

LaurTec

miniCOM USB

Autore : *Mauro Laurenti*

ID: PJ4001-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.

Indice

Introduzione	4
Specifiche Tecniche	4
Analisi del progetto	5
Il microcontrollore.....	5
Il quarzo	8
L'alimentatore.....	9
Pulsanti.....	12
Altre periferiche.....	13
Connettore di espansione.....	14
Connettore di Programmazione e Debug.....	16
Layout Periferiche.....	17
Istruzioni per il montaggio	18
Collaudo e messa in funzione	20
Prima messa in funzione dopo il montaggio.....	20
Messa in funzione ordinaria.....	20
Software ed esempi	21
Esempio 1: Classe CDC, Input Output Demo	22
Esempio 2: miniCom UART	26
Esempio 3: Classe HID, miniCOM Relay	29
...non ci scordiamo del PID, VID	31
Bibliografia	34
History	35

Introduzione

La porta USB è ormai il sistema più usato per lo scambio d'informazioni tra sistemi elettronici e per connettere ogni qualsivoglia dispositivo ad un PC. Anche se non di facile utilizzo come la porta parallela e la porta seriale, librerie e hardware dedicati possono agevolare notevolmente l'apprendimento e l'utilizzo della porta USB. Utilizzare l'interfaccia USB permette di rendere un nuovo progetto al passo con i tempi, permettendo di seguire le aspettative dell'utente finale.

La scheda di sviluppo miniCOM USB, fornisce l'essenziale per iniziare l'avventura USB grazie all'utilizzo del PIC18F14K50. Il connettore di espansione LTB (LaurTec Bus), permette di collegare la scheda a sistemi esterni per mezzo dei bus seriali più comuni (UART, SPI, I2C). I connettori laterali di espansione SV1 e SV2 permettono di collegare la scheda a sistemi più complessi. Le schede di espansione permettono di utilizzare miniCOM USB come mini scheda madre e strumento di laboratorio per l'analisi di bus seriali. La possibilità di programmare la scheda semplicemente per mezzo del *LaurTec PIC Bootloader GUI*, permette di cambiare l'applicazione e funzione della scheda per mezzo di pochi click e senza necessitare di un programmatore.

Specifiche Tecniche

Alimentazione : 5V

Assorbimento : 30mA con LED attivi

Dimensioni : 50 x 55 mm

Part Number : PJ4001

Versione : 2

Peso : 25g

Il sistema miniCOM supporta il seguente hardware:

- Supporto USB 2.0 Low Speed e Full Speed
- 1 pulsante + Reset
- 2 LED di utilizzo generico
- Connettore espansione LTB (LaurTec Bus)
- Programmabilità per mezzo dell'USB Bootloader
- Programmazione on-board e Debug compatibile con gli strumenti Microchip

Smaltimento



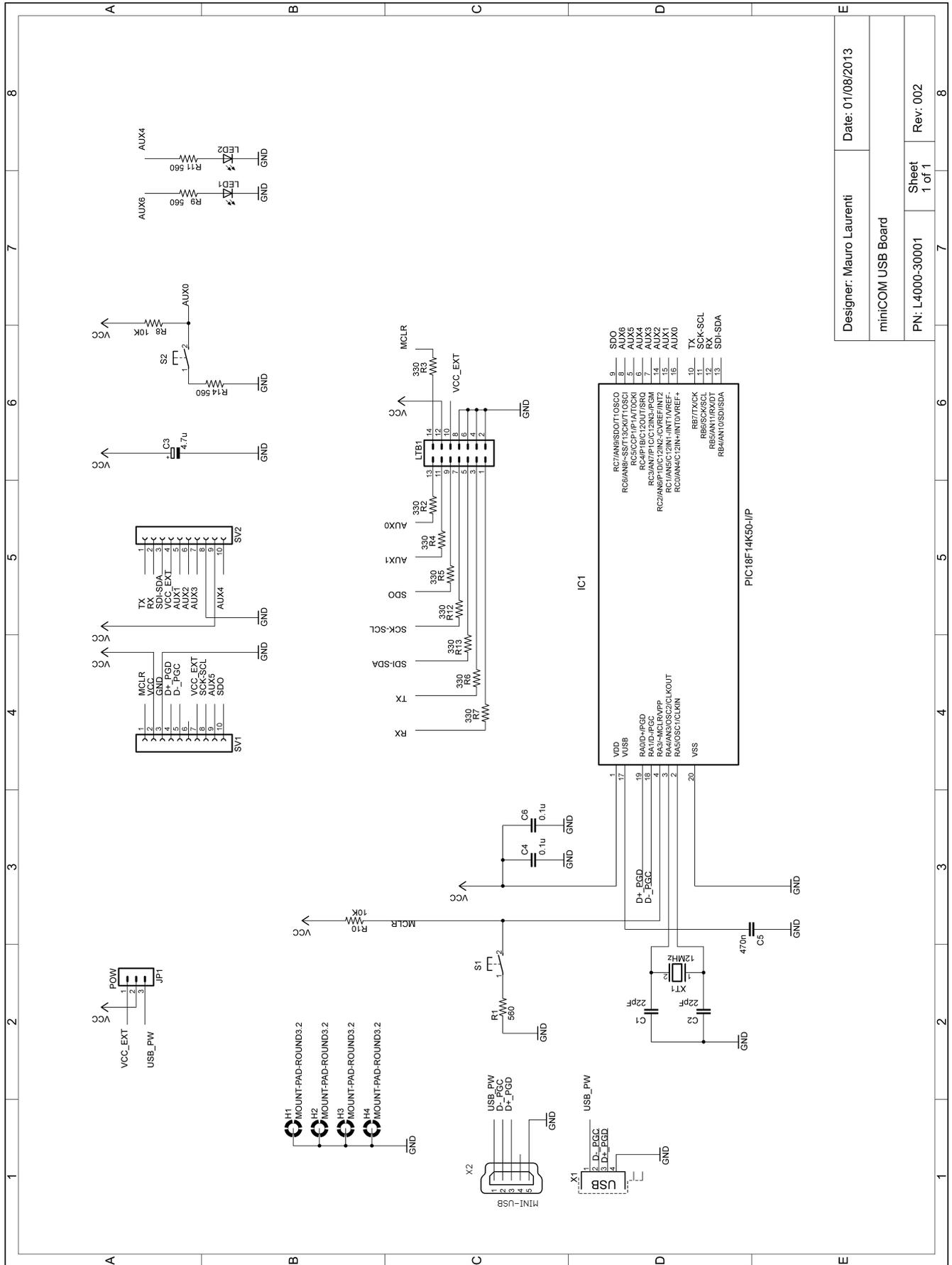
Secondo la Direttiva Europea 2002/96/EC tutti i dispositivi elettrici/elettronici devono essere considerati rifiuti speciali e non devono essere gettati tra i rifiuti domestici. La gestione e lo smaltimento dei rifiuti elettrici/elettronici viene a dipendere dalle autorità locali e governative. Un corretto smaltimento dei rifiuti permette di prevenire conseguenze negative per l'ambiente e ai suoi abitanti. E' obbligo morale, nonché legale, di ogni singolo cittadino, di attenersi alla seguente Direttiva. Per ulteriori chiarimenti l'Autore è a disposizione.

Analisi del progetto

In Figura 1 è riportato lo schema elettrico della scheda di sviluppo miniCOM USB. Dalle dimensioni ridotte è possibile subito capire che la scheda può essere utilizzata per espandere altri sistemi elettronici aggiungendo agli stessi la funzionalità di una porta USB. La scheda si presta, grazie alla presenza di 2 pulsanti, 2 LED, anche allo sviluppo di piccoli sistemi di prova. Il connettore di espansione LTB (LaurTec Bus) e i connettori di espansione laterali SV1 e SV2, permettono di collegare la scheda a sistemi più complessi e rappresentare l'interfaccia USB principale per mezzo della quale comunicare con il sistema stesso. I connettori SV1 e SV2 permettono di collegare schede di espansione quali miniCOM RS232, miniCOM Relays, miniCOM Extension Board, ottenendo utili strumenti da laboratorio e sistemi di sviluppo. Le applicazioni di miniCOM USB possono facilmente essere cambiate grazie al bootloader interno al PIC18F14K50 e *LaurTec PIC Bootloader GUI*.

Il microcontrollore

La scheda miniCOM USB nasce con lo scopo di permettere lo sviluppo di applicazioni USB per mezzo del PIC18F14K50 a 20 pin, che possiede il modulo USB ovvero il modulo SIE (Serial Interface Engine). Non è dunque pensata come le schede di sviluppo Freedom II, Freedom Light ed EasyUSB per il supporto di una più ampia famiglia di microcontrollori. L'hardware essenziale e mirato al PIC18F14K50 permette di avere un sistema semplice e privo di configurazioni speciali. Se da un lato si perde di flessibilità, si guadagna in semplicità, che per un primo momento può rappresentare uno strumento essenziale per garantire un successo rapido. Qualora si volesse una maggiore flessibilità o disponibilità di hardware si possono utilizzare schede più complesse, quali appunto Freedom II, Freedom Light e Easy USB.



Designer: Mauro Laurenti	Date: 01/08/2013
miniCOM USB Board	
PN: L4000-30001	Sheet 1 of 1
	Rev: 002

Figura 1: Schema elettrico di miniCOM USB.

Lista Componenti

Resistori

R1 = 560Ω %5 1/4W

R2 = 330Ω %5 1/4W

R3 = 330Ω %5 1/4W

R4 = 330Ω %5 1/4W

R5 = 330Ω %5 1/4W

R6 = 330Ω %5 1/4W

R7 = 330Ω %5 1/4W

R8 = 10KΩ %5 1/4W

R9 = 560Ω %5 1/4W

R10 = 10KΩ %5 1/4W

R11 = 560Ω %5 1/4W

R12 = 330Ω %5 1/4W

R13 = 330Ω %5 1/4W

R14 = 560Ω %5 1/4W

Condensatori

C1 = 22pF ceramico

C2 = 22pF ceramico

C3 = 4.7μF elettrolitico 50V

C4 = 0.1μF ceramico 50V

C5 = 470nF ceramico 50V

C6 = 0.1μF ceramico 50V

Circuiti Integrati

IC1 = PIC18F14K50

IC1-Z = Zocchetto IC 20 pin

Diodi

LED1 = LED 3mm verde

LED2 = LED 3mm verde

Quarzi

XT1 = 12MHz

Pulsanti

S1 = micro-pulsante a montaggio verticale per PCB

S2 = micro-pulsante a montaggio verticale per PCB

Connettori

LTB1 = IDC 14 pin (maschio)

X1 = USB mini B o Tipo B (vedi note sul montaggio)

JP1 = Jumper 3 pin maschio (passo 2.54)

JPx = Jumper femmina (passo 2.54)

SV1-SV2 = Connettore 10 pin femmina 8mm (passo 2.54)

Il quarzo

Ogni microcontrollore ha bisogno di un clock per poter funzionare. Il clock rappresenta la base del tempo che il microcontrollore utilizza per l'esecuzione delle istruzioni per cui è stato programmato. Alcuni microcontrollori della Microchip, tra cui il PIC18F14K50, hanno la possibilità di generare un clock interno, ciononostante al fine di utilizzare il modulo USB è necessario far uso di un cristallo esterno al fine di garantire una maggiore affidabilità della comunicazione. In Figura 2 è riportato lo schema a blocchi della circuiteria di clock interna al PIC18F14K50. Si può subito notare, confrontandolo con la circuiteria di un PIC18F4550, che risulta più essenziale e pone maggiori vincoli sul cristallo da utilizzare se paragonato al PIC18F4550.

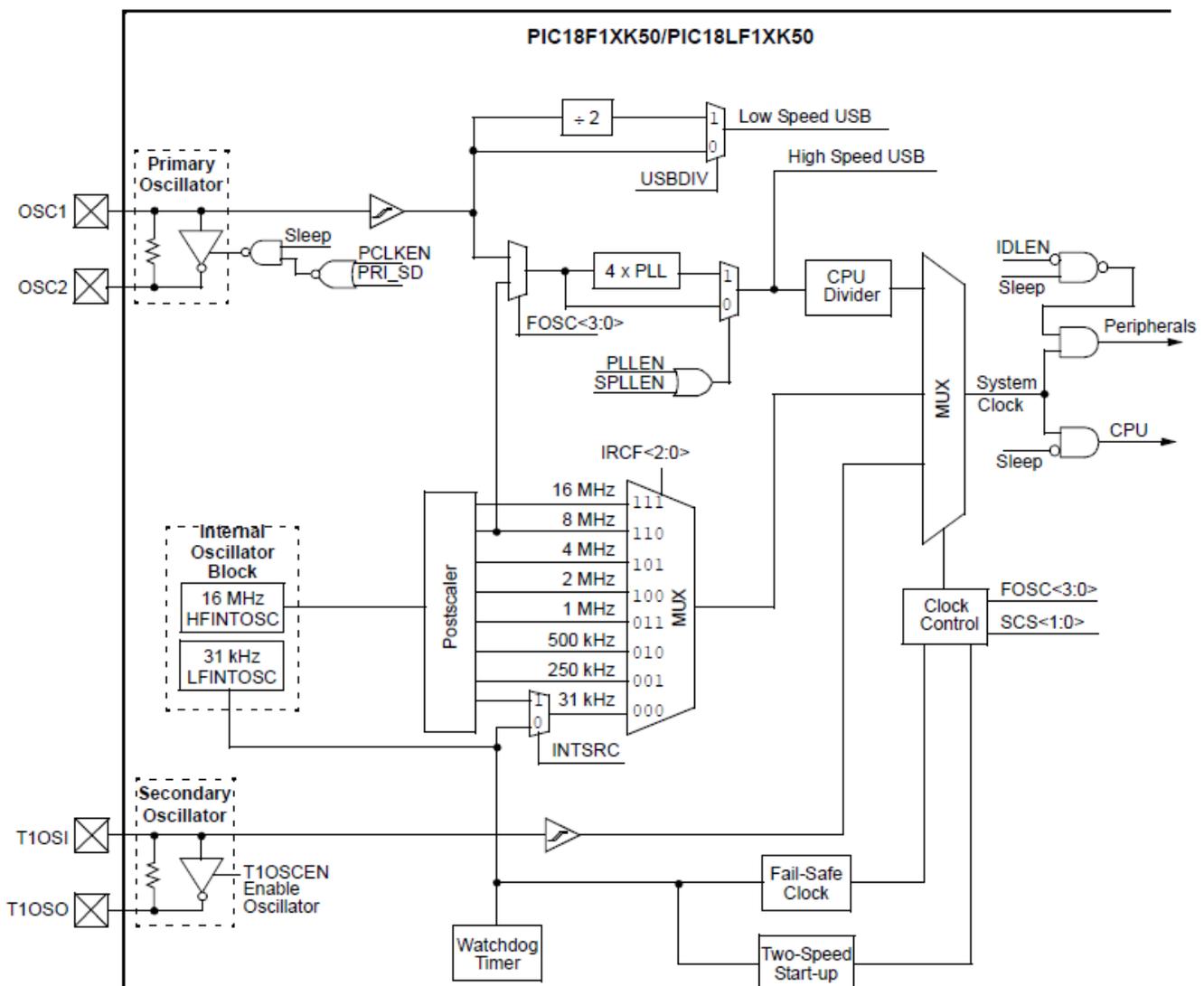


Figura 2: Schema a blocchi della circuiteria di clock, interna al PIC18F15K50.

Dal momento che il PLL moltiplica per 4 direttamente alla frequenza del cristallo, e per il modulo USB in modalità Full Speed sono necessari 48MHz, si ha la necessità di avere un cristallo di 12MHz. Il modulo del clock del PIC18F4550 permette invece di utilizzare anche il cristallo di 20MHz, grazie alla presenza di un prescaler ovvero un divisore di frequenza. Il resto delle opzioni interne rimane più o meno simile, ovvero si ha un oscillatore interno ad alta frequenza e un buffer secondario per un cristallo a bassa frequenza (tipicamente 32KHz).

L'alimentatore

La scheda miniCOM USB non possiede un vero e proprio alimentatore (convertitore DC-DC) visto che l'alimentazione viene prelevata direttamente dalla porta USB. Ciononostante è bene evidenziare alcune particolarità relative alle opzioni disponibili. Infatti il PIC18F14K50 può essere alimentato a 5V sia per mezzo della porta USB ovvero USB_PW che per mezzo di un sistema esterno che fornisce 5V per mezzo del connettore a 14 poli LTB1 o i connettori laterali SV1 e SV2, ovvero VCC_EXT. La scelta tra VCC_USB e VCC_EXT avviene per mezzo del Jumper JP1 nominato POW. Chiudendo il ponte tra 1-2 si ha il sistema alimentato per mezzo di VCC_EXT mentre chiudendo tra 2-3 si ha l'alimentazione via USB_PW. Al fine di rendere l'impostazione facile ed intuitiva, sul PCB è presente la serigrafia USB e EXT per evidenziare appunto USB_PW e VCC_EXT. Altra piccola accortezza per ricordare l'impostazione necessaria è nella posizione del Jumper, ovvero posizionandolo verso il lato del connettore USB, l'alimentatore viene dalla porta USB, mentre chiudendo verso il connettore LTB1, l'alimentazione viene dall'esterno.

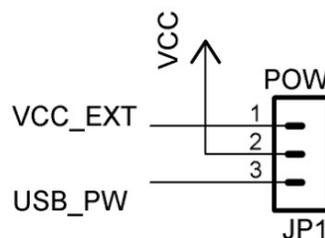


Figura 3: Connettore JP1 per selezionare l'alimentazione da USB o VCC_EXT.

In Figura 4 è riportato il connettore di espansione a 14 poli LTB1. Si noti come sia presente sia VCC_EXT che VCC. In particolare VCC, qualora JP1 sia posizionato su USB_PW, fornisce la tensione di alimentazione USB anche ad eventuali sistemi secondari. Quando JP1 sia posizionato su VCC_EXT, si ha che VCC e VCC_EXT sul connettore LTB1 coincidono per cui due pin sono a disposizione per poter supportare correnti maggiori.

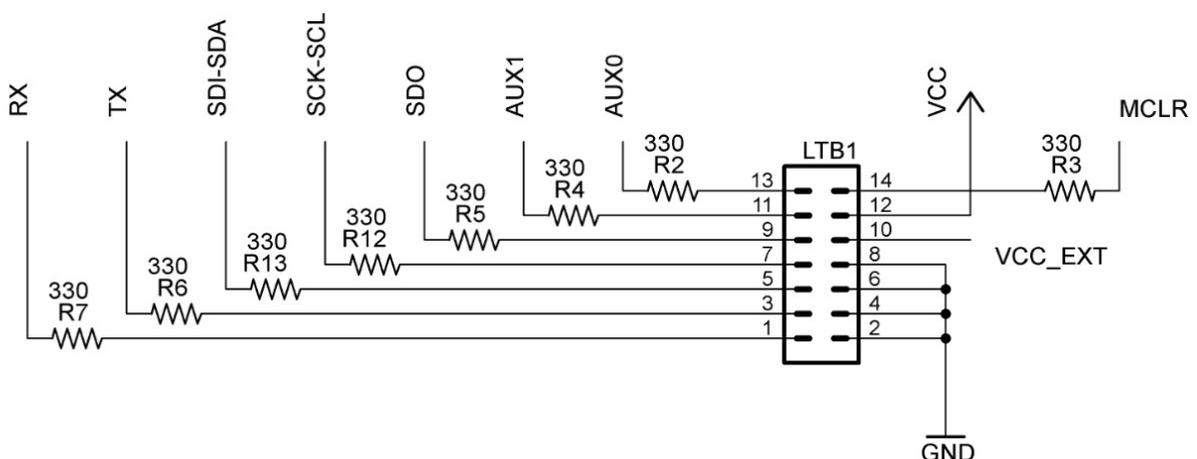


Figura 4: Connettore di espansione LTB1.

Il connettore LTB1 fornisce, oltre che all'alimentazione, il segnale di RESET ad un sistema secondario e le varie linee di comunicazione per mezzo dei protocolli seriali UART, SPI e I2C. Il

protocollo supportato dipende dalla particolare applicazione caricata nel PIC18F14K50, facilmente aggiornabile per mezzo del Bootloader interno e dell'applicazione *LaurTec PIC Bootloader* (PJ11005) scaricabile dal sito www.LaurTec.it.

In Figura 5 sono riportati i connettori di espansione SV1 e SV2 per mezzo dei quali è possibile collegare la scheda miniCOM USB ad un sistema secondario o collegare piccoli shield sulla scheda stessa. I connettori SV1-SV2 oltre alle linee disponibili nel connettore LTB1 possiede altri I/O per un utilizzo generico (maggiori dettagli sui connettori LTB1 e SV1-SV2 sono riportati nel paragrafo dedicato). In particolare è presente anche VCC e VCC_EXT. Si noti la simmetria delle alimentazioni VCC e VCC_EXT, la quale permette di collegare uno shield anche al contrario senza avere un corto circuito tra le alimentazioni.

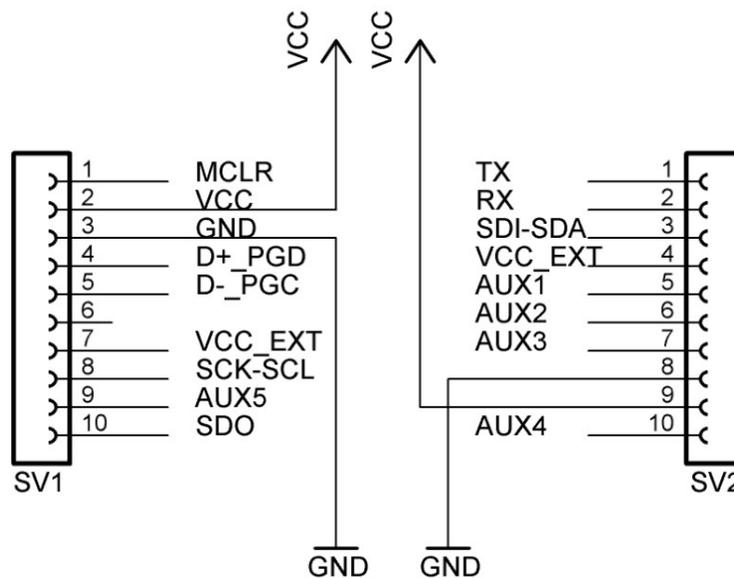


Figura 5: Connettori di espansione SV1 e SV2.

Limiti di corrente

Dal momento che è presente la possibilità di avere due tipi di alimentazione diversa è bene tenere a mente i seguenti casi, per i quali, i limiti di corrente sono differenti:

- **Alimentazione solo da USB**

Quando il sistema è alimentato solo da USB, il sistema deve assorbire al suo avvio non più di 100mA. Qualora si abbia bisogno di correnti maggiori è necessario richiederlo, e la richiesta deve essere approvata dall'Host o Hub. Solo dopo l'approvazione è possibile erogare la corrente richiesta fino al massimo di 500mA (limite per USB 2.0). Tale corrente rappresenta quella utilizzabile dal sistema e dalle periferiche connesse al connettore LTB1 o SV1-SV2. In questo caso la tensione VCC_EXT non è presente su LTB1 e SV1-SV2 visto che non è presente l'alimentazione esterna. Qualora il sistema collegato a miniCOM USB possieda un'alimentazione propria VCC_EXT potrebbe essere anche presente ma non viene utilizzata dal PIC18F14K50.

- **Utilizzo di un alimentatore esterno**

La tensione VCC_EXT di 5V può essere utilizzata solo se un sistema esterno la fornisce. La massima corrente che è possibile prelevare dal connettore LTB1 facendo uso dell'alimentazione proveniente da VCC_EXT è di 1A¹ se proviene da SV1 e SV2 mentre è di 200mA se proviene

¹ Naturalmente l'alimentatore esterno deve essere idoneo per erogare tale corrente. Nel bilancio della corrente del sistema esterno si deve considerare anche il consumo di miniCOM USB e relativi sub-sistemi collegati.

da LTB1.

Nota:

La scheda miniCOM USB non possiede alcun LED direttamente collegato a Vcc per cui per segnalare la presenza di Vcc è necessario utilizzare il LED1 o LED2 collegati rispettivamente a RC6 e RC4 (AUX6-AUX4). In questo modo qualora il sistema dovesse entrare in Suspend Mode è possibile farlo semplicemente spegnendo il LED (nella vecchia specifica USB, in Suspend Mode il sistema non dovrebbe assorbire più di 500uA mentre nelle specifiche attuali USB 2.0 tale specifica è stata modificata a 2mA). In molte applicazioni da laboratorio e sperimentali non si fa uso del Suspend Mode per cui si necessita di meno attenzioni.

Pulsanti

La scheda miniCOM USB possiede 1 pulsante per uso generico e un pulsante di RESET comodamente accessibili sul lato della scheda stessa. Nonostante la semplicità di questo hardware, la loro presenza permette di semplificare e accelerare lo sviluppo di nuove applicazioni. Infatti non è necessario avere fili volanti né tanto meno fare saldature. Lo schema elettrico relativo al pulsante è riportato in Figura 6.

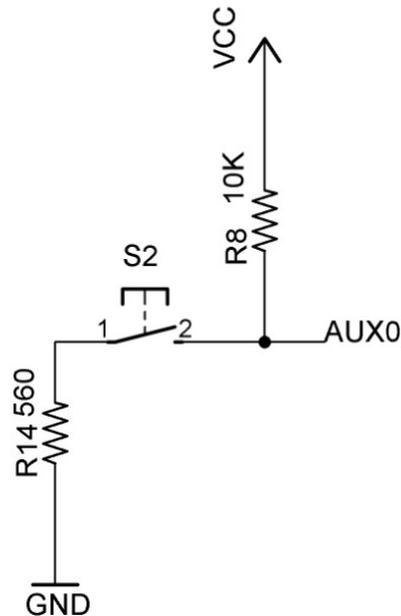


Figura 6: Schema elettrico del pulsante S2.

Si può osservare che il pulsante è collegato a AUX0 ovvero RC0. La scelta di questo pin discende dal fatto che la RC0 possiede l'opzione di INT0. Il resistore R8, rappresenta il resistore di pull-up mentre R14 è un resistore di protezione contro corto circuiti dell'uscita.

Sul PCB è previsto il montaggio verticale dei pulsanti. Qualora il sistema venga montato in una scatola di montaggio, utilizzare pulsanti verticali può risultare più pratico, visto che sono posti al lato del PCB stesso.

Il connettore USB sporge dal PCB per poter agevolare ulteriormente l'inserimento di miniCOM USB all'interno di una scatola di montaggio

Altre periferiche

Oltre alle periferiche appena descritte, il sistema miniCOM USB possiede altre piccole periferiche che è possibile utilizzare ed è bene riassumerle in qualche riga.

- Il tasto di RESET S1 permette di resettare il sistema qualora si sia verificato uno stallo dello stesso o si voglia semplicemente riavviare il sistema per altre ragioni. Tale pulsante è individuabile per mezzo della serigrafia RESET presente sul PCB. Il tasto di Reset è protetto da sovracorrenti per mezzo del resistore R1. Questo permette di proteggere il programmatore nel caso in cui, durante la programmazione, venisse premuto inavvertitamente il tasto di Reset. Infatti non tutti i programmatori sono protetti contro i corto circuiti. Il Bootloader Type_A per PIC18F14K50 fa uso del tasto RESET per avviare la programmazione per mezzo dell'applicazione *LaurTec PIC Bootloader*.
- Nonostante miniCOM USB sia progettata in maniera da ridurre al minimo l'Hardware superfluo e poter quindi essere utilizzata come scheda di espansione senza costi eccessivi e ingombri inutili, possiede 2 LED per uso generico.

Connettore di espansione

Il sistema miniCOM USB possiede a bordo l'hardware essenziale per ottenere periferiche USB di base. Hardware esterno può essere collegato per mezzo del connettore di espansione LTB1 o i connettori verticali SV1-SV2. In Figura 7 è riportata la piedinatura del connettore LTB1.

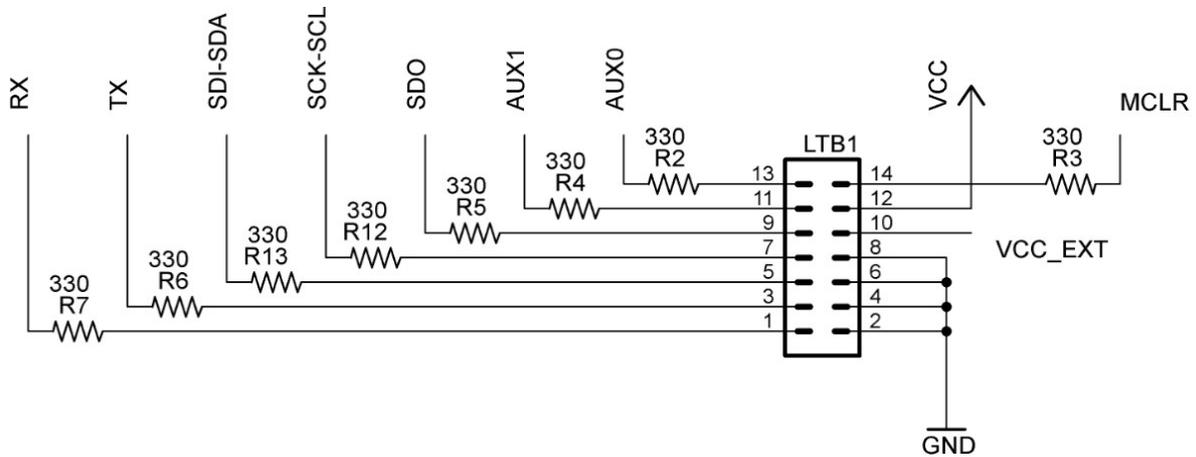


Figura 7: Connettore di espansione LTB1.

Oltre ai piedini di alimentazione e Reset, il connettore LTB1 fornisce varie linee di comunicazione per supportare vari protocolli seriali, ovvero UART, SPI e I2C. Sebbene la nomenclatura delle linee faccia riferimento ai relativi protocolli, a seconda dell'applicazione caricata nel PIC18F14K50 il loro utilizzo potrebbe differire. Per applicazioni generiche sono comunque presenti le linee AUX0 e AUX1, dove in particolare AUX0 è collegata al pulsante S2.

Dal momento che il connettore LTB1 può essere collegato a sistemi esterni per mezzo di un ribbon cable (cavo piatto) di lunghezza 10-20cm, sono previsti dei resistori da 330ohm su ogni linea, al fine di limitare eventuali extra-tensioni e campo elettromagnetico irradiato da variazioni troppo rapide della corrente. Al fine di limitare ulteriormente problemi dovuti ad irraggiamento e ad accoppiamento con campi elettromagnetici esterni, le linee di comunicazione sono intervallate con un segnale di massa.

In Figura 8 è riportato un dettaglio dei connettori di espansione verticali SV1 e SV2.

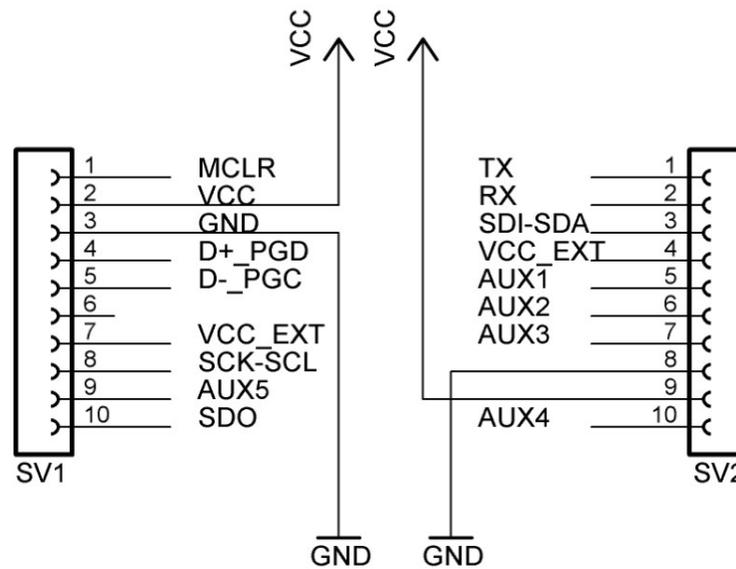


Figura 8: Connettori di espansione SV1 e SV2.

I connettori SV1 e SV2 possiedono tutte le linee riportate sul connettore LTB1 più altre linee ausiliari AUX1-AUX5. Si noti la simmetria delle alimentazioni. Tale simmetria permette di proteggere il sistema da eventuali cortocircuiti delle alimentazioni².

I connettori di espansione verticali, qualora siano di tipo femmina (di default), possono essere utilizzati per collegare semplici schede di espansione come shield. Usando connettori maschi si può collegare la scheda miniCOM USB direttamente su una scheda madre o sistema esterno più complesso.

Si noti che nel PIC18F14K50 i pin dedicati alla programmazione sono gli stessi utilizzati dal bus USB.

² A seconda delle impostazioni degli altri pin, l'inversione nel montaggio di una scheda di espansione potrebbe comunque creare problemi. La simmetria delle alimentazioni, limita i problemi associati a quest'ultima ma non copre tutte le casistiche di inversione dei pin.

Connettore di Programmazione e Debug

Un'altra importante caratteristica del sistema miniCOM USB è la sua integrazione con gli strumenti Microchip. In particolare per programmare la scheda si può fare uso del PICKIT 2, PICKIT 3 o programmatori compatibili. Il connettore SV1 ha in particolare la piedinatura disposta in maniera tale da poter collegare direttamente il programmatore. In particolare la freccia di riferimento del programmatore coincide con il pin 1 del connettore SV1

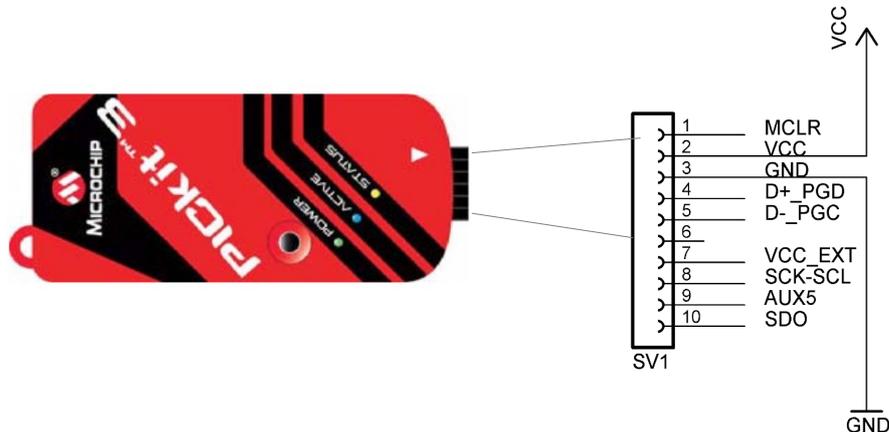


Figura 9: Connessione del programmatore al connettore SV1.

Confrontando infatti il connettore SV1 con la piedinatura del programmatore PICKIT 3 si può notare come questi coincidano.

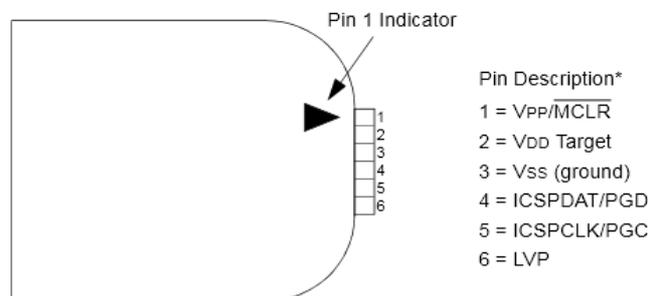


Figura 10: Piedinatura del programmatore PICKIT 2 e PICKIT 3

Il fatto di utilizzare il pin-out compatibile con i programmatori Microchip permette d'integrare facilmente il sistema con l'ambiente di sviluppo Microchip, MPLAB X[®]. Questo porta anche il vantaggio di avere sempre il supporto per gli ultimi modelli dei PIC a portata di mouse!

Si ricorda che la scheda miniCOM USB, oltre che poter essere programmata con il programmatore può essere programmata anche per mezzo dell'USB Bootloader. In particolare i KIT richiesti alla sezione servizi del sito www.LaurTec.it sono pre-programmati con il Bootloader, per cui i vari esempi presentati nel seguito di questo articolo possono essere caricati direttamente nel PIC18F14K50, senza l'utilizzo del programmatore. In fase di sviluppo è sempre raccomandabile avere un programmatore sotto mano al fine di poter riparare ad eventuali errori di programmazione. Se SV1 è un connettore femmina, al fine di poter inserire il programmatore in SV1 è necessario fare un piccolo adattatore maschio-maschio collegando tra loro due strip maschi diritti o a 90 gradi.

Volendo sacrificare qualche pin si potrebbe anche collegare un connettore di programmazione come Freedom II, ma dal momento che è possibile programmare la scheda anche per mezzo del Bootloader è meglio avere interamente il connettore SV1 ed effettuare la prima programmazione con un semplice adattatore.

Layout Periferiche

Durante la fase di sviluppo risulta di fondamentale importanza una guida rapida del Layout utilizzato per le varie periferiche. Sebbene questo possa essere ricavato anche dallo schema elettrico, aver a disposizione una Tabella in cui sono riportati direttamente i pin può risultare più utile. Si ricorda che a seconda delle scelte progettuali alcune parti della scheda potrebbero non essere disponibili. Si noti che non tutti i pin delle relative porte sono realmente implementati (o meglio non disponibili in uscita). In particolare RA2, RB0, RB1, RB2, RB3 non sono disponibili.

Pin	Periferica	Altre Funzioni
RA0	D+ USB	I/O, pin di programmazione PGD.
RA1	D+ USB	I/O, pin di programmazione PGC.
RA3	MCLR	I/O (usato dal bootloader Type A).
RA4	Cristallo	I/O o in. Analogico se si usa l'oscillatore interno.
RA5	Cristallo	I/O se si usa l'oscillatore interno.
RB4	SDI-SDA	I/O o in. Analogico se non si usa il modulo SPI o I2C.
RB5	RX	I/O o in. Analogico se non si usa il modulo UART.
RB6	SCK-SCL	I/O se non si usa il modulo SPI o I2C.
RB7	TX	I/O se non si usa il modulo UART.
RC0	AUX0 con pulsante S2	I/O o in Analogico se non si usa S2.
RC1	AUX1	I/O o in Analogico-Comparatore.
RC2	AUX2	I/O o in Analogico-Comparatore.
RC3	AUX3	I/O o in Analogico-Comparatore.
RC4	AUX4	I/O o uscita Comparatore.
RC5	AUX5	I/O o uscita PWM.
RC6	AUX6	I/O o in. Analogico.
RC7	SDO	I/O o in. Analogico se non si usa il modulo SPI.

Tabella 1: Tabella riassuntiva delle connessioni tra il PIC e le periferiche di sistema. Per una visione completa delle funzionalità di ogni pin si rimanda al datasheet del PIC utilizzato. In Tabella sono riportate solo le funzioni principali dei pin.

Istruzioni per il montaggio

La scheda di sviluppo miniCOM USB è un sistema realizzato su PCB a doppia faccia. Il PCB o il kit può essere richiesto alla sezione servizi del sito www.LaurTec.it. Il suo assemblaggio non risulta particolarmente complicato ma necessita certamente di attenzione.

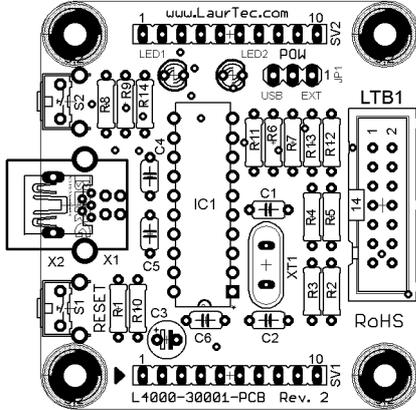


Figura 11: Serigrafia di miniCOM USB.

Per semplificare il montaggio, il PCB è realizzato con serigrafia dei componenti e relativo nome. Lo schema di montaggio è riportato in Figura 11. Il PCB reale di miniCOM USB è riportato in Figura 12.

Per il montaggio dei componenti è consigliabile seguire la regola legata all'altezza dei componenti stessi; dunque è bene iniziare dai resistori. Questa regola ha solamente un'utilità pratica associata al fatto che frequentemente, per fare le saldature, il PCB verrà posto sotto sopra. Per i resistori è bene accertarsi che il codice dei colori sia corretto. Nel saldare i diodi LED fare attenzione alla posizione dell'anodo e del catodo, in particolare la smussatura del contenitore del LED è visibile sulla serigrafia stessa³.

Successivamente si può procedere al montaggio delle capacità di filtro da 0.1uF e delle capacità ceramiche. Dal momento che il condensatore C5 da 0.47uF (identificato dal numero 474) è molto simile ai condensatori da 0.1uF (identificati dal numero 104) è bene isolarlo dagli altri e montarlo come primo condensatore. Si ricorda che l'ordine fin qui proposto non è obbligatorio ma può risultare pratico per il montaggio.

Il PIC18F14K50 (IC1) è bene montarlo su apposito zocchetto, in modo da poterlo cambiare ed evitare il suo danneggiamento in fase di saldatura. Gli integrati sono infatti sensibili alla temperatura, come d'altronde tutti i semiconduttori. Un'esposizione ad alte temperature può infatti portare alla rottura dell'integrato.

Ulteriore accorgimento va riservato per le capacità polarizzate, per le quali bisogna rispettare il verso legato alla polarizzazione. Sulla serigrafia del PCB di Figura 11 è facilmente individuabile il terminale positivo delle capacità indicato con un +. Se sul PCB non si dovesse ben leggere qualche carattere a causa di via⁴ far sempre riferimento alla Figura 11.

A montaggio completato miniCOM USB apparirà simile alla Figura 13. Si possono avere differenze da montaggio a montaggio a seconda dei componenti che si vuole montare o dalla versione del circuito stampato stesso⁵.

In particolare la scheda miniCOM USB supporta sia il connettore USB tipo B che mini B. Il kit completo viene fornito con il connettore mini B al fine di ottimizzare lo spazio sovrastante e rendere più agevole l'inserimento di schede di espansione. Il connettore tipo B ritorna più utile nel caso in cui si volesse utilizzare la scheda in una scatola di montaggio e non utilizzare le schede di espansione.

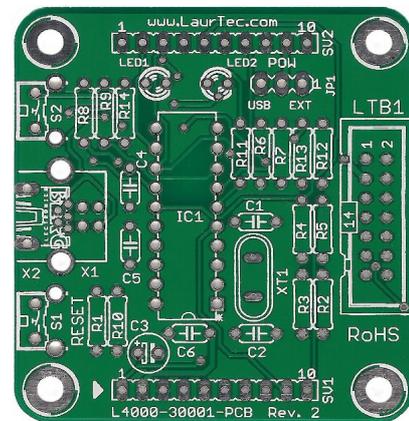


Figura 12: PCB miniCOM USB.

³ I diodi LED hanno il catodo segnalato da una smussatura sulla capsula del diodo stesso. Un altro modo per individuare l'anodo e il catodo è controllare la lunghezza dei terminali. L'anodo risulta il terminale più lungo.

⁴ I via rappresentano i fori metallizzati che permettono la realizzazione dei PCB a doppia faccia. Il loro uso permette infatti ad un segnale di poter passare da un lato all'altro del PCB.

⁵ Il circuito stampato può subire variazioni senza che il seguente articolo venga aggiornato.



Figura 13: Scheda miniCOM USB a montaggio ultimato con connettore tipo B.

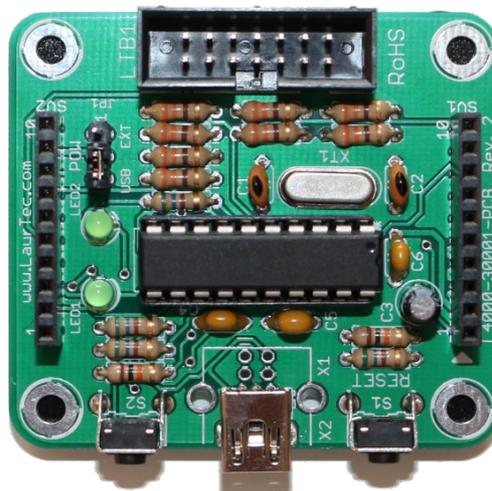


Figura 14: Scheda miniCOM USB a montaggio ultimato con connettore mini B.

Maggiori dettagli sugli strumenti e tecniche per il montaggio dei KIT elettronici possono essere trovati nel Tutorial AN9001-IT “*Strumenti e Tecniche per il montaggio di KIT elettronici*” scaricabile gratuitamente dal sito www.LaurTec.it.

Nota

- Per agevolare il montaggio la variante del Kit con connettore mini B viene fornito con il connettore già saldato. Il connettore tipo B standard viene invece fornito non saldato.
- Per garantire la compatibilità meccanica con tutte le schede di espansione è bene avere a disposizione almeno una scheda con connettore mini B.
- Il PIC18F14K50, incluso nei KIT, viene fornito con bootloader già installato.

Collaudo e messa in funzione

Una volta montato il sistema miniCOM USB, sebbene non sia particolarmente complesso, è bene fare delle misure preliminari prima della messa in funzione. Alcuni dettagli sul montaggio dei KIT elettronici e la loro messa in funzione possono essere trovati nel sito www.LaurTec.it nel Tutorial “Strumenti e Tecniche per il montaggio di KIT elettronici”.

Prima messa in funzione dopo il montaggio

1. Non montare IC1 prima dei test preliminari.
2. Controllare con il tester, prima di collegare la scheda all'alimentatore o ad una porta USB del PC, che il terminale GND e VCC non siano in corto.
3. Collegare il connettore USB accertandosi che il Jumper JP1 (POW) sia tra i pin 2-3 (ovvero posizione per alimentazioni USB). Misurare la tensione tra GND e VCC e constatare che sia 5V.
4. Ad alimentazione scollegata inserire il microcontrollore e caricare il programma di Test per mezzo del bootloader e l'applicazione *LaurTec PIC Bootloader* ovvero il file⁶:

00-Type-A-PIC18F14K50-miniCOM_USB_Test_Suite_v-1.0.hex.

All'avvio del programma di test si ha:

- Accensione LED1.
- Spegnimento LED1 e accensione LED 2.
- Spegnimento LED1 e LED2.
- Lampeggio LED1 e LED2.
- Premendo il tasto S1 di RESET si accende LED 1.
- Premendo il tasto S2 si accende LED 2.

Nota

- Non precedere al passo successivo qualora un test fallisca. In particolare se il punto 2 mostra un corto, bisogna cercare e rimuovere eventuali cortocircuiti derivanti da saldature.
- Per avviare il Bootloader Type-A staccare il connettore USB. Tenere premuto il tasto S1 (RESET) e ricollegare il cavo. Rilasciare poi il tasto S1

Messa in funzione ordinaria

Il normale funzionamento della scheda non richiede particolari attenzioni. A seconda delle esigenze è possibile installare un programma diverso facendo uso del Bootloader o del programmatore⁷.

⁶ Qualora abbiate un PIC18F14K50 senza Bootloader, caricare PIC18F14K50_Bootloader_Type_A.hex presente nel progetto *LaurTec PIC Bootloader*.

⁷ Si ricorda che programmi compilati per funzionare per il Bootloader non funzionano correttamente se installati per mezzo del programmatore piuttosto che il Bootloader.

Software ed esempi

Un semplice esempio da cui partire è senza dubbio lo stesso programma di test utilizzato per il collaudo della scheda. Infatti il programma mostra in maniera semplice come poter leggere i pulsanti e pilotare i LED. Il testo C18 Step by Step con i relativi esempi possono essere usati al fine di procedere passo passo all'utilizzo di ogni singola parte della scheda, iniziando con il programma "Hello World", per poi procedere con esempi più complessi.

In questa sezione sono mostrati esempi di utilizzo della scheda miniCOM USB con particolare attenzione all'utilizzo della porta USB. Gli esempi sono i seguenti:

- Classe CDC, Input Output Demo
- Classe CDC, miniCOM UART
- Classe HID, miniCOM Relays

Oltre a questi esempi è possibile caricare nella scheda anche gli esempi delle schede di espansione già disponibili e trattate nelle relative schede tecniche oltre che nel "Manuale Utente" PJ4000.

Si fa notare che gli esempi mostrati sono basati sui programmi di esempio della stessa Microchip⁸, opportunamente modificati al fine di mostrare le applicazioni con la scheda miniCOM USB. I programmi di esempio scaricabili dal sito LaurTec sono stati sviluppati utilizzando l'IDE MPLAB X e il compilatore C18⁹, i quali devono essere dunque propriamente installati sul proprio PC. Al fine di compilare i programmi di esempio è necessario scaricare le MAL della Microchip¹⁰. Accertarsi inoltre che i seguenti percorsi siano propriamente impostati nella sezione relativa ai percorsi da includere per il compilatore:

- percorso che punta al progetto
- [*percorso installazione MAL*]\Microchip

Al fine di limitare problemi di compilazione è bene salvare i progetti all'interno della cartella:

- [*percorso installazione MAL*]\USB

I percorsi potrebbero variare a seconda della versione MAL utilizzata.

Tutti gli esempi che seguiranno, sebbene possano essere compilati e modificati sono anche forniti precompilati e sono disponibili all'interno del progetto LaurTec_PIC_Bootloader_x.x.x scaricabile separatamente. Il progetto rappresenta la raccolta di tutti progetti per ogni scheda LaurTec che possiede il connettore USB. Il progetto LaurTec_PIC_Bootloader_x.x.x possiede inoltre la raccolta dei Linker file necessari per scrivere il progetto per il Bootloader, oltre ai bootloader stessi.

L'applicazione *LaurTec PIC Bootloader* permette di caricare programmi facendo uso del relativo bootloader HID fornito con il progetto. In particolare rappresenta il Bootloader Microchip con PID e VID modificati con quelli assegnati a LaurTec:

- PID = 0xFC5D
- VID = 0x04D8

⁸ La libreria MAL (Microchip Application Library) utilizzata è `microchip_solutions_v2013-02-15`.

⁹ Al tempo in cui è stato scritto l'articolo le MAL Microchip `microchip_solutions_v2013-02-15` non supportano ancora lo stack USB per XC8. Lo stack per PIC18 supporta solo il C18 mentre per i PIC a 16 e 32 bit c'è già il supporto dei compilatori XC16 e XC32.

¹⁰ Per maggiori informazioni sulle MAL, far riferimento al tutorial "Leggere e scrivere dati via USB" scaricabile dal sito www.LaurTec.it.

Esempio 1: Classe CDC, Input Output Demo

In questo primo esempio vedremo come poter leggere informazioni dalla scheda miniCOM USB simulando una comunicazione con la stessa per mezzo di una porta seriale, ovvero facendo uso della Classe CDC.

Caratteristiche e funzioni del progetto

Questo primo progetto permette di inviare alla porta seriale il valore del tasto premuto scrivendo sul terminale seriale il testo BUTTON: 1 o BUTTON: 2 a seconda che il tasto premuto sia S1 o S2. Inoltre premendo sulla tastiera del computer il numero 1 e 2 è possibile rispettivamente accendere il LED1 e il LED2. Per spegnere i due LED basta premere un tasto diverso da 1 e 2.

Caricare il progetto precompilato

Il progetto precompilato da caricare dalla directory LaurTec_PIC_Bootloader del progetto *LaurTec PIC Bootloader* è:

03-Type-A-PIC18F14K50-miniCOM_USB_CDC_Class_Input_Output_v-1.0

presente nella directory miniCOM USB. Il progetto senza modifiche può essere caricato dopo aver programmato il Bootloader nel PIC. Il Bootloader è già installato nel PIC18F14K50 qualora si sia acquistato un KIT di montaggio.

Modifica e comprensione del progetto

Il progetto presentato può essere modificato cambiando a proprio piacimento il testo o rilevando per esempio altri pulsanti o accendere altri LED. Per cambiare il progetto aprire e rendere attivo il progetto:

“01-CDC_miniCOM_USB_Input_Output_Demo”

Se necessario cambiare i percorsi delle librerie e compilarlo. Se tutti i percorsi sono stati propriamente settati il progetto verrà propriamente compilato e sarà possibile caricarlo nel PIC18F14K50 per mezzo del Bootloader¹¹.

Si noti che solo la funzione main è stata cambiata al fine di eseguire la funzione desiderata. Tutto il resto del programma appartiene al Framework USB Microchip. Prima del main si sono definite delle costanti solo a scopo di chiarezza del programma.

```
#define LED_1 LATCbits.LATC6
#define LED_2 LATCbits.LATC4

#define SWITCH_1 PORTAbits.RA3
#define SWITCH_2 PORTCbits.RC0

#define RELAY_1 PORTCbits.RC2
#define RELAY_2 PORTCbits.RC3

#define LED_ON 0x01
#define LED_OFF 0x00
```

¹¹ Il progetto è fornito con un file di linker modificato per poter lavorare con il bootloader Type-A fornito per il PIC18F14K50.

```
void main(void) {

    // Inbut data Buffer
    char dataIN = 0x00;

    // Output data array - used for the string
    char dataOUT[11];
    char data_sent_flag = 0;

    // USB module initialization
    InitializeSystem();

    USBDeviceAttach();

    // Output Text Initialization
    dataOUT[0] = 'B';
    dataOUT[1] = 'U';
    dataOUT[2] = 'T';
    dataOUT[3] = 'T';
    dataOUT[4] = 'O';
    dataOUT[5] = 'N';
    dataOUT[6] = ':';
    dataOUT[7] = ' ';
    dataOUT[9] = 13;
    dataOUT[10] = '\0';

    // Initialize PORTA
    LATA = 0x00;
    TRISA = 0xFF;

    // Initialize PORTB
    LATB = 0x00;
    TRISB = 0xFF;

    // Initialize PORTC
    LATC = 0x00;
    TRISC = 0b10100011;

    // Enable the pins as digital inputs
    ANSELbits.ANS4 = 0;
    ANSELbits.ANS6 = 0;
    ANSELbits.ANS7 = 0;
    ANSELHbits.ANS9 = 0;

    while (1) {
        CDCTxService();

        // Check if S1 is pressed and send the message if pressed
        if (SWITCH_1 == 0 && data_sent_flag ==0)
            if(USBUSARTIsTxTrfReady()) {
                dataOUT[8] = '1';
                putsUSBUSART(dataOUT);
                data_sent_flag = 1;
            }

        // Check if S2 is pressed and send the message if pressed
        if (SWITCH_2 == 0 && data_sent_flag ==0)
            if(USBUSARTIsTxTrfReady()) {
                dataOUT[8] = '2';
                putsUSBUSART(dataOUT);
            }
    }
}
```

```
        data_sent_flag = 1;
    }

    // Once I release the buttons I reset the flag for sending the message out
    if (SWITCH_1 == 1 && SWITCH_2 == 1)
        data_sent_flag = 0;

        dataIN = 0;

    // Read one byte from the input buffer
    getsUSBUSART (&dataIN, 1);

    // Check if button 1 or 2 is pressed
    if (dataIN == 0x31)
        LED_1 = LED_ON;

    if (dataIN == 0x32)
        LED_2 = LED_ON;

    if (dataIN > 0x32){
        LED_1 = LED_OFF;
        LED_2 = LED_OFF;
    }
}

}

} //end main
```

Come visibile dal programma, l'applicazione non fa altro che controllare la pressione dei pulsanti S1 e S2 ed invia lo stato della pressione via RS232 (porta emulata). La pressione dei pulsanti non è filtrata, per semplicità, per cui trasmissioni spurie potrebbero essere presenti. Per eliminarle è necessario realizzare un piccolo filtraggio sulla lettura dei pulsanti in maniera da non creare ritardi bloccanti.

Una volta caricato il programma nel PIC18F14K50 è possibile resettare la scheda per mezzo dell'applicazione *LaurTec PIC Bootloader*. Se è la prima volta che si collega la scheda alla porta USB del computer verrà richiesto il file .inf presente nella cartella del progetto stesso (cartella nominata inf) o nella cartella Boards → Drivers del progetto *LaurTec_PIC_Bootloader*. Una volta terminata l'installazione della scheda non verrà più richiesta l'installazione del driver.

A questo punto è possibile aprire un terminal per il controllo della porta seriale, in particolare si può far uso del programma *RS232 Terminal* come riportato in Figura 15. RS232 Terminal può essere scaricato gratuitamente dal sito www.LaurTec.it. Le impostazioni della porta seriale possono essere anche cambiate, infatti la scheda miniCOM USB si adatterà automaticamente alle nuove impostazioni. Nel caso specifico le impostazioni della porta COM sono : 115200, 8, 1, NONE, NONE.

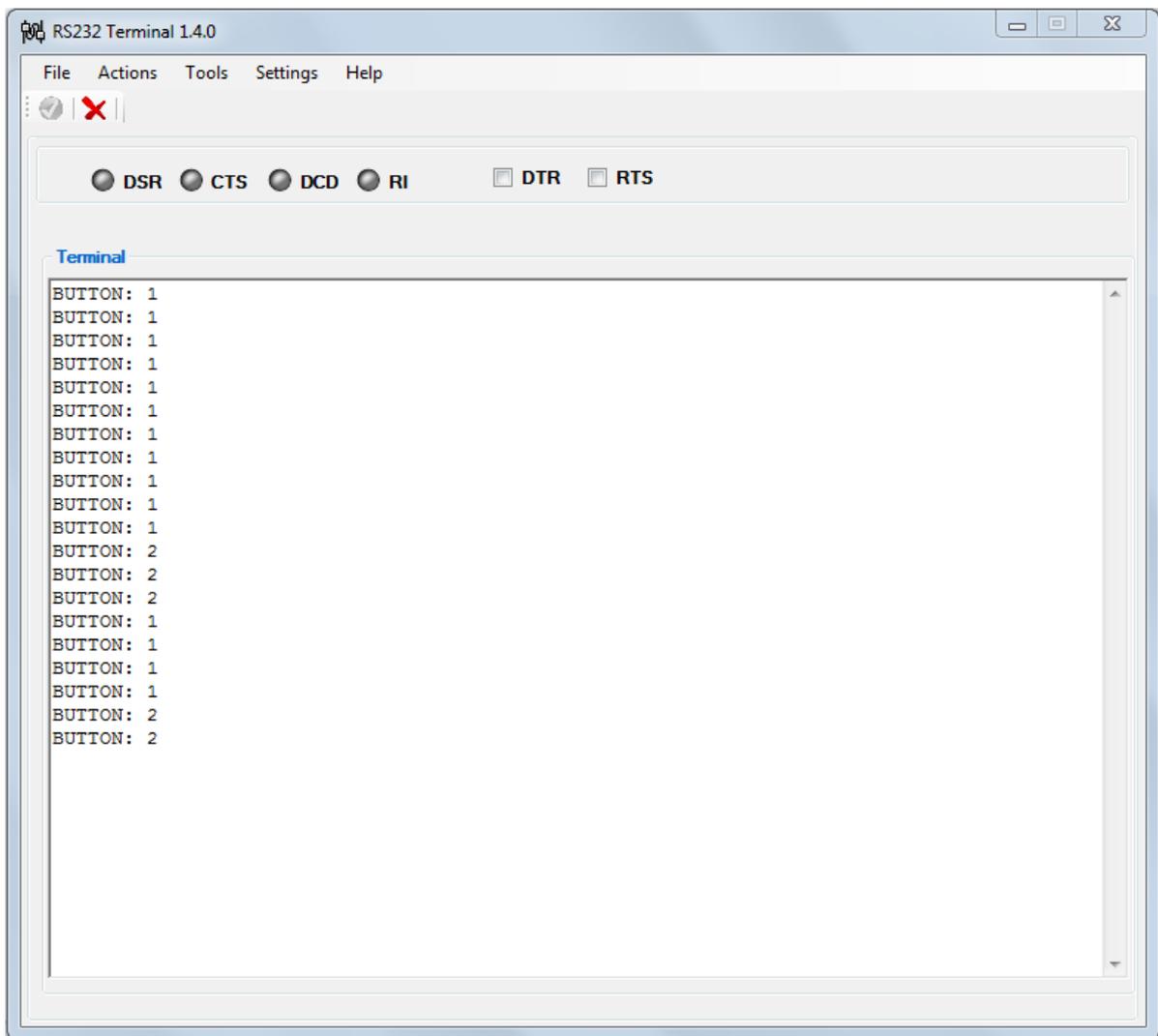


Figura 15: RS232 Terminal impostato per monitorare l'output della scheda miniCOM USB.

Come detto, il programma di esempio permette anche di accendere i due LED1-LED2 sulla scheda miniCOM USB, scrivendo semplicemente nel Terminal il numero 1 o 2. Premendo un numero diverso da 1, 2 i LED sono invece spenti.

Il programma può essere facilmente cambiato al fine di poter inviare informazioni diverse o pilotare un numero maggiore di uscite.

Esempio 2: miniCom UART

In questo secondo esempio viene realizzata una porta seriale emulata. Per testare il progetto sono necessarie due schede miniCOM USB in cui il pin TX della prima scheda deve essere collegato al pin RX della seconda scheda mentre il pin RX deve essere collegato al pin TX della seconda scheda. Le connessioni possono essere fatte direttamente dal connettore LTB1 o dal connettore laterale SV2.

In alternativa è possibile installare la scheda di espansione miniCOM RS232 per ottenere una porta virtuale seriale RS232 da collegare al PC. In questo secondo caso è necessario impostare la scheda miniCOM RS232 in posizione Null Modem ovvero con tutti i Jumper allineati nella posizione B.

Aprire due *RS232 Terminal* in modalità Terminal e selezionare i stessi parametri di Baud Rate sulle due porte. Attivare anche la funzione Local Echo nel Tab *Write Settings*.

Caratteristiche e funzioni del progetto

Questo secondo progetto permette di realizzare una porta seriale emulata. Tutti i dati ricevuti dalla scheda sulla linea RX sono presentati sul Terminal. Allo stesso modo tutti i dati scritti sul Terminal sono inviati sulla linea TX. Questo esempio permette di usare la scheda e comandare sistemi in cui siano presenti solo le linee TX e RX. In questo modo si può effettuare un semplice upgrade del sistema da UART ad USB. Aggiungendo il modulo miniCOM RS232 si ottiene una porta virtuale RS232.

Caricare il progetto precompilato

Il progetto precompilato da caricare dalla directory `LaurTec_PIC_Bootloader` è:

```
01-Type-A-PIC18F14K50-miniCOM_RS232_Demo_CDC_Class_v-1.0
```

presente nella directory miniCOM RS232. Il progetto senza modifiche può essere caricato dopo aver programmato il Bootloader nel PIC. Il Bootloader è già installato nel PIC18F14K50 qualora si sia acquistato un KIT di montaggio.

Una volta caricato il programma nel PIC18F14K50 è possibile resettare la scheda per mezzo dell'applicazione *LaurTec PIC Bootloader*. Se è la prima volta che si collega la scheda alla porta USB del computer verrà richiesto il file `.inf` presente nella cartella del progetto stesso (cartella nominata `inf`) o nella cartella *Boards* → *Drivers* del progetto *LaurTec PIC Bootloader*. Una volta terminata l'installazione della scheda non verrà più richiesta l'installazione del driver.

Se il software è propriamente installato, dopo il Reset della scheda il LED 2 lampeggerà.

Modifica e comprensione del progetto

Il progetto non richiede alcuna modifica qualora si voglia usare la porta emulata con le sole linee RX e TX. Qualora si voglia avere anche l'opzione Data Flow Hardware Control, ovvero per mezzo delle linee CTS e RTS è necessario ricompilare il progetto e definire:

```
USB_CDC_SUPPORT_HARDWARE_FLOW_CONTROL
```

Il progetto da ricompilare è:

“01-CDC_miniCOM_RS232_Serial_Emulator”

Se necessario cambiare i percorsi delle librerie e compilarlo. Se tutti i percorsi sono stati propriamente settati il progetto verrà propriamente compilato e sarà possibile caricarlo nel PIC18F14K50 per mezzo del Bootloader¹².

Un esempio di Output con due terminali *RS232 Terminal* è riportato in Figura 16.

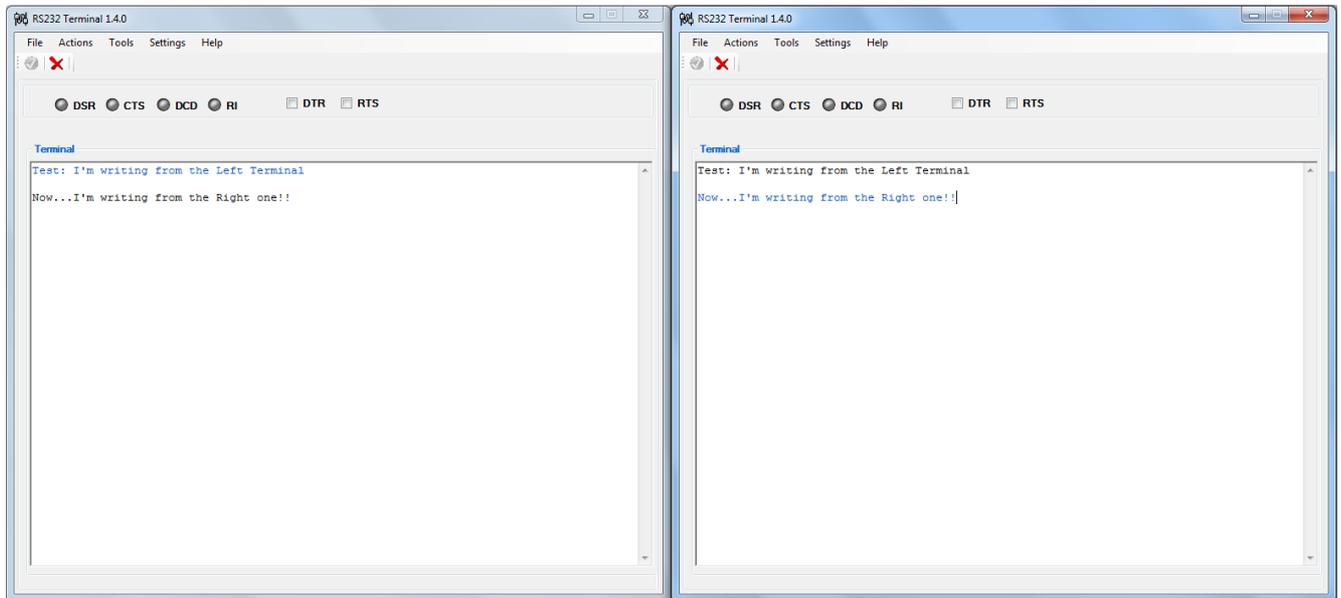


Figura 16: Due RS232 Terminal utilizzati per monitorare due schede miniCOM USB.

Tutto il testo scritto in un Terminal comparirà nell'altro.

In mancanza di due schede o della scheda miniCOM RS232 è possibile anche fare un semplice loop-back sulla singola scheda, ovvero collegando tra loro la linea TX e RX. In questo modo abilitando la funzione Local Echo, dal momento che ogni carattere inviato viene ricevuto dallo stesso Terminal, si ha la scritta “doppia” di Figura 17.

¹² Il progetto è fornito con un file di linker modificato per poter lavorare con il Bootloader Type-A fornito per il PIC18F14K50.

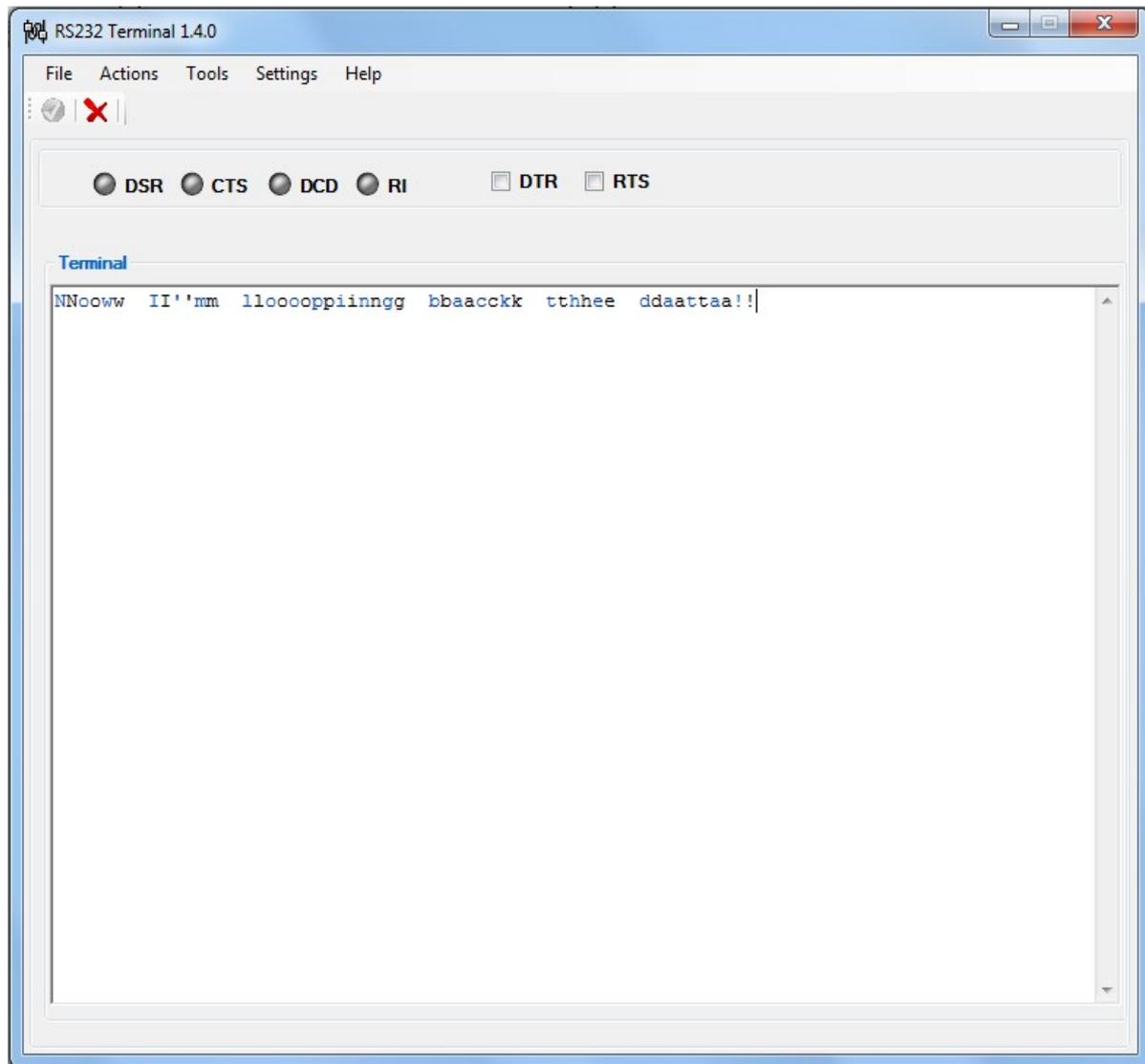


Figura 17: *RS232 Terminal utilizzato per monitorare miniCOM USB in loop-back.*

Esempio 3: Classe HID, miniCOM Relay

In questo terzo esempio vedremo un'applicazione che fa uso della porta USB classe HID¹³ (Human Interface Device). Tale classe è la stessa utilizzata per periferiche come il mouse e la tastiera. La caratteristica principale è che il sistema miniCOM USB programmato come periferica HID non richiede alcuna installazione di Driver né di file inf. Il sistema una volta collegato al PC è pronto per l'uso! Periferiche di tipo HID hanno anche il vantaggio di essere facilmente portabili tra vari sistemi operativi (Windows, Linux, MAC), dal momento che ogni sistema possiede già i driver per periferiche HID. Una periferica HID ha però lo svantaggio di non poter comunicare con il PC come una porta seriale COM (come abbiamo fatto negli esempi precedenti). Per tale ragione al fine di leggere i dati del sistema è necessario creare un'applicazione ad hoc sviluppata per il progetto, ovvero l'applicazione *miniCOM Relays*.

Caratteristiche e funzioni del progetto

Questo progetto permette di comandare direttamente i due Relay della scheda miniCOM Relays che deve essere pertanto montata direttamente sulla scheda miniCOM USB rispettando il verso del connettore SV1 e SV2. I Relay possono essere attivati e disattivati semplicemente facendo uso dell'interfaccia grafica di Figura 18.

Caricare il progetto precompilato

Il progetto precompilato da caricare dalla directory LaurTec_PIC_Bootloader è:

02-Type-A-PIC18F14K50-miniCOM_Relays_HID_Class_v-1.0

presente nella directory miniCOM Relays. In questa stessa directory si trova la GUI per il controllo della scheda.

Una volta caricato il programma nel PIC18F14K50 è possibile resettare la scheda per mezzo dell'applicazione *LaurTec PIC Bootloader*.

Modifica e comprensione del progetto

Il progetto non richiede alcuna modifica qualora si voglia usare l'esempio base. Qualora si volessero apportare delle modifiche bisogna notare che l'applicazione questa volta è scritta nella funzione `ProcessIO()` richiamata dalla funzione `main`. Si noti che l'intera applicazione è molto semplice e si basa semplicemente nel controllare il contenuto del buffer *ReceivedDataBuffer* e nello scrivere nel buffer *ToSendDataBuffer* opportunamente letti/aggiornati una volta verificata la ricezione e la possibilità di inviare dati. Questi due buffer sono di dimensioni pari a 64Byte.

Il progetto da ricompilare è:

“02-HID-miniCOM_Relays”

Se necessario cambiare i percorsi delle librerie e compilarlo. Se tutti i percorsi sono stati propriamente settati il progetto verrà propriamente compilato e sarà possibile caricarlo nel PIC18F14K50 per mezzo del bootloader¹⁴.

¹³ Si ricorda che la Classe HID fa uso dell'Interrupt Transfer Mode dello standard USB. Pacchetti di 64Byte sono inviati in maniera regolare ogni millisecondo, permettendo un data throughput di 64KB/s.

¹⁴ Il progetto è fornito con un file di linker modificato per poter lavorare con il bootloader Type-A fornito per il PIC18F14K50.

Una volta compilato il programma di esempio e caricato lo stesso nel PIC18F14K50 è possibile collegare la scheda al PC. Come detto non verrà richiesto alcun driver anche se non si sono eseguiti gli esempi 1 e 2. Per visualizzare i dati inviati dal sistema miniCOM USB, impostata come periferica HID è necessario eseguire l'applicazione miniCOM Relays come riportato in Figura 18. Diversamente dagli esempi precedenti non è necessario impostare alcuna “porta di comunicazione”, la scheda viene automaticamente rilevata alla connessione.

Nell'esempio di Figura 18 è mostrato il caso in cui il Relay 1 è attivato e il tasto S1 è stato premuto sulla scheda.



Figura 18: Esempio d'uso dell'applicazione miniCOM Relays.

I Relay possono essere attivati o disattivati semplicemente attivando/disattivando il relativo check box. Il sorgente dell'applicazione miniCOM Relays può essere variato al fine di realizzare una interfaccia più accattivante e poter gestire anche più Relay direttamente da PC.

...non ci scordiamo del PID, VID

Tutti contenti abbiamo i nostri progetti funzionanti...però non bisogna dimenticarsi del VID (Vendor ID) e PID (Product ID), visto che ogni progetto, se commerciale, deve avere una combinazione unica VID, PID. I progettisti che vogliono mettere sul mercato un prodotto devono attenersi alle regole del consorzio USB, che rilascia sotto pagamento il numero VID, accompagnato da 65536 combinazioni di PID, ovvero 2 alla 16 combinazioni. Il prezzo del VID dipende dal tipo di contratto che viene stipulato ma il costo minimo è di circa 2000\$. Fortunatamente per casi particolari la Microchip come anche altre case costruttrici di dispositivi USB possiedono un programma di Sublicensing per mezzo del quale è possibile utilizzare il VID del produttore ed ottenere un PID. Tra le varie condizioni richieste nell'accordo di Sublicensing vi è la condizione che VID e PID siano utilizzati su prodotti utilizzando il componente del produttore.

Per il driver della classe CDC il PID e VID sono contenuti all'interno del file .inf utilizzato per installare propriamente il driver. Il formato con cui lo si trova è USB\VID_xxxx&PID_yyyy. Un esempio è:

```
%DESCRIPTION%=DriverInstall, USB\VID_04D8&PID_F5B9
```

che è appunto il VID e PID utilizzato dalla LaurTec all'interno del file .inf. L'informazione del VID e PID oltre che essere propriamente scritti all'interno del file .inf devono essere scritti anche all'interno del nostro codice (questo vale anche per un dispositivo HID). Infatti il dispositivo hardware comunica queste informazioni all'Host (PC), che le utilizza per la ricerca del driver opportuno, ovvero il driver che secondo il file .inf supporta il VID e PID del dispositivo hardware (in realtà nella scelta del driver concorre anche il tipo di classe del dispositivo). Nella libreria USB versione 2.9 della Microchip, il VID e il PID possono essere trovati all'intero del file usb_descriptors.c, come sotto riportato (numeri rossi).

```
/* Device Descriptor */
ROM USB_DEVICE_DESCRIPTOR device_dsc =
{
    0x12,                // Size of this descriptor in bytes
    USB_DESCRIPTOR_DEVICE, // DEVICE descriptor type
    0x0200,              // USB Spec Release Number in BCD format
    CDC_DEVICE,         // Class Code
    0x00,                // Subclass code
    0x00,                // Protocol code
    USB_EP0_BUFF_SIZE,  // Max packet size for EP0, see usb_config.h
    0x04D8,              // Vendor ID
    0xF5B9,              // Product ID: CDC RS-232 Emulation Demo
    0x0100,              // Device release number in BCD format
    0x01,                // Manufacturer string index
    0x02,                // Product string index
    0x00,                // Device serial number string index
    0x01                // Number of possible configurations
};
```

Nella libreria USB versione 1.2 il VID e il PID possono essere trovati all'intero del file usbdsc.c.

In questi file, oltre al VID e PID è possibile anche cambiare gli identificatori che descrivono il tipo di dispositivo e il produttore.

Nota:

Le schede e software di esempi LaurTec per le schede della serie miniCOM sono forniti con una combinazione PID-VID unica fornita dalla Microchip ed assegnata a LaurTec, per cui non possono essere utilizzate in applicazioni commerciali riadattando gli esempi proposti nei Progetti LaurTec.

A partire da Ottobre 2013 LaurTec utilizza il seguente standard:

PIC18 bootloader

- PID = 0xFC5D
- VID = 0x04D8

Classe CDC

- PID = 0xF5B9
- VID = 0x04D8

Classe HID

- PID = 0xF750
- VID = 0x04D8

L'utilizzo di PID VID differenti permette di discriminare il Bootloader e relativi driver necessari senza creare conflitti tra driver appartenenti a classi differenti e combinazioni PID VID uguali.

Indice Alfabetico

A		Product ID.....	31
Alimentazione.....	4	programmatori Microchip.....	16
Altre periferiche.....	13	pulsante.....	12
Assorbimento.....	4	Pulsanti.....	7
C		Q	
Circuiti Integrati.....	7	Quarzi.....	7
clock.....	8	R	
clock interno.....	8	ReceivedDataBuffer.....	29
Condensatori.....	7	RESET.....	13
Connettore di espansione.....	14	Resistori.....	7
Connettori.....	7	RS232 Terminal.....	24
D		S	
Dimensioni.....	4	schema di montaggio.....	18
Diodi.....	7	schema elettrico.....	5
DriverInstall.....	31	Serial Interface Engine.....	5
H		serigrafia.....	18
Hardware esterno.....	14	SIE.....	5
HID.....	29	strumenti Microchip.....	16
Human Interface Device.....	29	Suspend Mode.....	11
L		T	
Layout Periferiche.....	17	ToSendDataBuffer.....	29
Limiti di corrente.....	10	U	
Lista Componenti.....	7	USB.....	31
LTB1.....	14	USB classe HID.....	29
M		usbds.c.....	31
MPLAB X®.....	16	Utilizzo di un alimentatore esterno.....	10
P		V	
Part Number.....	4	Vendor ID.....	31
PCB.....	18	Versione.....	4
Peso.....	4	via.....	18
PID.....	31	VID.....	31
ProcessIO.....	29		

Bibliografia

[1] www.LaurTec.com : sito ufficiale di EasyUSB dove poter scaricare ogni aggiornamento e applicazione. Il PCB di EasyUSB è reso disponibile alla sezione servizi sotto donazione di supporto al sito stesso.

[2] www.microchip.com : sito dove scaricare i datasheet del PIC18F4550 e il Framework USB.

[3] www.usb.org : sito ufficiale del consorzio USB

History

Data	Versione	Nome	Descrizione Cambiamento
18.11.13	1.0a	Mauro Laurenti	Aggiornata lista componenti. Cambiato il nome Q1 in XT1. Aggiunto il valore mancante di R14.
29.06.13	1.0	Mauro Laurenti	Versione Originale.