

## **INTERFACCIA SERIALE ASCII – MODBUS**

### **NOTE E DEFINIZIONI COMUNI AI DUE PROTOCOLLI**

Questa relazione intende descrivere il funzionamento dell'interfaccia seriale degli strumenti DC50 (1/16DIN=48x48mm) e DC500 (1/8DIN=96x48mm). Verranno descritti i protocolli e le sequenze di caratteri necessarie al colloquio con lo strumento. Nella trattazione si chiamerà "terminale" l'apparecchio intelligente collegato allo strumento, che potrà essere, ad esempio, un personal computer, un pocket PC, un terminale, un PLC o una qualunque macchina in grado di gestire le comunicazioni via seriale.

*NOTA BENE: Da qui in avanti ci riferiremo al DC500 sottintendendo anche lo strumento DC50. Eventuali differenze saranno di volta in volta espressamente specificate. Si noti che le diciture relative ai nomi di famiglie, gruppi o parametri, e gli mnemonici di alcuni dei parametri stessi sono riportati per esteso così come appaiono sullo strumento DC500. Per il DC50 tali diciture risultano ovviamente abbreviate, ma comunque facilmente identificabili.*

Le interfacce seriali disponibili sono due (mutualmente esclusive):

- lo standard V.24 (RS232C)
- lo standard RS485.

Per strumenti tipo DC50 l'interfaccia RS232 consente di realizzare un collegamento punto-punto mentre per strumenti tipo DC500 l'interfaccia RS232 consente di realizzare un collegamento a catena di più strumenti, in quanto il segnale proveniente dal terminale è bufferizzato e ritrasmesso su un canale di uscita (che potrà quindi essere collegato ad un altro strumento), mentre il segnale diretto verso il terminale si inserisce sulla linea eventualmente collegata ad altri strumenti (l'interfaccia è progettata in modo da evitare conflitti hardware di trasmissione). Sebbene lo standard RS232C e l'interfaccia stessa permettano la trasmissione full-duplex, la comunicazione prevede una risposta dello strumento solo dopo l'interpretazione di una richiesta da parte del terminale; è quindi evidente che la trasmissione sarà in pratica di tipo half-duplex. L'interfaccia RS232 sarà operativa con protocollo ASCII oppure MODBUS (mutualmente esclusivi) in accordo alla sigla di ordinazione dell'apparecchio.

L'interfaccia RS485 consente anch'essa il collegamento di più strumenti con il terminale, ma con il vantaggio di usare un semplice doppino per tale collegamento. Per questo tipo di interfaccia, però, il terminale dovrà gestire correttamente la commutazione trasmissione-ricezione, in quanto il canale RS485 così realizzato è, per definizione, di tipo half-duplex. Il ritardo minimo di commutazione tra ricezione e trasmissione degli strumenti è impostabile da 1 a 10 ms. L'interfaccia RS485 sarà operativa con protocollo ASCII oppure MODBUS (mutualmente esclusivi) in accordo alla sigla di ordinazione dell'apparecchio.

### **NOTE PER IL COLLEGAMENTO HARDWARE DEL DC50x AD UN PERSONAL COMPUTER**

Si tenga presente che lo strumento suppone di comunicare con il terminale senza usare nessun tipo di handshake software o hardware: per cui sarà necessario cortocircuitare il segnale RTS con CTS ed il segnale DTR con DSR dell'interfaccia seriale del Personal Computer, per evitare che alcuni sistemi operativi blocchino la comunicazione in attesa dei segnali CTS o DSR, che normalmente arrivano in risposta dai dispositivi collegati.

# Interfaccia Seriale ASCII

## PARAMETRI DI CONFIGURAZIONE

Nella famiglia **ConFiG** si trova il gruppo di parametri che impostano e regolano la comunicazione seriale, indicato con '**rS CoM**'; i parametri disponibili sono:

<b>Protoc</b>	<i>OFF</i> : Interfaccia seriale disabilitata ASCII: Interfaccia seriale abilitata con protocollo di comunicazione software 'ASCII'
<b>Addr</b>	1 - 255: Numero di identificazione dello strumento
<b>bAud</b>	Velocità di trasmissione: 300, 600, 1200, 2400, 4800, 9600 baud
<b>ModE</b>	<i>LocAL</i> : La tastiera dello strumento è completamente operativa mentre il terminale può solo leggere dati e parametri, ma non può modificarli (fatta eccezione per i parametri <b>kEyLok</b> del gruppo <b>diSPL</b> e <b>ModE</b> del gruppo <b>rS CoM</b> ) <i>rEMotE</i> : La tastiera dello strumento permette di vedere i parametri, ma non ne permette la modifica (fatta eccezione per i parametri <b>kEyLok</b> del gruppo <b>diSPL</b> e <b>ModE</b> del gruppo <b>rS CoM</b> ) mentre il terminale può leggere e modificare tutti i parametri.
<b>dELAY</b>	Ritardo alla risposta: 1, 2, 3, 4, 5, 6, 8, 10 ms

## PROTOCOLLO DI COMUNICAZIONE HARDWARE

La trasmissione è di tipo seriale asincrona half-duplex (lo strumento che sta ricevendo non trasmette, mentre se sta trasmettendo non riceve dati), con 1 bit di start, 8 bits per carattere, nessun bit di parità ed 1 bit di stop; la velocità della comunicazione può essere impostata cambiando un parametro dello strumento e può essere di 300, 600, 1200, 2400, 4800 o 9600 baud. Il protocollo di comunicazione è lo standard RS232 o lo standard RS485 a seconda della scheda installata nello strumento.

## RICHIESTA DI UN PARAMETRO DA PARTE DEL TERMINALE

La sequenza di caratteri da mandare allo strumento è la seguente:

- 1) carattere di start (ascii STX, 2 dec, 0x02 hex)
- 2) primo carattere identificativo strumento (tradotto in esadecimale)
- 3) secondo carattere identificativo strumento (tradotto in esadecimale)
- 4) comando di lettura (ascii 'R', 82 dec, hex. 0x52 hex)
- 5) primo carattere identificativo della locazione da leggere (tradotto in esadecimale)
- 6) secondo carattere identificativo della locazione da leggere (tradotto in esadecimale)
- 7) carattere di stop (ascii ETX, 3 dec, 0x03 hex)
- 8) carattere di controllo, calcolato come XOR di tutti i precedenti caratteri

Esempio: supponiamo che lo strumento abbia come indirizzo 123 dec, e che si voglia leggere il parametro 'SEt' dell'allarme 1.

L'indirizzo 123 dec corrisponde, in esadecimale, al valore 0x7B; inoltre dalla tabella locazioni si vede che il parametro 'SEt' dell'allarme 1 è accessibile alla locazione esadecimale 0x4B.

Per cui i caratteri da mandare saranno:

ascii	dec.	hex
-----		
STX	2	02
'7'	55	37
'B'	66	42
'R'	82	52
'4'	52	34
'B'	66	42
ETX	3	03
	80	50

### RISPOSTA DELLO STRUMENTO AD UNA RICHIESTA DI UN PARAMETRO

Supponendo, ad esempio, che il valore 'Set 1' richiesto sia 184542 dec, lo strumento risponderà con la sequenza di caratteri:

- 1) carattere di start (ascii STX, 2 dec, 0x02 hex)
- 2) segno, che sarà il carattere '+' per valori positivi (come in questo caso) o '-' per valori negativi
- 3, 4, 5, 6, 7, 8) cifre che compongono il valore desiderato, in questo caso '184542'
- 9) carattere di stop (ascii ETX, 3 dec, 0x03 hex)
- 10) carattere di controllo, calcolato come XOR di tutti i precedenti caratteri

I caratteri che lo strumento manderà saranno quindi:

ascii	dec.	hex
-----		
STX	2	02
'+'	43	2B
'1'	49	31
'8'	56	38
'4'	52	34
'5'	53	35
'4'	52	34
'2'	50	32
ETX	3	03
	36	24

**Nota per DC50:** Se lo strumento è un DC50, le prime due cifre del valore resteranno sempre a 0. Ad esempio, se il valore che lo strumento deve trasmettere è pari a 1845, lo strumento manterrà il formato a 6 cifre (per la compatibilità col DC500), ma con le prime 2 forzate a 0. Quindi si avrà la sequenza di caratteri STX, '+', '0', '0', '1', '8', '4', '5' ETX e Check Value calcolato come per il DC500, cioè XOR di tutti i caratteri precedenti (compresi i due caratteri forzati a '0').

## RICHIESTA DI SETTAGGIO DI UN PARAMETRO DA PARTE DEL TERMINALE

La sequenza di caratteri da mandare allo strumento è la seguente:

- 1) carattere di start (ascii STX, 2 dec, 0x02 hex)
- 2) primo carattere identificativo strumento (tradotto in esadecimale)
- 3) secondo carattere identificativo strumento (tradotto in esadecimale)
- 4) comando di scrittura (ascii 'W', 87 dec, hex. 0x57 hex)
- 5) primo carattere identificativo della locazione da scrivere (tradotto in esadecimale)
- 6) secondo carattere identificativo della locazione da scrivere (tradotto in esadecimale)
- 7) carattere ascii '=' (61 dec, 0x3d hex)
- 8) segno, che sarà il carattere '+' per valori positivi o '-' per valori negativi
- 9, 10, 11, 12, 13, 14) cifre che compongono il valore desiderato
- 15) carattere di stop (ascii ETX, 3 dec, 0x03 hex)
- 16) carattere di controllo, calcolato come XOR di tutti i precedenti caratteri

Esempio: supponiamo che lo strumento abbia come indirizzo 14 dec, e che si voglia settare il parametro 'SEt Lo' dell'allarme 4 con il valore -125025.

L'indirizzo 14 dec corrisponde, in esadecimale, al valore 0x0E; inoltre dalla tabella 1 si vede che il parametro 'SEt Lo' dell'allarme 4 è accessibile alla locazione 139 dec, che in esadecimale diventa 0x8B.

Per cui i caratteri da mandare saranno:

ascii	dec.	hex
-----		
STX	2	02
'0'	48	30
'E'	69	45
'W'	87	57
'8'	53	35
'B'	51	33
'='	61	3D
'-'	45	2D
'1'	49	31
'2'	50	32
'5'	53	35
'0'	48	30
'2'	52	32
'5'	53	35
ETX	3	03
	52	34

**Nota per DC50:** Se lo strumento è un DC50, le prime due cifre del valore resteranno sempre a 0. Ad esempio, se il valore che si vuole memorizzare è -1250, lo strumento manterrà il formato a 6 cifre (per la compatibilità col DC500), ma con le prime 2 forzate a 0. Quindi si avrà la sequenza di caratteri STX, '-', '0', '0', '1', '2', '5', '0' ETX e Check Value calcolato come per il DC500, cioè XOR di tutti i caratteri precedenti (compresi i due caratteri forzati a '0').

## RISPOSTA DELLO STRUMENTO AD UNA RICHIESTA DI SETTAGGIO DI UN PARAMETRO

Lo strumento risponderà con la sequenza di caratteri:

- 1) carattere di start (ascii STX, 2 dec, 0x02 hex)
- 2) carattere 'E' (69 dec, 0x45 hex)
- 3, 4) carattere '0' (48 dec, 0x30 hex)
- 5) codice errore, che potrà essere '0', '1', '2', e così via, secondo la 'tabella errori' sotto riportata
- 6) carattere di stop (ascii ETX, 3 dec, 0x03 hex)
- 7) carattere di controllo, calcolato come XOR di tutti i precedenti caratteri

Se non ci sono impedimenti alla memorizzazione del parametro specificato con il valore desiderato (cioè se il parametro 'ModE' del gruppo 'rS CoM' è impostato su 'rEMotE', il parametro non è protetto dalla chiave strumento ed il valore è all'interno dei limiti consentiti) lo strumento risponderà con la sequenza di caratteri

ascii dec. hex

ascii	dec.	hex
STX	2	02
'E'	69	45
'0'	48	30
'0'	48	30
'0'	48	30
ETX	3	03
	116	74

## TABELLA ERRORI

Lo strumento può rispondere con codici di errore diversi da 0; più precisamente si potranno avere i seguenti errori:

'E000'	Nessun errore: scrittura eseguita correttamente
'E001'	Comando non riconosciuto
'E002'	Errore in scrittura: valore del parametro fuori dai limiti permessi
'E003'	Errore in scrittura: il parametro in questione è protetto in scrittura
'E004'	Errore in lettura: il parametro in questione è protetto in lettura

## INTERPRETAZIONE DELLE RISPOSTE

Molti dei parametri dello strumento non sono numeri puri, per cui il numero che viene trasmesso dallo strumento in risposta all'interrogazione del terminale va interpretato secondo le sigle mnemoniche che gli danno un significato preciso.

I valori possibili dei parametri mnemonici ed i loro corrispondenti significati sono riportati nelle tabelle 1, 2, ..., 48.

# Interfaccia Seriale ModBus

## PARAMETRI DI CONFIGURAZIONE

Nella famiglia **ConFIG** si trova il gruppo di parametri che impostano e regolano la comunicazione seriale, indicato con '**rS CoM**'; i parametri disponibili sono:

<b>Protoc</b>	<i>OFF</i> :	Interfaccia seriale disabilitata
	<i>Modbus</i> :	Interfaccia seriale abilitata con protocollo ModBus RTU
<b>Addr</b>	<i>1 - 255</i> :	Numero di identificazione dello strumento
<b>bAud</b>		Velocità di trasmissione: 300, 600, 1200, 2400, 4800, 9600 baud
<b>ModE</b>	<i>LocAL</i> :	La tastiera dello strumento è completamente operativa mentre il terminale può solo leggere dati e parametri, ma non può modificarli (fatta eccezione per i parametri <b>kEyLok</b> del gruppo <b>diSPL</b> e <b>ModE</b> del gruppo <b>rS CoM</b> )
	<i>rEMotE</i> :	La tastiera dello strumento permette di vedere i parametri, ma non ne permette la modifica (fatta eccezione per i parametri <b>kEyLok</b> del gruppo <b>diSPL</b> e <b>ModE</b> del gruppo <b>rS CoM</b> ) mentre il terminale può leggere e modificare tutti i parametri.
<b>dELAy</b>		Ritardo alla risposta: 1, 2, 3, 4, 5, 6, 8, 10 ms

## PROTOCOLLO DI COMUNICAZIONE HARDWARE

E' stato preso come riferimento il protocollo MODBUS standard Modicon in versione RTU (Remote Terminal Unit). La trasmissione è di tipo seriale asincrona half-duplex (lo strumento che sta ricevendo non trasmette, mentre se sta trasmettendo non riceve dati), con 1 bit di start, 8 bits per carattere, nessun bit di parità ed 1 bit di stop; la velocità della comunicazione può essere impostata cambiando un parametro dello strumento e può essere di 300, 600, 1200, 2400, 4800 o 9600 baud. Il protocollo di comunicazione è lo standard RS232 o lo standard RS485 a seconda della scheda installata nello strumento.

## PROTOCOLLO DI COMUNICAZIONE SOFTWARE

Con riferimento al protocollo MODBUS standard Modicon RTU le funzioni riconosciute sono la 3, la 4 e la 6 (N.B. la 3 e la 4 vengono interpretate come se fossero lo stesso comando, cioè *words reading*; la 6 è la funzione *single word writing*).

Ci sono due differenze tra il protocollo implementato sui DC50x ed il protocollo standard:

- 1) non è stata prevista la funzione di *broadcasting*, cioè non è previsto che un comando mandato all'indirizzo 0 venga riconosciuto da tutti gli strumenti in rete, indipendentemente dal parametro **Addr** impostato nei vari strumenti;
- 2) il numero di words che si possono richiedere con i comandi 3 e 4 è limitato a 1; cioè ogni operazione di lettura è relativa ad un solo parametro;
- 3) se il terminale manda un codice funzione diverso da quelli riconosciuti (3, 4 e 6), lo strumento manda un error reply con codice di errore 1.

Tutti i dati vengono trasmessi in modo binario.

La condizione di START è riconosciuta quando il ritardo tra due caratteri consecutivi trasmessi supera 1 T.U. (Time Unit = tempo necessario a trasmettere 1 carattere, es. 9600baud: 10bit/car / 9600bit/s = 1.04ms/car).

Una richiesta del terminale ha sempre il seguente formato:

- 1° byte: indirizzo strumento (per selezionare, fra molti, lo strumento desiderato)
- 2° byte: codice funzione (3, 4 o 6)
- 3° byte: byte più significativo della word da leggere o da scrivere
- 4° byte: byte meno significativo della word da leggere o da scrivere
- 5° byte: lettura: parte alta del numero di words da leggere;  
scrittura: parte alta del valore da scrivere
- 6° byte: lettura: parte bassa del numero di words da leggere;  
scrittura: parte bassa del valore da scrivere
- 7° byte: parte bassa del CRC-16
- 8° byte: parte alta del CRC-16.

Il calcolo del CRC-16 bit viene effettuato secondo le specifiche del MODBUS.

I valori dei parametri verranno sempre espressi sotto forma binaria a 32 bit con segno: ciò vuol dire che si potranno avere valori compresi tra  $-2.147.483.648$  (0x80000000) e  $+2.147.483.647$  (0x7FFFFFFF). Bisognerà perciò porre attenzione al byte alto trasmesso, perché se il suo valore supera 127 (7FHex), il valore finale del parametro sarà negativo.

### Funzione '3' o '4': words reading

La risposta dello strumento alla funzione di lettura consisterà in un frame di 9 bytes, il cui significato è:

- 1° byte: indirizzo strumento
- 2° byte: codice funzione (3 o 4)
- 3° byte: numero di byte per i dati trasmessi (pari a 4N, dove N è il numero di words rich.)
- 4°, 5°, 6°, 7°: valore della word richiesta (dal byte più alto a quello più basso rispettivamente)
- 8°, 9°: CRC-16 (byte basso e alto rispettivamente)

### Funzione 6: single word writing

La risposta dello strumento alla funzione di scrittura consisterà in un frame di 10 bytes:

- 1° byte: indirizzo strumento
- 2° byte: codice funzione
- 3°, 4° byte: indirizzo word da scrivere (byte alto e byte basso rispettivamente)
- 5°, 6°, 7°, 8°: valore scritto (dal byte più alto a quello più basso rispettivamente)
- 9°, 10° byte: CRC-16 (byte basso e byte alto rispettivamente)

### Errore in risposta - error reply

Se il comando non può andare a buon fine, viene ritornato al terminale un frame di 5 bytes che specificano che tipo di errore si è verificato:

- 1° byte: indirizzo strumento
- 2° byte: codice funzione con il bit più significativo settato (codice funzione + 80Hex)
- 3° byte: codice errore
- 4°, 5° byte: CRC-16 (byte basso e byte alto rispettivamente)

## CODICI ERRORI

Valore Errore associato

- 1 Codice comando non riconosciuto
- 2 Indirizzo illegale
- 3 Valore illegale
- 9 Numero di dati richiesti illegale
- 10 Dato protetto in scrittura

## Esempi

### Esempio di lettura

Si supponga di voler leggere il parametro 'PrESEt' del contatore 1 da uno strumento che ha per indirizzo seriale '04'. Il parametro desiderato ha per indirizzo 1020Hex.

L'indirizzo di partenza delle letture è quindi 0x1020, il numero di words da leggere forzato a 1.

La sequenza di bytes da spedire allo strumento è:

```
id   com  < address > < words > < CRC >
0x04 0x03 0x10 0x20 0x00 0x01 0x81 0x55
```

La risposta dello strumento sarà:

```
id   com  bytes < msb  DATO   lsb > < CRC >
0x04 0x03 0x04 0x00 0x00 0x01 0xF4 0xAF 0x24
```

Quindi, per questo esempio, il PrESEt del contatore 1 è 0x000001F4Hex, cioè 500dec.

### Esempio di scrittura

L'unico comando di scrittura è il comando "0x06", che è il comando di scrittura per una singola word. Non è prevista la scrittura di più parametri con un solo comando.

Si supponga di voler scrivere lo stesso parametro letto prima (cioè il PrESEt del contatore 1, locazione 0x1020) con il valore di 1000dec (cioè 0x000003E8Hex) sullo stesso strumento di prima (id = 4). La sequenza di bytes da spedire allo strumento è:

```
id   com  < address > < msb  DATO   lsb > < CRC >
0x04 0x06 0x10 0x20 0x00 0x00 0x03 0xe8 0xA4 0x11
```

La risposta dello strumento, nel caso dell'avvenuta scrittura, sarà identica alla richiesta fatta, cioè:

```
id   com  < address > < msb  DATO   lsb > < CRC >
0x04 0x06 0x10 0x20 0x00 0x00 0x03 0xe8 0xA4 0x11
```

Nel caso in cui, invece, i parametri di controllo scrittura (LocAL / rEMotE o kEyLok), oppure il valore del parametro stesso non rientri nel range permesso, lo strumento risponderà con un messaggio di errore di tipo "error reply".

### Consigli di carattere generale

- in caso di problemi di risposta o di comportamento è spesso utile riprovare da una situazione certa di partenza e quindi caricare i parametri di default ecc. ecc.

- evitare nel modo più assoluto le situazioni di conflitto o di eventuale incertezza operativa. In linea di massima lo strumento gestisce autonomamente queste situazioni ma non sempre il comportamento (es. priorità e/o precedenze) segue una logica facilmente identificabile e tantomeno facilmente descrivibile.

- il progetto di riferimento che supporta questa famiglia di strumenti è costantemente sottoposto ad aggiunte, miglioramenti e revisioni. E' nostra primaria cura tenere regolarmente aggiornata la documentazione relativa che può essere oggetto di modifiche, anche importanti, senza preavviso.

## Tabella locazioni lettura/scrittura parametri dello strumento

Famiglia	Gruppo	Parametro	Ascii Lettura	Ascii Scrittura	MdBus	Significato	
InPut	InP 1	tArGEt	R00	W00	0x1000	Tab. 1	
		ModE	R01	W01	0x1001	Tab. 2	
		EdGE	R02	W02	0x1002	Tab. 3	
		dEbou	R03	W03	0x1003	Tab. 4	
		t1	R04	W04	0x1004	numerico	
		t2	R05	W05	0x1005	numerico	
	toGGLE	R06	W06	0x1006	Tab. 5		
	InP 2	tArGEt	R07	W07	0x1007	Tab. 1	
		ModE	R08	W08	0x1008	Tab. 2	
		PoL3-4	R09	W09	0x1009	Tab. 3	
		dEbou	R0A	W0A	0x100A	Tab. 4	
		t1	R0B	W0B	0x100B	numerico	
		t2	R0C	W0C	0x100C	numerico	
	toGGLE	R0D	W0D	0x100D	Tab. 5		
	InP 3	tArGEt	R10	W10	0x1010	Tab. 1	
		ModE	R11	W11	0x1011	Tab. 2	
		PoL3-4	R12	W12	0x1012	Tab. 3	
		dEbou	R13	W13	0x1013	Tab. 4	
		t1	R14	W14	0x1014	numerico	
		t2	R15	W15	0x1015	numerico	
	toGGLE	R16	W16	0x1016	Tab. 5		
	InP 4	tArGEt	R17	W17	0x1017	Tab. 1	
		ModE	R18	W18	0x1018	Tab. 2	
		EdGE	R19	W19	0x1019	Tab. 3	
		dEbou	R1A	W1A	0x101A	Tab. 4	
		t1	R1B	W1B	0x101B	numerico	
		t2	R1C	W1C	0x101C	numerico	
	toGGLE	R1D	W1D	0x101D	Tab. 5		
	Count	Cnt 1	PrESEt	R20	W20	0x1020	numerico
			OFFSEt	R21	W21	0x1021	numerico
			OvEr	R22	W22	0x1022	numerico
			UndEr	R23	W23	0x1023	numerico
			MuPLy	R24	W24	0x1024	numerico
			divid	R25	W25	0x1025	numerico
			dP PoS	R26	W26	0x1026	Tab. 6
			rSt In	R27	W27	0x1027	Tab. 7
HLd In			R28	W28	0x1028	Tab. 8	
rAnGE			R29	W29	0x1029	Tab. 9	
Funct			R2A	W2A	0x102A	Tab. 10	
PowEr		R2B	W2B	0x102B	Tab. 11		
Cnt 2		PrESEt	R30	W30	0x1030	numerico	
		OFFSEt	R31	W31	0x1031	numerico	
		OvEr	R32	W32	0x1032	numerico	
		UndEr	R33	W33	0x1033	numerico	
		MuPLy	R34	W34	0x1034	numerico	
		divid	R35	W35	0x1035	numerico	
		dP PoS	R36	W36	0x1036	Tab. 6	
		rSt In	R37	W37	0x1037	Tab. 7	
		HLd In	R38	W38	0x1038	Tab. 8	
		rAnGE	R39	W39	0x1039	Tab. 9	
		Funct	R3A	W3A	0x103A	Tab. 10	
		PowEr	R3B	W3B	0x103B	Tab. 11	

ALArM	ALrM 1	tyPE	R40	W40	0x1040	Tab. 12
		SourcE	R41	W41	0x1041	Tab. 13
		Inhib	R42	W42	0x1042	Tab. 14
		Funct	R43	W43	0x1043	Tab. 15
		rEFEr	R44	W44	0x1044	Tab. 16
		rESEt	R45	W45	0x1045	Tab. 17
		EdGE	R46	W46	0x1046	Tab. 18
		StoP	R47	W47	0x1047	Tab. 19
		bAck	R48	W48	0x1048	Tab. 20
		On dLy	R49	W49	0x1049	numerico
		OFFdLy	R4A	W4a	0x104A	numerico
		SEt	R4B	W4b	0x104B	numerico
		Hy Hi	R4C	W4C	0x104C	numerico
		Hy Lo	R4D	W4D	0x104D	numerico
		SEt Hi	R4E	W4E	0x104E	numerico
		SEt Lo	R4F	W4F	0x104F	numerico
		LoGOut	R50	W50	0x1050	Tab. 21
		PhiOut	R51	W51	0x1051	Tab. 22
			ALrM 2	tyPE	R54	W54
	SourcE	R55		W55	0x1055	Tab. 13
	Inhib	R56		W56	0x1056	Tab. 14
	Funct	R57		W57	0x1057	Tab. 15
	rEFEr	R58		W58	0x1058	Tab. 16
	rESEt	R59		W59	0x1059	Tab. 17
	EdGE	R5A		W5A	0x105A	Tab. 18
	StoP	R5B		W5B	0x105B	Tab. 19
	bAck	R5C		W5C	0x105C	Tab. 20
	On dLy	R5D		W5D	0x105D	numerico
	OFFdLy	R5E		W5E	0x105E	numerico
	SEt	R5F		W5F	0x105F	numerico
	Hy Hi	R60		W60	0x1060	numerico
	Hy Lo	R61		W61	0x1061	numerico
	SEt Hi	R62		W62	0x1062	numerico
	SEt Lo	R63		W63	0x1063	numerico
	LoGOut	R64		W64	0x1064	Tab. 21
	PhiOut	R65		W65	0x1065	Tab. 22
	ALrM 3	tyPE		R68	W68	0x1068
		SourcE	R69	W69	0x1069	Tab. 13
		Inhib	R6A	W6A	0x106A	Tab. 14
		Funct	R6B	W6B	0x106B	Tab. 15
		rEFEr	R6C	W6C	0x106C	Tab. 16
		rESEt	R6D	W6D	0x106D	Tab. 17
		EdGE	R6E	W6E	0x106E	Tab. 18
		StoP	R6F	W6F	0x106F	Tab. 19
		bAck	R70	W70	0x1070	Tab. 20
		On dLy	R71	W71	0x1071	numerico
		OFFdLy	R72	W72	0x1072	numerico
		SEt	R73	W73	0x1073	numerico
		Hy Hi	R74	W74	0x1074	numerico
		Hy Lo	R75	W75	0x1075	numerico
		SEt Hi	R76	W76	0x1076	numerico
		SEt Lo	R77	W77	0x1077	numerico
		LoGOut	R78	W78	0x1078	Tab. 21
		PhiOut	R79	W79	0x1079	Tab. 22

ALrM 4	tyPE	R7C	W7C	0x107C	Tab. 12	
	SourcE	R7D	W7D	0x107D	Tab. 13	
	Inhib	R7E	W7E	0x107E	Tab. 14	
	Funct	R7F	W7F	0x107F	Tab. 15	
	rEFEr	R80	W80	0x1080	Tab. 16	
	rESEt	R81	W81	0x1081	Tab. 17	
	EdGE	R82	W82	0x1082	Tab. 18	
	StoP	R83	W83	0x1083	Tab. 19	
	bAck	R84	W84	0x1084	Tab. 20	
	On dLy	R85	W85	0x1085	numerico	
	OFFdLy	R86	W86	0x1086	numerico	
	SEt	R87	W87	0x1087	numerico	
	Hy Hi	R88	W88	0x1088	numerico	
	Hy Lo	R89	W89	0x1089	numerico	
	SEt Hi	R8A	W8A	0x108A	numerico	
	SEt Lo	R8B	W8B	0x108B	numerico	
	LoGOut	R8C	W8C	0x108C	Tab. 21	
	PhiOut	R8D	W8D	0x108D	Tab. 22	
ConFiG	USEr	ScEnE	R90	W90	0x1090	Tab. 23
		tbASE	R91	W91	0x1091	Tab. 24
		tiMEr	R92	W92	0x1092	Tab. 25 (DC500)
		tGAtE	R93	W93	0x1093	numerico
diSPL	kEyLok	R94	W94	0x1094	Tab. 26	
	tiMout	R95	W95	0x1095	Tab. 27	
	kEy o	R96	W96	0x1096	Tab. 28	
	kEy x	R97	W97	0x1097	Tab. 29	
	kbSEnS	R98	W98	0x1098	Tab. 30	
	StorE	R99	W99	0x1099	Tab. 31	
	HidE	R9A	W9A	0x109A	Tab. 32	
	Cnt dS	R9B	W9B	0x109B	Tab. 33	
	rAtEdS	R9C	W9C	0x109C	Tab. 34	
	-ModE	R9D	W9D	0x109D	Tab. 35	
	FilTEr	R9E	W9E	0x109E	Tab. 36	
	n AvG	R9F	W9F	0x109F	numerico	
	2.diS	RAD	WAD	0x10AD	Tab. 41	
(DC50)	rS CoM	Protoc	RA0	WA0	0x10A0	Tab. 37
		Addr	RA1	WA1	0x10A1	numerico
		bAud	RA2	WA2	0x10A2	Tab. 38
		ModE	RA3	WA3	0x10A3	Tab. 39
		dELAy	RA4	WA4	0x10A4	Tab. 40
An Out	SourcE	RA8	WA8	0x10A8	Tab. 41	
	diSPLo	RA9	WA9	0x10A9	numerico	
	Out Lo	RAA	WAA	0x10AA	numerico	
	diSPHi	RAB	WAB	0x10AB	numerico	
	Out Hi	RAC	WAC	0x10AC	numerico	

## Tabella locazioni lettura/scrittura variabili operative DC500

Variabile	Ascii Lettura	Ascii Scrittura	ModBus	Significato
Situazione relé	RFF	WFF	0x20FF	Tab. 42
Situazione Leds	RFE	WFE	0x20FE	Tab. 43
Situazione display unità	RFD	WFD	0x20FD	Tab. 44
Situazione display decine	RFC	WFC	0x20FC	Tab. 44
Situazione display centinaia	RFB	WFB	0x20FB	Tab. 44
Situazione display migliaia	RFA	WFA	0x20FA	Tab. 44
Situazione display decine di migliaia	RF9	WF9	0x20F9	Tab. 44
Situazione display centinaia di migliaia	RF8	WF8	0x20F8	Tab. 44
Contatore 1	RF7	read only	0x20F7	numerico
Contatore 2	RF6	read only	0x20F6	numerico
Stato degli ingressi	RF5	read only	0x20F5	Tab. 45
Flags situazione strumento	RF4	read only	0x20F4	Tab. 46
Situazione allarmi	RF3	read only	0x20F3	Tab. 47
Situazione tasti	RF2	read only	0x20F2	Tab. 48
Caricamento parametri di Default	write only	WF0	0x20F0	

## Tabella locazioni lettura/scrittura variabili operative DC50

Variabile	Ascii Lettura	Ascii Scrittura	ModBus	Significato
Situazione relé	RFF	WFF	0x20FF	Tab. 42
Situazione Leds	RFE	WFE	0x20FE	Tab. 43
Situazione display unità superiori	RFD	WFD	0x20FD	Tab. 44
Situazione display decine sup.	RFC	WFC	0x20FC	Tab. 44
Situazione display centinaia sup.	RFB	WFB	0x20FB	Tab. 44
Situazione display migliaia sup.	RFA	WFA	0x20FA	Tab. 44
Situazione display unità inferiori	RF9	WF9	0x20F9	Tab. 44
Situazione display decine inf.	RF8	WF8	0x20F8	Tab. 44
Situazione display centinaia inf.	REF	WEF	0x20EF	Tab. 44
Situazione display migliaia inf.	REE	WEE	0x20EE	Tab. 44
Contatore 1	RF7	read only	0x20F7	numerico
Contatore 2	RF6	read only	0x20F6	numerico
Stato degli ingressi	RF5	read only	0x20F5	Tab. 45
Flags situazione strumento	RF4	read only	0x20F4	Tab. 46
Situazione allarmi	RF3	read only	0x20F3	Tab. 47
Situazione tasti	RF2	read only	0x20F2	Tab. 48
Caricamento parametri di Default	write only	WF0	0x20F0	

**NB** Scrivendo la locazione 0xFF (situazione relé) si forzano i relé per un tempo di 2 secondi. Scaduto questo tempo lo strumento riassume il controllo dei relé.

Scrivendo una qualsiasi delle locazioni dei displays si forza l'indicazione del display interessato per un tempo di 2 secondi. Scaduto questo tempo lo strumento riassume il controllo del display interessato.

E' quindi evidente che se si vuole mantenere il controllo delle risorse per un lungo periodo di tempo, il terminale dovrà mandare i comandi appropriati prima dello scadere del tempo di timeout (refresh di scrittura).

Per caricare i parametri di default mandare il comando "W80=+000001"; verranno aggiornati tutti i parametri ad eccezione del gruppo rSCoM, in cui verrà solamente forzato "ModE" su "LocAL".

Per riarmare un relé in blocco (in attesa del reset manuale) basta forzare a 0 il bit di blocco del relé interessato (locazione RFF, bits 4, 5, 6, 7).

**Tabella 1**

<i>Valore</i>	<i>Significato</i>
0	OFF
1	Cnt 1
2	Cnt 2
3	Cnt 12

**Tabella 2**

0	Add
1	Sub
2	Ph
3	Ph2
4	Ph4
5	dir
6	rESEt
7	PrESEt
8	Inhib
9	HoLd

**Tabella 3**

0	riSE
1	FALL
2	both

**Tabella 4**

0	FrEE
1	1000
2	100
3	10
4	1
5	t1-t2

**Tabella 5**

0	diSAb
1	EnAbL

**Tabella 6**

0	999999
1	99999.9
2	9999.99
3	999.999
4	99.9999
5	9.99999

**Tabella 7**

0	diSAb
1	InSt
2	HoLd

**Tabella 8**

Vedi tabella 7

**Tabella 9**

0	PoSnEG
1	PoS

**Tabella 10**

<i>Valore</i>	<i>Significato</i>
0	SiNGL
1	MuLti

**Tabella 11**

0	rESEt
1	PrESEt
2	MEMo

**Tabella 12**

0	OFF
1	ALrHi
2	ALrLo
3	bAnd

**Tabella 13**

0	Cnt 1
1	Cnt 2

**Tabella 14**

0	OFF
1	On

**Tabella 15**

0	PoS
1	nEG

**Tabella 16**

0	AbSLt
1	SEt 1
2	SEt 2
3	SEt 3
4	SEt 4

**Tabella 17**

0	AutoM
1	MAnuAL

**Tabella 18**

0	riSE
1	FALL

**Tabella 19**

0	no
1	yES

**Tabella 20**

<i>Valore</i>	<i>Significato</i>
	Vedi tabella 19

**Tabella 21**

0	OFF
1	rESet1
2	rESet2
3	PrESt1
4	PrESt2

**Tabella 22**

0	OFF
1	Out 1
2	Out 2
3	Out 3
4	Out 4

**Tabella 23**

0	ctUPdn
1	ctUPUP
2	ctdndn
3	quotUP
4	quotdn
5	FrEq
6	PEriod
7	tAchoM
8	tMr UP
9	tMr dn

**Tabella 24**

0	rEciPr
1	0.01
2	0.1
3	1
4	10

**Tabella 25**

0	tbASE
1	H-M-S

**Tabella 26**

0	OFF
1	Lo
2	Hi

**Tabella 27**

0	30
1	20
2	10
3	5

**Tabella 28**

<i>Valore</i>	<i>Significato</i>	(tasto "o")
0	diSAB	
1	HoLd	
2	Inhbit	
3	rESEt	
4	PrESEt	
5	rStALr	
6	rStAL1	
7	rStAL2	
8	rStAL3	
9	rStAL4	

**Tabella 29**

Vedi tabella 28 (tasto "x" solo per DC500)

**Tabella 30**

0	rELEAS
1	PrESS

**Tabella 31**

0	OFF
1	On

**Tabella 32**

0	OFF
1	On

**Tabella 33**

0	OFF
1	On

**Tabella 34**

0	0.01
1	0.1
2	1

**Tabella 35**

0	-199999
1	LEd

**Tabella 36**

0	OFF
1	diSP
2	n AvG

**Tabella 37**

0	OFF
1	ASCII oppure Modbus

**Tabella 38**

Valore	Significato
0	9600
1	4800
2	2400
3	1200
4	600
5	300

**Tabella 39**

0	LocAL
1	rEMotE

**Tabella 40**

0	1
1	2
2	3
3	4
4	5
5	6
6	8
7	10

**Tabella 41**

0	OFF
1	Cnt 1
2	Cnt 2
3	SEt 1
4	SEt 2
5	SEt 3
6	SEt 4

**Tabella 42**

Il dato (ricevuto o trasmesso) va interpretato come insieme di bit:

bit 0	Rele 1: 0 -> Off, 1 -> On
bit 1	Rele 2: 0 -> Off, 1 -> On
bit 2	Rele 3: 0 -> Off, 1 -> On
bit 3	Rele 4: 0 -> Off, 1 -> On

**Tabella 43**

Il dato (ricevuto o trasmesso) va interpretato come insieme di bit:

bit 0	Led 'H':	0 -> On	1 -> Off
bit 1	Led '4':	0 -> On	1 -> Off
bit 2	Led 'P':	0 -> On	1 -> Off
bit 3	Led 'L':	0 -> On	1 -> Off
bit 4	Led '3':	0 -> On	1 -> Off
bit 5	Led '1':	0 -> On	1 -> Off
bit 6	Led '2':	0 -> On	1 -> Off
bit 7	Led 'LK':	0 -> On	1 -> Off

**Tabella 44**

Il valore in risposta (il byte meno significativo) va interpretato come insieme di bit; ogni bit corrisponde ad un segmento.

Se bit = 0 il segmento è spento, se bit = 1 il segmento è acceso.

bit 0	segmento "c"
bit 1	segmento "e"
bit 2	segmento "d"
bit 3	segmento "p" (punto decimale)
bit 4	segmento "a"
bit 5	segmento "f"
bit 6	segmento "b"
bit 7	segmento "g"

**Tabella 45**

Il dato ricevuto va interpretato come insieme di bit:

bit 0	input 1:	0 -> Inattivo	1 -> Attivo
bit 1	input 2:	0 -> Inattivo	1 -> Attivo
bit 2	input 3:	0 -> Inattivo	1 -> Attivo
bit 3	input 4:	0 -> Inattivo	1 -> Attivo

**Tabella 46**

Il dato ricevuto va interpretato come insieme di bit:

bit 0	wait1	1 -> uscita 1 in attesa d un reset manuale
bit 1	wait2	1 -> uscita 2 in attesa d un reset manuale
bit 2	wait3	1 -> uscita 3 in attesa d un reset manuale
bit 3	wait4	1 -> uscita 4 in attesa d un reset manuale
bit 4	inhib1	1 -> uscita 1 non ha oltrepassato la soglia
bit 5	inhib2	1 -> uscita 2 non ha oltrepassato la soglia
bit 6	inhib3	1 -> uscita 3 non ha oltrepassato la soglia
bit 7	inhib4	1 -> uscita 4 non ha oltrepassato la soglia

**Tabella 47**

Il dato ricevuto va interpretato come insieme di bit:

bit 0	outAct1	1 -> è in atto sequenza attivazione per out1
bit 1	outAct2	1 -> è in atto sequenza attivazione per out2
bit 2	outAct3	1 -> è in atto sequenza attivazione per out3
bit 3	outAct4	1 -> è in atto sequenza attivazione per out4

**Tabella 48**

Il dato ricevuto va interpretato come insieme di bit:

bit 0	Tasto "x" (solo per DC500)
bit 1	Tasto INC
bit 2	Tasto DEC/CIFRA
bit 3	Tasto ENTER
bit 4	Tasto "o"

## Esempio di programma "C"

```

/*
21 giu 2006, esempio.c - rev. 21 giu 06

Testato il 21-06-06 con Pentium 100 Mhz,
O. S. MS-DOS Ver 6.20

Scritto per BORLAND C++ Version 1.00

Lo strumento deve avere settati i parametri seguenti
con i valori specificati:
'kEyLoK' (gruppo 'diSPL', famiglia 'ConFiG') -> 'OFF'
'Protoc' (gruppo 'rS CoM', famiglia 'ConFiG') -> 'ASCII'
'Addr' (gruppo 'rS CoM', famiglia 'ConFiG') -> '000001'
'bAud' (gruppo 'rS CoM', famiglia 'ConFiG') -> '9600'
'ModE' (gruppo 'rS CoM', famiglia 'ConFiG') -> 'rEMotE'

A titolo di esempio, verrà scritto il parametro 'SEt'
del gruppo 'ALrM 1', famiglia 'ALArM'
con il valore +184542.
Successivamente verrà letto il parametro 'SEt Hi'
del gruppo 'ALrM 1', famiglia 'ALArM'

La porta seriale COM1 verrà settata con:
velocità di comunicazione di 9600 baud,
8 bits per carattere, 1 bit di stop, nessuna parità.
*/

#include <stdio.h>
#include <bios.h>

void main()
{
    int caratteri_rx;
    unsigned char risposta[10];

    bioscom(0x00, 0xE3, 0x00); // inizializza porta seriale

    // trasmissione 1° comando:
    scrive parametro 'SEt' (comando W4B)
    del gruppo 'ALrM 1', famiglia 'ALArM' con il
    valore +184542

    bioscom(1, 2, 0);
    bioscom(1, '0', 0);
    bioscom(1, '1', 0);
    bioscom(1, 'W', 0);
    bioscom(1, '4', 0);
    bioscom(1, 'B', 0);
    bioscom(1, '=', 0);
    bioscom(1, '+', 0);
    bioscom(1, '1', 0);
    bioscom(1, '8', 0);
    bioscom(1, '4', 0);
    bioscom(1, '5', 0);
    bioscom(1, '4', 0);
    bioscom(1, '2', 0);
    bioscom(1, 3, 0);
    bioscom(1, 2 ^ '0' ^ '1' ^ 'W' ^ '4' ^ 'B' ^ '=' ^ '+' ^ '1' ^ '8'
        ^ '4' ^ '5' ^ '4' ^ '2' ^ 3, 0);

    for (caratteri_rx = 0; caratteri_rx < 7; caratteri_rx++)
        risposta[caratteri_rx] = bioscom(2, 0, 0);

    // stampa l'esito dell'operazione
    printf("Esito del comando W4B=+184542: ");
    if (caratteri_rx == 7) {
        for (caratteri_rx=1; caratteri_rx <=4; caratteri_rx++)
            printf("%c", risposta[caratteri_rx]);
    } else
        printf("Errore nella ricezione seriale");
    printf("\n");

    delay(1000); // ritardo 1 s

    // pulisce il buffer di ricezione
    bioscom(2, 0, 0);

    // trasmissione 2° comando: legge parametro 'SEt Hi'
    (comando R4E)
    del gruppo 'ALrM 1', famiglia 'ALArM'

    bioscom(1, 2, 0);
    bioscom(1, '0', 0);
    bioscom(1, '1', 0);
    bioscom(1, 'R', 0);
    bioscom(1, '4', 0);
    bioscom(1, 'E', 0);
    bioscom(1, 3, 0);
    bioscom(1, 2 ^ '0' ^ '1' ^ 'R' ^ '4' ^ 'E' ^ 3, 0);

    for (caratteri_rx = 0; caratteri_rx < 10; caratteri_rx++)
        risposta[caratteri_rx] = bioscom(2, 0, 0);

    printf("\nValore letto: %c%c%c%c%c%c%c\n",
        risposta[1],
        risposta[2],
        risposta[3],
        risposta[4],
        risposta[5],
        risposta[6],
        risposta[7] );
}

```

## Esempio di programma "QBASIC"

```

REM 21 giu 2006, comunica.bas - rev. 21 giu 06

REM Scritto per MSDOS QBASIC Version 1.1 di MSDOS.
REM Lo strumento deve avere settati i parametri seguenti
    con i valori specificati:
REM 'kEyLoK' (gruppo 'diSPL', famiglia 'ConFiG') -> 'OFF'
REM 'Protoc' (gruppo 'rS CoM', famiglia 'ConFiG') -> 'ASCII'
REM 'Addr' (gruppo 'rS CoM', famiglia 'ConFiG') -> '000001'
REM 'bAud' (gruppo 'rS CoM', famiglia 'ConFiG') -> '9600'
REM 'ModE' (gruppo 'rS CoM', famiglia 'ConFiG') -> 'rEMotE'

REM A titolo di esempio, verrà scritto il parametro 'SEt'
REM del gruppo 'ALrM 1', famiglia 'ALArM'
REM con il valore +184542.
REM Successivamente verrà letto il parametro 'SEt Hi'
REM del gruppo 'ALrM 1', famiglia 'ALArM'

REM La porta seriale COM1 verrà settata con una velocità di
    comunicazione di 9600 baud, 8 bits per carattere,
    1 bit di stop, nessuna parità.

10 DIM risposta(10)
20 CLS

100 OPEN "com1: 9600,n,8,1,cd,cs,ds,rs"
    FOR RANDOM AS #1

REM trasmissione 1ø comando:
    scrive parametro 'SEt' (comando W4B)
    del gruppo 'ALrM 1', famiglia 'ALArM'
    con il valore +184542

200 PRINT #1, CHR$(2);
205 PRINT #1, "0";
210 PRINT #1, "1";
215 PRINT #1, "W";
220 PRINT #1, "4";
225 PRINT #1, "B";
230 PRINT #1, "=";
235 PRINT #1, "+";
240 PRINT #1, "1";
245 PRINT #1, "8";
250 PRINT #1, "4";
255 PRINT #1, "5";
260 PRINT #1, "4";
262 PRINT #1, "2";
265 PRINT #1, CHR$(3);
270 PRINT #1, CHR$(2 XOR ASC("0") XOR ASC("1") XOR
    ASC("W") XOR ASC("4") XOR ASC("B") XOR ASC("=")
    XOR ASC("+") XOR ASC("1") XOR ASC("8") XOR
    ASC("4") XOR ASC("5") XOR ASC("4") XOR ASC("2")
    XOR 3);

300 FOR caratteri = 1 TO 7
310 risposta(caratteri) = ASC(INPUT$(1, 1))
320 NEXT caratteri

350 REM Stampa l'esito
355 PRINT "Esito comando W4B=+184542: ";
360 FOR caratteri = 2 TO 5
365 PRINT CHR$(risposta(caratteri));
370 NEXT caratteri

400 REM Pausa di 1 secondo
410 SLEEP (1)

420 PRINT
430 PRINT

REM trasmissione 2ø comando:
    legge parametro 'SEt Hi' (comando R4E)
    del gruppo 'ALrM 1', famiglia 'ALArM'

500 PRINT #1, CHR$(2);
510 PRINT #1, "0";
520 PRINT #1, "1";
530 PRINT #1, "R";
540 PRINT #1, "4";
550 PRINT #1, "E";
560 PRINT #1, CHR$(3);
570 PRINT #1, CHR$(2 XOR ASC("0") XOR
    ASC("1") XOR ASC("R") XOR ASC("4")
    XOR ASC("E") XOR 3);

600 FOR caratteri = 1 TO 10
610 risposta(caratteri) = ASC(INPUT$(1, 1))
620 NEXT caratteri

630 REM Stampa l'esito
635 PRINT "Esito comando R4E: ";
640 FOR caratteri = 2 TO 8
650 PRINT CHR$(risposta(caratteri));
660 NEXT caratteri

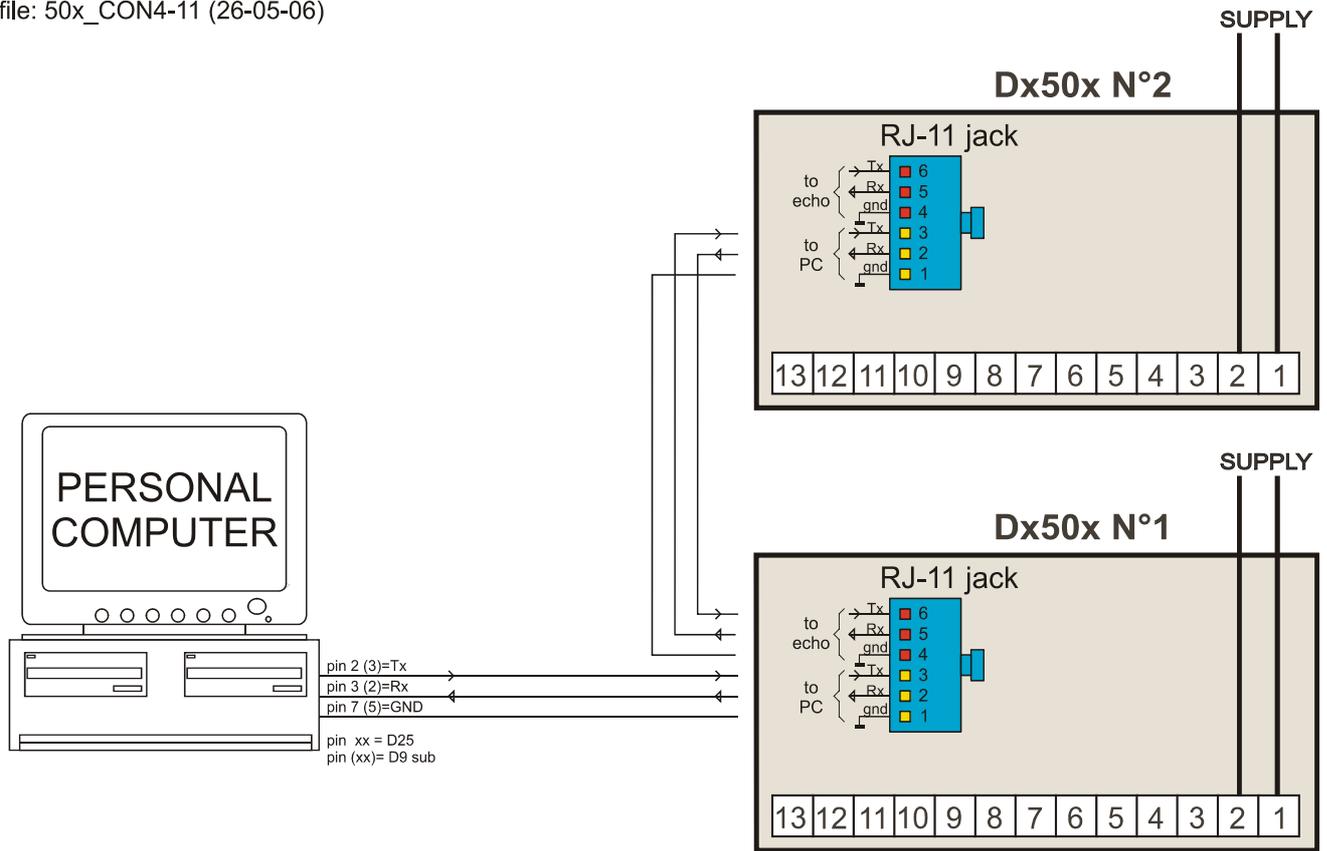
700 CLOSE (1)

800 END

```

# RS 232 INTERFACE CONNECTION and MULTI DROP EXAMPLE

file: 50x\_CON4-11 (26-05-06)



# RS 485 INTERFACE CONNECTION and MULTI DROP EXAMPLE

