

Istruzioni di gestione delle memorie dei dati

In una sezione precedente, abbiamo trattato le istruzioni che dobbiamo utilizzare per prendere e fornire dati sulle porte di ingresso/uscita. E se i dati non fossero introdotti da un utente? Se invece di visualizzarli li dovessimo memorizzare per un utilizzo successivo? In questo caso dovremmo utilizzare alcune memorie e le istruzioni corrispondenti, che ci permetteranno di immagazzinare e recuperare questi dati.

Differenti memorie di dati

Fra tutti i tipi di memorie che si possono trovare nei differenti sistemi di controllo, due sono quelle che principalmente si utilizzano nell'immagazzinamento dei dati: le memorie tipo RAM e le EEPROM. Le prime perdono i loro dati quando si toglie l'alimentazione al sistema in cui si trovano — in altre parole quando viene spento — da qui deriva il loro nome di memorie volatili. La memoria EEPROM invece conserva i dati sino a quando vogliamo noi, anche se non c'è alimentazione. Come può essere sfruttata questa caratteristica? Pensate ad esempio a un personal computer, voi sapete che ha una memoria RAM necessaria ai programmi per funzionare, e il disco rigido, dove sono contenuti

i dati e i programmi: è in fin dei conti una specie di memoria EEPROM. Siete sicuri di non avere attorno a voi elettrodomestici che utilizzano qualcuno di questi tipi di memoria? Cosa mi dite del televisore? Pensate se si dovessero sintonizzare i canali ogni volta che si accende! E il caricatore dei CD della macchina che si ricorda dell'ultima canzone quando lo riaccendete?



Molti elettrodomestici lavorano con memorie per la conservazione dei dati.

Lavoro con la memoria RAM dei dati

Il microcontroller PIC con cui stiamo lavorando, contengono questi due tipi di memorie per immagazzinare i dati: la RAM e

la EEPROM. La gestione della memoria RAM è più rapida e semplice, come abbiamo già visto con gli esercizi. Ogni volta che definiamo un registro, questo appartiene alla memoria

```
File Edit Compile Options Help
[Icons]
1 device 16F84
2
3 DIM A, B, C
4
5 A=PEEK (3) ; assegnamo ad A il valore del registro 3
6 ; della memoria RAM
7
8 B=6 ; carichiamo B con il valore 6
9 C=PEEK (B) ; assegnamo a C il valore del registro B,
10 ; che è il registro 6 della memoria RAM
```

Tempo fa erano necessarie delle istruzioni speciali per prendere il dato da un registro specifico.

RAM, ed è conosciuto come "registro generale", che l'utente può utilizzare come meglio crede.

Nelle versioni precedenti del LetPicBasic erano necessarie istruzioni specifiche per la gestione di questi registri, quindi l'utilizzo di registri RAM oppure di registri EEPROM non comportava nessun vantaggio dal punto di vista dell'utente. L'istruzione "PEEK" aveva come parametro il numero del registro o il nome di una variabile, e si utilizzava per prendere il valore contenuto in questo registro o variabile per

passarlo a un'altra variabile. L'istruzione "POKE" realizzava la funzione contraria, cioè assegnava un valore a un registro. Il primo parametro era l'indirizzo di destinazione e il secondo il dato da introdurre in questo indirizzo.

Quindi a una variabile definita dall'utente si poteva assegnare un valore direttamente, mentre per un registro specifico bisognava utilizzare l'istruzione "POKE".

Queste erano istruzioni molto importanti, tuttavia nelle nuove versioni del software possiamo dimenticarci di esse, e lavorare direttamente sui registri, sia con quelli definiti dall'utente con il comando "DIM" sia con quelli specifici del microcontroller, che hanno già un nome proprio.

```
File Edit Compile Options Help
PIC-BASIC LITE V7.1
1 device 16F84
2
3 DIM A
4
5 A=10 ; assegnamo ad A il valore 10
6
7 POKE (3,6) ; assegnamo al registro 3 della RAM
8 ; il valore 6
9 POKE (3,A) ; sul registro 3 della RAM introduciamo
10 ; il valore della variabile A, in questo
11 ; caso il numero 10
12
```

Anche l'assegnazione di valori ai registri specifici si faceva con istruzioni speciali.

```
PIC-BASIC LITE V7.1 - C:\Programmi\LE.T. PIC Basic Lite\programmi di prova\PE14 CP013_3.bas
File Edit Compile Options Help
1 device 16F84
2
3 DIM A, B, C
4
5 A=STATUS ; carichiamo su A il valore del registro 3
6 ; della memoria RAM, che è il registro STATUS
7
8 B=6 ; assegnamo a B il valore 6
9 C=PORTB ; assegnamo a C il valore del registro PORTB
10 ; che è il registro 6 della memoria RAM
11 C=B ; assegnamo a C il valore della variabile B,
12 ; che è 6
13
14 STATUS=B ; al registro STATUS si assegna il valore della
15 ; variabile B, che è 6
16
17 STATUS=6 ; assegnamo al registro STATUS il valore 6
```

Nelle versioni nuove si può lavorare direttamente sui registri.

Immagazzinamento dell'informazione



Lavoro con la memoria EEPROM dei dati

Dopo aver letto nel capitolo precedente il motivo per cui sono utilizzate le

Spesso utilizziamo dispositivi che funzionano grazie alla memoria EEPROM che hanno all'interno.