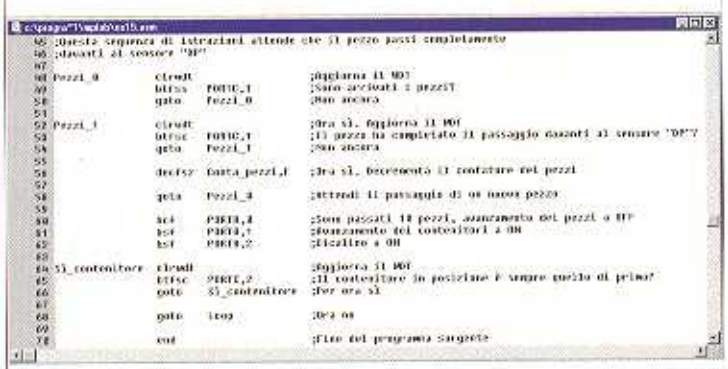
Esercizi di apprendimento

```
;
;Macchina di confezionamento
;
;Due relè "H1" (RB0)  e "H2" (RB1), comandano due motori che fanno muovere due nastri
;trasportatori. "H1" (RB0) trasporta pezzi e "H2" (RB1) imballaggi. Un sensore "DP" (RC1)
;rileva il passaggio dei pezzi e l'altro, "DE" (RC2), rileva il corretto posizionamento di
;un contenitore. Dopo che sono passati 10 pezzi, la confezione si considera piena, si
;attiva un segnale acustico, "H" (RB2), e il nastro che trasporta gli imballaggi parte,
;sino a posizionare un nuovo imballo vuoto. In questo momento si disattiva il segnale
;acustico "H" (RB2), e avanza nuovamente il nastro dei pezzi ripetendo così il ciclo.
;Un interruttore "I" (RC0) attiva o disattiva l'intero sistema.

                LIST    p=16F870                ;Tipo di processore
                include "P16F870.INC"   ;Definizione dei registri interni

Conta_pezzi     equ     0x20            ;Variabile N° di pezzi

                org     0x00

Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona banco 1
                clrf    TRISB           ;Configura la Porta B come uscita
                movlw   0xFF
                movwf   TRISC           ;Configura la Porta A come ingresso
                bcf     STATUS,RP0      ;Seleziona banco 0
```

Gli esercizi che abbiamo realizzato sinora erano esempi combinatori. In questo tipo di esercizi si disponeva di un insieme di segnali di ingresso che venivano letti una volta, si operava su di essi e si ottenevano dei risultati che erano visualizzati tramite i diodi led. Questo tipo di funzionamento è comune in molti algoritmi per i robot.
Si legge lo stato dei sensori e in funzione della combinazione letta si eseguono diversi ordini sui motori. Ora realizzeremo un tipo diverso di esercizi, quelli chiamati sequenziali. Le azioni sono determinate dal modo in cui si attivano i sensori.

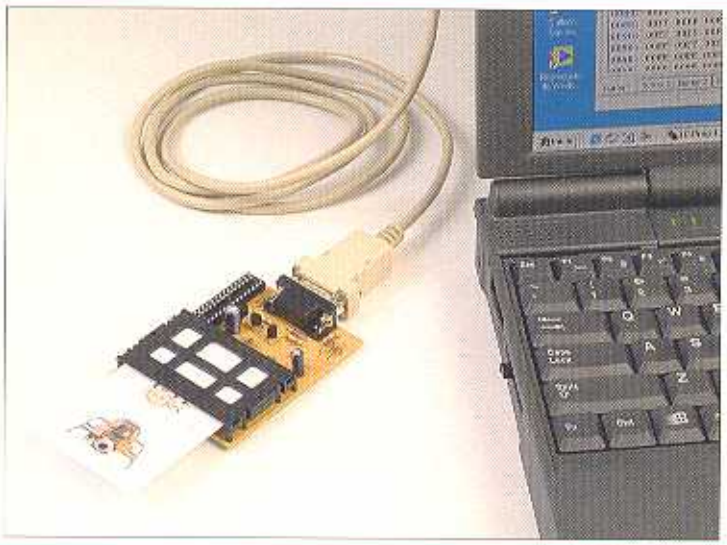
```
Loop_0          clrf    PORTB           ;Resetta le uscite
Loop:           clrwdt                  ;Aggiorna il WDT
                movlw   d'10'
                movwf   Conta_pezzi     ;Inizializza la variabile N° di pezzi
                btfss   PORTC,0         ;Testa lo stato dell'interruttore I
                goto    Loop_0          ;Il su OFF, sistema fermo

                bcf     PORTB,0         ;Avanzamento dei pezzi a OFF
                bsf     PORTB,1         ;Avanzamento dei contenitori a ON

Ha_contenitore  clrwdt                  ;Aggiorna il WDT
                btfss   PORTC,2         ;Contenitore in posizione?
                goto    Ha_contenitore  ;No.

                bsf     PORTB,0         ;Sì. Avanzamento dei pezzi a ON
                bcf     PORTB,1         ;Avanzamento dei contenitori a OFF
                bcf     PORTB,2         ;Cicalino a OFF
```

L'esercizio sequenziale che vogliamo provare si trova sul CDROM sotto il nome di es15.asm. Simula una macchina di confezionamento che trasporta pezzi su due nastri trasportatori. È necessario realizzare una serie di operazioni sui pezzi, non tutte assieme ma in modo sequenziale, quindi secondo come si attivano i sensori (simulati con gli interruttori della porta C), entreranno in funzione diversi attuatori (simulati con i diodi LED della porta B).

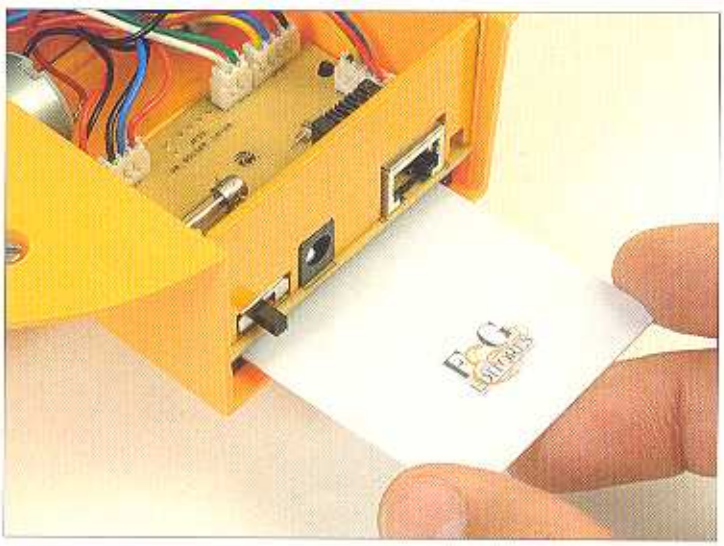


```
;Questa sequenza di istruzioni attende che il pezzo passi completamente
;davanti al sensore "DP"

Pezzi_0         clrwdt                  ;Aggiorna il WDT
                btfss   PORTC,1         ;Sono arrivati i pezzi?
                goto    Pezzi_0         ;Non ancora

Pezzi_1         clrwdt                  ;Ora sì. Aggiorna il WDT
                btfsc   PORTC,1         ;Il pezzo ha completato il passaggio davanti al sensore "DP"?
                goto    Pezzi_1         ;Non ancora

                decfsz  Conta_pezzi,f   ;Ora sì. Decremento il contatore dei pezzi

                goto    Pezzi_0         ;Attendi il passaggio di un nuovo pezzo

                bcf     PORTB,0         ;Sono passati 10 pezzi, avanzamento dei pezzi a OFF
                bsf     PORTB,1         ;Avanzamento dei contenitori a ON
                bsf     PORTB,2         ;Cicalino a ON

Sì_contenitore  clrwdt                  ;Aggiorna il WDT
                btfsc   PORTC,2         ;Il contenitore in posizione è sempre quello di prima?
                goto    Sì_contenitore  ;Per ora sì

                goto    Loop            ;Ora no

                end                     ;Fine del programma sorgente
```

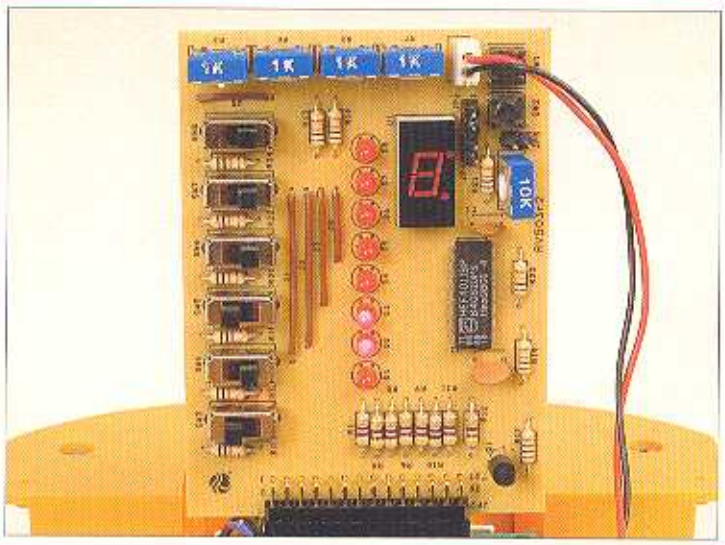
All'inizio del programma si realizzano le configurazioni: porta C come ingresso per leggere i sensori e porta B come uscita per inviare segnali agli attuatori.
In seguito si entra in una serie di cicli di attesa con le istruzioni di salto btfss e btfsc.
In questo modo, finché i sensori non hanno il valore appropriato al momento giusto, il programma rimane in attesa. Secondo come si compiono le condizioni di attivazione di ogni sensore vengono attivate le uscite corrispondenti, e si passa al ciclo successivo.



Per compilare il programma copieremo il file es15.asm sull'hard disk. Nella cartella in cui si copia dovrà essere inserito anche il file P16F870.INC, che contiene tutte le definizioni dei registri del microcontroller. Potremo generare un progetto e inserire in esso il file o semplicemente aprirlo e selezionare l'opzione Build Node del menù Project per compilare l'esercizio. Dopo aver ottenuto il file es15.hex, daremo seguito alla programmazione della Smartcard utilizzando il programma ICPROG e la scheda di scrittura con lo stesso procedimento che conosciamo già.



Per verificare questo esercizio, la scheda di ingresso e uscita dovrà essere collegata alla scheda di interfaccia tramite il connettore JP13. Il microcontroller dovrà essere inserito sulla scheda di controllo con il programma uploader.hex caricato. La Smartcard verrà inserita nella scheda di alimentazione con il programma es15.hex scritto. Dopo aver alimentato il robot, il microcontroller impiegherà alcuni secondi a leggere il contenuto della scheda Smartcard e a iniziare l'esecuzione del programma. Sulla scheda di ingresso e uscita dovrà essere chiuso il jumper JP1 e aperti i jumper JP4 e JP5.



Quando si inizia a provare l'esercizio, tutti gli interruttori dovranno essere posizionati in modo da inviare uno zero logico (la levetta dell'interruttore verso l'interno della scheda). Ora procederemo ad attivare gli interruttori nell'ordine adeguato, simulando l'attivazione dei sensori del nastro trasportatore. I diodi LED si attiveranno in modo sequenziale, simulando le diverse operazioni che realizzano gli attuatori sui pezzi presenti sul nastro trasportatore. A causa dell'effetto rimbalzo dei sensori meccanici, che vi abbiamo spiegato in passato, in alcuni casi anche con meno di dieci attivazioni dell'interruttore, il sistema conterà ugualmente dieci pezzi. In altri esercizi sarà applicata la soluzione software per correggere l'effetto rimbalzo.