

LaurTec

Freedom Light

Autore : *Mauro Laurenti*

ID: PJ7012-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
Smaltimento	4
Analisi del progetto	5
Il microcontrollore.....	5
Il quarzo	8
L'alimentatore.....	8
Modulo LCD alfanumerico 16x2.....	10
Pulsanti.....	11
Altre periferiche.....	12
Connettore di espansione.....	13
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.....	21
Software ed esempi	22
Esempio 1: Classe CDC, Input Output Demo	23
Esempio 2: Classe CDC, Analog Demo	26
Esempio 3: Classe HID, Input Output Demo	30
...non ci scordiamo del PID, VID	31
Bibliografia	33
History	34

Introduzione

La porta USB è ormai il sistema più usato per lo scambio d'informazione 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 Freedom Light, fonde le caratteristiche della scheda Freedom II e EasyUSB mantenendo l'essenziale per iniziare a programmare i microcontrollori PIC18 e agevolare lo studio della porta USB. Il connettore di espansione compatibile con Freedom II e EasyUSB la rende anche idonea quale valida alternativa economica per avere una mini piattaforma di sviluppo con la quale esercitarsi.

Specifiche Tecniche

Alimentazione : 9V-12V DC 

Assorbimento : 160mA medio @ 12V (LCD con retroilluminazione attiva)

Dimensioni : 73 x 84 mm

Part Number : L1000-30012

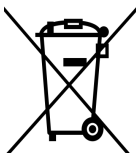
Versione : 3

Peso : 53g (90g incluso LCD)

Il sistema Freedom Light supporta il seguente hardware:

- Supporto USB 2.0 low speed e full speed
- 2 pulsanti + Reset
- 2 Led di utilizzo generico
- LCD alfanumerico 16x2
- Trimmer per contrasto LCD o ingresso analogico
- Buzzer
- Connettore espansione 40 pin
- 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 Freedom Light. Dalle dimensioni ridotte e dall'hardware disponibile 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 e display alfanumerico LCD 16x2, anche allo sviluppo di piccoli sistemi di prova, per poi raggiungere una qualsivoglia complessità grazie al connettore di espansione EX1, compatibile con la scheda di sviluppo Freedom II e EasyUSB.

Per una corretta descrizione e facile comprensione dello schema, l'hardware verrà introdotto a blocchi funzionali.

Il microcontrollore

Come prima cosa cerchiamo di capire con quali microcontrollori sia possibile utilizzare la scheda di sviluppo Freedom Light. La scheda nasce con lo scopo primario di permettere lo sviluppo di applicazioni USB, pertanto è ottimizzata per il supporto di microcontrollori della famiglia PIC18 a 40 pin che possiedono il modulo USB ovvero il modulo SIE (Serial Interface Engine), in particolare per il PIC18F4455, PIC18F4550 e PIC18F4553. Ciononostante altri PIC potrebbero essere utilizzati allo scopo di realizzare sistemi di altra natura, come per esempio il PIC16F877, ma non tutto l'hardware potrebbe risultare utilizzabile, come per esempio il modulo USB.

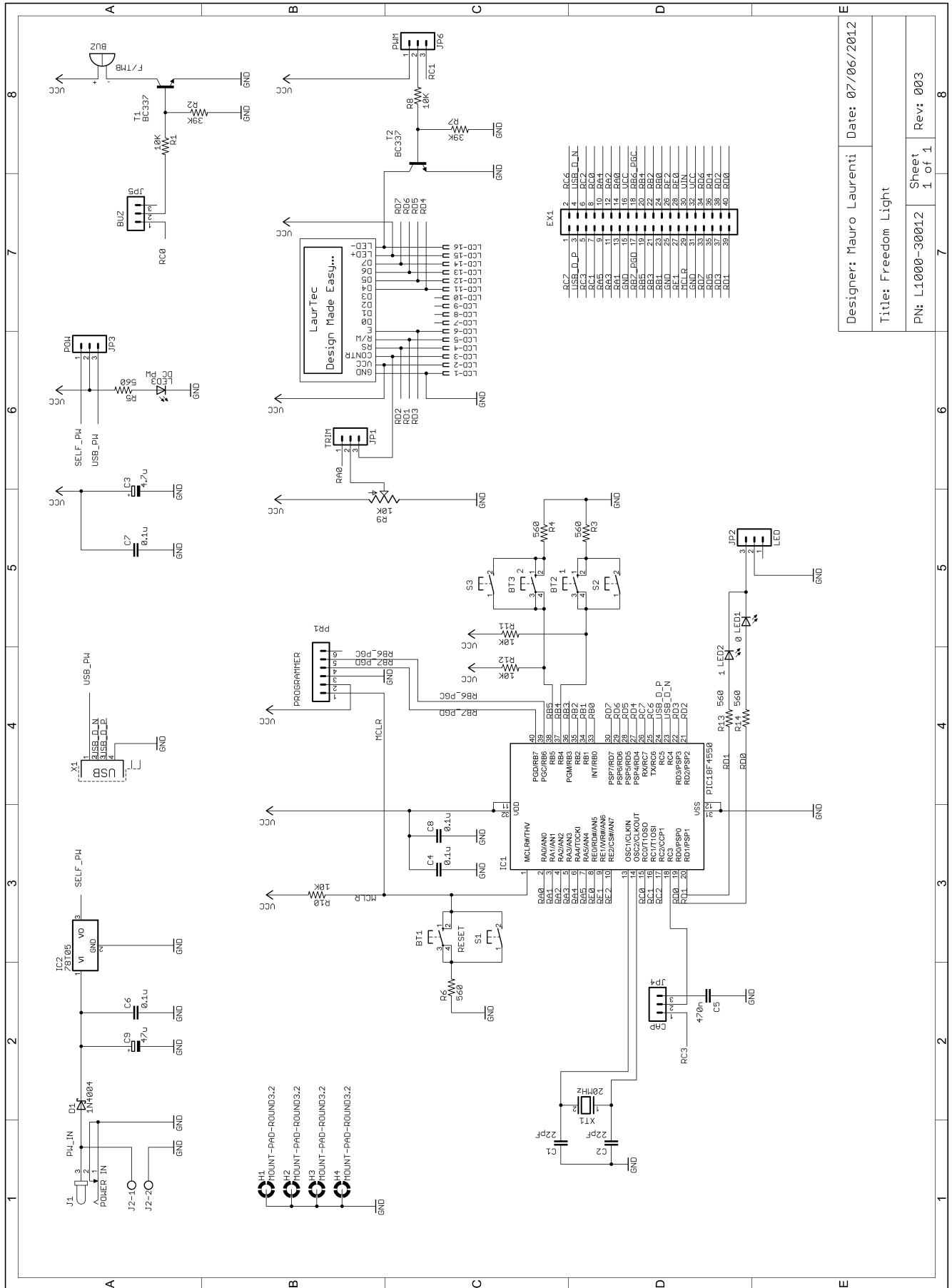


Figura 1: Schema elettrico di Freedom Light

Lista Componenti

Resistori

R1 = 10K Ω %5 1/4W
R2 = 39K Ω %5 1/4W
R3 = 560 Ω %5 1/4W
R4 = 560 Ω %5 1/4W
R5 = 560 Ω %5 1/4W
R6 = 560 Ω %5 1/4W
R7 = 39K Ω %5 1/4W
R8 = 10K Ω %5 1/4W
R9 = 10K Ω Trimmer
R10 = 10K Ω %5 1/4W
R11 = 10K Ω %5 1/4W
R12 = 10K Ω %5 1/4W
R13 = 560 Ω %5 1/4W
R14 = 560 Ω %5 1/4W

Condensatori

C1 = 22pF ceramico
C2 = 22pF ceramico
C3 = 4.7 μ F elettrolitico 50V
C4 = 0.1 μ F ceramico 50V
C5 = 470nF ceramico 50V
C6 = 0.1 μ F ceramico 50V
C7 = 0.1 μ F ceramico 50V
C8 = 0.1 μ F ceramico 50V
C9 = 47 μ F elettrolitico 35V

Circuiti Integrati

IC1 = PIC18F4550
IC2 = 7805

Transistor

T1 = Transistor NPN BC337
T2 = Transistor NPN BC337

Diodi

LED1 = LED 3mm verde
LED2 = LED 3mm verde
LED3 = LED 3mm rosso
D1 = 1N4004

Quarzi

XT1 = 20MHz

Pulsanti

BT1 = micro-pulsante per PCB verticale (opzionale)
BT2 = micro-pulsante per PCB verticale (opzionale)
BT3 = micro-pulsante per PCB verticale (opzionale)
S1 = micro-pulsante per PCB
S2 = micro-pulsante per PCB
S3 = micro-pulsante per PCB

Connettori

EX1 = ICD 40 pin (maschio)
X1 = Connettore USB per periferiche (tipo B)
J1 = Connettore cilindrico alimentatore 2.1mm
J2 = Connettore Conwago (opzionale)
JP1 -JP6 = Jumper 3 pin maschio (passo 2.54)
JPx = Jumper femmina (Qty. 6)
PR1 = Jumper 6 pin 90 gradi
LCDx = Connettore 16 pin maschio (passo 2.54)

Altro

LCD = LCD 16x2 controllore HD44780
BUZ = Buzzer 5V auto-oscillante
GM = 4 gommini spaziatori

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 PIC18F4550, hanno la possibilità di generare un clock interno. Questo significa che lo schema base per utilizzare il microcontrollore consiste semplicemente nella connessione dell'alimentazione e della circuiteria di Reset¹. Ciononostante la presenza del quarzo può in ogni modo essere necessaria per quelle applicazioni in cui la stabilità del clock è importante (per esempio l'utilizzo del modulo USB). Infatti, per mezzo del quarzo esterno, è possibile ottenere una stabilità del clock superiore a quella ottenibile utilizzando l'oscillatore interno.

L'alimentatore

La sezione di alimentazione della scheda Freedom Light è molto più semplice se paragonata con la scheda EasyUSB, questo è a favore della semplicità di uso. Lo schema riportato in Figura 2 mostra che tale sezione è composta del solo regolatore lineare 7805 al fine di stabilizzare la tensione in ingresso a 5V, necessari per il corretto funzionamento del microcontrollore.

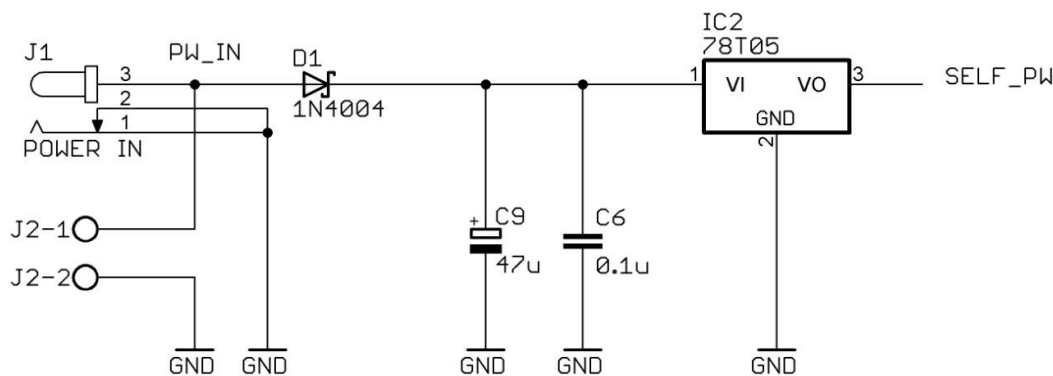


Figura 2: Schema elettrico della sezione di alimentazione.

Nonostante la semplicità dello schema, sono presenti alcuni accorgimenti:

- Il diodo di protezione D1 permette di proteggere il sistema da eventuali inversioni di polarità dell'alimentatore.
- Il connettore di alimentazione può essere sia J1 che un normale connettore a vite (tipo Conwago)
- La scheda può essere alimentata sia per mezzo di un alimentatore esterno che per mezzo della porta USB grazie al Jumper JP3, come riportato in Figura 3. In particolare posizionando JP3 tra i pin 1-2, l'alimentazione proviene dall'alimentatore esterno, mentre collegando JP3 tra 2-3 l'alimentazione viene prelevata dalla porta USB.

¹ Dal momento che i PIC possiedono una circuiteria di Reset interna, quella esterna non è obbligatoria....ma dal momento che non è escluso che il nostro programma si blocchi, è bene prevedere un Reset esterno.

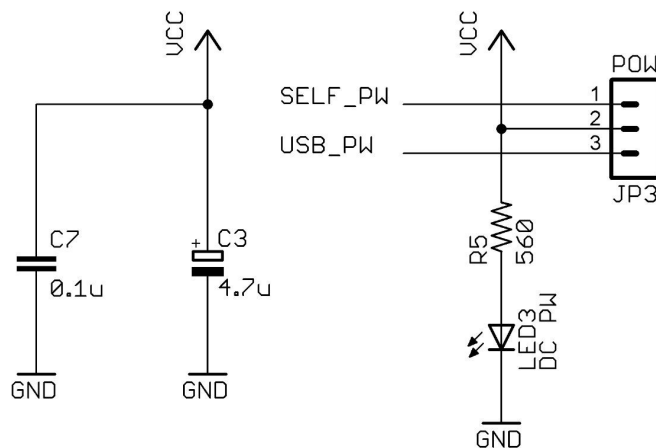


Figura 3: Schema elettrico relativo alla scelta dell'alimentazione Esterna o USB.

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. Tale corrente rappresenta quella utilizzabile dal sistema Freedom Light e dalle periferiche connesse al connettore EX1. In questo caso la tensione 9V-12V non è presente su EX1 visto che non è presente l'alimentatore esterno.

- **Utilizzo di un alimentatore esterno**

La tensione dell'alimentatore 9V-12V può naturalmente essere utilizzata solo se l'alimentatore di rete è presente. La massima corrente che è possibile prelevare dal connettore EX1 facendo uso dell'alimentazione proveniente da POWER_IN è di 1A².

Nota:

Diversamente dalla scheda EasyUSB, Freedom Light ha il LED3 per l'alimentazione collegato direttamente a 5V. Quindi non è necessario pilotarlo per segnalare l'alimentazione. D'altro lato però se si volesse realizzare un sistema che in stato di Suspend Mode rientri nelle specifiche USB (500uA max) è necessario rimuoverlo, e usare LED1 e/o LED2 per la segnalazione dell'alimentazione. Infatti la corrente erogata dal LED3 è di circa 10mA ovvero molto maggiore dei 500uA consentiti dal Suspend Mode. In molte applicazioni da laboratorio e sperimentali non si fa uso del Suspend Mode per cui è meglio privilegiare un LED3 che segnala direttamente l'alimentazione piuttosto che doverne accendere uno per mezzo del programma.

² Naturalmente l'alimentatore esterno deve essere idoneo per erogare tale corrente. Qualora si faccia uso di una tensione superiore a 9V e si utilizzano correnti superiori a 500mA, è bene prevedere un dissipatore di raffreddamento esterno.

Modulo LCD alfanumerico 16x2

Nonostante le ridotte dimensioni della scheda è presente anche un modulo LCD alfanumerico 16x2, per mezzo del quale è possibile scrivere messaggi e realizzare semplici menu. In Figura 4 è riportato il dettaglio dello schema elettrico relativo al modulo LCD.

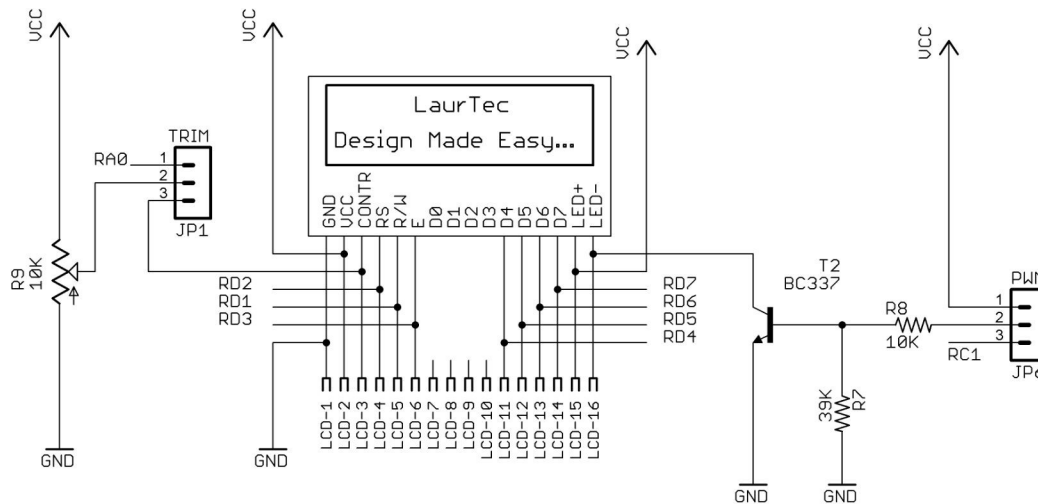


Figura 4: Schema del modulo LCD alfanumerico.

Il Trimmer per il contrasto è montato sulla scheda e grazie al foro sottostante può essere variato anche quando il modulo LCD è montato. Il Trimmer è collegato anche al Jumper JP1 poiché potrebbe essere utilizzato anche per altre applicazioni, quali per esempio testare gli ingressi analogici. In particolare JP1 permette di collegare il Trimmer o al modulo LCD (posizione 2-3) o all'ingresso analogico del microcontrollore AN0 (posizione 1-2).

Il Jumper JP6 permette di collegare la retroilluminazione o direttamente a Vcc, ovvero sempre accesa (posizione 1-2), o al pin RC1 del microcontrollore. Il pin RC1 oltre che a poter accendere e spegnere la retroilluminazione può, grazie alla presenza del modulo PWM, attivabile sul pin RC1, controllare anche al sua intensità.

Pulsanti

La scheda Freedom Light possiede 2 pulsanti, comodamente accessibili, che possono essere utilizzati per applicazioni generiche. 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 ai pulsanti è riportato in Figura 5.

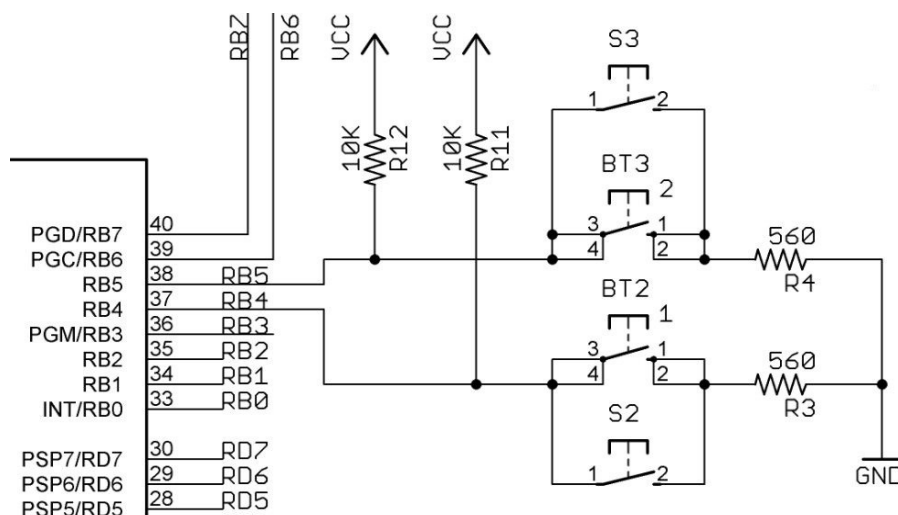


Figura 5: Schema elettrico dei pulsanti

Si può osservare che i pulsanti sono collegati rispettivamente agli ingressi RB4-RB5. La scelta di questi pin discende dal fatto che la PORTB possiede linee d'interrupt sulla variazione del valore logico del pin ed inoltre il Bootloader USB della Microchip possiede di default il controllo del pin RB4 al fine di verificare la necessità o meno di avviare il Bootloader. Un'altra peculiarità della PORTB è che possiede resistori di pull-up interni, dunque si possono risparmiare i resistori di pull-up esterni. Diversamente da Freedom II, si è preferito utilizzare in ogni modo dei resistori di pull-up in maniera da rendere il sistema compatibile con la scheda di sviluppo USB della Microchip (PICDEM™ FS USB).

I resistori R3, R4 risultano fondamentali per la protezione del PIC nel caso in cui si utilizzino i pin RB4-RB5 come output invece di input. In questa situazione la pressione di un tasto potrebbe cortocircuitare verso massa l'uscita, causandone la rottura. Le resistenze R3, R4 evitano che questo possa accadere, limitando la corrente di cortocircuito.

Sul PCB è previsto sia il montaggio orizzontale che il montaggio verticale dei pulsanti. Il montaggio orizzontale risulta più pratico in fase di sviluppo mentre, qualora il sistema venga montato in una scatola di montaggio³, utilizzare pulsanti verticali può risultare più pratico, visto che sono posti al lato del PCB stesso. Questo accorgimento spiega perché siano presenti i pulsanti BT2 e BT3 rispettivamente in parallelo ai pulsanti S2 e S3. Sulla serigrafia, vista la vicinanza dei vari pulsanti è riportata solo la nomenclatura dei pulsanti S2 e S3. Anche il tasto di Reset può essere montato sia orizzontalmente che verticalmente (S1, BT1).

³ Il connettore USB e J1 sporgono dal PCB per poter agevolare ulteriormente l'inserimento di Freedom Light all'interno di una scatola di montaggio.

Altre periferiche

Oltre alle periferiche appena descritte, il sistema Freedom Light possiede altre piccole periferiche che è possibile sfruttare nelle proprie applicazioni.

- 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 R6. 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.
- Power LED, per la segnalazione della presenza dell'alimentatore di rete. Il LED3, nominato *DC PW*, diversamente dagli altri presenti sulla scheda non è controllato dal PIC.
- Nonostante Freedom Light 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. I LED possono essere abilitati o disabilitati per mezzo del Jumper JP2 nominato LED. Ponendo il Jumper sulla posizione 2-3 i LED risultano attivi e controllabili dal PIC per mezzo dei rispettivi pin RD0-RD1. Chiudendo il Jumper tra i pin 1-2 i LED risultano disattivi.
- Trimmer per segnali analogici. Come visto nella sezione relativa al modulo LCD alfanumerico, qualora non lo si utilizzi, è possibile utilizzare il Trimmer per pilotare l'ingresso analogico AN0, piuttosto che il contrasto del modulo LCD. Per posizionare il Trimmer sul pin AN0 è necessario chiudere il Jumper JP1 tra 1-2, mentre chiudendo JP1 tra 2-3 si pilota il contrasto del modulo LCD.
- Un Buzzer auto-oscillante permette di dare semplice voce alla scheda. Risulta collegato al pin RC0 ma può essere attivato e disattivato per mezzo del Jumper JP5. Collegato il Jumper JP5 tra 1-2 si attiva il Buzzer mentre collegando JP5 tra 2-3 lo si disattiva. Un dettaglio del Buzzer è riportato in Figura 6.

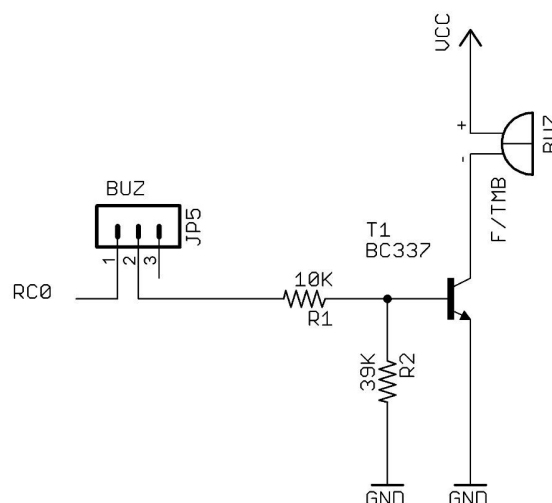


Figura 6: Buzzer collegato al pin RC0.

Connettore di espansione

Il sistema Freedom Light possiede a bordo l'hardware essenziale per ottenere periferiche USB di base. Hardware esterno può essere collegato per mezzo del connettore di espansione EX1 compatibile con il sistema Freedom II e EasyUSB. In Figura 7 è riportata la piedinatura. Per tale ragione le varie schede di espansione disponibili per i sistemi Freedom II e EasyUSB, possono essere utilizzate anche per Freedom Light.

EX1			
RC7	1	2	RC6
USB_D_P	3	4	USB_D_N
RC3	5	6	RC2
RC1	7	8	RC0
RA5	9	10	RA4
RA3	11	12	RA2
RA1	13	14	RA0
GND	15	16	UCC
RB7_PGD	17	18	RB6_PGC
RB5	19	20	RB4
RB3	21	22	RB2
RB1	23	24	RB0
GND	25	26	RF2
RF1	27	28	RF0
MCLR	29	30	VIN
GND	31	32	UCC
RD7	33	34	RD6
RD5	35	36	RD4
RD3	37	38	RD2
RD1	39	40	RD0

Figura 7: Connettore di espansione EX1.

Il progetto PJ7007⁴ riportato in Figura 8, può essere utilizzato come scheda di espansione per i vari prototipi USB.

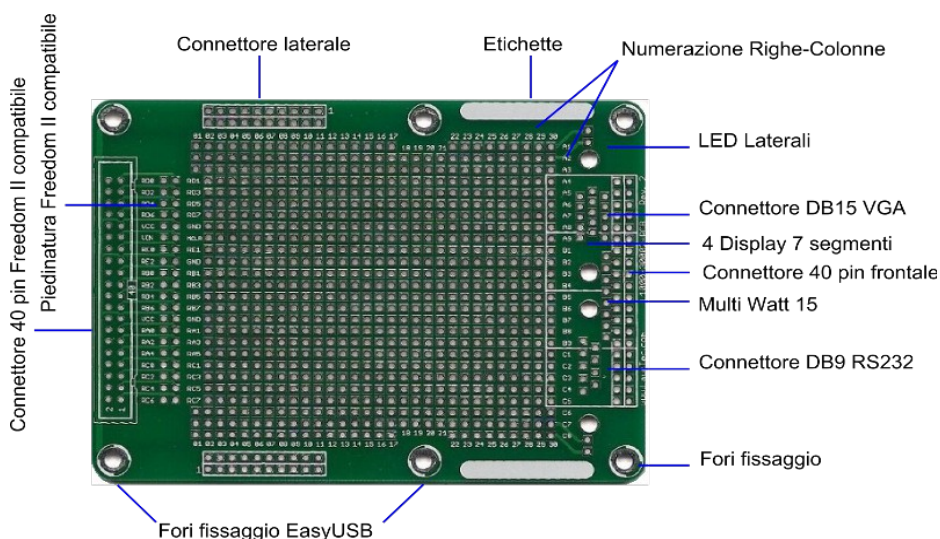


Figura 8: Scheda di prototipizzazione per la scheda Freedom Light-EasyUSB (compatibile Freedom II)

Oltre progetto PJ7007 è disponibile anche la scheda di espansione dedicata a Freedom Light o la

⁴ La scheda di espansione PJ7007 può essere richiesta alla sezione servizi in forma di KIT. Il KIT include tutti i connettori e cavo di connessione con la scheda EasyUSB.

scheda EasyUSB, ovvero il progetto PJ7009. Il vantaggio della scheda di espansione Freedom Light è legato al fatto che questa è delle stesse dimensioni di Freedom Light, permettendo di ottenere prototipi compatti senza necessitare di cavi aggiuntivi.

Un esempio di montaggio è riportato in Figura 10. Maggiori dettagli sul progetto PJ7009 possono essere trovati nella scheda tecnica dedicata⁵.

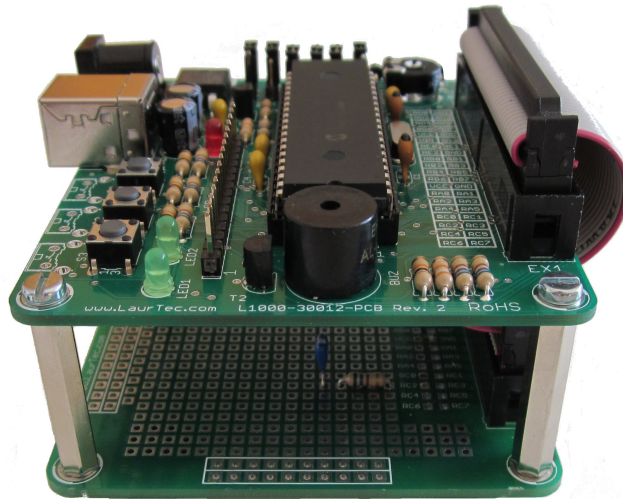


Figura 9: Scheda di espansione EasyUSB.

Per rendere tutto più compatto è anche disponibile la scheda PJ7013, la quale, come visibile in Figura 10 e Figura 11 può essere montata direttamente sulla scheda Freedom Light. Sul lato top del PCB è presente la serigrafia e il nome dei pin del microcontrollore, al fine di facilitare il montaggio di schede sperimentali (il PCB usato nel KIT PJ7013 è lo stesso usato nel KIT PJ7009, ma il contenuto dei due KIT differisce per gli altri componenti).

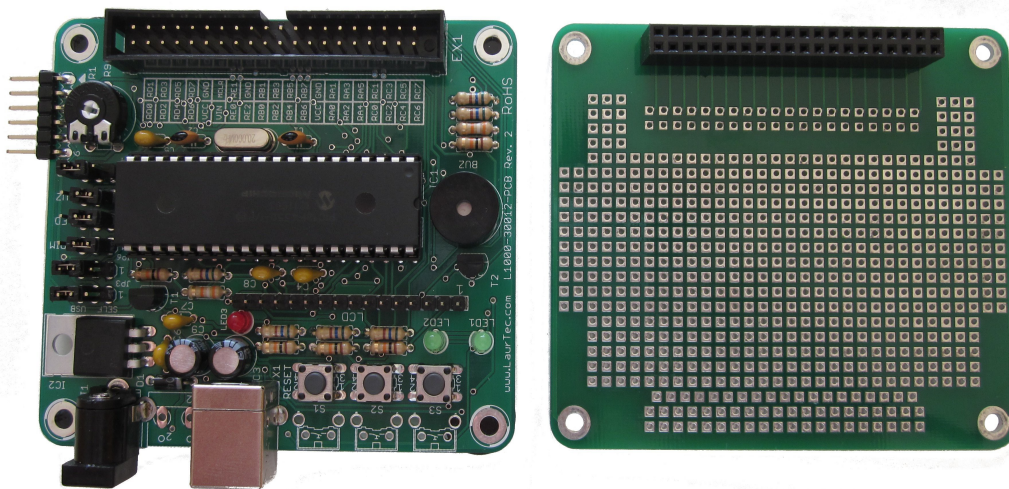


Figura 10: Scheda di espansione Freedom Light.

⁵ La scheda di espansione PJ7009 può essere richiesta alla sezione servizi in forma di KIT. Il KIT include tutti i connettori e cavo di connessione con la scheda Freedom Light.

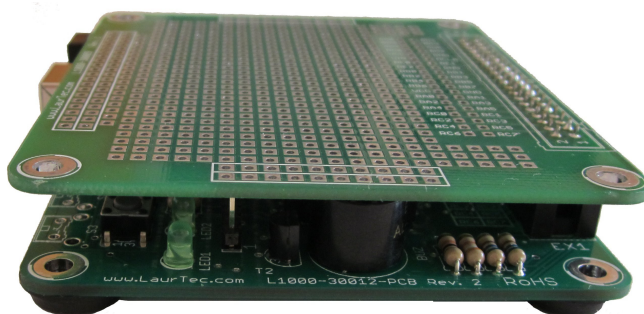


Figura 11: Scheda di espansione Freedom Light montata.

Si fa notare che il connettore EX1, oltre a tutti i pin del PIC porta in uscita GND, +5V e VIN+, permettendo così di alimentare direttamente l'hardware esterno. E' a cura dell'utilizzatore accertarsi che il consumo dell'hardware esterno rientri nei limiti di potenza ammessi dalla scheda Freedom Light e dalle specifiche USB 2.0.

In Figura 12 è riportata la scheda di espansione PJ7011 che possiede una breadboard con pulsanti e LED al fine di poter aggiungere altri componenti discreti senza dover effettuare alcuna saldatura. Maggiori informazioni sulla scheda di espansione possono essere trovati nella documentazione ufficiale PJ7011-IT scaricabile dal sito LaurTec⁶.

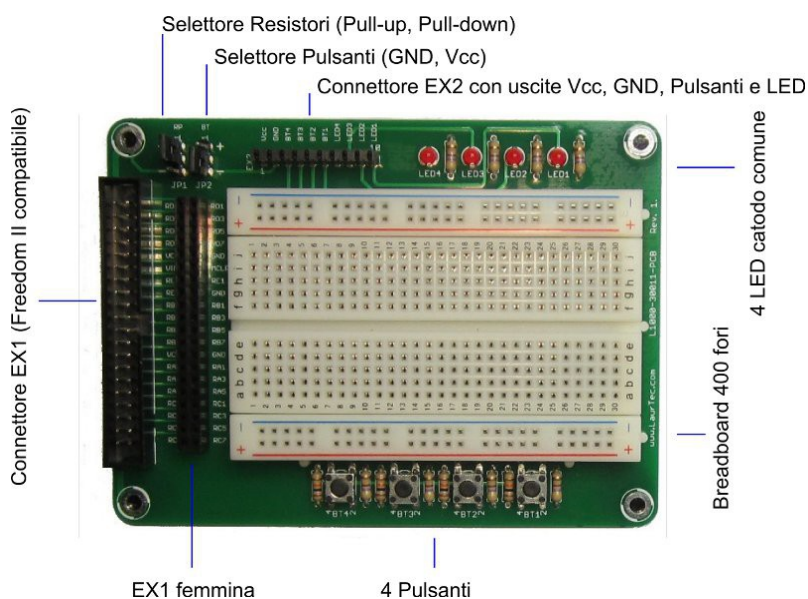


Figura 12: Scheda di espansione PJ7011.

⁶ La scheda di espansione PJ7011 può essere richiesta alla sezione servizi in forma di KIT. Il KIT include tutti i connettori e cavo di connessione con la scheda Freedom Light.

Connettore di Programmazione e Debug

Un'altra importante caratteristica del sistema Freedom Light è la sua integrazione con gli strumenti Microchip. In particolare il suo connettore di programmazione e debug è conforme alle specifiche utilizzate per i programmatori PICKIT, ICD e loro derivati. Il connettore utilizzato è in particolare la versione del connettore lineare a 6 pin come riportato in Figura 13⁷.

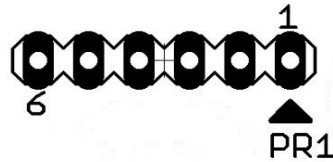


Figura 13: Serigrafia del connettore del programmatore/debug

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 Freedom Light oltre che poter essere programmata con il programmatore può essere programmata anche per mezzo dell'USB Bootloader (maggiori dettagli sono riportati nella scheda tecnica del progetto EasyUSB).

Nota:

Durante la fase di programmazione è bene non toccare la scheda al fine di evitare errori di scrittura e lettura. Mantenere un tavolo di lavoro ordinato aiuterà anche a limitare errori.

⁷ Si raccomanda l'utilizzo di un connettore a 90 gradi in maniera da collegare il programmatore senza problemi.

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, in particolar modo se si dovesse far uso di hardware esterno.

Pin	Periferica	Altre Funzioni
RA0	Trimmer	I/O o in. analogico
RA1	Disponibile	I/O o in. analogico
RA2	Disponibile	I/O o in. analogico
RA3	Disponibile	I/O o in. analogico
RA4	Disponibile	I/O o in. TOKI
RA5	Disponibile	I/O o in. analogico
RB0	Disponibile	SDA per bus I2C in PIC con USB;
RB1	Disponibile	SCL per bus I2C in PIC con USB;
RB2	Disponibile	I/O o INT
RB3	Disponibile	I/O
RB4	Pulsante BT1	I/O vincolato
RB5	Pulsante BT2	I/O vincolato
RB6	Disponibile	I/O, INT su PORTB
RB7	Disponibile	I/O, INT su PORTB
RC0	Buzzer	I/O
RC1	Retroilluminazione LCD	I/O, PWM 2
RC2	Disponibile	I/O o PWM
RC3	Condensatore LDO	I/O vincolato
RC4	D- in PIC con USB.	I/O vincolato
RC5	D+ in PIC con USB.	I/O vincolato
RC6	Disponibile	I/O, linea TX per trasmissioni RS232
RC7	Disponibile	I/O, linea RX per trasmissioni RS232
RD0	LED1	I/O se non si usano i LED
RD1	LED2	I/O se non si usano i LED
RD2	Disponibile	I/O
RD3	Disponibile	I/O
RD4	Disponibile	I/O
RD5	Disponibile	I/O
RD6	Disponibile	I/O
RD7	Disponibile	I/O
RE0	Disponibile	I/O o in. analogico
RE1	Disponibile	I/O o in. analogico
RE2	Disponibile	I/O o in. analogico

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

Freedom Light è un sistema realizzato su PCB a doppia faccia. Il PCB può essere richiesto alla sezione servizi del sito www.LaurTec.it per mezzo di semplice donazione di supporto. Il suo assemblaggio non risulta particolarmente complicato ma necessita certamente di attenzione.

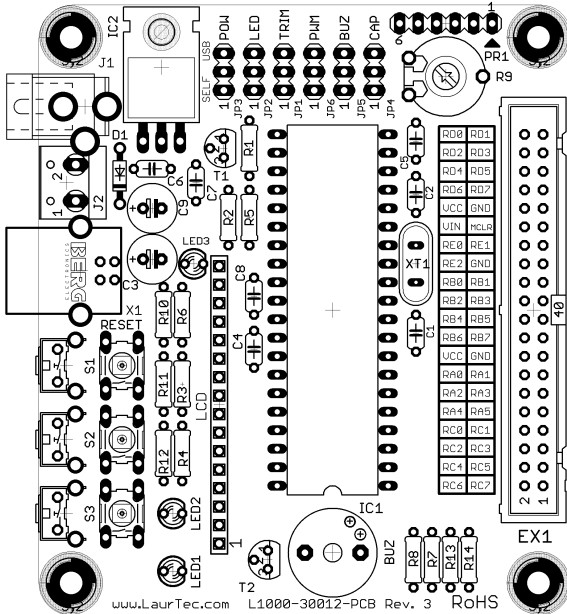


Figura 14: Serigrafia di Freedom Light

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 (identificato 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.

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 14 è facilmente individuabile il terminale positivo delle capacità indicato con un +. Se sul PCB non si dovesse ben leggere qualche carattere a causa di via⁹ far sempre riferimento alla Figura 14.

A montaggio completato Freedom Light apparirà simile a Figura 16. Si possono avere differenze da montaggio a montaggio a seconda dei componenti che si vuole montare o dalla versione del circuito

Per semplificare il montaggio, il PCB è realizzato con serigrafia dei componenti e relativo nome. Lo schema di montaggio è riportato in Figura 14. Il PCB reale di Freedom Light è riportato in Figura 15.

Per il montaggio dei componenti è consigliabile seguire la regola legata all'altezza dei componenti stessi; dunque è bene iniziare dai resistori per poi passare ai diodi. Questa regola ha solamente un'utilità pratica associata al fatto che frequentemente, per fare le saldature, il PCB verrà posto sotto sopra. Per i primi componenti è bene accertarsi che il codice dei colori sia corretto mentre per i diodi è necessario che il verso d'inserzione sia rispettato secondo la serigrafia riportata in Figura 14. In particolare i diodi hanno un anello colorato che segnala la posizione del catodo, questo anello è riportato anche sulla serigrafia⁸.

Successivamente si può procedere al montaggio delle capacità di filtro

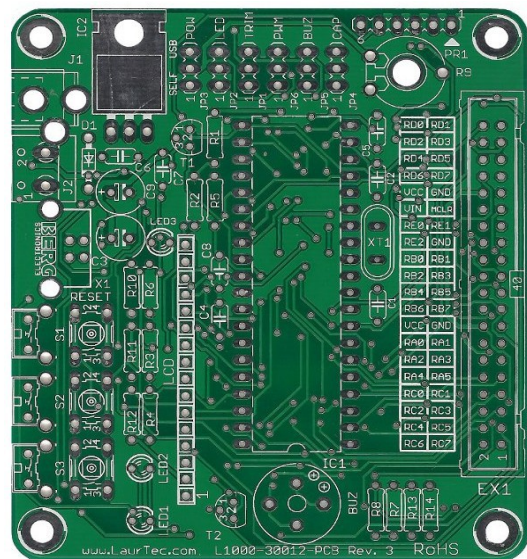


Figura 15: PCB Freedom Light

⁸ 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.

⁹ 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.

stampato stesso¹⁰. In ultimo al fine di non montare storto il regolatore 7805 si consiglia di saldare un solo pin e di controllare se il package è rimasto allineato. Come spessore per piegare i pin utilizzare il PCB stesso. Avere componenti montati in maniera simmetrica renderà il montaggio più professionale.

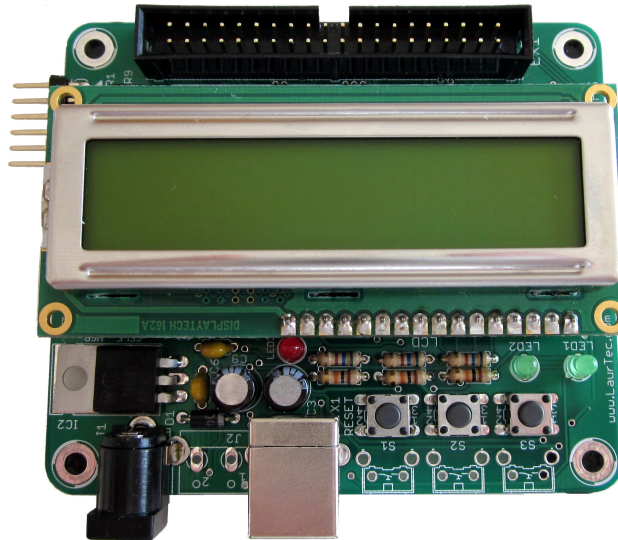


Figura 16: Scheda Freedom Light a montaggio ultimato (con LCD).

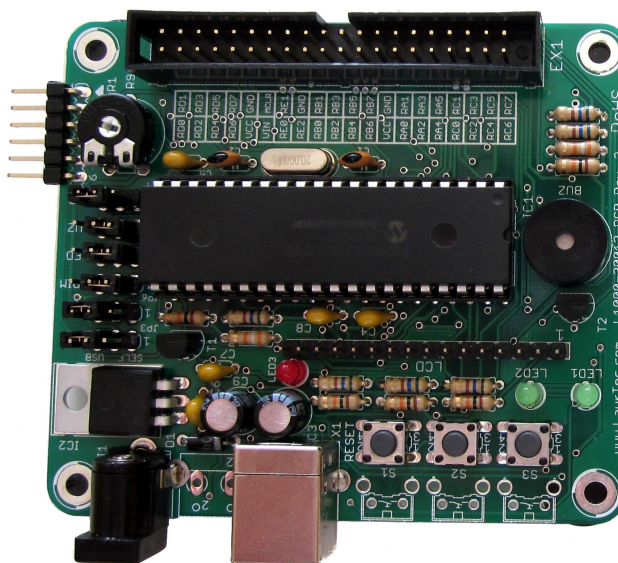


Figura 17: Scheda Freedom Light a montaggio ultimato (senza LCD).

Maggiori dettagli sugli strumenti e tecniche per il montaggio dei KIT elettronici possono essere trovate nel Tutorial AN9001-IT scaricabile gratuitamente dal sito www.LaurTec.it.

¹⁰ Il circuito stampato può subire variazioni senza che il seguente articolo venga aggiornato.

Collaudo e messa in funzione

Una volta montato il sistema Freedom Light, 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, che il terminale + e – non siano in corto.
3. Collegare l'alimentazione con tensione 12V accertandosi che il Jumper JP3 (POW) sia tra il pin 1-2 (ovvero posizione per alimentazioni esterna). Misurando la corrente con un amperometro si deve misurare 10mA e il LED3 rosso si deve accendere.
4. Ad alimentazione scollegata inserire il microcontrollore e caricare il programma di Test nel PIC18F4550 (Freedom_Light_test_suite). La scheda deve avere i Jumper collegati nel seguente modo:

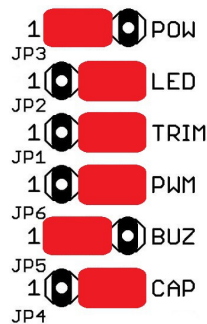


Figura 18: Posizione Jumper per testare la scheda.

Il contrasto del modulo LCD, se presente, deve essere opportunamente regolato al fine di far comparire i messaggi. All'avvio del programma le seguenti funzioni sono testate:

- Beep cicalino
- Accensione sequenziale LED1- LED2
- Lampeggio LED1 – LED2
- Lampeggio retroilluminazione LCD (Se collegato)
- Comparsa scrittura LaurTec e nome scheda
- Richiesta pressione tasto S2
- Richiesta pressione tasto S3
- Scrittura fine Test

Ripetere il test alimentando la scheda da USB (posizionare Jumper JP3 tra 2-3).

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.

Messa in funzione ordinaria

Il normale funzionamento della scheda non richiede particolari attenzioni. Accertarsi che i ponticelli JP1-JP6 siano posizionati come richiesto dall'applicazione e che la corrente di sistema non ecceda le specifiche della parte di alimentazione.

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, pilotare i LED, il cicalino oltre che utilizzare lo stesso display alfanumerico LCD. Oltre a questo esempio base, vista la compatibilità quasi totale della scheda Freedom Light con EasyUSB è possibile usare i progetti e spiegazioni di quest'ultima scheda, con particolare riferimento all'utilizzo del bootloader, qualora questo voglia essere utilizzato. Il testo C18 Step by Step con 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 verranno aggiunti esempi di utilizzo della scheda Freedom Light con particolare attenzione all'utilizzo della porta USB. Gli esempi sono i seguenti:

- Classe CDC, Input Output Demo
- Classe CDC, Analog Demo
- Classe HID, Input Output Demo

Si fa notare che i software mostrati sono basati sui programmi di esempio della stessa Microchip¹¹, opportunamente modificati al fine di mostrare le applicazioni con la scheda Freedom Light. I programmi di esempio scaricabili dal sito LaurTec sono stati sviluppati utilizzando l'IDE MPLAB X e il compilatore C18, i quali devono essere dunque propriamente installati sul proprio PC. Al fine di compilare i programmi di esempio è necessario scaricare le MAL della Microchip¹². 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

¹¹ La libreria MAL (Microchip Application Library) utilizzata è Microchip Solutions v2012-08-22.

¹² 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 Freedom Light simulando una comunicazione con la stessa per mezzo di una porta seriale, ovvero facendo uso della Classe CDC. Per maggiori dettagli sulla classe CDC fare riferimento al Tutorial “Leggere e scrivere dati via USB” scaricabile dal sito www.LaurTec.it.

Una volta aperto e reso attivo il progetto “00-FL-CDC-Input Output” è possibile compilarlo. Se tutti i percorsi sono stati propriamente settati il progetto verrà propriamente compilato e sarà possibile caricarlo nel PIC18F4550.

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.

```
void main(void) {  
  
    // Variabile utilizzata alla lettura del Buffer in Ingresso  
    char dataIN = 0x00;  
    // Variabile da trasmettere contenente il testo  
    char dataOUT[11];  
  
    char data_sent_flag = 0;  
  
    // Inizializzazione base del Modulo USB  
    InitializeSystem();  
    // Inizializzazione del modulo USB per utilizzo interruzioni  
    USBDeviceAttach();  
  
    // Inizializzazione Testo  
    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';  
  
    // Imposto PORTA tutti ingressi  
    LATA = 0x00;  
    TRISA = 0xFF;  
  
    // Imposto PORTB tutti ingressi  
    LATB = 0x00;  
    TRISB = 0xFF;  
  
    // Imposto PORTC tutti ingressi  
    LATC = 0x00;  
    TRISC = 0xFF;  
  
    // Imposto PORTD tutte uscite  
    LATD = 0x00;  
    TRISD = 0x00;  
  
    // Imposto PORTE tutti ingressi  
    LATE = 0x00;  
    TRISE = 0xFF;  
}
```

```
// Abilita i resistori di pull-up sulla PORTB
EnablePullups();

while (1) {
    CDCTxService();

    // Controllo pressione S2
    if (PORTBbits.RB4 == 0 && data_sent_flag ==0)
    if(USBUSARTIsTxTrfReady()) {
        dataOUT[8] = '2';
        putsUSBUSART(dataOUT);
        data_sent_flag = 1;
    }
    // Controllo pressione S3
    if (PORTBbits.RB5 == 0 && data_sent_flag ==0)
    if(USBUSARTIsTxTrfReady()) {
        dataOUT[8] = '3';
        putsUSBUSART(dataOUT);
        data_sent_flag = 1;
    }
}

// Una volta rilasciati i pulsanti riazzero il flag
if (PORTBbits.RB4 == 1 && PORTBbits.RB5 == 1)
    data_sent_flag = 0;

dataIN = 0;
// Leggo un byte dal buffer in ingresso
getsUSBUSART (&dataIN, 1);

// Controllo se è stato premuto 1-2 o altro, sulla tastiera
if (dataIN == 0x31)
    LATD = 0x01;

if (dataIN == 0x32)
    LATD = 0x02;

if (dataIN > 0x32)
    LATD = 0x00;

}
}
```

Come visibile dal programma, l'applicazione non fa altro che controllare la pressione dei pulsanti S2 e S3 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 PIC18F4550 è possibile collegare la porta USB della scheda Freedom Light al PC. Se è la prima volta che si collega la scheda alla porta USB verrà richiesto il file .inf presente nella cartella del progetto stesso (cartella nominata inf). 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, come *RS232 Terminal* e monitorare la porta seriale virtuale creata dalla scheda/driver, come riportato in Figura 19. RS232 Terminal può essere scaricato gratuitamente dal sito www.LaurTec.it. Le impostazioni della porta seriale possono essere anche cambiate, infatti la scheda si adatterà automaticamente alle nuove impostazioni. Nel caso specifico le impostazioni della porta COM sono : 115200,8,1, NONE, NONE.

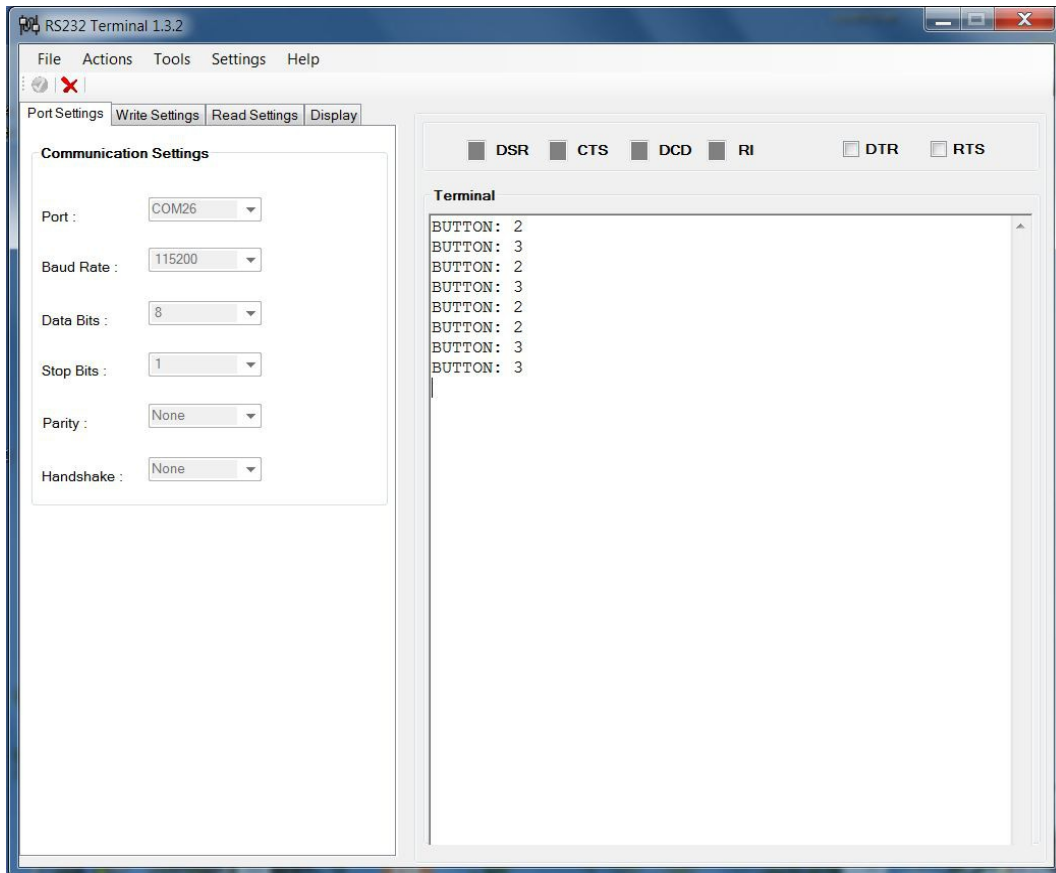


Figura 19: RS232 Terminal impostato per monitorare l'output della scheda Freedom Light.

Il programma di esempio permette anche di accendere i due LED1-LED2 sulla scheda Freedom Light, 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: Classe CDC, Analog Demo

In questo secondo esempio viene mostrata un'altra applicazione basata sulla classe CDC, in cui vengono inviati i dati letti dal convertitore Analogico digitale. L'empio rappresenta dunque una semplice base per poter realizzare un multimetro (lettura di tensione, corrente...) che possa essere collegato al PC. Anche in questo caso si è scritto il programma tutto all'interno della funzione main al fine di rendere l'esempio compatto e di facile comprensione.

```
void main(void) {

    unsigned int waiting_state = 0;
    unsigned char index = 0;
    int adc_read_out = 0;
    unsigned char adc_read_out_string[11];
    unsigned char adc_conversion_running = 0;
    unsigned char measure[16];

    // Inizializzazione base del Modulo USB
    InitializeSystem();
    // Inizializzazione del modulo USB per utilizzo interruzioni
    USBDeviceAttach();

    measure[0] = 0x18; //Delete ASCII code
    measure[1] = 'M';
    measure[2] = 'e';
    measure[3] = 'a';
    measure[4] = 's';
    measure[5] = 'u';
    measure[6] = 'r';
    measure[7] = 'e';
    measure[8] = ':';
    measure[9] = ' ';
    measure[10] = ' ';
    measure[11] = ' ';
    measure[12] = ' ';
    measure[13] = ' ';
    measure[14] = 13; //Return ASCII code
    measure[15] = '\0';

    // Imposto PORTA tutti ingressi
    LATA = 0x00;
    TRISA = 0xFF;

    // Imposto PORTB tutti ingressi
    LATB = 0x00;
    TRISB = 0xFF;

    // Imposto PORTC tutti ingressi
    LATC = 0x00;
    TRISC = 0xFF;

    // Imposto PORTD tutte uscite
    LATD = 0x00;
    TRISD = 0x00;

    // Imposto PORTE tutti ingressi
    LATE = 0x00;
    TRISE = 0xFF;

    // Abilito A0 come ingresso analogico
```

```

// VREF sono impostate a massa e VCC
ADCON1 = 0b00001110;
// Selezione AN0 come ingresso
ADCON0 = 0b00000000;
// Imposto i tempi di conversione e giustificazione a destra
// TAD : FOSC/4
// TACQ: 16 TAD
ADCON2 = 0b10110100;
// Abilito l' ADC
ADCON0 |= 0b00000001;

while (1) {
    CDCTxService();
    waiting_state++;

    if (adc_conversion_running ==0){
        ADCON0bits.GO = 1;
        adc_conversion_running = 0x01;
    } else if (ADCON0bits.GO == 0) {

        adc_conversion_running = 0;

        // Prelevo il valore della conversione
        adc_read_out = (((int) ADRESH) << 8) | ADRESL;

        index = 0;
        itoa (adc_read_out, (char*) adc_read_out_string);

        for (index = 10; index <14; index++) {
            if (adc_read_out_string[index-10])
                measure[index] = adc_read_out_string[index-10];
            else
                measure[index] = ' ';
        }
    }
    // Invio il valore letto dall'ADC
    if (waiting_state == 1000)
    if(USBUSARTIsTxTrfReady()) {
        putsUSBUSART(measure);
        waiting_state = 0;
    }
}
}

```

Una volta compilato il programma è possibile programmare il PIC18F4550. Al fine di poter eseguire il programma di esempio è necessario impostare i Jumper J1-J6, come riportato in Figura 20, ovvero collegando il Trimmer all'ingresso analogico e non al contrasto dell'LCD (LCD deve essere rimosso).

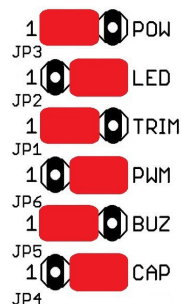


Figura 20: Posizione Jumper per testare l'Esempio 2.

Programmato il PIC è possibile collegare la porta USB al PC. Qualora si sia già eseguito con successo l'Esempio 1 si potrà sentire il suono di sistema di collegamento dell'hardware esterno ma non verrà richiesta alcuna installazione di driver.

Per monitorare l'output del sistema, ovvero la lettura della posizione del nostro Trimmer, è possibile utilizzare RS232 Terminal, come riportato in Figura 21.

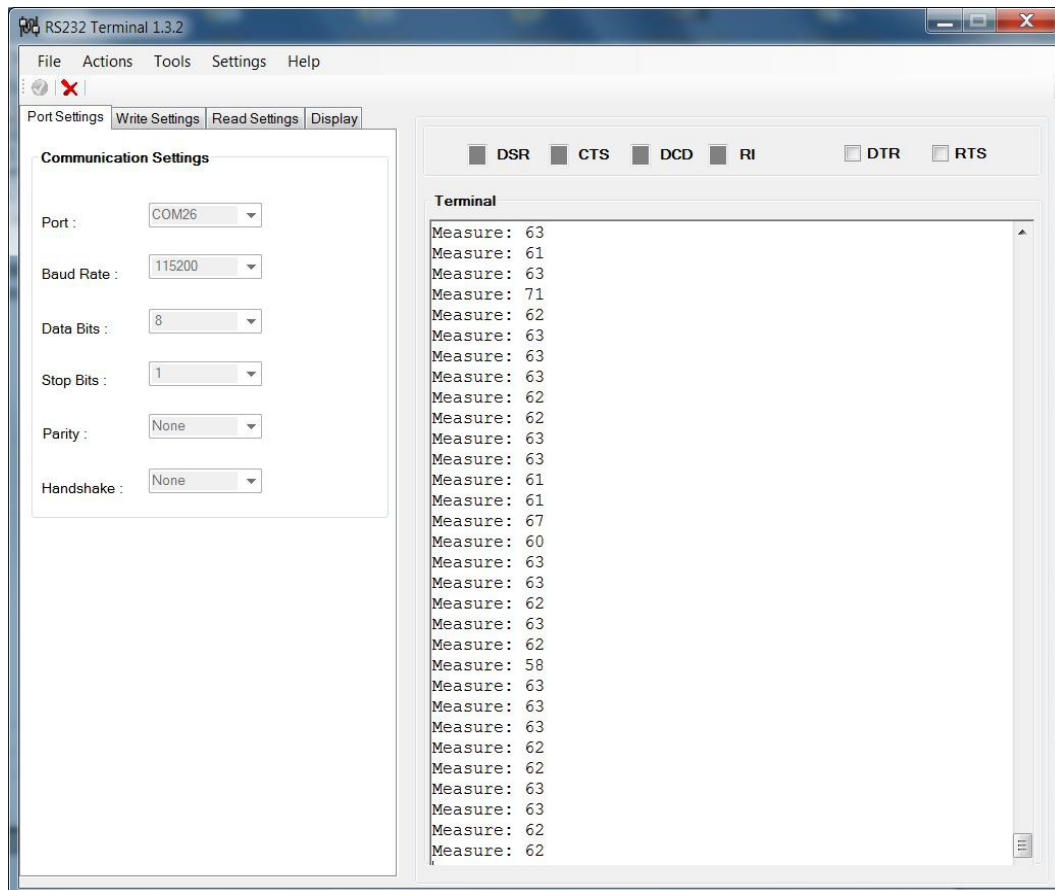


Figura 21: RS232 Terminal impostato per monitorare l'output della scheda Freedom Light.

Si noti come le misure sono scritte in successione sulla schermata del Terminal. RS232 Terminal possiede però una funzione particolare che permette di aggiornare il Terminal ogni volta che viene ricevuto il carattere ASCII <CANC> ovvero il codice esadecimale 0x18. Questa è la ragione per cui il nostro Array di uscita è stato definito nel seguente modo:

```
measure[0] = 0x18; //Delete ASCII code
measure[1] = 'M';
measure[2] = 'e';
measure[3] = 'a';
measure[4] = 's';
measure[5] = 'u';
measure[6] = 'r';
measure[7] = 'e';
measure[8] = ':';
measure[9] = ' ';
measure[10] = ' ';
measure[11] = ' ';
measure[12] = ' ';
```

```
measure[13] = ' ';  
measure[14] = 13; //Return ASCII code  
measure[15] = '\0';
```

Ovvero scrivendo il codice 0x18 nell'array alla posizione 0¹³ permette di cancellare il buffer in ingresso di *RS232 Terminal*, ad ogni nuova ricezione dati. Impostando *RS232 Terminal* come riportato in Figura 22 è possibile avere il nuovo Output sempre su una riga.

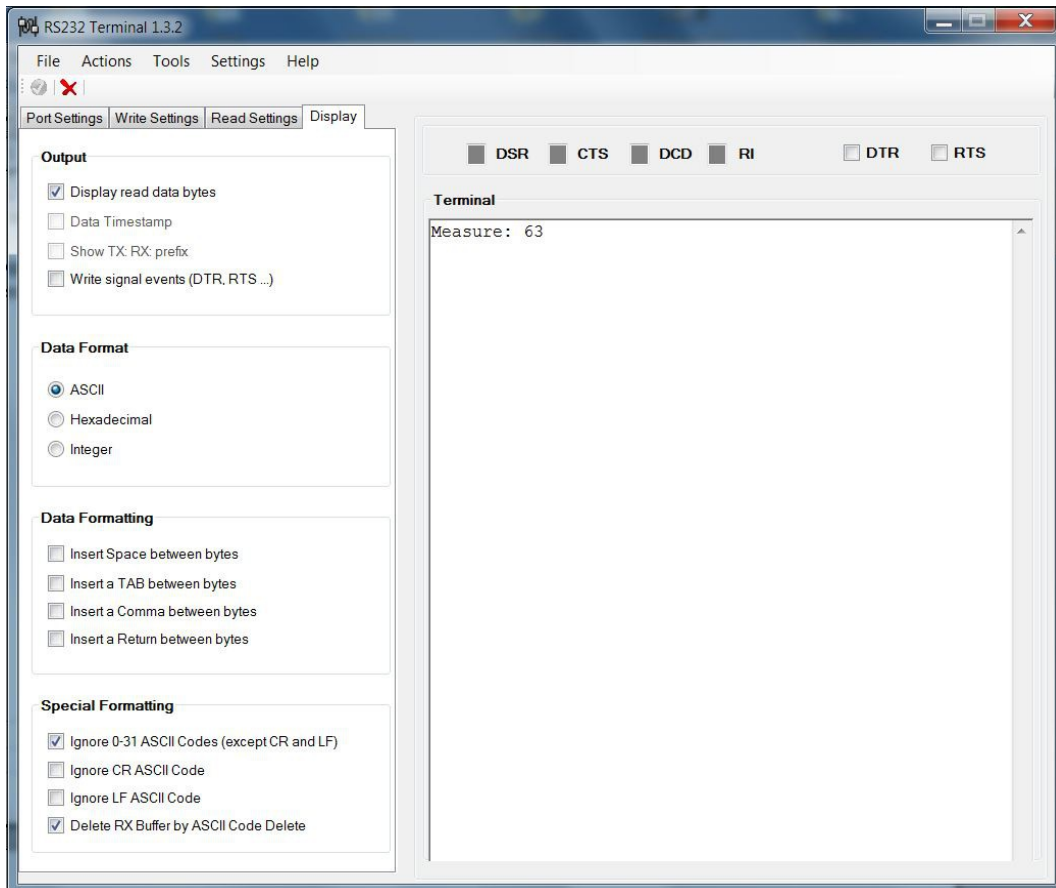


Figura 22: *RS232 Terminal* impostato per monitorare l'output della scheda *Freedom Light*.

Girando il Trimmer sulla scheda *Freedom Light* è possibile comodamente leggere il suo valore sul Terminal. Come detto l'applicazione può essere modificata al fine di realizzare altri sistemi di misura basati su USB e classe CDC.

¹³ Si noti che il carattere 0x18 non viene visualizzato dall'applicazione *RS232 Terminal*, poiché di default l'impostazione "Ignore 0-31 ASCII code" del Tab Display, è normalmente attiva.

Esempio 3: Classe HID, Input Output Demo

In questo terzo esempio vedremo un'applicazione che racchiude i contenuti dei due esempi precedenti ma facendo uso della porta USB classe HID¹⁴ (Human Interface Device). Tale classe è la stessa utilizzata per periferiche come il mouse e la tastiera. La caratteristica principale è che il sistema Freedom Light, 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 più 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. L'esempio 02-FL-HID-Input Output Demo possiede oltre che al Firmware un'applicazione scritta in VB .NET 2010 (Framework 3.5) basata sul demo della stessa Microchip ma estesa per poter comandare direttamente Freedom Light (LED1, LED2, S2, S3 e il Trimmer).

Il Firmware questa volta è scritto 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.

Una volta compilato il programma di esempio e caricato lo stesso nel PIC18F4450 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 Freedom Light, impostata come periferica HID è necessario eseguire l'applicazione Freedom Light come riportato in Figura 23. Diversamente dagli esempi precedenti non è necessario impostare alcuna “porta di comunicazione”, la scheda viene automaticamente rilevata alla connessione ed avviene la trasmissione dei dati.



Figura 23: RS232 Terminal impostato per monitorare l'output della scheda Freedom Light.

L'applicazione permette di leggere in una sola schermata lo stato dei pulsanti S2, S3 e del convertitore analogico (posizione Trimmer). I LED1, LED2 possono essere accesi o spenti semplicemente attivando/disattivando il relativo check box. Il sorgente dell'applicazione Freedom Light può essere variato al fine di realizzare una interfaccia più accattivante e realizzare uno strumento basato su PC. Altra applicazione interessante potrebbe essere quella di realizzare una scheda Relay controllabile via USB.

¹⁴ Si ricorda che la Classe HID fa uso dell'Interrupt Transfer Mode dello standard USB. Pacchetti di 64Byte sono inviati in maniera regolare ogni ms, permettendo data throughput di 64KB/s.

...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.

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_F750
```

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. 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'interno 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
    0xF750,              // 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'interno 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 Freedom Light sono forniti con una combinazione PID-VID unica fornita dalla Microchip ed assegnata a LaurTec, per cui non può essere utilizzata in applicazioni commerciali riadattando gli esempi proposti per i Progetti LaurTec.

Indice Alfabetico

A		PIC18F4455.....	5
alimentatore esterno.....	8	PIC18F4550.....	5
Alimentazione.....	4	PIC18F4553.....	5
Altre periferiche.....	12	PICKIT.....	16
Altro.....	7	PID.....	31
Assorbimento.....	4	ProcessIO.....	30
B		Product ID.....	31
Buzzer auto-oscillante.....	12	programmatori Microchip.....	16
C		programmazione e debug.....	16
Circuiti Integrati.....	7	pulsanti.....	11
clock.....	8	Pulsanti.....	7
clock interno.....	8	Q	
Condensatori.....	7	Quarzi.....	7
Connettore di espansione.....	13	R	
Connettori.....	7	ReceivedDataBuffer.....	30
D		RESET.....	12
Dimensioni.....	4	Resistori.....	7
Diodi.....	7	RS232 Terminal.....	24
diodo di protezione.....	8	S	
DriverInstall.....	31	schema di montaggio.....	18
E		schema elettrico.....	5
EX1.....	13	Serial Interface Engine.....	5
H		serigrafia.....	18
Hardware esterno.....	13	SIE.....	5
HID.....	30	strumenti Microchip.....	16
Human Interface Device.....	30	T	
I		ToSendDataBuffer.....	30
ICD.....	16	Transistor.....	7
L		Trimmer per segnali analogici.....	12
Layout Periferiche.....	17	U	
Limiti di corrente.....	9	USB.....	31
Lista Componenti.....	7	USB classe HID.....	30
M		usbds.c.....	31
microcontrollori.....	5	Utilizzo di un alimentatore esterno.....	9
MPLAB X®.....	16	V	
P		Vendor ID.....	31
Part Number.....	4	Versione.....	4
PCB.....	18	via.....	18
Peso.....	4	VID.....	31

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
12.06.14	1.0b	Mauro Laurenti	Cambiato il nome di Q1 nella lista dei componenti, in XT1.
24.01.13	1.0a	Mauro Laurenti	Correzione errori di battitura.
10.10.12	1.0	Mauro Laurenti	Versione Originale.