

PROGRAMMATORE DEMOBOARD PER PIC

Prezzo: 39.34 €

Tasse: 8.66 €

Prezzo totale (con tasse): 48.00 €



Versatile programmatore per microcontrollori Microchip® FLASH PIC (tutti i moderni microcontrollori con memoria programma di tipo Flash) in grado di funzionare anche come demo-board per la verifica dei programmi più semplici. Il sistema si interfaccia alla porta seriale di qualsiasi PC nel quale andrà caricato l'apposito software (compreso nella confezione): l'utente potrà così programmare, leggere e testare la maggior parte dei micro della Microchip. Dispone di quattro zoccoli in grado di accogliere micro da 8, 14, 18 e 28 pin. Il kit comprende anche un micro vergine PIC16F627 riprogrammabile oltre 1.000 volte. Versione già montata e collaudata.

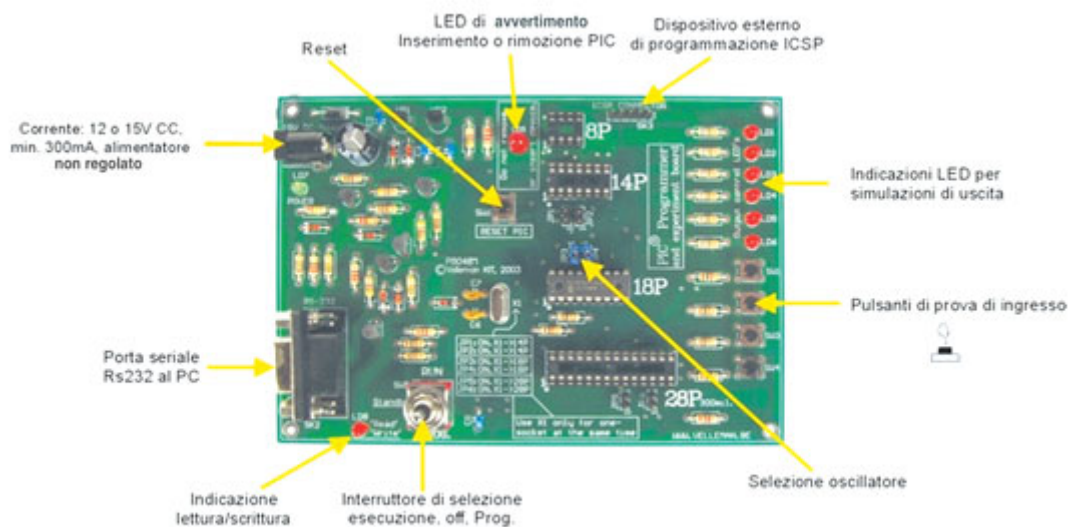
CARATTERISTICHE TECNICHE

- Adatto per la programmazione di microcontrollori Microchip® FLASH PIC™;
- Supporta 4 differenti formati: 4+4pin, 7+7pin 9+9pin e 14 + 14 pin;
- Quattro pulsanti e sei diodi LED per eseguire esperimenti con i programmi più semplici;
- Si collega facilmente a qualsiasi PC tramite la porta seriale;
- Include un microcontroller PIC16F627 che può essere riprogrammato fino a 1000 volte;
- Completo di software di compilazione e di programmazione;
- Alimentatore: 12÷15V cc, minimo 300mA, non stabilizzato (alimentatore non compreso);
- Supporta le seguenti famiglie di micro FLASH: PIC12F629, PIC12F675, PIC16F83, PIC16F84(A), PIC16F871, PIC16F872, PIC16F873, PIC16F874, PIC16F876, PIC16F627(A), PIC16F628(A), PIC16F630, ecc.
- Dimensioni: 145 mm x 100 mm.

Minimi requisiti di sistema:

- PC IBM Compatibile, processore Pentium o superiore.
 - Sistema operativo Windows™ 95/98/ME/NT/2000/XP.
 - Una porta RS232 libera.
-

CONNESSIONI E CONTROLLI



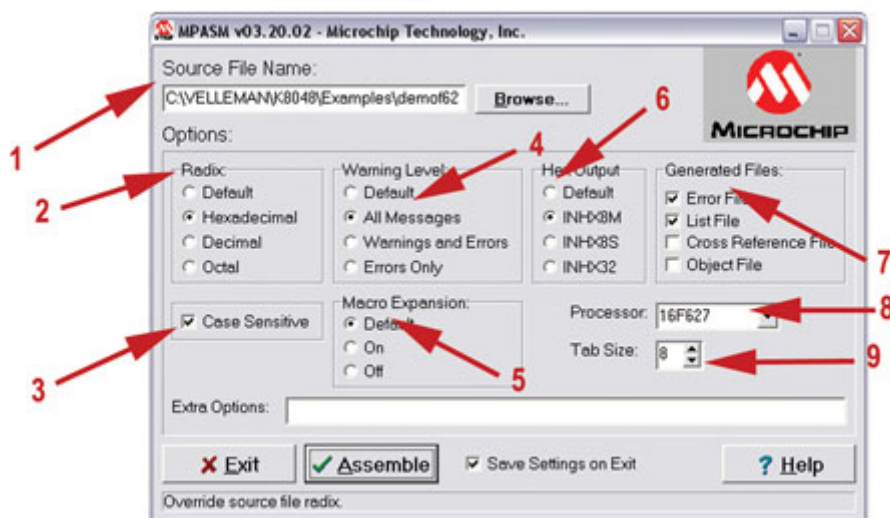
Nell'immagine sono evidenziate tutte le prese di ingresso/uscita, i led di segnalazione ed i vari controlli. Il PIC da programmare va inserito in uno dei quattro zocchi disponibili. Mediante i quattro pulsanti ed i sei led posti sul lato destro della basetta è possibile testare i programmi più semplici. Per l'alimentazione del programmatore è necessario utilizzare un adattatore da rete in grado di fornire una tensione non stabilizzata compresa tra 12 e 15V.

PROGRAMMAZIONE IN-CIRCUIT

Mediante questa tecnica è possibile programmare, cancellare e riprogrammare qualsiasi micro Flash già inserito nel suo circuito applicativo senza doverlo dissaldare dalla scheda. Ciò, tra l'altro, consente di operare con dispositivi di tipo Flash che per numero di piedini o configurazione del "case" non sono compatibili col nostro programmatore. Citiamo, ad esempio, i micro con passo 600 mil o quelli in contenitore QFP o PLCC come, ad esempio, il PIC16F876. La programmazione in-circuit, tuttavia, proprio perché effettuata nel circuito applicativo del micro, richiede una serie di precauzioni che riguardano le connessioni relative all'alimentazione ed alle porte RB6 e RB7 mediante le quali viene effettuata la programmazione. In altre parole quando si progetta e realizza il dispositivo finale nel quale andrà ad operare il micro bisogna prevedere l'impiego di due diodi supplementari e di due resistenze come indicato nel disegno in basso. Il diodo sull'alimentazione fa sì che la tensione VDD (+5V) proveniente dal programmatore alimenti esclusivamente il micro e non il resto del circuito; lo stesso concetto vale per la tensione di programmazione VPP che, grazie alla resistenza ed al diodo collegati al pin MCLR, può giungere esclusivamente al terminale di programmazione e non agli altri dispositivi presenti attorno al micro. Per quanto riguarda le due porte utilizzate per la programmazione (RB6 e RB7, rispettivamente ingresso di clock e connessione dati input/output), spesso questi I/O vengono impiegati anche per altri scopi (pilotare un led, leggere un livello logico, ecc). Per evitare che, durante la programmazione, questi circuiti possano "caricare" eccessivamente l'uscita del nostro dispositivo, è necessario prevedere due resistenze di disaccoppiamento da almeno 1 Kohm tra i pin del micro e la restante parte del circuito. A questo punto sulla scheda dove è montato il micro potremo prevedere l'impiego di un connettore a 5 poli (non bisogna dimenticare la massa!) e di un flat-cable col quale effettuare la connessione col nostro programmatore.

Ultimata questa prima fase, bisogna compilare il programma, ovvero convertire il listato che è stato scritto in un linguaggio comprensibile in codice macchina. A tale scopo possiamo utilizzare il programma MPASMWIN.EXE fornito con il kit . Dalla schermata principale possiamo selezionare tutti i parametri necessari per questa scopo:

- 1) Selezionare il nome ed il percorso del programma sorgente;
- 2) Selezionare il formato del codice macchina che si vuole generare (solitamente 'HEXADEMICAL');
- 3) Spuntare questa opzione se si desidera che vengano controllati gli errori di ortografia delle variabili;
- 4) Scegliere la tipologia di messaggi di errore che il programma deve generare (solitamente Error Only);
- 5) Scegliere una tipologia di funzionamento per la funzione Macro Expansion (nell'impostazione 'ON' le macro sono compilate nel file LIST - xxx.LIST);
- 6) Selezionare la lunghezza (in byte) della parola. Nel nostro caso impostare sempre 'INHX8M': altre scelte daranno luogo ad un codice macchina illeggibile o non funzionante.
- 7) Selezionare i files che si vogliono generare in aggiunta al codice macchina HEX. Non selezionare mai le opzioni "Cross Reference" e "Object" in quanto incompatibili con questo programmatore.
- 8) Selezionare il microprocessore utilizzato.
- 9) Indicare la larghezza di colonna (numero di caratteri) nei files di testo generati.



A questo punto è possibile cliccare su <ASSEMBLE> e se tutti i parametri sono corretti e non vi sono errori nel codice sorgente appare la schermata riportata in basso a sinistra. Il programma assembler genera i seguenti files:

.HEX è il file che contiene i codici operativi da inviare al PIC tramite il programmatore.

.LST è un file di testo in cui viene riportato il sorgente e la relativa traduzione in opcode. Non è utilizzabile per la programmazione del PIC ma è molto utile per verificare i processi di compilazione.

.ERR contiene la lista degli errori di compilazione riscontrati ed il numero di linea all'interno del sorgente in cui sono stati rilevati.

.COD file del codice utilizzato unicamente nell'ambiente di programmazione.

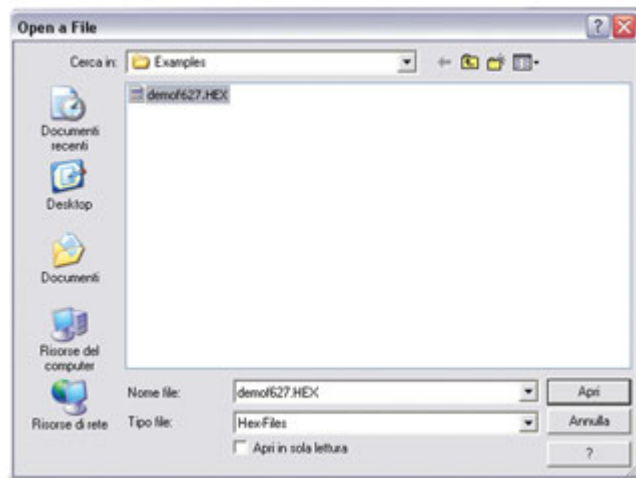
Solo il file .HEX viene utilizzato realmente per programmare il PIC. Vediamo come.

Il codice macchina (file .HEX) viene memorizzato nel processore tramite il PC e la scheda di programmazione utilizzando il software PROGIC2.EXE. A tale scopo è necessario:

- Collegare la scheda di programmazione al PC tramite la porta seriale libera tenendo presente che la linea IRQ di questa porta seriale (IRQ 3 o 4) non deve essere utilizzata da un altro dispositivo

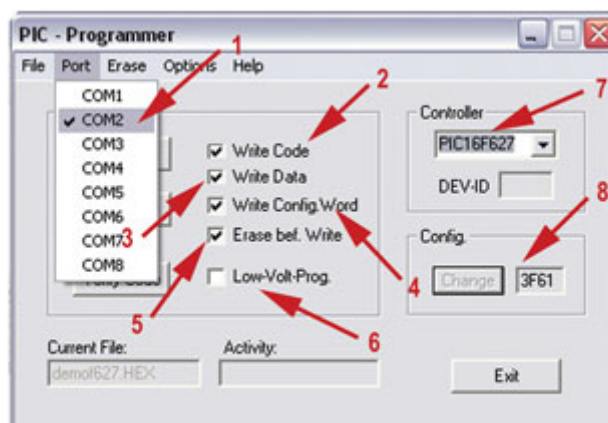
(ad esempio da un modem interno).

- Collegare l'alimentatore adatto e verificare che il led di alimentazione LD7 si attivi.
- Lanciare il programma 'PROGPIC2, PIC programmer' e cliccare su 'FILE' -> 'OPEN' e selezionare il programma compilato (Hex file 8M) che si vuole programmare (schermata in basso a destra).

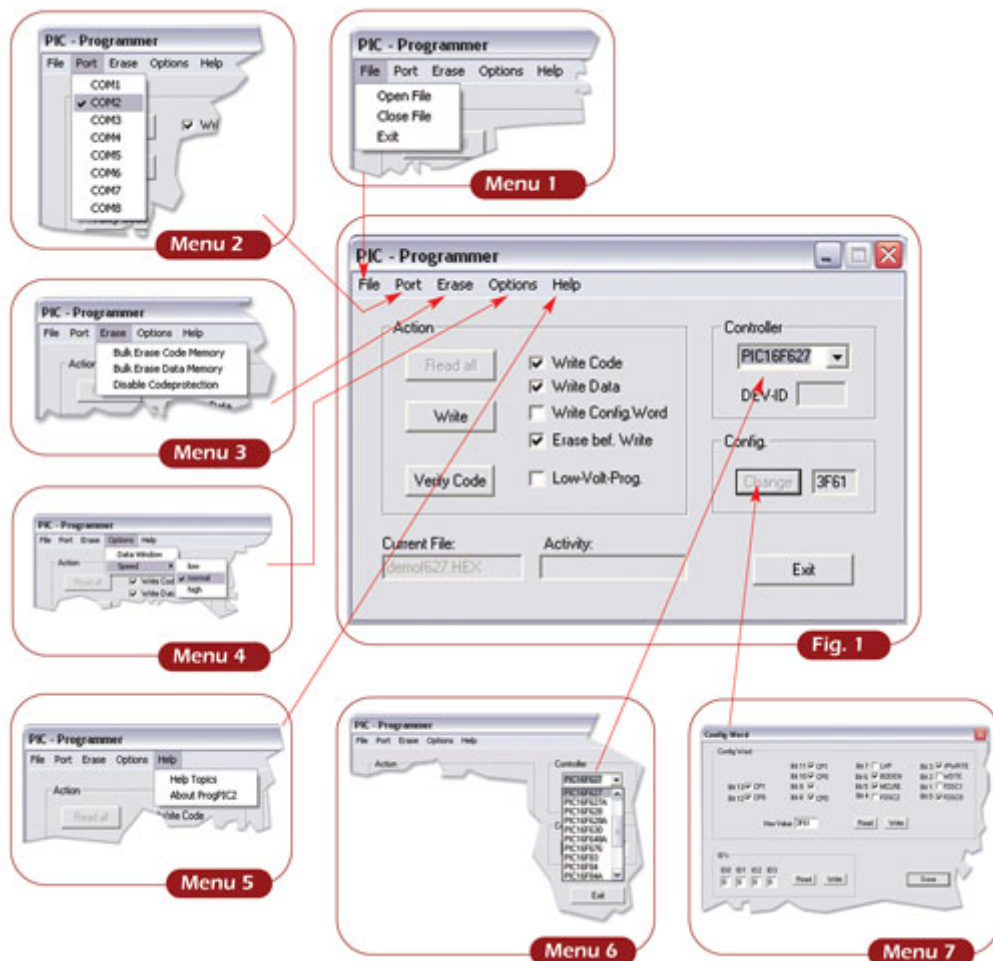


A questo punto, mediante la schermata riportata in basso, è necessario controllare i vari parametri:

- 1) Impostare il numero della porta seriale alla quale è collegato il programmatore.
- 2) Spuntare per memorizzare il codice (normalmente ON).
- 3) Spuntare per memorizzare i dati EEPROM (normalmente ON).
- 4) Spuntare per memorizzare i parametri di configurazione (normalmente ON).
- 5) Spuntare se si desidera cancellare completamente il micro prima di procedere con la scrittura (normalmente ON).
- 6) Spuntare per effettuare la programmazione in bassa tensione (Non selezionare: la nostra scheda utilizza una tensione di programmazione VPP di 13 V e non di 5 V).
- 7) Selezionare il microcontrollore utilizzato. Durante le prime prove utilizzare il micro fornito a corredo della scheda (un PIC16F627 o un PIC16F627A). Assicurarsi sempre di aver selezionato la sigla corretta.
- 8) Utilizzare per le definizioni/configurazioni del microcontrollore se queste non sono già state definite nel programma sorgente (come di solito accade).



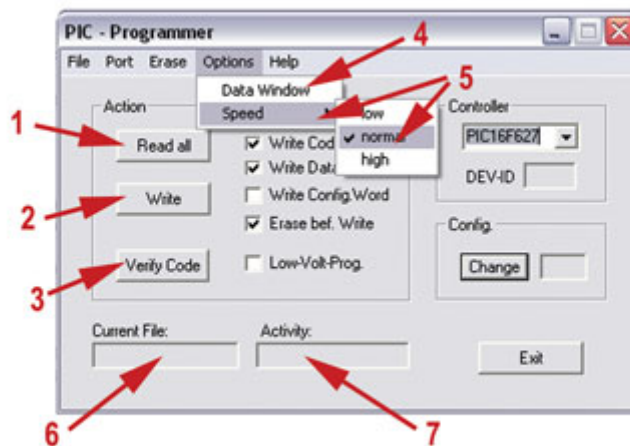
Nell'immagine che segue riportiamo il dettaglio dei vari menu:



A questo punto bisogna assicurarsi che il deviatore SW5 si trovi nella posizione centrale (standby); potremo quindi inserire il microcontrollore nello zoccolo adatto:

- utilizzare lo zoccolo IC1 nel caso di dispositivi a 4+4 pin;
- utilizzare lo zoccolo IC2 nel caso di dispositivi a 7+7 pin;
- utilizzare lo zoccolo IC3 nel caso di dispositivi a 9+9 pin;
- utilizzare lo zoccolo IC4 nel caso di dispositivi a 14+14 pin.

Successivamente dovremo spostare il deviatore SW5 nella posizione PROG: il led LD9 inizierà a lampeggiare segnalando che da questo momento non potremo spostare (inserire o togliere) il microcontrollore dallo zoccolo. Cliccando su WRITE il micro verrà programmato. Questo processo viene monitorato nella finestra ACTIVITY ed il led LD8 (READ/WRITE) lampeggia durante tutta la fase di scrittura. Una volta ultimata la programmazione spostate SW5 in posizione centrale (STANDBY): solo allora sarà possibile rimuovere il micro o passare alla modalità di prova (vedere i pulsanti di prova ed i LED di diagnostica dal menu Experiments). Se il programma è stato scritto per essere utilizzato con le risorse presenti sulla scheda (pulsanti e led), è possibile testare il dispositivo programmato spostando SW5 nella posizione RUN. Vediamo ora, facendo riferimento alla figura in basso, le funzioni corrispondenti ai vari comandi disponibili:



- 1) Read all: questa funzione consente di richiamare il programma presente nel micro purché questo non sia protetto (il codice macchina esadecimale viene salvato in un file di testo); i PIC per i quali il bit di protezione è stato attivato possono solamente essere cancellati.
- 2) Write: consente di programmare il micro. Ricordiamo che per utilizzare questa funzione è necessario mettere l'SW5 nella posizione PROG. Si può monitorare questo processo nel riquadro ACTIVITY. Il led LD8 (READ/WRITE) lampeggia durante questa fase.
- 3) Verify Code: permette di verificare se il micro è stato programmato correttamente. Questa operazione può essere effettuata esclusivamente se il micro non è protetto in lettura; ricordiamo anche che SW5 va spostato nella posizione PROG.
- 4) Data Window: permette all'utente di visualizzare e modificare i dati memorizzati nell' EEPROM del micro.
- 5) Speed: consente di modificare la velocità di programmazione adeguandola alle caratteristiche del PC utilizzato. Di solito va lasciata nella posizione NORMAL.
- 6) Activity: indica il nome del file caricato in memoria utilizzato per la programmazione.
- 7) Current File: permette di vedere quale indirizzo o area di indirizzi viene utilizzato dal programma per leggere o scrivere.

COLLEGAMENTI DEI PIN DI I/O

Di seguito riportiamo la tabella con la numerazione dei pin dei vari zoccoli a cui sono connessi i 4 pulsanti ed i 6 led utilizzati per verificare il funzionamento dei programmi più semplici. A ciascun pin corrisponde ovviamente una porta di I/O che varia a seconda dello zoccolo preso in considerazione.

	Pres a PIC 8 pin	Pres a PIC 14 pin	Pres a PIC 18 pin	Pres a PIC 28 pin
LD1	GP2 (pin 5)	RC0 (pin 10)	RB0 (pin 6)	RA0 (pin 2)
LD2	GP4 (pin 3)	RC1 (pin 9)	RB1 (pin 7)	RA1 (pin 3)
LD3	NC	RC2 (pin 8)	RB2 (pin 8)	RA2 (pin 4)
LD4	NC	RC3 (pin 7)	RB3 (pin 9)	RA3 (pin 5)
LD5	NC	RC4 (pin 6)	RB4 (pin 10)	RA4 (pin 6)
LD6	NC	NC	RB5 (pin 11)	RA5 (pin 7)
SW1	GP5 (pin 2)	RC5 (pin 5)	RA0 (pin 17)	RB0 (pin 21)
SW2	NC	RA2 (pin 11)	RA1 (pin 18)	RB1 (pin 22)
SW3	NC	NC	RA2 (pin 1)	RB2 (pin 23)
SW4	NC	NC	RA3 (pin 2)	RB4 (pin 25)

DOWNLOAD

Istruzioni in italiano, tutti i programmi, data-sheet, ecc (File Zip 5,86 MB): [K8048/VM111](#)