

## Similitudini e differenze fra il LetPicBasic e il PicBasicPlus

Il PicBasicPlus è un compilatore più evoluto del LetPicBasic. Oltre a permetterci di lavorare con più dispositivi, è stata anche migliorata la facilità di programmazione grazie al numero di istruzioni e strutture di controllo aggiuntive.

La prima differenza, come abbiamo già visto installando entrambi i programmi, si trova nell'ambiente di lavoro. Vediamone alcune tra le più importanti.

### Struttura di un programma in PicBasicPlus

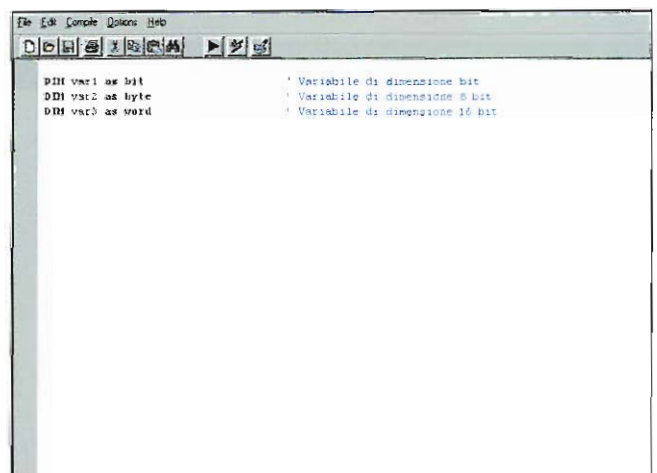
Così come nel LetPicBasic, esiste un ordine degli elementi da inserire in un programma; continueremo ad aver bisogno della direttiva "DEVICE" per indicare il tipo di microcontroller con cui vogliamo lavorare, perché da esso dipenderanno, ad esempio, i dispositivi che potremo utilizzare o il modo in cui si troveranno i piedini e i registri reinizializzando il sistema. Inoltre è possibile anche utilizzare la direttiva "INCLUDE", però con un significato diverso da quello che aveva nel LetPicBasic, e che vedremo in un altro capitolo.

Gli identificatori, che siano essi etichette, nomi di variabili o costanti, potranno contenere sia lettere che numeri e anche alcuni caratteri speciali, sempre

PP0, PP0H, PP1, PP1H, PP2, PP2H, PP3, PP3H, PP4, PP4H, PP5, PP5H, PP6, PP6H, PP7, PP7H, GEN, GENH, GEN2, GEN2H, GEN4, GEN4H, GPR, BPF.

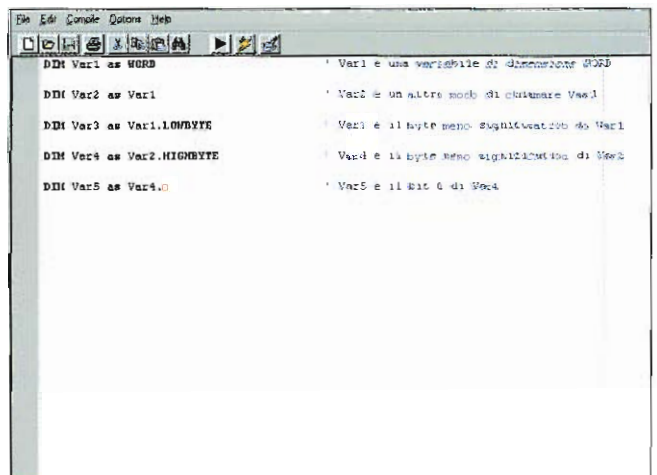
**Parole riservate** che non possono essere utilizzate come nomi.

**Dichiarazioni di variabili con "DIM".**



```
File Edit Compile Options Help
[Icons]
DIM Var1 as bit           ' Variabile di dimensione bit
DIM Var2 as byte         ' Variabile di dimensione 8 bit
DIM Var3 as word         ' Variabile di dimensione 16 bit
```

**Altri utilizzi di "DIM".**



```
File Edit Compile Options Help
[Icons]
DIM Var1 as WORD         ' Var1 è una variabile di dimensione WORD
DIM Var2 as Var1         ' Var2 è un altro modo di dichiarare Var1
DIM Var3 as Var1.LOWBYTE ' Var3 è il byte meno significativo di Var1
DIM Var4 as Var2.HIGHBYTE ' Var4 è il byte meno significativo di Var2
DIM Var5 as Var4.0       ' Var5 è il bit 0 di Var4
```

che non inizino con dei numeri. Sono riconosciuti i primi 32 caratteri, e possono essere maiuscoli o minuscoli indistintamente. Ci sono alcuni nomi che non possono essere utilizzati come identificatori,

trattandosi di parole riservate. Le etichette si dichiarano come nel LetPicBasic. Le variabili del programma si dichiarano con "DIM", e si dovrà scegliere la dimensione che meglio si adegua fra "bit", "byte" o "word",

```
File Edit Compile Options Help
DIM Var1 as 1      * Si potrà scrivere "1" o "Var1" indistintamente
DIM Var100 as Var1 * 100  * "Var100" sarà una costante di valore 100
```

**Definizioni di costanti.**

```
File Edit Compile Options Help
DIM Var1 as 5      * Var1 è una costante di valore 5
PORTA=3           * Si imposta il valore 3 in PORTA
DIM Var2 as Var1 + PORTA  * Var2 sarà una costante di valore 8
SYMBOL CCM = (PORTA + Var1) * CCM assumerà valore 10 (5+5)
```

**Differenza fra DIM e SYMBOL.**

```
File Edit Compile Options Help
TRISB = 00000000  * Si definisce la porta B come uscita
TRISA = 00000101  * Si definisce la porta A come una miscela di ingressi e uscite
```

**Dichiarazioni delle linee di ingresso / uscita.**

accedere così più comodamente ai valori. Le costanti si definiranno ugualmente con "DIM", come si può vedere dalla figura, e non potranno variare il loro valore. "SYMBOL" permette di utilizzare altri nomi per le variabili già definite con "DIM" ed è un altro modo di definire le costanti, però non utilizza indirizzi della memoria RAM.

Tuttavia bisogna fare attenzione, perché questa non è l'unica variazione dato che, ad esempio, utilizzando espressioni con il nome di variabili o registri, non si ottiene lo stesso risultato utilizzando "SYMBOL" oppure utilizzando "DIM". Come avveniva con il LetPicBasic, "SYMBOL" ci può aiutare a definire pin indipendenti di una porta o di un registro. Non sarà più necessaria la direttiva "DEFINE" per dichiarare i pin delle porte come ingresso o uscita, dato che questo verrà fatto direttamente con un'espressione di uguaglianza. Gli zeri (0) continueranno a significare uscita e gli uno (1) ingresso. "DATA" continuerà a essere utilizzato per definire le tabelle dei dati alfanumerici che potranno essere trattate individualmente, anche se determinati comandi potranno far uso di catene di caratteri inseriti fra virgolette, e trattati come una sola unità. Questi sono gli unici casi in cui il compilatore differenzia fra maiuscole e minuscole.

in altre parole, la dimensione di 1, 8 o 16 bit. Il numero totale di variabili da dichiarare dipenderà dalle loro dimensioni e dal microcontroller utilizzato, inoltre bisogna tener conto che il PicBasicPlus

riserva 26 indirizzi della RAM per il suo uso interno, e alcune altre variabili temporali per i calcoli delle equazioni complesse. "DIM" può essere utilizzato anche per definire parti di altre variabili, e