



LADDERWORK CON IL CONTROLLORE INDUSTRIALE GRIFO® GPC553

a cura della REDAZIONE -PARTE Ia

Comincia con questo numero la prima parte di questa interessante trattazione. Utile per tutti gli appassionati di elettronica aiuta a capirne di più di quel mondo, abbastanza sconosciuto, dei microprocessori e della loro programmazione.

BREVI CENNI STORICI SUL LINGUAGGIO LADDER

La nostra vita è sempre maggiormente governata dall'elettronica. Dispositivi di automazione elettronici trovano sempre più larga applicazione sia nei grossi impianti che nei comuni elettrodomestici. L'elemento comune che caratterizza questi sistemi è la presenza di un dispositivo denominato microprocessore. I microprocessori sono componenti elet-

tronici relativamente complessi che solo persone con un certo grado di conoscenza sono in grado di programmare. Per fare questo si ricorre all'utilizzo di linguaggi di programmazione come ad esempio tra i più conosciuti il BASIC, il C e il FORTH. Questi linguaggi di programmazione non sono però alla portata di tutti e richiedono un certo livello di conoscenza in ambito informatico. Nel mercato sono stati introdotti da diversi anni alcuni dispositivi denominati PLC (Programmable Logic Controller).

L'obbiettivo di questi dispositivi era quello di mascherare la complessità di programmazione dei microprocessori attraverso una interfaccia utente più intuitiva per gli operatori che installavano questi sistemi.

A tale proposito, sempre più prepotentemente si sta diffondendo uno standard di linguaggio per questi dispositivi PLC denominato linguaggio LADDER. Il linguaggio LADDER è nato alla fine degli anni 60.

L'obbiettivo era quello di creare un linguaggio simbolico visuale molto simile a quelle che erano le convenzioni simboliche degli impianti elettrici. Queste simbologie sono state poi modificate nel corso del tempo per ovviare a problemi di interpretazione logica. Per fare un esempio, il simbolo della lampadina, che voleva rappresentare un'uscita fisica, e'

stata sostituita con un simbolo costituito da due semicerchi verticali.

Questo nuovo simbolo ha però introdotto il concetto di uscita logica.

In pratica quel particolare oggetto, non è più legato ad una uscita reale ma bensì ad una uscita logica in genere che potrà essere legata a sua volta ad ingressi logici per permettere collegamenti di tipo FEED-BACK. Nello stesso modo anche il simbolo dell'interruttore che stava a significare un ingresso fisico, è stato sostituito con un simbolo più generico, caratterizzato da due barre verticali. Tale simbolo non rappresenta più a questo punto un reale nodo di collegamento elettrico del PLC con il mondo esterno, ma bensì rappresenta un punto di ingresso da poter relazionare con qualsiasi altro punto del circuito. A questo punto, il punto successivo è stato quello di aggiungere al linguaggio, tutta quella serie di elementi che erano già presenti sotto forma di dispositivi elettromeccanici nel mondo reale. Sono stati così introdotti simboli che avevano le funzionalità di linee di ritardo, temporizzatori, contatori e blocchi funzionali sempre più complessi.

La potenza del sistema, stava nel fatto che tutto quello che si stava creando, non necessitava di reali modifiche al cablaggio del sistema, ma la funzionalità poteva essere modificata in brevi istanti solamente spo-

stando un collegamento sullo schermo di un computer.

Va da sé che questo ha riscosso, nel mondo dell'automazione, un grande successo. Stabilito una volta sola il cablaggio con il mondo esterno, il moderno PLC così creato poteva cambiare la configurazione dei collegamenti di ingresso e uscita senza muovere un solo filo.

Il linguaggio LADDER permette a chiunque di avvicinarsi al mondo dei PLC attraverso una metodica di programmazione che è molto più vicina alla logica naturale dell'uomo.

Per programmare in linguaggio LADDER non è necessaria nessuna conoscenza informatica ma bensì bisogna solo avere chiaro il funzionamento di un interruttore, di una lampadina o di un relè.

IL SOFTWARE LADDERWORK

Il software LadderWORK, prodotto dalla MicroSHADOW Research di La Spezia, permette di scrivere applicazioni utilizzando la notazione LADDER (o più comunemente denominata logica a contatti) e generare codice per il microprocessore 8051. Tutta la sessione di sviluppo di una applicazione, dall'introduzione dello schema fino alla compilazione, viene eseguita comodamente all'interno di un unico ambiente integrato (IDE). La caratteristica peculiare di LadderWORK è quella di generare direttamente codice macchina per il processore 8051 senza dover passare prima da linguaggio intermedi come ad esempio il 'C'. Questa caratteri-

stica fa sì che il codice generato da LadderWORK sia sempre estremamente ottimizzato sia in dimensione che in velocità di esecuzione. In questo modo, attraverso LadderWORK, è possibile utilizzare controllori industriali di a basso costo ottenendo prestazioni comparabili a PLC di fascia alta. LadderWORK presenta la caratteristica di utilizzare al massimo le risorse dei microprocessori con la possibilità di lavorare con architetture realmente minime. La MicroSHADOW Research e la GRIFO® da molti anni lavorano in stretta collaborazione per poter offrire ai propri clienti una vasta gamma di soluzioni per coprire i problemi relativi all'automazione industriale.

In particolare, il controllore industriale GPC553® con estensione ZBX168, abbinato all'utilizzo di LadderWORK, è risultato essere un prodotto particolarmente versatile per risolvere una vasta gamma di problematiche. In questo articolo cercheremo di dare le linee guida per un primo contatto con questo tipo di programmazione. Nella seconda parte dell'articolo daremo le istruzioni pratiche per utilizzare il pacchetto LadderWORK con il controllore industriale della GRIFO®

GLI ELEMENTI DEL LINGUAGGIO LADDER

Il linguaggio LADDER è indiscutibilmente il linguaggio principe per la risoluzione delle problematiche relative all'automazione industriale. Una delle caratteristiche fondamentali di questo linguaggio è la sua veste

esclusivamente grafica. Infatti il linguaggio LADDER, a differenza di altri linguaggi come il 'C' oppure il BASIC che richiedono la battitura di un file sorgente in formato testo, basa il suo colloquio con l'utente attraverso il disegno di uno schema elettrico. Questo fa subito capire che il linguaggio LADDER ha come caratteristica saliente la semplicità. L'utente non deve in alcun modo preoccuparsi dell'architettura interna del controllore su cui sta lavorando, ma deve solamente pensare al suo progetto come ad uno schema elettrico. Tutta la logica di funzionamento di un particolare controllo viene descritta mediante l'utilizzo di primitive elementari come i contatti ed i relè, il software provvederà alla conversione dello schema elettrico in codice comprensibile dal controllore.

BARRE DI ALIMENTAZIONE

Nello standard LADDER due particolari elementi sono sempre presenti sul nostro foglio di lavoro: la barra di alimentazione positiva e la barra di alimentazione negativa. Questi due oggetti sono presenti rispettivamente sulla sinistra (+) e sulla destra (-) del nostro progetto. Come se si stesse lavorando effettivamente con componenti elettromeccanici reali, queste due barre mi forniscono l'alimentazione per il circuito in questione. In alternativa, per non dover collegare gli elementi sino alla barra di alimentazione negativa è possibile utilizzare il simbolo di GROUND (GND) direttamente in uscita ad un particolare componente.

INGRESSI ED USCITE NEL LINGUAGGIO LADDER

I due elementi sicuramente comuni a tutti gli ambienti di programmazione LADDER sono sicuramente i blocchi di ingresso (INPUT) ed i blocchi di uscita (OUTPUT). La notazione standard per questi due simboli può sembrare un po' strana (in partico-

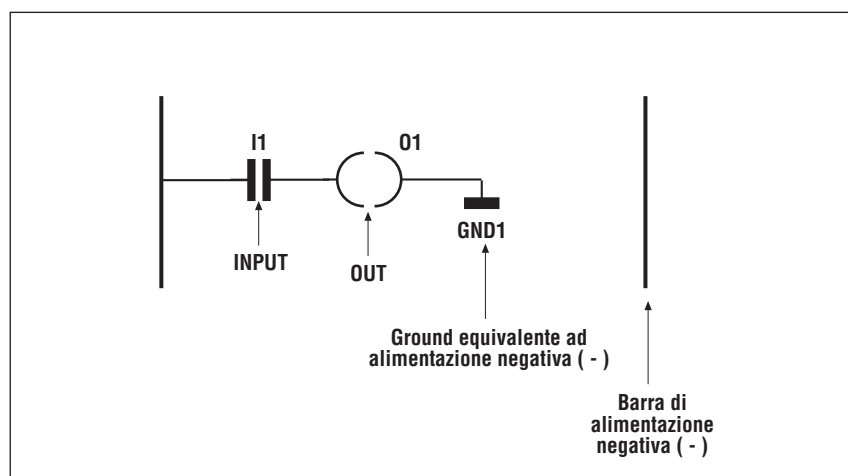


Figura 1. Gli elementi fondamentali del linguaggio LADDER.

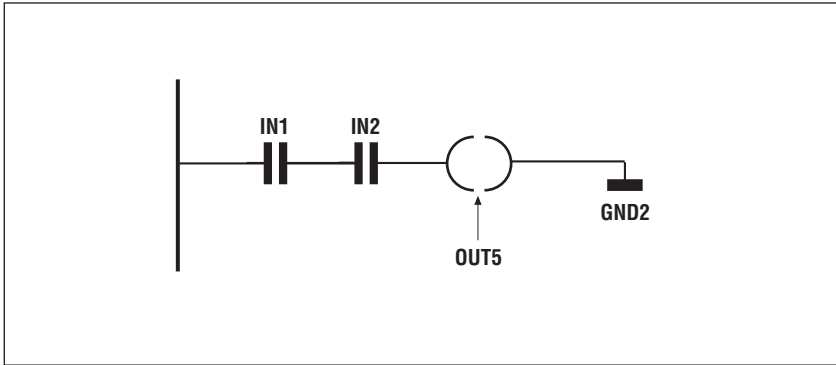


Figura 2. Interruttori in serie.

particolare interruttore. Nello stesso modo se sul foglio di lavoro mettiamo un blocco di OUTPUT e lo configuriamo in associazione al morsetto OUT5 ogni segnale in ingresso al piedino di questo componente andrà fisicamente a pilotare il relè asso-

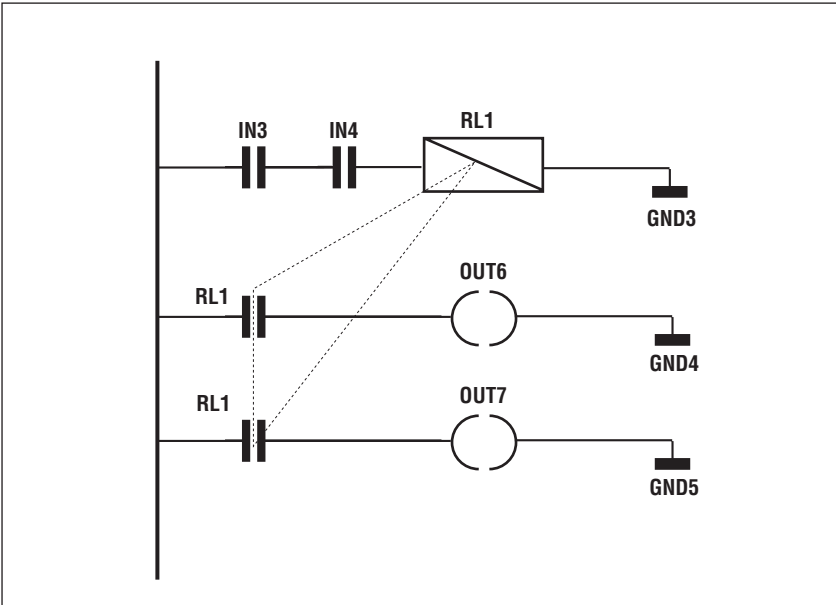


Figura 3. Utilizzo di bobine e contatti.

ciato a quel particolare morsetto. In generale i blocchi di INPUT trasferiscono il segnale applicato sul piedino di sinistra verso il piedino di destra solamente quando il segnale associato è attivo. Per esempio, nello schema di Figura 1, il piedino sinistro del blocco di OUTPUT viene alimentato, attivandone la relativa uscita, quando il segnale I1, associato al blocco di INPUT diventa attivo.

Il trasferimento dei segnali da sinistra verso destra permette di realizzare gli stessi collegamenti tipici degli impianti elettrici. Se per esempio vogliamo attivare una particolare uscita solamente quando due segnali sono entrambi attivi possiamo effettuare lo stesso collegamento che si farebbe in un normale circuito elettrico ovvero mettere in serie i due interruttori (**Figura 2**).

BOBINE E CONTATTI

Come negli impianti elettrici, anche nel linguaggio LADDER esiste il relè (Denominato anche COIL). Proprio come nei relè elettromeccanici è possibile utilizzare i contatti del relè stesso per attivare altre sezioni del circuito. La caratteristica fondamentale di questi componenti è la completa libertà di associare per ogni bobina un numero indeterminato di contatti (NO / NC). L'associazione tra la bobina ed il contatto avviene semplicemente configurando entrambi gli oggetti con lo stesso REFERENCE. Nel circuito di **Figura 3** viene mostrato un esempio di utilizzo

lare il blocco di ingresso ha lo stesso simbolo del condensatore degli schemi elettronici) ma è pur sempre uno standard riconosciuto a livello mondiale (**Figura 1**).

I blocchi di INPUT e OUTPUT possono rappresentare un elemento logico (Variabile) per il passaggio di segnali all'interno della rete oppure rappresentare un ingresso od uscita fisica sulla morsettiera del vostro PLC. Infatti il software riconosce

tutta una serie di identificatori (IN1, IN2, ... OUT1, OUT2 ... etc) che, quando applicati ad un particolare blocco di INPUT/OUTPUT, permettono di pilotare una particolare risorsa HARDWARE del vostro controllore. Se per esempio piazziamo sul nostro foglio di lavoro un blocco di INPUT e lo configuriamo con identificatore IN1, diciamo al sistema che il segnale proveniente dal morsetto omonimo andrà a pilotare quel

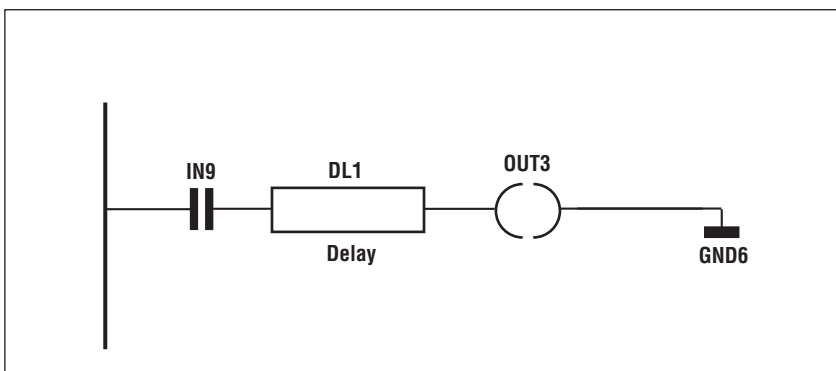


Figura 4. Circuito per luci scale.





Figura 5. Configurazione del componente DELAY per il circuito luci scale. ▲

dei relè. Nello schema, attivando contemporaneamente gli ingressi IN3 e IN4, viene eccitata la bobina del relè RL1 il quale provoca la chiusura dei contatti associati e conseguentemente l'accensione delle uscite OUT6 e OUT7.

TEMPORIZZATORI (DELAY)

I temporizzatori, in generale, sono componenti che permettono di ritardare in uscita, o comunque di modificare, il segnale applicato in ingresso. Esistono molti tipi di temporizzatori e LadderWORK ne mette a disposizione una gamma molto assortita. Il temporizzatore più elementare è denominato DELAY. Questo componente ha essenzialmente due modalità di funzionamento chiamate HOLD e DELAY. Nella modalità DELAY il segnale applicato in ingresso viene ripresentato in uscita dopo che il tempo programmato è trascorso.

Nella modalità DELAY il componente attiva la sua uscita immediatamente dopo la comparsa di un segnale in ingresso e mantiene questo stato per tutta la durata del tempo programmato. Per fare un esempio pratico, utilizzando la modalità DELAY di questo componente è possibile realizzare con tre soli blocchi la

funzionalità di LUCE SCALE, ovvero quel semplice meccanismo, normalmente realizzato con componenti elettromeccanici, che permette di tenere accese le luci delle scale per un certo numero di secondi dopo la pressione di un pulsante. Riferendosi allo schema di **Figura 4**, il pulsante di accensione collegato a IN9 avvia la temporizzazione del blocco DL1 il quale attiva la propria uscita, e quindi anche il blocco OUT3, per un tempo programmato. Se per esempio si desidera tenere accesa la luce per 30 secondi il componente DELAY deve essere configurato come da **Figura 5**.

CONTATORI

Questi componenti permettono di contare gli impulsi che vengono applicati al proprio ingresso. I contatori vengono utilizzati normalmente per produrre determinati effetti al raggiungimento di un particolare valore di conteggio. Per fare un piccolo esempio possiamo immaginare di

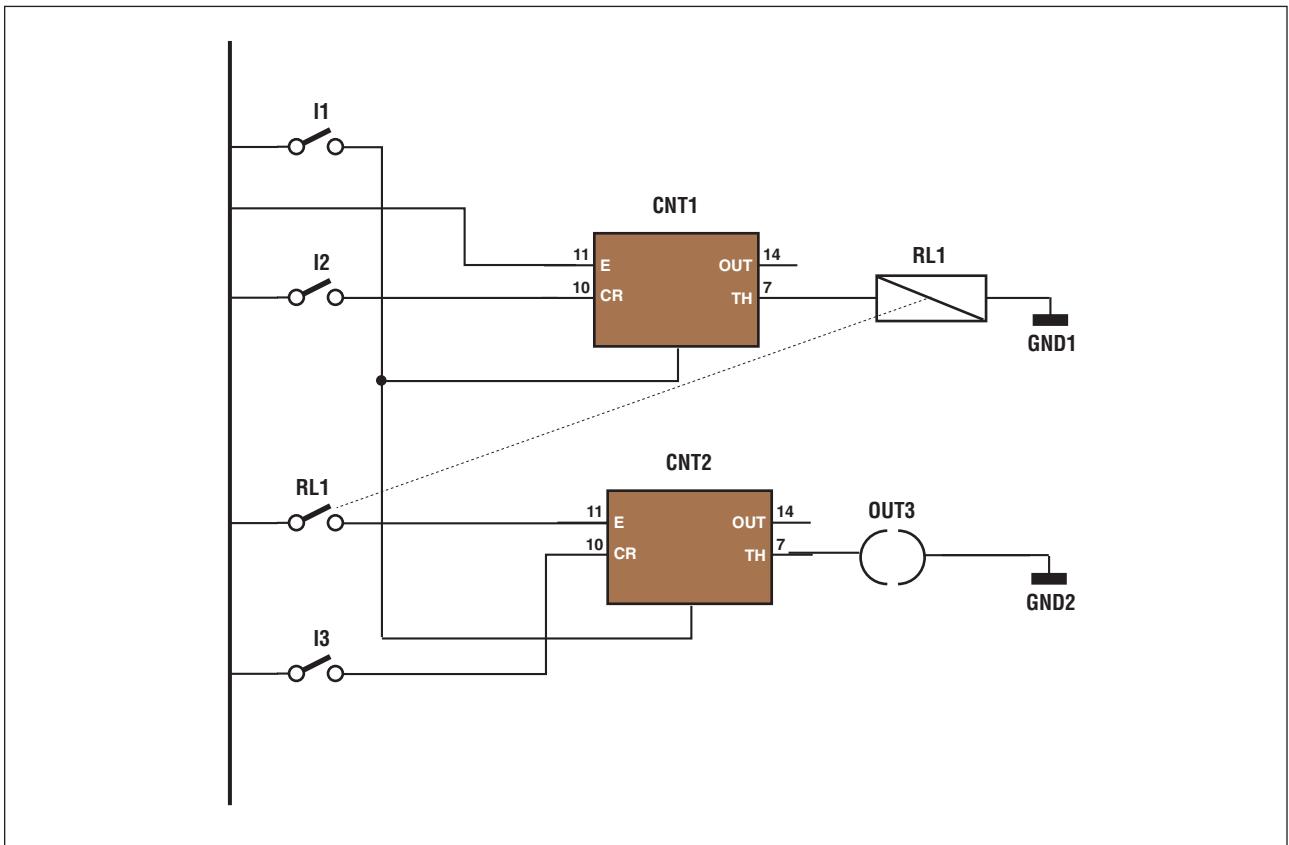


Figura 6. Chiave elettronica per porte.

realizzare un semplice circuito per una chiave elettronica in cui, attraverso tre pulsanti, bisogna impostare una combinazione segreta che comanderà l'apertura di una porta. Il software LadderWORK mette a disposizione una vasta gamma di questi dispositivi di conteggio ma in questo particolare esempio andremo ad utilizzare il più semplice. Il contatore più elementare è denominato COUNTER ed è caratterizzato dalla possibilità di contare sia in avanti che indietro con valori di conteggio massimo pari a 65535 impulsi. Questo contatore dispone di tre ingressi : E , CK , R e di due uscite OUT e TH. Il conteggio, che si ottiene mediante impulsi sul segnale CK, è abilitato quando il segnale E (ENABLE) è posto a livello attivo (+) . Il segnale OUT riporta il valore attuale del conteggio mentre l'uscita TH viene posta a livello attivo quando il contatore raggiunge il massimo valore impostato. Dopo il raggiungimento del valore impostato, al successivo impulso di clock, il contatore si posiziona

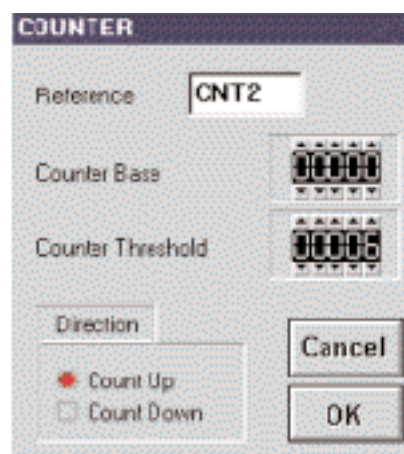
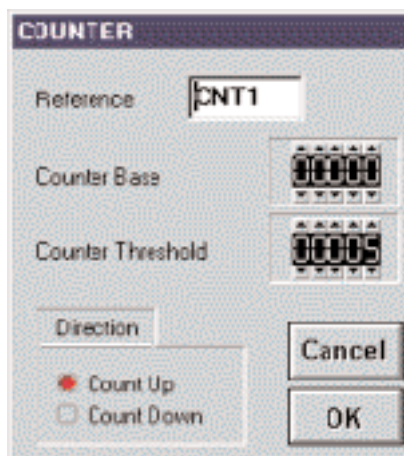


Figura 7a-b. Configurazione dei contatori CNT1 e CNT2 per la chiave elettronica.

nuovamente sul valore iniziale (RING COUNTER **Figura 6**). Il funzionamento del circuito è semplice. La presenza del segnale associato all'ingresso I1 permette di azzerare i due contatori CNT1 e CNT2. Attivando per cinque volte l'ingresso I2 si porta il segnale TH di CNT1 a livello attivo il quale attraverso RL1 abilita il secondo contatore CNT2. A questo punto sei successivi impulsi di clock mediante attivazione dell'ingresso I3 attiveranno l'uscita O1. Il circuito viene azzerato mediante pressione di I1 vedi **Figure 7a e b**. Finisce così questa prima parte, il prossimo mese parleremo dei generatori di impulsi, dei circuiti antirimbombo e tanti altri interessanti argomenti.