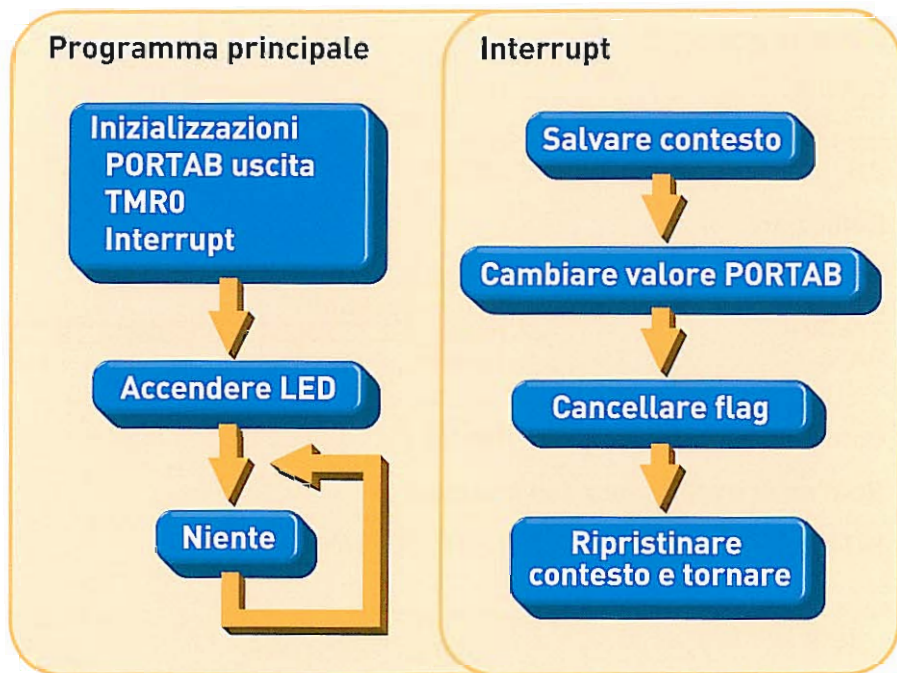


sul pin 1 della Porta B ogni volta che si produce l'overflow del TMR0. Nel programma principale dovremo solo realizzare le inizializzazioni pertinenti, come quelle della porta B, il TMR0, l'assegnazione delle abilitazioni di interrupt e dare un valore iniziale al LED posto sulla porta B. Si continuerà entrando in un ciclo infinito senza fare nulla, sino a quando si produce l'overflow del TMR0 e si entra nella subroutine di interrupt.

In essa si cambierà il valore della porta B, cioè, se il LED era acceso verrà spento e viceversa. Prima sarà necessario salvare l'ambiente di lavoro, che verrà ripristinato appena prima di tornare. Il flag dell'interrupt dovrà essere impostato a 0 prima di uscire.

Il programma continuerà quindi nel ciclo del programma principale, e ogni volta che il TMR0 andrà in overflow, entrerà nuovamente nella routine di servizio di interrupt, cambiando continuamente il valore del pin della porta B. Nella figura si può vedere il programma proposto risolto con istruzioni BASIC. Non è necessario determinare la causa dell'interrupt, perché è stato abilitato solo quello del TMR0.

Notate che, anche se le istruzioni BASIC ci permettono di fare le cose in modo più semplice rispetto all'assembler, quando si lavora con gli interrupt è ugualmente necessario conoscere la funzione di ognuno dei bit di un dispositivo, dato che si lavora con esso come si farebbe con il codice macchina.



Organigramma del programma principale e dell'interrupt.

```

File Edit Compile Options Help
[Icons]
ON INTERRUPT INTER 'Quando si genera l'interrupt si salterà a INTER
SYMBOL TOIE INTCON.5 'Bit di abilitazione dell'interrupt di THRO
SYMBOL TOIF INTCON.2 'Bit di segnalazione di overflow di THRO
SYMBOL GIE INTCON.7 'Bit di abilitazione globale di interrupt
SYMBOL LED PORTB.1

goto MAIN 'Salto al programma principale

INTER: CONTEXT SAVE 'Si salva il contesto
PORTB=PORTB^1 'Si cambia il valore della PortaB
TOIF=0 'Si cancella il flag
CONTEXT RESTORE 'Si ripristina il contesto e si ritorna dall'interru

MAIN: TRISB=%00000000 'PortaB come uscita
OPTION_REG=%00000111 'Configurazione THRO
TMR0=0 'Inizializzazione THRO
TOIE=1 'Abilitazione interrupt THRO
GIE=1 'Abilitazione globale di interrupt
HIGH LED 'Valore iniziale di LED
CICLO: goto CICLO 'Ciclo infinito

PICBASIC PLUS COMPILED OK. 41 Words used
26 Variables used in the DEFAULT 16F84 from a possible 68
  
```

Programma proposto dopo essere passati a istruzioni in PicBasicPlus.