memorie EEPROM dei dati, verrebbe da pensare che memorie di questo tipo siano da utilizzare sempre.

Tuttavia anch'esse hanno i loro inconvenienti, ad esempio hanno tempi di lettura e di scrittura più lunghi.

Inoltre permettono un numero di cicli di lettura/scrittura minore. Facciamo due esempi: è molto probabile che voi utilizziate spesso qualche tipo di Smartcard, ad esempio le schede telefoniche in cui sono contenuti alcuni tipi di dati, come ad esempio l'importo residuo.

Queste schede sono alimentate solamente quando vengono introdotte nell'apposito lettore, e quando si ritirano tornano ad essere senza alimentazione.

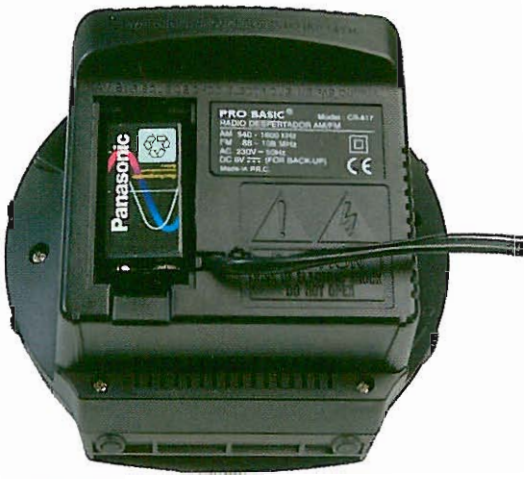
Questo è un classico esempio dove è necessario l'utilizzo della memoria EEPROM, infatti in una memoria normale il dato relativo al credito residuo dopo la nuova telefonata andrebbe perso al momento dell'estrazione della scheda dal lettore.

Se dal punto di vista dell'utente questo potrebbe essere un vantaggio, sicuramente non lo è per il gestore.

Invece che tipo di memoria credete che utilizzino le sveglie elettroniche per memorizzare l'ora attuale e quella di allarme? Se manca l'alimentazione ritorna allo zero, vero? Questo perché si utilizza una memoria RAM.

Sarebbe interessante che anche nel caso mancasse la tensione (magari mentre stiamo dormendo), l'ora rimanesse corretta, ma l'aggiornamento della EEPROM dovrebbe essere fatto in modo costante e continuativo, quindi si perderebbe tempo nello scrivere la EEPROM e questo si ripercuoterebbe sulle altre parti dell'applicazione.

La migliore opzione di questi elettrodomestici consiste nell'incorporare una pila ausiliaria che sostituisca l'alimentazione principale quando questa manca.



Non sempre è consigliabile l'utilizzo della memoria EEPROM.

Istruzioni di gestione della memoria EEPROM dei dati

Si utilizzano due istruzioni per la scrittura e la lettura delle memorie EEPROM: "STORE" e "EEDATA". Entrambe sono simili a "POKE" e "PEEK" rispettivamente.

Con "STORE" si scrive un dato nella EEPROM dei dati del microcontroller. Il primo parametro sarà l'indirizzo della memoria, che per un PIC16C84 o per un PIC16F84 deve essere compreso fra 0 e 63. Il secondo parametro sarà il valore da scrivere nella EEPROM. Si possono utilizzare direttamente numeri o variabili definiti dall'utente, a cui in precedenza è stato dato un valore.

Per contro con "EEDATA" possiamo leggere un dato della

```
File  Edit  Compile  Options  Help
1   device  16F84
2
3   DIM A,B
4
5   A=4 : B=6                   ; carichiamo la variabile A con il valore 4
6                               ; e la variabile B con il valore 6
7                               ; si può fare sulla stessa linea, oppure
8                               ; su linee differenti
9
10  STORE 10,20                 ; all'indirizzo 10 della memoria EEPROM
11                              ; introduciamo il valore 20
12
13  STORE A,3                   ; all'indirizzo 4 (valore di A) si introduce
14                              ; il valore 3
15
16  STORE B,A                   ; all'indirizzo 6 (valore di B) si introduce
17                              ; il valore 4 (valore di A)
18
```

L'istruzione "STORE" è simile alla "POKE".

EEPROM interna di un PIC 16C84 o PIC16F84.

I valori degli indirizzi dovranno anch'essi essere compresi fra 0 e 63, trattandosi di valori numerici o variabili a cui in precedenza è stato assegnato un dato. Il valore preso dalla memoria dovrà essere contenuto in una variabile.

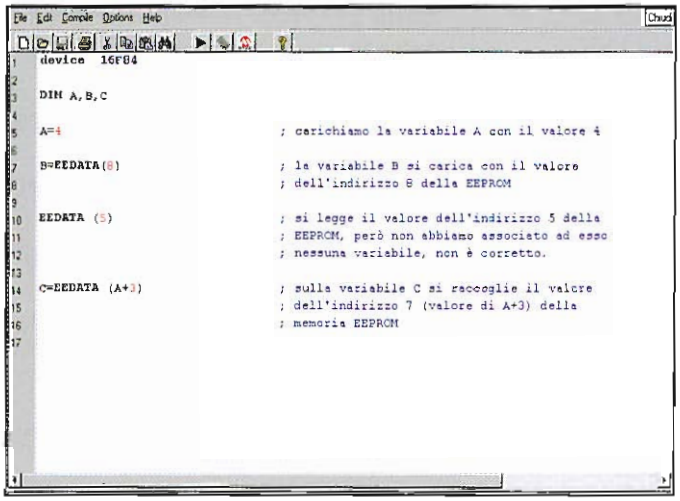
Se si prova a compilare le istruzioni che vi abbiamo spiegato prima, il compilatore ci avviserà dell'esistenza di errori. Riuscite a immaginare che cosa può essere successo? Effettivamente lo abbiamo visto quando abbiamo parlato delle varie parti di un programma.

Ci sono delle risorse che per essere utilizzate devono prima essere caricate e inizializzate. Una di queste è proprio la memoria EEPROM. Dovremo inserire i comandi "INCLUDE" e "INIT" prima che nel programma appaiano le istruzioni "STORE" o "EEDATA".

Con "INCLUDE", si assegnano variabili, routines e informazioni diverse da pacchetti predefiniti. Questi pacchetti, e quindi i parametri del comando, potranno essere: LCD, KEYPAD, A2D, I2CBUS, EEPROM e SERIALE. In ogni programma bisognerà includere i pacchetti di volta in volta necessari. Con "INIT" si assegna un codice ai pacchetti software inclusi in precedenza.

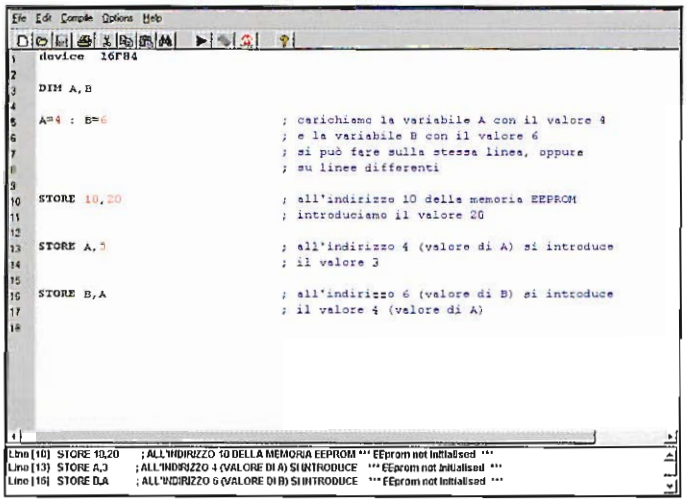
I parametri di "INIT" saranno il nome del pacchetto e altre informazioni complementari nel caso siano necessarie.

Per la EEPROM non è necessario un secondo parametro.



```
device  16F84

DIM A,B,C

A=4                         ; carichiamo la variabile A con il valore 4

B=EEDATA(8)                 ; la variabile B si carica con il valore
                            ; dell'indirizzo 8 della EEPROM

EEDATA (5)                  ; si legge il valore dell'indirizzo 5 della
                            ; EEPROM, però non abbiamo associato ad esso
                            ; nessuna variabile, non è corretto.

C=EEDATA (A+3)              ; sulla variabile C si raccoglie il valore
                            ; dell'indirizzo 7 (valore di A+3) della
                            ; memoria EEPROM
```

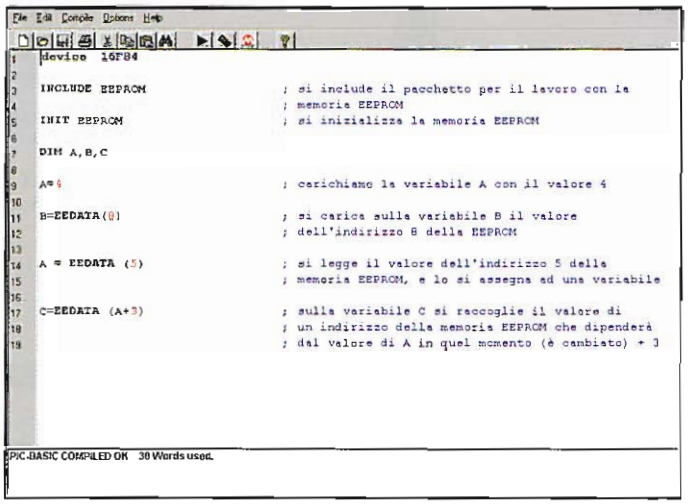
Esempio di utilizzo dell'istruzione "EEDATA".



```
device  16F84

DIM A,B

A=4 : B=6                   ; carichiamo la variabile A con il valore 4
                            ; e la variabile B con il valore 6
                            ; si può fare sulla stessa linea, oppure
                            ; su linee differenti

STORE 10,20                 ; all'indirizzo 10 della memoria EEPROM
                            ; introduciamo il valore 20

STORE A,3                   ; all'indirizzo 4 (valore di A) si introduce
                            ; il valore 3

STORE B,A                   ; all'indirizzo 6 (valore di B) si introduce
                            ; il valore 4 (valore di A)
```

```
Line (10)  STORE 10,20   ; ALL'INDIRIZZO 10 DELLA MEMORIA EEPROM *** EEprom not initialised ***
Line (13)  STORE A,3     ; ALL'INDIRIZZO 4 (VALORE DI A) SI INTRODUCE  *** EEprom not initialised ***
Line (15)  STORE B,A     ; ALL'INDIRIZZO 6 (VALORE DI B) SI INTRODUCE  *** EEprom not initialised ***
```

Se compiliamo gli esempi saremo avvisati di un errore.



```
device  16F84

INCLUDE EEPROM              ; si include il pacchetto per il lavoro con la
                            ; memoria EEPROM
INIT EEPROM                 ; si inizializza la memoria EEPROM

DIM A,B,C

A=4                         ; carichiamo la variabile A con il valore 4

B=EEDATA(8)                 ; si carica sulla variabile B il valore
                            ; dell'indirizzo 8 della EEPROM

A = EEDATA (5)              ; si legge il valore dell'indirizzo 5 della
                            ; memoria EEPROM, e lo si assegna ad una variabile

C=EEDATA (A+3)              ; sulla variabile C si raccoglie il valore di
                            ; un indirizzo della memoria EEPROM che dipenderà
                            ; dal valore di A in quel momento (è cambiato) + 3
```

```
PIC-BASIC COMPILED OK   30 Words used.
```

Modo corretto di utilizzare l'istruzione "EEDATA".