

**La guida di VISUAL PARSIC®.3.41**  
For Windows® 3.1/95/98/XP/NT/2000

**Preliminare non definitivo**

**Compilatore grafico per microPIC di Microchip®**

## **Tutorial**

**If you have any questions please feel free to contact us:  
In caso di necessita' potete contattarci ai seguenti recapiti:**

**Germany :**

Swen Gosh +49 (0)4121/22039  
e-mail : Swen.Gosh@t-online.de  
www. parsic.de

**Italy:**

Parsic Italia +39 0544 927468 +39 340 2455873  
Fax +39 178 6040 078  
e-mail : parsicitalia@libero.it  
www. parsicitalia.it

© Copyright 2001-2003

Parsic © V3.41

Copyright © 2001 Swen Gosh Nordestrasse, 23  
Elmshorn Germany European Community

All Rights Reserved

No part of this publication may be reproduced,stored in a retrival system,or transmitted,in any form or by any means,electronic,mechanical,photocopying,recording,scanning,digitizing,or otherwise,without the prior consense of **Parsic Swen Gosh** .

Windows is a registrered trademark of Microsoft Corporation.

Microchip is a registrered trademark of Microchip coporation.

## **Introduzione.**

### **Compilatore grafico per microprocessori PIC di Microchip**

Visual Parsic è un assembler grafico che permette di gestire in modo automatico la compilazione di un certo numero di ingressi analogici (variabili con continuità), I/O digitali (variabili a due valori). La limitazione nel numero di I/O gestibili dipende dal tipo di microprocessore impiegato. Il software permette di configurare blocchi logici software secondo le necessità del progettista, considerando, però, che il programma complessivamente realizzato non superi la capacità di memoria del microprocessore adottato.

E' opportuno, prima di procedere al progetto circuitale definitivo, realizzare uno schema elettrico di principio per avere l'idea, più accurata possibile, della quantità di blocchi logici necessari da impiegare nel progetto e, da qui, procedere alla scelta del microprocessore.

### **Operazioni pratiche per lo sviluppo di un progetto.**

Dopo l'installazione del programma sul vostro PC :

- 1 predisporre i **Settings** ;
- 2 procedere alla stesura del progetto impiegando i blocchi funzionali ;
- 3 procedere alla simulazione di funzionamento del circuito;
- 4 salvare il file sorgente **.PIC** (schema elettrico) ed il file di testo **.ASM**;
- 5 conversione del file sorgente **.ASM** in file **HEX**, utilizzando MPASM;
- 6 programmazione del microprocessore, impiegando la funzione **Build**.

### **Requisiti di sistema**

Parsic richiede i seguenti requisiti minimi di sistema:

- computer IBM-compatibile 486 o modello superiore;
- Windows 3.1/95/98/NT/2000;
- memoria tipo harddisk con almeno 10 Mbyte di spazio libero;
- scheda video con risoluzione grafica minima di 800x600;
- mouse doppio tasto.

### **Parsic versione Demo.**

La versione Demo di Visual Parsic consente lo sviluppo di qualunque schema elettrico. L'utente può valutare le potenzialità di Visual Parsic, ma non potrà accedere alle seguenti funzioni:

- conversione dello schema elettrico in formato testo ASM;
- salvare il file sorgente;
- stampare lo schema elettrico.

## **Parsic versione V3.xx**

La versione esecutiva di Visual Parsic consente di sviluppare i progetti senza alcuna limitazione (versione Home ed Advanced) ma, per funzionare correttamente, è necessario che l'utilizzatore sia in possesso del numero di codice operativo di abilitazione e della chiave hardware fornita assieme al CD di Visual Parsic. Il codice operativo e la chiave hardware vengono forniti all'acquisto del CD contenente il programma.

Visual Parsic si installa nel computer per mezzo di procedure automatiche di setup. La concessione delle licenze d'uso è rigorosa e le copie non autorizzate di Parsic ( duplicazione dei programmi) vengono perseguite a termine di Legge (DPR 518 del 29 dicembre 1992) . I dati identificativi delle licenze di concessione d'uso del programma (installati sul computer) , devono corrispondere ai dati identificativi riportati sui programmi originali (supporto CD) e relativa licenza sul supporto cartaceo. Eventuali discordanze rendono il programma privo di licenza (copia abusiva ). Gli unici documenti identificativi del programma riconosciuti dal Distributore Parsic Italia sono:

- 1. il codice CD riportato sulla licenza;**
- 2. il codice impresso a marcatura laser (mascherato) riportato sul supporto CD;**
- 3. il codice personale della licenza, inviato all'utente,all'atto dell'acquisto di Parsic;**
- 4. il codice della chiave hardware rilasciato all'utente.**

Per ogni tipo di assistenza, richiesta a Parsic Italia, l'utente dovrà fare riferimento ai codici personali rilasciati nella concessione d'uso. In caso di dubbia provenienza delle richieste,Parsic Italia non fornirà alcuna assistenza tecnica .

## Procedura per l'installazione di Visual Parsic

Per procedere all'installazione di Visual Parsic 3.41 è necessario preventivamente figurare come cliente registrato e disporre della chiave hardware/software .

La registrazione comporta che vengano comunicati:

- Per le licenze **Advance** il nome dell'**Azienda** o della **Scuola** intestataria della licenza ;
- Per le licenze **Home** il nome dell'intestatario ;
- Nome e Cognome dell'intestatario della licenza o di un referente nel caso di licenze in multiutenza o Scuola/Ente,la partita iva e/o codice fiscale;
- Città di residenza o sede dell'Azienda o della Scuola.

In base a queste informazioni, al tipo di licenza ( Advanced, Home,Scuola/Ente, Education) e al numero di copie acquistato viene generato un Codice Utente.

N.B. I dati comunicati a PARSIC ITALIA per la registrazione verranno inseriti in un archivio elettronico **non pubblico** e trattati nel rispetto della legge 675/96 sulla tutela dei dati personali.Parsic Italia si impegna,comunque, a non comunicare a terzi tali dati.



**La redistribuzione o rivendita dei codici di registrazione è proibita.**

**I dati di registrazione verranno memorizzati all'interno dei documenti creati con il software Visual Parsic e potranno essere visualizzati in qualsiasi momento,anche dopo la distribuzione a terzi.**

## **Installazione .**

### **Avvertenza:**

**Visual Parsic può essere installato in un solo computer.L´installazione prevede l´impiego di una chiave hardware che viene collegata dall´utente nella porta parallela del computer, ovvero nella porta HUB dello stesso. La mancata installazione della chiave hardware rende inservibile l´uso del programma. Gli utenti che desiderano installare il programma su piú computer devono richiedere,all´atto dell´acquisto, il duplicato della chiave hardware ( licenze aggiuntive ).**

Installate il CD nel lettore, aprite il contenuto del CdRom,accedere alla cartella Setup V3.41 e fare doppio click sulla stessa.La procedura di installazione è completamente automatica.Verrà creata una cartella PARSIC nella directory Windows\programmi. Ritornate al contenuto del CdRom; cliccate due volte sulla cartella Setup2 ed eseguite la seconda procedura di installazione. Al termine della seconda installazione,se di vostro interesse, potrete riversare ,con la semplice operazione di copia ed incolla di windows, il contenuto delle Applicazioni nella cartella del programma Parsic.

### **MPLAB**

Il setup del programma **MPLAB** è necessario per il completo funzionamento di Parsic. Se MPLAB ( MPASMWIN ) è stato già installato sul vostro computer, non è necessario eseguire questo Setup.

## **Predisporre Visual Parsic.**

### **Menu Imposta**

#### **Personalizza....**

Portare il puntatore del mouse sul menu **Imposta** e poi azionare il pulsante **Personalizza...** Attraverso la finestra di dialogo Personalizza si possono modificare le funzioni operative del programma, elencate qui di seguito:

nel riquadro **Service** :

- **Inserisci il wire-name automaticamente**
- **Cursore mouse centrato**
- **Seleziona senza shift**
- **Connessioni rettangolari.**

Si consiglia di spuntare nell'apposito riquadro tutte queste voci.

#### **Inserisci il wire-name automaticamente :**

attivando questa funzione, le linee di collegamento vengono codificate automaticamente con sigle significative diverse. Le linee codificate Sx.x sono orientate a segnali con singolo bit, quelle codificate Sx sono orientate a segnali con singolo byte.

#### **Cursore mouse centrato :**

attivando questa funzione, potremo spostare il centro del nostro piano di lavoro in un determinato punto dello schermo. Posizionando il cursore del mouse in una determinata area, cliccare con il tasto destro per portare l'area selezionata al centro del piano di lavoro.

#### **Seleziona senza shift :**

quando attivate questa funzione, potrete selezionare gli oggetti o le funzioni accessorie senza l'ausilio del tasto shift.

#### **Connessioni rettangolari:**

attivando questa funzione ( consigliata ) le connessioni delle linee virtuali sono ad angolo retto. Diversamente esse saranno libere di orientarsi su 360°

il riquadro **View** :

- **Visualizza in 3D**
- **Disponi automaticamente le windows**
- **Indica l'interruzione di connessione**

Si consiglia di spuntare i riquadri Disponi... Indica...

<b>Show in 3D</b>	: selezionando la visualizzazione in 3D gli oggetti e le linee di collegamento saranno presentate in grafica in 3D. Se non si ha una buona scheda grafica , ne sconsigliamo l'uso.
<b>Arrange Windows automatically</b>	: selezionando arrange, le dimensioni del piano di lavoro sono disposte automaticamente dal programma.
<b>Arrows at slanted connections</b>	: se non e' stato selezionato <b>connections rectangularly</b> , l'inizio e la fine di ogni linea di collegamento e' marcata da una freccia direzionale di orientamento.

### **Simulazione e Lingua**

- **Refresh**
- **Lingua**

<b>Simulations Refresh ms</b>	: impostando un periodo di tempo compreso tra 1 e 65000 ms si determina il periodo di aggiornamento funzionale del circuito, durante la fase di simulazione.
<b>Lingua</b>	: è la scelta della lingua operativa del programma. <b>Tedesco</b> <b>Inglese</b> <b>Italiano</b>

## Menù Imposta. Microcontroller:

Posizionare il puntatore del mouse sul menu **Imposta** e poi trascinarlo su **MICROCONTROLLER**, apparirà la finestra di dialogo Processor-settings con i seguenti riquadri :

### Configura :

- Watch-Dog –Timer
- Power-On-Timer
- Port B Pull-Up
- Brown Out Detect
- Code Protect

### Oscillatore :

- RC ext
- LP
- XT
- HS
- RC int

### Definisci :

- Ottimizza
- MPLAB ICD
- Clock-frequenza in Hz
- Calibrare l'oscillatore (hex)

Selezionando il pulsante raffigurato qui' di seguito si potrà scegliere uno dei 50 tipi di PIC che il programma e' in grado di assemblare.



Elenco dei PIC selezionabili:

PIC16C54C	PIC16C71	PIC16F84	PIC16C620A	PIC12CE508A
PIC16C55A	PIC16C72	PIC16F84A	PIC16C621A	PIC12CE509A
PIC1656A	PIC16C72A	PIC16F627	PIC16C622A	PIC12CE518
PIC16C57C	PIC16C73	PIC16F628	PIC16CE623	PIC12CE519
PIC16C58B	PIC16C73A	PIC16F873	PIC16CE624	PIC12C671
PIC16C61	PIC16C73B	PIC16F874	PIC16CE625	PIC12C672
PIC16C62B	PIC16C74	PIC16F876	PIC16C711	PIC12CE673
PIC16C63A	PIC16C74A	PIC16F877	PIC16C712	PIC12CE674
PIC16C64A	PIC16C74B		PIC16C715	
PIC16C65B	PIC16C76		PIC16C716	
PIC16C66	PIC16C77			
PIC16C67	PIC16C84			

## Menu imposta . Microcontroller : l'oscillatore.

Nel riquadro **Oscillatore** si imposta il tipo di oscillatore che si vuole impiegare e la sua frequenza di funzionamento :

Oscillator

RC (ext.)

LP

XT

HS

RC (int.)

Clock-frequency in Hz (mi.100)

4000000

Calibration value (hex)

00

<b>RC (ext)</b>	: la frequenza di clock e' determinata dalla rete RC collegata al PIC;
<b>LP</b>	: low-power osc. risuonatori con range fino a 200KHz ;
<b>XT</b>	: quarzi o risuonatori con range da 200 KHz a 4 MHz ;
<b>HS</b>	: quarzi o risuonatori con range da 4 a 20 MHz ;
<b>RC</b>	: oscillatore RC interno (non disponibile su tutti i PIC).
<b>Clock frequency</b>	: la frequenza di clock dell'oscillatore in Hz ;

**Calibration value** : questa opzione e' importante per quei Pic equipaggiati di oscillatore interno e/o che sono dotati di finestra di cancellazione (**Jw-tipe** versione finestrata del PIC) . Il valore (**hex**) da inserire nella opzione **calibration** viene indicato nelle specifiche Microchip. Dopo il reset del Pic, questo valore viene scritto nel **OSCCAL-register** in modo che l'oscillatore interno funzioni il più vicino possibile a 4 MHz. Un parametro elevato del **Calibration value (hex)** produce un corrispondente elevato valore di oscillazione. Questo valore deve essere posto alla fine dei dati contenuti nella ROM, sotto forma di comando e deve essere visto dal microprocessore prima dello start del programma.

## Configurazione .

Konfiguration

Watch-Dog-Timer

Power-On-Timer

PORTB-Pull-Up

Brown-Out-Detect

Code-Protected

<b>Watch-Dog Timer</b>	: letteralmente "cane da guardia".Interviene nel caso il programma si blocchi. L'impulso di reset viene prodotto dopo un periodo di circa 18ms, dall'arresto del programma.
<b>Power On Timer</b>	: se attivato,ritarda per un periodo di 72ms circa, lo start del micro, dopo il ciclo di reset.
<b>PORTB-PullUp</b>	: se attivato, i PortB RB.0....RB.7 vengono collegati internamente alle resistenze di pull-up (+Vdd).
<b>Brown-Out Detect</b>	: attiva il reset del micro in caso che la tensione di alimentazione scenda ad un valore inferiore a 4V. L'attivazione di questa opzione porta ad un aumento del consumo di corrente di circa 500 µA.

**Menu imposta.  
Il Master clear (MCLR).**



<b>MCLR ( Master Clear)</b>	: selezionando <b>MCLR = MCLR</b> le operazioni di reset del micro avvengono collegando elettricamente questo terminale al circuito esterno.
<b>MCLR = I/O Pin, MCLR int.</b>	: il terminale <b>MCLR</b> viene utilizzato come terminale di ingresso ed il <b>MCLR</b> viene collegato internamente al mcp, all'alimentazione +Vdd

**Ottimizza** ( Pic con banchi di memoria RAM, diversi)

Attivando **Ottimizza** Parsic provvede ad inserire le istruzioni dei comandi solo nei banchi di memoria che sono stati selezionati. ( nelle future versioni di Visual Parsic questo comando sarà del tutto automatico )

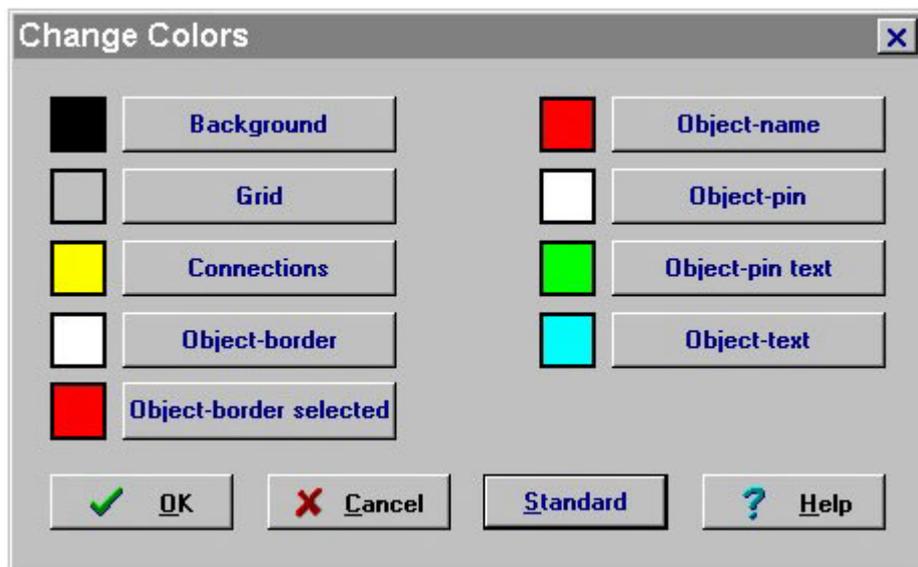
**MPLAB-ICD**

Se MPLAB è attivo vengono liberate le seguenti aree RAM e ROM per l' **Incircuit-Debugger**:  
NOP comand all'indirizzo 0;  
Indirizzo \$70 ed \$1EB nella RAM ( 1EF non é usato da Parsic)  
Le ultime 256 words dell'area ROM non sono usate.

**Menu imposta . Colori: come impostare i colori del piano di lavoro.**

Muovere il puntatore del mouse sul menu **Imposta**, cliccare sul pulsante **Colori**

Nella prima installazione del programma questa operazione è quasi sempre necessaria. Non trascurare questo passaggio, diversamente in alcuni casi non sarà possibile vedere gli oggetti e le linee di collegamento sullo schermo.



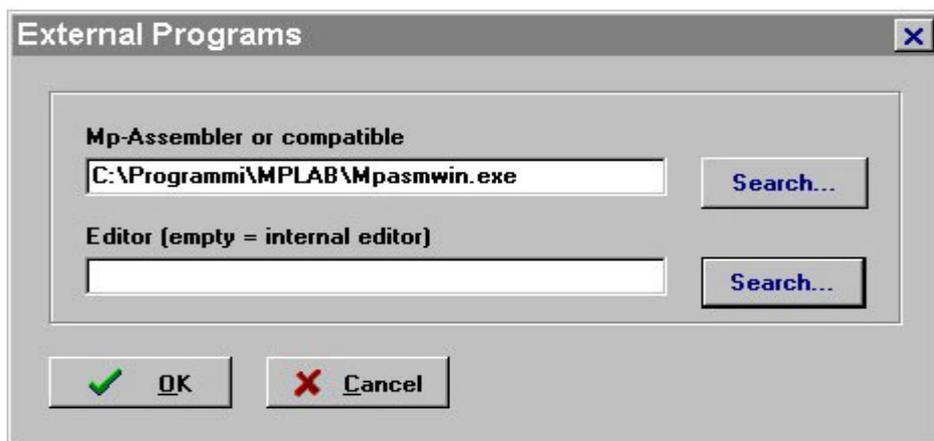
Suggeriamo, inizialmente, di operare con i colori **standard** del sistema .

## Menu Imposta. Programmi.

PARSIC consente di operare con 10 programmi applicativi diversi come, ad esempio, tra questi, **MPLAB**. Portarsi ancora con il puntatore su **Imposta** e poi su **Programmi**. Inserire il nome/percorso del programma **MPLAB** ( **MPASMWIN**) completo della sua estensione, così come raffigurato.

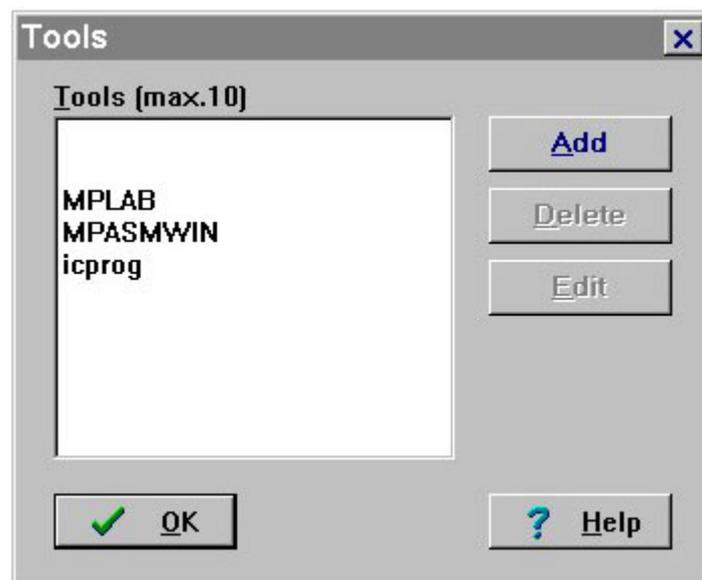
Utilizzare **CERCA**, per individuare nella libreria il percorso che PARSIC deve effettuare per aprire **MPASWIN** o un altro programma simile. Utilizzando questa procedura, al termine di un progetto, dopo l'operazione **SALVA** ( **Save As..**), provare ad attivare la funzione **BUILD..** ( **F10**).

Parsic provvederà a salvare il file sorgente **ASM** del circuito in progetto e lancerà automaticamente **MPASM** . Il programma, tra le altre funzioni, produce un file con estensione **.PJT** utile ad operare con **MPLAB**. Si potrà lanciare **MPLAB** oppure **MPASMWIN** o **ICPROG**, direttamente dalla barra delle applicazioni operando sul pulsante **TOOLS** (collegamenti) .



## Menu Imposta. Collegamenti ad altri programmi : Tools.

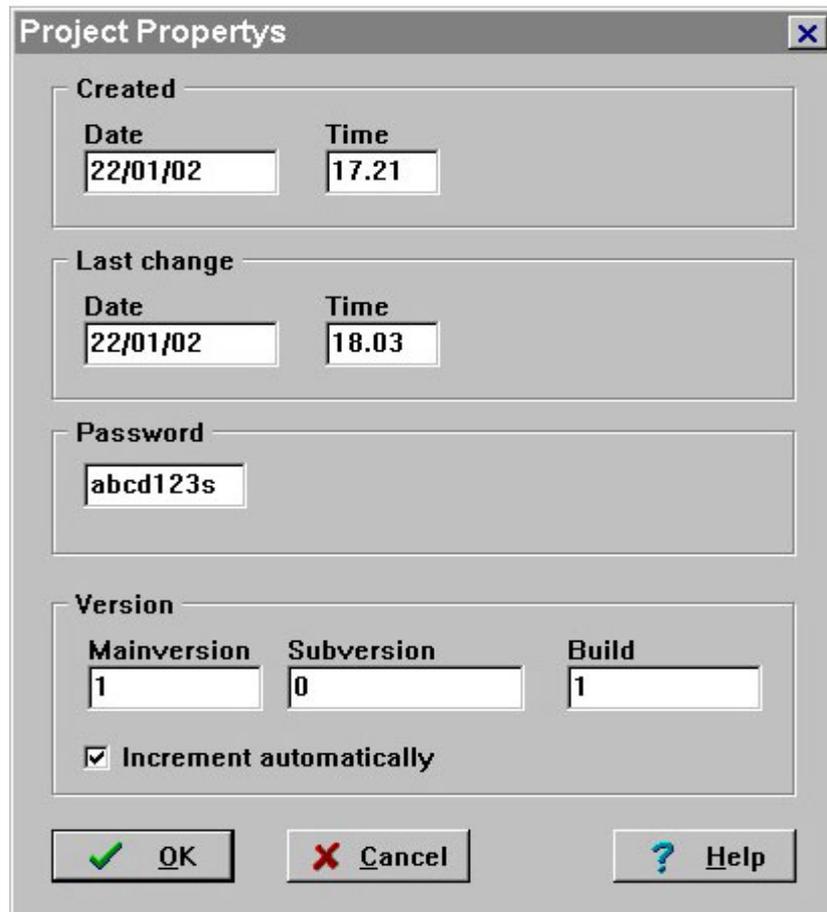
Portare il puntatore del mouse sul menu **Tools** e poi trascinarlo su **Configurazione Tools**, cliccare. All'interno della finestra di dialogo rappresentata in figura, potrete inserire i programmi che più utilizzate durante le operazioni progettuali (max 10).



## Menu Imposta.

### Project Propertys : i dati di progetto.

Spostare il puntatore del mouse sul menu **Imposta** e poi azionare il pulsante **Project**. I dati contenuti in questa maschera sono relativi al progetto che si vuole iniziare. Questi dati saranno aggiornati automaticamente ad ogni nuova sessione del progetto, se nel riquadro **Incrementa automaticamente** verra' spuntata la relativa casella. Inserendo la **Password** il progetto sara' protetto da accessi non autorizzati ( impiego in multiutenza di Parsic).



The image shows a dialog box titled "Project Propertys" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Created**: Contains two input fields: "Date" with the value "22/01/02" and "Time" with the value "17.21".
- Last change**: Contains two input fields: "Date" with the value "22/01/02" and "Time" with the value "18.03".
- Password**: Contains a single input field with the value "abcd123s".
- Version**: Contains three input fields: "Mainversion" with the value "1", "Subversion" with the value "0", and "Build" with the value "1". Below these fields is a checked checkbox labeled "Increment automatically".

At the bottom of the dialog, there are three buttons: "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "Help" (with a blue question mark icon).



## Mouse click.

Gran parte delle attività del programma vengono svolte impiegando il mouse.

Questo accessorio svolge l'importante funzione di accedere alle funzioni del programma con il semplice "click" dei suoi pulsanti. Il puntatore del mouse, secondo la sua posizione, assume tre funzioni diverse:

- selezione di una funzione programma o di un blocco logico;
- selezione di un collegamento circuitale ;
- selezione della modifica di un blocco logico (movimento del b.l. in un punto qualunque dello schema, cancellazione del b.l. , modifica di parametri,ecc.)

La selezione di un menu avviene portando il cursore su una delle risorse che si vuole attivare ( file, Impostazioni,tool, ecc.), mentre la selezione di un blocco logico funzionale dello schema elettrico, si attiva posizionandosi sui pulsanti della toolbar.

In ogni caso, dopo essersi posizionati su una qualsiasi funzione, si dovrà agire sul tasto sinistro del mouse per attivarla.

Operando sui pulsanti della toolbar, selezionata una funzione (tasto sinistro), si dovranno lasciare i tasti del mouse liberi e si dovrà proseguire trascinando l'oggetto nello schema elettrico. Una volta in posizione , cliccare col tasto sinistro del mouse per visualizzarlo. Più avanti, questa manovra verrà descritta dettagliatamente. Il cursore del mouse modifica la propria simbologia in base alla posizione assunta sul blocco logico:

-  portando il cursore sopra un blocco funzionale , attivando l'icona raffigurata a lato, si potranno ottenere le seguenti funzioni :
- mantenendo premuto il tasto sinistro del mouse, il blocco logico può essere spostato in un punto qualunque dello schema elettrico;
- cliccando una sola volta sul blocco logico con il tasto sinistro, e premendo sul tasto delete, esso può essere rimosso (cancellato) dallo schema elettrico;
- portando l'icona sul bordo dell'oggetto e cliccando il tasto destro, si attiva la finestra di dialogo dalla quale è possibile modificare i dati funzionali del blocco logico.

Utilizzare Cntrl +  per spostare le connessioni.



Questa icona si attiva quando il cursore del mouse è posizionato su una delle terminazioni del blocco logico. Mantenendo il tasto sinistro premuto e spostandosi con il cursore, procederemo al collegamento elettrico del terminale; invece, rimanendo sullo stesso terminale , cliccando con il tasto destro, procederemo alla identificazione del terminale, attraverso l'attivazione della finestra di dialogo che apparirà subito dopo la selezione ( **Connessione info** ).

## Uso delle combinazioni dei tasti.

Durante la costruzione dello schema elettrico l'uso combinato dei pulsanti della tastiera facilita ed abbrevia le operazioni di Window. A lato di ogni pulsante funzionale,nella barra dei menu , sono riportate le sequenze dei comandi abbreviati di ogni singola funzione.

## Uso dei pulsanti della Toolbar.

La toolbar si presenta come un complesso di pulsanti, sotto forma di icone, che vengono azionati con il tasto sinistro del mouse, quando il cursore è posizionato su di essi. La toolbar si divide in due gruppi di selezione :



seleziona il gruppo pulsanti della toolbar ;



inserisce o disabilita la **Griglia** di riferimento sullo schermo;



definisce la specifica del collegamento : si consiglia di lasciarlo sempre inserito;



quando azionato avvia il processo di **Simulazione** del circuito in progetto;



Selezione della funzione di **Zoom** in/out dello schema elettrico;



operatori booleani **AND, OR, XOR**;



**Flip-Flop** Set-Reset;



**Timer** o base dei tempi ( clock);



blocco **Monostabile** su evento;



**Contatori** con risoluzione a 8 oppure 16 bit ;



**Schift-register**;



blocco **Selezionatori ingressi analogici o costante digitale**;



blocco comparatore di ingresso digitale **Schmitt - trigger**;



gestione **Tabelle e Soubroutine**;



**De-Multiplexer** ;



## De- Codificatore digitale



funzione **Include** ;



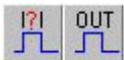
selezione dell'**editor** di commento;



selezione dei livelli di **potenziale di riferimento** ;



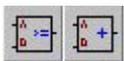
etichetta **Label** ;



condizionatore d'impulso **Impulse-in** , **Impulse-out** ;



monostabile di tipo **One-Shot** ;



blocco **Comparatore** digitale ed **Operatore Matematico** ;



gestione **EEPROM**;



funzione **Limiter** ;



funzioni modulo **Base LCD** e modulo **Testo LCD** ;



selezione della funzione **Sleep** ;



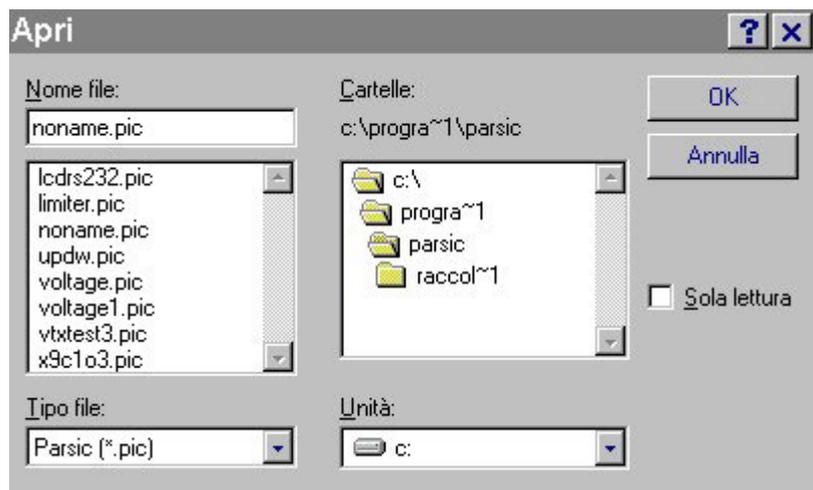
gestione delle linee seriali **RS232 e 485**.

La descrizione funzionale dei blocchi logici e' spiegata dettagliatamente in avanti.

**Load dei file, selezione dello zoom in-out.  
La Function Plan.**



Per accedere ad un progetto precedentemente memorizzato, portarsi con il puntatore del mouse sul menu **File** e poi scorrere fino al pulsante **Apri...**, cliccare.  
Parsic riconosce i file con estensione **.pic** che, una volta selezionati, li visualizzerà sul display.



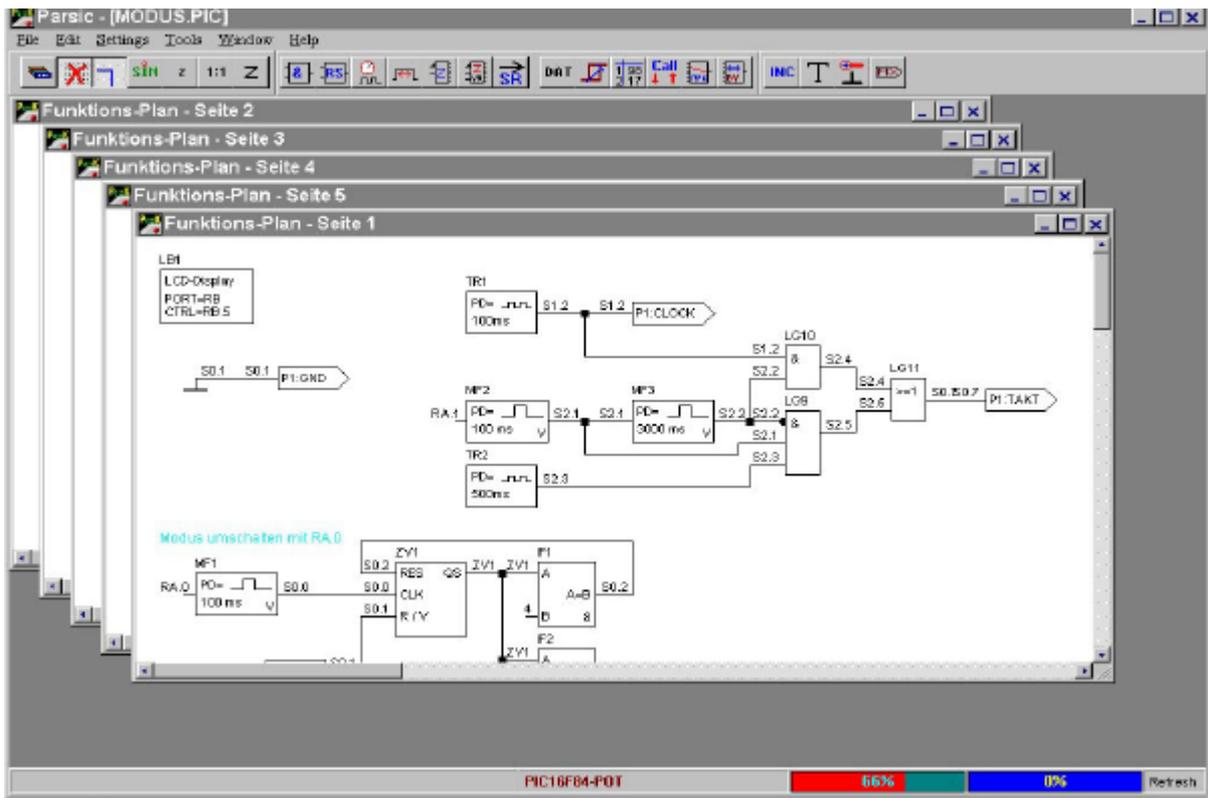
Se lo schema elettrico ha dimensioni molto grandi, tanto da individuare con difficoltà i blocchi circuitali, è bene usare i tasti (**Zoom**) **z** (in) e **Z** (out) . In caso di progetti che richiedono estensioni circuitali molto ampie, si consiglia di usare l'opzione **Seleziona Pagina** del menu **Window** , che permette la distribuzione dello schema elettrico su più fogli schematici. Attivando questa funzione, si raccomanda di utilizzare il blocco funzionale **Etichetta ( label )** , necessario per collegare le terminazioni elettriche tra una pagina e l'altra.

La funzione Etichetta ( label )



## Il menu Window : gestione dei fogli schematici.

Le utilita' del menu **Window** consentono di lavorare su schemi di grande dimensioni, operando su piani di lavoro diversi. Dopo avere aperto il file dimostrativo **Modus.pic**, portare il puntatore del mouse sul menu **Window**, cliccare con il tasto sinistro spostando il puntatore su **Cascade** (Shift + F5). Lo schema elettrico del file Modus.pic sara' distribuito su piu' pagine, come mostrato nella figura seguente:

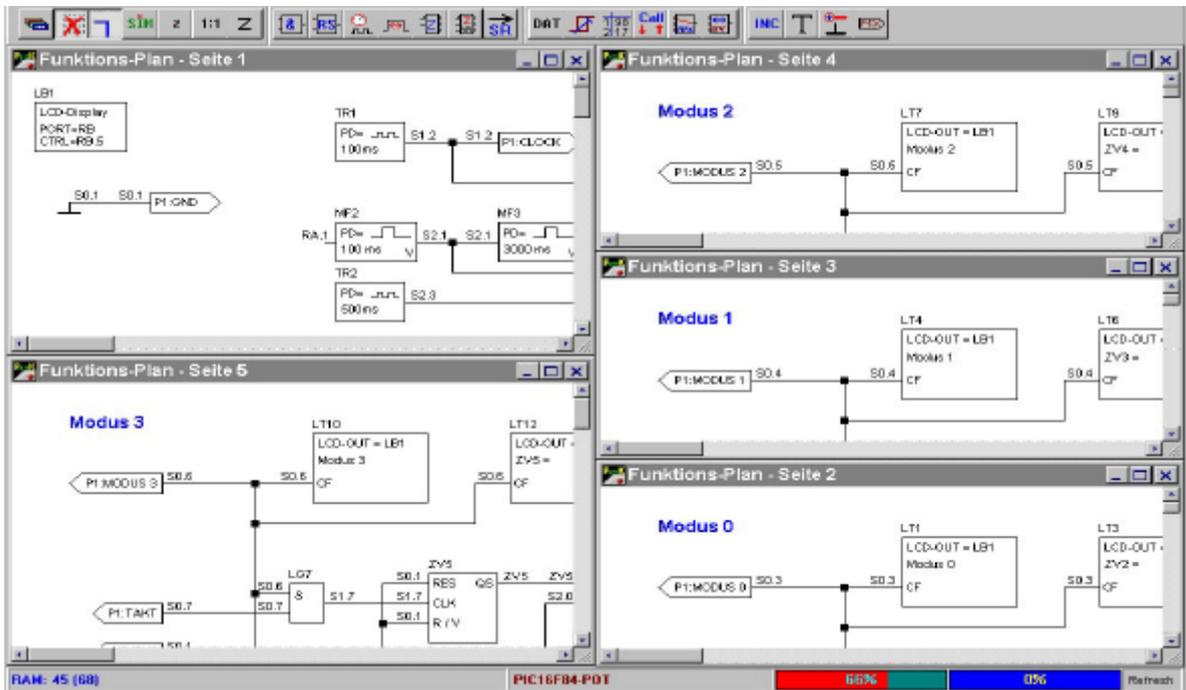


Posizionandosi sul bordo delle **Funktions-Plan**, si potranno visualizzare le pagine come se scorressero l'una su sull'altra.

Le terminazioni di continuita' elettrica, tra le pagine, sono distinte dall' etichetta di identificazione **Label**.



Portandosi di nuovo sul menu **Window** e posizionando il cursore del mouse sul pulsante **Tile** (Shift + F4). gli schemi elettrici si disporranno come nella figura seguente:



I pulsanti **Minimizza ed Ordina** servono a ridurre le pagine ad icone, il primo, mentre il secondo pulsante a disporre in primo piano il piano di lavoro. Per ritornare sulla pagina iniziale, a schermo pieno, è necessario riportare il puntatore del mouse sul menu **Window** e poi selezionare **Ripristina**. Utilizzando la Funktions Plan, **per commutare le pagine**, utilizzate i tastini numerici.

## Selezione della griglia e dei collegamenti virtuali.

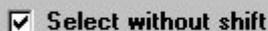


Selezionando la griglia di posizionamento, questa potrebbe essere di aiuto per posizionare ordinatamente i componenti nello schema elettrico. I collegamenti elettrici si dispongono con angolazioni a **90°** se nel menu **Imposta** è stata spuntata la voce **Connessioni rettangolari**. Se non viene selezionata la voce , allora il collegamento elettrico delle linee è libero di orientarsi su 360° : consigliamo il collegamento a 90°.

### Il comando Delete.

Precedentemente, nel paragrafo **mouse click**, e' stato spiegato come procedere alla cancellazione di un oggetto o di una linea di collegamento.

L'azione e' valida se nel menu **Imposta** è stata spuntata la voce **Seleziona senza shift**. Gli oggetti dello schema elettrico si possono cancellare diversamente, selezionando questi tenendo premuto il tasto **shift** della tastiera e premendo contemporaneamente il tasto **sinistro del mouse**. La rimozione si verifica premendo il tasto **delete** oppure portandosi con il cursore del mouse sulla barra dei menu, selezionando il menu **Edita** e poi **Delete**. Consigliamo di mantenere spuntata la voce **Seleziona senza shift**.



Volendo procedere alla cancellazione di vaste aree del circuito elettrico, posizionandosi con il cursore del mouse su un punto periferico dello schema , mantenere il tasto sinistro premuto, poi muoversi con il cursore verso il perimetro opposto dell'area in cui vi trovate.

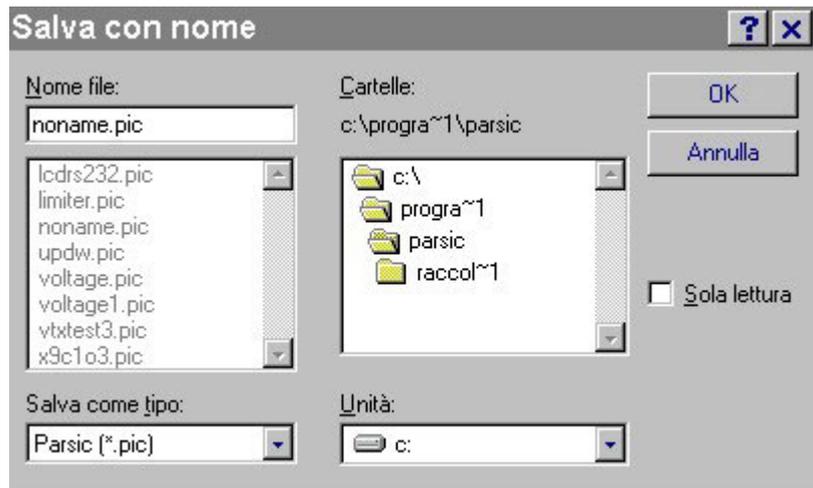
Mano a mano che si procede, gli oggetti che sono contenuti all'interno dell'area tracciata dal cursore del mouse saranno selezionati. Lasciando libero il tasto sinistro del mouse, per eliminare gli oggetti selezionati sarà necessario adoperare il tasto **delete** . Volendo procedere alla pulizia dello schermo, portare il puntatore del mouse sul menù **Edita**, trascinandolo poi sul pulsante **Seleziona tutto**, agire sul tasto **delete** per eliminare tutti gli oggetti dalla schermo.

### Il comando Salva.

Si raccomanda di usare spesso il comando Save durante la fase progettuale. Vi mette al riparo da errori di manovra, involontaria perdita dei dati, disattenzioni....

Per procedere, portando il puntatore del mouse, sul menu **File** selezionare **Salva (Salva con nome..)**. Il nome che si darà al file non potrà essere più lungo di 8 caratteri e deve necessariamente avere l'estensione **.pic** (Parsic non riconosce altre estensioni) .La selezione abbreviata per questa operazione e' **Cntr + S** .

Prendere la buona abitudine di salvare il file sorgente, fin dalla prima fase del vostro lavoro, ossia dalla selezione **Nuovo**. Il figura seguente mostra la finestra di dialogo **Salva**:



### Salvare il file di testo ASM.

Per salvare il file sorgente ASM (file di testo) prodotto da Parsic, portare il puntatore del mouse sul menu **File** e poi selezionare **Salva ASM (Salva ASM con nome.....)**. Il nome che si darà al file non potrà essere più lungo di 8 caratteri e deve avere l'estensione **ASM**. La selezione abbreviata per questa operazione è **Cntr + Q**.

### Conversione del file di testo ASM nel file compilato .HEX

La procedura per ottenere la conversione del file di testo **.ASM** in formato **Intel Hex** è molto semplice. Con Parsic è possibile applicare due metodi per ottenere lo stesso risultato.

**Il primo** consiste nell'avviare **MPASM**, selezionandolo dal menu **Tools**. All'apertura del menu, attivare **MPASM** e procedere nell'elaborazione del file **.ASM**, posizionato nella directory di **Parsic**. Si procede, da questo punto in poi, secondo le istruzioni di **MPASM**.

Il risultato che si ottiene, a video, è quello di una rapida presentazione del processo di traduzione del file **.ASM**, che terminerà con la produzione di una serie di file, così di seguito elencati, che saranno memorizzati nella directory di Parsic:

**nomefile. HEX**  
**nomefile. LST**  
**nomefile. ERR**  
**nomefile. COD**

In caso di errore, il file **.HEX** non sarà prodotto e l'utente dovrà impiegare il file di testo **.ERR** per leggere l'elenco degli errori.

**Il secondo metodo** : dopo aver salvato il file con estensione **.pic**, portarsi col il cursore del mouse sulla barra delle applicazioni, selezionare il menu **File** e poi **Build...** (selezione rapida **F10**)

Si aprirà la finestra di dialogo, **Salva ASM con nome** e si procederà a salvare il file di testo con la stessa procedura vista in precedenza. Completata l'operazione dando **OK**. Si noterà, nell'immediato, che il programma avvierà **MPASM** e che produrrà, in conseguenza, gli stessi file di cui sopra.

Ricordiamo, affinché questa procedura abbia successo, si dovranno seguire le operazioni di set suggerite al paragrafo, **Menu Imposta : Programmi...**

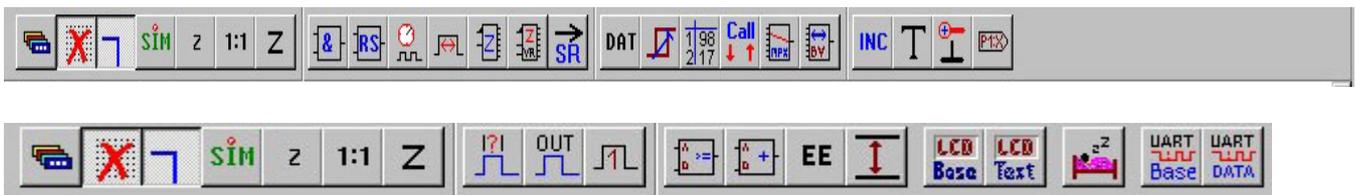
## Il comando SIM.



Prima di attivare la simulazione funzionale dei circuiti, è necessario che venga impostato correttamente il tempo di campionamento dei segnali, che dovrà essere compreso tra **1 e 65000 ms** ( **Simulazione** ). Diversamente, la funzione di simulazione dei circuiti potrebbe non funzionare . Per attivare il processo bisogna portare il cursore del mouse sulla toolbar e azionare il pulsante **SIM**. Le linee di collegamento elettrico virtuale si attiveranno, cambiando tonalità di colore, e si potranno notare le variazioni dello stato logico dei dispositivi . Durante la fase di simulazione si può intervenire sui componenti logici, forzando il livello dei segnali in circolo per mezzo del puntatore mouse, posizionandosi **direttamente all'ingresso** delle terminazioni dei blocchi logici. Se durante il processo di simulazione si attivano i messaggi di errore, il processo verrà interrotto : procedete alla correzione dei circuiti, secondo le indicazioni che verranno date.

## Costruzione dello schema a blocchi.

L' esempio pratico che segue, è utile a comprendere come procedere alla costruzione di uno schema logico funzionale. La **toolbar** consiste in un complesso di pulsanti con la forma di icone , che rappresentano i **blocchi logici funzionali** del programma. I pulsanti, quando azionati, modificano il loro aspetto cambiando colore. Il programma **non ammette** la selezione contemporanea di più blocchi funzione.

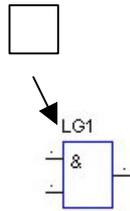


La procedura di costruzione di uno schema funzionale a blocchi è molto semplice e si basa sul funzionamento delle black-box. Posizionarsi con il cursore sulla toolbar e selezionare ( ad esempio) la funzione booleana **AND** .Essa si presenta, appunto, come una scatola nera la cui caratteristica è il segno grafico che la distingue, ed un numero ancora non definito di terminazioni elettriche in-out. Una volta selezionata, questa dovrà essere trascinata e liberata in un punto qualunque dello schermo, così come avviene negli ambienti di programmazione grafica con la funzione **Drag & Drop**. Selezionata la funzione, attaccato al puntatore del mouse, vi apparirà un oggetto di forma geometrica regolare. **Non azionate più i tasti del mouse** ma trascinate l'oggetto in un'area libera dello schermo. Trovandovi nella posizione desiderata, azionare **UNA SOLA VOLTA** il pulsante sinistro del mouse : l'oggetto viene liberato e visualizzato. **Per svincolare il puntatore del mouse, cliccare una volta il tasto destro.**

## Esecuzione pratica:



portare il puntatore mouse sul pulsante  e selezionarlo

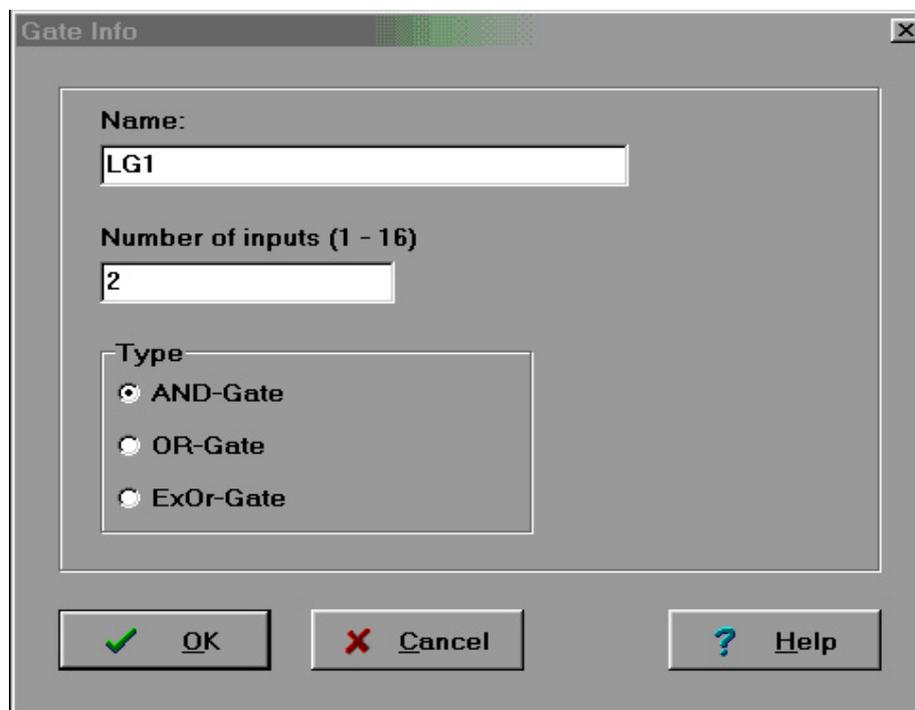


**senza agire piu' sui pulsanti** trascinare l'oggetto in un'area libera dello schermo. Premendo una sola volta il pulsante sinistro del mouse esso verrà liberato. Cliccate sul tasto destro per svincolare il puntatore mouse dalla funzione selezionata.

Portandosi con il puntatore sull'oggetto, esso cambia la geometria della sua icona secondo come viene posizionato. (vedere il paragrafo **mouse click**).



Quando il puntatore e' sopra l'oggetto, provare a cliccare con il tasto destro del mouse .Deve apparire questa finestra di dialogo nella quale si può modificare le caratteristiche elettriche dell'operatore booleano precedentemente selezionato. Si può scegliere un operatore booleano, diverso da quello che adoperemo nell'esempio, e si può modificare il numero dei gate di ingresso ( massimo 8 ) Il gate di uscita non è modificabile.Questa opzione consente di creare funzioni logiche booleane complesse:

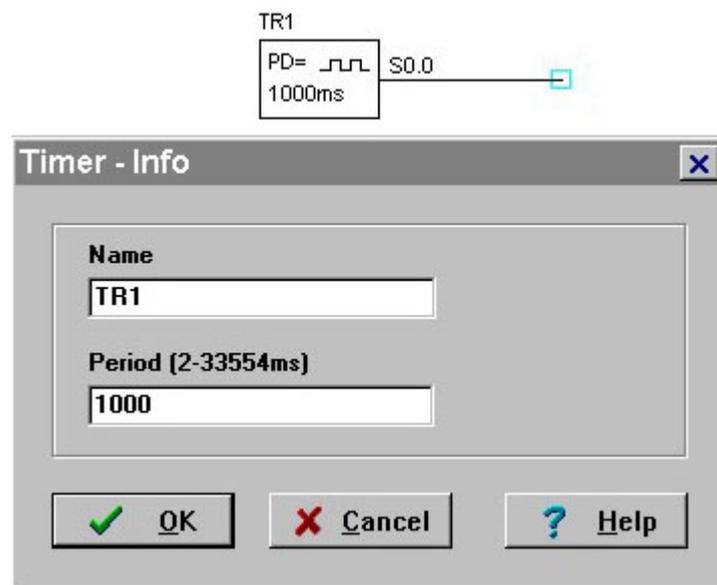


( lasciare inalterata la logica AND e procedete nell'esempio).

In caso di errore volendo eliminare un blocco funzionale o una linea: portarsi con il puntatore del mouse sull'oggetto, cliccare con il tasto sinistro. L'oggetto modificherà il colore del suo contorno geometrico: azionare il tasto delete sulla tastiera del computer per cancellare il dispositivo. Il funzionamento del tasto **delete** della tastiera dipende dalle predisposizioni che prima sono state impostate sul pannello **Imposta**. Controllare che sia spuntata la casella **Seleziona senza shift**

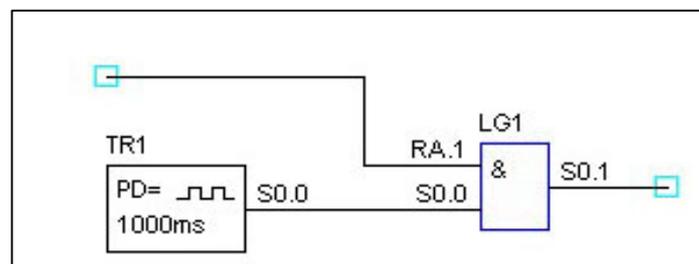
Posizionato l'operatore booleano **AND**, provare a collegarlo ad un **PORT** di ingresso del PIC ed a un generatore di **clock**. Posizionarsi sui pulsanti della toolbar, selezionare il generatore di **clock** (**TIMER**) e posizionarlo nelle vicinanze della logica **AND**.

Ora, provare a cambiare il **set del generatore**, posizionandosi su di esso con il puntatore mouse. Quando appare la finestra di dialogo, di seguito raffigurata, portare il **time base** del timer a **1000ms** (1 sec).

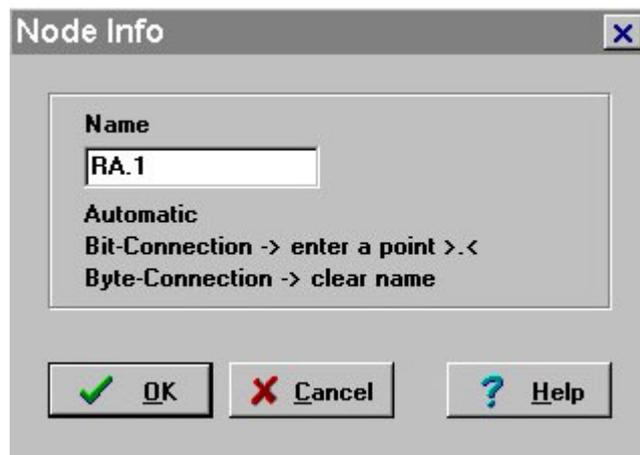


Con il cursore del mouse, collegare il terminale di uscita del timer ad uno dei pin di ingresso dell'operatore booleano. L'altro pin di ingresso della logica **AND**, sarà collegato al gate di ingresso del microchip: **PORT RA.1**

Il circuito, al termine dei collegamenti, dovrà apparire come in figura:

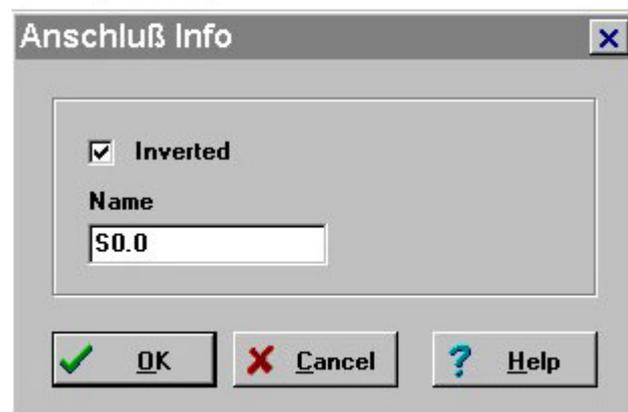
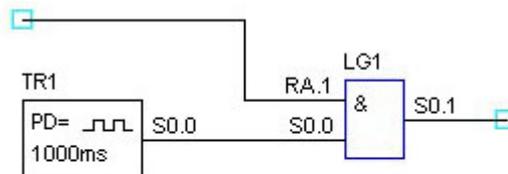


**Attenzione**, per specificare che **LG1**, pin 1, è collegato al **PORT RA.1**, bisogna portare il puntatore del mouse sul terminale 1 della logica **AND**. Quando il puntatore del mouse sarà posizionato sul terminale della logica, azionare il tasto destro. Aprendosi la finestra di dialogo definire l'indirizzo di collegamento del **PORT RA.1**, come mostrato nella figura seguente:

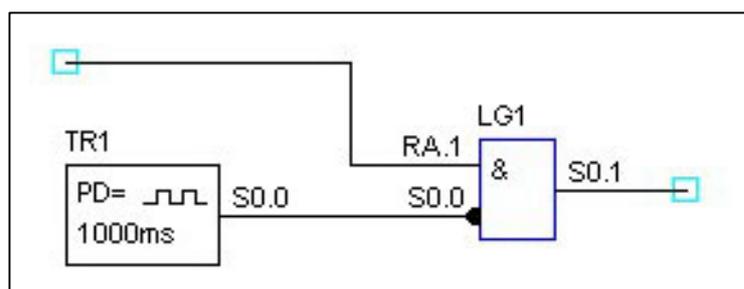


E' importante che tutti i gate d'ingresso o di uscita del micropic siano identificati con i nomi **RA.x**, **RB.x**, **RC.x ecc.**, riconosciuti dal Programma. Accertarsi **sempre** che queste indicazioni siano correttamente posizionate nei terminali dello schema elettrico funzionale, in particolare, prima di lanciare il programma MPASM .

Si puo' collegare un operatore booleano tipo **NOT** all'ingresso o all'uscita di uno qualsiasi degli oggetti logici utilizzati da PARSIC: ecco un esempio. Portare il puntatore del mouse sul collegamento di ingresso di **LG1**, **S0.0**, cliccare con il tasto destro, fino a che appare la seguente finestra di dialogo: spuntare **INVERTI**.



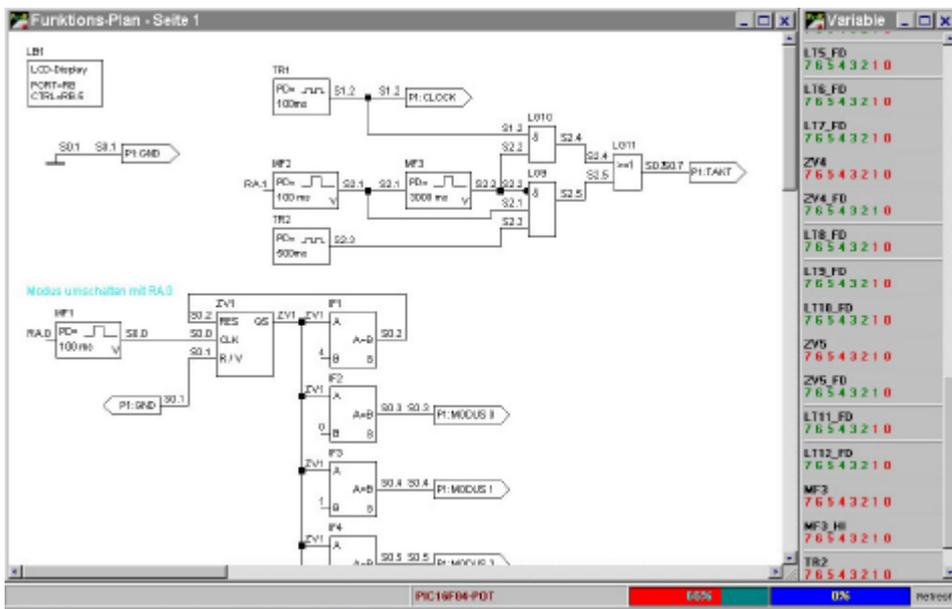
si otterrà il seguente risultato:



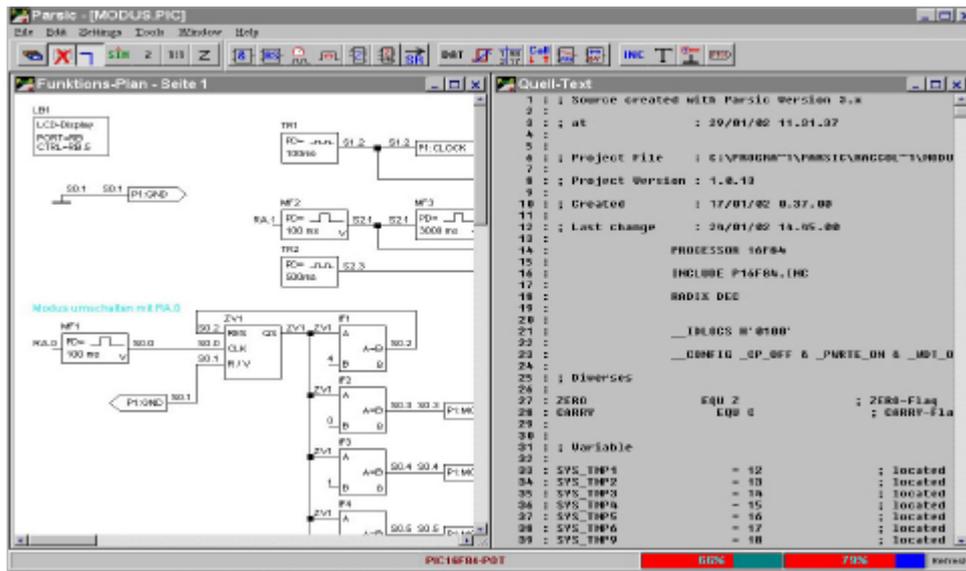
Una volta realizzato questo semplice circuito logico, si può procedere al collaudo funzionale dello stesso. Portarsi con il puntatore sulla barra delle applicazioni, premere il tasto **SIM**. Si procederà, alla **simulazione funzionale del circuito**. Nel caso siano stati commessi degli errori nei collegamenti o di impostazione dei blocchi logici funzionali, vi appariranno alcuni **messaggi di errore**.



Il menu **Window** contiene due funzioni utili per seguire la compilazione del sorgente assembler, mano a mano che si procede nello sviluppo dello schema elettrico. I due tasti funzionali sono **Verifica Variabile e Sorgente ASM**. Quando attive, alla destra dello schermo compaiono due documenti: il primo è una tabella dove sono specificate le variabili utilizzate nel sorgente, il secondo è il file sorgente dello schema elettrico in formato testo. Non tentare la modifica di questi documenti perché il programma non lo consente. Le immagini che seguono, mostrano come appaiono sullo schermo le funzioni **Verifica Variabile e Sorgente ASM**, una volta selezionate:



**Verifica variabile**



## Sorgente ASM

### Come salvare il file sorgente.

Parsic produce tre tipi di file:

il primo con estensione **.PIC**

il secondo con estensione **.PJT**

il terzo con estensione **.ASM**.

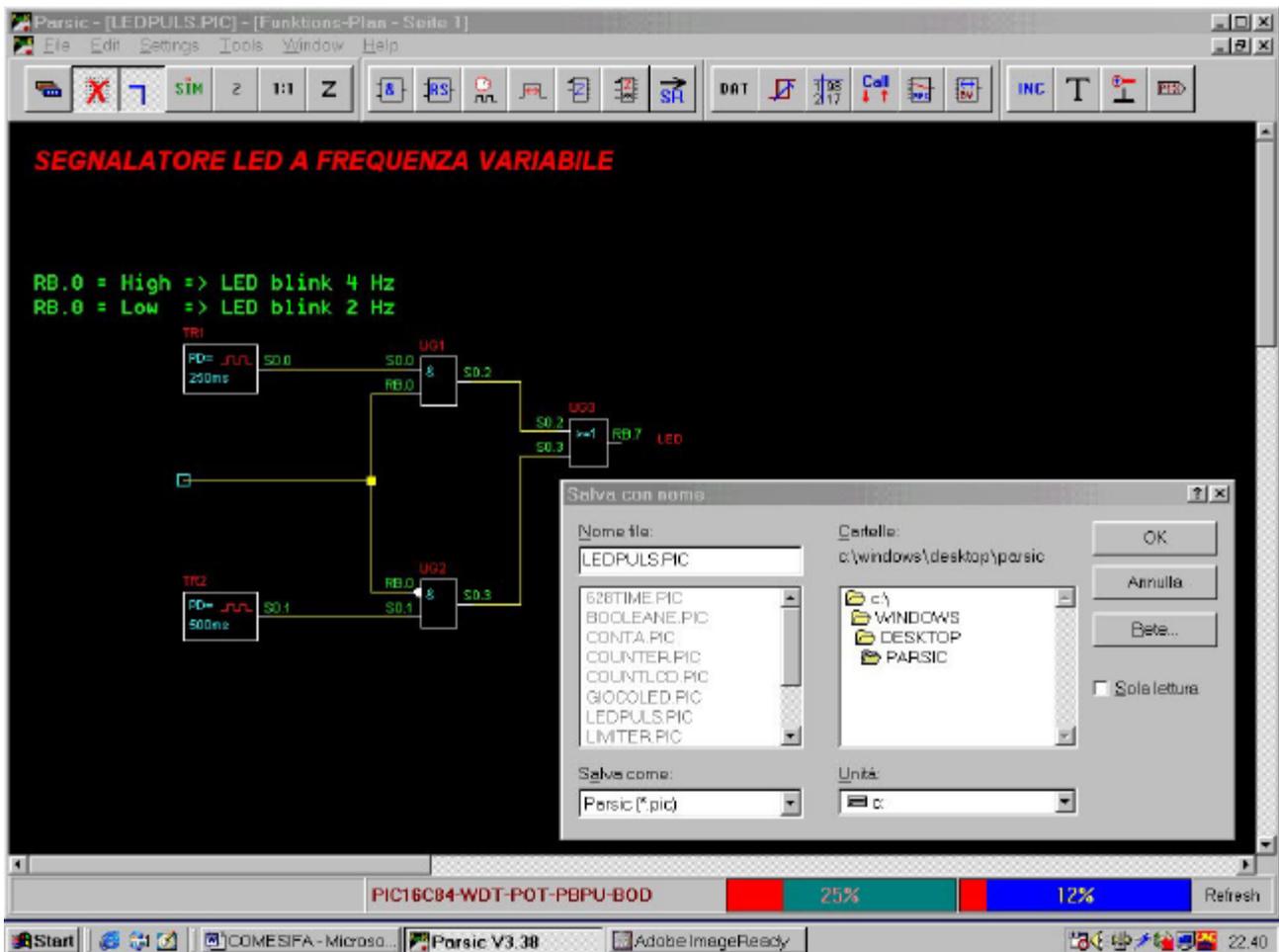
e' il file oggetto del circuito progettato;

e' il file di testo che si utilizzerà con MPLAB;

e' il file di testo che si utilizzerà con MPASMWIN.

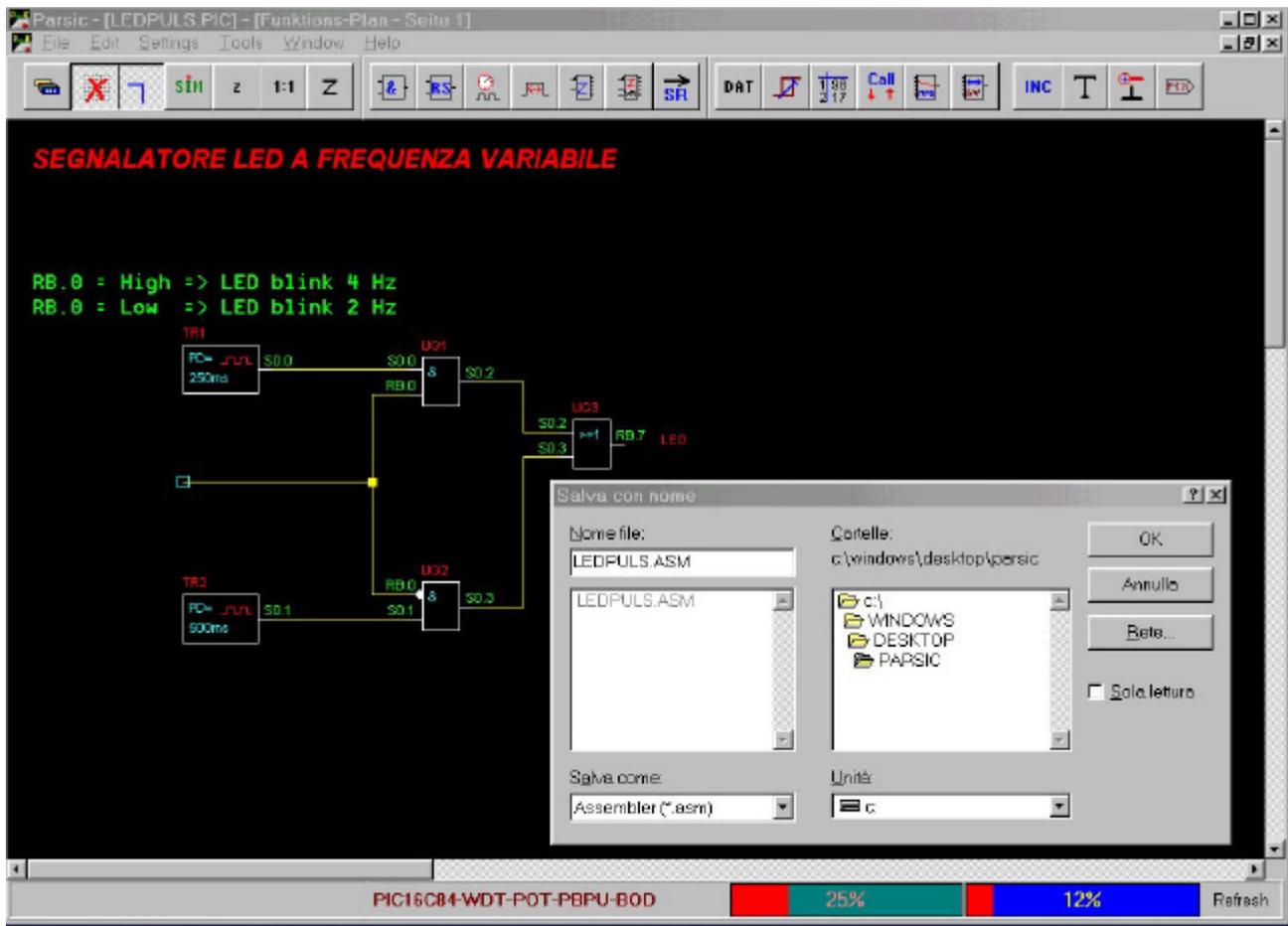
Gli esempi che seguono mostrano sono la sequenza pratica delle operazioni da compiere al termine della fase di progetto.

## La procedura Salva schema



Portare il puntatore del mouse sul menu **File** e poi agire sul pulsante **Salva schema (Salva schema con nome...)** (diversamente, operare con la selezione breve **Cntr + S.**) Nominare il progetto con 8 lettere e l'estensione **.pic**. Dare **OK** per chiudere l'operazione.

## La procedura Salva Asm:



Per eseguire questa operazione, portare il puntatore del mouse sul menu **File** e poi selezionare il pulsante **Salva ASM (Save ASM con nome)**. Diversamente operare con la selezione breve **Cntr. + Q**. Nominare il progetto (lo stesso nome del precedente file) con estensione **.asm**.

Si può utilizzare, per questa procedura, una **selezione più rapida** con lo scopo di produrre, oltre al codice **ASM**, anche il codice **HEX** necessario alla programmazione del pic ed il codice **.pjt** per operare con **MPLAB**.

Portarsi con il puntatore del mouse sulla barra delle applicazioni di window, selezionare **File** e poi agire sul pulsante **BUILD...**

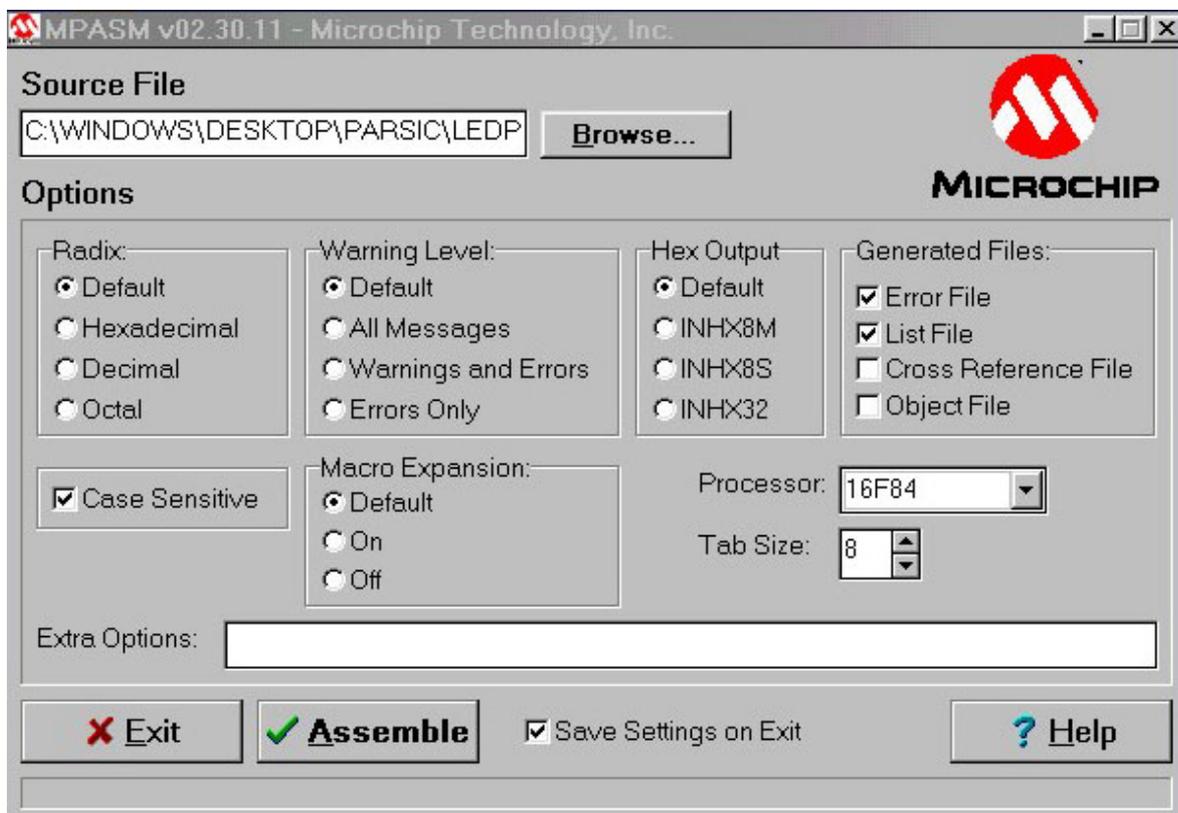
Si apre la stessa finestra di dialogo della figura sopra riportata. **nominare il file**, come spiegato in precedenza. Chiudendo l'operazione con **OK**, si vedrà attivare automaticamente l'**MPASM**. Tutti i file prodotti da **MPASM** saranno salvati nella **directory di Parsic**. Nella figura che segue, si possono osservare come sono stati prodotti i file del progetto **Ledpulse** e la maschera di presentazione di **MPASM**.

## I file prodotti da Parsic ed MPASM

 Ledpulse	6 KB	ASM File
 Ledpulse	10 KB	COD File
 Ledpulse	0 KB	ERR File
 Ledpulse	1 KB	PICpro Document
 Ledpulse	22 KB	LST File
 Ledpulse	1 KB	PJT File
 Ledpulse.wat	1 KB	WAT File

Questo e' l'elenco dei file prodotti durante il processo di conversione del file di testo Ledpulse.ASM  
Se MPASM produce il codice di errore .ERR, il file. HEX ( PICpro Document) non viene prodotto

## La finestra di dialogo di MPASM



Per l'uso corretto di questo programma, seguire le istruzioni di Microchip.

## **Descrizione dei blocchi logici di VISUAL PARSIC.**

Il software permette una grande varietà di utilizzazione, derivante dalla presenza al suo interno di una serie di funzioni implementabili attraverso l'impiego di moduli indipendenti e collegabili tra di loro in modo del tutto generale.

Per collegamento si intende una realizzazione logica di assiemaggio realizzata a programma, immessa da tastiera e mantenuta in memoria dal microprocessore, attraverso gli strumenti software ed hardware normalmente in uso per questo genere di applicazioni. Nella definizione di un progetto possiamo individuare le funzioni descritte di seguito.

### **Blocco logico:**

per blocco logico, dobbiamo intendere una funzione elementare o complessa, disponibile su Visual Parsic, che fornisce una o più uscite digitali, in funzione di uno o più ingressi analogici o digitali e di eventuali parametri impostabili.

### **Variabile Digitale:**

una variabile logica binaria che assume i soli due valori 0 e 1, corrispondenti ai valori logici Vero e Falso o viceversa.

### **Variabile Analogica:**

una variabile numerica che può assumere tutti i valori compresi tra 0% e 100%, con risoluzione pari a quella dell'AD converter del microprocessore utilizzato.

La risoluzione interna delle funzioni logico-aritmetiche è impostabile a 8 oppure 16 bit. Nelle realizzazioni delle funzioni, le variabili analogiche sono sempre viste (nella procedura di calcolo) come valore percentuale del fondo scala (0-100%) mentre, nella visualizzazione, possono essere convertite in qualunque unità numerica. Non sono possibili operazioni matematiche a virgola mobile se non con l'inserimento di opportune macro all'interno della funzione INC.

## Definizione dei blocchi circuitali di Visual Parsic :

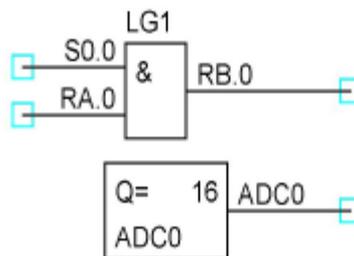
### Identificativi di collegamento utilizzati da Parsic:

ad intervalli di tempo fisso, il sistema esegue una lettura di tutti gli ingressi/uscite digitali effettivamente collegati . Parsic assegna automaticamente un codice numerico ad ogni linea di collegamento. In particolari circostanze quali, ad esempio, nella definizione dei port di ingresso e di uscita dal microprocessore, l'operatore deve forzare l'identificazione delle linee in osservanza delle istruzioni riconosciute dal PIC. Le linee di collegamento elettrico virtuale, vengono indicate con le seguenti diciture e la specifica delle stesse può essere automatica o manuale:

<b>S. xx / Sx</b>	specifica automatica: descrive il codice interno dei segnali assegnato dal programma. <b>La codifica Sx.x</b> e' orientata al bit, <b>Sx</b> al byte;
<b>Rx, GPx</b>	specifica manuale : l'operatore indica quale nome assegnare agli I/O;
<b>ADCx</b>	specifica manuale : l'operatore descrive quale nome assegnare agli input analogici.

Ogni collegamento e' distinto **SEMPRE** da una o piu' lettere seguita/e da uno o due numeri identificativi separati tra loro per mezzo di un punto : **S0.1** , **S7.5**, **S8.2**, **SI.1** , **UR.3**, **ADC0**, **GP4**, **RA.0**, **RC.5**, ecc.

### Esempio di elencazione degli I/O :



Parsic riconosce automaticamente se un gate e' stato programmato come terminale di ingresso oppure di uscita. I terminali del processore che non vengono utilizzati vengono riconosciuti dal programma come port di ingresso. Nota : Microchip consiglia di mantenere a livello logico alto o basso i terminali non utilizzati del processore.

### Specifiche:

<b>RA4</b>	se programmato come terminale di uscita necessita di un resistore di pull-up, in quanto e' configurato nel pic come open collector;
<b>RB0..7</b>	se impiegati come terminali di ingresso, il resistore di pull-up puo' essere programmato internamente;
<b>RB0,4..7</b>	possono essere programmati come trigger oppure interrupt.

### Corrispondenza degli ingressi digitali ad analogico:

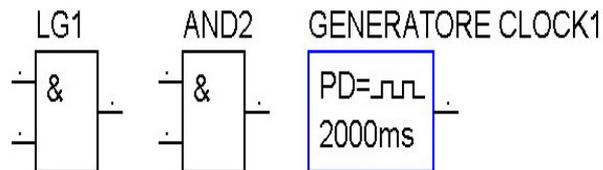
**RA.0 = ADC0**      **RA2 = ADC2**  
**RA.1 = ADC1**      **RA.3 = ADC3**

ecc.

## Denominazione dei blocchi.

I blocchi logici circuitali vengono nominati automaticamente dal software. E' possibile indicare con un nome diverso i blocchi circuitali, dando loro una denominazione diversa. Sono ammessi nomi con 30 caratteri, per ogni identificativo diverso dall'originale.

### Esempio di denominazione dei blocchi circuitali:



Questo è l'elenco dei comandi di Microchip assembler. Parsic accetta questi comandi ma, durante la conversione del file, l'assembler interno di Parsic, può inviare un messaggio di errore.

ADDLW	SUBLW	MOVWF
ANDLW	XORLW	NOP
CALL	ADDWF	RLF
CLRWDT	ANDWF	RRF
GOTO	CLRF	SUBWF
IORLW	CLRW	SWAPF
MOVLW	COMF	XORWF
OPTION	DECF	BCF
RETFIE	DECFSZ	BSF
RETLW	INCF	BTFSC
RETURN	INCFSZ	BTFSS
SLEEP	IORWF	
TRIS	MOVF	



### Operatori booleani AND,NAND,OR, NOR, XOR, XNOR :

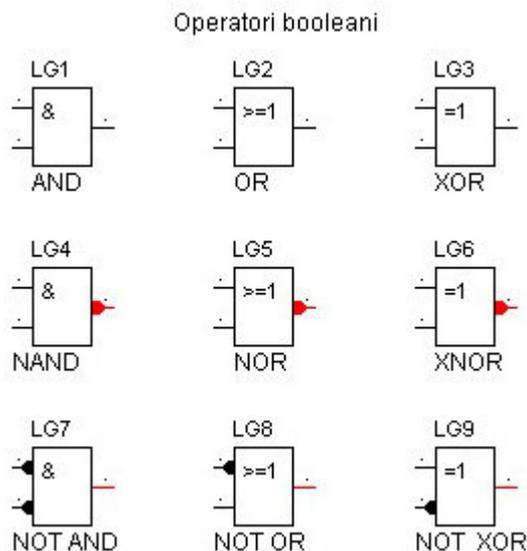
Sulle variabili digitali è possibile operare con variabili booleane di tipo AND,OR, e XOR. Questi blocchi forniscono in uscita anche il valore complementato dando così la possibilità di disporre anche di logiche di tipo NAND,NOR, XNOR.

Le tabelle della verità delle funzioni sono le seguenti :

	ingressi		uscite Sx.x	
OR-NOR :	a	b	or	nor
	0	0	0	1
	0	1	1	0
	1	0	1	0
	1	1	1	0
AND-NAND :			and	nand
	0	0	0	1
	0	1	0	1
	1	0	0	1
	1	1	1	0
XOR-XNOR			xor	xnor
	0	0	0	1
	0	1	1	0
	1	0	1	0
	1	1	0	1

La funzione NOT è liberamente collegabile ai gate di ingresso e uscita delle logiche,consentendo di realizzare funzioni booleane complesse.

### Esempio di predisposizione degli operatori booleani:





## Logiche con operatore matematico.

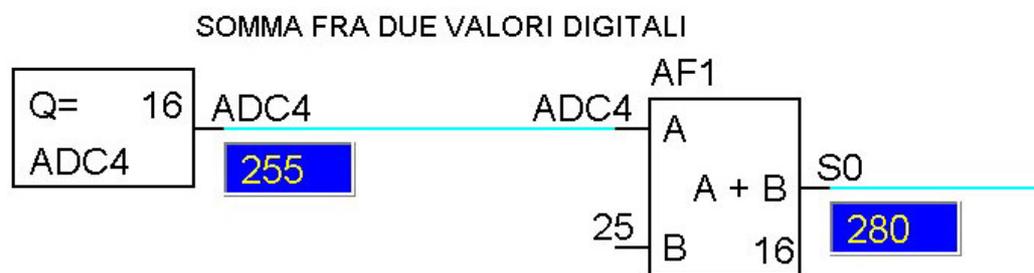
### Blocco sommatore digitale :

Il blocco sommatore esegue la somma di due variabili digitali e ne fornisce all' uscita il risultato :

$$S = a + b \quad \text{oppure} \quad S = k + a \quad (k = \text{costante})$$

La risoluzione puo' essere di 8 bit o 16 bit. Pertanto le variabili ammesse, nel primo caso, sono fino a 255 bit, nel secondo caso fino a 65535 bit.

### Esempio di somma di una variabile analogica e di una costante digitale:



### Blocco sottrattore digitale :

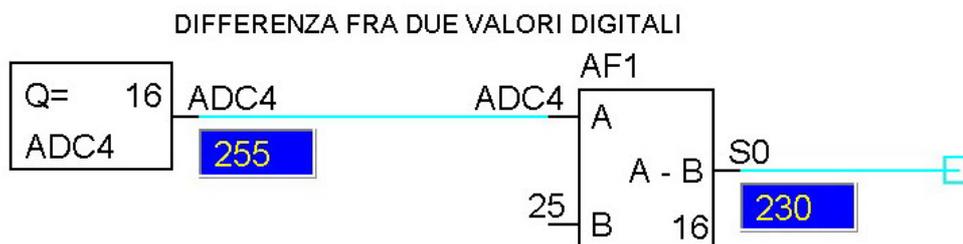
Il blocco sottrattore esegue la differenza di due variabili digitali e ne fornisce all'uscita il risultato.

La funzione e' valida solo se  $a > b$ , diversamente avremo un risultato non congruo :

$$S = 1 \quad \text{se} \quad a \geq 1 \quad b = 0$$

La risoluzione puo' essere di 8 o 16 bit.

### Esempio di differenza tra una variabile analogica ed una costante digitale:





### Blocco divisore digitale:

Esegue la divisione tra due variabili digitali collegate ai suoi ingressi. L'operazione e' valida se  $b > 0$

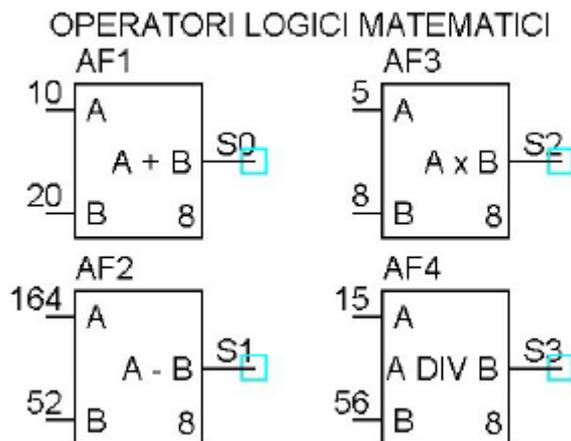
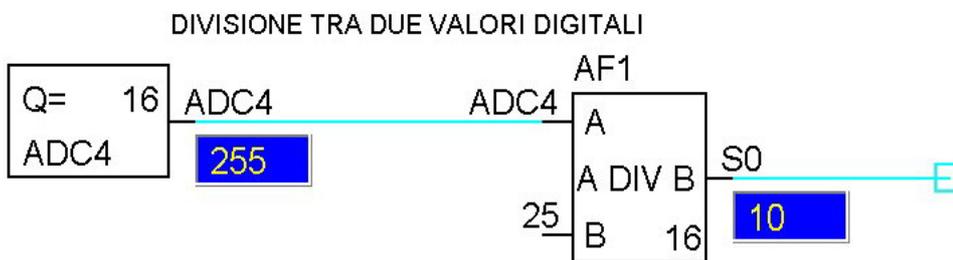
Diversamente avremo :

$S = 255$  se  $a \geq 0$   $b = 0$

$S = 0$  se  $b > a$

La risoluzione puo' essere di 8 o 16 bit.

### Esempio :





**Blocco moltiplicatore:**

Esegue il prodotto di due variabili digitali collegate ai suoi ingressi. Il risultato e' dato da  $S = a * b$   
La risoluzione puo' essere 8 o 16 bit.

**Esempio di moltiplicazione tra due valori digitali :**



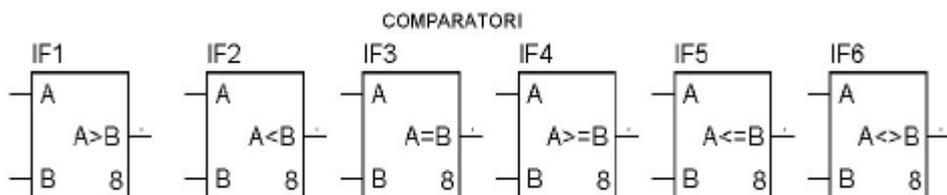
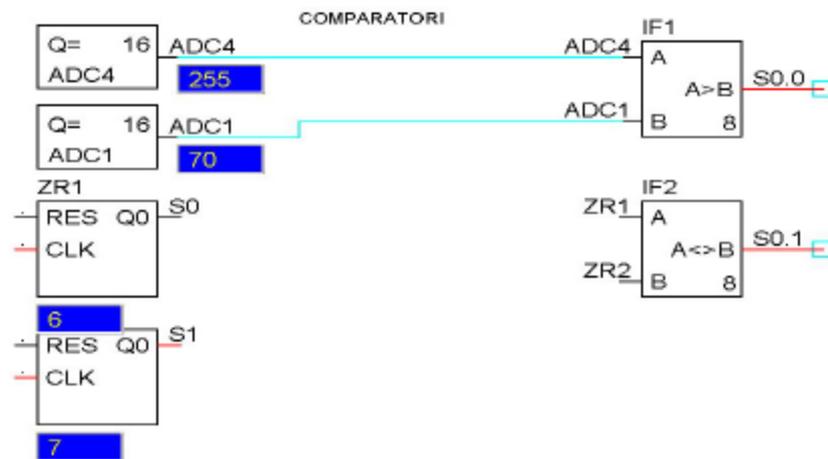


### Comparatore digitale:

Confronta gli ingressi collegati a due qualsiasi variabili digitali e fornisce in uscita un livello logico Vero o Falso secondo l'impostazione data :

S = 1 se	a > b
S = 1 se	a < b
S = 1 se	a ≥ b
S = 1 se	a ≤ b
S = 1 se	a = b
S = 1 se	a <> b

### Esempio di collegamento di blocchi comparatori :





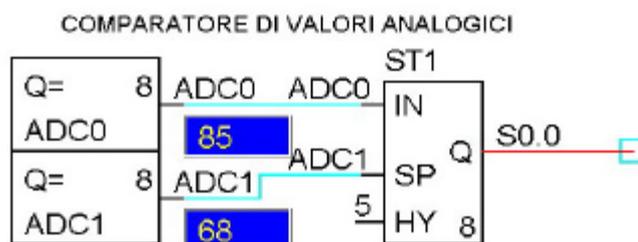
## Comparatore ingresso analogico ( Schmitt - trigger ) :

Confronta i due ingressi collegati a due qualsiasi variabili analogiche e fornisce in uscita un livello logico **Vero o Falso** secondo l'impostazione data. Inoltre e' prevista una isteresi parametrizzabile che evita instabilita' al circuito. Il valore dell'isteresi viene sottratto ad uno dei due ingressi ( **IN** ), in base allo stato del comparatore prima di fare il confronto, comportandosi in tal modo come un'area di banda morta. Per una corretta definizione dei valori di intervento dei comparatori e' da tenere presente che il valore **HY** si sottrae a uno dei due ingressi. Pertanto, per cambiare di stato, l'ingresso **IN** deve modificarsi in modo da differire dall'altro ingresso, meno il valore dell'isteresi:

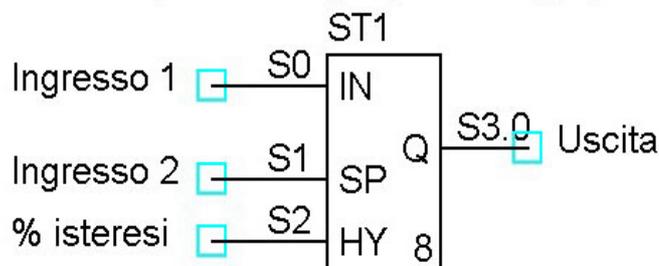
**IN** = variabile ingresso analogico ( - HY )  
**SP** = k (costante ) analogica o digitale  
**HY** = isteresi  
**Q** = uscita blocco comparatore

Il valore del contenuto del comparatore e' visualizzabile a video.

## Esempio di un possibile collegamento del blocco comparatore analogico ( Schmitt- trigger )



Comparatore analogico ( Schmitt- trigger )



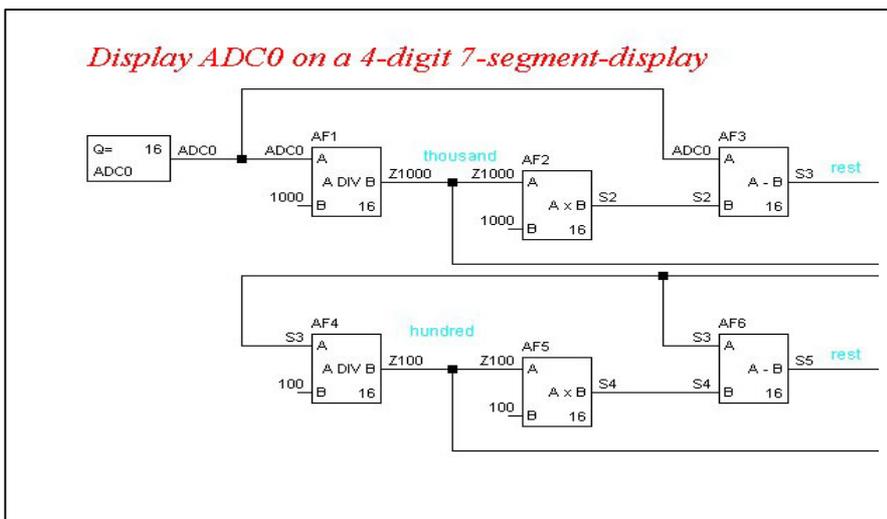
N.B. lo Schmitt-trigger, in alcune applicazioni può essere usato convenientemente per la comparazione di variabili digitali quali, ad esempio, quelle in uscita da due contatori 16 bit up/dwn.

**Selezionatore di ingressi o costante digitale :**

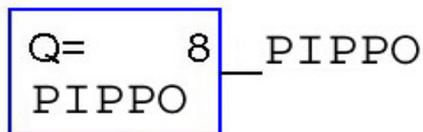
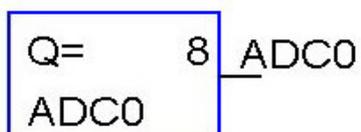
E' associato ad un ingresso analogico con risoluzione a 8 oppure 16 bit. oppure puo' essere utilizzato come costante digitale. Nel caso si desideri utilizzare il blocco DAT come costante digitale, basta semplicemente nominarlo con una sigla qualunque, ad esempio: Code1, Via3, Pippo1, ecc..

Il range dei valori sono compresi tra **0 - 255** per risoluzioni ad **8 bit**  
 " " " " " **0- 65535** per risoluzioni a **16 bit.**  
 Il valore del contenuto del selezionatore e' visualizzabile a video.

**Esempio applicativo di una conversione AD :**



AD CONVERTER O COSTANTE DIGITALE



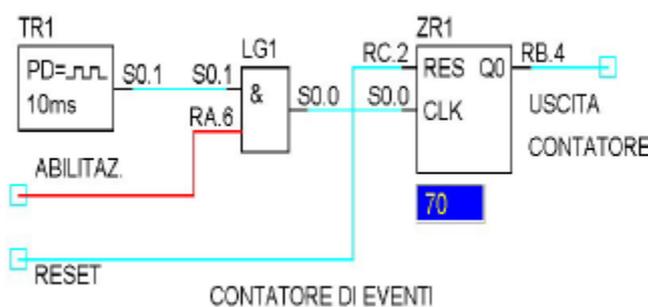


## Conteggio di eventi:

con questo blocco e' possibile contare le variazioni di stato di una qualsiasi variabile digitale uscita da un'altro blocco o applicata al gate di ingresso del microprocessore.

E' possibile visualizzare il valore conteggiato e forzarlo a zero attraverso l'ingresso digitale di reset. Il ritmo di scansione dipende dal segnale applicato all'ingresso, compatibile con il tipo di microprocessore utilizzato. Il valore del contenuto del contatore e' visualizzabile a video, con risoluzione **8 bit**.

## Esempio applicativo del blocco contatore 8 bit :

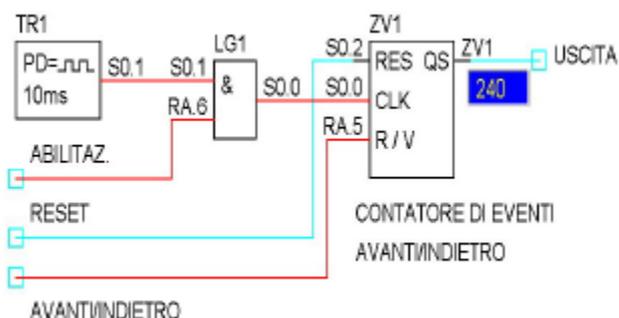


## Blocco conteggio di eventi avanti/indietro:

questo blocco permette di contare le variazioni di stato di una qualsiasi variabile digitale uscita da un'altro blocco o applicata al gate di ingresso del microprocessore.

Il conteggio puo' avvenire sia in avanti, in modo incrementale, oppure all'indietro in modo decrementale. Il contenuto del contatore e' visualizzabile su video ed e' possibile forzarlo a zero attraverso l'ingresso digitale di reset. Il ritmo di scansione dipende dal segnale applicato all'ingresso, compatibile con il tipo di microprocessore utilizzato. Il valore del contenuto del contatore e' visualizzabile a video con risoluzione a **8** oppure **16 bit**.

## Esempio applicativo del blocco contatore avanti - indietro :

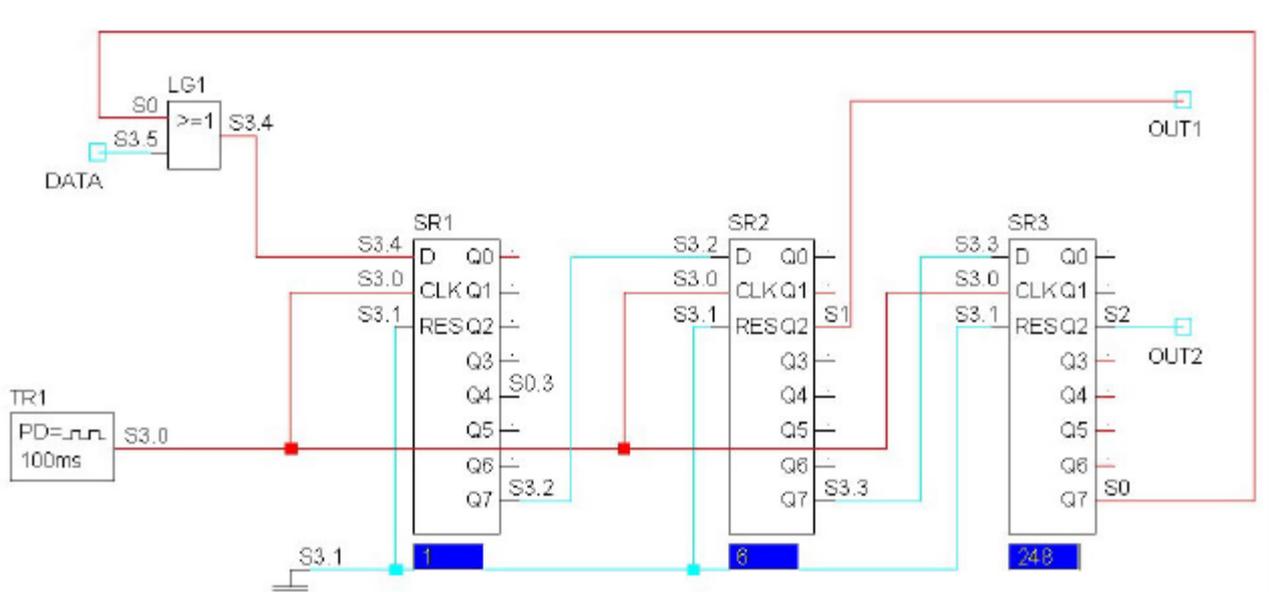




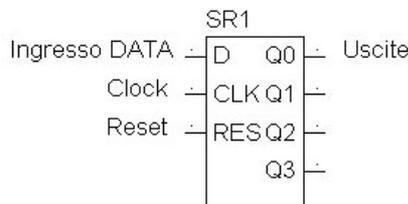
### Blocco schift register :

questo blocco permette di spostare lo stato logico di una variabile digitale, uscita da un'altro blocco, e di trasferirla sulle uscite dello schift register, in relazione alla frequenza di **clock** applicata al suo ingresso **Data**. L'ingresso **Reset** deve essere posto a livello logico zero durante la fase di schift. Se posto a livello alto il contatore viene azzerato. Lo schift register trova la sua particolare applicazione nelle operazioni di misura e conteggio nel dominio di tempo, nell'uso dei sequenziatori logici (robotica) .

### Esempio applicativo:



### SCHIFT REGISTER

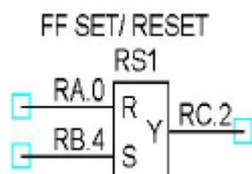




## Flip-Flop set-reset :

Sulle variabili digitali e' possibile operare con delle unità di memoria tipo set-reset. La funzione set e' attivata da un livello attivo sull'ingresso **S** mentre la funzione di reset e' attivata da un livello attivo sull'ingresso **R**. Nello stato **Set** si ha un'uscita **Y** a **livello alto**, mentre nello stato **Reset** si avra' l'esatto opposto. Quando gli ingressi sono **disattivi** lo stato del **flip-flop** viene mantenuto in quello **iniziale**.

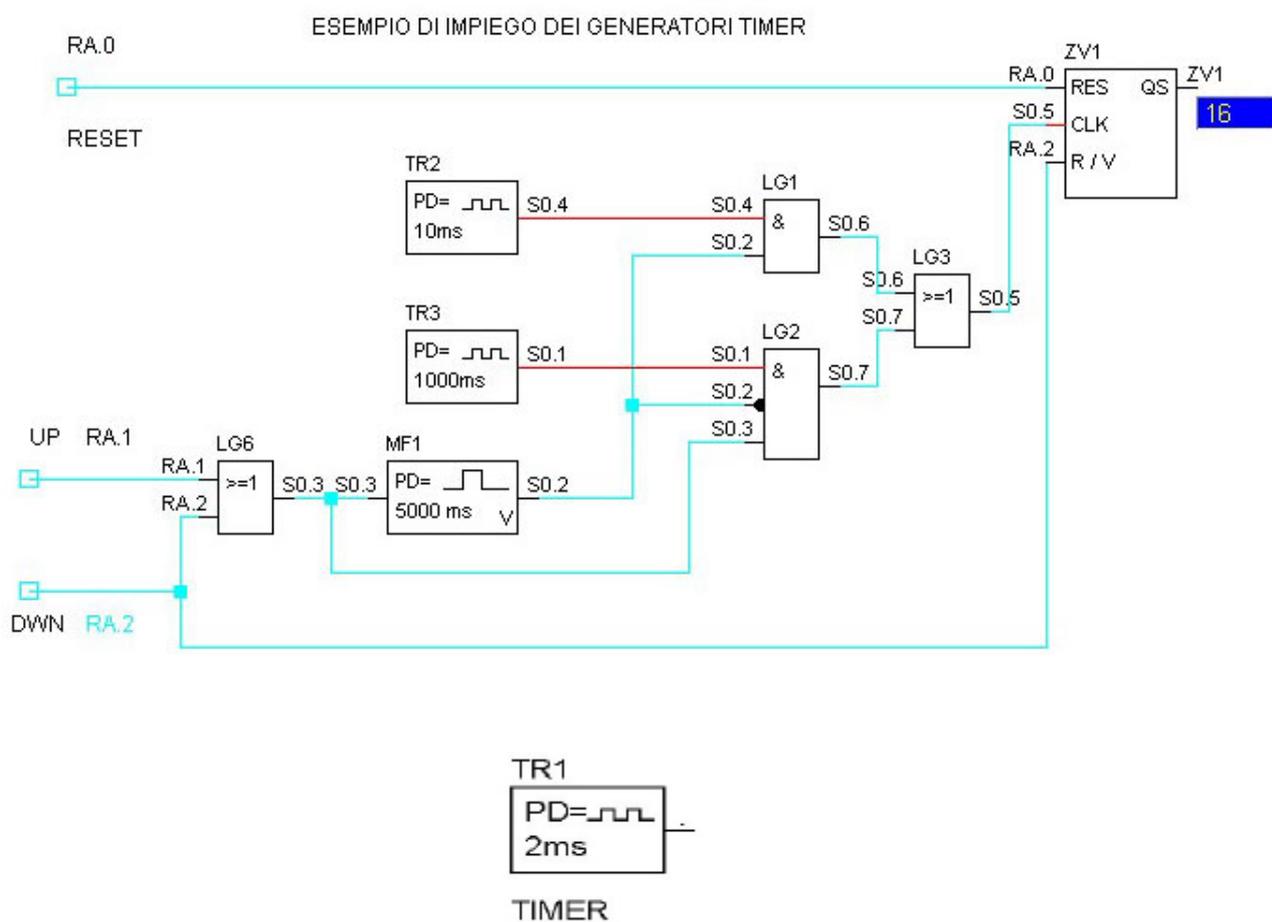
## Esempio applicativo:





## Timer o generatore base di tempi :

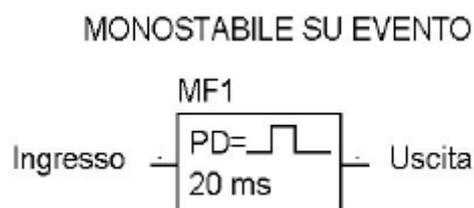
in uscita dal blocco **timer** è possibile prelevare un segnale digitale, di tipo **astabile**, e di durata ciclica perfettamente simmetrica. L'applicazione di questo segnale è, normalmente, quello di **base di tempi** (clock generator) per contatori, display LCD, multiplexer, ecc.





### Blocco circuito monostabile, su evento:

una variabile digitale applicata all'ingresso del **monostabile** determina, per un periodo limitato di tempo, la variazione dell'uscita, **da livello basso a quello alto**. Le funzioni svolte da questo blocco sono le seguenti :



### monostabile,su evento, retriggerabile:

l'applicazione di un impulso digitale in ingresso,di breve durata ( one shot), determina lo stato di **cambiamento** digitale dell'uscita **da livello logico basso a livello logico alto**. Tale livello logico **resta attivo** per tutta la durata della temporizzazione impostabile da **1 a 65535 ms**. Attivando il retrigger, l'uscita rimane a livello logico alto, tra il ripetersi di un impulso e l' altro, nel dominio di tempo di durata dell'impulso stabilito .

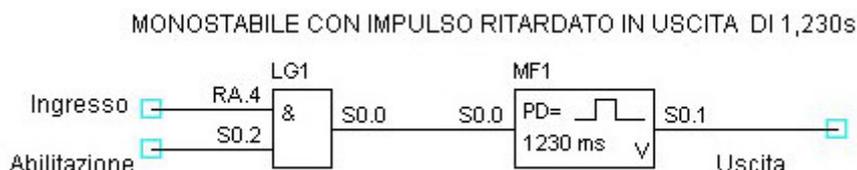
### Esempio di monostabile,retriggerabile:





### monostabile ,su evento,con attivazione ritardata

l'applicazione dell' impulso digitale in ingresso, determina lo stato di cambiamento digitale dell'uscita **da livello logico basso a livello logico alto**, solo **dopo un certo tempo**, determinato dall' impostazione della soglia di intervento che va' da **1 a 65535 ms**. **L'impulso** di ingresso **deve essere mantenuto** fino al cambiamento di stato dell'uscita. Togliendo l'impulso di ingresso l'uscita del monostabile torna a livello logico zero.



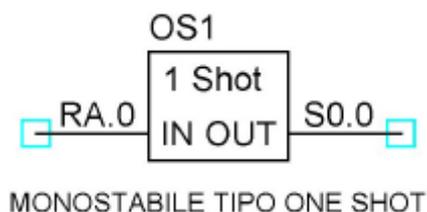
### monostabile ,su evento,con attivazione slope

l'applicazione dell'impulso digitale in ingresso, di breve durata ( one shot ) , determina lo stato di cambiamento digitale dell'uscita **da livello basso a livello alto**. Tale livello **resta attivo** per tutta la durata della temporizzazione impostabile da **1 a 65535 ms**.



### Monostabile di tipo one-shot :

una variabile digitale, applicata all'ingresso del monostabile, provoca il **cambiamento** di stato **dell'uscita per un breve periodo di tempo**, da livello logico basso a livello logico alto. L'impulso di uscita e' **transitorio** , anche se l'ingresso continua ad essere mantenuto a livello alto.





## Blocco condizionatore di impulsi in ingresso, impulse-in.

Questo modulo viene impiegato per determinare il tempo di durata di un impulso, applicato all'ingresso di uno dei **PortB4...7** , con una risoluzione di **16 bit**.

La precisione della misura dipende da diversi fattori, tra questi, la frequenza di clock ed il numero di timer e monoflop impiegati.

Impiegando un quarzo da 4 MHz la risoluzione della misura e' di 1 microsecondo ( a 20 MHz  $\geq$  20 nanosecondi )

La lunghezza dell'impulso da misurare viene compresa tra **50 e 65500** campionamenti/sec.

Il tempo di durata dell'impulso e' calcolato sul fronte positivo di salita dell'impulso .Sul fronte di discesa dell'impulso ha termine la misura che viene memorizzata in una variabile denominata **RBxTIME ed RBxTIME\_HI**.

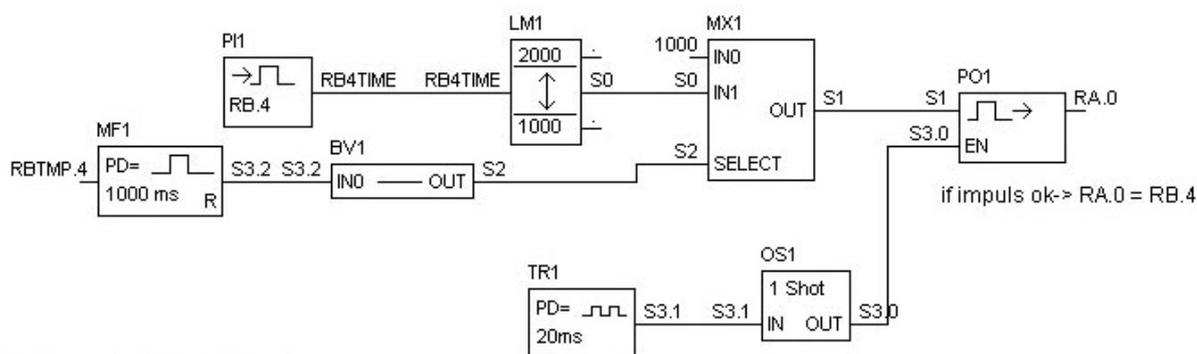
Portando il cursore del mouse sul blocco impulsin potrete selezionare, nella finestra di dialogo, se la misura deve essere attuata sul fronte di salita o di discesa dell'impulso. Inoltre sara' possibile selezionare quale degli ingressi **PortB** utilizzare nel circuito.

Parsic genera, via software, il codice **RBTMP.x** , che e' la copia dell'impulso ricondizionato, applicato al **PortB.x**. Questo codice viene normalmente utilizzato per attivare i circuiti di campionamento e conversione del segnale all'interno del circuito in progetto. Il blocco logico condizionatore di impulsi, nasce dall'esigenza applicativa nei circuiti di telecomando proporzionale quali, ad esempio, quelli del modellismo navale, aereo, automobilistico, e nella robotica (vedi RoboKit) ecc. Nell'esempio che segue, viene mostrato un circuito applicativo dove, oltre al condizionatore di impulsi ( siglato P11 ), vengono utilizzati anche i blocchi logici limiter ( LM1 ) ed impuls-out (P01) .

## Autopilota : impiego dei moduli IMPULSIN, IMPULSOUT, LIMITER

per applicazioni di radiocontrollo su modelli navali, aerei, automobili

Impuls on RB.4 : ogni 20 ms vengono campionati gli impulsi in ingresso di durata tra 1000 e 2000 microsecondi



RBTMP e' la copia del PORTB Input

esso viene generato da Parsic quando si seleziona il blocco Impul-in

Se non viene applicato l'impulso su RB.4, l'uscita di MF1 si pone allo zero logico

ed MX1 commuta P01 ad un valore di uscita costante RA.1 = 1000 microsec



## Blocco condizionatore di impulsi, impuls-out, e generatore PWM.

Questo circuito rielabora il tempo di durata di un impulso applicato al suo ingresso e lo restituisce in uscita con una precisione di campionamento a 16 bit.

Per migliorare la precisione dell'impulso di uscita, il programma del microcontrollore subisce una fase di arresto per tutta la durata del campionamento.

Viene generato un solo impulso di uscita ,quando l'ingresso di abilitazione EN e' posto a livello alto, ( fare riferimento allo schema elettrico applicativo impuls-in ).

Il tempo di campionamento dell'impulso dipende dalla frequenza di clock del microprocessore,che corrisponde ad una risoluzione di 1 microsecondo se il quarzo del microprocessore e' di 4MHz.

La lunghezza dell'impulso da applicare, all'ingresso del modulo, deve essere compresa tra 50 e 65550 campionamenti/sec.

## Generatore di impulsi PWM.

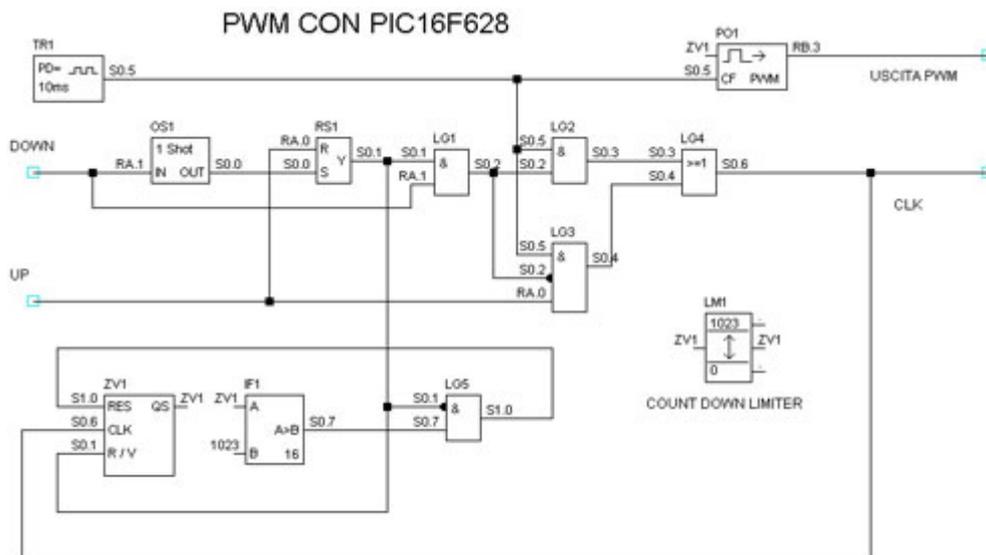
Questa funzione non e' implementabile su tutti i modelli di Pic.

Utilizzando il generatore **PWM** e' possibile ottenere dal terminale di uscita del modulo un impulso la cui durata dipende dal valore numerico della variabile digitale applicata al terminale di ingresso.

La frequenza di uscita dell'impulso e' costante **4KHz** con clock di 4MHz. Con questo modulo e' possibile ottenere valori di tensione analogici, collegando in uscita opportuni filtri livellatori d'onda.

Ad ogni variazione positiva,dell'impulso d'ingresso, il valore di uscita al modulo viene scritto nel **Duty-Cycle- Register**. Questo valore e' compreso tra **0 e 1023** bit.

## Esempio applicativo del generatore PWM



QUESTA CONFIGURAZIONE CONSENTE DI OTTENERE DEGLI IMPULSI DI DURATA VARIABILE SULL'USCITA RB.3  
IL CIRCUITO PUO' ESSERE USATO CONVENIENTEMENTE PER IL PILOTAGGIO DI ALIMENTATORI, DIMMER, CIRCUITI DI CONVERSIONE ecc.



## Limiter

Questo blocco funzionale controlla il **contenuto di una variabile** e stabilisce **due livelli di allarme** corrispondenti al valore **limite superiore ed inferiore** della variabile stessa.

Se il contenuto della variabile di ingresso è maggiore del valore di set impostato (top limit) ,l'uscita corrispondente si porterà livello logico alto.

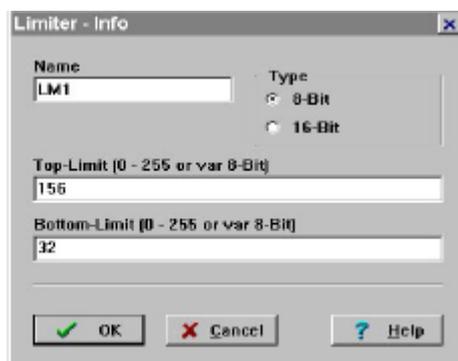
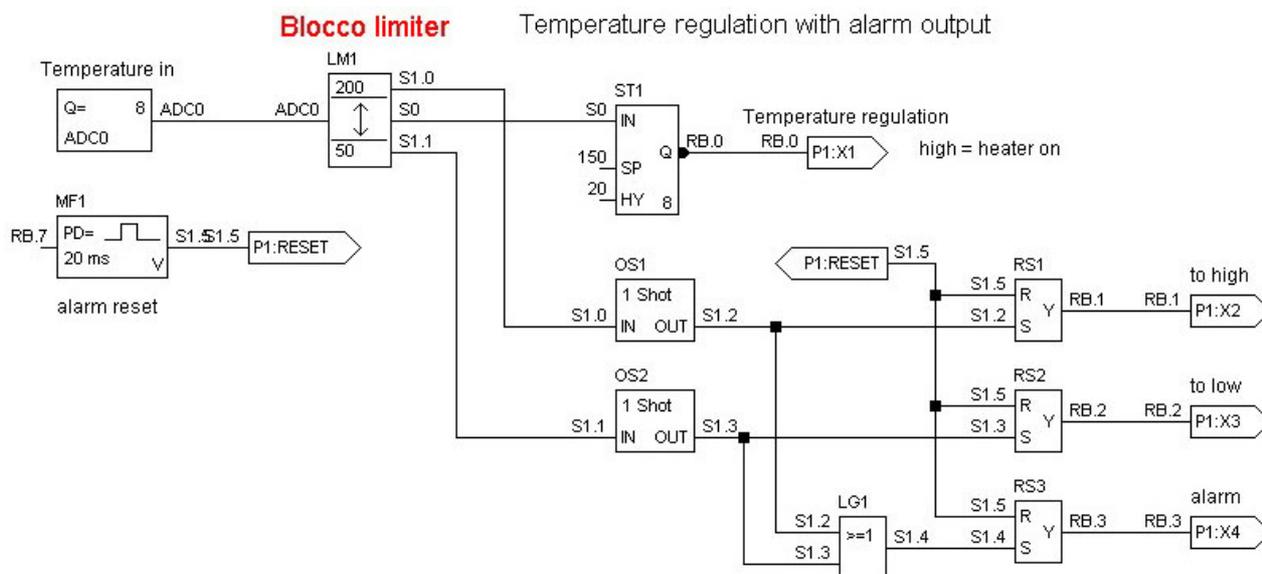
Se il contenuto della variabile di ingresso è inferiore al valore di set impostato (bottom limit) , l'uscita corrispondente si porterà a livello logico alto.

Il blocco funzionale limiter si comporta similmente ad un **filtro passa banda**: il valore numerico della variabile digitale, in uscita dal blocco, sarà contenuto all'interno dei valori di set (top e bottom limit) impostati, i valori inferiori e superiori al set saranno tagliati e non verranno considerati nell'elaborazione della variabile.

**L'uso delle uscite digitali di allarme sono del tutto opzionali.** Se esse non vengono impiegate Parsic non le utilizzerà e non verranno considerate nel file sorgente.

E' possibile stabilire il funzionamento del blocco con definizione a **8** oppure a **16 bit**.

### Esempio di impiego del blocco limiter:



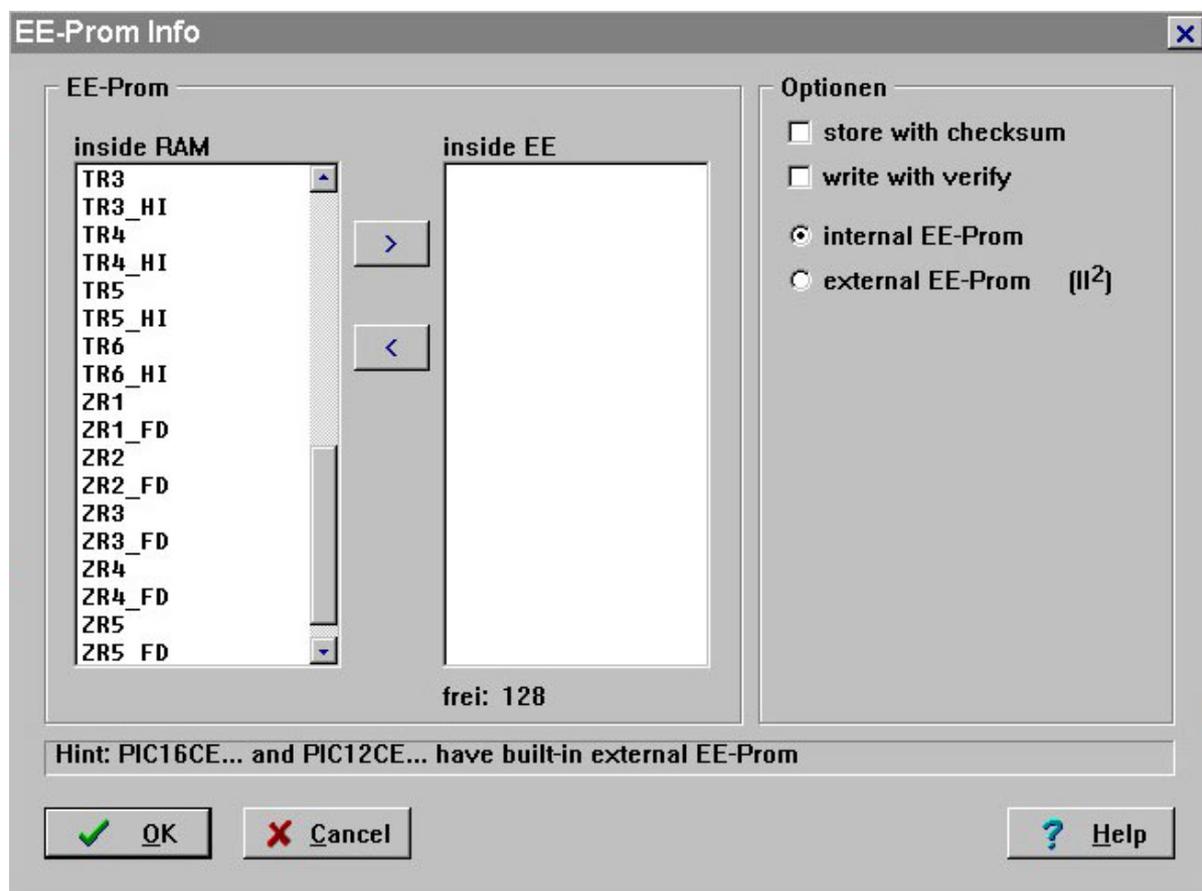
## Gestione EEPROM

La **EEPROM** dati è una particolare area di memoria del PIC nella quale è possibile scrivere i valori delle variabili digitali che non vogliamo perdere in caso di mancanza di tensione di alimentazione, ovvero allo spegnimento del circuito mpc.

Le modalità di accesso alla memoria **EEPROM** seguono speciali procedure onde evitare perdite di dati in condizione di funzionamento anomalo.

Con il blocco **EEPROM (EE)** si possono memorizzare i contenuti delle variabili digitali sia nelle memorie integrate nel chip del mpc, sia in quelle esterne. Le variabili contenute nella memoria **EE** sono richiamate ad ogni reset del mcp. Affinchè le variabili possano essere scritte nella memoria **EE** bisogna utilizzare la tabella **EE-PROM INFO**. Questa tabella si attiva con il tasto destro del mouse quando il puntatore si trova posizionato sul blocco funzionale **EExx**. Essa consente di spostare i dati dall'area **RAM** a quella **EEPROM** con semplici operazioni tabellari.

La figura seguente mostra la tabella EE-Prom-info che appare sullo schermo quando essa viene attivata:



Nel riquadro di sinistra della tabella sono elencati le variabili dei blocchi logici digitali facenti parte del circuito in progetto. Queste variabili, saranno codificati in una sequenza corretta di codici operativi (opcode) e memorizzati all'interno della memoria del pic.

Tale memoria di programma è composta, secondo il tipo di microprocessore, da un certo numero di locazioni, ognuna delle quali può contenere un opcode da eseguire. Ricordiamo brevemente che, l'architettura dei microcontrollori di Microchip è particolare, si basa su una CPU strettamente di tipo RISC, con separazione tra bus dati e bus istruzioni (tipo Harvard), nessuna possibilità di indirizzare memoria esterna ed un elevato numero di registri interni, che costituiscono di fatto l'unica area RAM utilizzabile.

### Gestione della EEprom interna.

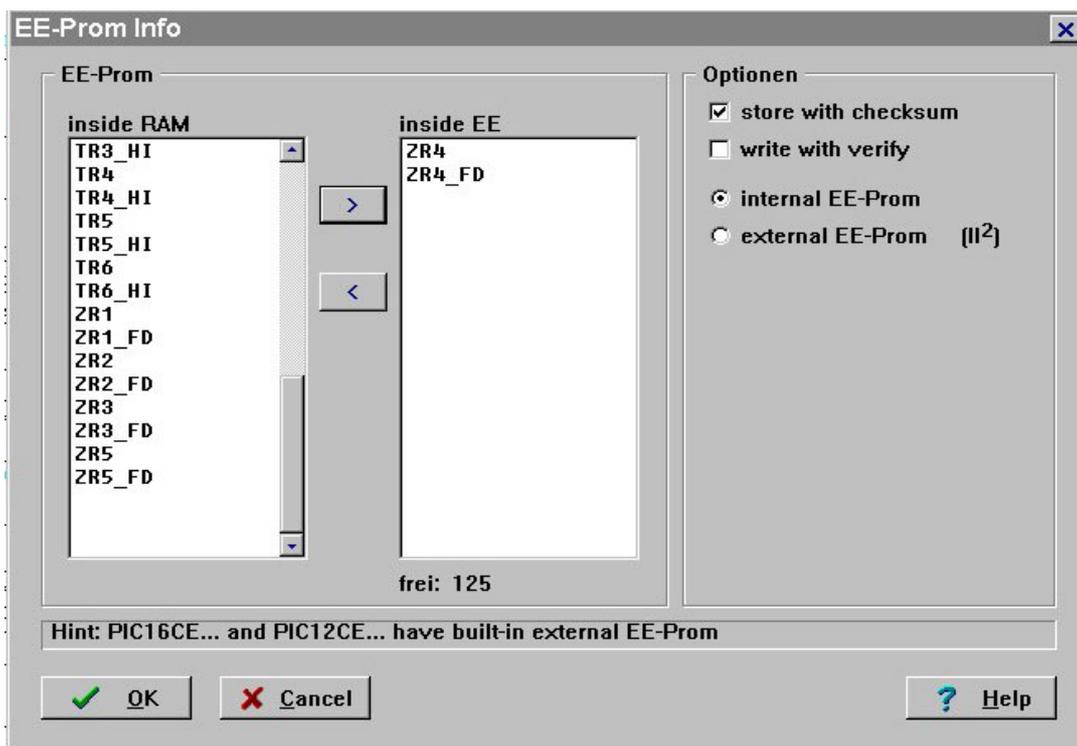
Parsic consente di poter effettuare il salvataggio delle variabili digitali, elaborate dal programma, predisponendo in modo opportuno il trasferimento dei dati dall'area di memoria RAM a quella EEprom.

#### Esempio:

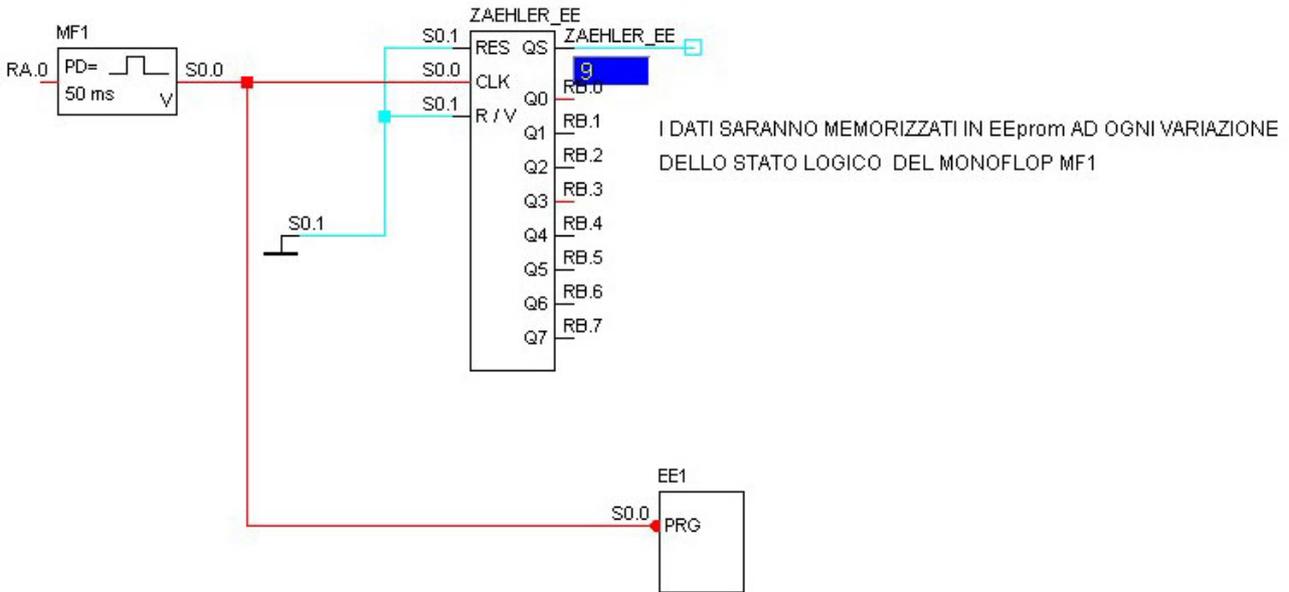
vogliamo trasferire il valore della variabile ZR4 in EE.

Le variabili ZR4 ed ZR4\_FD si possono individuare nella tabella di sinistra. Posizionarsi con il cursore del mouse su una delle due variabili e selezionarla (tasto sinistro del mouse). Ora portarsi sul pulsante di trasferimento, posto al centro della tabella, e spostate nella tabella di destra la variabile selezionata azionando il pulsante > : la variabile selezionata sara' trasferita immediatamente nell'area EEprom. Si ripete l'operazione con la seconda variabile e cosi' via, fino a trasferire in EEprom tutte le variabili di nostro interesse. Ovviamente si possono eseguire l'operazioni inverse, nel caso di trasferimenti non piu' necessari.

**Si consiglia di attivare la funzione Cecksum, specialmente per i dispositivi Pic vergini.**



**ESEMPIO DI COLLEGAMENTO DEL BLOCCO EEprom**



**Gestione della EEprom esterna.**

E' possibile la gestione esterna della EEPROM, per tutti quei dispositivi Pic (previsti a programma ) che non sono equipaggiati di EEprom interna ( PIC12CE e PIC16CE).

I dispositivi che adoperano l' EEprom esterna saranno collegati a questa per il mezzo del bus I<sup>2</sup>C ( Inter-Integrated Circuit ) adoperando lo standard Philips. Nella specifica dello standard I<sup>2</sup>C la velocità massima di trasferimento dei dati è di 100Kbps. Parsic rispetta il protocollo di comunicazione I<sup>2</sup>C . Di seguito specifichiamo le seguenti terminologie, proprie del bus I<sup>2</sup>C :

<b>Transmitter</b>	e' il dispositivo che invia i dati attraverso il Bus di collegamento;
<b>Receiver</b>	e' il dispositivo che riceve i dati dal Bus di collegamento;
<b>Master</b>	e' il dispositivo che inizia il trasferimento dei dati, genera il segnale di Clock, e termina il trasferimento dei dati;
<b>Slave</b>	e' il dispositivo indirizzabile attraverso il Master ;
<b>Multi-master</b>	sono piu' dispositivi Master che condividono lo stesso Bus di collegamento ed il cui sistema di controllo fa si che non si verifichino conflittualita' nel sistema di comunicazione;
<b>Arbitration</b>	e' una procedura che consente ad un solo dispositivo master di controllare il bus di collegamento e fa si che non si verifichino conflittualita' nel sistema di comunicazione;
<b>Synchronization</b>	procedura che consente di sincronizzare il segnale di clock a piu' dispositivi collegati tra loro.

**Nota :**

Parsic consente di operare con più dispositivi I<sup>2</sup>C collegati sullo stesso bus di comunicazione, però si consiglia di operare **sempre** con soli **due** dispositivi per volta, cioè con **un** dispositivo **Master** ed **un** dispositivo **Slave**. Nello schema elettrico funzionale del sistema distingueremo :

il circuito **Master** : ( Pic) trasmette e Slave riceve;  
 il circuito **Slave** : (EEProm) trasmette e Master riceve.

In ogni caso sarà sempre il **Master** a generare il segnale di clock.

Lo stadio di uscita del Clock si chiama **SCL** e quello dei dati **SDA**. Questi stadi sono costituiti da transistor configurati ad **open collector** oppure **open drain**, ed una resistenza esterna di pull-up deve essere collegata necessariamente alle porte di comunicazione.

**Denominazioni dei port di comunicazione.**

Bisogna distinguere la denominazione delle porte di comunicazione secondo il tipo di Pic che viene utilizzato. Per i Pic della serie **12CExxx** , Parsic denomina i port in questo modo:

**SDA = GPIO.6**  
**SCL = GPIO.7**

Optionen

store with checksum

write with verify

internal EE-Prom

external EE-Prom (I<sup>2</sup>)

Size in byte

Bus-Adresse

SDA

SCL

Per i Pic della serie **16CExx**, Parsic denomina i port in quest'altro modo:

**SDA = EEINTF.6**  
**SCL = EEINTF.**

The screenshot shows a dialog box titled "Optionen" with the following settings:

- store with checksum
- write with verify
- internal EE-Prom
- external EE-Prom (I<sup>2</sup>)
- Size in byte: 128
- Bus-Adresse: 0
- SDA: EEINTF.1
- SCL: EEINTF.2

Si possono comunque utilizzare altri port di comunicazioni sia per il terminale **SDA** che per **SCL**. Bisogna fare attenzione alla denominazione di questi port, nominandoli correttamente:

**PORTx.x oppure GPIO.x**

Altre denominazioni tipo Rx.0 oppure GP.x, ecc. **non sono ammesse**.

### Salvataggio con cecksum.

Salvando in eeprom i dati ed utilizzando questa opzione, viene generato il codice di cecksum di un byte. Questo codice viene utilizzato per confrontare il corretto valore dei dati memorizzati in eeprom. Se il dato memorizzato non e' corrispondente al cecksum, il dato viene azzerato.

### Check after writing ( verify)

Utilizzando questa opzione, il dato trasferito in EEprom viene controllato nella sua integrita'. Se questo dato non e' stato scritto corettamente, il processo di trasferimento viene ripetuto.

### Slave address

Le EEprom 24Cxx, devono essere identificate attraverso un indirizzo di collegamento. Questo indirizzo viene specificato nella casella Bus-Address con la denominazione A0, A1 e A2. Per default questa denominazione e' 0 ( zero).

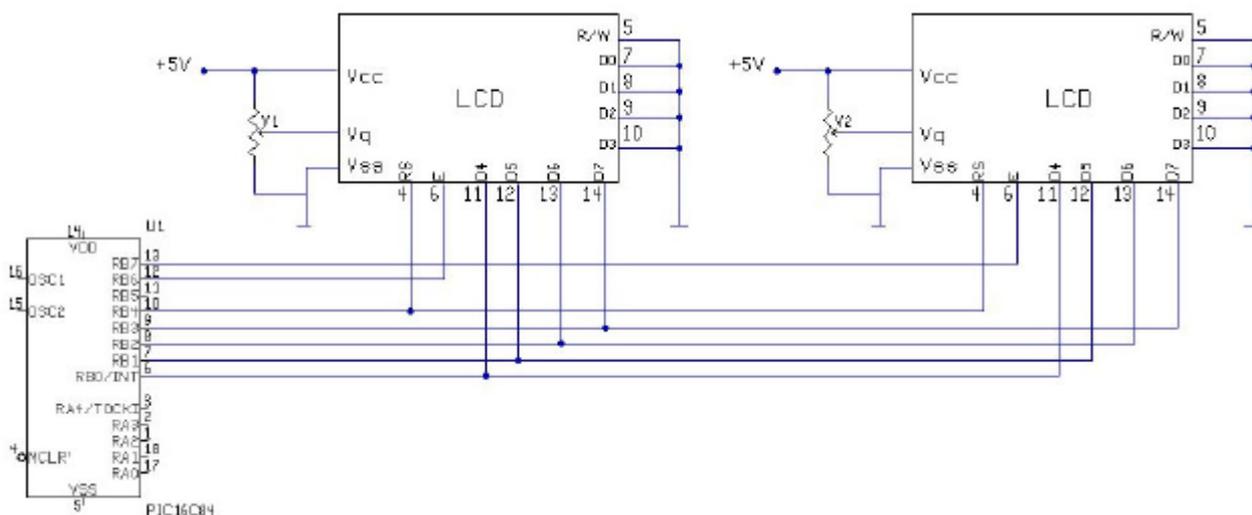


## Gestione display LCD alfanumerici.

I display LCD piu' comuni dispongono di un'interfaccia ideata da **Hitachi** che nel tempo e' diventata uno standard industriale utilizzato ovunque. Questo tipo di interfaccia prevede che il display sia collegato al micro tramite un bus di **4 o 8 linee**, piu' tre linee di controllo.

Parsic impiega, appunto, lo standard Hitachi, per la gestione dei moduli alfanumerici LCD.

Le linee di pilotaggio dei moduli sono ridotte a sei, con evidente risparmio delle risorse dei Pic. Il programma consente di collegare, all'uscita dello stesso dispositivo Pic, fino a **tre moduli LCD**. Le linee utilizzate per il pilotaggio, sono quelle relative ai **PortB** (default) ma, secondo le necessita' del progetto, e' possibile impiegare port diversi. Sono impiegati i port **Rx.0 ..... Rx.4** per i **dati (D)** ed i port **Rx.5...Rx.7 (E)** per le **linee di controllo**. Lo schema seguente mostra come sia possibile il collegamento di due display LCD utilizzando le linee dello stesso port:



Si possono utilizzare **LCD fino a 4 righe, da 8 a 40 caratteri**.

Dato che il tempo necessario alla corretta visualizzazione dei caratteri puo' variare secondo il tipo di LCD utilizzato, e' possibile variare il tempo di trasferimento dei dati dal microprocessore al modulo impostando tre tempi diversi di propagazione :

$$x 1 = 50 \mu\text{sec};$$

$$x 2 = 100 \mu\text{sec};$$

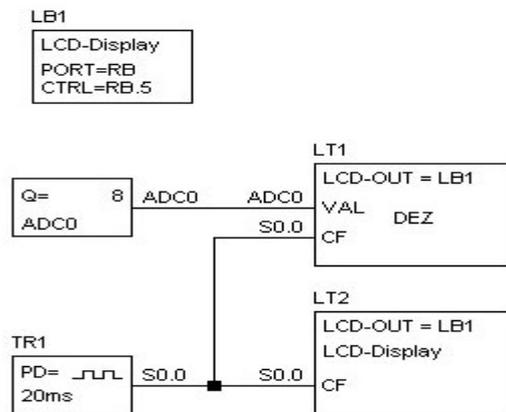
$$x 3 = 150 \mu\text{sec};$$

La tabella, di seguito descritta, indica brevemente le funzioni svolte dai pin normalmente disponibili su questo tipo di display ed utilizzate nel blocco funzionale LCD :

1	<b>GND</b>	massa
2	<b>Vcc</b>	alimentazione +5Vcc
3	<b>Vq</b>	regolazione contrasto LCD
4	<b>RS</b>	register select, e' una linea di controllo con cui si comunica al display se si sta inviando sul bus dati un comando RS = 0 oppure un dato RS = 1
5	<b>R/W</b>	non usato, viene collegato al GND
6	<b>Enable</b>	e' una linea di controllo e serve per abilitare il display ad accettare dati ed istruzioni dal bus
7..10	<b>DB.0...3</b>	sono le linee dati da 0 a 3 che non vengono utilizzate da Parsic
11..12	<b>DB.4...7</b>	sono le linee dati da 4 a 7 utilizzati per le comunicazioni con il display

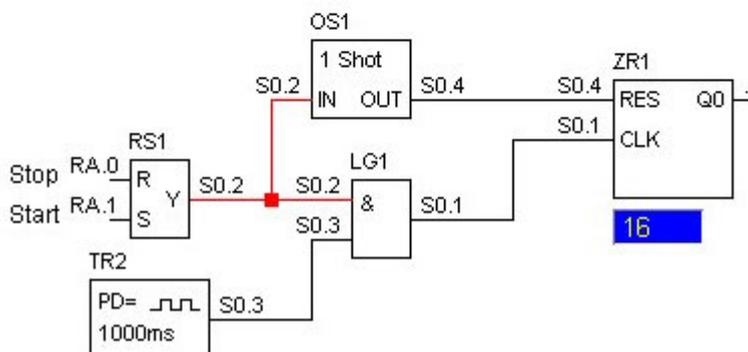
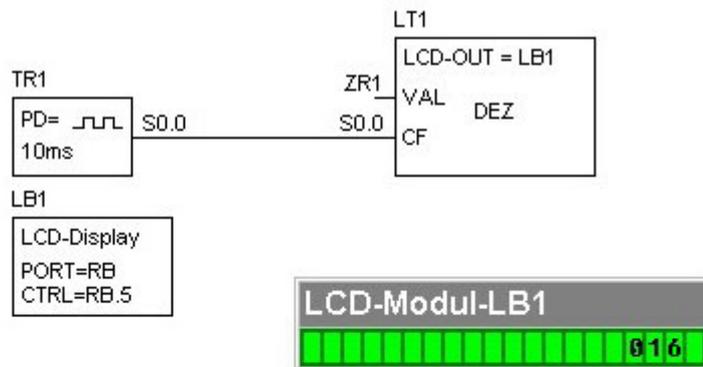


Lo schema elettrico seguente mostra il corretto collegamento dei blocchi logici LCD.



Bisogna considerare che, il dato che si vuole fare apparire sul rigo del display deve essere commutato applicando al terminale CF del blocco logico, una variabile digitale che, nel caso in esempio, e' stata prelevata all'uscita di un generatore di clock.

L'esempio reale di collegamento al blocco LT1, che riportiamo di seguito, dara' una indicazione piu' precisa di quanto ora esposto:



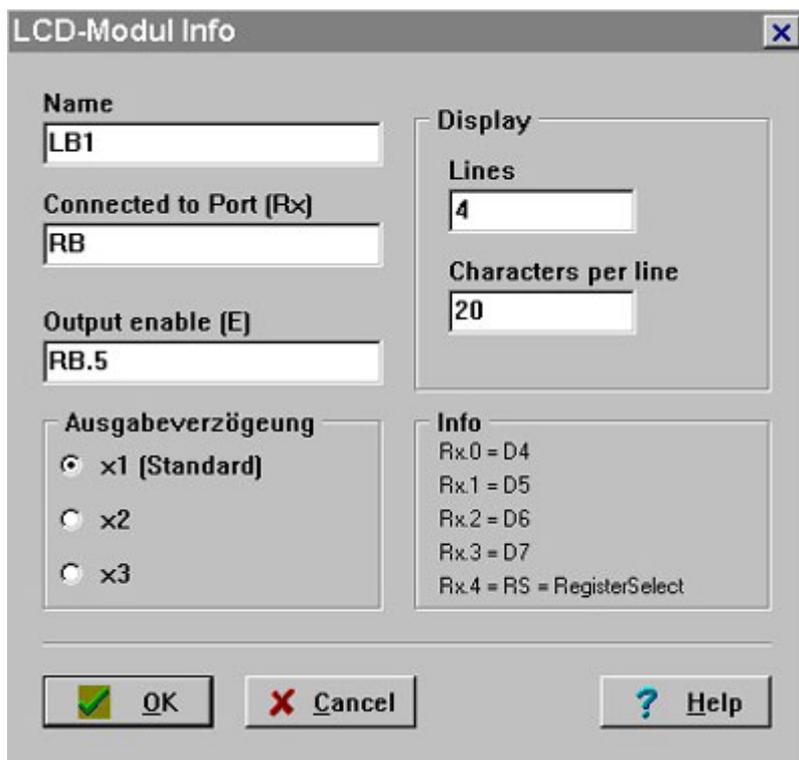
La visualizzazione del conteggio **non sarebbe affatto possibile** se il terminale CF di LT1 non venisse commutato dal generatore di clock TR2.



## Inserimento del testo nei display LCD

### Modulo base.

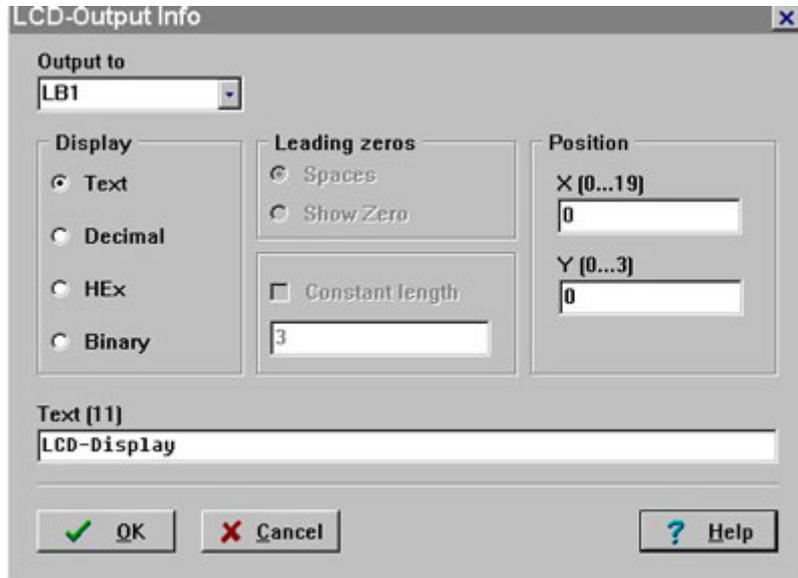
Per rendere funzionante un display LCD, e' necessario posizionare nello schema elettrico il **modulo base** che definisce le caratteristiche elettriche del dispositivo che vogliamo utilizzare. Nello schema elettrico precedente si distingue nettamente il modulo base **LB1** dal display **LT1**. In ogni schema elettrico e' necessario **un solo modulo base** ed **n moduli LTx**, tante quanto sono le frasi che vogliamo gestire.



La finestra di dialogo sopra riportata consente di definire il funzionamento del tipo di modulo LCD impiegato in progetto. Nel riquadro **Name** specificare il codice del dispositivo (LB1, LB2, LB3, ecc). Specificare su quale Port si vuole collegare il display, **Connected to Port (Rx)**, e quale Pin sarà specializzato per pilotare il terminale **Enable** del modulo, **Output enable (E)**. Si possono impiegare display da **1 a 4 righe, Lines**, fino a **40 Characters per line**.

Nel riquadro **Ausgabeverzögerung**, si possono selezionare 3 diversi tempi di scrittura dei caratteri sull'LCD. Per LCD di vecchia generazione si consiglia di utilizzare l'opzione x3.

## Modulo di testo.



**Per ogni rigo di testo**, che si vuole fare apparire sul display, bisogna posizionare nello schema elettrico **un modulo LT** , selezionandolo, di volta in volta ,dalla Toolbar. Una volta in posizione il modulo sara' gestito per mezzo della finestra di dialogo,come rappresentata in figura.

Specificate il nome del modulo che state trattando **Output to...**,in quanto lo stesso bus potrebbe gestire piu' di un display. Si possono impiegare caratteri in formato testo, esadecimale, o binario. I caratteri vanno descritti nel riquadro **Text** e dovranno occupare lo spazio necessario alla loro rappresentazione.

Essi vanno inseriti nella **Position**, che l'utente decide di assegnare . Prestare molta attenzione quando si trattera' di gestire messaggi commutati,ovverro diciture tipo ON-OFF oppure ALTO-BASSO, collocate nello stesso rigo e stessa posizione.

Le diciture devono perfettamente coincidere fra loro. Eventuali frasi piu' corte rispetto ad altre piu' lunghe saranno compensate immettendo degli spazi vuoti. In altri termini, se la prima frase occupa 12 spazi, anche la seconda frase dovra' occupare gli stessi spazi; se piu' corta si dovranno aggiungere n spazi fino a raggiungere la stessa lunghezza della prima ( e viceversa). Si possono inserire caratteri speciali specificando il codice del carattere preceduto e chiuso dal segno grafico #.

Esempio: # $E2$ # , #226# ,#226,S#



## Rappresentazioni numeriche decimali.

Per la visualizzazione dei dati decimali numerici ,ci sono tre modalita' di rappresentazione degli stessi:

- lunghezza variabile delle cifre : i numeri vengono rappresentati in crescendo,con shift a destra della cifra, nessun zero iniziale;  
**Spaces**
- rappresentazione degli zeri : la cifra viene rappresentata da una serie di zeri iniziali che vengono sostituiti dal valore numerico corrispondente mano a mano che procede il conteggio;  
**Show zero**
- lunghezza costante : Viene rappresentato un valore numerico di lunghezza costante.  
**Constant lenght** La rappresentazione numerica non puo' subire incrementi oltre la cifra impostata.

La rappresentazione dei dati numerici decimali, a lunghezza costante, e' molto utile quando si vogliono rappresentare valori numerici derivati, ad esempio, da una conversione analogico-digitale. Ad esempio, se desideriamo che del valore di conversione equivalente a 1223, venga rappresentato solo il valore 23, si dovra' spuntare la casella **Constant lenght** e si dovra' inserire la costante 2, nell'apposito riquadro.

## Blocco funzionale Sleep mode



L'istruzione **SLEEP** viene utilizzata per mettere i Pic in **Power Down Mode** e ridurre di conseguenza la corrente assorbita da Pic che passerà da pochi mA ad altrettanto pochi  $\mu\text{A}$ , ovvero **1000 volte in meno**, circa, il valore normalmente assorbito dal microprocessore.

Parsic provvede ad inserire l'istruzione **SLEEP** in un determinato punto del file sorgente, spegnendo tutti i circuiti interni, tranne quelli necessari a mantenere lo stato delle porte I/O ed a rilevare le condizioni di risveglio del processore.

Per ridurre il consumo energetico in questo stato, **non devono esserci collegati**, ovviamente, circuiti alle linee di uscita del Pic che assorbano corrente, o meglio, questi circuiti devono essere progettati in modo da limitare il loro assorbimento nelle condizioni di **Power Down**.

Si consiglia di collegare al positivo **Vdd** o al negativo **Vss** di alimentazione tutte le linee in alta impedenza non utilizzate.

Quando le linee di ingresso **sono tutte ad alto livello**, viene attivato lo **SLEEP MODE** ed il programma viene posto in condizione di **STOP**, **l'oscillatore e' posto ad off**.

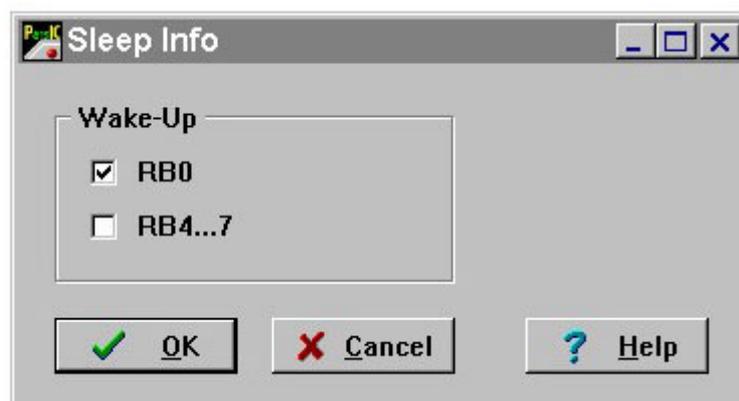
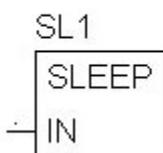
Per ripristinare le funzioni del microprocessore ci sono due possibilità:

**la prima** : applicare un impulso positivo al port **RB.0**;

**la seconda** : applicare una variazione di livello ad uno dei port **RB.4, RB.5, RB.6, RB.7**.

## Restrizioni dell'uso della funzione di Sleep mode.

Non e' possibile abilitare la funzione **Watch-Dog Timer** se viene utilizzato il **blocco funzione Sleep** . Questa funzione non e' implementata nei microcontrollori della serie **12CExxx**.





## Blocchi funzionali Table e Call (subroutines).

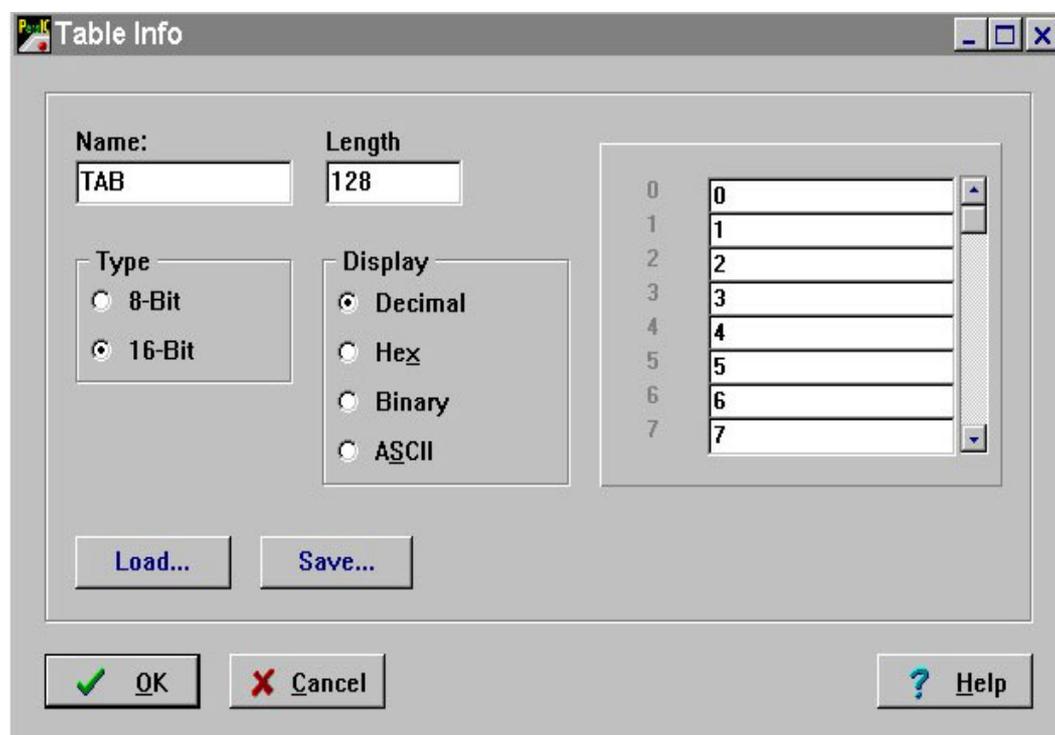
### Table.

Il blocco funzionale **Table** viene utilizzato per memorizzare nella ROM un determinato numero di variabili digitali che potranno essere richiamate dal blocco funzionale **Call**. Il valore espresso di queste variabili può essere in dicato in 8 oppure 16 bit.

Valori a 8 bit	vengono catalogati in una tabella contenente <b>256</b> locazioni (0...255) specificate in numerazione crescente da 1 a 256, il cui valore è compreso tra <b>0 e 255</b> ;
Valori a 16 bit	vengono catalogati in una tabella contenente <b>128</b> locazioni (1...128) specificate in numerazione crescente da <b>0 a 127</b> , il cui valore è compreso tra 0 e 65535. La sequenza dei valori è compresa in una scala contenente il low byte ed high byte .

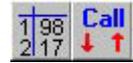
L'utilizzo di questo blocco è indicato per le seguenti funzioni:

**decoder per display a 7 segmenti LED;**  
**linearizzazione di valori analogici.**



Portarsi con il puntatore del mouse sul blocco funzionale **Tab** e cliccare con il tasto destro. Si aprirà la finestra di dialogo come sopra in figura. Procedere nel modo seguente:

## Table



<b>Name:</b>	<b>Length</b>
TAB	128

dare un nome identificativo alla tavola e definire la sua lunghezza, cioè il numero delle variabili che essa deve contenere.

<b>Type</b>
<input type="radio"/> 8-Bit
<input checked="" type="radio"/> 16-Bit

Definire il tipo di dato se a 8 oppure 16 bit

<b>Display</b>
<input checked="" type="radio"/> Decimal
<input type="radio"/> Hex
<input type="radio"/> Binary
<input type="radio"/> ASCII

Definire come si vuole rappresentare i dati in tabella

Decimale = numeri compresi tra 0 e 9;

Hex = valori numerici e lettere, preceduti dal simbolo \$ es. \$7F

Bin = valori numerici 0 - 1. Il bit più significativo viene posto a sinistra.

Asc = codici ASC , preceduti dal segno '

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Indipendentemente dal tipo di rappresentazione che si è scelto nel riquadro display, inserire i dati partendo dal rigo 0 della tabella.

<b>Load...</b>	<b>Save...</b>
----------------	----------------

Il contenuto della tabella può essere salvato in formato ASC. con il comando **Save...**

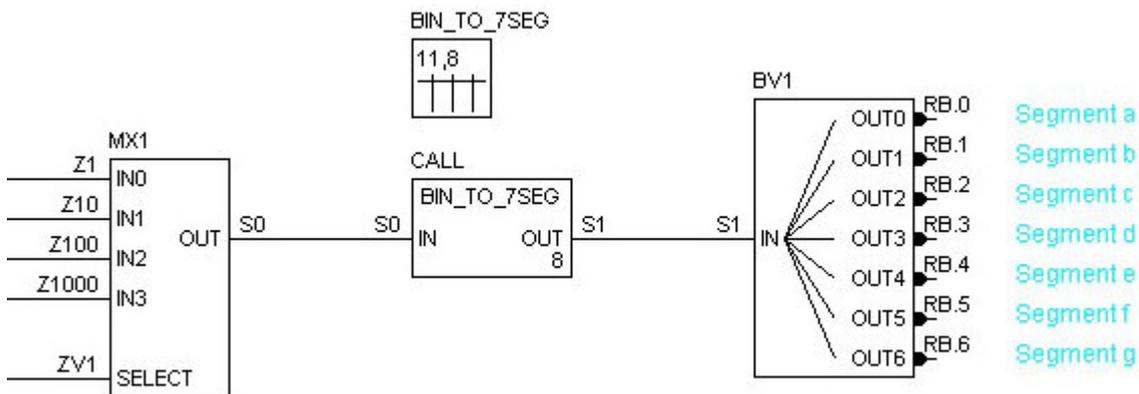
Il contenuto di una tabella, precedentemente salvato, viene ricollocato in tabella in formato ASC. I valori vengono riposti con lo stesso ordine in cui sono stati salvati. Le linee precedentemente non utilizzate vengono ignorate. Se la tabella è predisposta per valori ad 8 bit, i valori eccedenti vengono eliminati.



## Call.

Il blocco funzionale **Call** serve per richiamare i valori memorizzati nel blocco **Table** (souboutines).

I valori delle variabili contenuti in tabella, vengono richiamati dalla **Call**, attraverso il collegamento ai blocchi funzionali, come in esempio :



All'ingresso del modulo **MX1** vengono inviate variabili digitali ognuna del valore di 8 bit.

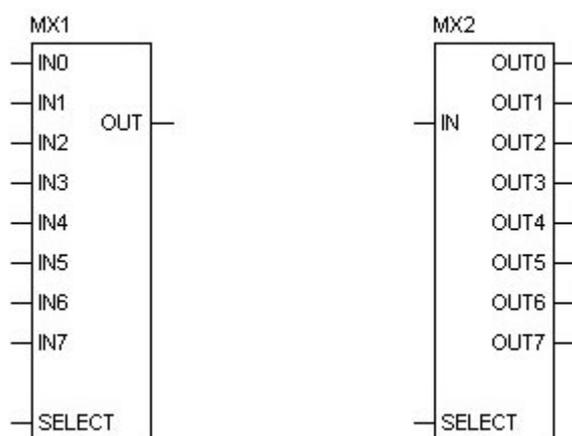
L'ingresso Select di **MX1** e' collegato ad un generatore di selezione che pone in uscita, una alla volta, i dati di **MX1**. Questi, entrando nel modulo **CALL** vengono confrontati con i valori memorizzati in tabella e convertiti, in un' altro valore digitale, corrispondente in questo caso alla decodifica 7 segmenti del display led.

Le variabili in ingresso al modulo **CALL** devono corrispondere al numero di variabili digitali memorizzate in tabella. Se i valori sono in eccedenza, il processo di decodifica viene resettato, con conseguente blocco del sistema. I valori delle variabili digitali in ingresso-uscita dal blocco sono da 8 oppure 16 bit .



## Multiplexer

Il blocco funzionale multiplexer viene impiegato come distributore di ingresso/uscita di variabili digitali. E' costituito da un circuito la cui caratteristica e' quella che le linee di ingresso/uscita vengono commutate al terminale di ingresso/uscita per mezzo di un commutatore elettronico ad alta velocita'. l'esempio circuitale che segue mostra le configurazioni che questo blocco puo' assumere:



Nel blocco di sinistra, le linee di ingresso sono commutate in uscita, ad una ad una, per mezzo della variabile di selezione applicato al terminale Select.

Nel blocco di destra, la linea di ingresso e' commutata alle uscite, utilizzando ancora il terminale Select. La selezione avviene sempre per una linea di uscita alla volta.

Se la selezione impostata al terminale Select e' piu' alta del numero di ingressi/uscite ammessi, essa viene ignorata dal programma. Gli ingressi del multiplexer vanno collegati ad una sorgente digitale il cui valore puo' essere di 8 oppure 16 bit. Agli stessi ingressi digitali, in alternativa, e' possibile collegare una costante digitale.

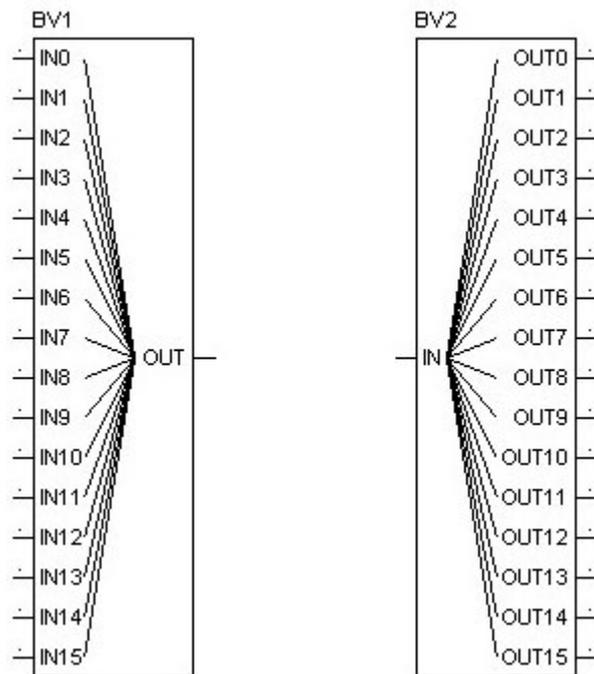
Terminale INx	ingresso della variabile o costante digitale al blocco;
Terminale OUT	uscita della variabile digitale dal blocco funzionale;
Terminale IN	ingresso della variabile digitale al blocco;
Terminale OUTx	uscita della variabile digitale dal blocco.
Select	selezione del terminale di ingresso oppure uscita.

Il multiplexer puo' essere rinominato con massimo 30 caratteri.



## Blocco funzionale de-codificatore **Bit > Byte < Bit**

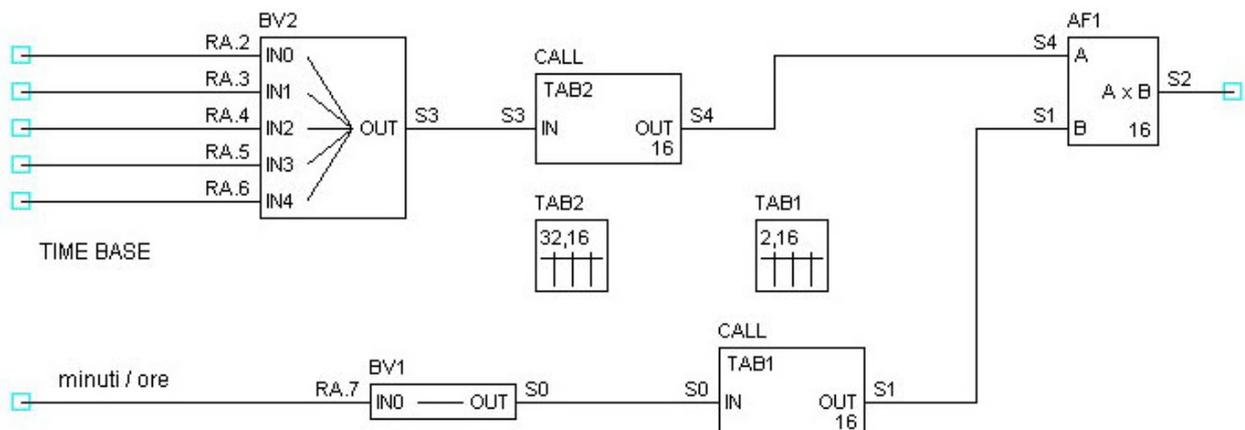
Questo blocco funzionale, svolge l'importante funzione di **codifica** o **decodifica** delle variabili digitali, semplificando notevolmente la formazione degli schemi funzionali, specialmente quando sono molti i blocchi logici da gestire. Lo schema seguente mostra il componente nella sua dimensione massima:



Il modulo di sinistra svolge il compito di **Codifica**: portando a livello alto uno o piu' terminali di ingresso in uscita si ottiene un determinato valore digitale, in dipendenza degli ingressi attivati.

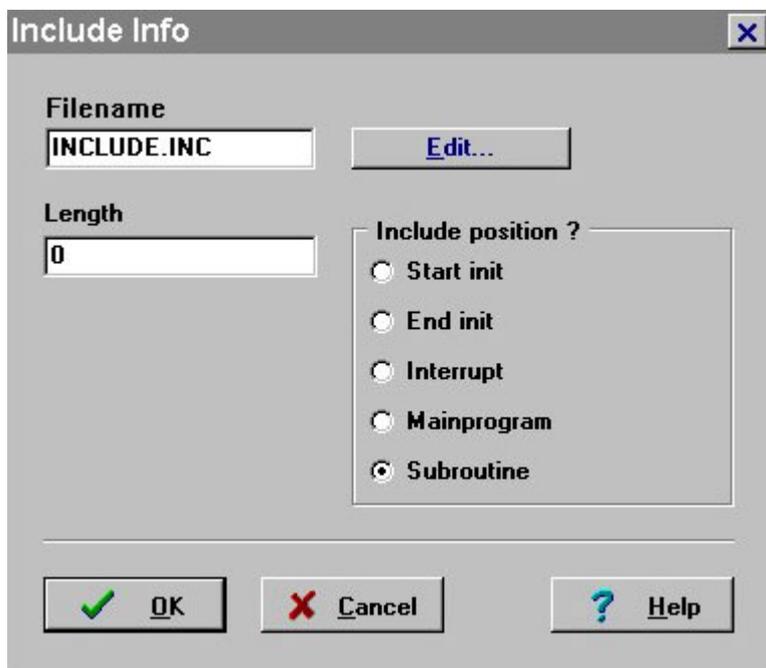
Nel modulo di destra si ha il funzionamento opposto, collegando al suo ingresso una variabile digitale, questa viene **Decodificata** ed il numero dei terminali di uscita attivi dipende dal valore della variabile.

Dal numero degli ingressi/uscite impiegate, dipende il valore massimo della variabile digitale impiegata che puo' essere di 8 oppure 16 bit. I terminali che non vengono utilizzati, non verranno considerati in via di compilazione del file sorgente. L'esempio circuitale che segue mostra l'impiego di due circuiti di codifica, utilizzati per selezionare il ciclo di un temporizzatore:



## Include

Utilizzando questa opzione, l'utente puo' inserire nel sorgente Parsic una soubroutine specializzata in una determinata funzione. Portandovi con il puntatore del mouse sopra il blocco include, cliccando con il tasto destro comparira' questa finestra di dialogo.



Nel riquadro **Include position ?** scegliere in quale posizione del file sorgente si desidera inserire il file Include;

Nel riquadro **Filename** specificare il nome del file ,che deve essere completato dall'estensione . INC ;

Utilizzare l'opzione **Edit...** per scrivere il file in formato testo ;

Nel riquadro **Length** inserire le dimensioni del file.

Nota: inserendo l'include nella posizione **Interrupt**, e' necessario che il file contenga un ciclo di temporizzazione . Diversamente il file include verrebbe ignorato dal programma.



## Comunicazione seriale. UART Base

Lo standard **RS232** definisce una serie di specifiche per la trasmissione seriale dei dati tra due dispositivi denominati **DTE** ( data terminal equipment) e **DCE** ( data communication equipment). Il **DCE** e' un dispositivo che si occupa di gestire una comunicazioni dati mentre il **DTE** e' un dispositivo che si occupa di generare o ricevere dati. In pratica l '**RS232** e' stata creata per connettere tra loro un terminale dati con un modem per la trasmissione a distanza dei dati generati.

Per la connessione tra due computer sono necessari cinque dispositivi di comunicazione: un computer **DTE**, un modem **DCE**, una linea di collegamento, ancora un'altro modem **DCE** ed infine un computer **DTE**.

Per collegare tra loro due computer ( microprocessori ) si puo' utilizzare una linea senza interporre alcun modem. Questo tipo di connessione si chiama **Null Modem** e si realizza con un cavo elettrico,invertente, in grado di scambiare i dati tra i due computer ( microprocessori ).

La comunicazione che Parsic e' in grado di gestire e' di tipo **RS232** asincrona, cioe' un tipo di trasmissione che non prevede affatto che la comunicazione tra due dispositivi diversi sia di tipo sincronizzata.

La trasmissione e la ricezione tra due dispositivi in comunicazione tra loro avviene per mezzo di due fili ed un collegamento comune riferito a massa. Il formato di trasmissione e' del tipo **8N1**:

### **8 bit data, nessuna parita', nessun bit stop**

Non tutti i microprocessori PIC sono dotati di porta di comunicazione seriale. Se, per errore, si seleziona un dispositivo non adatto all'impiego viene generato un segnale di errore.



### **Baud rate.**

La velocita' di trasmissione **Baud-rate** e' riferita al valore della frequenza di clock che corrisponde a quella dell'oscillatore al quarzo impiegato, cioe' 4MHz. Si raccomanda di utilizzare sempre un oscillatore al quarzo per le comunicazioni seriali. Portando il cursore del mouse sopra il blocco UART , cliccando con il tasto destro si apre la finestra di dialogo UART-Info.

## Baud-rate. Uart Base

### Predisposizione del modulo di comunicazione seriale.

with checksum  
 Self-defined  


---

Speed

 1200 Baud  
 2400 Baud  
 9600 Baud  
 19200 Baud

Self-defined  


---

Speed

 BRGH = 1  

SPBRG

La velocita' di trasmissione viene impostata nel riquadro Speed spuntando una delle velocita' gia preimpostate.  
 Se viene impostata l'opzione Self-defined, il Baud -rate viene definito con un semplice calcolo matematico.

Se BRGH viene mantenuto a 0 ( casella BRGH non selezionata ), per il calcolo del Boud-rate procedere in questo modo:

X =                    e' il valore da immettere in SPBRG ed e' compreso tra 0 e 225;  
 Fosc =                e' la frequenza esterna dell'oscillatore ( 4 MHz)  
 Baud-rate =     $Fosc / (64(X + 1))$

oppure  $(Fosc/64) / \text{Baud-rate} - 1$

Se BRGH viene mantenuto ad 1 ( casella BRGH selezionata) per il calcolo del Baud-rate procedere in questo modo:

Baud rate =     $Fosc / (16(X+1))$

oppure  $(Fosc/16) / \text{baud-rate} - 1$

Esempio:

Baud-rate 19200, frequenza esterna = 4 MHz, BRGH=1

$SPRG = (4000000 / 16) / 19200 - 1$

$SPRG = 250000 / 19200 - 1$

$SPRG = 13,0208 - 1$

$SPRG = \sim 12$

Il valore SPRG da inserire e' 12



## TX - RX Pin. Uart Base

Si puo' definire quale pin deve essere impiegato come terminale TX. Se viene usato il terminale hardware, definito dal produttore Microchip, questi viene denominato:

RB.2 per i dispositivi PIC a 18 pin;  
RC.6 per i dispositivi PIC a 20 e 40 pin.

Per il terminale RX valgono le stesse regole di cui sopra distinguendo diversamente i pin:

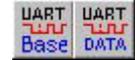
RB.1 per i dispositivi PIC a 18 pin;  
RC.7 per i dispositivi PIC a 20 e 40 pin.

## Standard RS485. Uart Base

Negli ambienti industriali ed in piccole reti di microcontrollori e' molto usato lo standard RS485. Lo standard RS485 e' uno standard half-duplex, cioe' la trasmissione dei dati e bidirezionale ma non contemporanea, occorre quindi che si indichi in modo esplicito se si vuole realizzare una ricezione o una trasmissione. L'utilizzo di un dispositivo hardware per la selezione della direzione ha il vantaggio di non richiedere cambiamenti software di comunicazione rispetto ad un normale cavo seriale null-modem, con le uniche avvertenze di disattivare il controllo hardware del flusso dei dati e di evitare di trasmettere e ricevere contemporaneamente.

Half-duplex (RS485)  
DIR-Pin

Selezionando lo standard RS485, nel riquadro **Half-duplex DIR-PIN**, si dovra' specificare quale pin di uscita al PIC si desidera utilizzare per il controllo Rx/Tx della porta seriale. Questo terminale e' normalmente a livello basso durante la ricezione del segnale, commuta a livello alto durante la trasmissione e resta a tale livello fino al termine. (vedi schema elettrico allegato).



## Checksum. Uart Base

Attivando questa opzione viene generato il cecksum (1 byte) al termine di ogni periodo di trasmissione.

### Ricezione :

codice di checksum inviato viene comparato, durante il processo di ricezione del segnale, con il valore di checksum calcolato dal microprocessore. Se la comparazione e' valida, i dati sono trasferiti nella variabile indicata a destra della tabella sottostante (riceive...):



Nel caso sia di interesse,il calcolo del codice di cecksum e' facilmente ricavabile dalla seguenti formule:

1. CS = 170 (HEX = \$AA)
2. CS = CS + 1st byte
3. CS = CS + 2nd byte
4. ECC.

CS= checksum

Esempio:

Il contatore ZV5 presenta un valore di conteggio = 1000.

Nel riquadro di sinistra,del modulo di ricezione, vedremo indicati:

ZV5 ( che rappresenta il low-byte del contatore, che al momento contiene il valore 232 (\$E8))

ZV5\_HI ( che contiene l'high-byte del contatore = 3)

il seguente calcolo mostra come ricavare il contenuto del contatore ZV5 ( $3 \times 256 + 232 = 1000$ )

il valore di checksum sara' cosi' determinato:

$CS = 170 + 232 + 3 = 405$   $405 - 256 = 149$  rappresenta il valore del (8) bit piu' basso trasmesso.

La sequenza di trasferimento conterra' i seguenti valori <232> <3> <149>



## UART data

Utilizzando questo modulo e' possibile ricevere o trasmettere i dati sulla linea seriale. Il modulo si presenta con due terminazioni :

il terminale ACT e' il pin di ingresso/uscita dei dati;

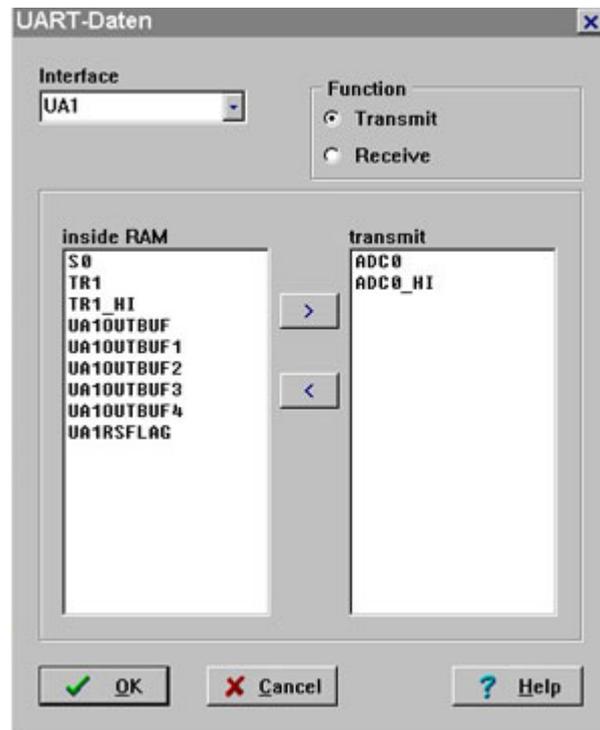
il terminale Enable e' il pin che abilita la ricezione/trasmissione dei dati.

Per tutto il periodo di trasmissione il terminale ACT-Out rimane a livello logico alto. Durante la fase di ricezione il terminale ACT-Out rimane a livello logico alto fino alla corretta ricezione del dato.

Utilizzando l' ACT si puo' determinare quale dei moduli, trasmittente-ricevente, dovra' restare attivo durante il trasferimento dei dati. Parsic utilizza questo terminale per il controllo del flusso dei dati.

### Trasferimento dei dati.

Le variabili del programma sono contenute in una lista, specificata nella UART-data.



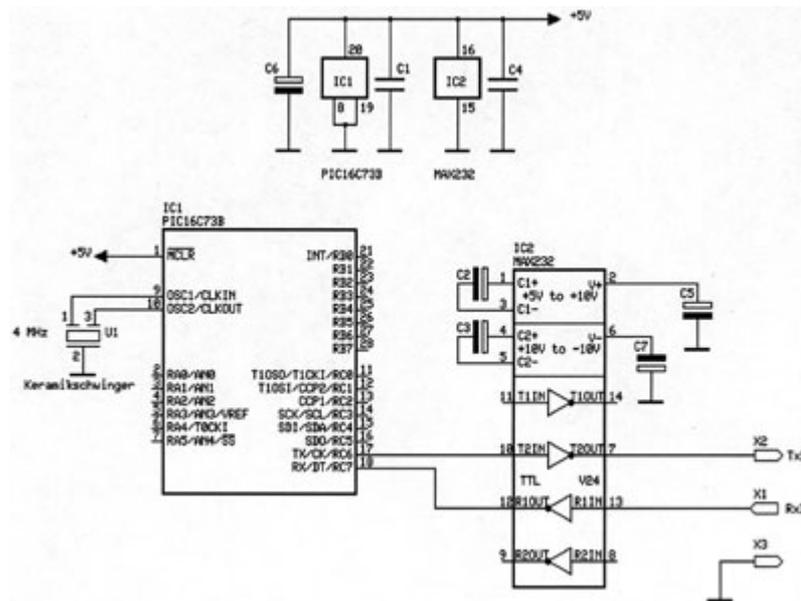
Le valori che si desidera trasmettere dovranno essere trasferiti nella tabella di destra, selezionando con il puntatore del mouse le variabili contenute nel riquadro Inside Ram, agendo sul pulsante > .

La trasmissione dei dati avverra' nello stesso ordine in cui essi vengono trasferiti nel riquadro Transmit. La trasmissione si attiva quando viene applicato un impulso positivo al terminale ENable ed al buffer di uscita saranno inviati i dati selezionati piu' il checksum. E' sufficiente un solo impulso per attivare la trasmissione e, durante la stessa, altri impulsi di abilitazione non saranno considerati fino al termine dell'operazione.

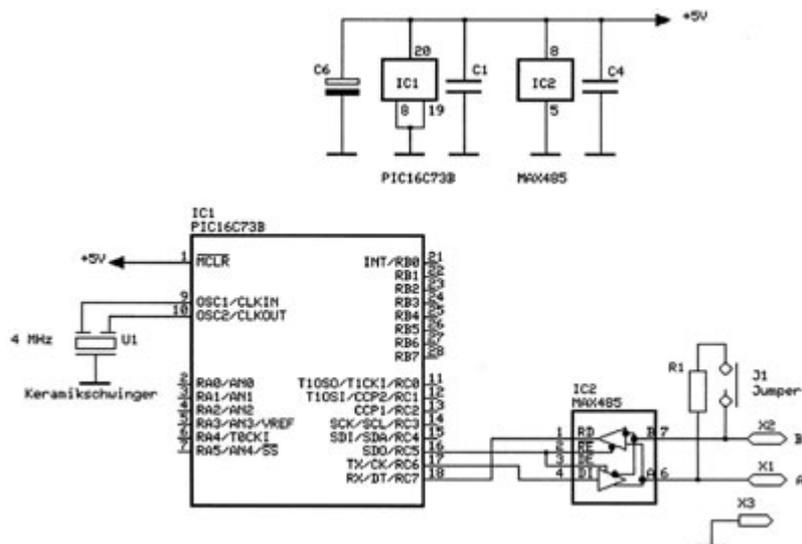
## Ricezione dei dati Uart Data.

I dati che si vogliono ricevere saranno trasferiti nel riquadro Receive, con le stesse modalita' specificate sopra. La ricezione dei dati ha inizio applicando al terminale ENable un impulso positivo e, durante la fase di ricezione, eventuali altri impulsi applicati al terminale ENable saranno ignorati. Se durante la ricezione si manifesta un'errore nei dati o di checksum, la ricezione viene interrotta ed il terminale ACT-Out si porta a livello logico basso.

## Schema elettrico di un collegamento seriale RS232



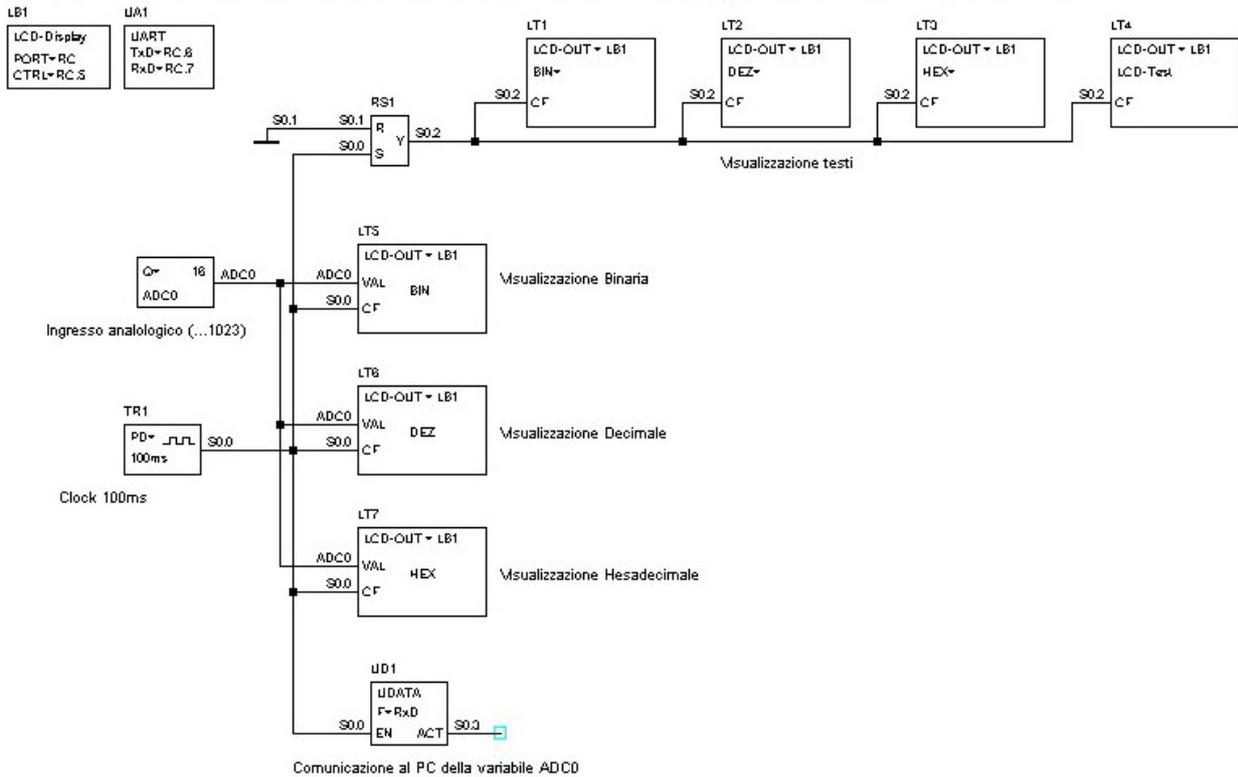
## Schema elettrico di un collegamento seriale RS485



Esempio di una trasmissione RS232 .

Il PIC viene utilizzato per trasmettere la variabile analogica ADC0 ad un PC.  
 Si noterà, nello schema funzionale, che il generatore di clock viene utilizzato contemporaneamente per il refresh dei display LCD e per attivare la trasmissione seriale RS232.

### Trasmissione seriale RS232 di una variabile analogica e visualizzazione LCD





### L'editor di testo.

Durante la costruzione dello schema funzionale si consiglia il frequente uso di questo editor. Consente di aggiungere commenti e segnalazioni lungo il tracciato dello schema. Inoltre sarà di valido aiuto quando saranno ripresi schemi archiviati da tempo di cui non si ricordano più i particolari di progetto. Queste tracce non occupano spazio nella memoria del PIC pertanto è possibile utilizzarle a proprio piacimento. Per l'impiego, è sufficiente posizionare l'etichetta vicino al componente o sezione di impianto che si desidera commentare, dopo avere selezionato la funzione sulla toolbar. Cliccate con il tasto destro del mouse sopra l'etichetta; quando appare la finestra di dialogo inserite la dicitura nella casella **Commento**.



### I potenziali di riferimento.

Utilizzando questa funzione si possono collegare ai terminali di ingresso dei blocchi funzionali i riferimenti di potenziali **GND** e **Vcc**.

Questi riferimenti non sono altro che lo **0 (zero)** oppure **1 (uno)** logico digitale. Servono, ad esempio, a forzare un livello logico su una porta di ingresso di un operatore booleano, a mantenere lo zero logico al terminale di reset di un contatore, ecc.

Dopo aver selezionato il riferimento di potenziale, portatevi con il puntatore del mouse su di esso, operate sul tasto destro del mouse e, quando compare la finestra di dialogo, definite l'orientamento da dare all'oggetto e la sua polarità'.

## **Indice degli argomenti**

<b>Introduzione:</b>	<b>pg. 3</b>
• indicazioni di carattere generale	
• requisiti di sistema	
• Parsic versione Demo	
• Parsic versione V3.39	
<b>Procedura per l'installazione di Parsic</b>	<b>pg. 5</b>
• indicazioni di carattere generale	
• impostazione della lingua operativa	
<b>Il menu Setting</b>	<b>pg. 7</b>
• le Environment	
• Handling	
• Presentation	
• Refresh	
• Microcontroller	
• L'oscillatore	
• Configurazione del microcontroller	
• Il master clear MCLR	
• Colori dello schema elettrico	
• I programmi accessori	
• I Tools	
• I dati di progetto	
<b>Funzionamento del MOUSE</b>	<b>pg. 15</b>
<b>I blocchi funzionali : la Toolbar</b>	<b>pg. 16</b>
<b>Load dei file</b>	<b>pg. 18</b>
• la selezione dello zoom	
• la Funktion Plan	
<b>Il menu Window</b>	<b>pg. 19</b>
<b>Selezione della griglia e dei collegamenti virtuali</b>	<b>pg. 21</b>
• il comando Delete	
• il comando Save	
• il file sorgente ASM	
• il file compilato .HEX	
• il comando SIM	
<b>Costruzione dello schema a blocchi</b>	<b>pg. 23</b>
• esecuzione pratica	
• come posizionare un blocco logico	
• come rimuovere un blocco logico	
• come collegare una linea virtuale	
• come definire una linea virtuale	
• come attivare la simulazione	
• come visualizzare le variabili ed il sorgente ASM	
<b>Come salvare il file sorgente</b>	<b>pg. 28</b>
• come salvare il file di testo ASM	
• come ottenere il file compilato .HEX usando MPASM	
<b>Descrizione dei blocchi logici di Visual Parsic</b>	<b>pg. 31</b>
• definizioni	
• identificativi di collegamento	

## denominazione dei blocchi

<b>I blocchi logici funzionali : gli operatori booleani</b>	<b>pg. 34</b>
<b>logiche con operatore matematico</b>	<b>pg. 35</b>
<b>comparatore digitale</b>	<b>pg. 38</b>
<b>comparatore analogico-digitale Schmitt-trigger</b>	<b>pg. 39</b>
<b>selezionatore di ingressi o costante analogica</b>	<b>pg. 40</b>
<b>conteggio di eventi contatori ad 8 bit</b>	<b>pg. 41</b>
• <b>conteggio di eventi contatori UP/DOWN 8-16 bit</b>	
<b>Schift-register</b>	<b>pg. 42</b>
<b>Flip-flop Set-Reset</b>	<b>pg. 43</b>
<b>generatore di clock Timer</b>	<b>pg. 44</b>
<b>monostabili su evento</b>	<b>pg. 45</b>
<b>monostabile One-Shot</b>	<b>pg. 46</b>
<b>condizionatore di impulsi Impulse-in</b>	<b>pg. 47</b>
<b>condizionatore di impulsi Impulse-out</b>	<b>pg. 48</b>
• <b>generatore impulsi PWM</b>	
<b>filtro digitale Limiter</b>	<b>pg. 49</b>
<b>gestione della EEprom</b>	<b>pg. 50</b>
<b>gestione dei moluli LCD</b>	<b>pg. 55</b>
<b>la funzione SLEEP</b>	<b>pg. 60</b>
<b>la funzione TABLE e CALL</b>	<b>pg. 61</b>
<b>il Multiplexer</b>	<b>pg. 64</b>
<b>il de-codificatore BV</b>	<b>pg. 65</b>
<b>l' Include</b>	<b>pg. 66</b>
<b>la comunicazione seriale RS232</b>	<b>pg. 67</b>
<b>l'editor</b>	<b>pg. 68</b>
<b>i potenziali di riferimento</b>	<b>pg. 74</b>