



informa@iw1axr.eu

Questo articolo è stato pubblicato su....



Programmatore "minimo" di PIC

Programma i PIC più comuni ed è collegato sulla porta parallela del PC

di Daniele Cappa IW1AXR

Sono anni che si vedono in giro minuscoli programmatore di PIC, quasi tutti collegati alla seriale, battezzati con i nomi più fantasiosi da ludipipo, multipipo e via di seguito, si tratta del JDM programmer. Tutti hanno la particolarità di impiegare solo alcune resistenze... fin qui nulla di strano, almeno una volta li abbiamo certamente utilizzati tutti.

Fino a che... il PIC non lo vuole più programmare. E da qui in poi le solite prove, cambia la seriale, reinstalla il programma, cambia il PC, cambia anche il software. Niente da fare.

Che questo tipo di programmatore sia "fuori specifiche" è cosa nota, il PIC viene programmato utilizzando la tensione della seriale, spesso il programmatore non è neppure alimentato; altre volte una pila da 9V alimenta il chip con la mediazione di una zenerino da 5V mentre i 12V di programmazione sono prelevati da un pin della seriale.

Altre volte sul PC nuovo della seriale non c'è traccia e i convertitori USB > seriale fanno quel che possono, in questi casi il più delle volte assolutamente nulla.

A questo punto restano due alternative, un programmatore serio, meglio se "intelligente" oppure un esemplare semplice, ma collegato alla parallela dove i segnali sono più stabili e soprattutto considerando che la porta parallela di solito è ancora presente sul computer!

Dato l'uso saltuario la scelta è

caduta su quest'ultima soluzione. Il programmatore sarà alimentato a 13,5V prevede una serie di zoccoli per programmare gli esemplari più comuni, una uscita a cinque pin è disponibile per collegare un eventuale adattatore per programmare altri modelli inizialmente non previsti o per un connettore per smartcard.

Nulla di definitivo, un prototipo adatto all'uso amatoriale assemblato con la cura necessaria a conferirgli una buona stabilità meccanica, senza troppo spazio all'estetica, che impieghi un software di gestione piccolo, leggero e libero.

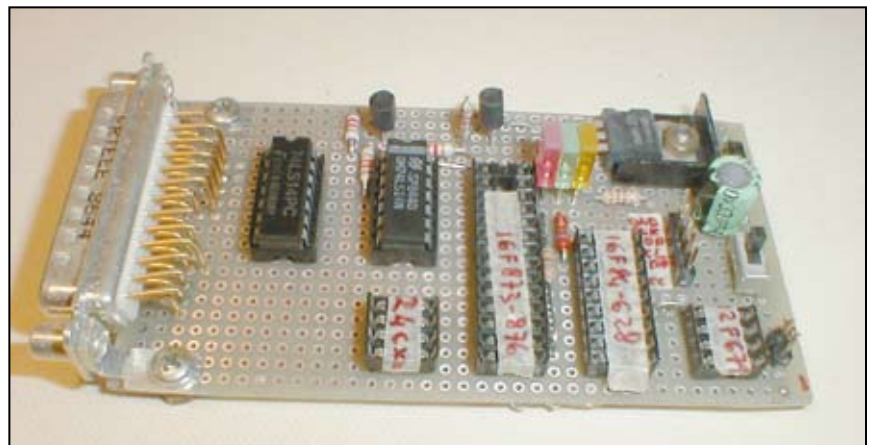
Lo schema elettrico

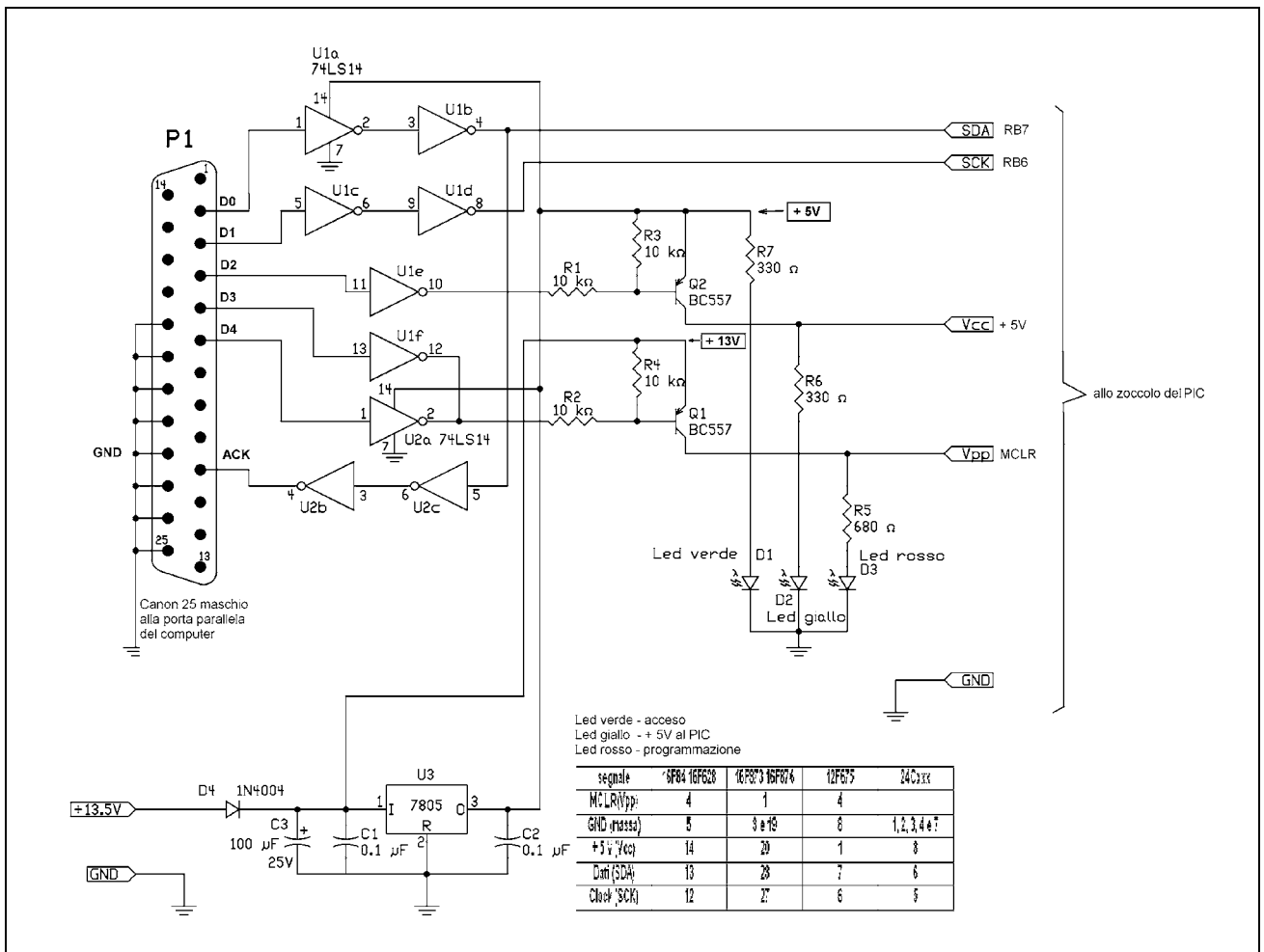
Anni fa ho avuto necessità di scrivere alcune EEPROM seriali (24C02 della microchip) e nell'occasione ho assemblato velocemente un programmatore parallelo, poi stupidamente mes-

so da parte, dunque la prima idea è stata quella di recuperare l'oggetto e verificare se era riutilizzabile. All'epoca i programmi di gestione dell'hardware erano sotto DOS e le EEPROM hanno un segnale di programmazione in meno rispetto ai PIC, scartato il recupero vediamo cosa c'è in giro oggi...

Tra i suoi kit la Gpe ne ha uno (MK 3160), si tratta di un programmatore a transistor su porta parallela con software di gestione in DOS e la gestione dei pin della parallela sono diversi. Lo schema è semplice e valido, la modifica è certamente possibile, ma vediamo cos'altro si trova...

Questo va meglio. Si tratta di una semplificazione del "pro-pic2" ad opera di Claudio e Dario (non ho altri riferimenti, non li conosco, ma li cito). Il chip utilizzato (7407) non è nei cassette, poi ha le uscite a collettore aperto, non mi piace, meglio un chip in più con la logica che si occupa





Elenco componenti

R1 = R2 = R3 = R4 = 10 kΩ
 R5 = 680 Ω
 R6 = R7 = 330 Ω
 C1 = C2 = 100 nF multistrato
 C3 = 100 μF 25V elettrolitico
 D1 = LED verde
 D2 = LED giallo
 D3 = LED rosso
 D4 = 1N4004
 Q1 = Q2 = BC557 (PNP al Si)
 U1 = U2 = 74 LS14 (o 74 LS04)
 U3 = LM7805
 P1 = connettore Canon 25 poli maschio
 Zoccoli DIL14 per U1 e U2
 Zoccoli per le PIC

dei livelli verso il PIC; contemporaneamente preferisco che la parallela sia, per quanto possibile, isolata da dove metterò le mani.

Due 74LS14 e due transistor è tutto quanto occorre, l'utilizzo di qualche porta in più permette di evitare il settaggio per inversione dei due segnali MCLR e VCC. Tre LED si occupano di indicare

lo stato del programmatore.

L'alimentazione è fornita dall'esterno, i 13,5V dell'alimentatore della stazione è perfetto, la corrente richiesta è minima, pochi mA ben filtrati. Il programmatore monta un 7805 da 1 A, abbondantemente sovradimensionato, non richiede alcun radiatore anche se sul prototipo è presente.

I pin 2 e 3 (segnali D0 e D1) della parallela sono rispettivamente i dati e il clock necessari alla scrittura del componente. I due segnali sono bufferizzati da due porte dell'LS14 che capovolgono due volte il segnale, così è anche per i dati in lettura che fanno capo due porte analoghe verso il pin 10 della parallela (segnale ACK)

L'alimentazione del componente (Vcc) è controllata dal pin 4 della parallela (segnale D2), la tensione di programmazione (Vpp) fa capo ai pin 5 e 6 (segna-

li D3 e D4). Come già espresso la presenza degli inverter evita il settaggio del programma per capovolgere i due segnali.

Le tensioni vengono fornite al componente per mezzo dei due transistor PNP: alla tensione di alimentazione (Vcc pari a 5V) fa capo il LED giallo, alla tensione di programmazione (Vpp pari a 13V) fa capo il LED rosso.

Il componente da programmare andrà sempre e comunque inserito nello zoccolo con il software sul PC già avviato e con il programmatore spento; questo andrà spento anche prima di rimuovere il PIC a programmazione eseguita.

Montaggio e collaudo

Come abbiamo visto lo schema elettrico è molto semplice, è sufficiente prestare la massima attenzione nel collegamento dei

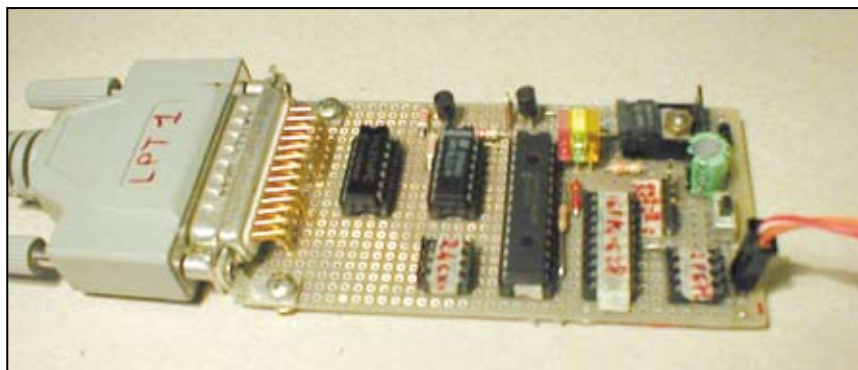
segnali verso gli zoccoli che ospiteranno i microcontrollori da programmare. Il prototipo è stato assemblato su un ritaglio di millefori da 100 x 55 mm, il connettore verso la parallela è un canon maschio 25 poli da stampato, in modo da poter sfruttare un vecchio cavo parallelo maschio/femmina di una unità I/O Mega Zip (o di un vecchio scanner parallelo). Sul connettore andranno tagliati tutti i pin lasciando integri solo i pin 2, 3, 4, 5, 6, 10 e le masse dal pin 18 al 25.

I due integrati sono identici, due 74LS14, ma due 74LS04 andranno ugualmente bene, è importante che si tratti del modello LS, non la versione CMOS 74HCxx. Sicuramente anche due vecchissimi 7414 o 7404 vanno ugualmente bene, anche se consumano qualche mA in più.

Si tratta di sei porte NOT, ovvero sei inverter, sono utilizzate a coppie di due per riportare i segnali allo stesso livello logico di partenza, isolando la porta parallela dal componente da programmare senza capovolgere lo stato logico del segnale. I due transistor effettuano a loro volta una inversione del segnale, dunque su questi segnali è presente una sola porta dell'LS14.

Il pin 10 è utilizzato durante il ciclo di lettura e di verifica del PIC.

Durante la fase di programmazione del PIC è importante che siano rispettate le tensioni di alimentazione e di programmazione. L'alimentazione è fornita da un 7805 tramite Q2, mentre quella di programmazione è prelevata direttamente dall'alimentazione esterna tramite Q1. I due transistor sono entrambi PNP al silicio da commutazione, due BC557 o analoghi. Come spesso accade qui non esiste alcuna limitazione e qualsiasi PNP in grado di sopportare un centinaio di mA andrà più che bene. È importante che le due resistenze collegate alla base siano presenti entrambe, sullo schema il valore suggerito è di 10 kΩ, sul prototipo ho utilizzato elementi da 8,2 kΩ, ma valori da 4,7 a 15 kΩ andranno



Programmatore all'opera

comunque bene. Nel caso se non dovessimo avere resistenze uguali avremo cura di collegare quella con il valore più alto tra la base e il positivo e quella con il valore più basso tra la base e il 74LS14 (4k7 e 22k in questa configurazione sono ancora valori accettabili). La commutazione è evidenziata dal LED rosso per la tensione di programmazione e giallo per quella di alimentazione.

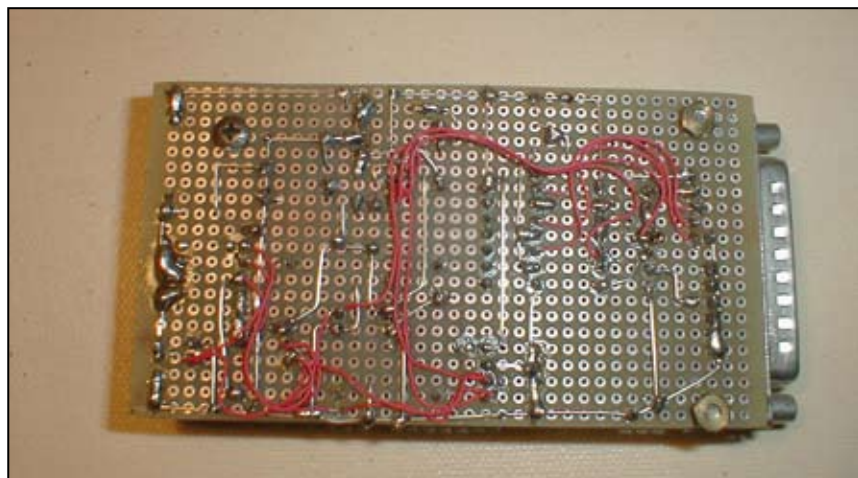
La tensione di programmazione standard è pari a 13 V (se non ricordo male), i limiti di tensione, dal datasheet Microchip, sono pari a 14V per la tensione di programmazione e 7,5 V per quella di alimentazione. Durante la programmazione si aumenta un poco la tensione di alimentazione fino a 6 V, un diodo collegato tra il comune e massa del 7805 aumenta la tensione di alimentazione dei soliti 0,6V, cosa che potrebbe talvolta essere utile; questa necessità non si è finora rivelata necessaria, ma la tengo pronta in un cassetto.

L'alimentazione esterna è fornita da un comune alimentatore in grado di fornire poche centinaia di mA. È necessario che questo non fornisca più di 14,5 V con cui si arriva vicino al limite superiore della tensione di programmazione massima del PIC.

In verità i PIC sono di solito piuttosto robusti e sopportano maltrattamenti che di solito distruggono altri componenti. Durante le prove ho fornito la tensione di programmazione (i 13 V) al pin sbagliato, ho inserito il PIC al contrario, e malgrado le scaldate è risultato alla fine perfettamente funzionante.

Questo non significa che dobbiamo sentirci in dovere di maltrattarlo a priori... è sempre una buona idea rimanere entro le specifiche del componente e prestare attenzione a quel che si fa.

Come espresso sopra il prototipo è stato assemblato a filo su un ritaglio di millefori, senza alcun contenitore che ne avrebbe complicato la realizzazione ob-



bligandomi a portare all'esterno solo i cinque collegamenti necessari alla programmazione e predisporre degli adattatori per ogni componente da programmare. I quattro zoccoli presenti sulla piastra mi permettono di operare sui PIC che per me sono più comuni ovvero la serie 16F84 e 16F628 sullo zoccolo a 18 pin, la serie 16F873/876 sullo zoccolo a 28 pin "stretto", la serie 12F675 sullo zoccolo a 8 pin di destra mentre quello di sinistra è adibito alla EEPROM 24Cxx (dalla 24C02 alla 24C256). Non ho montato a bordo il connettore per le smartcard riservandomi di utilizzarlo quale eventuale unità esterna...

È evidente che se le nostre esigenze in merito sono diverse basterà montare lo zoccolo per il PIC di nostro utilizzo e collegare solamente i pin di programmazione, vediamo nella tab. 1.

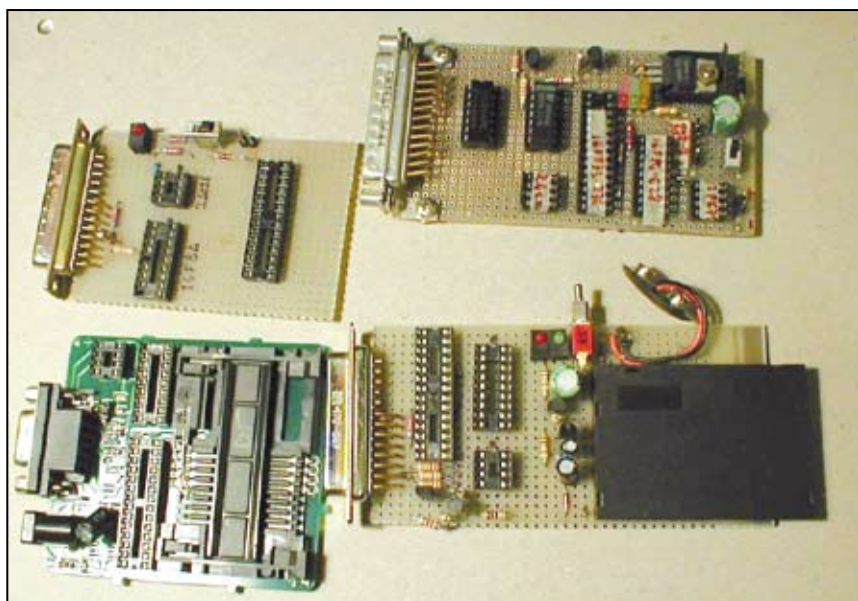
Per altri modelli basterà consultare un datasheet o il sito della Microchip per avere le connessioni di volta in volta necessarie.

Il software

Dopo un uso iniziale del vecchio icprog (ver. 1.06b) e su consiglio di Ezio, IK1FLG, sono passato a WinPic800 (ver.3.64e).

Il software è freeware ed è reperibile sul sito del produttore www.winpic800.com, compatibile con Win 98, NT, 2000, ME, XP e Vista, è multilingua tra cui l'italiano.

Appena lanciato per la prima volta il programma scegliamo il menù "impostazioni", quindi "hardware". Dobbiamo settare il tipo di programmatore (Propic2), settare la porta parallela in uso (tipicamente LPT1), eliminare il blocco della configurazione in alto a destra, quindi portarci nelle condizioni segnate in



Il programmatore con i suoi colleghi seriali.

figura 1, ovvero eliminando tutte le spunte dalla prima colonna, quella riferita alle inversioni dei segnali (colonna INV). A questo punto possiamo nuovamente bloccare la configurazione, cliccare su "applica" in basso a destra ed uscire da questo menù.

Dalla prima videata scegliamo ora la famiglia e il PIC di nostro interesse, quindi "file", "apri" e scegliamo il file da caricare sul componente. Sono accettati file con estensione .HEX, ovvero immagini delle tre aree di memoria del PIC.

Winpic800 prevede molte op-

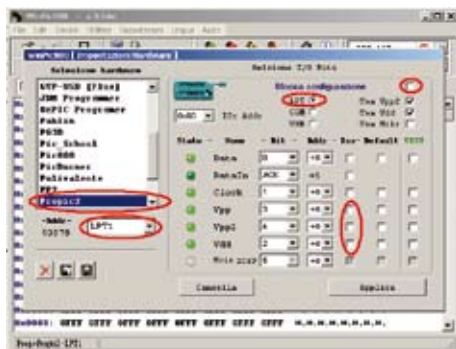
zioni tra cui l'identificazione del componente inserito nello zoccolo, dal menù "device" click su "rileva device" ci viene in risposta il componente che il software "vede" inserito nel programmatore. Se lo zoccolo è vuoto, o se il programmatore è spento (o scollegato) la risposta è "device sconosciuto".

In pochi minuti saremo certamente in grado di programmare il nostro primo PIC e in un'oretta gestiamo già il software con sufficiente padronanza.

Sicuramente winpic800 è più evoluto di icprog che comunque funziona perfettamente con questo programmatore, anche i settaggi dell'hardware sono praticamente identici.

Un vantaggio nell'uso è certamente la velocità, winpic800 insieme a propic2 è terribilmente più veloce del vecchio programmatore jdm con icprog!

La ragione scatenante di questo progetto è stata la realizzazione del rosmetro automatico LCD pubblicato su Radiokit di giugno 2005 a firma dei colleghi dell'ARI di Padova. Nell'occasione ringrazio (in ordine alfabetico) Beppe iwlego e Ezio ik1flg per lo stampato del rosmetro che entrambi mi hanno preparato. Poi Pino, ik1jns, per l'offerta del suo propic2 e tutti gli altri colleghi storici.



Tab. 1

segnale	16F84	16F628	16F873	16F876	12F675	24Cxxx
MCLR (Vpp)		4		1	4	
GND (massa)		5		8 e 19	8	1, 2, 3, 4 e 7
+ 5 V (Vcc)		14		20	1	8
Dati (SDA)		13		28	7	6
Clock (SCK)		12		27	6	5