

# FT215

# DEMOBOARD PER MICROCONTROLLORI PIC

Come anticipato nelle precedenti puntate del Corso, è nostra intenzione mettere in grado chiunque di apprendere le tecniche di programmazione dei microcontrollori PIC. Abbiamo quindi pensato che la soluzione migliore sia quella di realizzare una basetta di test, una demoboard, e di associare a tale scheda una serie di programmi didattici appositamente realizzati. In questo modo, partendo da un hardware affidabile e da dei listati software già ampiamente collaudati, l'apprendimento diventa veramente semplice, veloce ed anche, perché no, divertente. Una scheda di test multiuso quindi, adatta allo studio ma anche per fare qualche esperimento con i micro PIC e per crearsi una prima libreria di routine affidabili e funzionanti da utilizzare come punto di partenza per programmi più complessi. Prima di entrare nel vivo della demoboard, rammentiamo qualche prestazione del PIC 16C84, micro a cui questa demoboard è dedicata. E' certamente uno dei migliori, tra quelli ad 8 bit, perché ha un'architettura simile alla RISC (con un set di sole 35 istruzioni) dispone internamente, oltre alla solita RAM (registro 36x8 bit, oltre a 15 registri per funzioni speciali) di una EPROM nella quale memorizzare sia il programma che eventuali dati di caratterizzazione; ciascun componente può essere programmato e riprogrammato senza troppi problemi (la Casa garantisce addirittura 1.000.000 di operazioni di read/write). L'importanza del PIC16C84 ci ha spinti non solo ad applicarlo in diverse situazioni (alcune le avete viste, altre le vedrete...) e a dedicargli un programmatore, ma anche a progettare e proporre una scheda di test universale che permetta di verificare un programma appena scritto e caricato nella EPROM. Se quindi volete testare un software realizzato, ad esempio, per leggere una tastiera a matrice e pilotare un display non dovrete fare altro che inserire il microcontrollore nel proprio zoccolo, quindi abilitare il display (LCD o a led, a seconda del vostro programma) ed il latch della matrice. Ancora, se il vostro PIC16C84 deve visualizzare scritte ed attivare un relè, potete abilitare il solito display LCD, ben adatto a questo scopo, e il latch che interfaccia i relè con le porte di I/O. Insomma, la demoboard che proponiamo consente di simulare praticamente tutte le situazioni pratiche nelle quali si può impiegare il microcontrollore Microchip, e ciò è utilissimo perché consente al progettista di aggiustare eventuali errori software senza la necessità di ingegnerizzare uno specifico hardware. Per questo motivo lo consigliamo anche a chi abitualmente lavora con i micro, e soprattutto a quanti tra i nostri lettori vorranno seguire il nostro Corso, poiché diventerà il miglior ausilio didattico per mettere in pratica le nozioni apprese. Vediamo allora in pratica questa scheda di test, analizzandone lo schema elettrico: nonostante si tratti di un circuito grande e complesso, comprenderlo è piuttosto facile, dato che in sostanza non è altro che un microcontrollore (U8) contornato da una serie di buffer tri-state che, a seconda dell'impostazione che farete manualmente, possono applicare i suoi I/O una volta ad un sottoinsieme, una volta all'altro, ecc. Questi "sottoinsiemi" non sono altro che dei tipici circuiti controllabili con il PIC, e cioè: il display a cristalli liquidi (Display LCD) i due relè RL1 e RL2, il cicalino piezoelettrico BZ, il display 7-segmenti a led (Display) la tastiera a matrice di 4x4 con linee e colonne sulle resistenze R14, R15, R16, R17, R18, R19, R20, R21, e la barra di led LD1÷LD8; sono stati previsti anche un paio di pulsanti liberi quali dispositivi di ingresso, utili per sollecitare dall'esterno gli ingressi del microcontrollore e per verificare la bontà delle routine di antirimbando, e comunque di lettura dei fronti di salita/discisa e dei livelli logici TTL. Possiamo perciò

dire che la nostra Demo-Board è davvero completa. Il tutto funziona con l'alimentazione applicata tra il punto +V e la massa, ovvero con 12 volt c.c. che, passato il diodo di protezione D1 (serve contro l'inversione di polarità) alimentano le bobine dei relè ed il cicalino BZ; il regolatore U6 provvede quindi a ricavare 5 volt ben stabilizzati che servono tutta la logica, cioè il microcontrollore U8, il display LCD e quello a led, le resistenze di pull-up per i pulsanti P1/P2 e per quelli della tastiera a matrice, e quelle per i dip-switch ed il resto. Per consentire di far funzionare il PIC16C84 ora con una parte dell'hardware, ora con un'altra, i vari circuiti sono tutti separati dagli I/O mediante dei line-driver tri-state: i notissimi 74244, che hanno la caratteristica di comportarsi come semplici buffer quando sono abilitati, mentre disattivando i loro piedini di controllo assumono in uscita lo stato di alta impedenza, cioè non trasmettono più quanto ricevono agli ingressi. Le uscite non assumono né lo zero né tantomeno l'1 logico, ma vengono lasciate fluttuanti, come dei transistor open-collector. Praticamente, volendo fare un esempio, in condizioni normali dal piedino 18 esce lo stato logico applicato al 2, dal 16 quello del 4, ecc. Ciò quando il piedino 1 è collegato a massa, ovvero allo zero logico, mentre se viene posto a livello alto il relativo buffer è disattivato e i pin 18 e 16 (ma anche il 12 ed il 14) sono posti nella condizione tri-state, cioè ad alta impedenza, e si comportano come dei transistor open-collector. Va notato che ogni line-driver di quelli utilizzati è scomposto in due parti, ciascuna fatta di 4 buffer; esternamente si presenta in contenitore dip a 10+10 piedini, 16 dei quali riservati agli I/O, 2 per l'abilitazione (uno per ciascun gruppo di 4 buffer) ed altri due per l'alimentazione a 5 volt. I piedini 2, 4, 6, 8, sono ingressi e fanno capo rispettivamente alle uscite pin 18, 16, 14, 12; il piedino d'abilitazione per questo blocco di 4 elementi è l'1 (1/G) attivo a 0 logico e disabilitato ad 1. Il secondo gruppo, con ingressi ai piedini 11, 13, 15 e 17, ed uscite rispettivamente ai pin 9, 7, 5 e 3, si abilita invece tramite il piedino 19 (2/G). Nella nostra scheda di test sono impiegati 5 di questi line-driver, dei quali uno (U1) è dedicato interamente alla barra di led (LD1÷LD8) un altro (U2) e parte dell'U4 (U4b) sono destinati all'interfaccia parallela con il display a cristalli liquidi, un altro ancora (U3) è usato per la gestione di righe e colonne della tastiera a matrice, e l'ultimo, U5, è impiegato parte per i pulsanti (U5a) indipendenti P1 e P2, e parte (U5b) per il controllo dei transistor che pilotano i due relè ed il cicalino piezo. L'altra metà dell'U4 (U4a) è impiegata come buffer tra il microcontrollore ed il decoder BCD/7-segmenti (74LS47) posto nel circuito per controllare appunto il display a led. Bene, a questo punto possiamo subito vedere in che modo si imposta la scheda di test, ovvero come si assegnano le periferiche agli I/O del microcontrollore U8. Allora, sono possibili 3 diverse combinazioni, scelte sulla base di quelle che ci sono sembrate le applicazioni più comuni del PIC16C84; tutte si selezionano, con l'aiuto dei buffer tri-state, mediante i jumper JP1, JP2 e JP3: - Ponticellando JP1 si attivano i line-driver U3a, U3b, ed U4a; in pratica si rendono disponibili: 1) la tastiera a matrice (da collegare alle righe ed alle colonne portate alle rispettive piazzole dello stampato) collegata alla porta B del PIC, ovvero alle rispettive linee RB0÷RB7; 2) la cifra decimale (gestita da 4 degli 8 bit della porta A, cioè RA0÷RA3) ovvero il display 7-segmenti a led pilotato tramite U4a e il decoder BCD/7-segmenti siglato U7. - Ponticellando JP2 si rendono disponibili: 1) il display LCD, collegato alle linee RB0÷RB7 per quanto riguarda il bus-dati, e ad RA0÷RA2 per la gestione delle linee di controllo RS, R/W, ed E; 2) i



pulsanti P1 e P2, collegati alle linee RA3 ed RA4, configurate queste come ingressi. - Chiudendo JP3 si abilitano infine: 1) gli 8 led collegati alla porta B (RB0÷RB7) mediante il buffer U1; 2) i pulsanti P1 e P2 gestiti ancora da RA3 ed RA4; 3) i due relè RL1 ed RL2, attivati tramite i transistor T1 e T2, con le linee RA1 ed RA2 del microcontrollore; 4) il cicalino piezoelettrico BZ, comandato dal transistor T3, pilotato a sua volta dalla linea RA0 della porta A. Notate che per come è fatto il simulatore le selezioni possono essere fatte una sola alla volta, cioè si può chiudere uno soltanto dei 3 jumper, e non più d'uno contemporaneamente. Notate ancora che i ponticelli JP1, JP2 e JP3 sono provvisti ciascuno di una resistenza di pull-up, e comandano direttamente, o tramite una semplice logica a diodi, i piedini di abilitazione dei buffer tri-state: chiudendo il primo si abilita l'intero U3, e la prima parte dell'U4 (U4a) permettendo la scansione della tastiera a matrice e l'uso del display a led, mentre U2, U4a, U1 ed U5 sono in tri-state perché con JP2 e JP3 aperti le rispettive resistenze di pull-up ne pongono i pin di controllo a livello alto. Praticamente R11 tiene ad 1 logico il piedino 1 dell'U5b ed 1 e 19 dell'U1, oltre al catodo del D4; R12 fa lo stesso con il catodo del diodo D5, con il piedino 19 dell'U4b, e con 1 e 19 dell'U2. La resistenza R8 assicura il livello alto al pin 19 dell'U5a, poiché D4 e D5 sono interdetti. Chiudendo il solo JP2 l'U5a viene attivato con lo zero portato dal D5, ed anche U4b ed U2 vengono abilitati mediante i piedini 1 per il primo, ed 1/19 per il secondo. Gli altri buffer sono disattivati. Con JP3 chiuso si abilita ancora U5a, stavolta tramite lo zero logico portato dal diodo D4, ma anche U1 (i pin 1 e 19 vengono messi a livello basso) ed U5b. Riassumendo possiamo pertanto dire che con JP1 è possibile testare ad esempio un software per gestire una tastiera a matrice di 4 righe per 3 colonne, oppure 4x4, che visualizzi sul display a led il numero del tasto battuto di volta in volta; oppure un programma che comunque debba gestire la matrice o solo il display 7-segmenti. Con le funzioni legate a JP2 si può lanciare invece ogni software di visualizzazione su display LCD intelligente, tipo quello della Clover (CDL4162) a 2 righe per 16 caratteri, che preveda magari la lettura di due pulsanti: insomma il funzionamento di un PIC16C84 come temporizzatore, orologio o sveglia programmabile, contatore, semplice visualizzatore, ecc. Infine, con la selezione fatta da JP3 si ha la possibilità di verificare programmi di vario tipo, che debbano accendere led o comandare relè, leggere pulsanti o generare note acustiche per pilotare cicalini o sirene.

**L'articolo completo è stato pubblicato su Elettronica In N. 27 marzo '98**