

Esercizi di apprendimento

```

1 ;
2 ;Generazione di numeri casuali. Il dado elettronico
3 ;
4 ;SS tratta di generare un numero casuale fra 1 e 6. Quando R00 è a "1", sopra il
5 ;display a 7 segmenti collegato alla porta B, si visualizzano in modo sequenziale
6 ;il numeri da 1 a 6, con intervalli di 0,25". Dopo la R00 passa a livello "0", si visualizzerà
7 ;il numero casuale ottenuto per un tempo di 3". Dopo il display si spegne e la
8 ;sequenza si ripete.
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;
27 ;
28 ;
29 ;
30 ;
31 ;
32 ;
33 ;
34 ;
35 ;
36 ;
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 ;
48 ;
49 ;
50 ;
51 ;
52 ;
53 ;
54 ;
55 ;
56 ;
57 ;
58 ;
59 ;
60 ;
61 ;
62 ;
63 ;
64 ;
65 ;
66 ;
67 ;
68 ;
69 ;
70 ;
71 ;
72 ;
73 ;
74 ;
75 ;
76 ;
77 ;
78 ;
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ;
89 ;
90 ;
91 ;
92 ;
93 ;
94 ;
95 ;
96 ;
97 ;
98 ;
99 ;
100 ;

```

Questo esercizio combina quasi tutte le conoscenze sulla programmazione che abbiamo acquisito sino a questo momento. Si tratta di sviluppare un dado elettronico che genererà numeri casuali fra 1 e 6, visualizzati tramite il display a 7 segmenti della scheda di ingressi e uscite. Avremo bisogno di leggere l'interruttore SW3 per attivare il dado, eliminare l'effetto rimbalzo, gestire una routine di generazione di numeri casuali, fare in modo che il numero sia compreso fra 1 e 6 e realizzare una temporizzazione per fare spegnere il display dopo 3 secondi.

```

20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;
27 ;
28 ;
29 ;
30 ;
31 ;
32 ;
33 ;
34 ;
35 ;
36 ;
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 ;
48 ;
49 ;
50 ;
51 ;
52 ;
53 ;
54 ;
55 ;
56 ;
57 ;
58 ;
59 ;
60 ;
61 ;
62 ;
63 ;
64 ;
65 ;
66 ;
67 ;
68 ;
69 ;
70 ;
71 ;
72 ;
73 ;
74 ;
75 ;
76 ;
77 ;
78 ;
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ;
89 ;
90 ;
91 ;
92 ;
93 ;
94 ;
95 ;
96 ;
97 ;
98 ;
99 ;
100 ;

```

Nella routine di inizio realizzeremo le configurazioni dei registri per il programma. La porta C come ingresso, la porta B come uscita e il prescaler a 256 per il Timer0. Nel ciclo "Loop" si attende che l'interruttore RC0 passi a 1. In questo momento si acquisisce il valore del Timer0, portandolo sulla variabile "Numero", detto valore sarà un numero casuale fra 0 e 255. Il dato da visualizzare sul display a 7 segmenti deve essere compreso fra 1 e 6, per questo motivo nella routine "Dividi" si opera con la variabile "Numero" sino a che il suo valore casuale sia compreso in questo intervallo.

```

40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 ;
48 ;
49 ;
50 ;
51 ;
52 ;
53 ;
54 ;
55 ;
56 ;
57 ;
58 ;
59 ;
60 ;
61 ;
62 ;
63 ;
64 ;
65 ;
66 ;
67 ;
68 ;
69 ;
70 ;
71 ;
72 ;
73 ;
74 ;
75 ;
76 ;
77 ;
78 ;
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ;
89 ;
90 ;
91 ;
92 ;
93 ;
94 ;
95 ;
96 ;
97 ;
98 ;
99 ;
100 ;

```

Sino a quando l'interruttore RC0 non passa a valore 0, non si deve visualizzare il numero casuale sul display a 7 segmenti. Nell'attesa che ciò accada, la routine dado opera sul display visualizzando numeri casuali ad alta velocità; all'interno di questo ciclo si testa RC0, in modo che quando l'interruttore SW3 passa a valore 0, termini il ciclo e si visualizzi il numero casuale generato, scritto nella variabile "Numero", tramite il display a 7 segmenti. Dopo aver visualizzato il numero casuale si temporizzano tre secondi e si spegne il display.



```

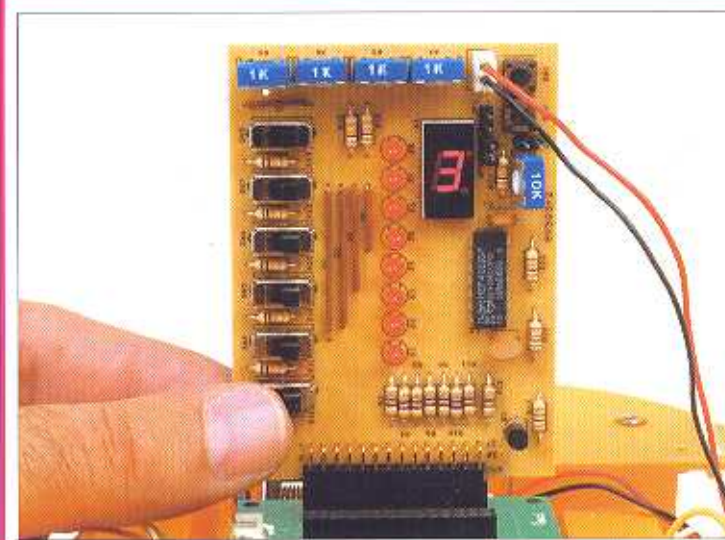
70 ;Tabella: Questa routine converte il codice binario presente sui 4 bit meno significativi
80 del reg. R nel suo equivalente a 7 segmenti. Il codice a 7 segmenti viene anche scritto
81 sul reg. R
82
83 ;Tabella:
84 movlw 0x00000000 ;spostamento sopra la tabella
85 movwf 0x00000000 ;digit 0
86 movlw 0x00000010 ;digit 1
87 movwf 0x00000000 ;digit 2
88 movlw 0x00000011 ;digit 3
89 movwf 0x00000000 ;digit 4
90 movlw 0x00000010 ;digit 5
91 movwf 0x00000000 ;digit 6
92
93 ;Delay 20 ms: questa routine di ritardo ha come obiettivo eliminare l'effetto rimbalzo",
94 caratteristica dei componenti elettronici. Realizza un ritardo di 20 ms.
95 Se il PIC lavora ad una frequenza di 4 MHz, il 1000 si aggiorna ogni µs. Se vogliamo
96 temporizzare 20000 µs (20 ms) con un prescaler di 256, il 1000 dovrà contare 74 eventi
97 (10k * 256). Il valore 20 equivale a 100 hex, e dato che il 1000 è ascendente lo dividerò
98 caricherò con il suo complemento a 1 (0001 hex.).
99
100 Delay_20_ms: bcf INTCON,TOIF ;resetta il flag di overflow del 1000
101 movlw 0x01 ;complemento hex. di 20
102 movwf 1000 ;carica il 1000
103 Delay_20_ms_1: cfsrwf ;aggiorna il 1001
104 ultra INTCON,TOIF ;overflow del 1000?
105 goto Delay_20_ms_1 ;Non ancora
106 return
    
```

Negli esercizi precedenti abbiamo utilizzato le due routines mostrate nell'immagine. La prima di esse è la routine di conversione dei numeri binari a numeri in codice per il display a 7 segmenti. È necessario richiamare questa routine prima di inviare il numero casuale alla porta B. L'altra funzione è una temporizzazione di 20 ms utilizzata per eliminare l'effetto rimbalzo dell'interruttore.

```

107
108 ;Delay_var: Questa routine di utilizzo generale realizza una temporizzazione variabile
109 fra 50 ms e 12,0". Si utilizza un prescaler di 256 e sul 1000 si carica 195.
110 La velocità di lavoro è di 4 MHz e quindi il 1000 si incrementa ogni µs. In
111 questo modo, il 1000 deve contare 745 eventi che, con un prescaler di 128 danno un
112 intervallo totale di 50000 µs = 50 ms (195 * 256). Il valore 195 lo dobbiamo esprimere
113 in hex. (c3) e, dato che il 1000 è ascendente dovremo caricare il suo complemento a 1 (3c hex.)
114 Questo intervallo di 50 ms si ripete tante volte quante indicate dalla variabile "Delay_cont".
115 Per questo il ritardo minimo è di 50 ms ("Delay_cont=1") e quello massimo di 12,0 sec.
116 ("Delay_cont=255").
117
118 Delay_var: bcf INTCON,TOIF ;resetta il flag di overflow
119 movlw 0xc3 ;complemento hex. di 195
120 movwf 1000 ;carica il 1000
121 intervallo: cfsrwf ;aggiorna il 1001
122 ultra INTCON,TOIF ;overflow del 1000?
123 goto intervallo ;Non ancora
124 decfz Delay_cont,F ;decrementa il contatore degli intervalli
125 goto Delay_var ;ripete l'intervallo di 50 ms
126 return
127
128 ;fine del programma sorgente
    
```

Questa funzione è già stata utilizzata in precedenti programmi di generazione di numeri casuali, con essa realiziamo una temporizzazione di tre secondi che sarà il tempo per cui rimarrà il dato visualizzato sul display a 7 segmenti. Questa funzione realizza una temporizzazione sulla base di 50 ms che si ripete tante volte quante indicate nella variabile Delay_cont. Nel caso si volessero temporizzare tre secondi, è necessario inizializzare questa variabile con il valore 60, prima di entrare nella funzione. Se vogliamo temporizzare solamente un secondo, caricheremo il dato 20 in questa variabile.



Dopo aver compilato il programma procederemo alla sua scrittura sulla Smartcard, utilizzando il software ICPROG e la scheda di scrittura, seguendo la stessa procedura già applicata per il resto degli esercizi. Per provare il programma dobbiamo aprire i jumper JP1 e JP5 della scheda di ingressi e uscite e chiudere il jumper JP4 della stessa scheda. In questo modo attiveremo il display a 7 segmenti. Utilizzeremo l'interruttore SW3 per iniziare la generazione e la visualizzazione dei numeri casuali sul display.

