

***LaurTec***

**RS232-CAN  
Protocol Converter**

**Autore :** *Mauro Laurenti*

**email:** [info.laurtec@gmail.com](mailto:info.laurtec@gmail.com)

**ID:** PJ2003-IT

## INFORMATIVA

Come prescritto dall'art. 1, comma 1, della legge 21 maggio 2004 n.128, l'autore avvisa di aver assolto, per la seguente opera dell'ingegno, a tutti gli obblighi della legge 22 Aprile del 1941 n. 633, sulla tutela del diritto d'autore.

Tutti i diritti di questa opera sono riservati. Ogni riproduzione ed ogni altra forma di diffusione al pubblico dell'opera, o parte di essa, senza un'autorizzazione scritta dell'autore, rappresenta una violazione della legge che tutela il diritto d'autore, in particolare non ne è consentito un utilizzo per trarne profitto.

La mancata osservanza della legge 22 Aprile del 1941 n. 633 è perseguibile con la reclusione o sanzione pecuniaria, come descritto al Titolo III, Capo III, Sezione II.

A norma dell'art. 70 è comunque consentito, per scopi di critica o discussione, il riassunto e la citazione, accompagnati dalla menzione del titolo dell'opera e dal nome dell'autore.

## AVVERTENZE

I progetti presentati non hanno la certificazione CE, quindi non possono essere utilizzati per scopi commerciali nella Comunità Economica Europea.

Chiunque decida di far uso delle nozioni riportate nella seguente opera o decida di realizzare i circuiti proposti, è tenuto pertanto a prestare la massima attenzione in osservanza alle normative in vigore sulla sicurezza.

L'autore declina ogni responsabilità per eventuali danni causati a persone, animali o cose derivante dall'utilizzo diretto o indiretto del materiale, dei dispositivi o del software presentati nella seguente opera.

Si fa inoltre presente che quanto riportato viene fornito così com'è, a solo scopo didattico e formativo, senza garanzia alcuna della sua correttezza.

L'autore ringrazia anticipatamente per la segnalazione di ogni errore.

Tutti i marchi citati in quest'opera sono dei rispettivi proprietari.

## Introduzione

Ogni volta che bisogna interfacciare il computer con un sistema che possiede il CAN bus è necessario un convertitore di protocollo che permetta la comunicazione del PC con il sistema. Nel seguente progetto (L2000-30004) viene presentato un convertitore RS232-CAN. Il progetto è realizzato per mezzo della scheda Freedom opportunamente estesa con un CAN transceiver. Il sistema supporta il CAN 2.0A e B e può essere utilizzato per la realizzazione di un sistema di diagnosi di bus CAN.

## Uno sguardo d'insieme

Nella seguente documentazione si considera che il lettore abbia familiarità con quanto esposto nel Tutorial "CAN Bus" e con la scheda di sviluppo "Freedom"<sup>1</sup>.

Il Protocol Converter ha lo scopo di convertire i dati inviati dal PC, per mezzo della porta seriale, secondo le specifiche del bus CAN e viceversa. Tale dispositivo risulta a tutti gli effetti un nuovo nodo sul bus CAN. Diversamente dagli altri nodi non possiede un proprio indirizzo dal momento che ha lo scopo di ricevere tutti i dati sul Bus indipendentemente dall'identifier<sup>2</sup>. Lo schema a blocchi del Protocol Converter L2000-30004 è riportato in Figura 1<sup>3</sup>

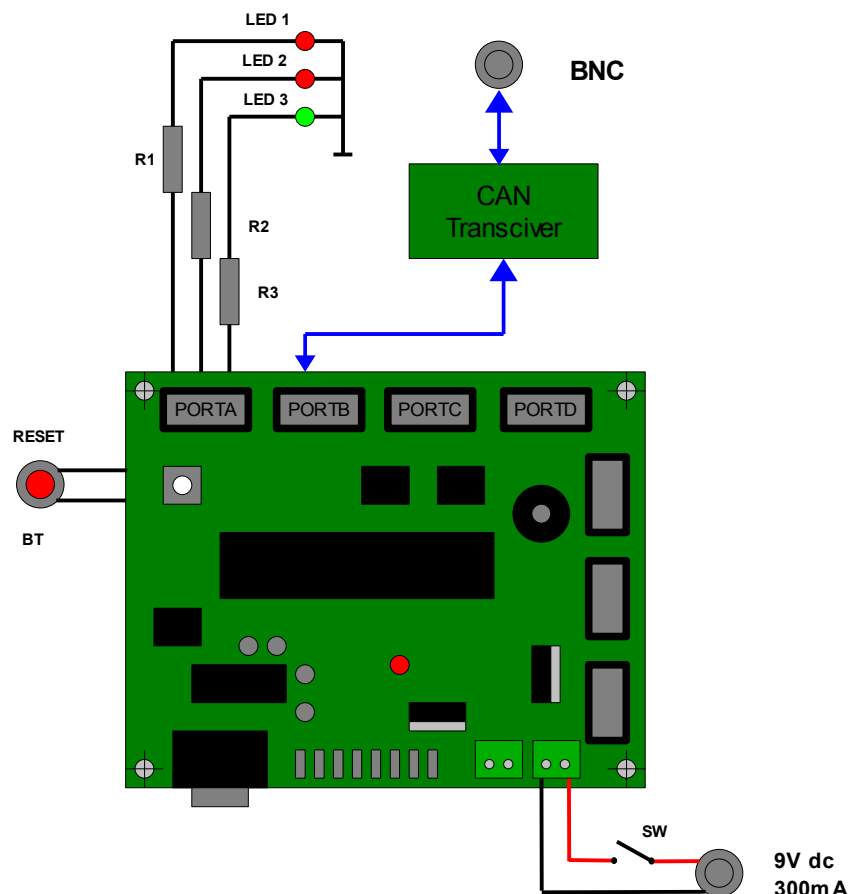


Figura 1: Schema a blocchi del sistema L2000-30004

<sup>1</sup> La documentazione può essere scaricata dal sito [www.LaurTec.com](http://www.LaurTec.com).

<sup>2</sup> Il software potrebbe essere modificato in modo da supportare l'indirizzamento.

<sup>3</sup> Si fa notare che il sistema non possiede la resistenza da 120ohm per terminare il bus. Questo poiché si è fatta l'assunzione che il bus dove si collega il sistema L2000-30004 sia già propriamente terminato. In ogni modo qualora sia necessario la resistenza può essere facilmente aggiunta. Il circuito potrebbe essere modificato anche per poter controllare via software la resistenza di terminazione.

Il Protocol Converter è ottenuto per mezzo della demoboard Freedom sulla quale è stato montato un PIC18F4580 che possiede al suo interno un modulo CAN. Lo schema del sistema è riportato in Figura 2<sup>4</sup>.

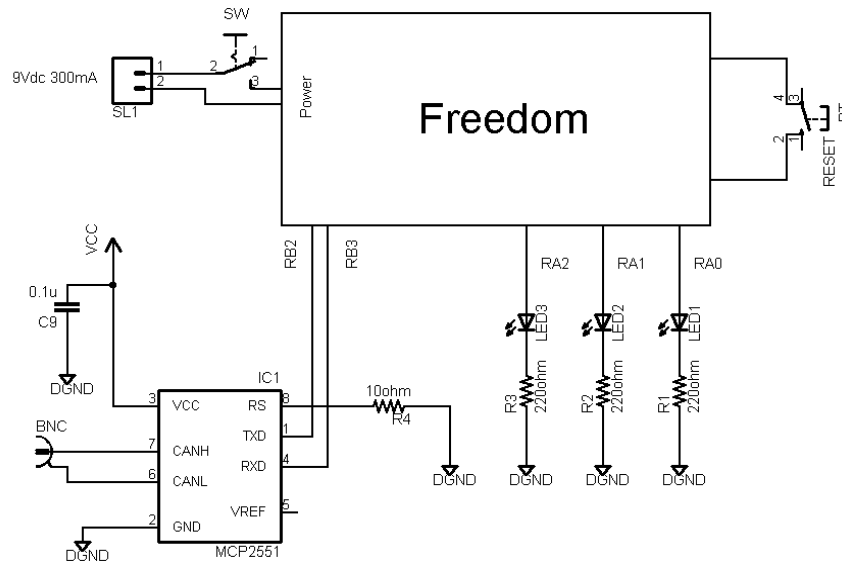


Figura 2: Schema elettrico del sistema

E' possibile vedere che il sistema risulta particolarmente semplice e consiste semplicemente nell'aggiunta di tre led, un pulsante di reset e del CAN transceiver MCP2515 posto alle uscite del modulo CAN. Il pulsante di reset è collegato in parallelo al pulsante già presente sulla scheda Freedom. E' stato aggiunto solo per permettere il reset del sistema dall'esterno della scatola di montaggio. La resistenza R4 da 10Ω permette di regolare la ripidità dei fronti di salita dei bit trasmessi dal transceiver. Questa resistenza potrebbe essere variata a seconda delle esigenze del progetto.

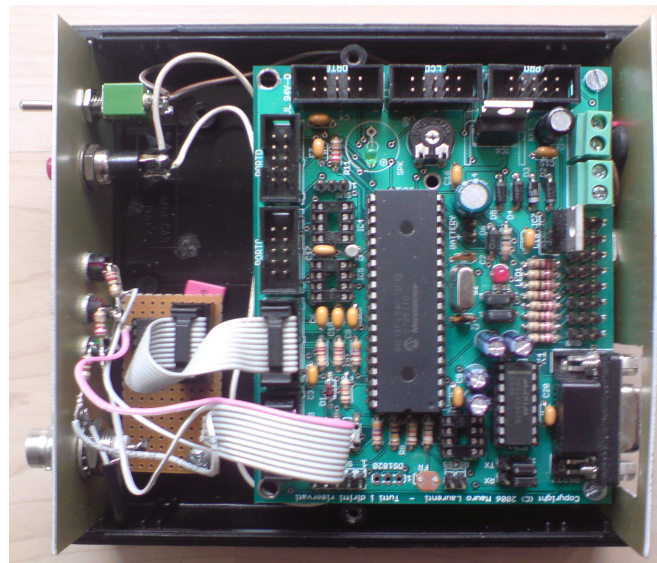


Figura 3: Sistema montato

<sup>4</sup> Si fa notare che come connettore si è fatto uso di un BNC standard da 50ohm. Per applicazioni professionali è bene sostituire questo connettore con connettori dedicati al bus CAN.

## Interfaccia LED

I led collegati sul pannello frontale riportati in Figura 4, hanno il seguente significato:



**Figura 4: Led di segnalazione frontali**

### LED1

Collegato su RA0. Rappresenta TX\_MESSAGE flag, ovvero viene acceso ogni volta che il PC trasmette un messaggio attraverso la porta seriale. Il led viene dunque acceso quando un messaggio passa dalla porta seriale RS232 al bus CAN. Il colore del led è rosso.

### LED2

Collegato su RA1. Rappresenta RX\_MESSAGE flag, ovvero viene acceso ogni volta che il PC riceve un messaggio attraverso la porta seriale. Il led viene dunque acceso quando un messaggio passa dal bus CAN alla porta seriale. Il colore del led è rosso.

### LED3

Collegato su RA2. Rappresenta HEART flag. Quando il sistema funziona correttamente il led lampeggia simulando il battito del cuore. Il colore del led è verde.

Oltre alle funzioni ora descritte si ha che:

- Al momento dell'accensione del sistema tutti e tre i led si accendono e rimangono accesi per circa un secondo. Dopodiché si accendono in sequenza con lo scopo di mostrare che il PIC è funzionante.
- Quando si verifica un errore il LED3 smette di lampeggiare e rimane sempre acceso. Il sistema potrebbe essere ancora funzionante a seconda dell'errore accaduto. Per ripristinare il lampeggio è necessario premere il tasto di reset.

Gli errori riconosciuti sono:

- **Comando non riconosciuto**  
Attivato quando il sistema riceve dalla porta seriale un comando non valido
- **TX error counter**  
Attivato quando il modulo CAN ha attivato lo stato di errore in trasmissione.
- **RX error counter**  
Attivato quando il modulo CAN ha attivato lo stato di errore in ricezione.

Come si vedrà a breve lo stato di errore è memorizzato all'interno di una variabile error che è possibile leggere attraverso opportuno comando. Il TX e RX counter possono essere letti con ulteriori comandi.

## Utilizzo del Protocol Converter

Si capisce che per poter convertire dei dati da RS232 secondo le specifiche CAN bus non è solo necessario cambiare le caratteristiche elettriche del segnale ma è necessario anche rispettare un determinato formato nella trasmissione dei byte, in modo da poter trasmettere l'identificatore, i byte dati e i rispettivi settaggi tipici del protocollo CAN.

In Figura 5 è riportata la struttura dati che il PC deve trasmettere al sistema L2000-30004 e ricevere da quest'ultimo. La struttura dati è fissa ovvero tutti i byte devono essere sempre trasmessi anche se non contengono informazioni utili. Il ruolo del microcontrollore è quello di formattare i dati ricevuti dal PC secondo il protocollo CAN e viceversa. Il Command Type è trasmesso per primo e gli altri sono a seguire. La porta seriale del PC deve essere impostata per trasmissioni a 19000 b/s, 8bit, senza bit di parità ed un bit di stop. Si capisce che il sistema pur supportando il CAN 2.0A e B non può trasmettere a pari velocità le informazioni al PC<sup>5</sup>.

Command Type	Data Info	Identifier Field	Data Field
1 Byte	1 Byte	4 Byte	8 Byte

Figura 5: Struttura dati sul bus RS232

### Command Type

Questo campo specifica il comando inviato dal PC alla scheda L2000-30004 e consiste in un byte. I comandi supportati dal sistema L2000-30004 sono riportati in Tabella 1.

Comandi	Hex Code
Invia messaggio (dal PC alla scheda L2000-30004)	0xAA
Messaggio ricevuto (inviato dalla scheda al PC)	0x99
Richiesta contatori errori	0xA0
Richiesta Error Flags	0xA1
Imposta modalità "loop mode"	0xA2
Imposta modalità "normal mode"	0xA3

Tabella 1: Comandi RS232-CAN

- **Comando 0xAA**

Questo comando viene utilizzato dal PC per comunicare alla scheda L2000-30004 il messaggio da inviare sul bus CAN.

#### Valore di ritorno

Nessuno

- **Comando 0x99**

Questo comando viene utilizzato dalla scheda L2000-30004 per segnalare al PC il messaggio ricevuto sul bus CAN.

<sup>5</sup> La versione attuale del firmware non supporta nessun buffer interno oltre a quelli presenti nel modulo CAN.

**Valore di ritorno**

Dati dal bus CAN. I byte inviati dalla scheda hanno lo stesso formato dei byte inviati dal PC. Tutti i campi vengono impostati a seconda del messaggio ricevuto<sup>6</sup>.

- **Comando 0xA0**

Questo comando viene utilizzato dal PC per richiedere alla scheda L2000-30004 i contatori d'errore in trasmissione e ricezione.

**Valore di ritorno**

I due contatori RX e TX vengono inviati rispettivamente nella posizione del data Byte 0 e del data Byte 1 (si veda Figura 8). Il campo "Command Type" assume il valore 0xA0. L'identificatore come anche gli altri data byte devono essere ignorati.

- **Comando 0xA1**

Questo comando viene utilizzato dal PC per richiedere alla scheda L2000-30004 il flag di errori.

**Valore di ritorno**

Il flag è costituito da un byte che viene trasmesso nella posizione del data Byte 0 (si veda Figura 8). Il campo "Command Type" assume il valore 0xA1. L'identificatore come anche gli altri data byte devono essere ignorati. I bit del flag error hanno il seguente significato:

**Bit 0**

1 : Comando non riconosciuto  
0 : Comando riconosciuto

**Bit 1**

1 : TX error Counter ha superato il livello di error  
0 : TX Counter è al di sotto del livello di error

**Bit 2**

1 : RX error Counter ha superato il livello di error  
0 : RX Counter è al di sotto del livello di error

**Bit 3**

1 : Il nodo CAN L2000-30004 è bus off  
0 : Il nodo CAN è operativo

- **Comando 0xA2**

Questo comando viene utilizzato dal PC per impostare la scheda L2000-30004 in modalità "loop mode". Questa modalità può essere utilizzata per ragioni diagnostiche.

**Valore di ritorno**

Nessuno

- **Comando 0xA3**

Questo comando viene utilizzato dal PC per impostare la scheda L2000-30004 in modalità normale.

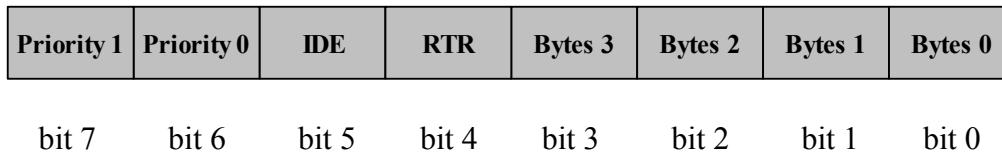
<sup>6</sup> Si fa notare che la versione attuale del Firmware supporta solo lo standard 2.0B alla velocità di 125KHz. Il limite è semplicemente software e può essere migliorato.

**Valore di ritorno**

Nessuno

**Data Info**

Il campo Data Info è composto da un byte strutturato come riportato in Figura 6.

**Figura 6: Struttura del Data Info**

**Priority 0-1** : Questi bit sono utilizzati per indicare la priorità del messaggio. Questa priorità non ha nulla a che fare con il protocollo CAN ma è legata al PIC. Il suo valore può essere compreso tra 0-3<sup>7</sup>. Se non si ha interesse nella priorità si può utilizzare un qualunque valore valido come valore di default.

**IDE**: Questo bit indica il tipo di messaggio.

1 : indica extended format

0 : indica standard format.

**RTR** : Questo bit indica se il messaggio è associato a un remote frame

1 : indica remote frame (campo dati è ignorato)

0 : indica normal message

**Byte 3-0** : Questi bit indicano il numero di byte contenuti dalla struttura dati. I byte trasmessi dal PC alla scheda L2000-30004 sono sempre 8, come mostrato in Figura 8 ma il numero di byte fisicamente trasmessi sul bus CAN dipenderà dal numero definito da questi bit.

**Identifier**

Questo campo rappresenta l'identificatore del messaggio e consiste di 4 byte, I quali vengono trasmessi secondo l'ordine riportato in Figura 7. Il numero di byte utilizzati dipende dal tipo di messaggio utilizzato, esteso o standard. Per messaggi standard solo i primi 11 bit saranno trasmessi attraverso il bus CAN mentre se il formato del messaggio è esteso verranno utilizzati 29 bit. Il PC trasmetterà e riceverà i 4 byte associati all'identificatore indipendentemente dal formato. Sarà al software estrarre l'informazione a seconda del bit IDE precedentemente descritto.

**Figura 7: Struttura dell'identifier**

<sup>7</sup> Il livello di priorità aumenta dal livello 0 a 3. Per maggiori informazioni si faccia riferimento al datasheet del PIC18F2580.



**Data Field**

Questo campo consiste di 8 byte. Come detto in precedenza gli 8 byte vengono trasmessi/ricevuti dal PC indipendentemente dalla lunghezza effettiva della struttura dati. In ogni modo I dati effettivamente trasmessi sul bus CAN saranno quelli specificati dal valore dei bit Byte 3-0 precedentemente descritti. I byte sono trasmessi dal Byte 0 al Byte7 come mostrato in Figura 8.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
--------	--------	--------	--------	--------	--------	--------	--------

**Figura 8: Struttura del Data Field**

**Bibliografia**

[www.LaurTec.com](http://www.LaurTec.com) : sito di elettronica dove poter scaricare gli altri articoli menzionati, aggiornamenti e progetti.