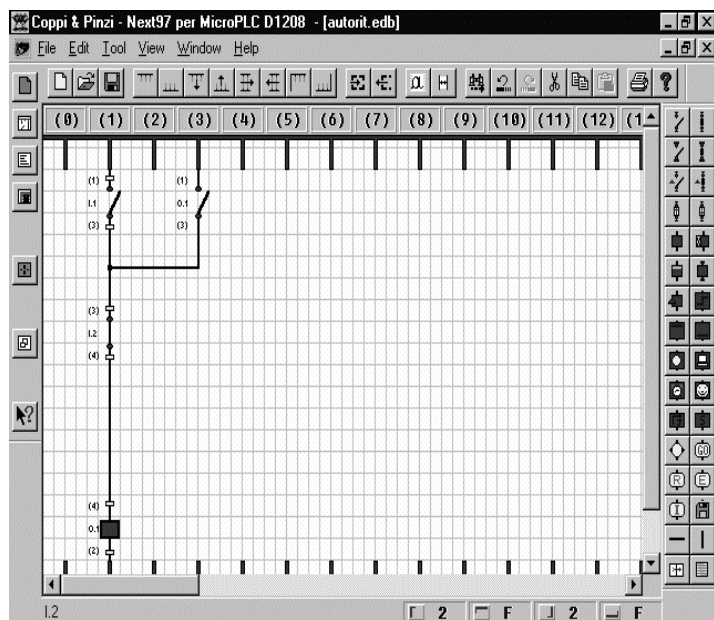


ESEMPIO 1 Circuito ad autoritenuta

Uno degli esempi più immediati e semplici da realizzare è senza dubbio il circuito ad autoritenuta.

In questo applicativo vengono utilizzati N° 2 ingressi e N° 1 uscite. Lo schema è quello rappresentato in figura 1.1, l'esempio può essere caricato utilizzando il programma NEXT97 dalla directory C&P96\ESEMPI\AUTORIT.EDB.



Descrizione:

INGRESSO 1 = *START* INGRESSO 2 = *STOP*

Premendo il pulsante di start viene abilitata l'uscita 1 che si automantiene per mezzo del proprio contatto posto in parallelo all'ingresso di start.

Premendo il pulsante di stop posto in serie alla abilitazione l'uscita 1 verrà disabilitata.

Va sottolineato che, realizzando il cablaggio esterno degli ingressi, non sarà necessario utilizzare un pulsante di STOP normalmente chiuso; infatti è il programma che determina lo stato iniziale degli ingressi.

Pur utilizzando il contatto aperto anche per l'ingresso di STOP basterà rappresentarlo normalmente chiuso nello schema elettrico per avere l'effetto desiderato.

COME REALIZZARE QUESTO ESEMPIO:

Aprire il programma NEXT97 fornito in dotazione al PLC. La prima cosa da fare quando si inizia un nuovo progetto è quella di definire la grandezza del foglio di lavoro. Mentre è sempre possibile, anche successivamente, dimensionare la quantità di colonne e quindi la larghezza del disegno questo non lo è per quello che riguarda il numero di righe, e cioè, la quantità di componenti che possono essere

inseriti in senso verticale. Per fare questo è necessario, solo la prima volta all'inizio, selezionare dal menu *SetUP*, la voce *Impostazioni*. Dopo questa selezione comparirà il seguente dialog. Inserire nella casella *Ampiezza Foglio Disegno* un numero che dipende dalla risoluzione della vostra scheda VGA (nel caso di 640x480 $N=8$), questo servirà a visualizzare i due fili comuni superiore ed inferiore contemporaneamente senza il bisogno di utilizzare i comandi di scorrimento verticale. Sul resto del dialog è anche possibile personalizzare l'intestazione dei fogli disegno in fase di stampa, semplicemente inserendo i dati richiesti. Confermare cliccando su *OK*. Selezionare *File*, *nuovo*, *disegno*, dal menu. Dopo questa operazione si aprirà la pagina di disegno NEXT dove sarà possibile inserire il disegno rappresentato in FIG 1.1.

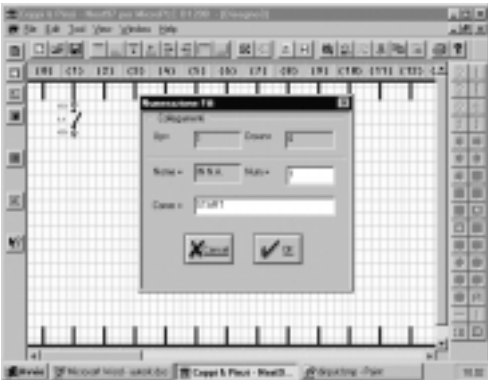
Dopo questa breve descrizione vedremo come disegnare con il programma NEXT realizzando lo schema dell' autoritenuta. Si ricorda, comunque, che di tutti gli esempi applicativi di questo manuale esistono i file sia in formato disegno *.edb sia in linguaggio pseudoassembler *.esm che in formato compilato per plc *.epc.

Per comprendere il significato degli strumenti di disegno messi a disposizione dal programma NEXT si può fare riferimento alla mappa degli strumenti plc DI1208 raffigurata nell'allegato A. Come indicato nella



descrizione dell'esempio, per la realizzazione di questo circuito si utilizzano due ingressi ed una uscita. Tali componenti verranno prelevati dagli strumenti posizionati sulla destra della pagina.

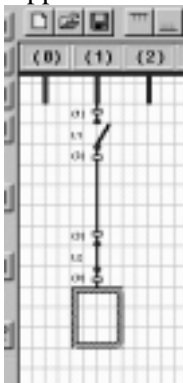
Per prima cosa preleveremo un contatto normalmente aperto. I contatti di ingresso si differenziano dai contatti generici perché hanno dei piccoli rettangoli blu ai due estremi superiore ed inferiore. Per prelevare il contatto (ed ogni tipo di strumento) è sufficiente posizionarsi al centro con il mouse e poi cliccare una volta. Dopo questa operazione la barra degli strumenti diventerà completamente grigia. A questo punto spostare la freccia sotto alla colonna 1 e cliccare con il pulsante sinistro del mouse. In questo modo avremo piazzato il primo componente (il contatto di ingresso NA). Cliccando due volte con il pulsante sinistro sul componente installato si potrà accedere al dialog di impostazione dati come raffigurato nella foto sottostante.



All'interno di questo dialog è possibile inserire il numero di ingresso associato al contatto ed un commento che identificherà la funzione del contatto stesso (per esempio START).

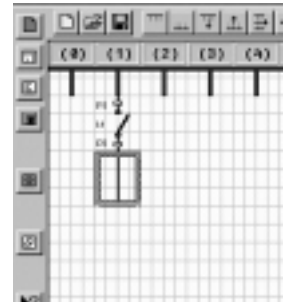
Adesso bisognerà inserire un filo verticale sotto all'ingresso N°1. Il filo può essere prelevato dalla solita barra strumenti e con le stesse modalità spiegate precedentemente. Adesso portarsi con il puntatore del mouse sotto al contatto precedentemente inserito e circa a metà del rettangolo evidenziato sulla foto di destra.

A questo punto cliccando con il pulsante sinistro del mouse otterrete il piazzamento del filo verticale come raffigurato. Adesso è possibile prelevare il contatto normalmente chiuso e posizionarlo sotto al filo verticale procedendo sempre allo stesso modo. Il risultato ottenuto sarà quello rappresentato nel disegno raffigurato sotto a sinistra.

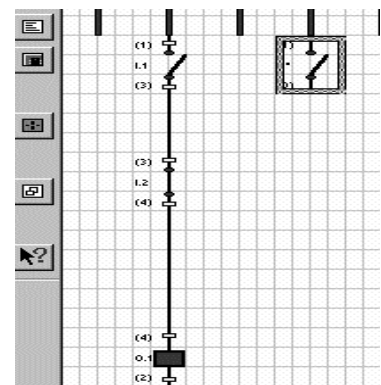


Dopo aver inserito il numero di ingresso ed il relativo commento come per l'ingresso 1 possiamo passare all'inserimento della bobina di uscita. Come per gli ingressi, le uscite si differenziano dalle bobine generiche perché contraddistinte alle estremità da due rettangolini blu. Prelevare per mezzo del mouse la bobina uscita, portarsi immediatamente sotto all'ingresso 2 (nella casella vuota raffigurata), premere il pulsante sinistro del mouse.

Il risultato sarà il piazzamento della bobina in fondo al disegno con l'aggiunta automatica di un nuovo filo. L'ultimo componente da piazzare è il contatto di automantenimento **O1** in parallelo all'ingresso N°1. Questo contatto può essere prelevato cliccando sul contatto NA interno generico (il primo in alto a sinistra nel tool strumenti), e, utilizzando le stesse modalità precedenti può essere collocato



nella colonna N° 3 in alto a destra di **II** come rappresentato nel disegno qui a fianco. Il contatto così piazzato non corrisponde a nessun componente se non si provvede ad assegnarli un nome ed un indirizzo, per far questo è necessario, come per gli ingressi, cliccare due volte sul componente con il pulsante sinistro del mouse sino a veder comparire sul video la finestra di dialogo qui sotto raffigurata. Con questo dialog è possibile impostare il tipo ed il numero di contatto interno che interessa.



Per selezionare il contatto di uscita N°1 è sufficiente cliccare sul cerchietto *Out* ed inserire il N° 1 nella casella *NUM=*.

Convalidando con il pulsante *OK* otterremo l'assegnazione al contatto interno precedentemente inserito. L'ultima cosa da fare è il collegamento del contatto all'ingresso N°1 così da poter espletare alla funzione di autoritenuta dello schema generale di FIG. 1.1.

Per fare questa operazione bisognerà tenere conto di una regola molto importante che varrà poi anche per i collegamenti verticali più lunghi di un singolo filo.



IMPORTANTE : Per il prolungamento di qualunque tipo di filo sia in senso orizzontale che verticale non è necessario utilizzare altri fili ma solo allungare quello esistente procedendo nella maniera seguente:

- *Prolungamento di un filo verticale:* cliccare sul filo interessato, la freccia del puntatore mouse si trasformerà in una matita, portarsi ora con il mouse in corrispondenza del componente da collegare (meglio se leggermente all'interno del componente stesso) e cliccare nuovamente, si vedrà il filo allungarsi dal componente superiore, dove era stato collocato, al componente da collegare.
- *Prolungamento di un filo verticale con diramazione verso destra:* cliccare sul filo interessato, la freccia del puntatore mouse si trasformerà in una matita, portarsi ora con il mouse in orizzontale verso destra e sotto al componente da collegare, cliccare nuovamente per vedere il filo che si allunga sino alla matita. Eseguire nuovamente la procedura dall'ultima posizione raggiunta e spostarsi verso l'alto sotto al componente interessato (meglio se leggermente all'interno del componente stesso) per cliccare nuovamente e vedere finalmente l'ultima parte del filo collegarsi al componente.

Una volta ricordate queste regole il disegno risulterà abbastanza semplice e rapido da realizzare.

Con questa ultima procedura abbiamo realizzato il nostro primo disegno che ora è pronto per essere compilato e inviato al PLC.

Prima di compilare questo disegno bisognerà salvarlo selezionando *File, Salva con nome*, dal menu in alto. Sarà poi sufficiente inserire il nome del file ricordandosi di mantenere l'estensione .EDB, che sta ad indicare

che quello salvato è un file disegno, e facendo attenzione a collocarlo all'interno della sottodirectory desiderata. Tutte queste procedure seguono le stesse modalità di tutti i programmi che funzionano in ambiente WIN95.

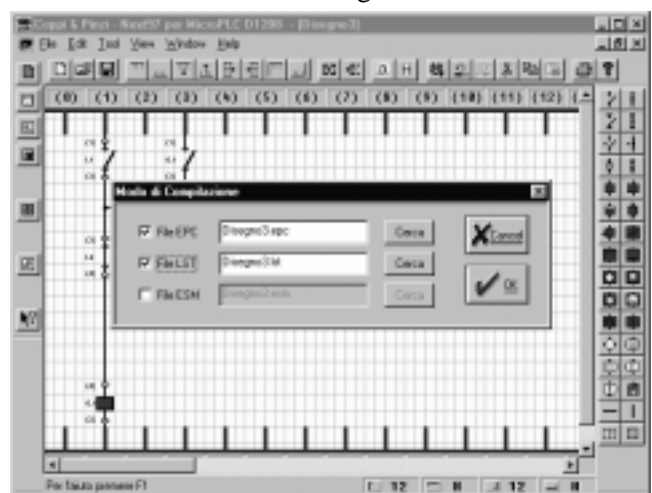
Per passare alla fase di compilazione è sufficiente cliccare sull'ultimo strumento in basso a destra raffigurante un foglio, ad operazione effettuata comparirà la finestra di dialog sottostante.

Qui è possibile inserire il nome del file compilato (file avente estensione .EPC, il nome del file List (contenente il linguaggio macchina) ed il nome del file assembler avente estensione .ESM, è da ricordare che gli ultimi due file sono opzionali ed il loro inserimento è possibile cliccando sulla apposita casellina che li contraddistingue.

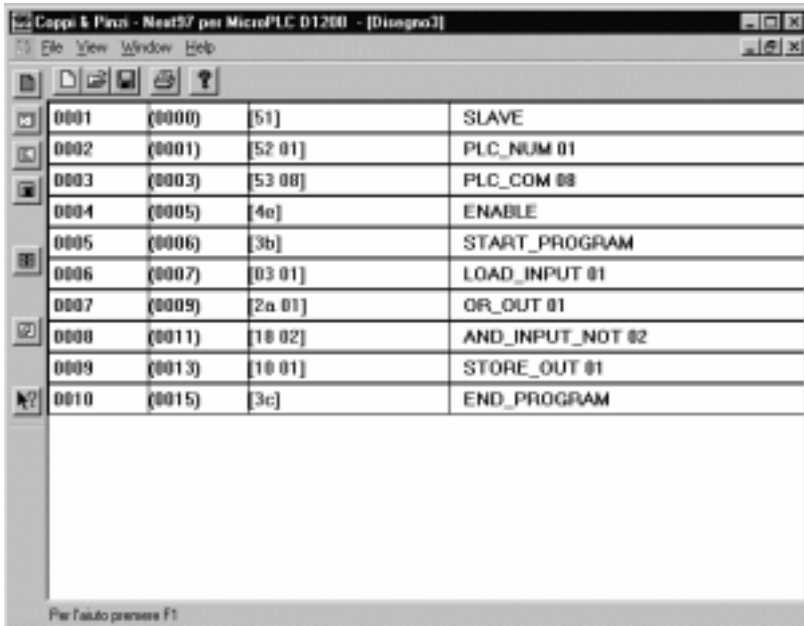


La conferma sul tasto OK lancerà la fase di compilazione. La procedura appena seguita ritornerà una pagina raffigurante l'elenco istruzioni in formato pseudo assembler se l'esito della compilazione è stato positivo. Il ritorno alla pagina disegno con la finestra riportante lo stato degli errori segnaleranno un eventuale esito negativo della compilazione.

Per semplicità abbiamo riportato nella pagina successiva la finestra riguardante la compilazione avvenuta con esito positivo. Questo ci servirà ad esaminare lo stesso programma realizzato per mezzo dello pseudo assembler. Riporteremo nel capitolo "trattamento degli errori" il caso in cui lo schema contiene uno o più errori ed esamineremo le modalità per risolverli.



Quella riportata sotto è la finestra che compare a compilazione avvenuta. Viene rappresentato il listato nel linguaggio pseudo assembler (la colonna 4), il codice oggetto in linguaggio macchina (colonna 3), l'indirizzo del contatore di programma che risulta utile quando esistono istruzioni di jump per vedere a quale parte di programma salta il microprocessore (colonna 2) ed infine il numero di riga del programma (colonna 1).



Linea	Indirizzo	Codice	Istruzione
0001	(0000)	[51]	SLAVE
0002	(0001)	[52 01]	PLC_NUM 01
0003	(0003)	[53 08]	PLC_COM 08
0004	(0005)	[4e]	ENABLE
0005	(0006)	[3b]	START_PROGRAM
0006	(0007)	[03 01]	LOAD_INPUT 01
0007	(0009)	[2a 01]	OR_OUT 01
0008	(0011)	[18 02]	AND_INPUT_NOT 02
0009	(0013)	[10 01]	STORE_OUT 01
0010	(0015)	[3c]	END_PROGRAM

Una volta esaminata, questa finestra può essere chiusa per ritornare al disegno precedentemente realizzato.

A questo punto si è creato il programma **AUTORIT.EPC**, programma in linguaggio macchina pronto per essere caricato dentro al PLC.

Per poter eseguire il programma appena realizzato bisogna passare al programma SIMULA che permette il dialogo tra Personal computer e PLC DI1208, questo può essere fatto senza chiudere il programma NEXT97, ma semplicemente riducendolo ad icona (procedura comune a tutti i programmi su piattaforma WIN95), per poi passare al program manager ed aprire il programma SIM96.

Alla apertura il programma SIM96 presenterà la finestra riportata sotto. Prima di proseguire con il programma bisognerà collegare il PLC alla alimentazione ed alla porta seriale come spiegato nel manuale d'uso. A collegamenti ultimati bisognerà procedere nella seguente maniera:

- Selezionare il menu *Azione, Interroga..* Comparirà una finestra di dialogo per l'inserimento del numero di PLC e della velocità seriale di trasmissione/ricezione dati.
- Impostare (di solito sono già i valori predefiniti) velocità seriale = 57600 Baud e numero plc = 1, confermare con il tasto OK. La comparsa del display orologio sulla finestra ORA PLC e la illuminazione



dei led indicanti gli INGRESSI e le USCITE sulle rispettive finestre sta ad indicare che il collegamento tra personal computer e PLC è attivo, e quindi che il programma compilato può essere trasferito al plc.

Selezionare nuovamente il menu *Azione*, seguito da *Carica programma*, inserire il nome del file compilato precedentemente ed avente estensione .EPC (autorit.EPC), confermare con il tasto OK ed attendere il caricamento che sarà evidenziato dalla finestra di caricamento raffigurata qui sotto. Lo scorrimento di una barra orizzontale verde seguita da una barra orizzontale rossa e la successiva chiusura della finestra di dialogo confermeranno

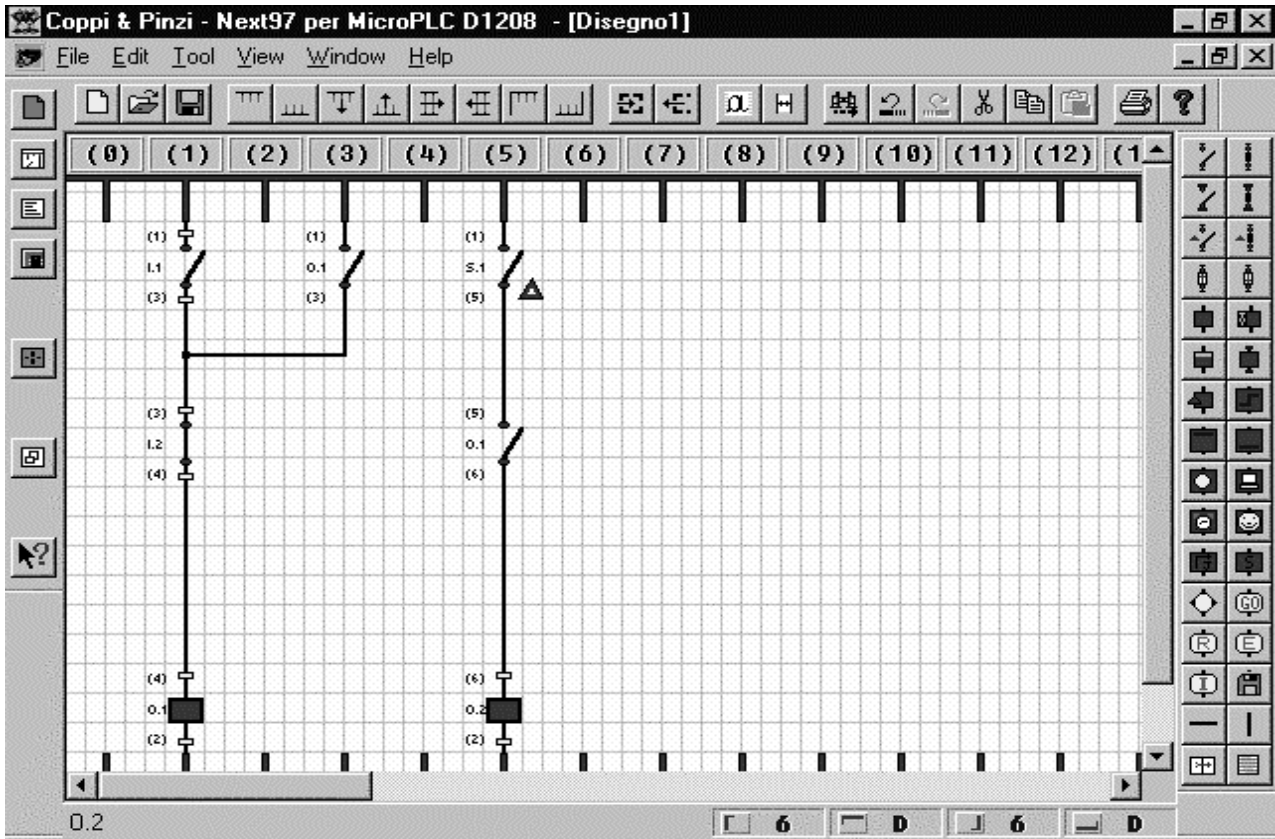
l'esito riuscito del caricamento.

A questo punto non resta che spostarsi con il mouse sul selettore RUN (in alto a destra) e trascinarlo verso l'alto tenendo premuto il pulsante sinistro del mouse. In questo modo il plc è entrato in modo RUN, questa condizione può essere verificata sul PLC osservando che il led verde RUN è acceso fisso, mentre le segnalazioni ERR e ENABLE dovranno rimanere spente. Non rimane che verificare la funzionalità del programma collegando i due pulsanti esterni di START e STOP per vedere la spia rossa della uscita N°1 accendersi premendo lo START e spegnersi premendo lo STOP.



ESEMPIO 2 Circuito ad autoritenuta con lampeggio su uscita 2

Per acquisire un po' di familiarità con il programma NEXT97 si consiglia di aggiungere al precedente programma una variante che consiste nell'inserimento di una uscita lampeggiante (USCITA N° 2) utile come



segnalazione per lo stato di marcia di un eventuale dispositivo.

Per facilitare il compito ai programmatori si può attingere ad una serie di bit interni, denominati contatti speciali e bobine speciali, i quali sono contraddistinti da un piccolo triangolo. In questo esempio viene utilizzato un contatto speciale che può essere prelevato dalla barra strumenti e piazzato sul punto prescelto seguendo le stesse modalità dell'esempio precedente.

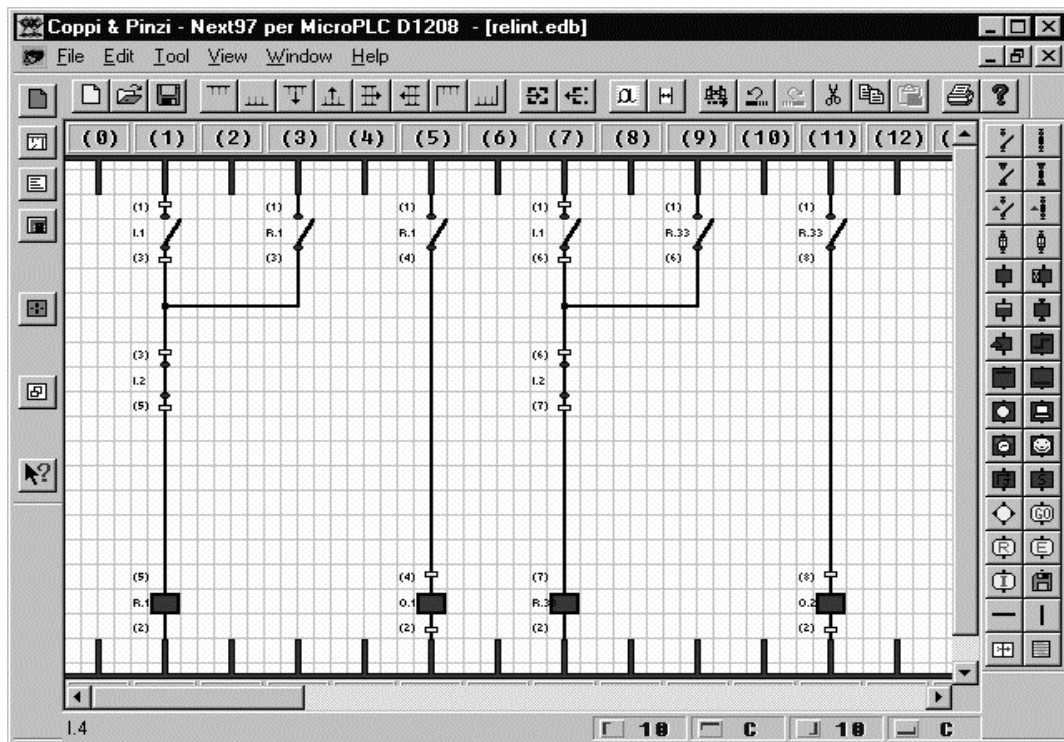
A piazzamento effettuato è possibile accedere al data base dei contatti speciali esercitando un doppio click con il pulsante sinistro del mouse sul componente piazzato. Si aprirà, così, una finestra di dialog indicante l'elenco completo dei contatti speciali. Per il nostro esempio sceglieremo il bit speciale **S1** (**contatto temporizzato che inverte il suo stato ogni 0,5 secondi**). Proseguendo nello stesso modo possiamo completare la realizzazione della parte mancante sino ad ottenere il disegno rappresentato in figura.

Una volta ultimato, lo schema può essere ricompilato e trasferito nuovamente nel PLC.

Si ricorda che il passaggio da NEXT97 e SIM96 può avvenire semplicemente riducendo ad icona l'uno e/o l'altro programma, in questo modo le eventuali modifiche ed i successivi trasferimenti risulteranno molto più rapidi.

ESEMPIO 3 Utilizzo dei relè interni

All'interno di un plc DI1208 vengono emulati 64 relè interni (bit) che possono essere utilizzati dal programma NEXT97. Ogni relè interno può avere un numero di contatti aperti/chiusi senza alcuna limitazione se non quella imposta dalla capacità del programma utente che può contenere sino a 700 byte di memoria programma. Dei 64 relè disponibili solo il secondo gruppo (dal 33 al 64) non sono mai azzerati dal sistema operativo, così questi mantengono il loro stato anche alla ripartenza del PLC dopo una interruzione avvenuta per mancanza di energia elettrica. Questa funzione risulta particolarmente utile quando è necessario mantenere traccia di una sequenza dopo una imprevista interruzione del sistema. Il primo gruppo di relè



(dall'1 al 32) vengono, invece, volutamente forzati a zero all'accensione del PLC.

A titolo di esempio abbiamo realizzato lo schema qui riportato. Sono stati realizzati due circuiti identici che agiscono con gli stessi ingressi (1 START, 2 STOP) su due uscite diverse.

La uscita N° 1 viene comandata dal relè interno N° 1 che si automantiene dopo che noi avremo premuto l'ingresso START.

La uscita N° 2 viene comandata dal relè interno N° 33 (primo del gruppo di relè memorizzati), che si automantiene dopo che noi avremo premuto lo START.

Non appena premuto il pulsante di START le 2 uscite si accenderanno istantaneamente.

Non appena premuto lo STOP le 2 uscite si spegneranno istantaneamente.

Proviamo, adesso, a riaccendere le 2 uscite premendo nuovamente lo START. Una volta eseguita questa operazione possiamo togliere la tensione di alimentazione al PLC, attendiamo alcuni secondi e alimentiamo nuovamente il sistema.

Attendiamo almeno 15sec perchè il PLC entri in modo RUN (fase di autodiagnostica contraddistinta dal lampeggio alternato dei LED RUN ed ENABLE), raggiunta la fase di RUN possiamo constatare che la uscita N°1 rimane spenta, questo perché il relè N°1 viene riazzerato dopo una ripartenza. La uscita N° 2, invece, si riaccende, questo perché è associata al relè N° 33 (lo stesso sarebbe accaduto per qualunque relè nel range da 33 a 64) che mantiene memorizzata la condizione precedente.

Potrete caricare il file RELINT sia in formato disegno (RELINT.EDB) che il file compilato (RELINT.EPC).

Entrambi i file si trovano dentro alla sottodirectory ESEMPI.

ESEMPIO 4 come avviene la scansione di programma

Il PLC non esegue in tempo reale il programma contenuto al suo interno. Egli esegue una istruzione per volta scandendo lo schema dalla sinistra verso destra e una colonna doppi l'altra. Raggiunta la fine dello schema il programma ritorna all'inizio per iniziare una nuova scansione e così di seguito sino a che il programma rimane nello stato RUN.

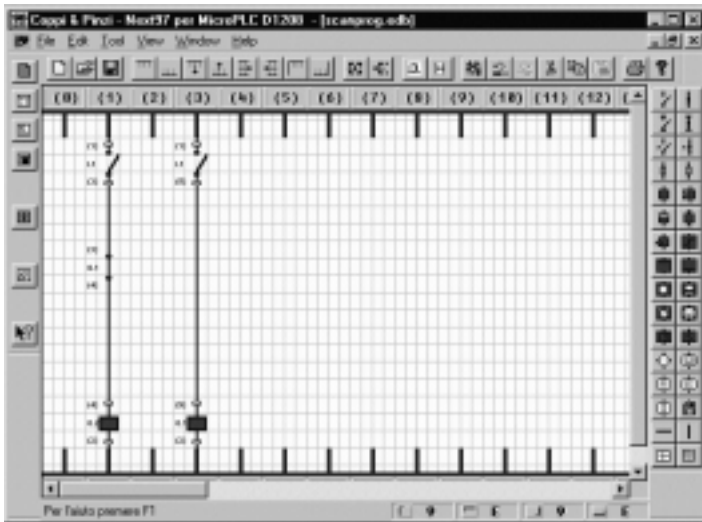
All'inizio di ogni ciclo di programma il compilatore inserisce una istruzione chiamata START_PROGRAM. Durante l'esecuzione di questa istruzione il PLC aggiorna lo stato degli ingressi, trasferendo lo stato HARDWARE di questi ultimi all'interno della RAM del PLC, per tutta una scansione il PLC terrà conto dei valori letti in questa locazione per condizionare tutti gli altri bit dipendenti dagli ingressi.

Alla fine di ogni ciclo di programma il compilatore inserisce una istruzione chiamata END_PROGRAM.

Durante l'esecuzione di questa istruzione viene copiato il contenuto delle locazioni di memoria con al suo interno lo stato delle uscite nelle uscite HARDWARE del PLC.

In pratica se fosse possibile realizzare un programma senza START_PROGRAM non potremmo leggere gli ingressi, analogamente un programma senza END_PROGRAM non potrebbe attuare nessuna uscita.

Il programma rappresentato qui sotto simula un FRONTE DI SALITA e si presta a capire come funziona la scansione di programma.

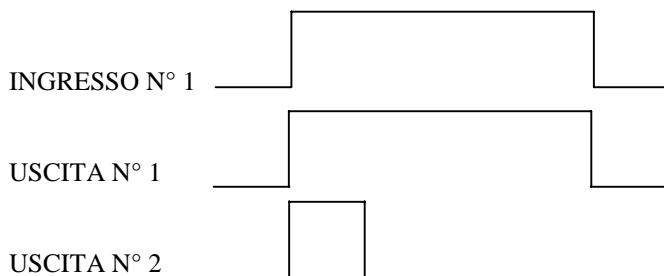


Il FRONTE DI SALITA non è altro che un impulso che viene generato per una sola scansione di programma.

Come si è detto precedentemente il PLC esegue una colonna alla volta da sinistra verso destra, al termine dell'ultima colonna il contatore di programma ritorna alla prima istruzione.

Supponiamo di aver chiuso l'ingresso N° 1. Sulla prima colonna (N° 1) viene letto lo stato di I1, il valore ottenuto dalla lettura di questo viene messo in AND logico con il contatto NC della USCITA 1 che ancora non è abilitata. Il risultato di questa operazione abiliterà la uscita N° 2. Successivamente viene scandita la

colonna N° 3 dove troviamo ancora l'ingresso 1 che stavolta abilita direttamente la uscita N° 1. Al termine della prima scansione avremo tutte e due le uscite abilitate. Inizio della seconda scansione. Si suppone che l'ingresso 1 sia ancora chiuso. Il contatto della uscita N° 1 (abilitata nella scansione precedente) si apre interrompendo il collegamento verso la uscita N° 2 la quale si disabiliterà rimanendo in questo stato finché il pulsante associato all'ingresso N° 1 non viene rilasciato. Osservando le uscite si noterà che la N° 1 rimane accesa per tutto il tempo che I1 rimane premuto, mentre la N° 2 farà un piccolo lampeggio e poi rimarrà spenta sino a che I1 non viene rilasciato e poi nuovamente premuto.

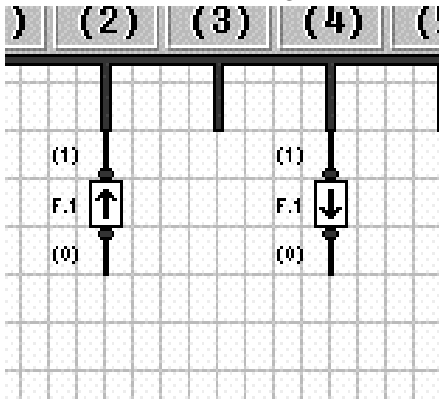


Il diagramma temporale a fianco riportato rappresenta lo stato delle 2 uscite in funzione dell'ingresso N° 1

Il tempo che l'uscita N° 2 rimane allo stato logico è il tempo di scansione del programma all'interno del PLC.

ESEMPIO 5 I fronti di salita e di discesa

Nell'esempio N° 3 si è parlato dei fronti di salita e si è visto come è possibile crearli utilizzando i vantaggi della scansione di programma. Va però detto che questi dispositivi sono già stati implementati nel software del PLC e quindi non è necessario crearli. Ogni PLC DI1208 contiene al suo interno 16 fronti, di cui, 8 in salita ed 8 in discesa. Ogni fronte ha un simbolo che lo contraddistingue come rappresentato in figura.

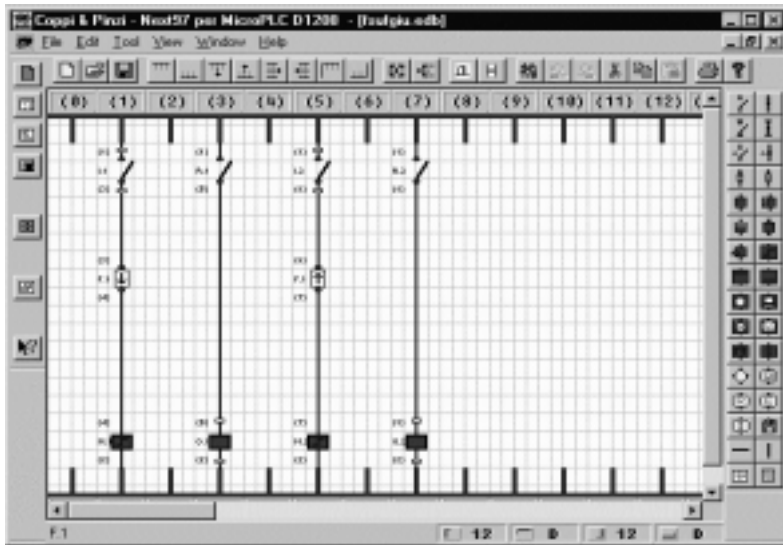


Il rettangolo con la freccia verso l'alto contraddistingue il *FRONTE DI SALITA* mentre quella verso il basso il *FRONTE DI DISCESA*.

Cliccando due volte sul componente già piazzato si può accedere al dialog di impostazione dati del componente. All'interno della finestra di dialogo va inserito il numero di componente (da 1 a 8) e l'eventuale commento. A differenza dei contatti, non è possibile ripetere N volte il piazzamento di un fronte; infatti, ogni volta che se ne piazza uno gli va assegnato un nome diverso. E' possibile nominare nello stesso modo un fronte di salita ed uno di discesa poiché in pratica questi debbono essere considerati due dispositivi diversi. Con lo schema applicativo FSUFGIU.EDB si può verificare il funzionamento dei fronti.

Nella prima colonna l'ingresso N°1 è posto in serie ad un fronte di discesa che va a comandare un relè *PASSO PASSO*. Nella seconda colonna il contatto del relè passo passo comanderà l'uscita N° 1.

In questa configurazione il relè riceverà il comando solo quando verrà rilasciato il pulsante di ingresso N° 1. Il relè rimarrà abilitato per una sola scansione di programma a causa della presenza del fronte di discesa, tempo comunque sufficiente a far sì che possa mantenere memorizzata la condizione. Nella seconda configurazione (schema colonne 5 e 7) viene utilizzato l'ingresso N°2 in serie ad un *FRONTE DI SALITA*.

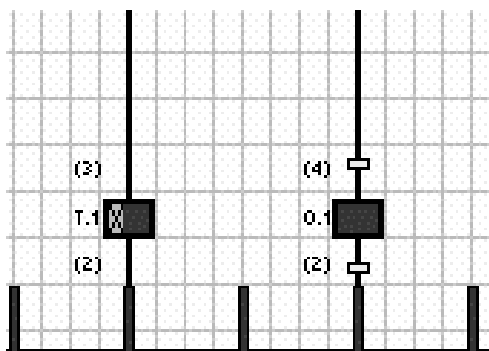


Come nella parte precedente questa catena alimenta un secondo relè *PASSO PASSO*. In questo caso il relè riceverà l'impulso di abilitazione per una sola scansione ma appena premuto il pulsante di ingresso N°2. Anche in questo caso, pur essendo l'impulso di breve durata, la condizione verrà memorizzata dal relè N°2. Il funzionamento del relè *PASSO PASSO* verrà esaminata successivamente in un esempio dedicato.

Si può aggiungere che in generale i fronti sono utili in tutte quelle applicazioni dove risulta necessario eliminare impulsi ripetuti che possano compromettere il normale funzionamento di un programma, e comunque, in tutte quelle applicazioni dove bisogna eseguire delle operazioni una sola volta e per una sola scansione di programma. Queste condizioni si presenteranno nel caso dell'utilizzo del blocco USER per eseguire operazioni aritmetiche, logiche (funzioni di shift register, rotazioni di byte, ecc), e manipolazioni delle WORD interne al PLC.

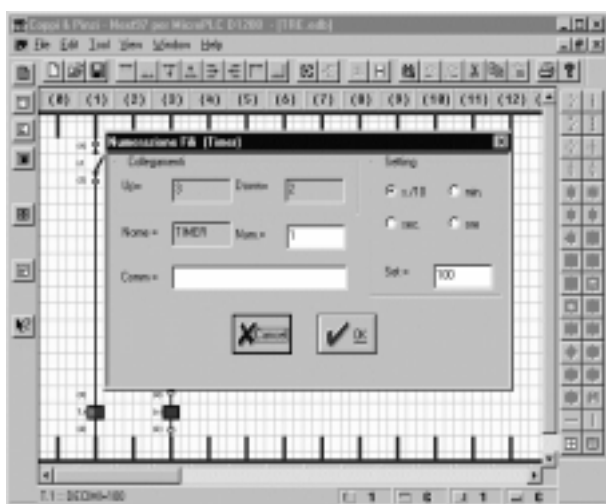
ESEMPIO 6 I Timer

Ogni PLC contiene al suo interno **8 temporizzatori**. I temporizzatori sono programmabili nel range compreso tra 1 e 32768, e con basi dei tempi di 100 mS, Secondi, minuti, ore. Come per i relè interni ed in



genere come per tutti i componenti del DI1208, ogni **TIMER** può disporre di un numero illimitato di contatti aperti e/o chiusi (sempre in rapporto alla capacità di memoria di programma). Il simbolo che contraddistingue il timer è quello rappresentato dal disegno a fianco e contrassegnato dal nome **T1**. All'interno del PLC i timer vengono emulati dal software, in tempo reale e rispecchiando il funzionamento dei timer elettromeccanici classici. Il numero di identificazione, ed il valore di SET con la relativa base dei tempi, possono essere impostati cliccando due volte all'interno del rettangolo che rappresenta il timer. Più propriamente il rettangolo raffigurato e denominato **T1** è solo la bobina del temporizzatore. Come per i **TIMER** tradizionali anche

questo componente è costituito nel suo insieme dalla bobina e dai contatti NA, NC e dai contatti istantanei. Come vedremo negli esempi che seguiranno, la combinazione di tutti questi elementi permette la realizzazione di dispositivi con funzionalità comparabili a tutti i tipi di **TIMER** esistenti in commercio. La finestra di dialogo per la impostazione dei valori di SET è raffigurata sotto; è ben evidente che cliccando



all'interno di uno dei 4 cerchietti è possibile selezionare la base dei tempi, mentre dentro alla finestra SET è possibile inserire il valore di preset del temporizzatore. Come per gli altri componenti visti in precedenza è possibile immettere il numero di identificazione (all'interno della finestra NUM) ed un eventuale commento allo scopo di facilitare la lettura del programma.

Per ottenere un ottimo grado di precisione del tempo richiesto è importante tenere conto di una regola essenziale:

Il valore di preset dovrà essere scelto in funzione della base dei tempi selezionata; infatti, non è corretto impostare il tempo di preset pari ad 1 secondo selezionando la base dei tempi a secondi, ma è più corretto scegliere una base dei tempi di

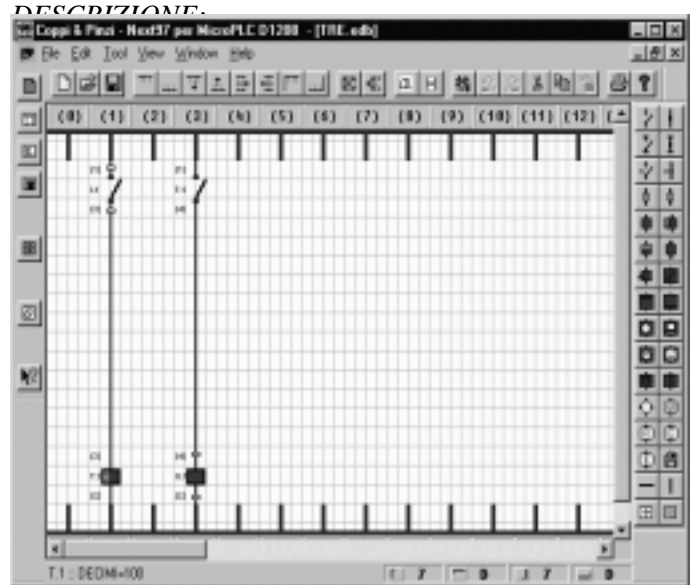
100ms ed inserire un valore di preset pari a 10. Pur non cambiando il risultato dal punto di vista funzionale avremo comunque una risoluzione migliore.

Per capire il funzionamento del **TIMER** dobbiamo tenere conto di alcune considerazioni:

- *Il temporizzatore fa partire il proprio conteggio quando viene abilitata la bobina. Questo è possibile solo quando tutta la logica a monte della bobina stessa ha come risultato il valore logico 1.*
- *Analogamente il **TIMER** cessa il proprio conteggio e si riazzera quando la sua bobina non è più abilitata, e cioè quando tutta la logica a monte ha come risultato il valore logico 0.*
- *I contatti normalmente aperti si chiudono quando il timer ha raggiunto, con il suo conteggio, il valore di PRESET, e rimangono in questo stato sino a quando la bobina viene disabilitata.*
- *I contatti normalmente chiusi si aprono a fine conteggio del timer, e rimangono in questo stato sino a quando la bobina non viene disabilitata.*
- *I contatti istantanei del timer si chiudono (NA) o si aprono (NC) istantaneamente nel momento in cui la bobina viene "virtualmente" alimentata. **E' importante ricordare che i contatti istantanei dei timer sono disponibili come contatti speciali all'interno del data base di questi ultimi e possono essere selezionati nell'ordine, dal contatto speciale N° 16 sino al contatto speciale N° 23 rispettivamente per i timer da 1 ad 8.***

ESEMPIO 7 Timer ritardato alla eccitazione

Realizzare lo schema di figura o caricarlo da NEXT97 scegliendo il file *TRE.EDB*.

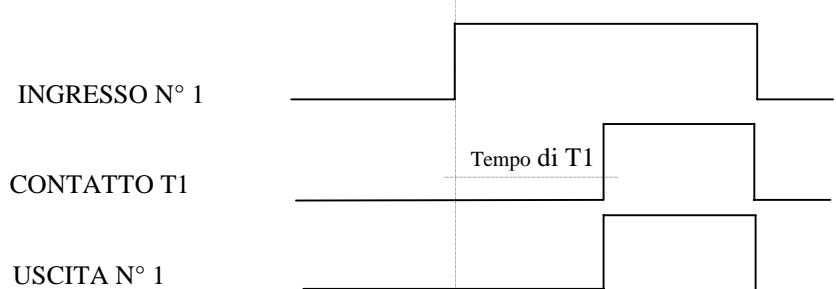


alto nella colonna 3 si chiuderà abilitando l'uscita N° 1. Il diagramma dei tempi rappresentato sotto renderà più semplice la comprensione del funzionamento di questo

applicativo. Per verificare quanto realizzato è sufficiente compilare il programma con le solite modalità, trasferirlo nella memoria programma del PLC utilizzando il SIM96 e avviare il sistema.

A trasferimento effettuato è possibile effettuare la verifica funzionale del programma. E' anche possibile leggere il valore del timer mentre sta lavorando. Per fare questo è necessario eseguire le seguenti operazioni:

Diagramma temporale del TIMER ritardato alla eccitazione



- Dal pannello di controllo del programma SIM96 spostare l'indicatore del mouse, sino a



ro **TIMER 1.8.**

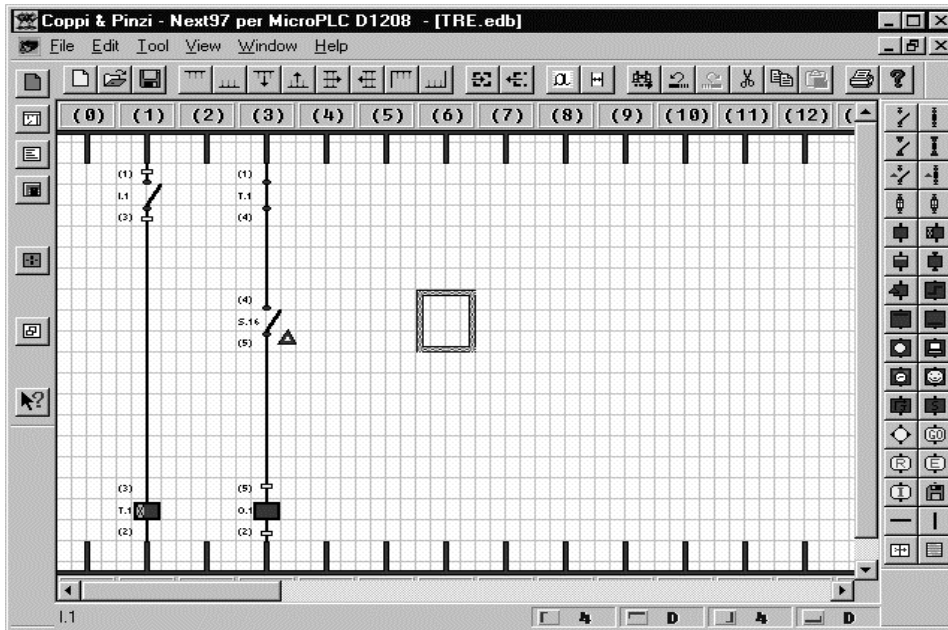
che il puntatore del mouse si trasforma e diventa una mano che indica il condizione è sufficiente cliccare con il pulsante sinistro per far aprire la del TIMER 1. Oltre a poter visualizzare lo stato corrente del TIMER è re di preset. In questo caso bisogna selezionare il menu *IMPOSTAZIONI* o la voce *MODIFICA TIMER* che si aprirà vi impone di inserire il re, e, sol



clickare sul tasto CAMBIA e di seguito sul tasto sottostante. Il nuovo valore rimarrà sino a quando il PLC non viene rimesso a STOP. In questo caso riprenderà dal programma il vecchio valore impostato.

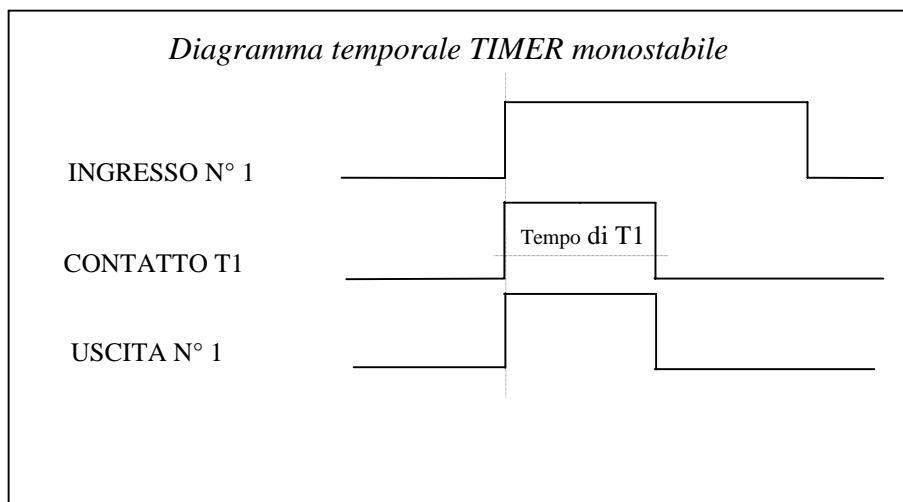
ESEMPIO 8 Timer monostabile

Realizzare lo schema di figura o caricarlo da NEXT97 scegliendo il file *TIMMON.EDB*.



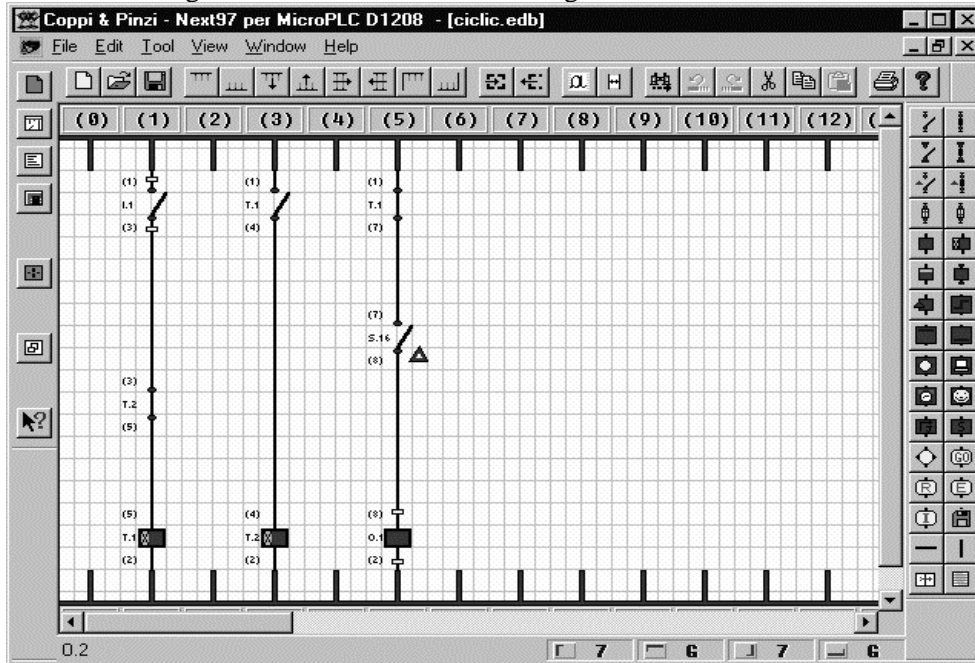
DESCRIZIONE:

Lo schema raffigura la connessione di un timer in configurazione monostabile. Alla chiusura dell'ingresso N°1 il temporizzatore inizierà il conteggio in funzione della sua base dei tempi e del valore impostato, contemporaneamente nella colonna 3 si chiuderà anche il contatto istantaneo di T1 (denominato S16). Il contatto NC temporizzato di T1 rimarrà chiuso sino a che il temporizzatore non raggiungerà il valore di preset. La serie dei due contatti chiusi abiliterà l'uscita O1 per tutto il tempo conteggiato da T1. Al raggiungimento del valore prefissato si avrà la riapertura dell' uscita. Il timer si riavverrà immediatamente (anche a tempo di lavoro non ultimato) se l'ingresso N° 1 viene riaperto. Il diagramma dei tempi chiarisce quanto fin qui esposto. Su questo diagramma non è rappresentato il reset prima del fine conteggio, perché questa rappresentazione è la copia esatta del diagramma dell'ingresso 1 in funzione del tempo che sta abilitato. Come per l'esempio precedente è possibile verificare il funzionamento di questo circuito e monitorare il valore di T1 che si incrementa. Per fare questo è sufficiente seguire le modalità già esposte nell'esempio 5.1 " *Timer ritardato alla eccitazione* " e riguardanti il programma SIM 96



ESEMPIO 9 Timer ciclico

Realizzare lo schema di figura o caricarlo da NEXT97 scegliendo il file *CICLIC.EDB*.

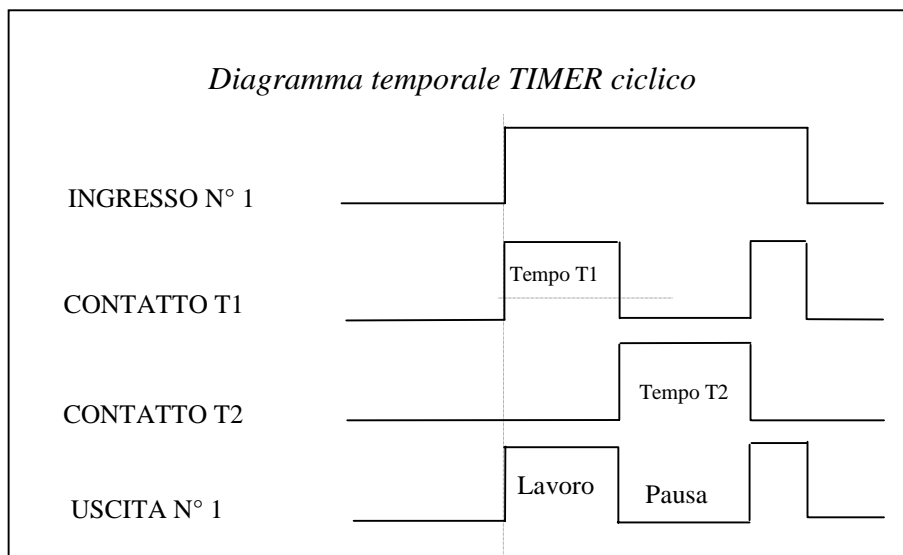


DESCRIZIONE:

Lo schema raffigura come realizzare la funzione di temporizzazione ciclica *LAVORO, PAUSA*.

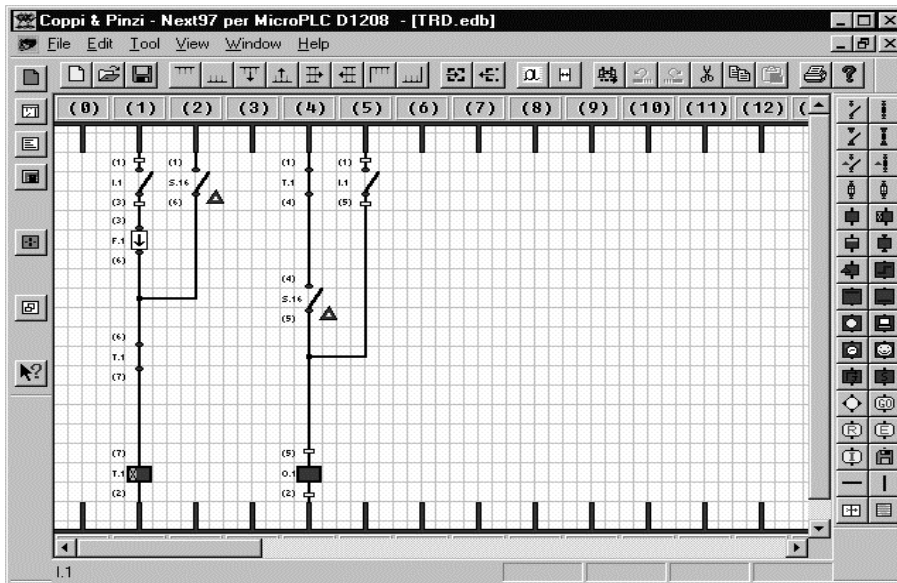
Per questo applicativo è necessario impiegare due temporizzatori. Il temporizzatore T1 viene utilizzato per il *LAVORO*, il temporizzatore T2 per il tempo di *PAUSA*. Alla chiusura dell'ingresso N°1 viene subito alimentato il TIMER 1, questo è possibile perché il contatto temporizzato del *TIMER 2* posto in serie all'ingresso N° 1 risulta chiuso essendo questo temporizzatore ancora disabilitato. L'uscita 1 è asservita dal *TIMER 1* in configurazione monostabile (vedi esempio 5.2), per cui viene immediatamente alimentata dalla serie **T1 N.C.** ed **S16 N.A.** Al raggiungimento del tempo impostato il *TIMER 2* viene comandato dal contatto **N.A. di T1**, (colonna 3). Questo evento dà inizio alla fase di *PAUSA*, che si protrarrà per tutto il tempo di **T2**. A fine tempo **T2** aprirà il proprio contatto posto sulla colonna 1 in serie all'ingresso 1. L'apertura di questo contatto resetterà il *TIMER T1*. Questa condizione riavvierà nuovamente i due cicli.

Tutta la funzione ciclica rimarrà attiva sino a quando non viene aperto l'ingresso 1. Aprire il programma SIM96 e caricare il programma *CICLIC.EPC* per verificare il funzionamento dell'applicazione appena esposta.



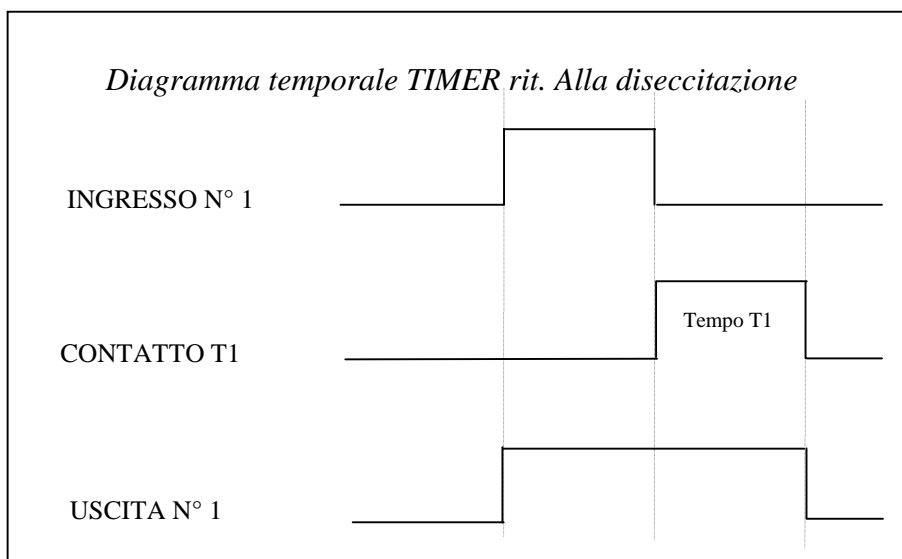
ESEMPIO 10 *Timer ritardato alla diseccitazione*

Realizzare lo schema di figura o caricarlo da NEXT97 scegliendo il file *TRD.EDB*.



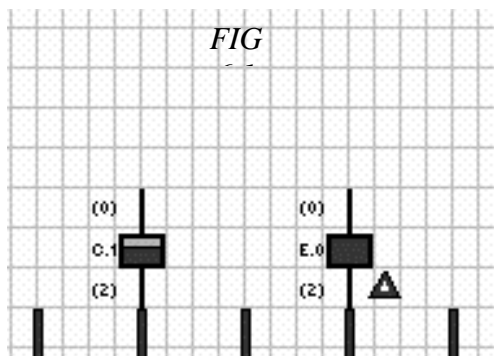
DESCRIZIONE:

Dal diagramma temporale si può risalire al funzionamento del timer ritardato alla diseccitazione. Come si può constatare l'uscita viene abilitata dall'ingresso 1, al suo rilascio l'uscita non si disabilita, ma rimane nello stato ON per tutto il tempo impostato sul timer N° 1. Inizialmente l'uscita N° 1 viene abilitata solo dall'ingresso 1 premuto (colonna 5), il timer risulta ancora disabilitato perché l'altro contatto dell'ingresso N° 1 (colonna 1) ha in serie un fronte di discesa (esempio 4). Al rilascio del contatto di ingresso il fronte abilita immediatamente il TIMER 1, il quale per mezzo del proprio contatto istantaneo **S16** si automantiene. In questa seconda fase l'uscita è mantenuta ad ON dalla serie **T1 N.C.** ed **S16** (colonna 4). Questo stato si protrarrà sino al raggiungimento del valore impostato su **T1**, sino a quando, cioè, il contatto NC di **T1** (colonna 1) si aprirà per azzerare tutta la sequenza. Tutta la procedura potrà essere riavviata solo richiudendo l'ingresso **I1**. Aprire il programma SIM96 e caricare il programma TRD.EPC per verificare il funzionamento dell'applicazione appena esposta.

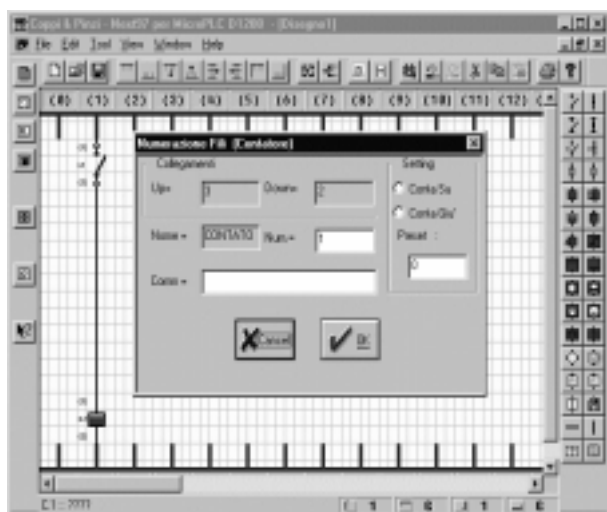


ESEMPIO 11 I Contatori

Ogni PLC contiene al suo interno **8 contatori**. I contatori sono programmabili nel range compreso tra 1 e 32768, ed in modalità UP e DOWN. Come per i relè interni ed in genere come per tutti i componenti del DI1208, ogni *CONTATORE* può disporre di un numero illimitato di contatti aperti e/o chiusi (sempre in rapporto alla capacità di memoria di programma). Il simbolo che contraddistingue il contatore è quello rappresentato nel disegno a fianco e contrassegnato dal nome **C1**. All'interno del PLC i contatori vengono emulati dal software, in tempo reale e rispecchiando il funzionamento dei contatori tradizionali. Il numero di identificazione, ed il valore di PRESET con la relativa modalità di conteggio, possono essere impostati cliccando due volte all'interno del rettangolo che rappresenta il timer. Più propriamente il rettangolo raffigurato e denominato **C1** è solo la bobina del modulo di conteggio. Come per i TIMER tradizionali



anche questo componente è costituito nel suo insieme dalla bobina e dai contatti NA, NC. Il contatore potrà essere resettato per mezzo di una bobina di reset (bobina speciale), la quale una volta abilitata riporta il conteggio al suo valore iniziale. La finestra di dialogo per la impostazione dei valori di SET è raffigurata sotto; è ben evidente che cliccando all'interno di uno dei 2 cerchietti è possibile selezionare la modalità di conteggio, mentre dentro alla finestra PRESET è possibile inserire il valore di preset del contatore.



Come per gli altri componenti visti in precedenza è possibile immettere il numero di identificazione (all'interno della finestra NUM) ed un eventuale commento allo scopo di facilitare la lettura del programma.

Il funzionamento dei contatori è molto semplice e dipende dalla modalità impostata.

Modalità CONTASU (UP)

- Il contatore inizia il suo conteggio da 0.
- Ogni volta che la bobina C1 viene abilitata il contatore si incrementa di 1. La funzione di antirimbasso è già supportata internamente dal

software e per il corretto conteggio degli impulsi non è necessario inserire nessun fronte di salita.

- Quando il conteggio ha raggiunto il valore di PRESET - 1 i contatti normalmente aperti si chiudono e quelli chiusi si aprono.
- Al raggiungimento del valore prefissato, se la bobina non viene interrotta, il conteggio continua sino al valore massimo (32768) per poi ritornare a 0, quindi, per non incorrere in problemi software, è molto importante seguire questa regola generale:

IN SERIE ALLA BOBINA DI CONTEGGIO BISOGNA SEMPRE INSERIRE UN CONTATTO NORMALMENTE CHIUSO DEL CONTATORE STESSO, IN MODO CHE A CONTEGGIO ULTIMATO QUESTO SI APRA ED IMPEDISCA UN ULTERIORE INCREMENTO DEL CONTATORE.

- I contatti aperti e/o chiusi ritornano nel suo stato iniziale solo dopo che il contatore è stato resettato per mezzo della bobina speciale a lui dedicata. Nella FIG 6.1 si può vedere il simbolo della *bobina speciale* che, come per i contatti speciali ha un data base interno (accessibile con doppio click da pulsante sinistro) all'interno del quale sono situate le 8 bobine di reset (da 0 a 7) dei rispettivi contatori.

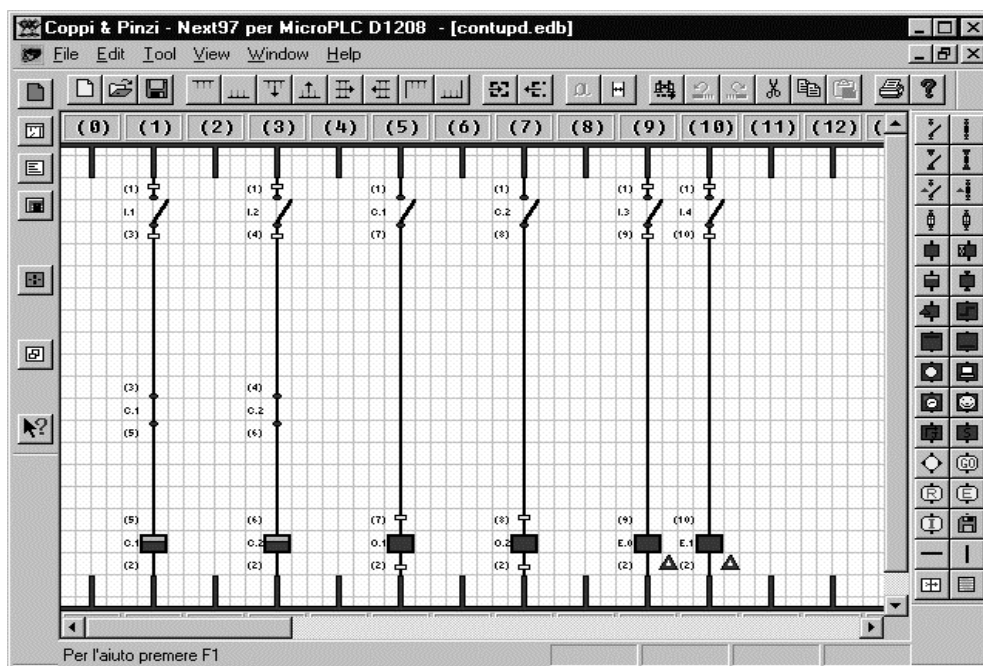
Modalità CONTAGIU (DOWN)

- Il contatore inizia il suo conteggio dal valore di preset

- Ogni volta che la bobina di C1 viene abilitata il contatore si decrementa di 1. La funzione di antirimbando è già supportata internamente dal software e per il corretto conteggio degli impulsi non è necessario inserire nessun fronte di salita.
- Quando il conteggio ha raggiunto il valore di 0 i contatti normalmente aperti si chiudono e quelli chiusi si aprono.
- Al raggiungimento del valore prefissato, se la bobina non viene interrotta, il conteggio continua passando per il massimo valore (32768) per poi ritornare a 0, quindi, per non incorrere in problemi software, è molto importante seguire questa regola generale:
IN SERIE ALLA BOBINA DI CONTEGGIO BISOGNA SEMPRE INSERIRE UN CONTATTO NORMALMENTE CHIUSO DEL CONTATORE STESSO, IN MODO CHE A CONTEGGIO ULTIMATO QUESTO SI APRA ED IMPEDISCA UN ULTERIORE INCREMENTO DEL CONTATORE .
- I contatti aperti e/o chiusi ritornano nel suo stato iniziale solo dopo che il contatore è stato resettato per mezzo della bobina speciale a lui dedicata. Nella FIG 6.1 si può vedere il simbolo della *bobina speciale* che, come per i contatti speciali ha un data base interno (accessibile con doppio click da pulsante sinistro) all'interno del quale sono situate le 8 bobine di reset (da 0 a 7) dei rispettivi contatori.

ESEMPIO 12 Contatori UP e DOWN

Realizzare lo schema di figura o caricarlo da NEXT97 scegliendo il file CONTUPD.EDB.



DESCRIZIONE:

Il contatore N°1, che è programmato in modo UP con preset = 10, viene incrementato dall'ingresso N° 1. Chiudendo per 9 volte l'ingresso N°1 (PRESET-1), il contatto aperto da C1 posto nella colonna 5 si chiuderà abilitando l'uscita N° 1. Chiudendo l'ingresso N°3 è possibile effettuare il reset del contatore N° 1 per mezzo della rispettiva bobina di reset.

Il contatore N°2, che è programmato in modo DOWN con preset = 5, viene decrementato dall'ingresso N° 2. Chiudendo per 4 volte l'ingresso N°2 (PRESET-1), il contatto aperto da C2 posto nella colonna 7 si chiuderà abilitando l'uscita N° 2. Chiudendo l'ingresso N°4 è possibile effettuare il reset del contatore N° 2 per mezzo della rispettiva bobina di reset.

In entrambe le colonne di abilitazione conteggio dei due contatori (colonne 1 e 3) è possibile notare il contatto chiuso dei rispettivi contatori. Come è evidente questi contatti si aprono a fine conteggio per impedire il proseguimento del conteggio stesso. Per verificare quanto realizzato è sufficiente compilare il programma per mezzo dell'apposito comando da programma NEXT97 e trasferirlo nella memoria programma del P.L.C. utilizzando il SIMULATORE SIM96.

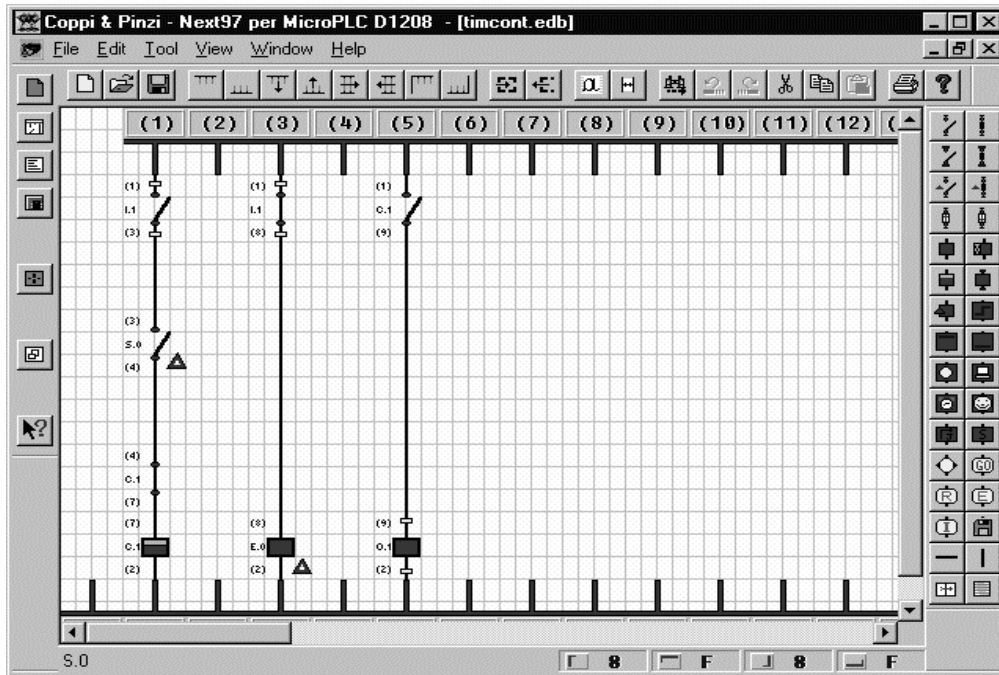
Una volta trasferito su P.L.C. è possibile sperimentare il funzionamento del programma descritto. Come per i timer è anche possibile vedere realmente il conteggio.

Per verificare il conteggio del contatore C1 posizionarsi con l'indicatore sul primo led a sinistra del riquadro **CONTATORI 1..8**. Dopo aver fatto questo si noterà che il puntatore del mouse si trasforma in una mano che indica il led interessato. In questa condizione è sufficiente cliccare con il pulsante sinistro per far aprire la finestra che visualizza lo stato del contatore N°1. Nella figura accanto è rappresentata la finestra di dialogo che indica lo stato del contatore N°1. Selezionando il menu IMPOSTAZIONI e di seguito la voce MOD. COUNTER è anche possibile impostare un nuovo valore di preset o modificare il valore attuale del conteggio in corso.



ESEMPIO 13 Contatore in modo timer

Utilizzando la funzione di conteggio associata con i bit speciali a tempo è possibile realizzare dei timer. Questa possibilità può risultare utile quando stiamo realizzando dei programmi che impiegano più degli 8 timer disponibili nel PLC DI1208. Per realizzare questo applicativo disegnare lo schema raffigurato o caricarlo da NEXT97 scegliendo il file *TIMCONT.EDB*.



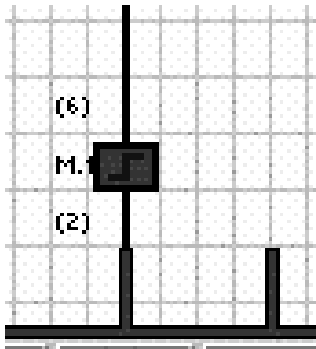
DESCRIZIONE:

In questa particolare applicazione il contatore assolve alla funzione di timer ritardato alla eccitazione. Il contatore N°1 è stato preettato al valore 51 ed impostato in modo UP. L'ingresso N°1 da l'avvio al conteggio ma è il contatto speciale **S0** aprendosi e chiudendosi ogni decimo di secondo, a stabilire la base dei tempi. Questo è possibile poichè il contatore che riceve un impulso ogni 2/10 di secondo si incrementa con questa velocità. Una volta che il contatore avrà raggiunto il valore di PRESET-1 abiliterà l'uscita 1 per mezzo del proprio contatto NA. Il reset del contatore avverrà, per rendere la funzione compatibile con il timer, aprendo l'ingresso 1. Si può notare, infatti, che togliendo la abilitazione a questo ingresso verrà abilitata la bobina speciale E0 la quale forzerà a zero il proprio contatore.

ESEMPIO 14 Relè passo passo

I relè passo passo, o relè salterello, cambiano il loro stato ad ogni impulso ricevuto. Avevamo già realizzato una applicazione con questo tipo di relè nell'esempio 4 " *I fronti di salita e di discesa* ", ma vediamo più in dettaglio come funziona. Il simbolo che contraddistingue questo tipo di relè è quello rappresentato nella figura sottoriportata. Nel programma NEXT il relè passo passo non è considerato come un componente, perché è sempre da associare ad uno dei 64 relè interni.

Provate, dal programma NEXT, a piazzare il componente con il simbolo rappresentato in figura. Una volta piazzato cliccateci sopra due volte per aprire la finestra delle proprietà come già è stato fatto per gli altri componenti.



Comparirà il dialog box raffigurato sulla destra. In corrispondenza della finestra **RELE USCITA** è possibile inserire un numero (da 1 a 64) che identifica il numero di relè interno al quale il relè passo passo sarà associato. Naturalmente una volta assegnato un numero il relè corrispondente non potrà più venire utilizzato come bobina per altri tipi di funzioni, e tutti i contatti aperti e chiusi che si riferiscono al numero assegnato si

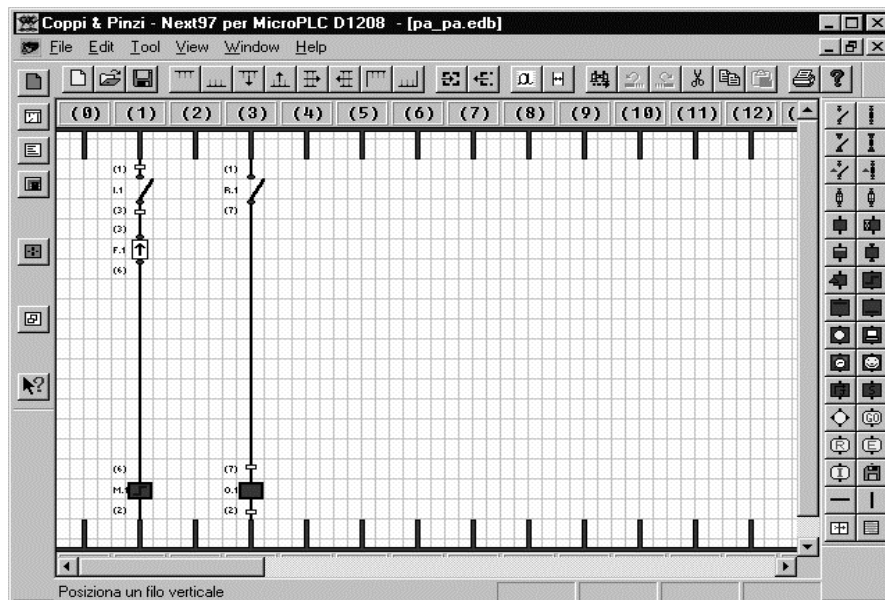


intendono associati al relè passo passo scelto ed indicato dal **NUM** nella finestra di figura. Per evitare errori di assegnazione, e per rendere lo schema più leggibile si consiglia di assegnare al relè anche un commento digitandolo nella apposita finestra.

IMPORTANTE:

A causa dei rimbalzi di tutti i componenti elettromeccanici, si consiglia di inserire un fronte di salita in serie ad ogni contatto di ingresso che abilita un relè passo passo.

Per facilitare la comprensione dei relè passo passo si può fare riferimento all'esempio sottoindicato.

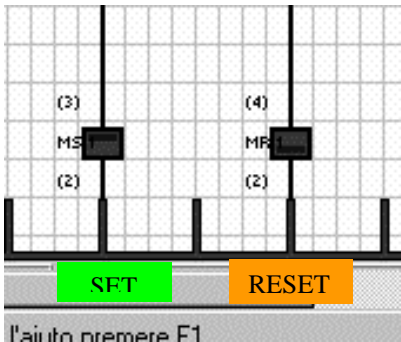


Disegnare lo schema o caricare dal dischetto esempi il file PA_PA.EDB. Compilare il file e trasferirlo sul PLC per mezzo del programma SIMULA. L'esempio è di facile comprensione e non necessita di alcun commento. E' importante ricordare le regole già menzionate in precedenza:

- Non assegnare ad un relè interno od ad un altro componente lo stesso relè interno associato al passo passo che lo utilizza. *Nel nostro esempio sarebbe un errore disegnare una bobina di un relè interno chiamata R1.*

ESEMPIO 15 Relè bistabile

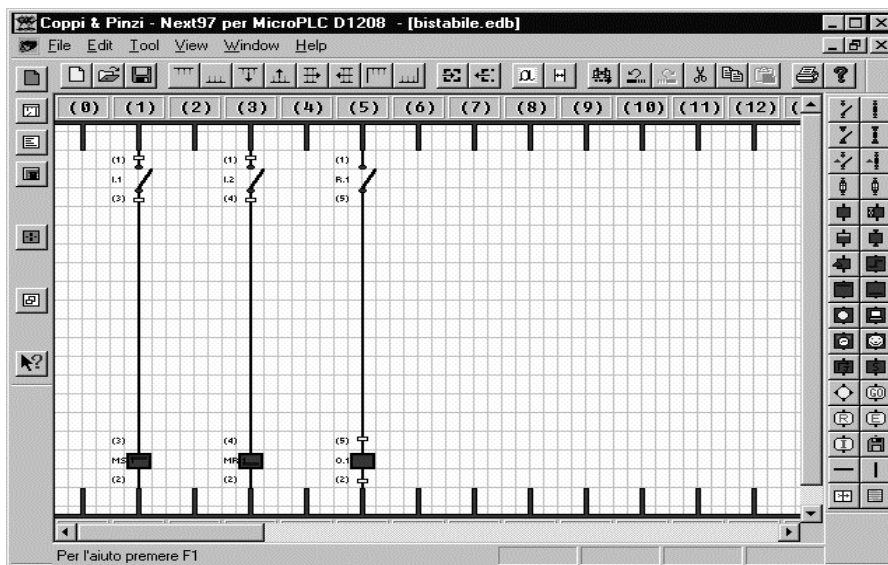
Il relè bistabile è costituito da 2 bobine, una di SET l'altra di RESET. Anche questo componente come il relè passo passo si deve appoggiare ad un relè interno per poter lavorare correttamente. Il funzionamento del relè bistabile è identico a quello di un FLIP-FLOP SET,RESET. Abilitando la bobina di SET il contatto NA ad essa associato si chiude, e si mantiene in questo stato sino a quando non viene abilitata la bobina di RESET. In pratica questo tipo di relè ha 2 bobine ed un contatto ad esse comune. Naturalmente come per gli altri componenti è possibile associare a questo un numero di contatti aperti e/o chiusi dipendente esclusivamente dalla quantità di memoria a disposizione. I simboli che rappresentano le due bobine (indicati in figura) sono



evidenziati da una linea orizzontale di colore blu che, nel caso della bobina di SET, è disegnata in alto, mentre in quella di reset in basso.

Per assegnare lo stesso relè interno ad ambedue le bobine è per forza necessario scriverlo su ogni bobina in modo separato ripetendo due volte la procedura di assegnazione. Per fare questo le modalità sono quelle comuni a tutti i componenti, e cioè, fare un doppio click sulla bobina interessata e compilare la finestra di dialogo che comparirà. Nel caso dei relè bistabili il dialog box è lo stesso dei relè passo passo e le regole da seguire sono identiche. Considerando che in questo caso 2 bobine distinte devono fare riferimento allo stesso relè, è ancora più importante riempire lo spazio dedicato al commento, così da poter meglio identificarne la funzione.

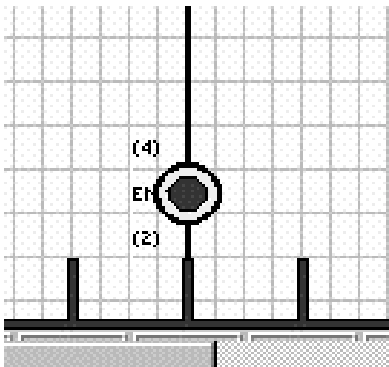
Per meglio comprendere il funzionamento di questo componente è possibile aprire il file BISTABILE.EDB



da programma NEXT97. Comparirà lo schema sottoriportato che è già pronto per essere compilato e quindi trasferito sul PLC. Mettendo in RUN il controllore ed abilitando l'ingresso N° 1 si avrà l'accensione della uscita N°1, questo perché il relè N°1, che la comanda, è stato selezionato all'interno delle 2 bobine del bistabile. Per poter spegnere l'uscita basta abilitare l'ingresso N° 2 che in questo caso comanda la bobina di RESET del relè bistabile.

ESEMPIO 16 Blocco Enable

Quello raffigurato sotto è il componente ENABLE. La funzione di questo blocco è quella di forzare lo stato di tutte le uscite a zero quando non è abilitato. Questo blocco può essere utilizzato quando vogliamo essere certi che, a prescindere dallo stato di ciascuna uscita, un unico comando ne forzi la condizione allo 0 logico.



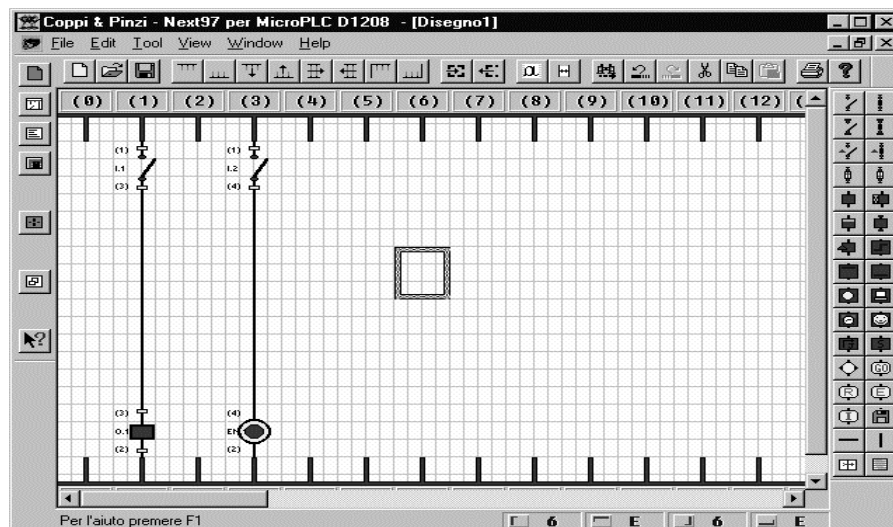
Generalmente nel programma simula è contenuta una serie di istruzioni che provvedono alla inizializzazione di alcuni parametri nel PLC. Le istruzioni, che sono quattro, possono essere manipolate selezionando il menu EDIT e successivamente il sottomenu INIZIALIZZAZIONE. Una di queste istruzioni è "ENABLE". Per default i parametri sul PLC sono inizializzati come nella finestra qui sotto riportata, e cioè:

- **TIPO DI PLC** = SLAVE (master solo quando si devono collegare in rete più PLC o espansioni ad un unico PLC).
- **NUMERO DI PLC** = 1 stabilisce di quali PLC si tratta.
- **VEL COM. PLC** = 57600 Velocità seriale.
- **OUTPUT** = Abilitati (ENABLE) o disabilitati.

E' importante ricordare che la modifica di questi parametri non viene salvata nel disegno, per cui ad ogni nuova riapertura dello stesso questi vanno nuovamente settati nella maniera voluta.

Il blocco ENABLE è l'unico parametro di inizializzazione che può essere cambiato anche da schema elettrico, e questo è possibile per mezzo della bobina descritta in precedenza.

Per comprendere meglio questa funzione abbiamo realizzato un piccolo esempio che abbiamo chiamato ENABLE.EDB e che è contenuto nel disco ESEMPI. Lo schema di questo esempio è quello qui sottoriportato.



DESCRIZIONE:

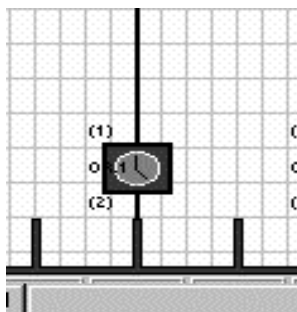
Dopo avere trasferito il programma sul PLC sarà sufficiente metterlo in RUN per rendersi subito conto di una differenza rispetto agli esempi sino a qui esaminati, e cioè, il lampeggio del led verde ENABLE.

Questo significa che le uscite sono disabilitate, e questo a causa dell'ingresso N° 2 che è inizialmente aperto. Chiudendo anche l'ingresso N°1 che abilita l'uscita corrispondente il risultato non cambia poiché la bobina ENABLE è ancora disabilitata.

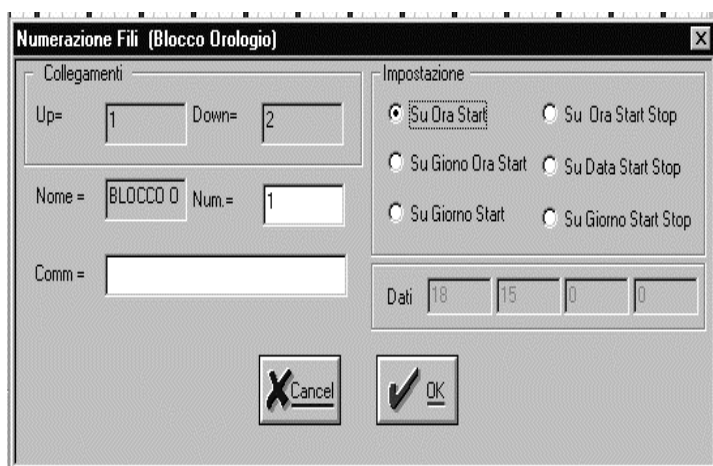
Per uscire da questa condizione è necessario chiudere l'ingresso 2 che agisce direttamente sulla bobina di abilitazione (il cambiamento di stato è anche evidenziato dallo spegnimento del led verde ENABLE).

ESEMPIO 17 Orologio interno

Il PLC DI1208 è dotato di un orologio interno che mantiene memorizzati data ed ora in tempo reale anche in assenza di alimentazione. Utilizzando le funzioni dell'orologio interno sono stati implementati quattro blocchi bobina orologio che offrono notevoli possibilità di programmazione. I quattro blocchi sono perfettamente identici ed ognuno possiede un numero illimitato di contatti NA ed NC. Ognuno di questi orologi è impostabile in diverse modalità come è possibile vedere dal disegno raffigurante la finestra di dialogo che, naturalmente, è accessibile cliccando due volte con il pulsante sinistro del mouse sulla relativa bobina. Dal dialog box, all'interno del riquadro impostazione, sono visibili i modi di configurazione del modulo orologio.



- **Su ora start** impostando l'ora ed i minuti all'interno del riquadro DATI stabiliremo il momento in cui i contatti NA dell'orologio definito da "NUM" (1,2,3 o 4) si chiudono, e viceversa, quelli NC si aprono. In questa modalità i contatti utilizzati restano attivi per un solo minuto, questo perché i dati dell'orologio sono aggiornati con cadenza al minuto.
- **Su ora start stop** in questa modalità va definito il momento di start inserendo l'ora ed i minuti nelle prime due caselle sulla sinistra (all'interno della finestra dati), e l'ora ed i minuti di stop nelle caselle rimanenti. In questo modo il funzionamento è analogo a quello di un orologio giornaliero a cavalieri.
- **Su Giorno Ora Start** il funzionamento è quello di un orologio annuale. Vanno definiti: ore, minuti, giorno del mese e mese dell'anno. Anche in questo caso i contatti restano attivi per il solo minuto corrente nel quale avviene lo start.
- **Su Data Start Stop** in questo modo l'orologio rimarrà attivo per l'intervallo stabilito dai giorni e mesi di start e di stop definiti nelle finestre DATI. In questo caso la base dei tempi è scandita dal passaggio per la mezzanotte del giorno impostato.
- **Su Giorno Start** In questo caso ci riferisce al giorno settimanale che è un numero compreso tra 1 e 7. Anche in questo caso lo start avviene al passaggio delle ore 24.
- **Su Giorno Start Stop** come la selezione precedente ma i bit interni dell'orologio selezionato vengono abilitati nell'intervallo dei giorni settimanali stabiliti.

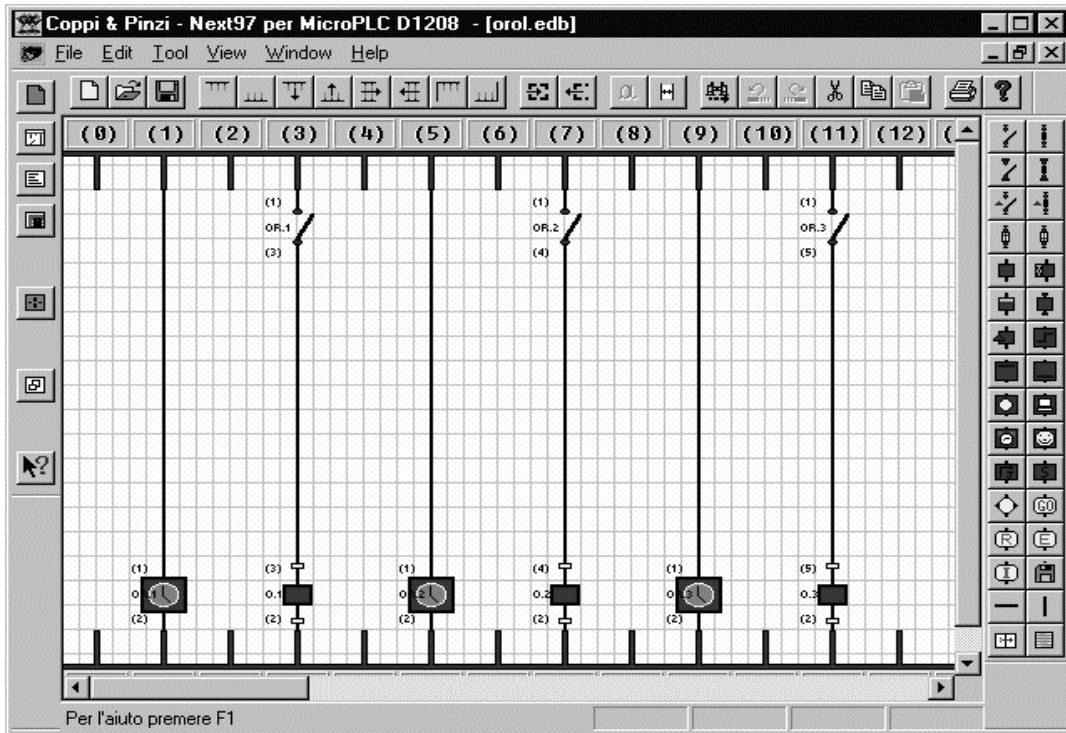


Come per tutti gli altri blocchi funzione è possibile inserire un commento per identificare qual è la funzione svolta dall'orologio inserito; inoltre è necessario inserire il numero che identifica l'orologio e che per questo tipo di PLC va da 1 a 4. Nelle pagine seguenti vedremo un esempio applicativo che tratta i moduli orologio.

ESEMPIO 18 Impiego dei blocchi orologio

In questo esempio vengono utilizzati tre moduli “orologio” impostati in modi diversi l’uno dall’altro per poter meglio comprenderne il funzionamento.

Aprire dal programma *NEXT97* il file “*OROLEDB*”, comparirà il disegno illustrato di seguito.

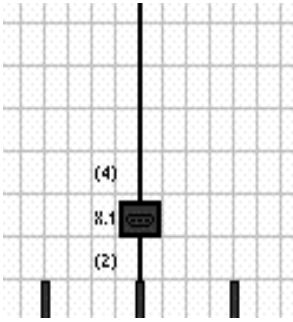


DESCRIZIONE :

Lo schema è molto semplice; l’orologio N°1 il cui contatto NA abilita la uscita N°1 (colonne 1 e 3) è impostato in modalità *ORA START* per le ore 18:15. Raggiunta l’ora prestabilita abiliterà l’uscita N°1 per un solo minuto. Naturalmente nel caso che occorra prolungare il tempo di abilitazione si può ricorrere alla funzione *ORA START STOP*. Questo è il modo con cui è stato programmato il blocco orologio N°2. Operando un doppio click su questa bobina è possibile verificare o reimpostare i tempi di lavoro. L’uscita interessata dall’orologio N°2 è l’uscita N°2, anche questa uscita raggiunto il tempo prefissato (sempre all’interno del minuto corrente) si abiliterà accendendo il led ed il relè interessati.

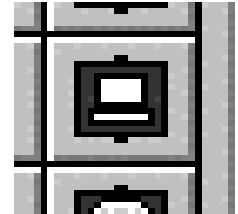
Infine l’ultimo orologio utilizzato in questo esempio, il numero 3, abilita la uscita N°3 ed è impostato per rendere attivo il proprio contatto per le 18:15 del giorno 15 Maggio. Anche in questo caso il contatto NA si chiuderà raggiunto il tempo prefissato e sempre all’interno del minuto corrente. Dopo avere compilato il file appena esposto è possibile trasferirlo dal personal al PLC utilizzando il solito programma “*SIMULA*”. Qui la fase di verifica implica per forza di adeguare l’ora e la data correnti del PLC ai tempi impostati, questo per evitare lunghe attese in fase di debug del programma nel caso i tempi risultino molto lontani. Sempre da *SIMULA* è possibile verificare lo stato e la eventuale impostazione dei dati semplicemente cliccando su uno degli 8 LED (per il DI1208 solo i primi 4) all’interno della finestra *OROLOGI 1..8* nel pannello sinottico “virtuale” del programma *SIMULA*.

ESEMPIO 19 Blocco trasmissione seriale



Il simbolo qui rappresentato sulla sinistra raffigura il blocco che gestisce la trasmissione di dati attraverso la porta seriale. Nel gruppo strumenti il blocco di trasmissione seriale è rappresentato da un simbolo raffigurante un terminale video (vedi figura a destra).

La funzione di questo dispositivo è quella di inviare dati attraverso la porta seriale del PLC. La velocità di comunicazione è impostabile dal menu di inizializzazione come già spiegato nell'esempio 9 a proposito del "Blocco Enable". Le modalità ed il tipo di trasmissione sono invece impostabili direttamente dal



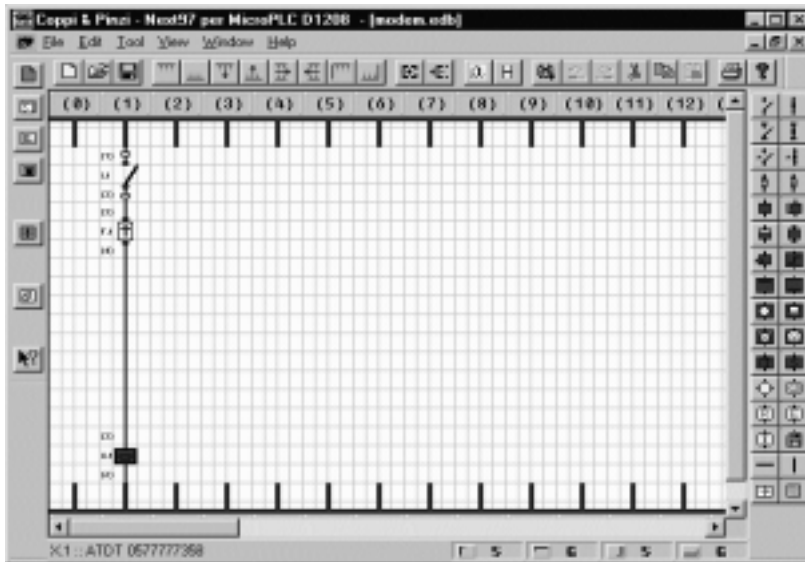
blocco trasmissione cliccando due volte con il pulsante SX del mouse sopra alla bobina.

Per il corretto utilizzo di questo componente bisognerà tener conto di un particolare importante:

Onde poter mantenere anche la comunicazione con il personal computer durante la fase di messa a punto di un programma contenente uno o più blocchi di trasmissione seriale, è necessario che le loro bobine non rimangano abilitate per lunghi periodi. Per rendere possibile questo inserire in serie ad ogni bobina un fronte di salita per fare sì che la trasmissione avvenga per una sola scansione di programma. **Mantenere sempre abilitato un blocco di trasmissione significa non poter permettere nessun collegamento tra personal e PLC. All'interno del blocco trasmissione, attraverso la finestra di dialogo, è possibile impostare solo la modalità di Setting:**

- *TX TESTO* : permette la trasmissione di una stringa ascii. Questa possibilità torna molto utile quando si devono inviare stringhe per il settaggio di modem o per la stampa di eventi su stampanti seriali o terminali seriali in genere. La composizione della stringa è possibile attraverso la composizione di un testo dentro la apposita finestra.

L'esempio riportato qui sotto indica come inviare una chiamata telefonica attraverso un modem.



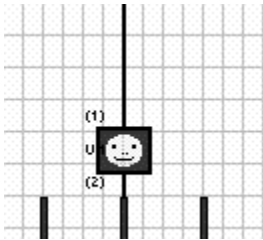
DESCRIZIONE:

L'ingresso N°1 ed il rispettivo fronte di salita abilitano il blocco testo N°1 all'interno del quale è contenuta la stringa di configurazione per il modem. Come spiegato in precedenza l'inserimento del fronte di salita serve ad inviare la stringa ogni volta che viene premuto il pulsante di ingresso N°1 ma solo per una scansione di programma. **Questo programma può essere aperto dal dischetto ESEMPI con il nome "MODEM.EDB". Sempre dentro al disco ESEMPI è possibile trovare un versione ancora più evoluta di questo file che è stata denominata**

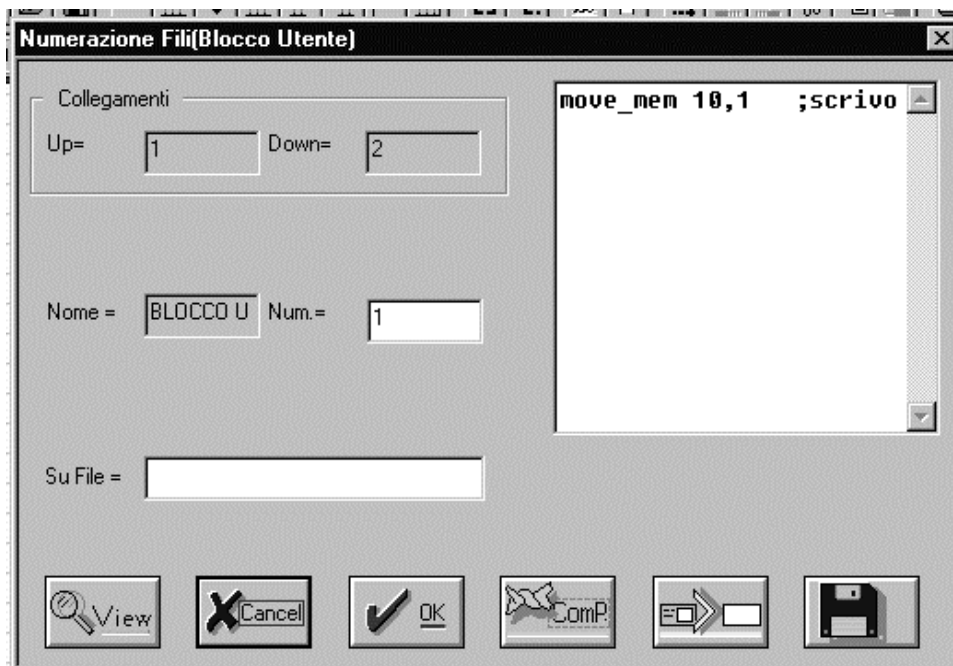
"MODEM1.EDB". Questa seconda versione utilizza più blocchi testo per trasmettere la configurazione completa di un modem. La comprensione di questo ultimo esempio implica la conoscenza di istruzioni più evolute di quelle sin qui trattate.

ESEMPIO 19 Blocco User

Questa bobina, che all'apparenza è un semplice blocchetto come quelli visti in precedenza, in realtà è un vero e proprio compilatore per linguaggio evoluto "LOGASM". All'interno di questo blocco è possibile inserire tutte le istruzioni supportate dal linguaggio pseudoassembler per DI1208, e di conseguenza si aprono nuove possibilità per la risoluzione di altri tipi di problemi. Con il linguaggio "LOGASM" è possibile implementare operazioni matematiche e logiche a 16 bit (WORD), memorizzare valori sui 40 registri disponibili, fare operazioni di rotazione, shift, conversioni di formato e quant'altro il linguaggio "LOGASM" rende possibile. Tutte le istruzioni supportate da questo linguaggio sono elencate nel manuale d'uso e nel manuale fornito sul disco di installazione "MANUALE.HLP". Oltre alle modalità di configurazione di questo blocco vedremo anche alcuni esempi che serviranno a chiarire il modo di impiego della bobina in questione. Il simbolo che contraddistingue il blocco user è raffigurato nel disegno di sinistra. In un programma non ci sono limiti al numero di blocchi utilizzati se non quelli imposti dalla capacità di memoria.



Cliccando sul blocco impostato si aprirà la finestra di dialog che permette l'inserimento della parte di programma. Esaminando il disegno che rappresenta il dialog box vedremo più in dettaglio ogni singolo comando di impostazione.



- **VIEW** importa un file in formato EPC. All'interno della finestra "Sufile" va digitato il nome del file da aprire nel formato sopra specificato.
- **COMP** compila la parte di programma editata all'interno della apposita finestra. Nella figura sopra riportata è stata inserita la riga:
MOVE_MEM 10,1
- > questo simbolo (nel bottom a destra di **COMP**) serve ad allargare a tutto schermo la finestra per l'inserimento delle istruzioni. Per ritornare alla dimensione originale è sufficiente ripremere nuovamente lo stesso tasto.
- **Simbolo "dischetto"** salva su un file specificato la parte di programma editata. Questa funzione è utile quando si vogliono conservare una serie di istruzioni che possono essere adattate a più programmi (questa procedura in gergo viene definita "MACRO").

Prima di esporre alcuni esempi applicativi utilizzanti il blocco **USER**, è opportuno chiarire alcuni aspetti che riguardano l'architettura interna del PLC DI1208. Una volta esposta la sua costituzione interna, le istruzioni editate all'interno degli blocchi **USER** dovrebbero risultare più comprensibili,

consentendo di risolvere problemi di notevole complessità e difficilmente affrontabili con il software *NEXT97*.

20 Architettura interna DI1208

ACCUMULATORE (16 bit)
REGISTRO AUX (16 bit)
INGRESSI da 1...8 (8 bit)
INGRESSI da 9 ...12 (8 bit)
USCITE da 1...8 (8 bit)
MEMORIA 0 (16 bit)
MEMORIA 1 (16 bit)
....
....
MEMORIA 39 (16 bit)
REGISTRO DI STACK 1 (16 bit)
REGISTRO DI STACK ... (16 bit)
REGISTRO DI STACK 16 (16 bit)

La tabella raffigurata rappresenta la architettura del PLC DI1208. Parte della memoria interna al PLC è stata dedicata alla emulazione di un microprocessore. L'utente può contare su una quantità di registri interni specializzati e/o di uso generale per eseguire una serie di operazioni.

Tutti i registri sono a 16 bit (WORD) tranne quelli contenenti gli stati delle uscite e degli ingressi.

Il cuore del sistema è senza dubbio il registro definito **ACCUMULATORE**, su questo registro si appoggiano quasi tutte le operazioni del PLC. Questo registro è dimensionato per contenere numeri a 16bit (65535 decimale, oppure 0xFFFF esadecimale). Le operazioni che si appoggiano all'accumulatore sono di tre dimensioni: 16 bit (WORD), 8

bit (BYTE), 1 bit.

Tutte le combinazioni logiche disegnate con il programma NEXT97 sono operazioni tra bit (ad un solo bit) e transitano tutte per il registro accumulatore. Dal momento che queste combinazioni logiche possono essere create facilmente da NEXT97 non verranno prese in considerazione negli esempi dedicati al blocco *USER*

Per la memorizzazione di WORD variabili e costanti l'utente ha a disposizione 40 registri (da 0 a 39) ai quali fanno riferimento tutte le istruzioni che hanno come estensione la parola "MEM", per esempio:

- **READ_MEM Nmem** che trasferisce il contenuto della memoria numero Nmem nel registro accumul.
- **WRITE_MEM Nmem** che trasferisce il contenuto dell'accumulatore nella memoria numero Nmem
- **MOVE_MEM Nmem,K** che scrive la costante K dentro la memoria Nmem.

E' importante ricordare che le 40 memorie utente non sono azzerate dal sistema operativo, (questo perché possano essere utilizzate anche come WORD costanti), quindi un eventuale reset o la forzatura a valori predefiniti di questi registri dovrà essere a discrezione del programmatore.

Le locazioni di memoria dove sono contenuti gli stati degli ingressi e delle uscite sono dimensionate ad 8 bit. Per talune applicazioni può essere indispensabile poter acquisire contemporaneamente un certo numero di ingressi o viceversa poter trasferire il contenuto di una locazione di memoria verso le 8 uscite. Tutto ciò è possibile utilizzando istruzioni dedicate a questo scopo come per esempio:

- **READ_INPUT Ngruppo** che trasferisce il contenuto del gruppo ingressi Ngruppo dentro l'accumulatore
- **WRITE_OUTPUT** che scrive il contenuto dell'accumulatore nelle 8 uscite disponibili.

Oltre alle 40 locazioni di memoria ne esistono altre 16 (lo Stack) alle quali è possibile accedere per mezzo di due istruzioni speciali :

- **PUSH** scrivi nello stack ed incrementane il contatore.
- **POP** leggi dallo stack e decrementane il contatore.

Utilizzando lo stack come serbatoio di dati non è poi possibile stabilire l'ordine di accesso a questi valori memorizzati. L'ordine viene deciso dal sistema operativo secondo una gerarchia del tipo : *IL PRIMO DATO AD ESSERE SCRITTO SARA' L'ULTIMO AD ESSERE RILETTO.*

Infine il registro ausiliario è un registro di supporto all'accumulatore per l'appoggio temporaneo di variabili.

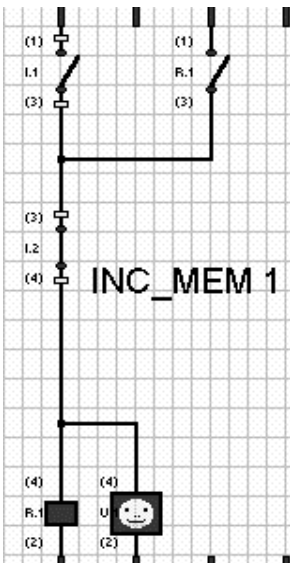
Anche questo registro come tutti gli altri è dimensionato per svolgere operazioni a 16 bit.

Come regola generale c'è da tener conto che in tutte le operazioni aritmetiche e nei confronti tra variabili il sistema operativo non tiene conto del segno, ma svolge tutti i calcoli per numeri interi positivi a 16 bit.

All'interno dei blocchi *USER* i dati numerici sono supportati in vari formati, decimale, binario, esadecimale, oppure sottoforma di espressioni numeriche. I suffissi che contraddistinguono i formati supportati sono le seguenti:

- *Nessuna suffisso* formato decimale esempio 10230
- *Suffisso 0x* formato esadecimale esempio 0xf0cd
- *Suffisso 0b* formato binario esempio 0b111000

I blocchi user sono delle comode vie per aggirare problemi complessi e difficilmente risolvibili con la sola programmazione a schema elettrico, senza stravolgere comunque il tipo di programmazione. Il blocco user è una bobina ma è lasciato al programmatore la decisione se utilizzare il risultato presente nell'accumulatore dato dalla catena comunque complessa presente per abilitare questa bobina ad esempio:

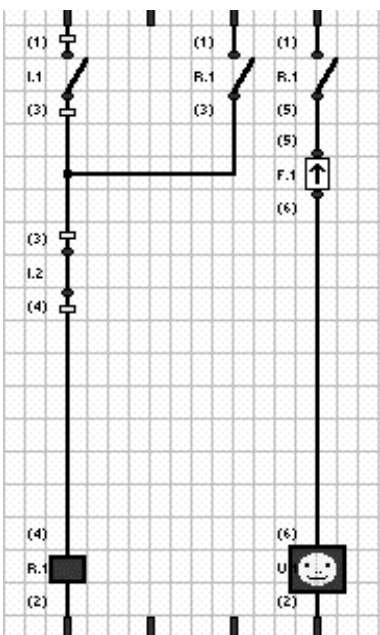


Il seguente circuito di autoritenuta attiva R.1 se I.1 è ON mentre se I.2 è ON R.1 si disattiva, ma se ad esempio il blocco user è in parallelo a R.1 come in figura con una istruzione INC_MEM 1 la memoria 1 sarà incrementata ad ogni scansione di programma, mentre se si sa che la memoria sia incrementata ogni volta che R1 è attivo (o in generale che il blocco user sia attivo ogni volta che la catena di abilitazione è attiva) bisogna implementare il seguente codice:

```
#label          NOABIL
IF_RESET_JUMP  .NOABIL          ;ACC OFF
INC_MEM        1                 ;ACC ON
.NOABIL
```

In questo modo quando viene processato il blocco user viene eseguito un test sull'accumulatore per vedere se è attivo, nel caso l'accumulatore sia zero e quindi la catena di abilitazione da risultato falso non viene eseguita l'istruzione INC_MEM 1. Con questa modifica il programma è cambiato nel seguente modo: La memoria 1 è incrementata ad ogni scansione di programma se R.1 è attivo. Nel caso in cui MEM 1 debba essere un contatore degli stati attivi

di R.1 allora bisogna inserire un fronte di salita in serie al blocco user. ATTENZIONE non si può inserire direttamente un fronte di serie all' blocco user direttamente nello schema di figura, in questo caso si deriverebbe un risultato parziale ca una colonna complessa e il compilatore dopo una serie di prove non troverebbe soluzione. Bisogna invece modificare lo schema come in figura:



Adesso MEM 1 conta quante volte viene attivato R.1.

Il programma che genera il compilatore è il seguente:

0001	{0000}	[51]	SLAVE
0002	{0001}	[52 01]	PLC_NUM 01
0003	{0003}	[53 08]	PLC_COM 08
0004	{0005}	[4e]	ENABLE
0005	{0006}	[3b]	START_PROGRAM
0006	{0007}	[03 01]	LOAD_INPUT 01
0007	{0009}	[26 01]	OR_RELE 01
0008	{0011}	[18 02]	AND_INPUT_NOT 02
0009	{0013}	[0f 01]	STORE_RELE 01
0010	{0015}	[01 01]	LOAD_RELE 01
0011	{0017}	[23 01]	AND_F_UP 01
0012	{0019}	[40 00 18]	IF_RESET_JUMP 0024
0013	{0022}	[80 01]	INC_MEM 01
0014	{0024}	[3c]	END_PROGRAM