

# Manuale

**Protocollo di comunicazione per DA 480-R su bus 485 (firmware 2.35)**  
(v. 2.0 – febbraio 2008)

## Introduzione

Il bus 485 è composto da un cavo a due conduttori intrecciati al quale sono connesse un certo numero di unità remote; una di queste unità (master), ha il compito di controllare tutte le altre ed è l'unica autorizzata ad occupare/utilizzare il bus, mentre tutte gli altri dispositivi connessi (slave) occupano il bus solo su esplicita richiesta del master, tramite un tipo di interrogazione che viene descritto in seguito.

Due resistenze di terminazione da 120 ohm devono essere poste ai due capi estremi della linea di connessione; tutti i dispositivi che si trovano in mezzo devono essere collegati direttamente alla linea.

Il collegamento corretto deve essere fatto unendo i due conduttori di arrivo con i due conduttori in partenza, direttamente sui morsetti. Connessioni eseguite parzialmente o completamente 'a stella' non garantiscono un corretto funzionamento del bus.

All'interno di ogni dispositivo è presente un driver che garantisce comunicazioni sicure anche con diverse centinaia di metri di cavo.

Il protocollo di comunicazione è a 9600 baud, 8 bit, 1 start bit, 1 stop bit, parità N.

Ogni comando inviato dal master (MA) ha in testa un identificativo del dispositivo (IDSCH) a cui è diretto il comando (IDCOM), seguito da un certo numero di byte costituenti il comando stesso.

Ad ogni comando ricevuto, lo slave (SL) risponde con un OK quando i dati sono corretti ed il comando è stato interpretato, oppure con un ER in caso di errori. Fanno eccezione i comandi riguardanti la richiesta di dati da parte del MA allo SL: per poter leggere dalle periferiche SL, esistono degli appositi comandi; in questo modo il MA ha la possibilità di interrogare in maniera ciclica (polling) il bus e ricevere molteplici informazioni da tutti gli SL connessi. Se necessario il polling può avvenire con estrema rapidità, in considerazione del fatto che una comunicazione normale richiesta/risposta può avere, in media, una durata di 10/15ms.

La stringa è così composta:

Start stringa	ID scheda	ID comando	Dato 1	Dato 2	Dato N	Checksum	End stringa
<b>STX</b>	<b>IDSCH</b>	<b>IDCOM</b>	<b>DATA1</b>	<b>DATA2</b>	<b>DATAN</b>	<b>CKS</b>	<b>ETX</b>
0x02	Ascii	Hex	Ascii	Ascii	Ascii	Ascii	0x03
1 byte	2 byte	1 byte	2 byte	2 byte	2 byte	2 byte	1 byte

STX (Start TX) e ETX (End TX) sono due campi che permettono di controllare la stringa e renderla dinamica (non a lunghezza di byte fissa): vengono gestiti con 1 singolo byte.

IDSCH è un campo che identifica l'unità a cui il comando è diretto: deve essere gestito in ASCII (due byte).

IDCOM è un campo che identifica il comando; viene gestito con 1 byte e deve essere sempre maggiore di 0x30.

Tutti gli altri campi di dati possono avere valore da 0x00 a 0xFF: vanno convertiti ed inviati in ASCII, ovvero vengono divisi in low nibble e high nibble e trasmessi con i relativi codici ASCII.

Esempi:

DATO = 0x36 = '3'+ '6' = 0x30 + 0x36

DATO = 0x4B = '4'+ 'B' = 0x34 + 0x42

DATO = 0xCF = 'C'+ 'F' = 0x43 + 0x46

Riferimenti tabella ASCII:

CHR	HEX	CHR	HEX	CHR	HEX	CHR	HEX
0	0x30	4	0x34	8	0x38	C	0x43
1	0x31	5	0x35	9	0x39	D	0x44
2	0x32	6	0x36	A	0x41	E	0x45
3	0x33	7	0x37	B	0x42	F	0x46
, (virgola)	0x2C	. (punto)	0x2E				

Viene utilizzata la conversione dei dati in ASCII in modo che tutti i codici inferiori a 0x30 sono da considerarsi codici di controllo stringa, compreso il tipo di comando.

### **Campo identificativo scheda IDSCH**

L'IDSCH è composto da 8 bit per cui è possibile ottenere 256 ID diversi da assegnare alle unità SL attraverso i dip-switch interni.

Per poter inviare a tutte le unità un comando di tipo Multicast, viene riservato l'ID 0x00: quando l'unità riceve questo ID esegue il comando corrispondente, senza però dare nessun tipo di risposta al MA ed impegnare quindi il bus. In questo modo, con un'unica stringa, è possibile inviare il medesimo comando a tutte le periferiche SL del bus

### **Campo comando IDCOM**

L>IDCOM è composto da 8 bit e per ogni SL si possono implementare  $(0xFF-0x30) = 207$  comandi diversi.

### **Campo controllo di errore (checksum) CKS**

Il checksum è un codice di controllo che viene ottenuto sommando tutti i byte della stringa a partire da IDSCH (compreso), fino a CKS (escluso). Come valore potrebbe anche assumere 0x02 o 0x03 che corrispondono ai codici di controllo stringa ed essere interpretato erroneamente: quindi anche il CKS deve essere gestito in ASCII. Nella somma non viene tenuto conto dell'overflow ed ha validità il byte meno significativo risultante dal calcolo.

Esempio di calcolo:

Start stringa	ID scheda	ID comando	Dato 1	Dato 2	Dato N	Checksum	End stringa
STX	IDSCH	IDCOM	DATA1	DATA2	DATAN	CKS	ETX
0x02	0x24	0x35	0x18	0x0D	0xC4	0xBF	0x03
0x02	0x32 0x34	0x35	0x31 0x38	0x00 0x44	0x43 0x34	0x42 0x46	0x03
1 byte	2 byte	1 byte	2 byte	2 byte	2 byte	2 byte	1 byte

### **Risposte dello slave dopo un comando ricevuto**

Dopo aver ricevuto un comando dal MA, lo SL dovrà rispondere entro un tempo di circa 10ms. Il timeout massimo previsto per la risposta è di 20ms. La situazione ottimale si ha con risposte entro 2 o 3ms in modo da mantenere i tempi totali di comunicazione piuttosto bassi: questo garantirebbe una buona velocità nel caso siano presenti sul bus molte unità e debbano essere controllate in modo continuo dal MA.

Lo SL deve sempre sincronizzarsi su STX (0x02) come primo byte del record e memorizzare i dati successivi in un buffer di lunghezza limitata (32/64 byte), in modo da poter facilmente individuare nella stringa errori di comunicazione e non rimanere bloccato. Qualsiasi tipo di risposta da parte dello SL deve avvenire solo dopo l'arrivo di un ETX.

Nel caso lo SL interpreti solo la parte iniziale del record, e verificata la corrispondenza dell' IDSCH, in presenza di errori nel resto del record dovrà rispondere con un codice di errore, ma sempre dopo l'arrivo di ETX. In tutti gli altri casi di mancata interpretazione della stringa, lo SL non risponde.

Le due risposte dello SL in caso di comunicazione corretta od errata sono:

Start stringa	ID scheda	Dato 1	Dato 2	End stringa	
STX	IDSCH	'O'	'K'	ETX	Comunicazione corretta senza errori
0x02	Ascii	0x4F	0x4B	0x03	
STX	IDSCH	'E'	'R'	ETX	Comunicazione errata
0x02	Ascii	0x45	0x52	0x03	
1 byte	2 byte	1 byte	1 byte	1 byte	

In queste stringhe di risposta non si rende necessaria la codifica in ASCII e non è presente il CKS in quanto i due caratteri 'OK' o 'ER' vengono verificati entrambi dal MA e questo è già garanzia di dati corretti.

Nel caso di risposta negativa o di mancanza di risposta, il MA può effettuare 4 tentativi di comunicazione, falliti i quali disconosce l'unità SL.

### 0x30 Comando richiesta di presenza dello slave

Questo comando è indispensabile quando si desidera implementare una procedura di auto-configurazione del sistema, in cui il MA effettua una ricerca (polling) di tutte le periferiche SL collegate al bus e che in seguito dovrà controllare.

Il comando dovrà essere riconosciuto in modo universale da tutti gli SL per cui IDCOM = 0x30.

La stringa sarà così composta:

Start stringa	ID scheda	ID comando	Checksum		End stringa
STX	IDSCH	IDCOM	CKS		ETX
0x02	0xXX	0x30	0xXY		0x03
0x02	0xXX	0x30	0x3X	0x3Y	0x03
1 byte	2 byte	1 byte	2 byte		1 byte

Il MA invierà quindi una sequenza di stringhe incrementando IDSCH a partire da 0x01 a 0xFF.

Quando uno SL è presente sul bus e riceve questo comando dovrà solo rispondere con una conferma (OK).

### 0x32 Comando di lettura dati da periferica slave

Questo comando permette la lettura di alcuni parametri dallo SL. E' un comando unificato per tutti gli SL ma le risposte dagli SL potranno essere diverse in base ai parametri da loro gestiti. Se i parametri da leggere sono molti e solo per alcuni è richiesta una lettura continua e ripetitiva, conviene suddividere il comando in due o più comandi distinti in base alle esigenze di gestione della periferica.

Il comando sarà così strutturato:

Start stringa	ID scheda	ID comando	Checksum		End stringa
STX	IDSCH	IDCOM	CKS		ETX
0x02	0xXX	0x32	0xXY		0x03
0x02	0xXX	0x32	0x3X	0x3Y	0x03
1 byte	2 byte	1 byte	2 byte		1 byte

## Comandi di impostazione e richiesta dati

Per il controllo del DA 480-R fare riferimento alla seguente tabella:

<b>IDSCH</b>	<b>0x--</b>	Possibilità di cambiare l'indirizzo da <b>0x01 a 0xFF</b> (0x00 per modalità multicast).
IDCOM	0x30	Richiesta di presenza slave.
IDCOM	0x31	Lettura dati: versione firmware.
IDCOM	0x32	Lettura dati: livelli volume impostati sui 4 canali, stato flag funzionamento volume (VCA/remote), mute.
IDCOM	0x33	Comando: impostazione flag di funzionamento volume (VCA/remote), mute.
IDCOM	0x34	Comando: impostazione livelli di volume per i 4 canali.
IDCOM	0x35	Lettura dati: valori dei 4 ingressi analogici VCA.
IDCOM	0x36	Lettura dati: stati signal-present e peak canali di ingresso, stati mute e stati fault canali di uscita.
IDCOM	0x37	Lettura dati: livello temperatura interna, stato velocità ventola, allarme temperatura massima e stato relè altoparlanti.
IDCOM	0x38	Lettura dati: stato uscite logiche open-collector e relè.
IDCOM	0x39	Comando: gestione relè altoparlanti.
IDCOM	0x3A	Comando: scrittura etichetta da 8 caratteri.
IDCOM	0x3B	Lettura dati: lettura etichetta da 8 caratteri.
IDCOM	0x3C	Lettura dati unificati: stati signal-present e peak canali di ingresso, stati mute e stati fault canali di uscita, livello temperatura interna, stato velocità ventola, allarme temperatura massima e stato relè altoparlanti.
IDCOM	0x3D	Lettura dati: livelli volume impostati da remoto sui 4 canali.
IDCOM	0x3E	Lettura dati: lettura valori timer vita in ore e power-on timer in minuti.
IDCOM	0xF0	Lettura dati: Service Serial Number SSN.
IDCOM	0xF1	Lettura dati: Factory Serial Number FSN.

### 0x31 Lettura versione firmware

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x31	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	CHAR1	CHAR2	CHAR3	CHAR4	CHAR5	CKS	ETX
0x02	Ascii (2)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (2)	0x03

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x31,0x39,0x32,0x03

*Risposta del dispositivo:* 0x02,0x30,0x34,0x30,0x32,0x2E,0x33,0x35,0x35,0x43,0x03

Unità con ID = 1, versione firmware 02.35.

### 0x32 Lettura livelli di volume impostati sui 4 canali e flags funzionamento (sia sotto controllo VCA che remote)

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x32	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	CFLAG1	VOL CH1	VOL CH2	VOL CH3	VOL CH4	CKS	ETX
0x02	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	0x03

CONTROL FLAGS 1	
<b>Bit 0</b> = CH1 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 4</b> = CH1 mute check 0= no mute 1= mute mode
<b>Bit 1</b> = CH2 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 5</b> = CH2 mute check 0= no mute 1= mute mode
<b>Bit 2</b> = CH3 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 6</b> = CH3 mute check 0= no mute 1= mute mode
<b>Bit 3</b> = CH4 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 7</b> = CH4 mute check 0= no mute 1= mute mode

Il byte CFLAG1 è il CONTROL FLAGS 1: in questo comando è gestito in lettura e consente di conoscere la modalità a cui il valore del volume si riferisce (VCA o remoto) e lo stato di mute del canale. I parametri VOL CHx indicano i reali valori di volume che l'amplificatore sta gestendo. Attenzione: i valori da 0 a 5 vengono interpretati da DA480R sempre come volume 0.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x32,0x39,0x33,0x03

*Risposta del dispositivo:*

0x02,0x30,0x31,0x30,0x46,0x35,0x43,0x38,0x35,0x38,0x37,0x36,0x31,0x39,0x32,0x03

Unità con ID = 1, VCA disabilitati, mute = 0, volumi: 35%,51%,52%,37%.

### 0x33 Comando impostazione flag di funzionamento volume – commutazione VCA/remote e mute

STX	IDSCH	IDCOM	CONTROL FLAGS 1	CKS	ETX
0x02	Ascii (2)	0x33	Ascii (2)	Ascii (2)	0x03

Con questo comando è possibile commutare la gestione del volume VCA/remoto e mettere in mute i canali.

CONTROL FLAGS 1	
<b>Bit 0</b> = CH1 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 4</b> = Set CH1 Mute mode 0= disable 1= enable
<b>Bit 1</b> = CH2 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 5</b> = Set CH2 Mute mode 0= disable 1= enable
<b>Bit 2</b> = CH3 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 6</b> = Set CH3 Mute mode 0= disable 1= enable
<b>Bit 3</b> = CH4 volume remote control via RS485 0= disable (VCA enable) 1= enable (VCA disable)	<b>Bit 7</b> = Set CH4 Mute mode 0= disable 1= enable

Attenzione: quando viene forzato un mute da remoto (bit 4-7), il DA480R entra in modalità MUTE sia quando il volume viene regolato da VCA sia quando viene regolato da remoto. Quando si esce dallo stato di MUTE (bit=0), il valore del volume riprodotto dal DA480R sarà quello precedentemente impostato dai VCA o da remoto: a tale scopo utilizzare i flags VCA/remoto (bit 0-3) ed i relativi livelli di volume per ottenere il funzionamento desiderato.

Questa impostazione viene memorizzata in modo permanente finché non viene inviato un comando di modifica dei flag: anche dopo un power-cycle i settaggi verranno mantenuti.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x33,0x30,0x46,0x30,0x41,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x4F,0x4B,0x03

Unità con ID = 1, disabilita i VCA ed il mute per i 4 canali.

### 0x34 Comando impostazione livelli di volume per i 4 canali

STX	IDSCH	IDCOM	VOL. CH1	VOL. CH2	VOL. CH3	VOL. CH4	CKS	ETX
0x02	Ascii (2)	0x34	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	0x03

Segue risposta OK o ER da parte dello SL.

Attenzione: i valori inviati che vanno da 0 a 5 vengono interpretati da DA480R sempre come volume 0 e viene restituito il valore 0 anche in caso di lettura tramite comando 0x32.

Questo settaggi vengono memorizzati in modo permanente finché non viene inviato un comando di modifica dei dati: anche dopo un power-cycle i settaggi verranno mantenuti.

Tipo di dato	Formato in cui viene gestito	Range	Valori di default	Modo di gestione
Regolazione volume	DEC	Da 0% a 100% Byte da 00 a FF	0%	256 posizioni compreso 0

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x34,0x39,0x34,0x35,0x45,0x41,0x33,0x34,0x30,0x35,0x34,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x4F,0x4B,0x03

Unità con ID = 1, volumi: 58%,37%,64%,25%.

### 0x35 Lettura valori dei 4 ingressi analogici VCA

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x35	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	VCA CH1	VCA CH2	VCA CH3	VCA CH4	CKS	ETX
0x02	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	0x03

I valori letti con questo comando non subiscono l'influenza del ponticello di inversione: quindi per valori di tensione 0-10V la corrispondenza è 0-100%.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x35,0x39,0x36,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x39,0x34,0x39,0x35,0x30,0x31,0x30,0x31,0x46,0x45,0x03

Unità con ID = 1, lettura volumi VCA: 57%,58%,0%,0%.

### 0x36 Lettura stati signal-present e peak canali di ingresso, stati mute e stati fault canali di uscita

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x36	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	STS INPUTS	STS MUTE & FAULTS	CKS	ETX
0x02	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	0x03

Per l'interpretazione dei flag vedere la seguente tabella:

STS INPUTS	STS MUTE & FAULTS
<b>Bit 0</b> = CH1 input signal present 0= no signal 1= signal present	<b>Bit 0</b> = CH1 mute check 0= no mute 1= mute mode
<b>Bit 1</b> = CH1 input peak 0= no peak 1= peak present	<b>Bit 1</b> = CH2 mute check 0= no mute 1= mute mode
<b>Bit 2</b> = CH2 input signal present 0= no signal 1= signal present	<b>Bit 2</b> = CH3 mute check 0= no muting 1= mute mode
<b>Bit 3</b> = CH2 input peak 0= no peak 1= peak present	<b>Bit 3</b> = CH4 mute check 0= no mute 1= mute mode
<b>Bit 4</b> = CH3 input signal present 0= no signal 1= signal present	<b>Bit 4</b> = CH1 fault check 0= no fault 1= fault occurrence
<b>Bit 5</b> = CH3 input peak 0= no peak 1= peak present	<b>Bit 5</b> = CH2 fault check 0= no fault 1= fault occurrence
<b>Bit 6</b> = CH4 input signal present 0= no signal 1= signal present	<b>Bit 6</b> = CH3 fault check 0= no fault 1= fault occurrence
<b>Bit 7</b> = CH4 input peak 0= no peak 1= peak present	<b>Bit 7</b> = CH4 fault check 0= no fault 1= fault occurrence

Le informazioni di mute mode sono le stesse che si possono avere anche con il comando 0x32.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x36,0x39,0x37,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x30,0x30,0x30,0x30,0x32,0x31,0x03

Unità con ID = 1, signal present, peak, mute e fault = 0 per tutti e 4 i canali.



**0x37 Lettura livello temperatura interna, stato velocità ventola, allarme temperatura massima e stato relè altoparlanti**

<b>STX</b>	<b>IDSCH</b>	<b>IDCOM</b>	<b>CKS</b>	<b>ETX</b>
0x02	Ascii (2)	0x37	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

<b>STX</b>	<b>IDSCH</b>	<b>TENS</b>	<b>UNITS</b>	<b>POINT</b>	<b>DEC</b>	<b>F &amp; OVT &amp; SR</b>	<b>CKS</b>	<b>ETX</b>
0x02	Ascii (2)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (2)	Ascii (2)	0x03

ATTENZIONE: il range di temperatura misurato va da 20° a 80°C.

Per l'interpretazione dei flag vedere la seguente tabella:

<b>FAN &amp; OVER-TEMPERATURE &amp; SPEAKERS RELAYS STATUS</b>	
<b>Bit 0</b> = fan speed 1 (low) 0= off 1= on	<b>Bit 4</b> = Speakers relay CH 1/2 status 0=disable (load disconn.) 1=enable (load connected)
<b>Bit 1</b> = fan speed 2 (mid) 0= off 1= on	<b>Bit 5</b> = Speakers relay CH 3/4 status 0=disable (load disconn.) 1=enable (load connected)
<b>Bit 2</b> = fan speed 3 (high) 0= off 1= on	<b>Bit 6</b> =
<b>Bit 3</b> = over-temperature flag 0= no alarm 1= over-temperature alarm	<b>Bit 7</b> =

I bit 4 e 5 si riferiscono allo stato dei relè posti tra le uscite dello stadio finale ed il carico (altoparlanti): conoscere lo stato di questi relè è importante per sapere quando il carico è connesso.

**Esempio:**

*Stringa comando:* 0x02,0x30,0x31,0x37,0x39,0x38,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x32,0x39,0x2e,0x35,0x33,0x30,0x39,0x32,0x03

Unità con ID = 1, il cui livello di temperatura è 29.5 °C , ventola spenta e relè di carico connessi per le due coppie di canali.

### 0x38 Lettura stato uscite logiche open-collector e relè

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x38	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	STS L&R OUTPUTS	CKS	ETX
0x02	Ascii (2)	Ascii (2)	Ascii (2)	0x03

STS LOGIC & RELAYS OUTPUTS	
<b>Bit 0</b> = Logic output TTL 1/2 0= disable (open) 1= enable (closed to GND)	<b>Bit 4</b> =
<b>Bit 1</b> = Logic output TTL 3/4 0= disable (open) 1= enable (closed to GND)	<b>Bit 5</b> =
<b>Bit 2</b> = Logic relay 1/2 (see bit 6 for inverted mode) 0=disable (C closed to NC) 1=enable (C closed to NO)	<b>Bit 6</b> = Logic relay 1/2 inverted mode (for safety use) 0= standard mode 1= inverted mode
<b>Bit 3</b> = Logic relay 3/4 (see bit 7 for inverted mode) 0=disable (C closed to NC) 1=enable (C closed to NO)	<b>Bit 7</b> = Logic relay 3/4 inverted mode (for safety use) 0= standard mode 1= inverted mode

I relè logici legano il loro funzionamento a 2 bit: il primo (bit 2 e 3) serve come abilitazione e disabilitazione mentre il secondo (bit 6 e 7) serve per invertirne lo stato in caso di impiego in modalità di sicurezza. In questo secondo caso la condizione di normale utilizzo sarà quella con il relè chiuso tra C e NO: sia in caso di malfunzionamento che in caso di mancanza di alimentazione, il relè si apre dando una segnalazione di allarme al sistema che supervisiona il DA480R.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x38,0x39,0x39,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x43,0x30,0x44,0x34,0x03

Unità con ID = 1, con le uscite ½ e ¾ aperte, i relay logici ½ e ¾ in modalità inversa (C closed to NC).

### 0x39 Comando gestione relè altoparlanti

STX	IDSCH	IDCOM	SPEAKERS RELAYS FLAGS	CKS	ETX
0x02	Ascii (2)	0x39	Ascii (2)	Ascii (2)	0x03

Con questo comando è possibile commutare e gestire i relè altoparlanti, per consentire il controllo da micro interno o da remoto. In controllo da remoto è possibile collegare e scollegare i carichi.

SPEAKERS RELAYS FLAGS	
<b>Bit 0</b> = Speak. relay CH1/2 remote control via RS485 0= disable (micro control) 1= enable (remote control)	<b>Bit 4</b> =
<b>Bit 1</b> = Speak. relay CH3/4 remote control via RS485 0= disable (micro control) 1= enable (remote control)	<b>Bit 5</b> =
<b>Bit 2</b> = Speak. relay CH1/2 (only in remote control) 0=disable (load disconn.) 1=enable (load connected)	<b>Bit 6</b> =
<b>Bit 3</b> = Speak. relay CH3/4 (only in remote control) 0=disable (load disconn.) 1=enable (load connected)	<b>Bit 7</b> =

Attenzione: nel caso di eventi di fault i relè verranno sganciati indipendentemente dal tipo di controllo abilitato nei flag sopra riportati. Al rientro dagli eventi di fault verrà ripristinato il funzionamento impostato dai flag.

Questa impostazione viene memorizzata in modo permanente finché non viene inviato un comando di modifica dei flag: anche dopo un power-cycle i settaggi verranno mantenuti.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x39,0x30,0x33,0x46,0x44,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x4F,0x4B,0x03

Unità con ID = 1, impostazione relè di carico: controllo remoto abilitato per le coppie di canali e relè disconnessi per entrambe le coppie.

### 0x3A Comando scrittura etichetta da 8 caratteri

STX	IDSCH	IDCOM	POS	LABEL	CKS	ETX
0x02	Ascii (2)	0x3A	Ascii (1)	Ascii (8)	Ascii (2)	0x03

Con questo comando è possibile salvare in memoria permanente 5 etichette composte da 8 caratteri. Il campo POS è la posizione (da 0 a 4) dell'etichetta che deve essere scritta e il campo LABEL sono gli 8 caratteri. Devono essere utilizzati solo caratteri Ascii e si raccomanda di utilizzare solo lettere (sia minuscole che maiuscole) e numeri.

Attenzione: la posizione 0 è quella relativa al nome del dispositivo DA480-R mentre le altre sono relative ai 4 canali.

Segue risposta OK o ER da parte dello SL: viene risposto ER anche nel caso in cui la posizione che si cerca di scrivere è scorretta.

#### Esempio:

*Stringa comando:*

0x02,0x30,0x31,0x3a,0x30,0x44,0x41,0x34,0x38,0x30,0x52,0x31,0x20,0x0,0x0,0x0,0x38,0x46,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x4F,0x4B,0x03

Unità con ID = 1, scrittura etichetta 0 (nome del dispositivo), stringa "DA480R1".

### 0x3B Comando lettura etichetta da 8 caratteri

STX	IDSCH	IDCOM	POS	CKS	ETX
0x02	Ascii (2)	0x3B	Ascii (1)	Ascii (2)	0x03

Con questo comando è possibile leggere dalla memoria permanente le 5 etichette composte da 8 caratteri. Il campo POS è la posizione (da 0 a 4) dell'etichetta che deve essere letta.

La stringa ricevuta ha il seguente formato:

STX	IDSCH	LABEL	CKS	ETX
0x02	Ascii (2)	Ascii (8)	Ascii (2)	0x03

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x3b,0x30,0x43,0x43,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x44,0x41,0x34,0x38,0x30,0x52,0x31,0x20,0x32,0x35,0x03

Unità con ID = 1, lettura etichetta 0 (nome del dispositivo), valore "DA480R1".

### 0x3C Lettura dati unificati: stati signal-present e peak canali di ingresso, stati mute e stati fault canali di uscita, livello temperatura interna, stato velocità ventola, allarme temperatura massima e stato relè altoparlanti

Questo è un comando riassuntivo dei due comandi 0x36 e 0x37 ed è utile nel momento in cui si debba velocizzare al massimo l'interrogazione di molti dispositivi.

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x3C	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	STS INPUTS	STS MUTE & FAULTS	TENS	UNITS	POINT	DEC	F & OVT & SR	CKS	ETX
0x02	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (1)	Ascii (2)	Ascii (2)	0x03

Per l'interpretazione dei flags vedere le corrispondenti tabelle sopra riportate.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x3c,0x39,0x44,0x03

*Risposta del dispositivo:*

0x02,0x30,0x31,0x30,0x30,0x30,0x35,0x34,0x31,0x2e,0x32,0x30,0x30,0x34,0x42,0x03

Unità con ID = 1, valori di stato degli ingressi = nessun segnale, canali 1 e 3 in mute, temperatura = 41,2 °C, velocità ventola = 0, stato relè di carico = disconnessi tutti i canali.

### 0x3D Lettura livelli di volume impostati da remoto sui 4 canali

Questo comando serve per poter conoscere i livelli di volume che sono stati impostati da remoto e memorizzati dall'unità.

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x3D	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	REM VOL CH1	REM VOL CH2	REM VOL CH3	REM VOL CH4	CKS	ETX
0x02	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	Ascii (2)	0x03

I parametri REM VOL CHx indicano i valori di volume che sono stati inviati da remoto all'amplificatore. Attenzione: i valori da 0 a 5 vengono interpretati dal DA480R sempre come volume = 0.

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x3d,0x39,0x45,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x39,0x34,0x35,0x45,0x41,0x33,0x34,0x30,0x32,0x30,0x03

Unità con ID = 1, valori dei canali 58%,37%,64%,25%.

### 0x3E Lettura valori timer vita in ore e power-on timer in minuti

Con questo comando è possibile conoscere i valori del timer vita e del power-on timer; tali valori vengono inviati dallo slave in formato BCD, in modo da poter essere facilmente interpretati.

Entrambe i timer possono arrivare al valore 983040 che corrisponde a 112 anni per il timer vita e 682 giorni per il power-on timer. Il timer vita viene memorizzato in eeprom per renderlo permanente. Il power-on timer, invece, riparte da 0 ad ogni accensione.

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0x3E	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	H5	H4	H3	H2	H1	H0	M5	M4	M3	M2	M1	M0	CKS	ETX
0x02	Ascii (2)	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii	Ascii (2)	0x03

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0x3e,0x39,0x46,0x03

*Risposta del dispositivo:*

0x02,0x31,0x30,0x30,0x30,0x30,0x35,0x32,0x30,0x30,0x30,0x30,0x39,0x31,0x42,0x03

Unità con ID = 1, timer vita = 000052 ore, timer accensione = 000091 min.

### 0xF0 Lettura del Service Serial Number SSN

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0xF0	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	SERVICE SERIAL NUMBER								CKS	ETX
0x02	Ascii (2)	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii (2)	0x03

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0xf0,0x35,0x31,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x37,0x35,0x30,0x30,0x31,0x37,0x35,0x20,0x45,0x41,0x03

Unità con ID = 1, SSN = 7500175.

### 0xF1 Lettura del Factory Serial Number FSN

STX	IDSCH	IDCOM	CKS	ETX
0x02	Ascii (2)	0xF1	Ascii (2)	0x03

La stringa ricevuta ha il seguente formato:

STX	IDSCH	FACTORY SERIAL NUMBER								CKS	ETX
0x02	Ascii (2)	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii1	Ascii (2)	0x03

#### Esempio:

*Stringa comando:* 0x02,0x30,0x31,0xf0,0x35,0x31,0x03

*Risposta del dispositivo:* 0x02,0x30,0x31,0x37,0x35,0x30,0x30,0x31,0x37,0x35,0x20,0x45,0x41,0x03

Unità con ID = 1, FSN = 7500175.