



Steve Blackbird (TardoFreak)

MICROCONTROLLORI COME CIRCUITI LOGICI

23 September 2010

L'evoluzione tecnologica nel campo dei **microcontrollori** e la loro massiccia diffusione hanno determinato anche la riduzione del loro costo. Oggi troviamo in commercio microcontrollori a prezzi inferiori al dollaro che ci permettono di affrontare la **progettazione di apparecchiature in un modo nuovo**, progettando **sistemi composti da più microcontrollori**. In buona sostanza possiamo guardare al micro non solo come l'unico "cervello" dell'apparecchiatura, come colui che tutto svolge e tutto controlla ma anche come un **blocco funzionale** che svolge un certo numero di funzioni. In passato questo era difficile da fare anche solo per il fatto che i microcontrollori con memoria **EPROM** erano molto costosi. Oggi tutti i microcontrollori hanno come memoria di programma una **FLASH EPROM** che si cancella elettricamente e che rende tutto più semplice.

Esempi di come questo modo di realizzare le apparecchiature viene applicato, magari senza rendersene conto, sono i circuiti integrati che effettuano la conversione da **USB a RS232**, i **players MP3**, i controller per display LCD, i circuiti di misurazione della potenza ed altro ancora. In questi casi si demanda la gestione dell'USB o della decodifica MP3 a microcontrollori venduti come normali integrati digitali, ma che tanto normali non sono. Il vantaggio di questo approccio è quello di lasciare i lavori pesanti (come la gestione dell'USB) alle unità periferiche. In pratica il blocco funzionale della gestione di un macro-servizio viene spostato dall'interno del master all'esterno di esso e, se ben realizzato, può essere impiegato in altre apparecchiature simili, proprio come un blocco funzionale software. Nelle applicazioni hobbistiche può essere un bel vantaggio ed un modo interessante di sperimentare. Vedremo come mediante un semplice esempio di progettazione di circuito logico.

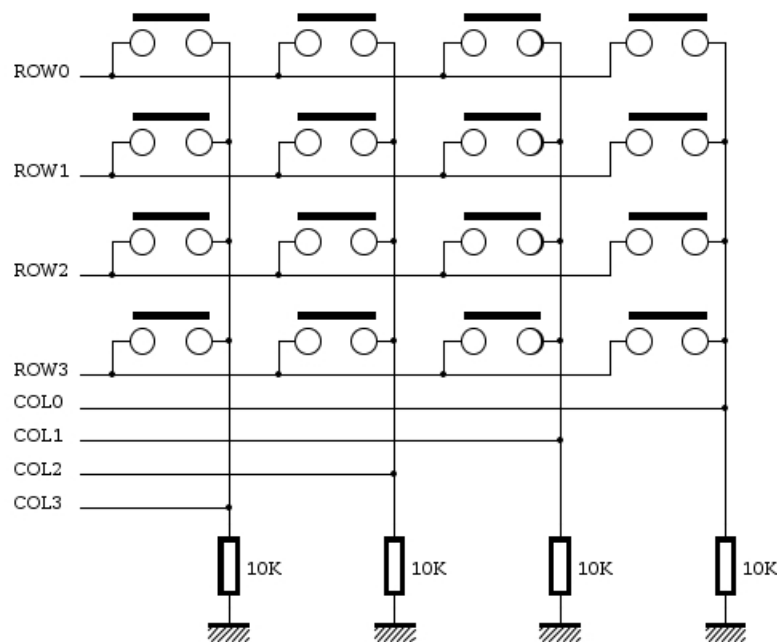
Circuiti logici complessi

In passato, ed ancora oggi in campo hobbistico, per realizzare i circuiti si usavano integrati logici più o meno complessi. L'integrato 74C926 era una benedizione del cielo! Questo integrato a 16 pin svolgeva la funzione di contatore a 4 digit con decodifica e circuiteria di multiplexing integrata. La possibilità di poter sostituire 4 contatori e 4 decodifiche/driver con un solo integrato e di risparmiare collegamenti grazie al multiplexing, permetteva di realizzare facilmente e a poco prezzo frequenzimetri, orologi digitali, timers e quant'altro. Un altro oggetto interessante usato ancora oggi era l'ICL7106 (LCD) o l'ICL7107 (LED), un convertitore A/D con un driver per un display da 3 cifre e mezza. Probabilmente il multimetro digitale

che avete sul banco o nella cassetta degli attrezzi usa quel componente. Un altro componente era il 74C922 (ringrazio gli amici del forum che mi hanno trovato la sigla) della National Semiconductor che gestiva una tastiera decimale o esadecimale fornendo il codice binario su 4 linee più uno strobe. Ed e' proprio questo l' oggetto che porterò come esempio in questo articolo ma ... con una funzionalita in più.

Le tastiere a matrice

Nella maggior parte delle applicazioni le tastiere hanno i pulsanti collegati o direttamente agli ingressi del micro o disposti a matrice. Il primo caso riguarda tastiere con un numero limitato di pulsanti. Essendo ogni pulsante collegato ad un ingresso del micro, la gestione di una tastiera di questo tipo è banale. Basta infatti leggere lo stato dell' ingresso per conoscere lo stato del pulsante. Se escludiamo l' algoritmo per eliminare gli inevitabili rimbalzi dovuti al contatto meccanico, il software non presenta nessun problema ed è di facile stesura. Purtroppo questo sistema è molto dispendioso in termini di linee di ingresso infatti una tastiera da 16 tasti ne impegnerebbe ben 16. La tastiera a matrice permette di ridurre il numero di linee e, nel caso di una tastiera a 16 pulsanti, bastano 8 linee per gestirla completamente. Con 10 linee si possono gestire 25 pulsanti e con 16 linee ben 64 pulsanti.



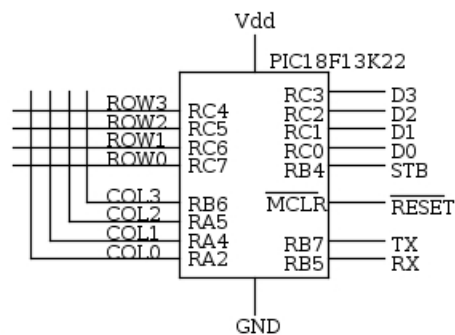
tast16.jpg

Il principio di funzionamento è semplice: si attiva una riga per volta e si leggono gli ingressi delle colonne per vedere quale tasto è premuto. Per esempio si attiva la linea ROW0 collegandola al positivo o dando un livello logico "1". Le linee da COL0

a COL3 saranno a livello "1" se uno o più tasti della riga in alto saranno premuti altrimenti saranno a "0". Non si possono usare delle semplici uscite digitali perchè possono essere messe in collegamento fra di loro dalla pressione dei tasti. Facciamo un esempio. Supponiamo di avere attivato la linea ROW0 e che i due tasti della prima colonna a sinistra siano ambedue premuti, la linea ROW0 e la linea ROW1 risulteranno collegate insieme. Due linee di uscita, una a livello logico "0" e l'altra ad "1" vanno in conflitto, fanno a testate insomma, e si possono quindi danneggiare. Un rimedio sarebbe quello di mettere dei diodi alle uscite oppure fare in modo che le uscite siano tali solo quando sono attivate e che a riposo si comportino come dei circuiti aperti, in pratica come delle linee di ingresso. Noi useremo questo sistema.

Il controller per tastiera

Il controller che ho realizzato, oltre ad avere le linee da collegare alla tastiera presenta quattro linee di uscita da D3 a D0, un segnale di strobe STB attivo alto che si attiva alla pressione di un pulsante e si disattiva quando non ci sono più tasti premuti, una linea di RESET (normalmente collegata alla Vdd tramite una resistenza da 10K), e due linee TX/RX per comunicazione seriale.



CtrlTast.jpg

Quando si preme un pulsante il codice corrispondente viene presentato sulle linee da D0 a D3 quindi il segnale STB viene attivato. Sul fronte di salita del segnale di STB è possibile catturare il dato. Il segnale STB torna a "0" quando si rilascia il pulsante. Nel caso della pressione di più pulsanti viene presentato il codice del primo pulsante che si è premuto. Nel caso di due tasti premuti contemporaneamente non vi sarà nessuna variazione sulle linee D0-D3 e sul segnale STB. Le linee da COL0 a COL3 devono essere collegate a massa tramite resistenze da 10K. Il micro svolge le stesse funzioni dell'integrato 74C922 con una funzione in più: l'invio del codice ASCII del tasto premuto tramite la linea di comunicazione seriale. La velocità di comunicazione è impostata in modo automatico tramite una procedura di riconoscimento della velocità (auto-bauding). L'applicazione funziona subito come controller che presenta i dati sulle linee D0-D3 ma, se riceve un carattere esadecimale 0x55 corrispondente

al carattere ASCII "U" dall' ingresso RX, riconosce la velocità e da quel momento in poi invierà anche il carattere ASCII attraverso l' uscita TX. Potremo così interfacciare il controller nel modo che più ci aggrada. Il micro utilizzato è il PIC18F13K22 per vari motivi. In primo luogo diciamo che ... non costa, o meglio, costa veramente poco (meno di 1,5 euro), può funzionare con una tensione di alimentazione che va da 1,8V (2,7V a qualsiasi frequenza) fino a 5,5V, ed ha l' oscillatore interno. L' uso dell' oscillatore interno permette di evitare di montare un quarzo. E' anche per questo motivo che ho pensato alla procedura di auto-bauding, infatti l' auto apprendimento della velocità permette di eliminare l' errore di frequenza dell' oscillatore oltre che fornire flessibilità. Il non montare il quarzo vuol anche dire avere più linee di I/O disponibili. Direi che per questo tipo di applicazioni può essere un ottimo candidato.

Il firmware

Il firmware è scritto in C per il compilatore MPLAB C18 e non è niente di speciale. D' altronde l' applicazione è quella che è: molto semplice. Il sorgente è scaricabile e può essere anche riutilizzato per essere integrato in un progetto che prevede la scansione della tastiera. Come mio costume ho previsto la completa programmabilità delle linee utilizzate in modo da rendere facile l' adattamento all' hardware e dare la possibilità di utilizzare qualsiasi linea di I/O. la tabella cod_tasti contiene i codici esadecimali dei tasti. Sono un pò sparsi per via della tastierina che utilizzo e, per trovare la corrispondenza di ogni tasto con la tabella ho usato un sistema semplice: dapprima ho scritto i codici da 0 a F in sequenza e poi, premendo i tasti, ho visto quale codice usciva. Ho poi sostituito i codici originali con quelli corrispondenti ai tasti. E' un sistema da "praticone" ma ... funziona! Da notare, all' interno della funzione accendi_riga, il fatto che le linee non utilizzate sono riconfigurate come ingressi per evitare, come spiegato prima, conflitti quando sono premuti più tasti. L' autobauding è una procedura molto semplice: si tratta di mettere il bit ABDEN a 1 ed aspettare che il flag di carattere ricevuto (RCIF del registro PIR1) vada a "1" ed il gioco e' quasi fatto. Bisogna infatti prendere il valore del registro SPBRG (in questo caso anche SPBRGH), sottrargli 1 e ricaricarlo.

Conclusioni

Utilizzare un microcontrollore per una funzione come questa potrebbe sembrare una scelta non molto intelligente ma, se ci pensiamo bene, non si discosta molto dalle applicazioni dei microcontrollori più piccoli (quelli a 8 pin) che vengono utilizzati negli alimentatori switching, nei controlli di motorini, varialuce e quant' altro. La possibilità di poter realizzare integrati che svolgono funzioni complesse (o semplici) è estremamente utile a livello hobbistico. Basti pensare al classico problema di realizzare un contatore a 4 cifre completo di decodifica, oppure un circuito per ottenere una frequenza programmabile. Ancora più interessante è l' applicazione del convertitore A/D del micro con il quale è facile realizzare voltmetri che, ad esempio,

pilotano direttamente i displays. Con questo approccio si può sviluppare il progetto una volta sola, conservare il programma e, invece di impazzire cercando un integrato o un circuito che svolga la funzione e che magari è introvabile o costoso, prendere un micro, programmarlo e montarlo nel circuito. In questo modo potremmo eliminare quasi tutti i cassettoni contenenti i circuiti logici, contatori, decodifiche, generatori di clock e tenerne uno solo con i micro vergini pronti all' uso. Un solo componente e tanti piccoli programmi.

Download

[TF_key01.c](#)

eprom: La EPROM, acronimo di Erasable Programmable Read Only Memory, è una memoria di sola lettura cancellabile tramite raggi ultravioletti. Si tratta di un'evoluzione della PROM che, una volta programmata, non poteva essere più modificata (salvo usare trucchi da alchimista). ...it.wikipedia.org/wiki/EPROMMore »

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Tardofreak:microcontrollori-come-blocchi-funzionali>"