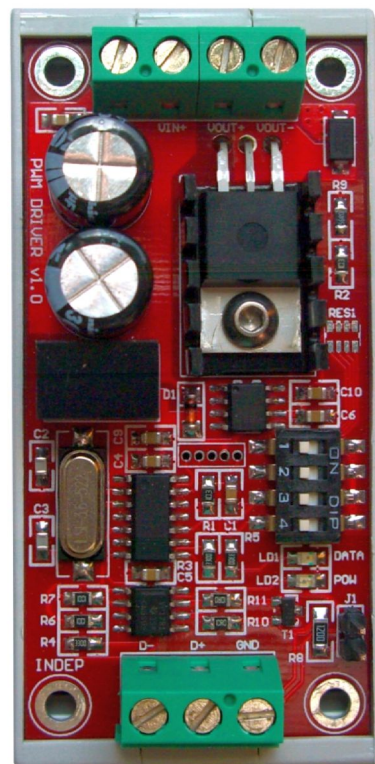


IDP-PWM1-DRIVER – Serie MINI DIN

CARATTERISTICHE:

- Canale PWM programmabile in frequenza e duty-cycle
- Elevata corrente: fino a 8 Ampere
- Ampia tensione di alimentazione: $7V_{dc} \sim 32V_{dc}$
- Fino a 16 moduli collegabili su bus seriale RS485
- Portata fino a 1200 metri
- Velocità di comunicazione seriale impostabile da comando
- Terminazione 120Ω On/Off integrata
- Semplici comandi in formato ASCII a stringa di caratteri
- Possibilità di controllo anche attraverso USB tramite modulo adattatore
- Salvataggio e ripristino automatico di tutte le impostazioni tramite EEPROM
- 240 bytes di memoria EEPROM programmabili dall'utente
- Contenitore per guida DIN



DESCRIZIONE:

Il dispositivo IDP-PWM1-DRIVER è un driver PWM di potenza con la possibilità di impostare sia la frequenza di oscillazione del segnale PWM (da 60Hz fino a 25KHz) sia il valore del duty-cycle (0-100%).

Il dimmer è controllabile tramite interfaccia RS485 con comandi in formato ASCII costituiti da semplici stringhe a caratteri.

Il collegamento tra HOST (tipicamente un PC o un PLC) e il/i modulo/i IDP-PWM1-DRIVER, si realizza sfruttando un bus RS485 a 2 fili (eventualmente con terzo filo di massa).

IDP-PWM1-DRIVER – Serie MINI DIN

E' possibile controllare il/i modulo/i anche attraverso una normale porta USB utilizzando un comune adattatore USB-RS485 o utilizzando il nostro adattatore USB485ISO, in grado di garantire elevate prestazioni ed elevata affidabilità grazie a un totale isolamento elettrico tra la porta USB e il bus seriale RS485.

I dispositivi IDP-PWM1-DRIVER, assieme ad un qualsiasi altro modulo della serie MINI-DIN, possono esser fra loro connessi con un semplice doppino filare per formare una rete che può raggiungere i 16 moduli sullo stesso Bus. In particolare ogni modulo IDP-PWM1-DRIVER è dotato di un DipSwitch a 4 levette che consente di configurare 16 diversi indirizzi.

I moduli possono essere forniti come semplice scheda o direttamente su contenitore plastico per guida DIN. Grazie al gancio per guida DIN, i moduli possono esser rapidamente bloccati su guida all'interno di armadi, rack o fissati su pareti o piani di lavoro.

La regolazione sui carichi avviene riportando sulle uscite la tensione di alimentazione in ingresso e modulandola in PWM (Pulse Width Modulation). In questo modo è possibile regolare la potenza applicata ai vari carichi in modo molto efficiente senza provocare surriscaldamento sul modulo di regolazione.

Il carico grazie anche alla programmabilità della frequenza del segnale PWM potrà essere di vario genere:

- Motori DC
- Ventole
- Riscaldatori
- Moduli Led

Come accennato in precedenza, il dispositivo dispone di 1 canale di uscita in modulazione PWM. Il canale di uscita PWM (VOUT+) è connesso alla tensione di alimentazione positiva +VIN. Il morsetto di uscita (-VOUT) viene chiuso a massa (-VIN) da un mosfet come mostrato in Figura1. La risoluzione su ogni canale pwm è di 256 livelli (0 = spegnimento, 255 = massima potenza): duty-cycle tra 0% e 100%.

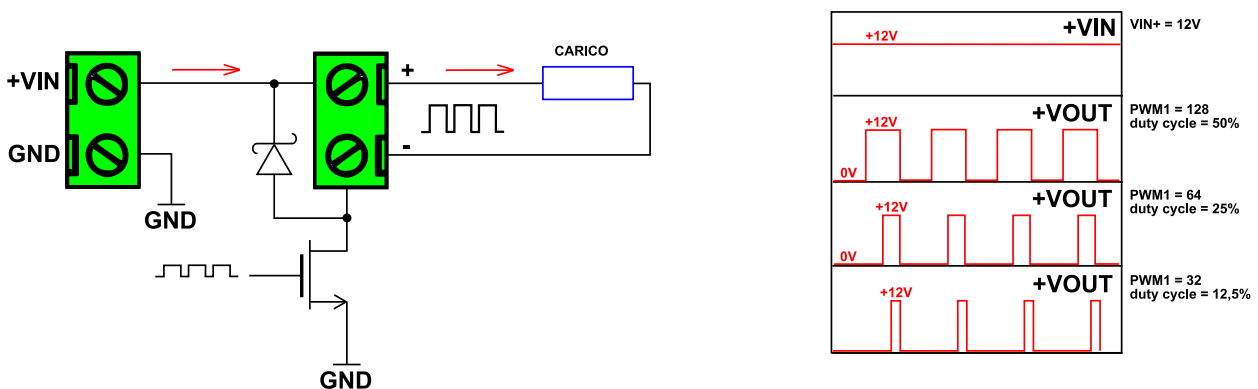


Figura1. Schema elettrico del canale PWM di uscita, esempio di forme d'onda in uscita.

IDP-PWM1-DRIVER – Serie MINI DIN

In Figura1 è possibile vedere come la tensione di alimentazione +VIN (nell'esempio +12V) venga applicata parzializzata su un canale di uscita tramite un mosfet che chiude il carico a massa (-VIN = GND).

In Figura2, di seguito, viene presentato uno schema di collegamento del Driver con vari possibili carichi.

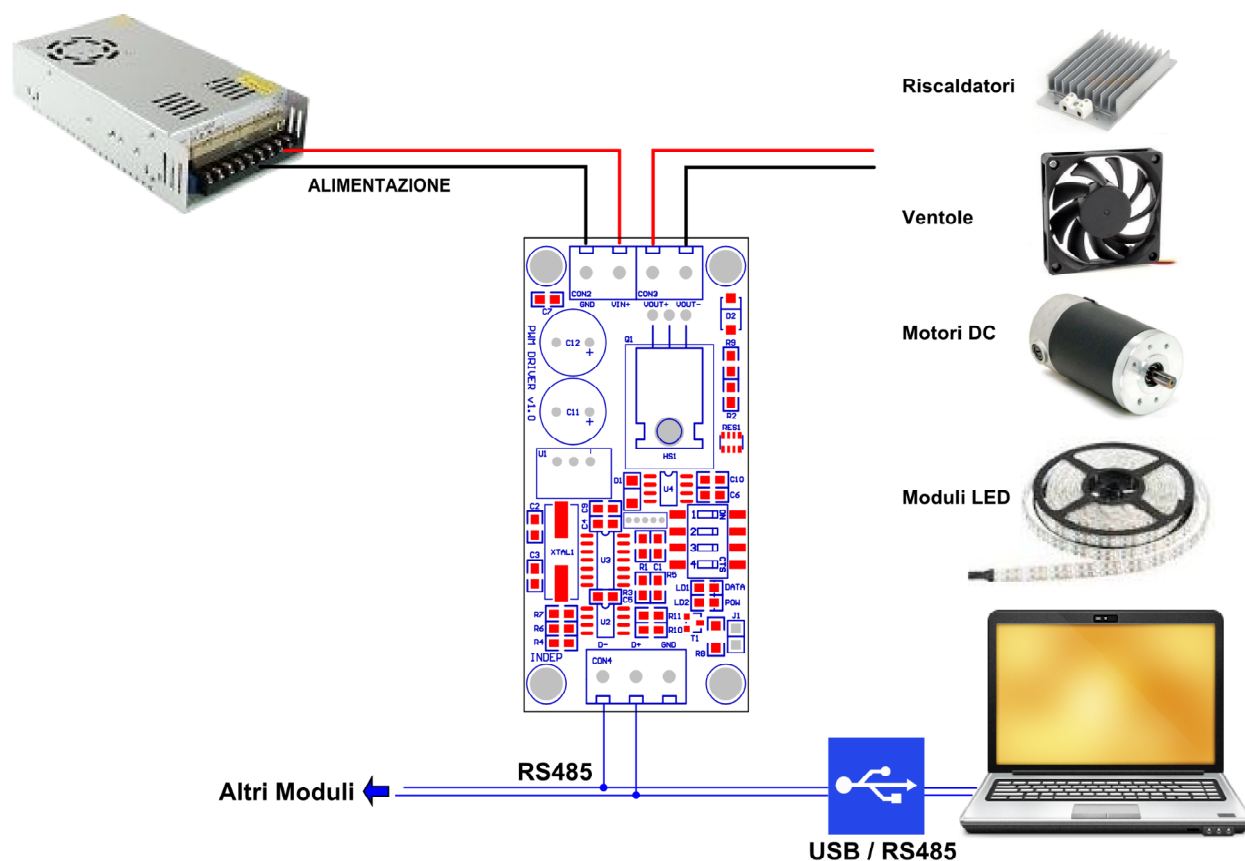
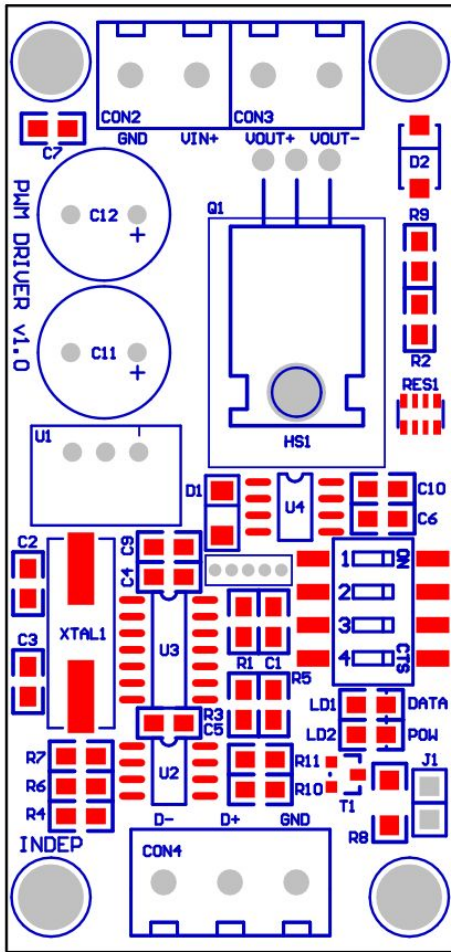


Figura2. Esempio di schema elettrico.

Nella Figura2 è possibile notare la presenza del Bus RS485 che consente di inserire fino a 16 diversi Driver sullo stesso doppino. Nello specifico, il controllo del modulo viene affidato ad un PC che comunica sul bus RS485 tramite un comune convertitore USB/RS485.

IDP-PWM1-DRIVER – Serie MINI DIN



Rif.	Descrizione	Note
GND/+VIN	Alimentazione 7V-32V	1
VOUT+	Uscita PWM Ch1, 8A Max	2
VOUT -	Uscita PWM Ch1, (GND)	3
J1	Jumper: Terminazione 120Ω	4
LD1	Dati sul Bus RS485	-
LD2	Presenza di alimentazione	-
SW1	Switch assegnazione indirizzo	5
D+ /D-	Bus comunicazione RS485	-
GND	Massa RS485	-

NOTE:

- 1) Su richiesta anche tensioni maggiori.
- 2) Connessa elettricamente a VIN+
- 3) Viene chiusa a GND dal MOSFET in modulazione PWM
- 4) Inserendo il ponticello J1 si chiude la linea differenziale RS485 su un resistore di terminazione da 120ohm.
- 5) DipSwitch per assegnare un indirizzo al modulo (default tutti ponticelli a 0). La codifica è Binaria. Le 4 levette sul DipSwitch consentono la definizione di 16 diversi indirizzi.

Figura3. Disposizione dei connettori, del dipswitch di indirizzo, del jumper di terminazione a 120 ohm e dei led di segnalazione.

CARATTERISTICHE ELETTRICHE:

Nella Tabella1, vengono riassunte la caratteristiche elettriche del modulo IDP-PWM1-DRIVER.

Parametri	Descrizione	Valore	Note
VIN	Tensione di alimentazione	7V _{dc} - 32V _{dc}	+VIN, GND
IIN	Corrente in ingresso (alim.)	8A (max)	+VIN, GND
IINo	Corrente in ingresso a vuoto	30mA (max)	+VIN, GND Nessun carico
IOUT1	Massima corrente in uscita CH1	8A (max)	VOUT+ = +VIN
F(pwm)	Frequenza modulazione pwm	60Hz – 25KHz	Impostabile a comando, 35 diversi valori di frequenza. Si veda il comando FROW
Duty-Cycle	Regolazione Luminosità	0%...100%	Il registro di controllo intensità si imposta tra 0 e 255
R(term)	Resistore di terminazione	120Ω	Linea RS485 (D+, D-)
T(amb)	Convezione in aria	-20°C +40°C	Temperatura ambiente

Tabella1. Parametri elettrici.

NOTE SUL BUS RS485:

Lo standard di comunicazione seriale RS485 è spesso usato in ambiente industriale per la semplicità di collegamento (solo due fili) e per l'alta affidabilità derivante da un'elevata immunità ai disturbi.

Il tipico schema di collegamento tra dispositivi è mostrato in Figura4. Si fa notare che la massima velocità di trasmissione è funzione del tipo di cavo e della lunghezza dello stesso.

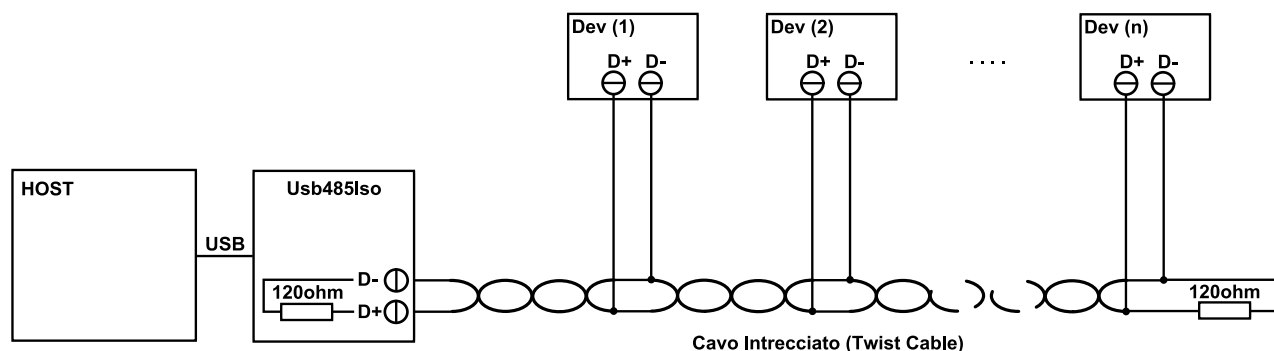


Figura4. Tipico schema di collegamento su bus RS485.

Per garantire le massime prestazioni (specie per tratte molto lunghe e con elevate velocità) usare cavo intrecciato con impedenza caratteristica attorno ai 120ohm ed abilitare la terminazione di linea sia sul lato Host (PC, PLC, ...) che alla fine della linea. E' altresì consigliata una schermatura del cavo (specie in ambienti particolarmente rumorosi). La terminazione alla fine della linea è facilmente ottenibile chiudendo il jumper di terminazione a 120 ohm presente sull'ultimo modulo IDP-PWM1-DRIVER inserito sul doppino del bus.

Per garantire massima affidabilità consigliamo il nostro modulo USB485ISO per collegare la porta USB di un PC con il bus RS485.

DIMENSIONI:

Di seguito vengono mostrate le dimensioni meccaniche del modulo. Come già accennato in precedenza il dispositivo può essere agganciato su una comune guida DIN.

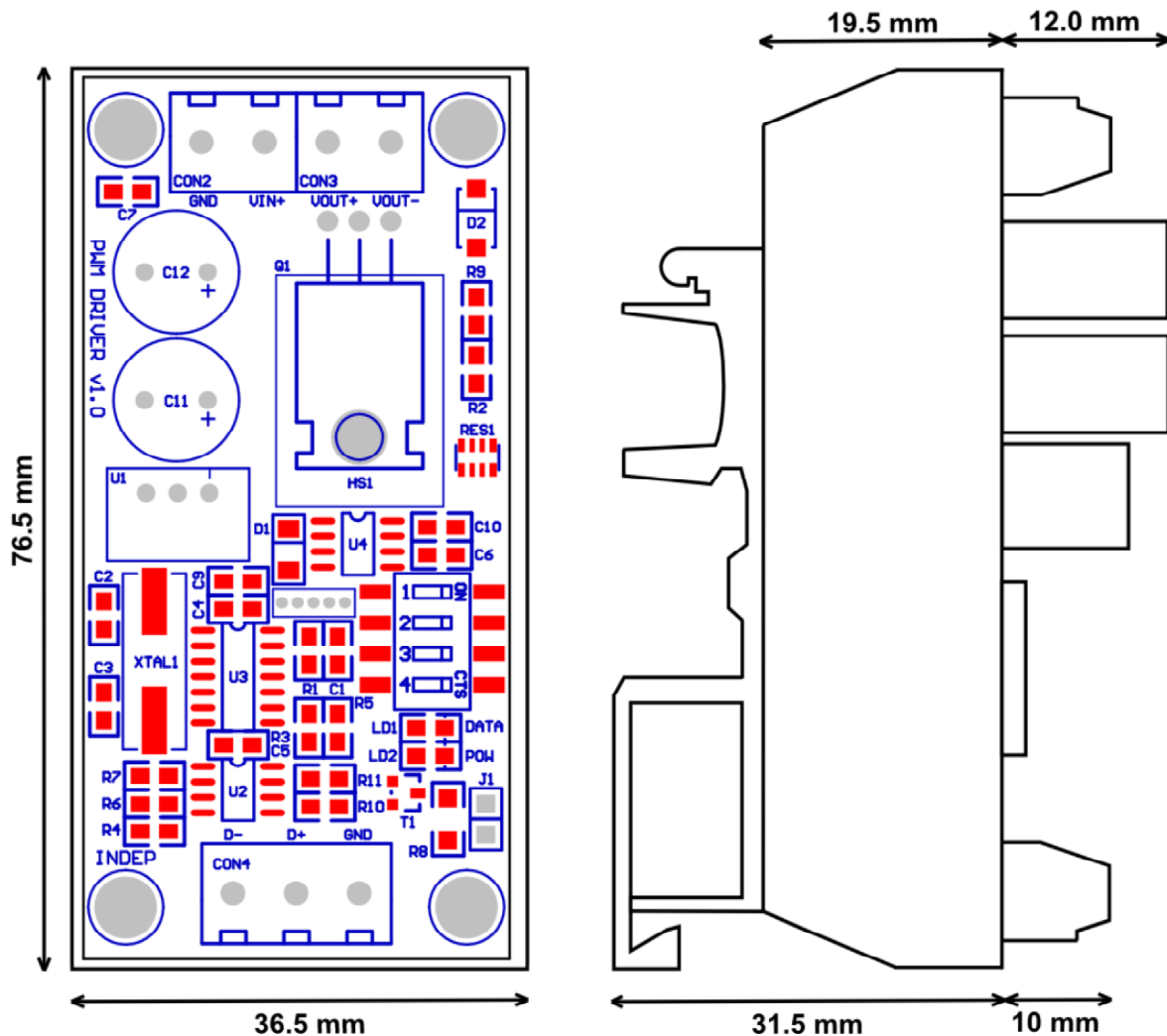


Figura5. Dimensioni del modulo IDP-PWM1-DRIVER montato sul contenitore per guida DIN.

L'ingombro totale della scheda, con il relativo contenitore a vaschetta, sono mostrate in Figura5.

Le dimensioni invece della sola scheda sono di 72.5mm x 34.5mm con un ingombro massimo in altezza di 18mm.

PROTOCOLLO DI COMUNICAZIONE:

Il modulo IDP-PWM1-DRIVER comunica serialmente verso un Host (tipicamente un PC o un PLC) tramite bus RS485. Il protocollo prevede che il master sia l'Host mentre tutti i dispositivi IDP-PWM1-DRIVER si comportano da slave. In pratica ogni modulo IDP-PWM1-DRIVER è in uno stato di ascolto e commuta in uno stato di trasmissione (pilotando il bus RS485) solo in risposta a specifici comandi provenienti dall'Host. In presenza di più dispositivi slave è fondamentale che ogni slave abbia un proprio indirizzo fisico in modo che il master (Host) possa scambiare informazioni in modo puntuale con ognuno di essi senza ambiguità.

BIN = 0000 DEC = 0 (default)	BIN = 0011 DEC = 3	BIN = 1000 DEC = 8	BIN = 1111 DEC = 15

Figura6. Esempi di indirizzo sul DipSwitch.

Per questa ragione ogni modulo IDP-PWM1-DRIVER è dotato di un DipSwitch a 4 levette che consente di assegnare ad ogni dispositivo un indirizzo sul bus. Per default le levette sono tutte OFF ovvero l'indirizzo binario è dato da 4 bit a zero (Indirizzo = 0). E' responsabilità dell'utente assegnare ad ogni dispositivo un indirizzo univoco. Alcuni esempi di indirizzo sul DipSwitch sono presentati in Figura6.

NOTE:

- Il protocollo di comunicazione prevede una configurazione con 8bit di dato, nessuna parità, un bit di start, un bit di stop e nessun controllo di flusso. La velocità del canale seriale è per default a 115200bps (configurabile a 57600bps, 19200bps, 9600bps).
- Tutti i pacchetti di dati sono composti da stringhe a caratteri in formato ASCII.
- Di default tutte le impostazioni sul modulo slave (dimmer IDP-PWM1-DRIVER) sono salvate su una memoria non volatile interna al modulo stesso. Questo consente il ripristino di tutte le impostazioni al successivo riavvio. Tale operazione può essere disabilitata tramite un opportuno comando (si veda oltre).
- La vita della memoria non volatile interna (EEPROM) è di circa 100000 cicli di scrittura. La lettura è infinita.

IDP-PWM1-DRIVER – Serie MINI DIN

STRUTTURA DEL PACCHETTO MASTER:

Ogni pacchetto spedito dal master (Host) verso uno slave (IDP-PWM1-DRIVER) è così composto:

\$	Indirizzo	Comando	Valore	<CR>
Carattere di inizio pacchetto ASCII 36.	Indirizzo dello slave a cui inviare il pacchetto.	Si veda più avanti la lista dei comandi supportati.	Non tutti i comandi prevedono l'invio di un valore numerico.	Carattere di fine pacchetto ASCII 13.

- 1) Tutti i valori numerici sono in base decimale e le cifre sono in caratteri ASCII.
- 2) Sono ammessi caratteri di spazio (codice ASCII 32) all'interno del pacchetto.
- 3) Sono ammessi zeri davanti a un valore numerico, ad esempio il valore numerico 12 può essere scritto come 12 o come 012.
- 4) Qualsiasi valore numerico non deve superare le 5 cifre, ad esempio sono valori formalmente corretti: 89, 0089, 00089. Non è considerato valido 000089.
- 5) Il pacchetto deve sempre iniziare col carattere "\$" (codice ASCII 36).
- 6) Il pacchetto deve sempre finire col carattere <INVIO> o <CR> Carriage Return (codice ASCII 13).
- 7) Le lettere del pacchetto devono essere tutte maiuscole.

STRUTTURA DEL PACCHETTO SLAVE:

Ogni pacchetto inviato da uno slave verso il master viene inviato solo in risposta a un pacchetto precedentemente spedito dal master allo slave.

#	Risposta	<CR>
Carattere di inizio pacchetto ASCII 35.	Si veda più avanti la lista delle risposte	Carattere di fine pacchetto ASCII 13.

- 1) Per ragioni di efficienza tutte le risposte da uno slave non prevedono caratteri di spazio (codice ASCII 32) né cifre di zero davanti a un valore numerico.
- 2) Sempre lo slave elabora una risposta in seguito a un pacchetto ricevuto a lui indirizzato.
- 3) Lo slave non risponde a un pacchetto che non sia a lui indirizzato.

RISPOSTE DELLO SLAVE:

#NOK<CR>	Errore.
----------	---------

Questa risposta segnala la presenza di uno o più errori, ad esempio un comando non riconosciuto o valore numerico fuori intervallo. In genere viene segnalato un errore quando uno dei 7 punti sopra riportati nel paragrafo STRUTTURA DEL PACCHETTO MASTER, non viene rispettato.

#OK<CR>	Comando eseguito.
---------	-------------------

Quando un comando ricevuto è valido e viene eseguito (senza prevedere il ritorno di un valore numerico) viene inviata questa risposta.

#<Msg><CR>	Risposta numerica.
------------	--------------------

Quando un comando ricevuto è valido e prevede la restituzione di un valore numerico viene spedita una risposta di questo tipo. Il campo <Msg> è composto da un valore numerico decimale (caratteri ASCII). Ad esempio se il master chiede a uno slave di inviare il valore di duty-cycle impostato su un canale, una possibile risposta potrebbe essere:

#125<CR> Ovvero un valore di registro pari a 125 (Valori validi da 0 a 255)

COMANDI DAL MASTER:

I comandi inviati dal master sono di due tipi. Comandi di scrittura e comandi di lettura. I comandi di scrittura prevedono anche il campo "Valore".

PWMR	Lettura Registro PWM
------	----------------------

Questo comando richiede allo slave di inviare il valore del registro pwm, relativo all'unico canale. Lo slave deve rispondere con un valore compreso tra 0 e 255.

Esempio:

MASTER (richiesta):	\$12 PWMR <CR>	Indirizzo dello slave: 12 (dec).
SLAVE (risposta):	#100<CR>	Il valore pwm è 100 (dec).
MASTER (richiesta):	\$8 PWMR <CR>	Indirizzo dello slave: 8 (dec).
SLAVE (risposta):	#250<CR>	Il valore pwm è 250 (dec).

IDP-PWM1-DRIVER – Serie MINI DIN

PWMW	<Valore>	Scrittura Registro PWM
------	----------	------------------------

Questo comando scrive nel registro pwm dello slave, un valore di duty-cycle specificato dal campo <Valore> = 0...255. Lo slave risponderà con #OK<CR> se l'operazione ha esito positivo.

Esempio:

MASTER (richiesta):	\$12 PWMW 90<CR>	Indirizzo dello slave: 12. Valore da scrivere: 90.
SLAVE (risposta):	#OK<CR>	Scrittura eseguita: pwm = 90.
MASTER (richiesta):	\$8 PWMW 255<CR>	Indirizzo dello slave: 8. Valore da scrivere: 255.
SLAVE (risposta):	#OK<CR>	Scrittura eseguita: pwm = 255.

SER	<Valore>	Scrittura Registro velocità seriale
-----	----------	-------------------------------------

Questo comando imposta la velocità della seriale. Il campo <Valore> ammette i seguenti valori:

0 = 115200bps (default)
1 = 57600bps
2 = 19200bps
3 = 9600bps

Lo slave risponderà con #OK<CR> se l'operazione ha esito positivo. Lo slave commuterà la velocità della seriale solo dopo aver dato la risposta.

Esempio:

MASTER (richiesta):	\$12 SER 3<CR>	Indirizzo dello slave: 12. Richiesta seriale a 9600bps.
SLAVE (risposta):	#OK<CR>	Comando valido. Seriale a 9600bps immediatamente dopo questa risposta.

DLYR	Lettura Registro ritardo risposta (DELAY)
------	---

Il comando legge il valore del registro di ritardo risposta (DELAY) sul canale seriale. Il registro in questione consente di ritardare, a passi di 10us, la commutazione da parte dello slave del bus RS485 dallo stato RX a TX. In pratica quando lo slave deve inviare una risposta al master, lo slave deve prendere il controllo del bus allo scopo di pilotarlo e inviare il pacchetto di risposta. Se il master è particolarmente lento a rilasciare il bus RS485 per consentire allo slave di rispondere, può esser utile imporre allo slave di attendere un po' di tempo prima di pilotare la linea differenziale RS485. Il registro in questione consente di ritardare la risposta da parte dello slave da 0us fino a 2550us (default). In pratica quando lo slave ha pronta la risposta lo si fa attendere un altro po' di tempo (in base al registro DELAY).

IDP-PWM1-DRIVER – Serie MINI DIN

Esempio:

MASTER (richiesta):	\$12 DLYR<CR>	Indirizzo dello slave: 12. Richiesta lettura Registro Ritardo.
SLAVE (risposta):	#100<CR>	Comando valido. Ritardo di 1000us.

DLYW	<Valore>	Scrittura Registro ritardo risposta (DELAY)
-------------	----------	---

Questo comando scrive il registro ritardo risposta (DELAY). Il campo <Valore> può assumere valori compresi tra 0 e 255. Un incremento unitario equivale a 10us. Come spiegato in precedenza tale ritardo viene introdotto nel momento in cui lo slave è pronto a inviare la risposta al master. Il tempo che intercorre tra la ricezione di un comando allo slave e la risposta di quest' ultimo al master, dipende dal tipo di comando e dal fatto che sia abilitato o meno il salvataggio in EEPROM del parametro (si veda avanti il comando EEON ed EEOFF). Quindi il ritardo impostato nel registro DELAY si sommerà al naturale ritardo dello slave per elaborare la risposta al comando.

Esempio:

MASTER (richiesta):	\$12 DLYW 22 <CR>	Indirizzo dello slave: 12. Richiesta scrittura Registro Ritardo (ritardo di 220us).
SLAVE (risposta):	#OK<CR>	Comando valido. Ritardo impostato.

EEON	Abilita salvataggio dei parametri in EEPROM
-------------	---

Questo comando consente di abilitare il salvataggio dei parametri di volta in volta inviati allo slave, in una memoria non volatile. Con questa opzione abilitata, ogni impostazione sullo slave, come ad esempio il valore del duty-cycle sui vari canali o la velocità della seriale, viene memorizzata e ritenuta anche in assenza di alimentazione per poter esser riproposta alle successive accensioni del modulo. Si tenga presente che l'operazione di scrittura in EEPROM impiega circa 2.8ms per esser eseguita, quindi il tempo di risposta dello slave a un comando di scrittura, come per esempio un PWMW, potrà impiegare circa 2.8ms in più o in meno a seconda che sia o meno abilitato il salvataggio del parametro in EEPROM. Questo comando non ha effetto sul comando EEW.

Esempio:

MASTER (richiesta):	\$12 EEON <CR>	Indirizzo dello slave: 12. Richiesta abilitazione salvataggio parametri in EEPROM.
SLAVE (risposta):	#OK<CR>	Comando valido. Salvataggio abilitato.

EEOFF	Disabilita salvataggio dei parametri in EEPROM
--------------	--

Il comando disabilita il salvataggio dei parametri in EEPROM. La disabilitazione può essere utile per velocizzare la risposta da parte dello slave, per tutti i comandi di scrittura o impostazione di un parametro. Allo stesso tempo è consigliata per un uso estremamente intensivo di operazioni in scrittura: per esempio se si decide in modo assiduo di generare delle rampe luminose o dei rapidi impulsi luminosi, il continuo aggiornamento in scrittura della EEPROM può portare a un progressivo deperimento della stessa in quanto progettata per 100K cicli in scrittura. Questo comando non ha effetto sul comando EEW.

Esempio:

MASTER (richiesta):	\$12 EEOFF <CR>	Indirizzo dello slave: 12. Richiesta disabilitazione salvataggio parametri in EEPROM.
SLAVE (risposta):	#OK<CR>	Comando valido. Salvataggio disabilitato.

EES	Stato salvataggio parametri in EEPROM
------------	---------------------------------------

L'invio di questo comando ha lo scopo di chiedere allo slave se il salvataggio in EEPROM dei parametri è attivo oppure no. Lo slave risponderà con #0<CR> per indicare che lo stato è OFF (ovvero i parametri non vengono salvati in EEPROM), oppure risponderà con #1<CR> per indicare che lo stato è ON (salvataggio abilitato).

Esempio:

MASTER (richiesta):	\$12 EES <CR>	Indirizzo dello slave: 12. Richiesta dello stato di salvataggio parametri in EEPROM.
SLAVE (risposta):	#1<CR>	Comando valido. Salvataggio abilitato (ON).

EEA	<Valore>	Puntatore locazione di memoria EEPROM utente
------------	----------	--

Il comando indica al modulo slave la locazione di memoria utente su EEPROM, che potrà successivamente essere letta o scritta tramite i comandi EER o EEW. Il campo <Valore> indica la locazione di memoria su EEPROM: I valori sono compresi tra 0 e 239, il valore iniziale di default del puntatore è 0.

Esempio:

MASTER (richiesta):	\$12 EEA 88 <CR>	Indirizzo dello slave: 12. Richiesta di puntare alla locazione 88 nella memoria EEPROM.
SLAVE (risposta):	#OK<CR>	Comando valido. Puntatore caricato.

EEW	<Valore>	Scrittura memoria utente su EEPROM
------------	----------	------------------------------------

Il comando consente di scrivere un byte di dato, indicato dal campo <Valore>, nella memoria utente precedentemente puntata dal comando EEA. I comandi EEON ed EEOFF non hanno effetto su questo comando (la scrittura con EEW è sempre attiva).

Esempio:

MASTER (richiesta):	\$12 EEW 100 <CR>	Indirizzo dello slave: 12. Richiesta di scrittura del valore 100 nella locazione puntata in precedenza dal comando EEA.
SLAVE (risposta):	#OK<CR>	Comando valido. Byte scritto.

EER	Lettura memoria utente su EEPROM	
------------	----------------------------------	--

L'invio del comando allo slave, consente di leggere il byte contenuto nella locazione di memoria utente (EEPROM) puntata in precedenza tramite il comando EEA. Le locazioni non scritte in precedenza hanno un valore di default pari a 255.

Esempio:

MASTER (richiesta):	\$12 EER <CR>	Indirizzo dello slave: 12. Richiesta di lettura del byte nella locazione puntata in precedenza dal comando EEA.
SLAVE (risposta):	#100<CR>	Comando valido. Byte letto.

VER	Versione HW e FW	
------------	------------------	--

Questo comando richiede allo slave di fornire la propria versione dell'hardware e del firmware. I primi due caratteri numerici indicano la versione Hardware mentre i successivi due caratteri quella Firmware.

Esempio:

MASTER (richiesta):	\$12 VER <CR>	Indirizzo dello slave: 12. Richiesta Versione.
SLAVE (risposta):	#1010<CR>	Comando valido. Versione HW = 1.0 Versione FW = 1.0

FRQW	<Valore>	Impostazione Frequenza PWM
-------------	----------	----------------------------

Il comando indica al modulo slave il valore di frequenza del segnale PWM da mandare in uscita. Tale valore potrà successivamente essere letto tramite il comando FRQR. Il campo <Valore> indica la frequenza da impostare in base alla seguente tabella.

TABELLA DELLE FREQUENZE

<Valore>	Valore della frequenza in uscita (PWM)
0	60Hz
1	100Hz
2	200Hz
3	300Hz
4	400Hz
5	500Hz
6	600Hz
7	700Hz
8	800Hz
9	900Hz
10	1KHz
11	2KHz
12	3KHz
13	4KHz
14	5KHz
15	6KHz
16	7KHz
17	8KHz
18	9KHz
19	10KHz
20	11KHz
21	12KHz
22	13KHz
23	14KHz
24	15KHz
25	16KHz
26	17KHz
27	18KHz
28	19KHz
29	20KHz
30	21KHz
31	22KHz
32	23KHz
33	24KHz
34	25KHz

Tutti i valori di frequenza sono entro una tolleranza di $\pm 1\%$.

Esempio:

MASTER (richiesta):	\$12 FRQW 32 <CR>	Indirizzo dello slave: 12. Richiesta di impostare la frequenza di 23KHz.
SLAVE (risposta):	#OK<CR>	Comando valido. Valore impostato.

FRQR	Lettura della Frequenza PWM
-------------	-----------------------------

Il comando indica al modulo slave di fornire il valore di frequenza del segnale PWM. Tale valore potrà successivamente essere modificato tramite il comando FRQW. Il <Valore> restituito indica la frequenza impostata in base alla tabella sopra riportata (comando FRQW).

Esempio:

MASTER (richiesta):	\$12 FRQR <CR>	Indirizzo dello slave: 12. Richiesta di lettura della frequenza PWM.
SLAVE (risposta):	#32<CR>	Comando valido. Valore impostato pari a 23KHz.

TABELLA RIASSUNTIVA:

Di seguito viene presentata una tabella riassuntiva dei comandi dal master verso lo slave e delle risposte dello slave verso il master. In tabella vengono riportati anche i valori di default presenti nel modulo dimmer (slave) alla prima accensione.

Comando	Tipo	Ritorno	Default
PWMR	Lettura	0...255	255 (max. power)
PWMW <0...255>	Scrittura	OK, NOK	-
SER <0...3>	Scrittura	OK, NOK	0 (115.2Kbps)
DLYR	Lettura	0...255	255 (ritardo 2.55ms)
DLYW <0...255>	Scrittura	OK, NOK	-
EEON	Scrittura	OK, NOK	-
EEOFF	Scrittura	OK, NOK	-
EES	Lettura	0...1	1 (ON) default
EEA <0...239>	Scrittura	OK, NOK	0 (default)
EEW <0...255>	Scrittura	OK, NOK	-
EER	Lettura	0...255	255 (mem. non prg.)
VER	Lettura	nnnn (n=0...9)	-
FRQW <0...34>	Scrittura	OK, NOK	Vedi tab. frequenze
FRQR	Lettura	0...34	Vedi tab. frequenze

Tabella2. Riassunto dei comandi.

IDP-PWM1-DRIVER – Serie MINI DIN

Sul mercato sono molti i programmi anche gratuiti che consentono di implementare un monitor seriale o un video terminale in grado di gestire un canale seriale. In particolare possiamo suggerire il nostro terminale seriale IDPTERM, disponibile gratuitamente nel nostro sito web.

L'utilizzo del modulo IDP-PWM1-DRIVER su un PC, richiede un'interfaccia USB/RS485. In commercio vi sono molti adattatori di tale tipo, da parte nostra possiamo proporre il nostro convertitore USB-RS485 (IDP-USB485ISO) che consente una elevata affidabilità grazie ad un circuito di isolamento galvanico tra PC e Bus RS485.

IDP-PWM1-DRIVER – Serie MINI DIN

Questo documento è stato scritto allo scopo di fornire una presentazione relativamente a prodotti realizzati da INDEP SRL.

Per ulteriori informazioni rivolgersi direttamente a INDEP SRL.

Le informazioni in questo documento si intendono accurate e affidabili. L'azienda comunque non si assume alcuna responsabilità per errori che possano comparire in questo documento. L'azienda si riserva il diritto di apportare variazioni sia ai dispositivi sia alle specifiche accluse in questo documento in ogni momento e senza preavviso.

Nessuna licenza a brevetti o a proprietà intellettuali di INDEP SRL sono dovute da parte dell'azienda in relazione alla vendita o alla visione dei propri prodotti. I prodotti di INDEP SRL non sono autorizzati per l'uso come componenti critici in dispositivi o sistemi vitali.

Alcuni nomi di prodotti menzionati in questo documento potrebbero risultare marchi registrati: in questo caso tali prodotti o marchi vengono usati solamente per puro riferimento.

INDEP SRL. Tutti i diritti sono riservati.