

**PIC LABS**

**C-170**

**Manuale  
Utente**

# SISTEMA C-170

Il sistema supporta le MCU PIC nei contenitori PDIP a 40, 28, 18, 14 e 8 pins. La programmazione dei microcontrollori può avvenire utilizzando la porta parallela del PC ed il software allegato, oppure collegando alla scheda l'In-Circuit Debugger ICD2 di Microchip. Ultimata la programmazione è possibile passare alla modalità RUN per debuggare il funzionamento del codice trasferito nella MCU. La scheda prevede un quarzo a 20 MHz e la possibilità di inserire un secondo quarzo del valore desiderato. Sono disponibili 5 connettori con tutte le porte di I/O dei microcontrollori. E' previsto un pulsante di Reset.

Il sistema risulta scomponibile in due parti:

- sezione di programmazione: scheda C-202
- sezione socket (con zoccoli): scheda C-203

La confezione non contiene: il cavo di collegamento alla porta parallela del PC.

Per utilizzare la scheda occorre disporre di: una sorgente di alimentazione a 17 Vdc minimo 300 mA; un PC IBM compatibile con Windows(TM) 95/98/ME/NT/2000/XP, lettore CDROM, una porta parallela libera.

## SOMMARIO

### SEZIONE DI PROGRAMMAZIONE

introduzione	pag. 3
Collegamento al PC	pag. 4
Segnalazioni	pag. 4
Jumper linee di programmazione	pag. 5
Jumper PGM	pag. 6
Connettore ICSP (In-Circuit Serial Programming)	pag. 7
Schema elettrico	pag. 8

### SEZIONE SOCKET

Introduzione	pag. 11
Socket per MCU a 40 pins	pag. 13
Socket per MCU a 28 pins	pag. 13
Socket per MCU a 18 pins	pag. 14
Socket per MCU a 14 pins	pag. 14
Socket per MCU a 8 pins	pag. 15
Connettore di ingresso	pag. 15
Connettori di uscita	pag. 15
Connettore per ICD2	pag. 16
Utilizzo di ICD2	pag. 17
Schema elettrico	pag. 19

### IL SOFTWARE DI PROGRAMMAZIONE

Introduzione	pag. 21
Installazione	pag. 21
Avvio	pag. 21
Configurazione	pag. 23

### PROGRAMMARE CON IC-PROG

Operazioni preliminari	pag. 24
MCU a 40 pins	pag. 26
MCU a 28 pins	pag. 30
MCU a 18 pins	pag. 32
MCU a 14 pins	pag. 34
MCU a 8 pins	pag. 36

Caratteristiche tecniche

pag. 38

# SEZIONE DI PROGRAMMAZIONE

## Introduzione

La scheda C-202 contiene l'elettronica necessaria per la programmazione dei microcontrollori PIC. Il sistema prevede l'utilizzo di due linee per il Clock e per i Dati e l'applicazione di una tensione di 13.2V al piedino MCLR.

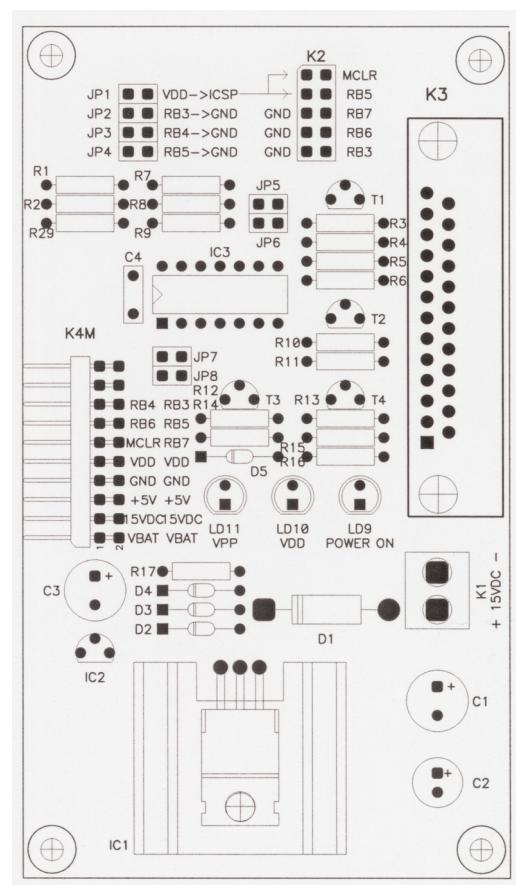
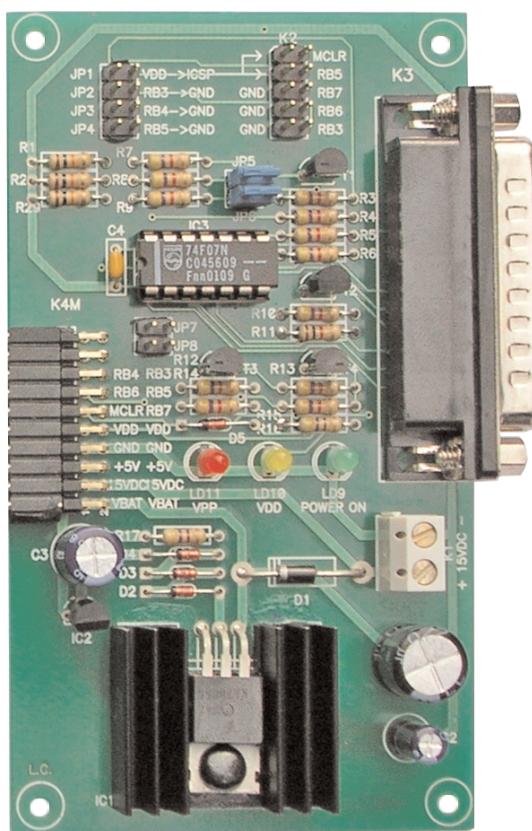
Per le MCU PIC che implementano anche la modalità di programmazione LVP (Low Voltage Programming) si rende necessario impostare sul piedino denominato PGM un livello logico basso. Il circuito necessita di una tensione di alimentazione di 17 Vdc.

La tensione di alimentazione va applicata al morsetto K1 rispettando la polarità.

Il diodo D1 protegge la scheda in caso di inversione di polarità.

Tramite l'integrato IC2 (7805) viene ricavata una tensione regolata di +5V, il diodo verde LD9 indica la presenza di tale tensione.

L'integrato IC1 (78L12) provvede a generare una tensione regolata di +13.2V necessaria nelle fasi di programmazione.



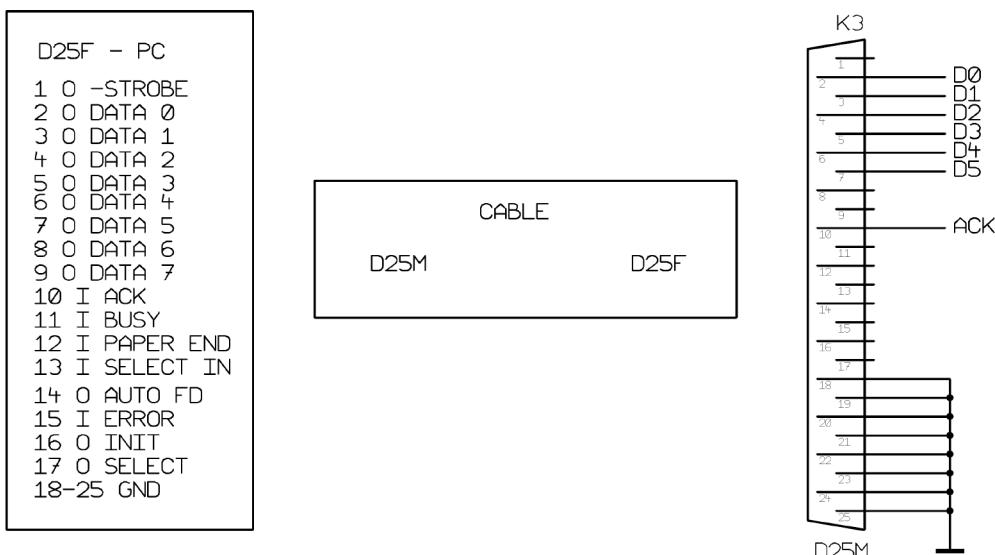
## Collegamento al PC

La scheda va collegata alla porta parallela di un PC IBM compatibile su cui dovrà essere installato il software IC-Prog. Per il collegamento utilizzare un cavo parallelo a 25 conduttori con un connettore maschio D25 da un lato (PC) e un connettore femmina D25 dall'altro (scheda C-202).

Della porta parallela vengono utilizzati i segnali di DATA da 0 a 5 e il segnale di ACK con i seguenti significati:

- **D0** segnale DATO dal PC verso il microcontrollore (viene applicato alla porta **RB7**)
- **ACK** segnale DATO dal microcontrollore verso il PC (viene prelevato dalla porta **RB7**)
- **D1** segnale CLK dal PC verso il microcontrollore (viene applicato alla porta **RB6**)
- **D2** controlla la tensione Vdd di alimentazione del microcontrollore (coincide con il piedino **VDD**)
- **D3, D4 e D5** controllano la tensione di programmazione del microcontrollore (coincide con il piedino **MCLR**)

Tutti questi segnali sono isolati dalla porta parallela dall'integrato IC3, in questo modo non sarà mai possibile danneggiare la porta parallela del computer.



## Segnalazioni

La scheda prevede tre LED di segnalazione:

**LD9** di colore verde si accende in presenza della tensione **+5V**

**LD10** di colore giallo si accende in presenza della tensione **VDD**

**LD11** di colore rosso si accende in presenza della tensione **VPP**

## Jumper linee di programmazione

La programmazione / lettura dei microcontrollori PIC avviene serialmente utilizzando i piedini **PGD** (Program Data), **PGC** (Program Clock). Risultano poi interessati alla programmazione il piedino di alimentazione positiva **VDD**, il piedino **MCLR/** (Memory Clear) a cui viene applicata la tensione di programmazione e, se disponibile, il piedino **PGM** (Program Mode).

**PGD** (Program Data) coincide con la porta **RB7** del microcontrollore

**PGC** (Program Clock) coincide con la porta **RB6** del microcontrollore

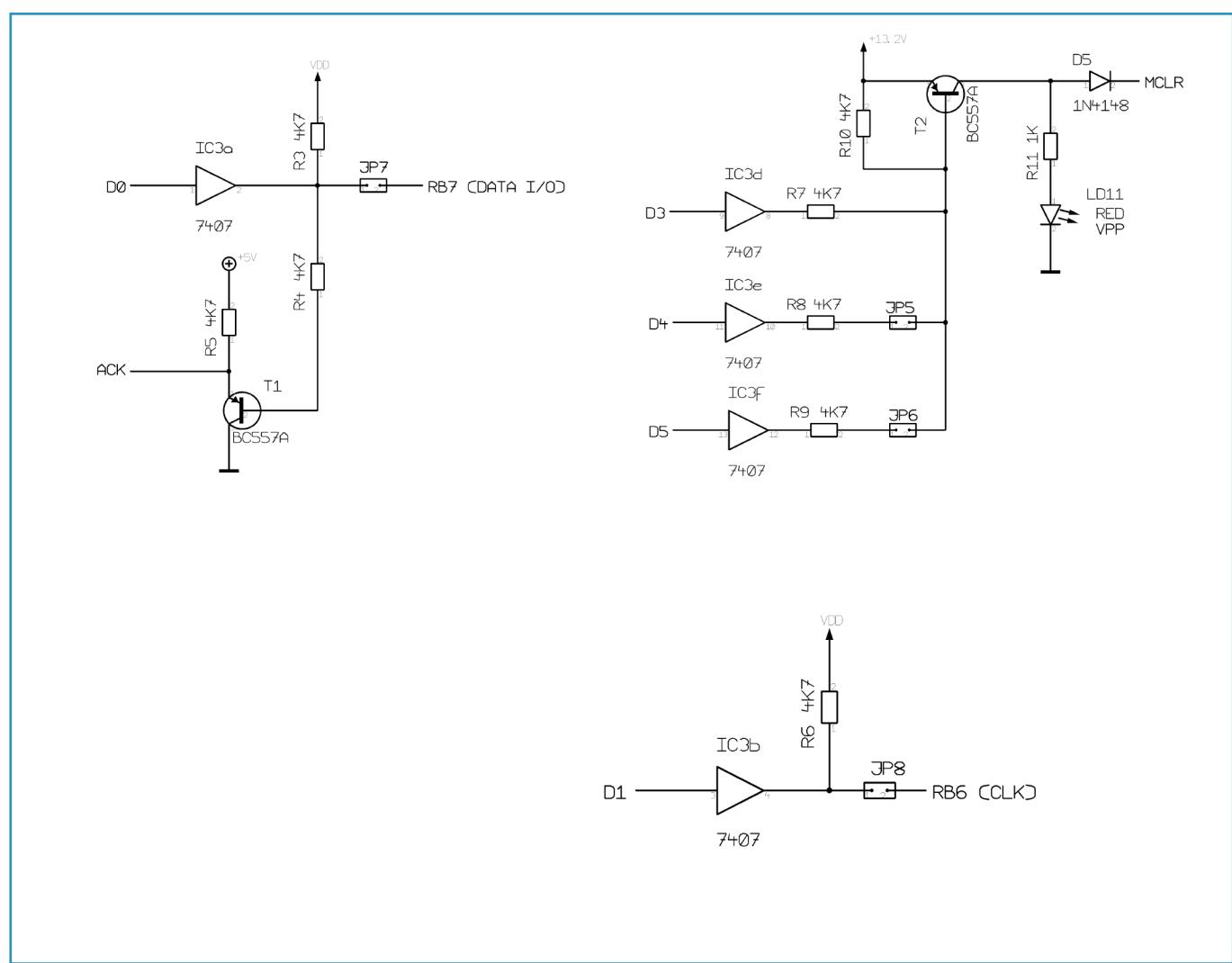
**VPP** (Tensione di programmazione) coincide con la linea **MCLR/** del microcontrollore

**PGM** (Program Mode) coincide con la porta **RB3** oppure **RB4** oppure **RB5** del microcontrollore

Le linee seriali PGD e PGC possono essere interrotte grazie ai jumper JP7 e JP8:

**JP7** interrompe la linea **PGD**

**JP8** interrompe la linea **PGC**



## Jumper PGM

La scheda prevede 3 jumper che consentono di collegare le porte RB3, RB4 e RB5 del microcontrollore a massa attraverso una resistenza da 10 Kohm.

chiudendo **JP2** viene portata a massa attraverso la resistenza R1 da 10K la porta **RB3** della MCU

chiudendo **JP3** viene portata a massa attraverso la resistenza R29 da 10K la porta **RB4** della MCU

chiudendo **JP4** viene portata a massa attraverso la resistenza R2 da 10K la porta **RB5** della MCU

Qualora il microcontrollore da programmare preveda la linea PGM è necessario chiudere il relativo jumper prima di iniziare la fase di programmazione. Per sapere se la linea PGM è prevista in un determinato microcontrollore occorre consultare il relativo datasheet.

In linea generale vale la seguente regola:

famiglia **PIC16F87X**:

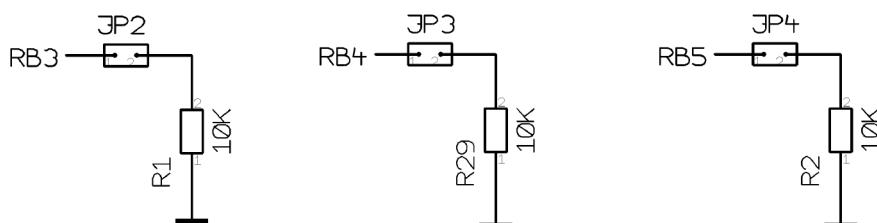
la linea PGM è prevista sulla porta **RB3**, chiudere il jumper JP2 in programmazione

famiglia **PIC16F62X**:

la linea PGM è prevista sulla porta **RB4**, chiudere il jumper JP3 in programmazione

famiglia **PIC18FXXX**:

la linea PGM è prevista sulla porta **RB5**, chiudere il jumper JP4 in programmazione



## Connettore ICSP (In-Circuit Serial programming)

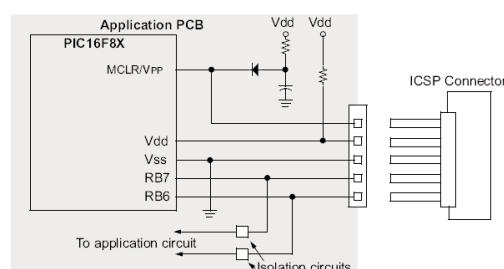
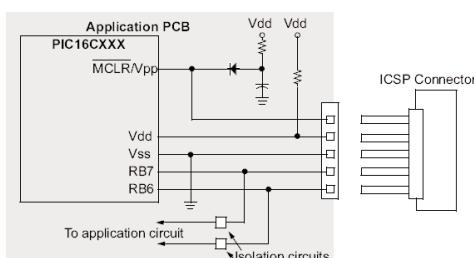
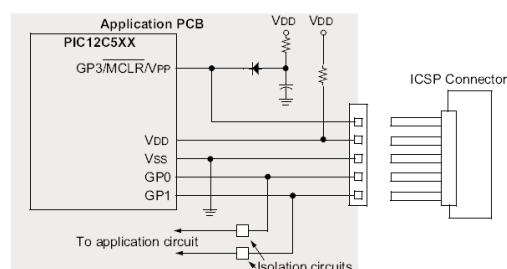
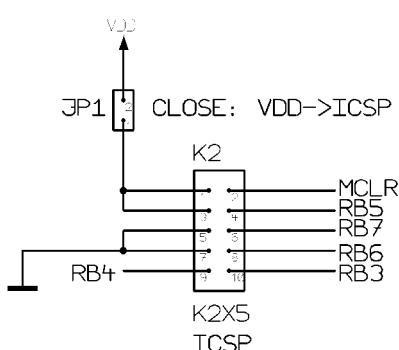
Tutti i segnali necessari alla programmazione / lettura dei microcontrollori PIC sono disponibili sul connettore **K2** a 10 piedini.

Risulta così possibile programmare il microcontrollore montato sulla scheda utente senza la necessità di rimuoverlo dalla scheda stessa.

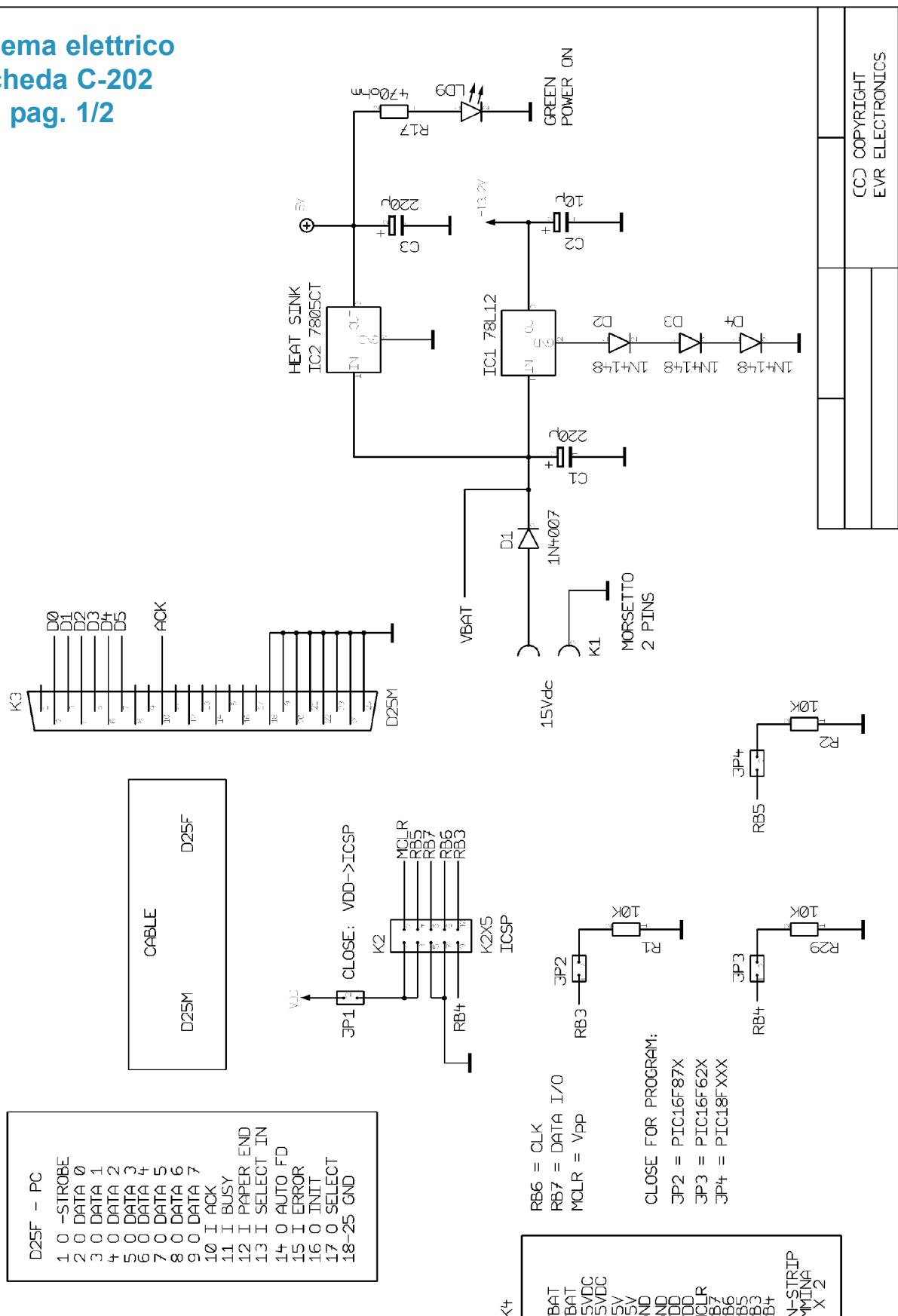
Il jumper **JP1** interrompe la tensione **VDD** verso il connettore K2.

Per la programmazione In-Circuit è necessario prevedere sulla scheda utente uno specifico Hardware, a tale scopo fare riferimento al documento **ICSP Guide** disponibile sul sito Microchip.

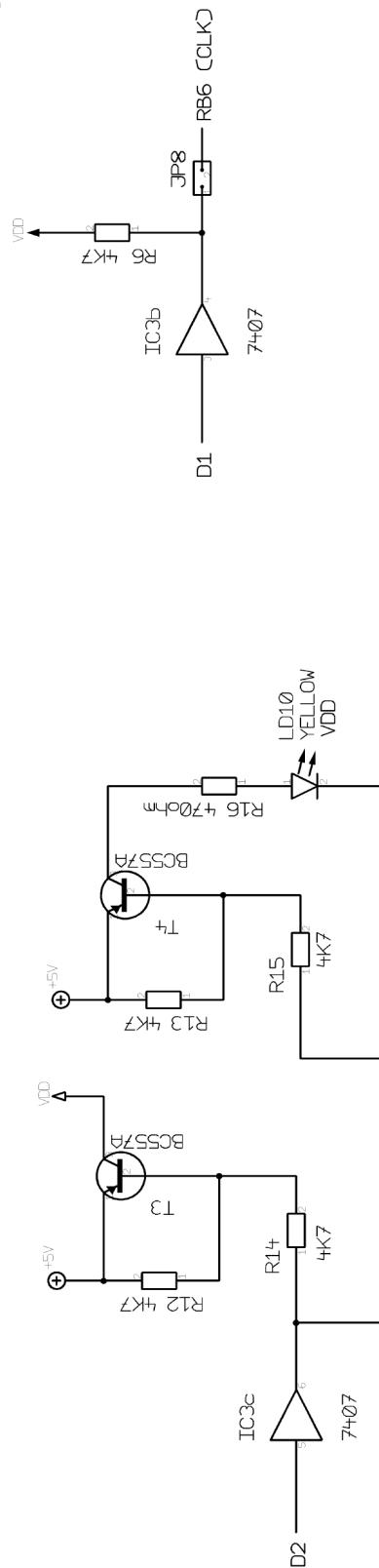
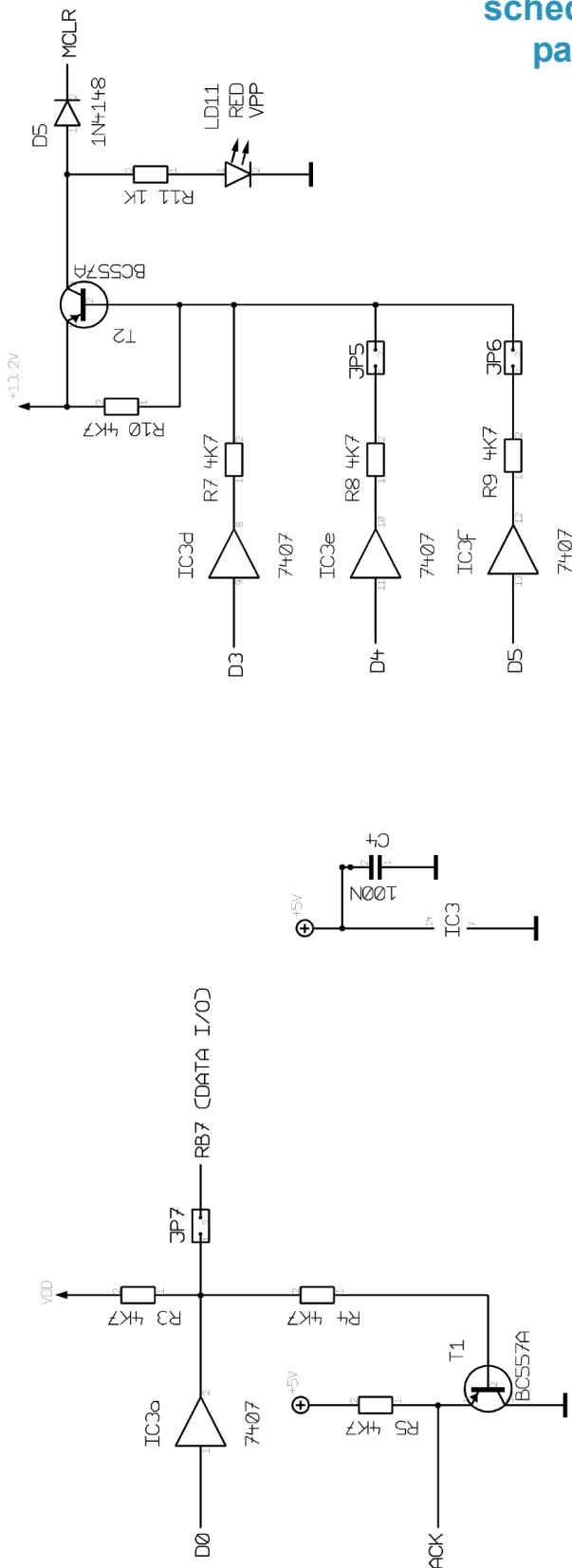
Riportiamo di seguito la piedinatura del connettore K2 e lo schema tipico richiesto dalla programmazione In-Circuit.



**Schema elettrico  
scheda C-202  
pag. 1/2**



**Schema elettrico  
scheda C-202  
pag. 2/2**



REV 1 - 27 SETTEMBRE 2004	C-202 PIC LAB PROG
(C) COPYRIGHT EVR ELECTRONICS	

**PART LIST C-202**

<b>R1:</b>	10 Kohm	$\frac{1}{4}$ W	<b>C1:</b>	220 $\mu$ F	35 V
<b>R2:</b>	10 Kohm	$\frac{1}{4}$ W	<b>C2:</b>	10 $\mu$ F	50 V
<b>R3:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>C3:</b>	220 $\mu$ F	35 V
<b>R4:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>C4:</b>	100 nF	
<b>R5:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>IC1:</b>	78L12	
<b>R6:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>IC2:</b>	7805	
<b>R7:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>IC3:</b>	7407	
<b>R8:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>T1:</b>	BC557A	
<b>R9:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>T2:</b>	BC557A	
<b>R10:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>T3:</b>	BC557A	
<b>R11:</b>	1 Kohm	$\frac{1}{4}$ W	<b>T4:</b>	BC557A	
<b>R12:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>K1:</b>	MORSETTO 2 PINS	
<b>R13:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>K2:</b>	PIN-STRIP M 5 + 5 PINS	
<b>R14:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>K3:</b>	D25M PINS 90 GRADI	
<b>R15:</b>	4,7 Kohm	$\frac{1}{4}$ W	<b>K4M:</b>	PIN-STRIP M 10 + 10 90°	
<b>R16:</b>	470 ohm	$\frac{1}{4}$ W	<b>JP1:</b>	PIN-STRIP 2 PINS	
<b>R17:</b>	470 ohm	$\frac{1}{4}$ W	<b>JP2:</b>	PIN-STRIP 2 PINS	
<b>R29:</b>	10 Kohm	$\frac{1}{4}$ W	<b>JP3:</b>	PIN-STRIP 2 PINS	
<b>D1:</b>	1N4007		<b>JP4:</b>	PIN-STRIP 2 PINS	
<b>D2:</b>	1N4148		<b>JP5:</b>	PIN-STRIP 2 PINS	
<b>D3:</b>	1N4148		<b>JP6:</b>	PIN-STRIP 2 PINS	
<b>D4:</b>	1N4148		<b>JP7:</b>	PIN-STRIP 2 PINS	
<b>D5:</b>	1N4148		<b>JP8:</b>	PIN-STRIP 2 PINS	
<b>LD9:</b>	LED GREEN 3 MM		-	SOCKET 7 + 7 PINS	
<b>LD10:</b>	LED YELLOW 3 MM		-	DISSIP 25 X 29 MM	
<b>LD11:</b>	LED RED 3 MM		-	N. 6 JUMPERS F	
			-	CS C202-1	

# SEZIONE SOCKET

## Introduzione

La scheda C-203 supporta le MCU PIC nei contenitori PDIP a 40, 28, 18, 14 e 8 pins.

E disponibile uno zoccolo tornito a 40 pins in cui possiamo inserire qualsiasi MCU in case PDIP40 come ad esempio:16F74, 16F77, 16F871, 16F874, 16F877, 18F4320, 18F442, 18F452, 18F448, 18F458 ...

E disponibile uno zoccolo tornito a 28 pins in cui possiamo inserire qualsiasi MCU in case PDIP28 come ad esempio:16F72, 16F73, 16F76, 16F870, 16F872, 16F873, 16F876, 18F2020, 18F2320, 18F242, 18F252, 18F248, 18F258 ...

E disponibile uno zoccolo tornito a 18 pins in cui possiamo inserire qualsiasi MCU in case PDIP18 come ad esempio:16C620, 16C622, 16C710, 16C711, 16C712, 16C715, 16C716, 16F627, 16F628, 16F84, 16F818, 16F819 ...

E disponibile uno zoccolo tornito a 14 pins in cui possiamo inserire qualsiasi MCU in case PDIP14 come ad esempio:16F684, 16F630, 16F688, 16F676, 16F636, 16F505 ...

E disponibile uno zoccolo tornito a 8 pins in cui possiamo inserire qualsiasi MCU in case PDIP8 come ad esempio:12C508, 12C509, 12C518, 12C519, 12C671, 12C672, 12C673, 12C674, 12F629, 12F675 ...

Se si utilizzano MCU in contenitore non PDIP è necessario procurarsi degli appositi adattatori.

Tutte le linee di ingresso / uscita sono disponibili su pratici connettori a passo 2,54 mm:  
K5 PortA; K6 PortB; K7 PortC; K8 PortD; K9 PortE.

Ogni linea di I/O viene riportata dagli zoccoli ai connettori K5, K6, K7, K8 e K9 e ai connettori di uscita K11M, K12M e K13M. La serigrafia della scheda permette una immediata identificazione di tutte le linee di I/O di ogni zoccolo e di ogni connettore.

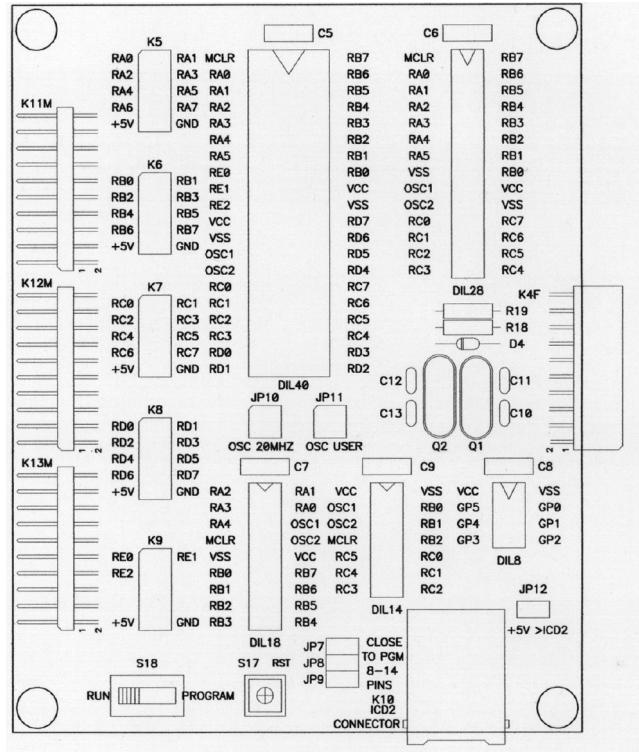
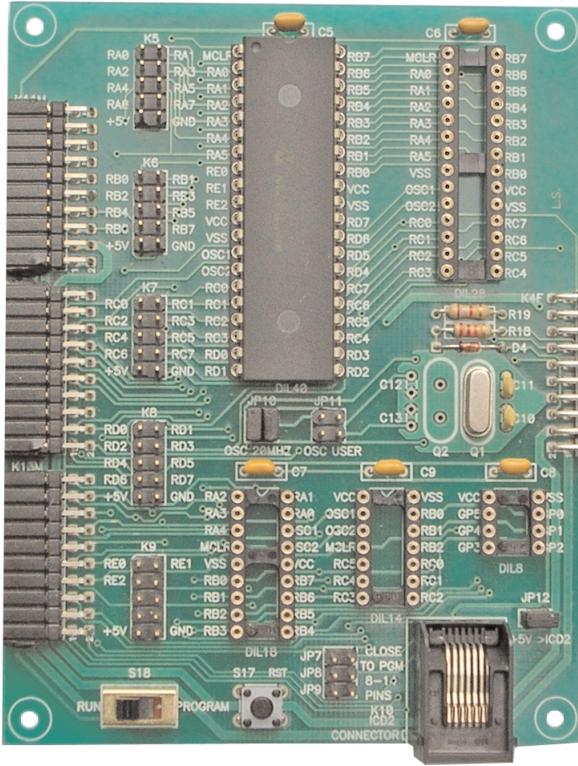
**Per quanto riguarda lo zoccolo ad 8 pins (DIL8) le linee GP sono state collegate alle linee PortB:  
GP0 = RB0; GP1 = RB1; GP2 = RB2; GP3 = RB3; GP4 = RB4; GP5 = RB5.**

**Per quanto riguarda lo zoccolo a 14 pins (DIL14) i piedini 13, 12, 11 sono collegati rispettivamente alle linee RB0, RB1 e RB2. Il piedino 4 è utilizzato come Master Clear (MCLR), mentre i piedini 2 e 3 sono utilizzati come OSC1 e OSC2.**

Tutte le linee OSC1 di ogni zoccolo (tranne DIL8) sono collegate tra loro e portate ai jumper JP10 e JP11; stesso discorso per le linee OSC2. Chiudendo JP10 (inserendo due ponticelli in posizione verticale) viene selezionato il quarzo da 20MHz montato sulla scheda; chiudendo JP11 (inserendo due ponticelli in posizione verticale) si seleziona il quarzo Q2 (non montato) con i relativi condensatori C12 e C13 (non montati).

I piedini di alimentazione di ogni zoccolo (pin VCC) sono collegati insieme tra loro e portati al comune del deviatore S18: in posizione PROGRAM le alimentazioni VCC risultano collegate con il segnale VDD gestito dalla scheda di programmazione; in posizione RUN le alimentazioni VCC risultano collegate al +5V fornito dalla scheda di programmazione.

## Note



## Socket per MCU a 40 pins

Piedinatura dei microcontrollori PIC a 40 pins.

La programmazione seriale utilizza le seguenti linee:

**PGD** (Program Data): RB7 **pin 40**

**PGC** (Program Clock): RB6 **pin 39**

**PGM** (Program Mode): RB3 **pin 36**

**VPP** (Tensione di programmazione): MCLR **pin 1**

**VDD** (Alimentazione positiva): Vdd **pin 11 e 32**

**GND** (Massa): Vss **pin 12 e 31**

MCLR	X <sub>1</sub> MCLR	RB7	40 X RB7
RA0	X <sub>2</sub> RA0	RB6	39 X RB6
RA1	X <sub>3</sub> RA1	RB5	38 X RB5
RA2	X <sub>4</sub> RA2	RB4	37 X RB4
RA3	X <sub>5</sub> RA3	RB3	36 X RB3
RA4	X <sub>6</sub> RA4	RB2	35 X RB2
RA5	X <sub>7</sub> RA5	RB1	34 X RB1
RE0	X <sub>8</sub> RE0	RB0	33 X RB0
RE1	X <sub>9</sub> RE1	Vdd	32 + VCC
RE2	X <sub>10</sub> RE2	Vss	31 VSS
VCC	X <sub>11</sub> Vdd	RD7	30 X RD7
VSS	X <sub>12</sub> Vss	RD6	29 X RD6
OSC1	X <sub>13</sub> OSC1	RD5	28 X RD5
OSC2	X <sub>14</sub> OSC2	RD4	27 X RD4
RC0	X <sub>15</sub> RC0	RC7	26 X RC7
RC1	X <sub>16</sub> RC1	RC6	25 X RC6
RC2	X <sub>17</sub> RC2	RC5	24 X RC5
RC3	X <sub>18</sub> RC3	RC4	23 X RC4
RD0	X <sub>19</sub> RD0	RD3	22 X RD3
RD1	X <sub>20</sub> RD1	RD2	21 X RD2

PIC(DIL40)  
SOCKET 40PIN

## Socket per MCU a 28 pins

Piedinatura dei microcontrollori PIC a 28 pins.

La programmazione seriale utilizza le seguenti linee:

**PGD** (Program Data): RB7 **pin 28**

**PGC** (Program Clock): RB6 **pin 27**

**PGM** (Program Mode): RB3 **pin 24**

**VPP** (Tensione di programmazione): MCLR **pin 1**

**VDD** (Alimentazione positiva): Vdd **pin 20**

**GND** (Massa): Vss **pin 8 e 19**

MCLR	X <sub>1</sub> MCLR	RB7	28 X RB7
RA0	X <sub>2</sub> RA0	RB6	27 X RB6
RA1	X <sub>3</sub> RA1	RB5	26 X RB5
RA2	X <sub>4</sub> RA2	RB4	25 X RB4
RA3	X <sub>5</sub> RA3	RB3	24 X RB3
RA4	X <sub>6</sub> RA4	RB2	23 X RB2
RA5	X <sub>7</sub> RA5	RB1	22 X RB1
VSS	X <sub>8</sub> Vss	RB0	21 X RB0
OSC1	X <sub>9</sub> OSC1	Vdd	20 + VCC
OSC2	X <sub>10</sub> OSC2	Vss	19 VSS
RC0	X <sub>11</sub> RC0	RC7	18 X RC7
RC1	X <sub>12</sub> RC1	RC6	17 X RC6
RC2	X <sub>13</sub> RC2	RC5	16 X RC5
RC3	X <sub>14</sub> RC3	RC4	15 X RC4

PIC(DIL28)  
SOCKET 28PIN

## Socket per MCU a 18 pins

Piedinatura dei microcontrollori PIC a 18 pins.

La programmazione seriale utilizza le seguenti linee:

**PGD** (Program Data): RB7 **pin 13**

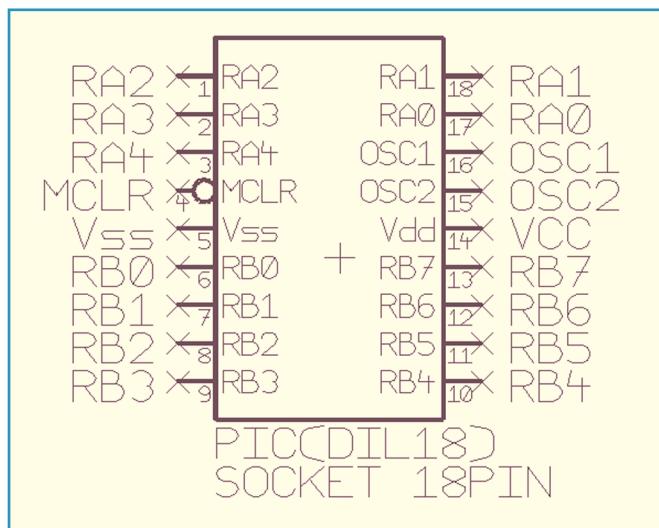
**PGC** (Program Clock): RB6 **pin 12**

**PGM** (Program Mode): RB4 **pin 10**

**VPP** (Tensione di programmazione): MCLR **pin 4**

**VDD** (Alimentazione positiva): Vdd **pin 14**

**GND** (Massa): Vss **pin 5**



## Socket per MCU a 14 pins

Piedinatura dei microcontrollori PIC a 14 pins.

La programmazione seriale utilizza le seguenti linee:

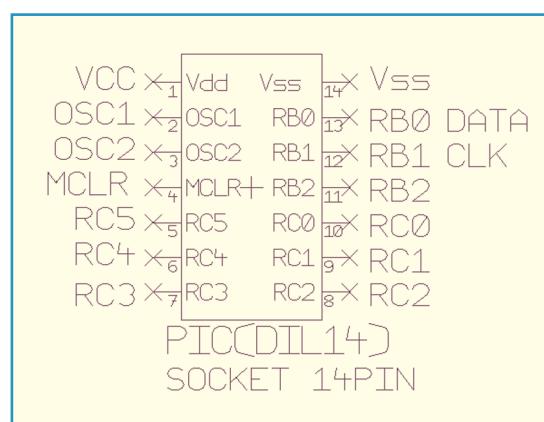
**PGD** (Program Data): RB0 **pin 13**

**PGC** (Program Clock): RB1 **pin 12**

**VPP** (Tensione di programmazione): MCLR **pin 4**

**VDD** (Alimentazione positiva): Vdd **pin 1**

**GND** (Massa): Vss **pin 14**



I piedini 13, 12, 11 sono collegati rispettivamente alle linee RB0, RB1 e RB2.

Il piedino 4 è utilizzato come Master Clear (MCLR), mentre i piedini 2 e 3 sono utilizzati come OSC1 e OSC2.

## Socket per MCU a 8 pins

Piedinatura dei microcontrollori PIC a 8 pins.

La programmazione seriale utilizza le seguenti linee:

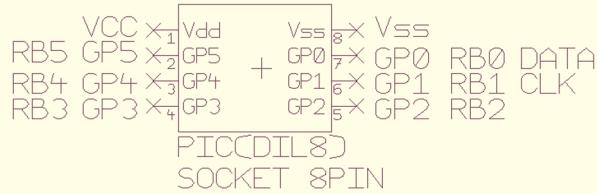
**PGD** (Program Data): GP0 pin 7

**PGC** (Program Clock): GP1 pin 6

**VPP** (Tensione di programmazione): GP3 pin 4

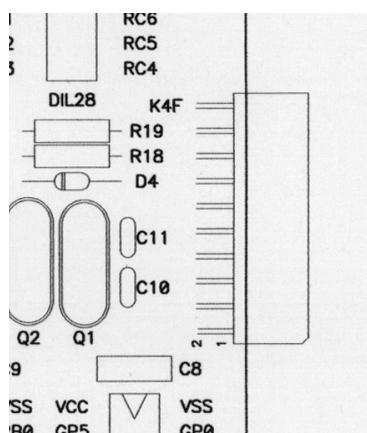
**VDD** (Alimentazione positiva): Vdd pin 1

**GND** (Massa): Vss pin 8



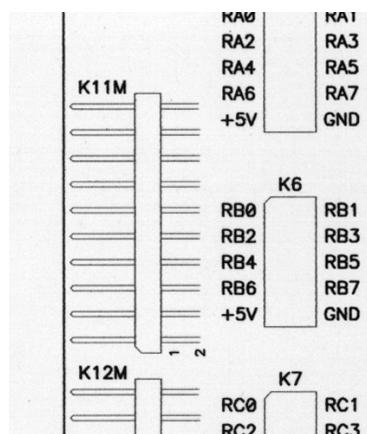
Le linee GP sono state collegate alle linee PortB:  
**GP0 = RB0; GP1 = RB1; GP2 = RB2;**  
**GP3 = RB3; GP4 = RB4; GP5 = RB5.**

## Connettore di ingresso K4F



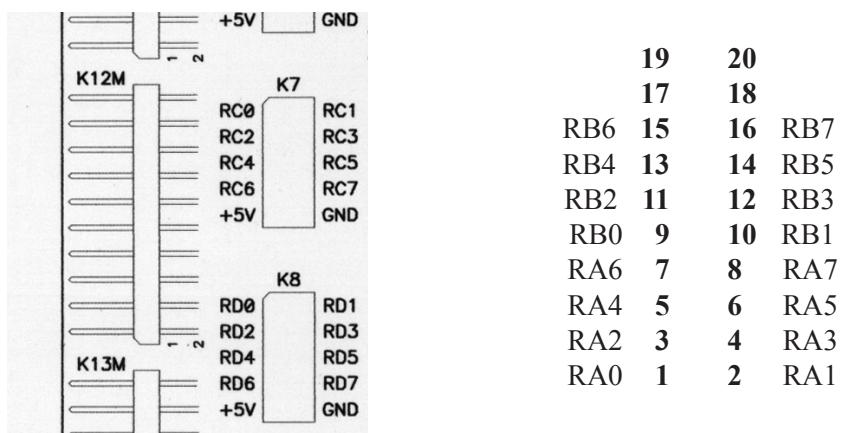
20	19
18	17
RB3	16 RB4
RB5	14 RB6
RB7	12 MCLR
VDD	10 VDD
GND	8 GND
+5V	6 +5V
15VDC	4 15VDC
VBAT	2 VBAT

## Connettore di uscita K11M

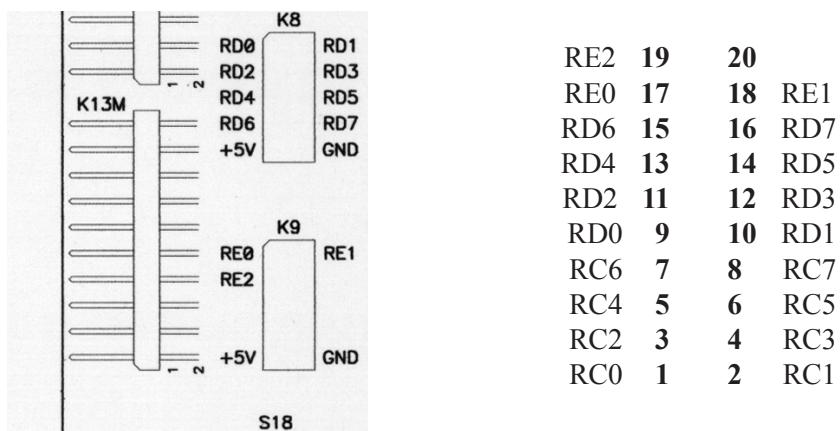


19	20
17	18
15	16
13	14
11	12
9	10
GND	8 GND
+5V	6 +5V
15VDC	4 15VDC
VBAT	2 VBAT

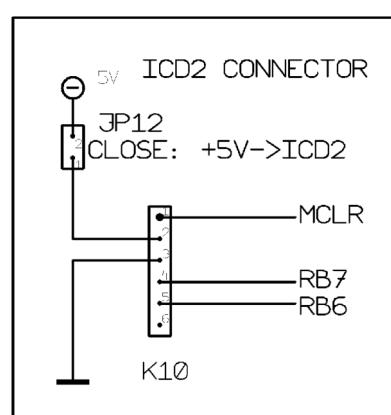
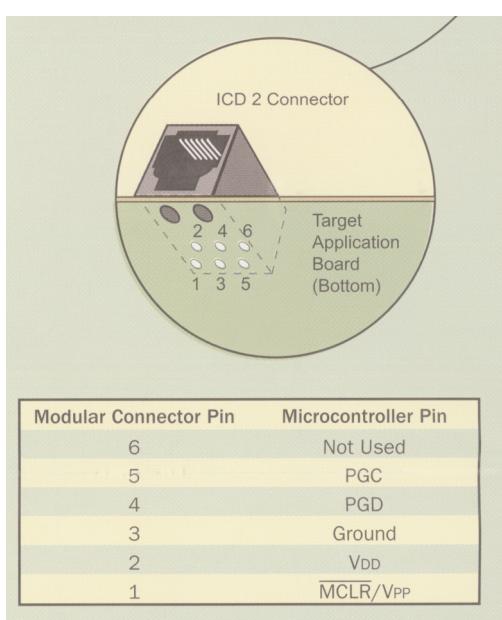
## Connettore di uscita K12M



## Connettore di uscita K13M



## Connettore per ICD2



## Utilizzo di ICD2

La scheda Socket C-203 può essere gestita dall'In-Circuit Debugger / Programmer ICD2 di Microchip.

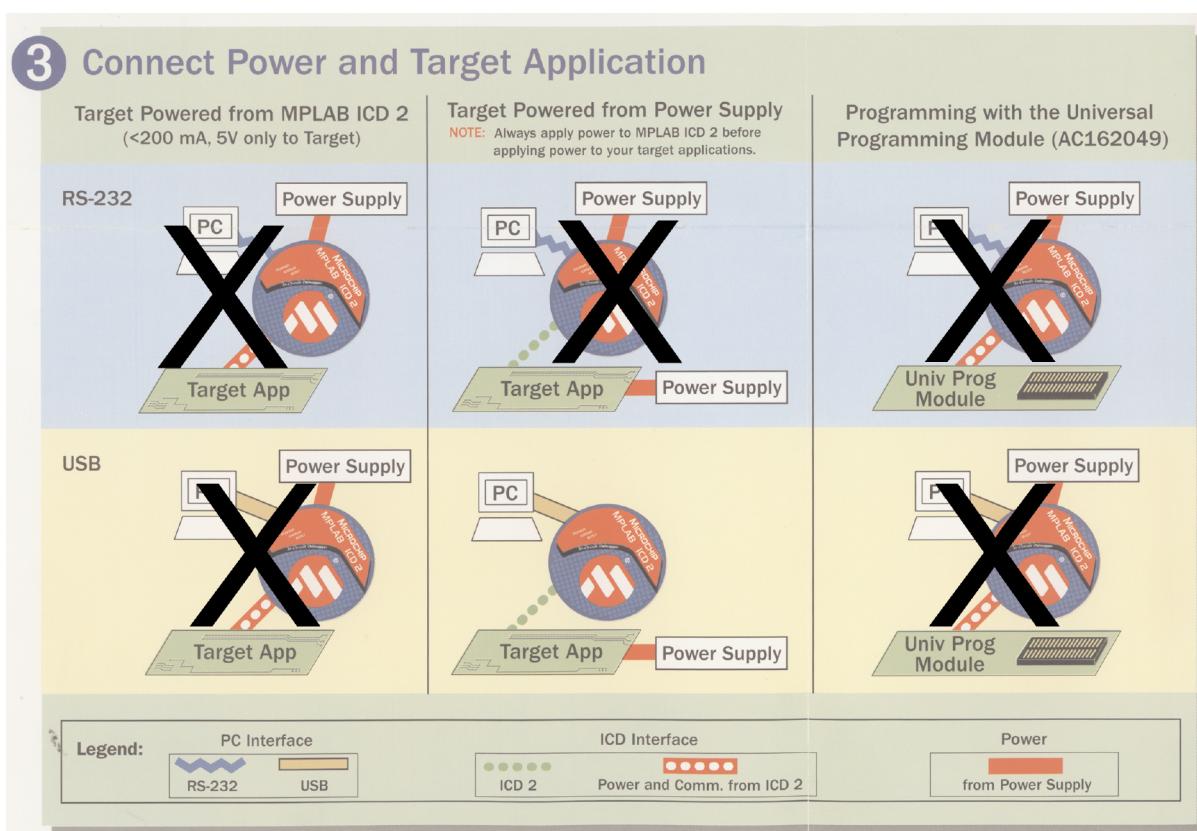
Prima di utilizzare ICD2 eseguire le seguenti configurazioni:

- **Scollegare il cavo che collega la porta parallela del PC alla scheda programmazione C-202** e verificare che il led rosso (VPP) sia spento. In questo modo la tensione di programmazione (gestita da ICD2) non andrà mai in conflitto con la VPP che potrebbe essere generata inavvertitamente dalla scheda di programmazione C-202.
- **Aprire il jumper JP7.** Il segnale PGD (Program Data) risulta scollegato dalla scheda di programmazione.
- **Aprire il jumper JP8.** Il segnale PGC (Program Clock) risulta scollegato dalla scheda di programmazione.
- **Lasciare sempre** (sia per la programmazione che per il debug) **il deviatore S18** (scheda Socket C-203) **su Run.** In questo modo i piedini VCC degli zoccoli risultano sempre collegati al +5V.

Il nostro sistema è stato studiato per interfacciarsi ad ICD2 in **comunicazione USB** e con il **Target alimentato**; il sistema va quindi organizzato nel seguente modo:

- **ICD2 collegato al PC tramite il cavo USB in dotazione;**
- **ICD2 non alimentato;**
- **Scheda programmazione C-202 alimentata;**

- **Jumper JP12 scheda Socket C-203 chiuso.** Questo consente di portare ai driver di ICD2 la tensione di +5V. Per l'installazione di MPLAB IDE, la configurazione di MPLAB IDE con ICD2 e l'installazione dei driver USB attenersi alle istruzioni riportate nel manuale di ICD2.



**Attenzione:**  
**quando si utilizza il sistema ICD2 è indispensabile scollegare il cavo parallelo.**

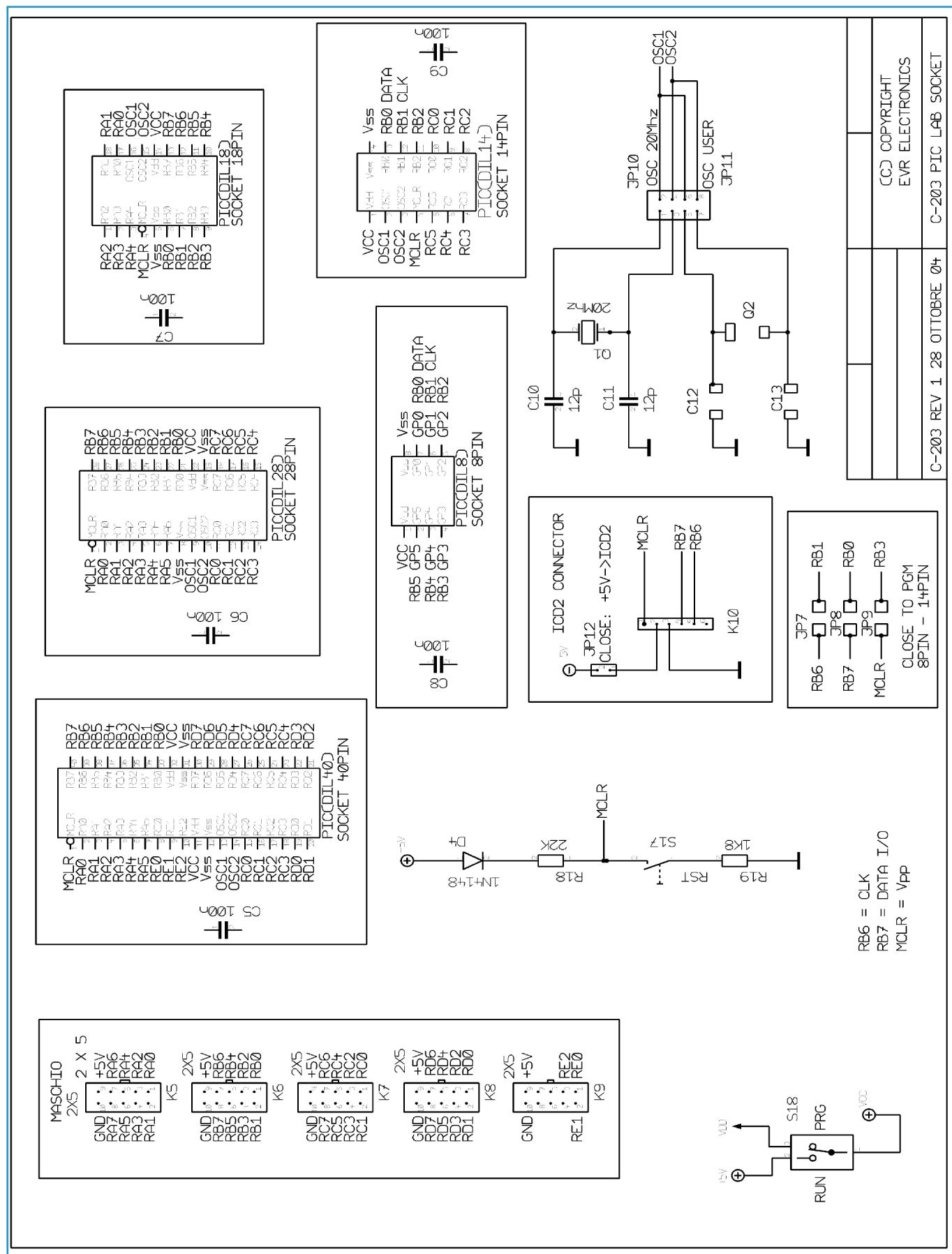
## Esempio di utilizzo di ICD2

Per meglio comprendere l'utilizzo di ICD2 abbiamo previsto un file progetto denominato T18F4320.MCP disponibile nella cartella ICD2. Questo programma di test attiva sequenzialmente ad intervalli di 2 secondi le linee della MCU. Vediamo ora come programmare un micro PIC18F4320-I/P partendo dal file progetto disponibile ed utilizzando ICD2.

**Queste operazioni vanno eseguite dopo l'installazione dell'ambiente MPLAB IDE e di ICD2 in configurazione USB e con alimentazione prelevata dal Target (vedi manuale di ICD2) e dopo aver configurato le schede C-202 e C-203 attenendosi alle informazioni riportate nella sezione Utilizzo di ICD2.**

- Avviare MPLAB IDE;
- Selezionare Project, di seguito Open e scegliere il file T18F4320.MCP;
- Nella finestra di progetto selezionare Source Files, premere il tasto destro del mouse, selezionare Add Files, selezionare T18F4320.ASM (questo file è disponibile nella cartella ICD2);
- Selezionare Configure, selezionare Select Device, scegliere PIC18F4320;
- Selezionare Configure, selezionare Configuration Bits, ed impostare i seguenti parametri: Oscillator = HS; Watchdog Timer = Disable; Low Voltage Program = Disable; tutti gli altri parametri possono essere lasciati nella condizione di default.
- Selezionare Project, selezionare Build All: il file T18F4320.ASM viene compilato;
- Verificare che la scheda programmazione C-202 e la scheda Socket C-203 siano configurate come descritto nella sezione Utilizzo di ICD2;
- Verificare che il cavo parallelo sia scollegato;**
  - Inserire una MCU PIC18F4320 nello zoccolo a 40 pins;
  - Chiudere il jumper JP2 della scheda di programmazione C-202, in questo modo il piedino PGM del microcontrollore viene portato a massa tramite una resistenza da 10K;
  - Alimentare la scheda di programmazione C-202 con una tensione continua e stabilizzata di 17Vdc;
  - Inserire il plug di ICD2 nel connettore K10 disponibile sulla scheda Socket C-203;
  - Selezionare Programmer, selezionare Connect;
  - Selezionare Programmer, selezionare Program; il file T18F4320 viene trasferito nella memoria programma del microcontrollore, la finestra di Output di MPLAB riporta l'esito di ogni operazione;
  - Aprire il jumper JP2 della scheda di programmazione C-202;
  - Togliere il plug di ICD2 dal connettore K10 disponibile sulla scheda Socket C-203;
  - Verificare il funzionamento del programma inserito nella MCU.

## Schema elettrico scheda Socket C-203



**PART LIST C-203**

**R18:** 22 Kohm

**R19:** 1,8 Kohm

**D4:** 1N4148

**C5:** 100 nF

**C6:** 100 nF

**C7:** 100 nF

**C8:** 100 nF

**C9:** 100 nF

**C10:** 12 pF

**C11:** 12 pF

**C12:** not mounted

**C13:** not mounted

**Q1:** 20 MHz

**Q2:** not mounted

**S17:** pushbutton cs

**S18:** shunter cs

**DIL8:** socket 8 pins

**DIL14:** socket 14 pins

**DIL18:** socket 18 pins

**DIL28:** socket 28 pins

**DIL40:** socket 40 pins

**K4F:** pin-strip F 90° 10+10 pins

**K5:** pin-strip M 5+5 pins

**K6:** pin-strip M 5+5 pins

**K7:** pin-strip M 5+5 pins

**K8:** pin-strip M 5+5 pins

**K9:** pin-strip M 5+5 pins

**K10:** connector FCC 6/6

**K11M:** pin-strip M 90° 10+10 pins

**K12M:** pin-strip M 90° 10+10 pins

**K13M:** pin-strip M 90° 10+10 pins

**JP7, JP8, JP9:** pin-strip M 3+3 pins

**JP10:** pin-strip M 2+2 pins

**JP11:** pin-strip M 2+2 pins

**JP12:** pin-strip M 2 pins

- CS C203-2

# IL SOFTWARE DI PROGRAMMAZIONE

## Introduzione

IC-Prog è un software estremamente versatile e semplice da utilizzare ed è in grado di programmare la maggior parte dei microcontrollori PIC di Microchip. Per informazioni approfondite sulle prestazioni di IC-Prog rimandiamo al file di Help allegato al programma. IC-Prog richiede un computer IBM compatibile e funziona in ambiente Windows 95, 98, ME, NT, 2000, XP.

## Installazione

Il programma è disponibile sotto forma di cartella compressa; sono presenti due file: icprog105.zip e icprog\_driver.zip.

Per installare il programma seguire i seguenti punti:

- Creare nel proprio PC, nella directory Programmi, la cartella IC-Prog.
- Decomprimere il file icprog105.zip e copiarne il contenuto nella cartella IC-Prog del proprio PC.
- Decomprimere il file icprog\_driver e copiarne il contenuto nella cartella IC-prog del proprio PC.

Selezionare il file icprog.exe nella cartella IC-Prog con il tasto destro del mouse; scegliere ora il comando

Proprietà e di seguito Compatibilità; con il mouse selezionare “Esegui il programma in modalità compatibilità per” e scegliere uno delle possibili modalità (Windows 95, Windows 98 / Windows ME, Windows NT, Windows 2000) in funzione del proprio sistema operativo.

## Avvio

Per avviare il programma fare doppio click sul file applicazione icprog.exe caratterizzato da un’icona che simboleggia un circuito integrato; apparirà la videata principale. L’area principale di IC-Prog mostra le informazioni di cui si ha bisogno per il dispositivo selezionato, nel nostro caso selezionando un dispositivo PIC qualsiasi l’area principale può sempre essere scomposta in quattro parti principali:

- la barra superiore dei comandi;
- l’area programma (Program Code) dove viene visualizzato il codice del programma che intendiamo trasferire nel microcontrollore o che abbiamo letto dal microcontrollore. Le informazioni sono visualizzate su più linee, in funzione della capacità di memoria programma del dispositivo selezionato. Ogni linea riporta nella prima colonna a sinistra l’indirizzo fisico della prima locazione di memoria indicata nella linea; di seguito troviamo il contenuto di 8 locazioni di memoria espresso in esadecimale; nell’ultima colonna troviamo il contenuto delle stesse 8 locazioni di memoria espresso in caratteri ASCII. Il contenuto di ogni locazione di memoria è rappresentato da una word composta da 16 bit; IC-Prog utilizza cifre esadecimali da 0000 a FFFF per rappresentare 16 bit. Ovviamente non tutti i dispositivi utilizzano word a 16 bit, ma ad esempio a 14, 12 e 8 bit per word. In tale condizione IC-Prog userà ugualmente una notazione esadecimale a 4 cifre, indicando rispettivamente nell’area di codice 3FFF, 0FFF, 0OFF per 14, 12, 8 bit.
- l’area dati (Eeprom Data) dove viene visualizzato il contenuto della memoria dati EEPROM del microcontrollore. Anche in questo caso le informazioni sono visualizzate su più linee, in funzione della capacità di memoria EEPROM del dispositivo. Ogni linea riporta nella prima colonna a sinistra l’indirizzo fisico della prima locazione di memoria indicata nella linea; di seguito troviamo il contenuto di 8 locazioni di memoria espresso in esadecimale; nell’ultima colonna troviamo il contenuto delle stesse 8 locazioni di memoria espresso in caratteri ASCII. Il contenuto di ogni locazione di memoria è rappresentato da un byte espresso in esadecimale, quindi con 2 cifre che vanno da 00 a FF.
- l’area di configurazione del dispositivo (Configuration) che varia in funzione del microcontrollore selezionato.

## Configurazione

Poichè IC-Prog supporta diverse soluzioni hardware è necessario configurarlo per la scheda di programmazione C-202.

A tale scopo dalla linea superiore dei comandi selezionare Setting e di seguito Hardware: comparirà la finestra di settaggio dell'hardware.

Selezionare come programmatore il ProPic 2 (Programmer = ProPic 2 Programmer).

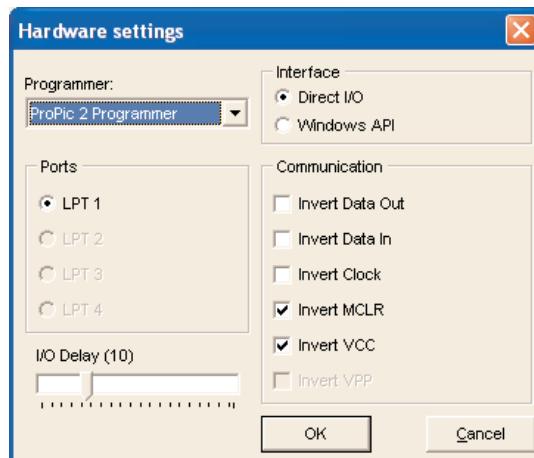
Selezionare la porta parallela a cui intendiamo collegare la scheda C-200 tra LPT1, LPT2, LPT3 e LPT4.

Impostare il ritardo nella commutazione tra ingresso e uscita a 10: I/O Delay (10); questo valore dovrebbe essere adatto per tutti i computer anche per quelli di ultima generazione, se in fase di programmazione si riscontrano dei problemi aumentare il valore del ritardo.

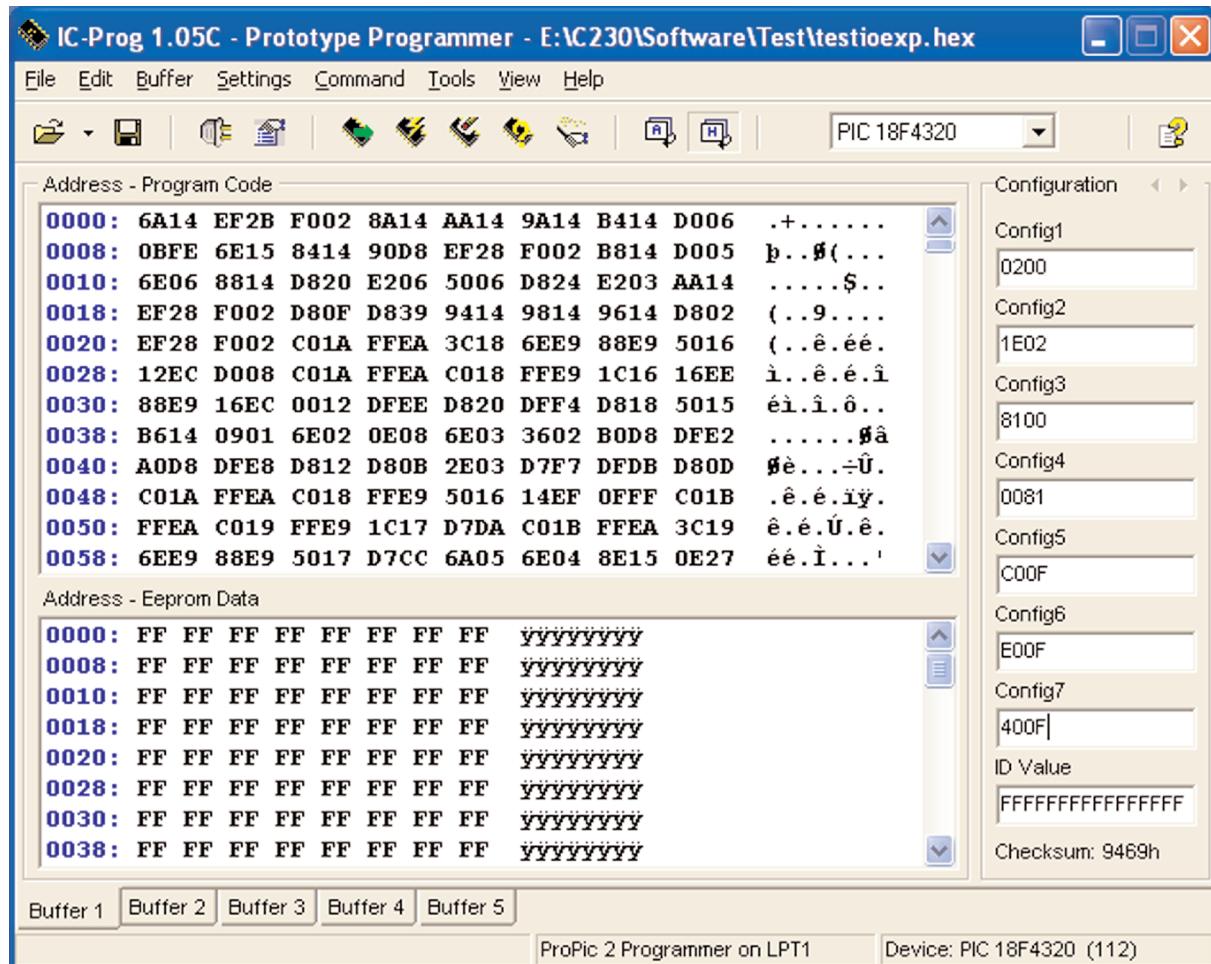
Selezionare l'interfaccia di tipo diretta: Interface = Direct I/O.

Impostare i parametri di comunicazione selezionando Invert MCLR e Invert Vcc.

Confermare i settaggi portando il mouse su OK: tutti i parametri impostati vengono salvati da IC-Prog anche per le sessioni future, non sarà quindi necessario impostarli nuovamente.



## Videata principale di IC-Prog



# PROGRAMMARE CON IC-PROG

## Operazioni preliminari

La scheda C-170 supporta le MCU PIC nei contenitori PDIP a 40, 28, 18, 14 e 8 pins.  
Qualora si utilizzino dispositivi in contenitori diversi da quelli elencati è necessario procurarsi  
degli opportuni adattatori.

Collegare la scheda di programmazione C-202 alla porta parallela del Computer.  
Procurarsi un alimentatore in grado di fornire una **tensione stabilizzata di 17 Vdc**  
e una corrente di almeno 300 mA.

Rimuovere la sezione risorse (esempio C-230 o C-180) dal resto della scheda.

Assicurarsi che i ponticelli siano configurati nel modo seguente:

### Sezione programmazione C-202:

**JP1** - VDD->ICSP = OPEN

**JP2** - RB3->GND = dipende dalla MCU da programmare vedi pagine seguenti

**JP3** - RB4->GND = dipende dalla MCU da programmare vedi pagine seguenti

**JP4** - RB5->GND = dipende dalla MCU da programmare vedi pagine seguenti

**JP5** - LPT D4 = CLOSE

**JP6** - LPT D5 = CLOSE

**JP7** - PGD = CLOSE

**JP8** - PGC = CLOSE

### Sezione socket C-203:

**JP7** - RB6->RB1 = OPEN per MCU 40-28-18 pins / CLOSE per MCU 14-8 pins

**JP8** - RB7->RB0 = OPEN per MCU 40-28-18 pins / CLOSE per MCU 14-8 pins

**JP9** - MCLR->RB3 = OPEN per MCU 40-28-18 pins / CLOSE per MCU 14-8 pins

**JP10** - OSC 20MHz = CLOSE chiudere con due ponticelli in verticale

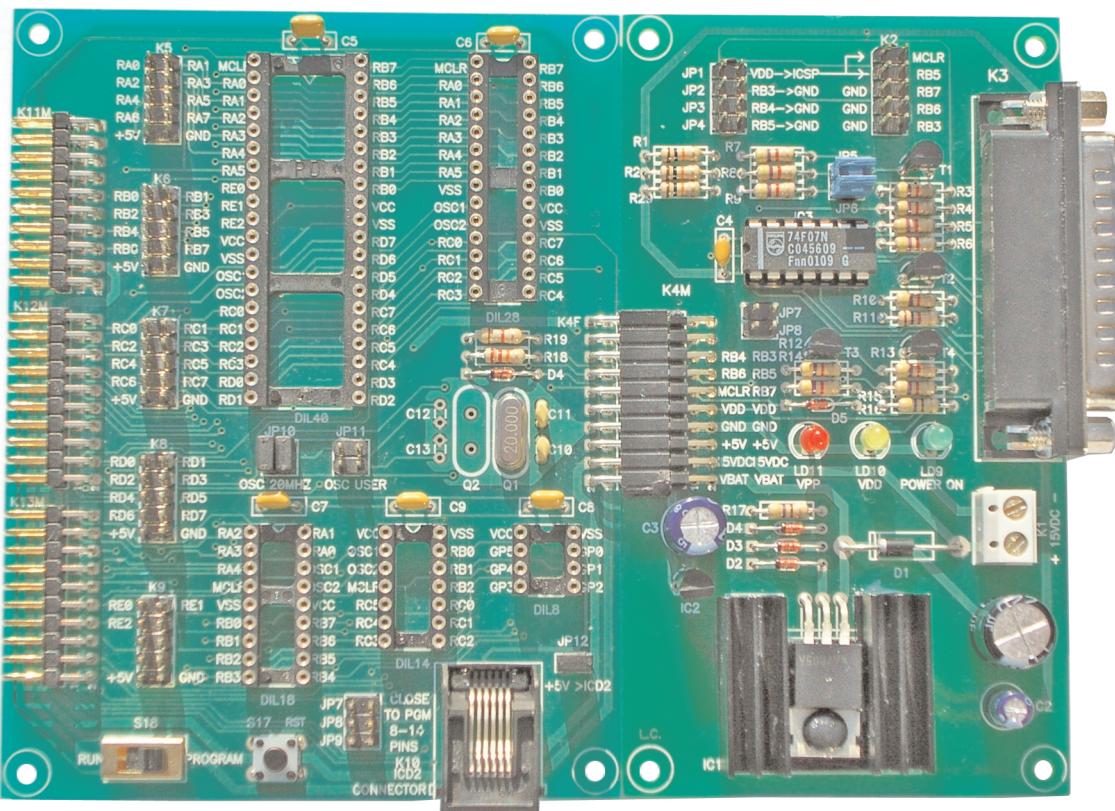
**JP11** - OSC USER = OPEN

**S18** = PROGRAM

Abbiamo previsto un programma di test per ogni piedinatura selezionando uno tra i micro più diffusi.

Il programma consiste nell'impostare tutte le linee di i/o del micro in uscita e nel far scorrere sulle uscite un 1 logico. I listati, le configurazioni dei jumper e del programma IC-Prog sono riportati nelle pagine seguenti.

## Note



## Microcontrollori a 40 pins (PIC18F4320)

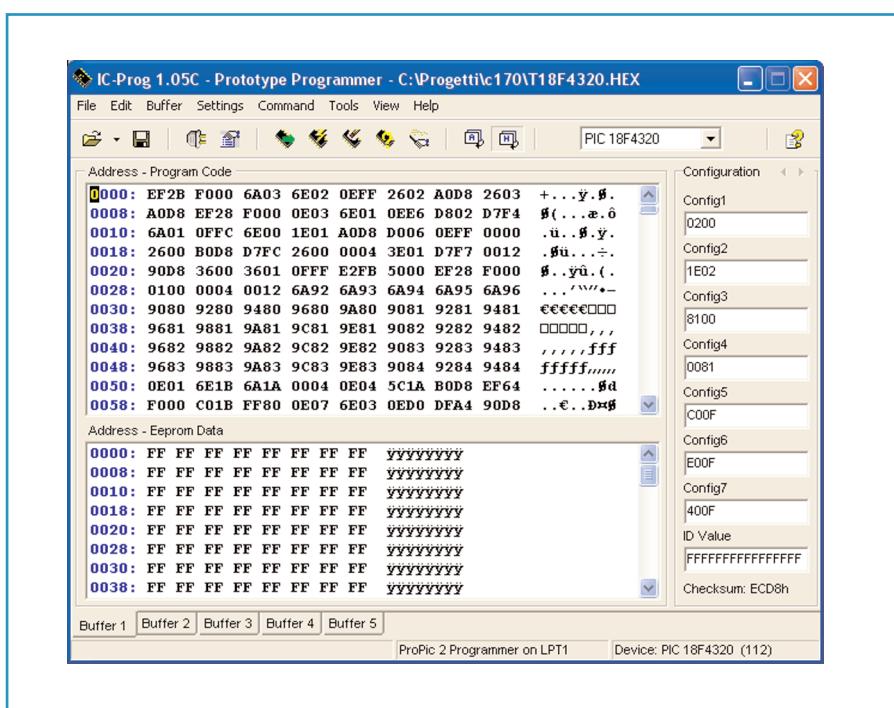
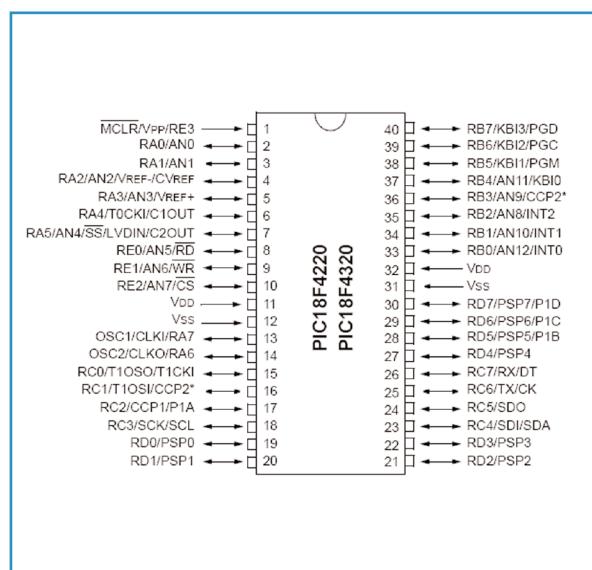
In questa sezione vediamo le fasi necessarie per programmare o leggere un microcontrollore PIC a 40 piedini. Le operazioni che ci accingiamo a descrivere sono valide per qualsiasi PIC a 40 piedini, per semplicità prendiamo in esame uno tra i microcontrollori più diffusi in questa fascia di contenitori e cioè il **PIC18F4320-I/P**. Abbiamo previsto un programma di test denominato **T18F4320** scritto in PicBasic Pro e disponibile nel CD sia in formato sorgente che eseguibile.

Vediamo ora le fasi necessarie per trasferire il file demo T18F4320.HEX nel microcontrollore.

- Avviare IC-Prog e selezionare il dispositivo **PIC18F4320**
- Selezionare il comando File in alto a sinistra
- Selezionare il comando Open File e di seguito il file **T18F4320.HEX**
- Impostare i bit di Config come indicato nella figura riportata in basso;
- Inserire un **PIC18F4320-I/P** nello zoccolo a 40 pins DIL40 rispettando la polarità
- Alimentare la scheda
- Verificare che i jumper **JP7** e **JP8** scheda C-202 siano chiusi
- Chiudere il jumper **JP4** della scheda di programmazione C-202 per portare il pin PGM del microcontrollore a massa
- Portare il selettor S18 in posizione **PROGRAM**
- Selezionare il comando **Program All**: comparirà la scritta "Do you really want to program the devices"; Selezionare **Yes**
- Inizia quindi la fase di programmazione che si articola in 3 parti ognuna contraddistinta da una barra di progressione: Programming Code; Programming Data; Programming Config
- Di seguito inizia la fase di verifica dei dati scritti anch'essa evidenziata da una barra di progressione.

- Se la verifica è andata a buon fine, compare la scritta Device Successfully Verified ed il microcontrollore è stato correttamente programmato
- Rimuovere i jumper **JP4**, **JP7**, **JP8** (scheda C-202) e portare il selettor S18 su **RUN**.
- Verificare con un oscilloscopio o con un tester che le linee del microcontrollore vadano sequenzialmente a livello logico alto.

Nota: la porta RA4 essendo di tipo open collector richiederebbe una resistenza esterna di pull-up e, per questo motivo, è stata esclusa dal test.



## T18F4320.BAS

```
*****
* File Name: T18F4320
* Notice : Copyright (c) 2005 EVR electronics
*           : All Rights Reserved
* Date   : 26/01/2005
*****
```

```
DEFINE OSC 20
```

```
TRISA = %00000000 'PORTA OUT
TRISB = %00000000 'PORTB OUT
TRISC = %00000000 'PORTC OUT
TRISD = %00000000 'PORTD OUT
TRISE = %00000000 'PORTE OUT
```

```
TEMP VAR BYTE
CONTA VAR BYTE
```

```
PORTA.0 = 0
PORTA.1 = 0
PORTA.2 = 0
PORTA.3 = 0
PORTA.5 = 0
```

```
PORTB.0 = 0
PORTB.1 = 0
PORTB.2 = 0
PORTB.3 = 0
PORTB.4 = 0
PORTB.5 = 0
PORTB.6 = 0
PORTB.7 = 0
```

```
PORTC.0 = 0
PORTC.1 = 0
PORTC.2 = 0
PORTC.3 = 0
PORTC.4 = 0
PORTC.5 = 0
PORTC.6 = 0
PORTC.7 = 0
```

```
PORTD.0 = 0
PORTD.1 = 0
PORTD.2 = 0
PORTD.3 = 0
PORTD.4 = 0
PORTD.5 = 0
PORTD.6 = 0
PORTD.7 = 0
```

```
PORTE.0 = 0
PORTE.1 = 0
PORTE.2 = 0
```

LOOP:

```
TEMP = 1
for CONTA = 0 TO 3
    PORTA = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
```

NEXT CONTa

```
PORTA.3 = 0
PORTA.5 = 1
PAUSE 2000
PORTA.5 = 0
PAUSE 2000
```

```
TEMP = 1
for CONTA = 0 TO 7
    PORTB = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
```

NEXT conta

```
PORTB.7 = 0
pause 2000
```

```
TEMP = 1
for CONTA = 0 TO 7
    PORTC = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
```

NEXT conta

```
PORTC.7 = 0
pause 2000
```

```
TEMP = 1
for CONTA = 0 TO 7
    PORTD = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
```

NEXT conta

```
PORTD.7 = 0
pause 2000
```

```
TEMP = 1
for CONTA = 0 TO 2
    PORTE = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
```

NEXT conta

```
PORTE.2 = 0
pause 2000
```

Goto LOOP

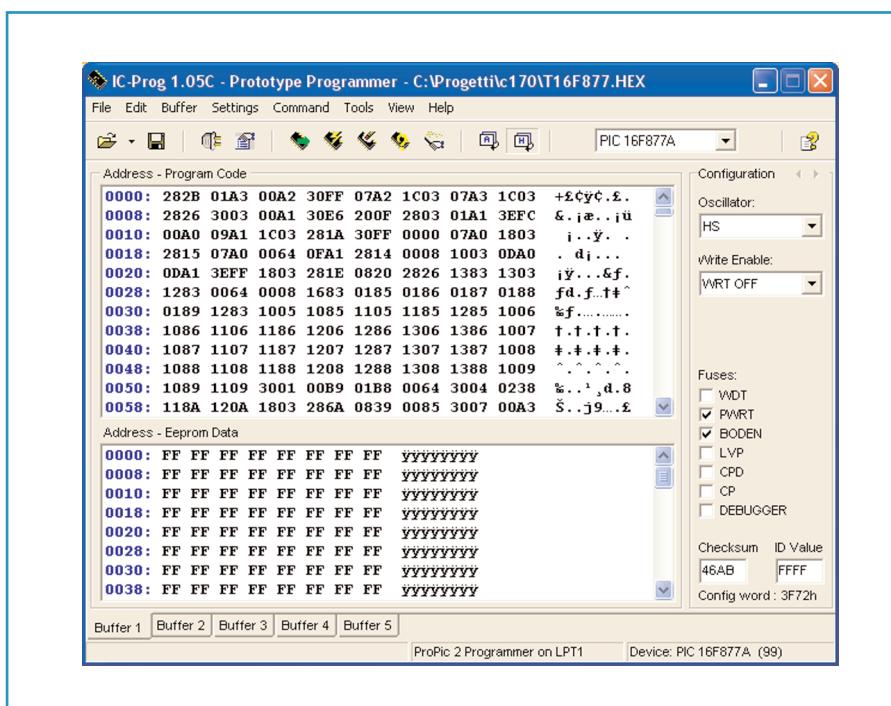
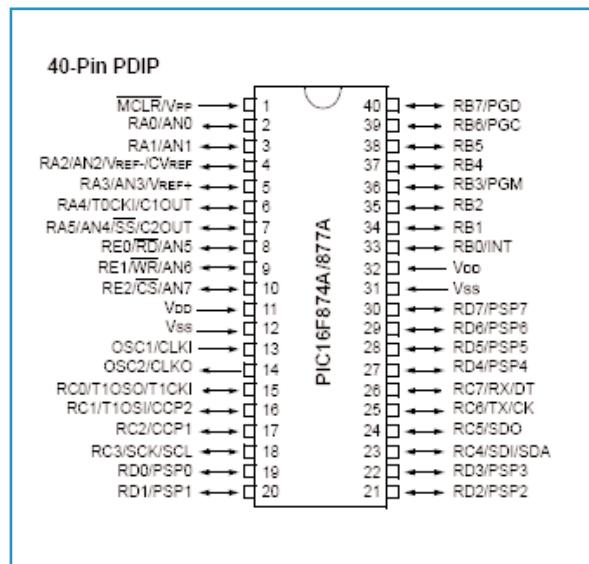
## Microcontrollori a 40 pins (PIC16F877A)

In questa sezione vediamo le fasi necessarie per programmare o leggere un microcontrollore PIC a 40 piedini. Le operazioni che ci accingiamo a descrivere sono valide per qualsiasi PIC a 40 piedini, per semplicità prendiamo in esame uno tra i microcontrollori più diffusi in questa fascia di contenitori e cioè il **PIC16F877A-I/P**. Abbiamo previsto un programma di test denominato **T16F877** scritto in Pic Basic e disponibile nel CD sia in formato sorgente che eseguibile.

Vediamo ora le fasi necessarie per trasferire il file demo T16F877.HEX nel microcontrollore.

- Avviare IC-Prog e selezionare il dispositivo **PIC16F877A**
- Selezionare il comando File in alto a sinistra
- Selezionare il comando Open File e di seguito il file **T16F877.HEX**
- Impostare oscillatore HS (**Oscillator = HS**)
- Disabilitare la scrittura in memoria (**Write Enable = WRT OFF**)
- Abilitare **PWRT** e **BODEN** e verificare che le altre opzioni del gruppo Fuses siano disabilitate
- Inserire un **PIC16F877A** nello zoccolo a 40 pins DIL40
- Alimentare la scheda
- Verificare che i jumper **JP7** e **JP8** scheda C-202 siano chiusi
- Chiudere il jumper **JP2** scheda programmazione C-202 per portare il pin PGM del microcontrollore a massa
- Portare il selettore **S18** in posizione **PROGRAM**
- Selezionare il comando **Program All**: comparirà la scritta "Do you really want to program the devices"; Selezionare **Yes**
- Inizia quindi la fase di programmazione che si articola in 3 parti ognuna contraddistinta da una barra di progressione: Programming Code; Programming Data; Programming Config
- Di seguito inizia la fase di verifica dei dati scritti anch'essa evidenziata da una barra di progressione.
- Se la verifica è andata a buon fine, compare la scritta Device Successfully Verified ed il microcontrollore è stato correttamente programmato
- Rimuovere i jumper **JP2**, **JP7**, **JP8** scheda programmazione e portare il selettore **S18** su **RUN**.
- Verificare con un oscilloscopio o con un tester che le linee del microcontrollore vadano sequenzialmente a livello logico alto.

Nota: la porta RA4 essendo di tipo open collector richiederebbe una resistenza esterna di pull-up e, per questo motivo, è stata esclusa dal test.



## T16F877.BAS

```
*****
* File Name: T16F877
* Notice : Copyright (c) 2005 EVR electronics
*           : All Rights Reserved
* Date  : 26/01/2005
*****
```

```
DEFINE OSC 20
```

```
TRISA = %00000000 'PORTA OUT
TRISB = %00000000 'PORTB OUT
TRISC = %00000000 'PORTC OUT
TRISD = %00000000 'PORTD OUT
TRISE = %00000000 'PORTE OUT
```

```
TEMP VAR BYTE
CONTA VAR BYTE
```

```
PORTA.0 = 0
PORTA.1 = 0
PORTA.2 = 0
PORTA.3 = 0
PORTA.5 = 0
```

```
PORTB.0 = 0
PORTB.1 = 0
PORTB.2 = 0
PORTB.3 = 0
PORTB.4 = 0
PORTB.5 = 0
PORTB.6 = 0
PORTB.7 = 0
```

```
PORTC.0 = 0
PORTC.1 = 0
PORTC.2 = 0
PORTC.3 = 0
PORTC.4 = 0
PORTC.5 = 0
PORTC.6 = 0
PORTC.7 = 0
```

```
PORTD.0 = 0
PORTD.1 = 0
PORTD.2 = 0
PORTD.3 = 0
PORTD.4 = 0
PORTD.5 = 0
PORTD.6 = 0
PORTD.7 = 0
```

```
PORTE.0 = 0
PORTE.1 = 0
PORTE.2 = 0
```

```
LOOP:
    TEMP = 1
    for CONTA = 0 TO 3
        PORTA = TEMP
        PAUSE 2000
        TEMP = TEMP << 1
    NEXT CONTA
```

```
    PORTA.3 = 0
    PORTA.5 = 1
    PAUSE 2000
    PORTA.5 = 0
    PAUSE 2000
```

```
    TEMP = 1
    for CONTA = 0 TO 7
        PORTB = TEMP
        PAUSE 2000
        TEMP = TEMP << 1
    NEXT conta
```

```
    PORTB.7 = 0
    pause 2000
```

```
    TEMP = 1
    for CONTA = 0 TO 7
        PORTC = TEMP
        PAUSE 2000
        TEMP = TEMP << 1
    NEXT conta
    PORTC.7 = 0
    pause 2000
```

```
    TEMP = 1
    for CONTA = 0 TO 7
        PORTD = TEMP
        PAUSE 2000
        TEMP = TEMP << 1
    NEXT conta
    PORTD.7 = 0
    pause 2000
```

```
    TEMP = 1
    for CONTA = 0 TO 2
        PORTE = TEMP
        PAUSE 2000
        TEMP = TEMP << 1
    NEXT conta
    PORTE.2 = 0
    pause 2000
    Goto LOOP
```

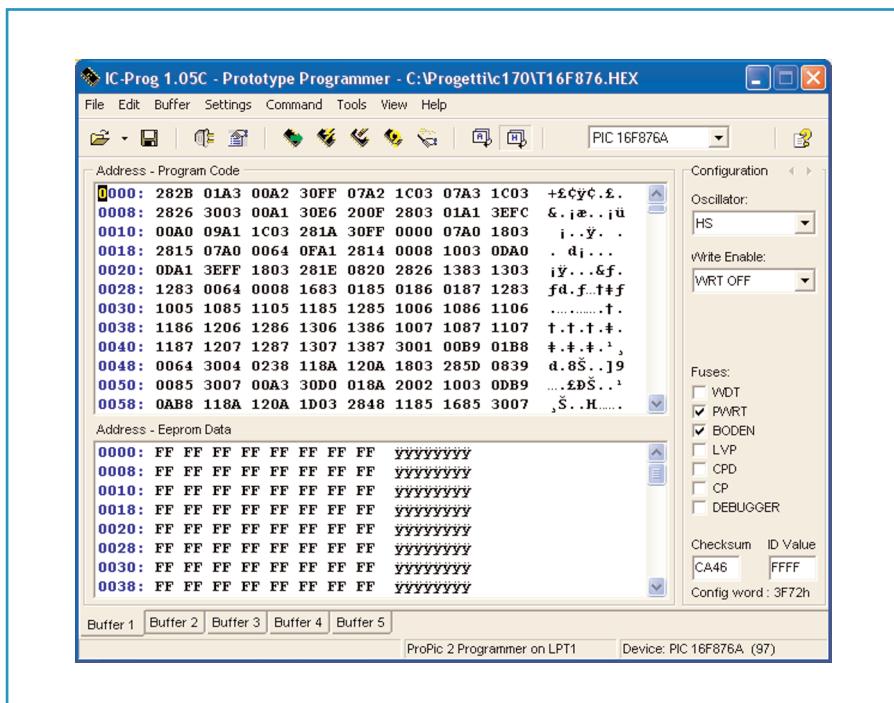
## Microcontrollori a 28 pins

In questa sezione vediamo le fasi necessarie per programmare o leggere un microcontrollore PIC a 28 piedini. Le operazioni che ci accingiamo a descrivere sono valide per qualsiasi PIC a 28 piedini, per semplicità prendiamo in esame uno tra i microcontrollori più diffusi in questa fascia di contenitori e cioè il **PIC16F876A-I/SP**. Abbiamo previsto un programma di test denominato **T16F876** scritto in Pic Basic e disponibile nel CD sia in formato sorgente che eseguibile.

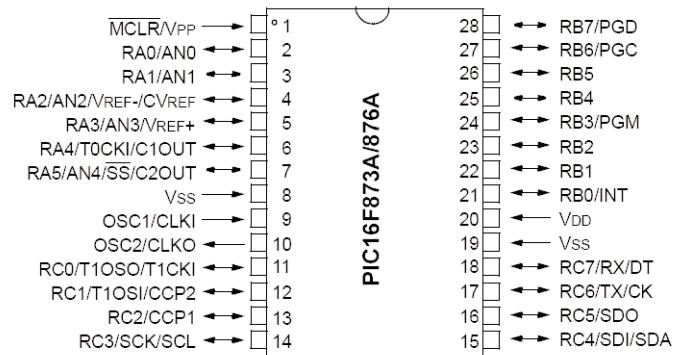
Vediamo ora le fasi necessarie per trasferire il file demo T16F876.HEX nel microcontrollore.

- Avviare IC-Prog e selezionare il dispositivo **PIC16F876A**
- Selezionare il comando File in alto a sinistra
- Selezionare il comando Open File e di seguito il file **T16F876.HEX**
- Impostare oscillatore HS (**Oscillator = HS**)
- Disabilitare la scrittura in memoria (**Write Enable = WRT OFF**)
- Abilitare **PWRT** e **BODEN** e verificare che le altre opzioni del gruppo Fuses siano disabilitate
- Inserire un **PIC16F876A** nello zoccolo a 28 pins DIL28 rispettando la polarità
- Alimentare la scheda
- Verificare che i jumper **JP7** e **JP8** scheda C-202 siano chiusi
- Chiudere il jumper **JP2** scheda programmazione C-202 per portare il pin PGM del microcontrollore a massa
- Portare il selettore **S18** in posizione **PROGRAM**
- Selezionare il comando **Program All**: comparirà la scritta "Do you really want to program the devices"; Selezionare **Yes**
- Inizia quindi la fase di programmazione che si articola in 3 parti ognuna contraddistinta da una barra di progressione: Programming Code; Programming Data; Programming Config
- Di seguito inizia la fase di verifica dei dati scritti anch'essa evidenziata da una barra di progressione
  
- Se la verifica è andata a buon fine, compare la scritta Device Successfully Verified ed il microcontrollore è stato correttamente programmato
- Rimuovere i jumper **JP2**, **JP7**, **JP8** scheda programmazione C-202 e portare il selettore S18 su **RUN**
- Verificare con un oscilloscopio o con un tester che le linee del microcontrollore vadano sequenzialmente a livello logico alto

Nota: la porta RA4 essendo di tipo open collector richiederebbe una resistenza esterna di pull-up e, per questo motivo, è stata esclusa dal test.



### 28-Pin PDIP, SOIC, SSOP



### T16F876.BAS

```
*****
* File Name: T16F876
* Notice : Copyright (c) 2005 EVR electronics
*          : All Rights Reserved
* Date   : 26/01/2005
*****
```

PORTC.6 = 0  
PORTC.7 = 0

**LOOP:**

TEMP = 1  
for CONTA = 0 TO 3  
PORTA = TEMP  
PAUSE 2000  
TEMP = TEMP << 1  
NEXT CONTA  
PORTA.3 = 0  
PORTA.5 = 1  
PAUSE 2000  
PORTA.5 = 0  
PAUSE 2000

TEMP = 1  
for CONTA = 0 TO 7  
PORTB = TEMP  
PAUSE 2000  
TEMP = TEMP << 1  
NEXT conta  
PORTB.7 = 0  
pause 2000

TEMP = 1  
for CONTA = 0 TO 7  
PORTC = TEMP  
PAUSE 2000  
TEMP = TEMP << 1  
NEXT conta  
PORTC.7 = 0  
pause 2000

Goto LOOP

DEFINE OSC 20

TRISA = %00000000 'PORTA OUT  
TRISB = %00000000 'PORTB OUT  
TRISC = %00000000 'PORTC OUT

TEMP VAR BYTE  
CONTA VAR BYTE

PORTA.0 = 0  
PORTA.1 = 0  
PORTA.2 = 0  
PORTA.3 = 0  
PORTA.5 = 0

PORTB.0 = 0  
PORTB.1 = 0  
PORTB.2 = 0  
PORTB.3 = 0  
PORTB.4 = 0  
PORTB.5 = 0  
PORTB.6 = 0  
PORTB.7 = 0

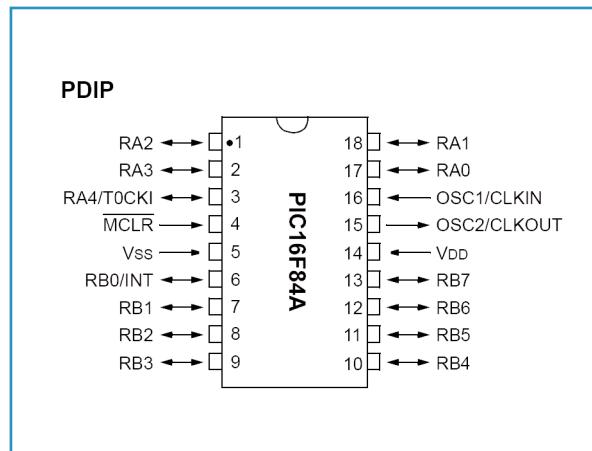
PORTC.0 = 0  
PORTC.1 = 0  
PORTC.2 = 0  
PORTC.3 = 0  
PORTC.4 = 0  
PORTC.5 = 0

## Microcontrollori a 18 pins

In questa sezione vediamo le fasi necessarie per programmare o leggere un microcontrollore PIC a 18 piedini. Le operazioni che ci accingiamo a descrivere sono valide per qualsiasi PIC a 18 piedini, per semplicità prendiamo in esame uno tra i microcontrollori più diffusi in questa fascia di contenitori e cioè il **PIC16F84A-20I/P**. Abbiamo previsto un programma di test denominato **T16F84** scritto in Basic e disponibile nel CD sia in formato sorgente che eseguibile.

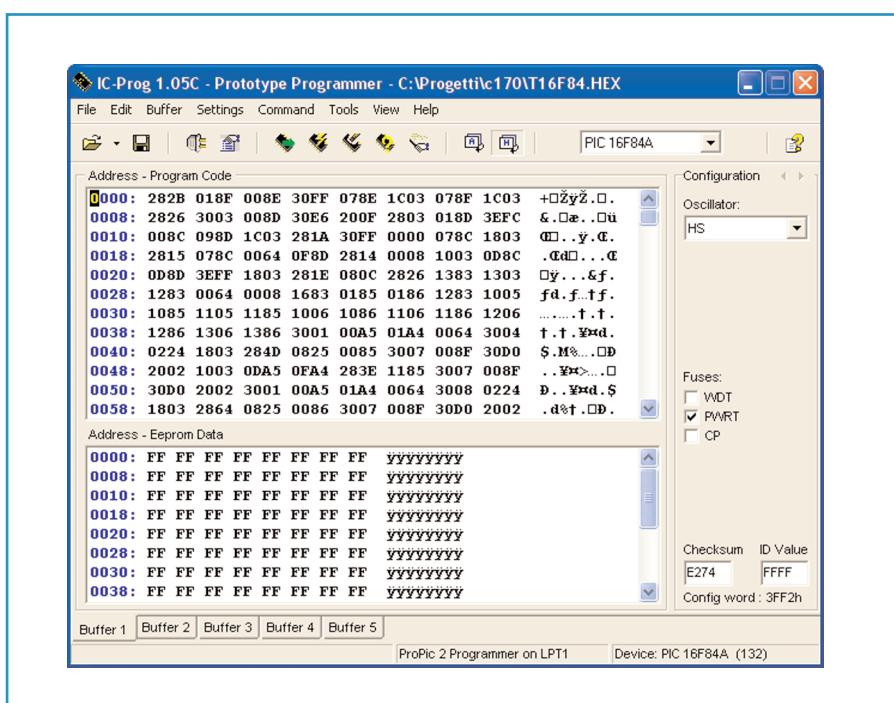
Vediamo ora le fasi necessarie per trasferire il file demo T16F84.HEX nel microcontrollore.

- Avviare IC-Prog e selezionare il dispositivo **PIC16F84A**
- Selezionare il comando File in alto a sinistra
- Selezionare il comando Open File e di seguito il file **T16F84.HEX**
- Impostare oscillatore HS (**Oscillator = HS**)
- Abilitare **PWRT** e verificare che le altre opzioni del gruppo Fuses siano disabilitate
- Inserire un **PIC16F84A** nello zoccolo a 18 pins DIL18 rispettando la polarità
- Alimentare la scheda
- Verificare che i jumper **JP7** e **JP8** scheda C-202 siano chiusi
- Portare il selettore **S18** in posizione **PROGRAM**
- Selezionare il comando **Program All**: comparirà la scritta "Do you really want to program the devices";  
Selezionare **Yes**
- Inizia quindi la fase di programmazione che si articola in 3 parti ognuna contraddistinta da una barra di progressione: Programming Code; Programming Data; Programming Config
- Di seguito inizia la fase di verifica dei dati scritti anch'essa evidenziata da una barra di progressione
- Se la verifica è andata a buon fine, compare la scritta Device Successfully Verified ed il microcontrollore è stato correttamente programmato
- rimuovere i jumper **JP7** e **JP8** scheda C-202



- Portare il selettore **S18** su **RUN**
- Verificare con un oscilloscopio o con un tester che le linee del microcontrollore vadano sequenzialmente a livello logico alto

Nota: la porta RA4 essendo di tipo open collector richiederebbe una resistenza esterna di pull-up e, per questo motivo, è stata esclusa dal test.



## T16F84.BAS

```
*****
* File Name: T16F84
* Notice : Copyright (c) 2005 EVR electronics
*          : All Rights Reserved
* Date   : 26/01/2005
*****
```

```
DEFINE OSC 20
```

```
TRISA = %00000000 'PORTA OUT
TRISB = %00000000 'PORTB OUT
```

```
TEMP VAR BYTE
CONTA VAR BYTE
```

```
PORATA.0 = 0
PORATA.1 = 0
PORATA.2 = 0
PORATA.3 = 0
```

```
PORTB.0 = 0
PORTB.1 = 0
PORTB.2 = 0
PORTB.3 = 0
PORTB.4 = 0
PORTB.5 = 0
PORTB.6 = 0
PORTB.7 = 0
```

LOOP:

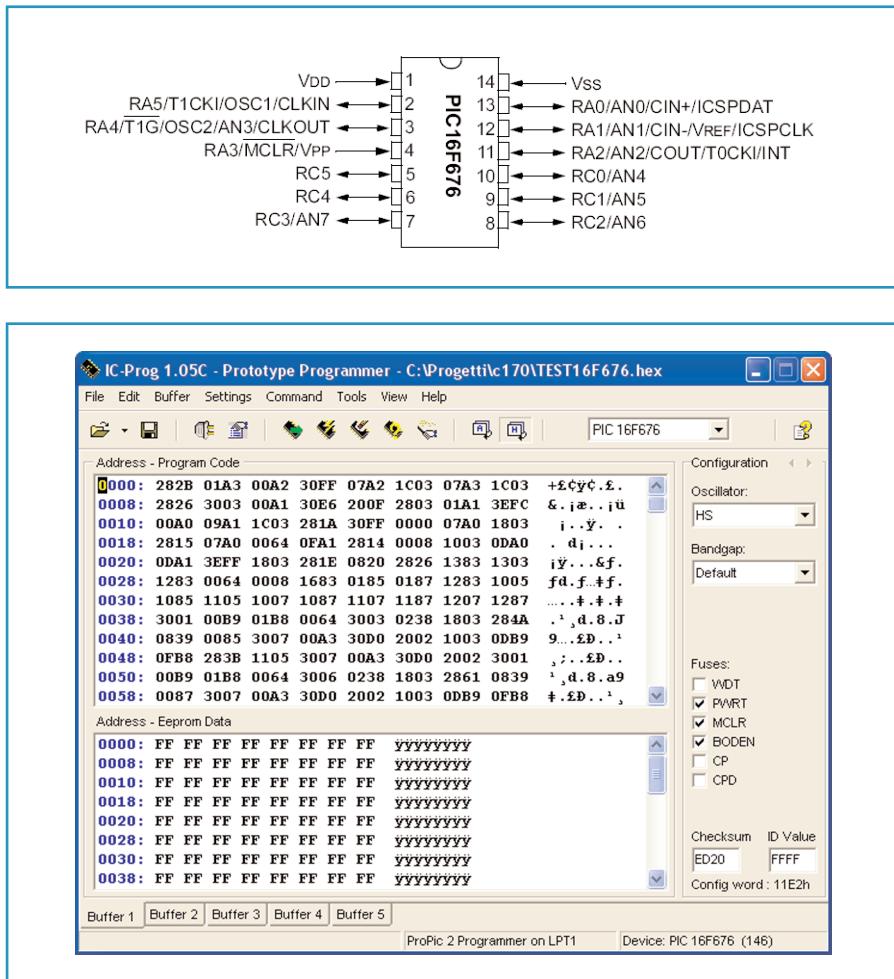
```
TEMP = 1
for CONTA = 0 TO 3
    PORATA = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
NEXT CONTA
PORATA.3 = 0
pause 2000
TEMP = 1
for CONTA = 0 TO 7
    PORTB = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
NEXT conta
PORTB.7 = 0
pause 2000
Goto LOOP
```

## Microcontrollori a 14 pins

In questa sezione vediamo le fasi necessarie per programmare o leggere un microcontrollore PIC a 14 piedini. Le operazioni che ci accingiamo a descrivere sono valide per qualsiasi PIC a 14 piedini, per semplicità prendiamo in esame uno tra i microcontrollori più diffusi in questa fascia di contenitori e cioè il **PIC16F676-I/P**. Abbiamo previsto un programma di test denominato **T16F676** scritto in Pic Basic e disponibile nel CD sia in formato sorgente che eseguibile.

Vediamo ora le fasi necessarie per trasferire il file demo T6F676.HEX nel microcontrollore.

- Avviare IC-Prog e selezionare il dispositivo **PIC16F676**
- Selezionare il comando File in alto a sinistra
- Selezionare il comando Open File e di seguito il file **T16F676.HEX**
- Impostare oscillatore HS (**Oscillator = HS**)
- Impostare Bandgap = Default
- Abilitare **MCLR, BODEN, PWRT** e verificare che le altre opzioni del gruppo Fuses siano disabilitate
- Inserire un **PIC16F676** nello zoccolo a 14 pins DIL14 rispettando la polarità
- Alimentare la scheda
- Verificare che i jumper **JP7** e **JP8** scheda programmazione C-202 siano chiusi
- Chiudere i jumper **JP7, JP8 e JP9** scheda socket C-203
- Portare il selettore **S18** in posizione **PROGRAM**
- Selezionare il comando Program All: comparirà la scritta "Do you really want to program the devices"; Selezionare **Yes**
- Comparirà la scritta "Oscillator calibration value = xxxx.Do you want to use value from file instead?" Rispondere **No** per mantenere il valore di calibrazione memorizzato dal costruttore nell'ultima locazione della memoria programma.
- Inizia quindi la fase di programmazione che si articola in 3 parti ognuna contraddistinta da una barra di progressione: Programming Code; Programming Data; Programming Config
- Di seguito inizia la fase di verifica dei dati scritti anch'essa evidenziata da una barra di progressione
- Se la verifica è andata a buon fine, compare la scritta Device Successfully Verified ed il microcontrollore è stato correttamente programmato
- Rimuovere i jumper **JP7, JP8 e JP9** scheda socket
- Rimuovere i jumper **JP7, JP8** scheda C-202
- Portare il selettore **S18** su **RUN**.
- Verificare con un oscilloscopio o con un tester che le linee del microcontrollore vadano sequenzialmente a livello logico 1



**Nota:**

I piedini 13, 12, 11 sono collegati rispettivamente alle linee RB0, RB1 e RB2.  
 Il piedino 4 è utilizzato come Master Clear (MCLR),  
 mentre i piedini 2 e 3 sono utilizzati come OSC1 e OSC2.

```
*****
* Name   : T16F676.BAS
* Notice : Copyright (c) 2005 EVR electronics
*          : All Rights Reserved
* Date   : 26/01/2005
*****
```

DEFINE OSC 20

```
TRISA = %00000000 'PORTA OUT
TRISC = %00000000 'PORTC OUT
```

```
TEMP VAR BYTE
CONTA VAR BYTE
```

```
PORATA.0 = 0
PORATA.1 = 0
PORATA.2 = 0
```

```
PORTC.0 = 0
PORTC.1 = 0
PORTC.2 = 0
PORTC.3 = 0
PORTC.4 = 0
PORTC.5 = 0
```

### T16F676.BAS

LOOP:

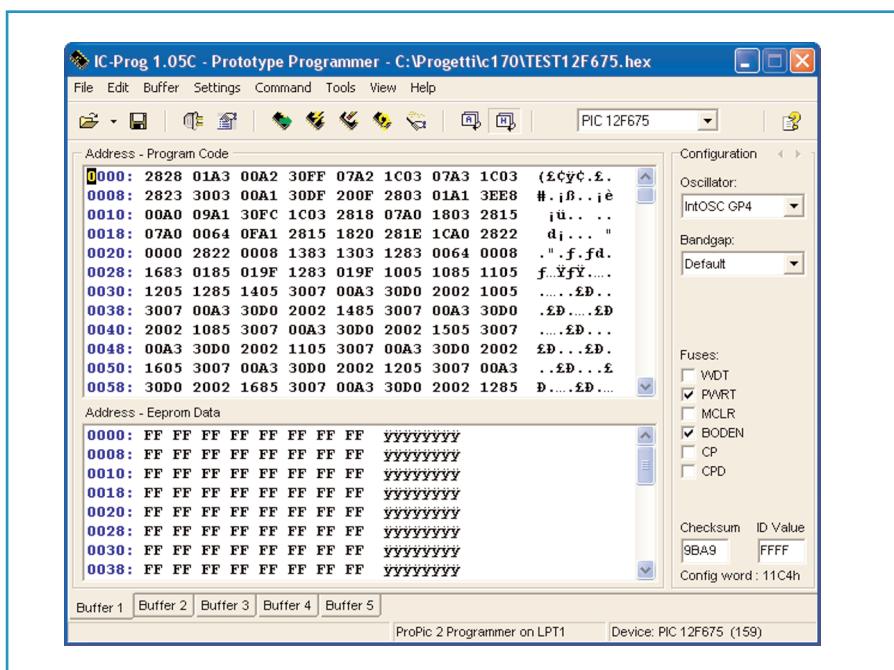
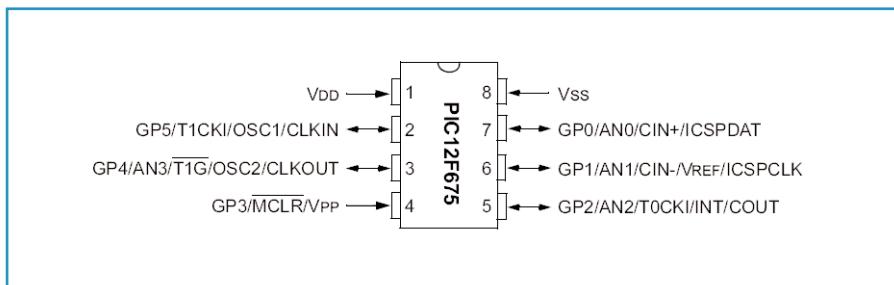
```
TEMP = 1
for CONTA = 0 TO 2
    PORTA = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
NEXT CONTA
PORTA.2 = 0
pause 2000
TEMP = 1
for CONTA = 0 TO 5
    PORTC = TEMP
    PAUSE 2000
    TEMP = TEMP << 1
NEXT conta
PORTC.5 = 0
pause 2000
Goto LOOP
```

## Microcontrollori a 8 pins

In questa sezione vediamo le fasi necessarie per programmare o leggere un microcontrollore PIC a 8 piedini. Le operazioni che ci accingiamo a descrivere sono valide per qualsiasi PIC a 8 piedini, per semplicità prendiamo in esame uno tra i microcontrollori più diffusi in questa fascia di contenitori e cioè il **PIC12F675-I/P**. Abbiamo previsto un programma di test denominato **T12F675** scritto in Pic Basic e disponibile nel CD sia in formato sorgente che eseguibile.

Vediamo ora le fasi necessarie per trasferire il file demo T12F675.HEX nel microcontrollore.

- Avviare IC-Prog e selezionare il dispositivo **PIC12F675**
- Selezionare il comando File in alto a sinistra
- Selezionare il comando Open File e di seguito il file **T12F675.HEX**
- Impostare oscillatore interno (**Oscillator = INTOSC**)
- Impostare **Bandgap = Default**
- Abilitare **PWRT** e **BODEN** e verificare che le altre opzioni del gruppo Fuses siano disabilitate
- Inserire un **PIC12F675** nello zoccolo a 8 pins DIL8 rispettando la polarità
- Alimentare la scheda
- Verificare che i jumper **JP7** e **JP8** scheda programmazione C-202 siano chiusi
- Chiudere i jumper **JP7**, **JP8** e **JP9** scheda socket C-203
- Portare il selettore **S18** in posizione **PROGRAM**
- Selezionare il comando Program All: comparirà la scritta "Do you really want to program the devices"; Selezionare **Yes**
- Comparirà la scritta "Oscillator calibration value = xxxx. Do you want to use value from file instead?" Rispondere **No** per mantenere il valore di calibrazione memorizzato dal costruttore nell'ultima locazione della memoria programma.
- Inizia quindi la fase di programmazione che si articola in 3 parti ognuna contraddistinta da una barra di progressione: Programming Code; Programming Data; Programming Config
- Di seguito inizia la fase di verifica dei dati scritti anch'essa evidenziata da una barra di progressione
- Se la verifica è andata a buon fine, compare la scritta Device Successfully Verified ed il microcontrollore è stato correttamente programmato
- Rimuovere i jumper **JP7**, **JP8** e **JP9** scheda socket C-203
- Rimuovere i jumper **JP7**, **JP8** scheda C-202
- Portare il selettore **S18** su **RUN**.
- Verificare con un oscilloscopio o con un tester che le linee del microcontrollore vadano sequenzialmente a livello logico1



Le linee GP risultano collegate alle linee PortB nel seguente modo:  
**GP0 = RB0; GP1 = RB1; GP2 = RB2; GP3 = RB3; GP4 = RB4; GP5 = RB5.**

Nota, la linea GP3 non è stata considerata nel test poichè può funzionare solo come ingresso.

```
*****
* Name  : T12F675.BAS
* Notice : Copyright (c) 2005 EVR electronics
*          : All Rights Reserved
* Date   : 26/01/2005
*****
```

TRISIO = 0 'All output

ANSEL = 0  
ADCON0 = 0

GPIO.0 = 0  
GPIO.1 = 0  
GPIO.2 = 0  
GPIO.4 = 0  
GPIO.5 = 0

LOOP:

GPIO.0 = 1  
PAUSE 2000  
GPIO.0 = 0  
PAUSE 2000  
GPIO.1 = 1  
PAUSE 2000  
GPIO.1 = 0  
PAUSE 2000  
GPIO.2 = 1  
PAUSE 2000  
GPIO.2 = 0  
PAUSE 2000  
GPIO.4 = 1  
PAUSE 2000  
GPIO.4 = 0  
PAUSE 2000  
GPIO.5 = 1  
PAUSE 2000  
GPIO.5 = 0  
PAUSE 2000

GOTO LOOP

**T12F675.BAS**

## Caratteristiche tecniche

Tensione di alimentazione: continua e regolata da 12Vdc a 17Vdc

- 12Vdc (solo normale funzionamento)

- 17Vdc (normale funzionamento e programmazione MCU)

Corrente di alimentazione: minimo 300mA

Temperatura di funzionamento: da 0°C a 40°C

Umidità massima: <RH80%

Dimensioni scheda programmazione C-202: 65 x 112 mm

Dimensioni scheda socket C-203: 89 x 112 mm

Peso scheda programmazione C-202: 70 gr.

Peso scheda socket C-203: 60 gr

## Garanzia

Questo prodotto è garantito contro difetti di componenti e assemblaggio come stabilito dei termini di Legge per un periodo di un anno dalla data della vendita. La garanzia è valida solo se l'utilizzatore dispone di una copia originale di prova di acquisto quale fattura o scontrino fiscale.

La responsabilità del costruttore è limitata alla riparazione del difetto o, se necessario, alla sostituzione o riparazione del componente difettoso. I costi e i rischi connessi al trasporto, alla rimozione o al riposizionamento del prodotto, e qualsiasi altro costo direttamente o indirettamente collegato alla riparazione, non può essere imputato al costruttore.

Il costruttore non può essere ritenuto responsabile di qualsiasi danno causato da un malfunzionamento del prodotto.

## Istruzioni sulla sicurezza

- Questo prodotto è stato progettato e realizzato per essere utilizzato in un laboratorio di elettronica e su di un banco di lavoro munito di protezioni contro le scariche elettrostatiche
- Tutti i cavi aggiunti al prodotto per collegarlo ad altri dispositivi o al personal computer devono essere muniti di ferriti per limitare le emissioni elettromagnetiche
- Non disassemblare o rimuovere delle parti o componenti dal prodotto
- Maneggiare il prodotto con cura e delicatezza
- Non superare i valori limite indicati nelle caratteristiche tecniche
- Non bagnare il prodotto
- Assicurarsi che il prodotto disponga della necessaria ventilazione
- Impedire che i bambini possano avvicinarsi al prodotto

## Supporto tecnico

Per supporto tecnico relativo al prodotto e per riparazioni potete contattarci al seguente indirizzo:

**support@evr-electronics.com**

PIC is a Registered trademark of Microchip company.

All other trademarks are the property of their respective owners.

Il costruttore si riserva il diritto di cambiare le specifiche dei prodotti o di cessare la produzione dei prodotti senza preavviso e di dovere incorporare o fornire le nuove funzioni o le nuove istruzioni nei prodotti già venduti. Il costruttore non può essere ritenuto responsabile per perdite o danni, diretti o indiretti, che possono derivare dall'uso dei prodotti. I prodotti non sono adatti per l'uso come parti di sistemi di supporto vitale, o sistemi che possono creare situazioni pericolose di qualsiasi tipo.



Con riserva di modifica ed errori tipografici:  
Copyright©2005 EVR Electronics  
All right reserved  
[www.evr-electronics.com](http://www.evr-electronics.com)  
[www.c-project.com](http://www.c-project.com)

Modello: C-170  
File: C-170 Manuale utente Rev1.qxp  
Data: 20 Gennaio 2005

