Esempi uPLC - Instruction List (IL)

Usa l'interfaccia grafica online per la programmazione ladder:
https://www.ni-lab.online/new_websmart/nilab_ladder_editor.php

Per scrivere o modificare il programma usa l'editor uPLC dentro il software NILAB Starter:
https://www.nilab.at/dokuwiki/doku.php?id=nilab_starter:uplc_editor

Esempio 1 - Copiare ingresso digitale su uscita digitale

; IN0 (bit0 di reg 4620) → OUT0 (bit0 di reg 4621)

LD 4620, 0 ; carica bit0 del registro ingressi digitali
OUT 4621, 0 ; scrivi il risultato su bit0 del registro uscite digitali
END



Esempio 2 — Ingresso invertito

; OUT1 è ON quando IN0 è OFF (logica contatto NC)

LDN 4620, 0 ; carica NOT(bit0 dell'ingresso digitale)
OUT 4621, 1 ; scrivi il risultato su bit1 del registro uscite digitali
END



Esempio 3 — Logica AND tra due ingressi

; OUT0 = IN0 AND IN1

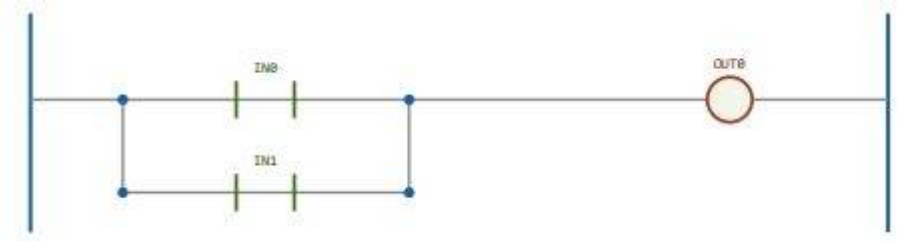
LD 4620, 0 ; carica IN0
AND 4620, 1 ; AND con IN1
OUT 4621, 0 ; scrivi il risultato su OUT0
END



Esempio 4 — Logica OR tra due ingressi

```
; OUT0 = IN0 OR IN1

LD 4620, 0 ; carica IN0
OR 4620, 1 ; OR con IN1
OUT 4621, 0 ; scrivi il risultato su OUT0
END
```



Esempio 5 — Latch SET/RESET

```
; SET: IN0 imposta OUT0
; RESET: IN1 azzera OUT0
; l'uscita mantiene il suo stato quando entrambi gli ingressi sono OFF

LD 4620, 0 ; carica IN0
SET 4621, 0 ; se stack=1, imposta OUT0 (latch ON)

LD 4620, 1 ; carica IN1
RES 4621, 0 ; se stack=1, azzera OUT0 (latch OFF)
END
```



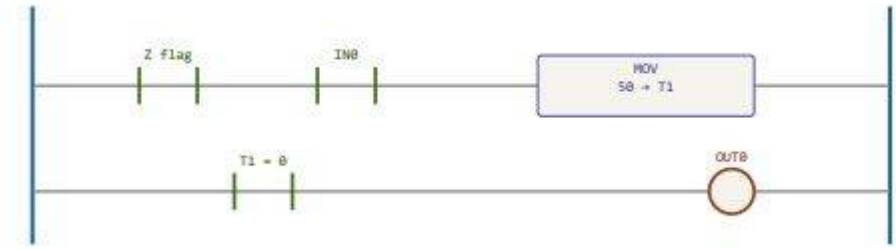
Esempio 6 — Timer di ritardo all'accensione

```
; Quando IN0 va alto, carica Timer1 con 50 cicli di scan (~300ms)
```

; OUT0 diventa ON quando il timer scade

; Ricarica il timer solo quando IN0 è alto e il timer è scaduto
LD 8671, 3 ; carica flag Z: impostato quando Timer1 == 0 (scaduto)
AND 4620, 0 ; AND con IN0
MOV 8678, 8028 ; ricarica Timer1 con costante 10 (reg 8678 = 10) ; sostituisci 8678 con un registro utente per ritardi più lunghi

LD 8671, 0 ; carica bit0 del registro di stato: Timer1 scaduto
OUT 4621, 0 ; OUT0 ON quando il timer è scaduto
END

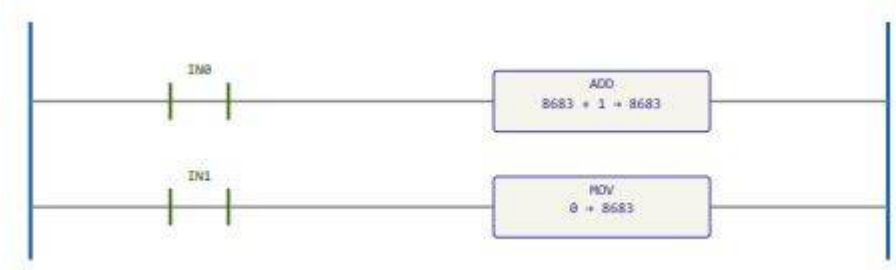


Esempio 7 — Contatore di ciclo

; Incrementa il registro utente 8683 a ogni ciclo di scan quando IN0 è attivo

LD 4620, 0 ; condizione: IN0 attivo
ADD 8683, 8676, 8683 ; 8683 = 8683 + 1 (costante 1 memorizzata in reg 8676)

; Azzera il contatore quando IN1 è attivo
LD 4620, 1 ; carica IN1
MOV 8675, 8683 ; 8683 = 0 (costante 0 memorizzata in reg 8675)
END

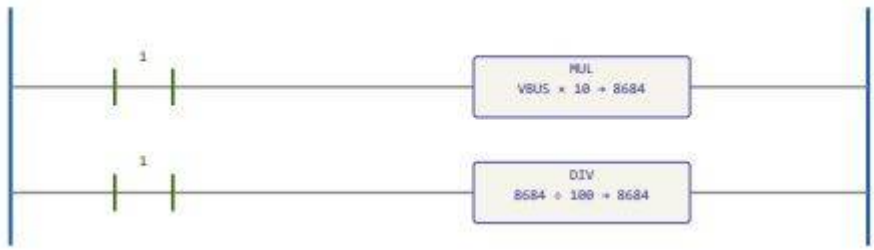


Esempio 8 — Ridimensionamento valore analogico

; Ridimensiona la tensione VBUS (reg 4631) per fattore 1/10
; formula: 8684 = 4631 * 10 / 100

LD 8676, 0 ; stack = 1 (sempre vero, costante 1 in reg 8676)
MUL 4631, 8678, 8684 ; 8684 = VBUS * 10 (costante 10 in reg 8678)
DIV 8684, 8679, 8684 ; 8684 = (VBUS * 10) / 100 (costante 100 in reg 8679)

END

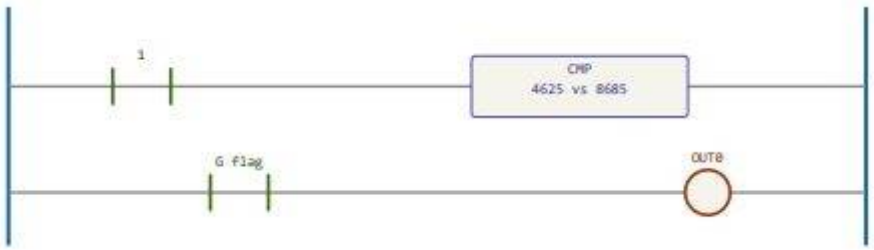


Esempio 9 — CMP: uscita quando la posizione supera la soglia

; OUT0 diventa ON quando la posizione effettiva del motore (reg 4625, low word)
; supera il valore di soglia memorizzato nel registro utente 8685

LD 8676, 0 ; stack = 1 (sempre vero)
CMP 4625, 8685 ; confronta posizione vs soglia, imposta bit4 (flag G) in reg 8671 se 4625 > 8685

LD 8671, 4 ; carica flag G (bit4): 1 se posizione > soglia
OUT 4621, 0 ; OUT0 ON quando la posizione supera la soglia
END



Esempio 10 — Sequencer a due fasi con timer

; Ciclo automatico a due fasi:
; Fase 1: OUT0 ON per 50 cicli di scan (~300ms)
; Fase 2: OUT1 ON per 100 cicli di scan (~600ms)
; La sequenza si ripete continuamente
;
; Variabili usate:
; reg 8683 bit0 = flag fase corrente (0=fase1, 1=fase2)
; Timer1 (reg 8028) controlla entrambe le durate di fase

; — Ricarica il timer quando è scaduto e siamo in fase 1 —
LD 8671, 3 ; flag Z: Timer1 == 0 (scaduto)
ANDN 8683, 0 ; AND NOT flag fase (siamo in fase 1)
MOV 8678, 8028 ; ricarica Timer1 con 10 cicli ; sostituisci 8678 con un registro utente impostato a 50

in produzione

```
; — Passa alla fase 2 quando Timer1 scade durante la fase 1 —
LD 8671, 0 ; Timer1 scaduto (bit0 del reg di stato)
ANDN 8683, 0 ; siamo attualmente in fase 1
SET 8683, 0 ; imposta flag fase → entra in fase 2

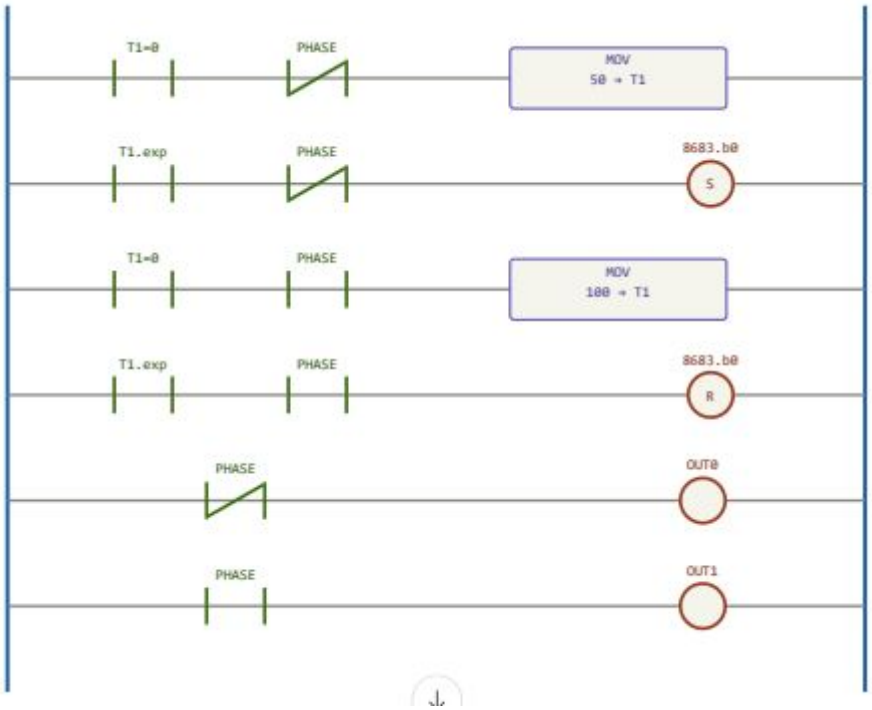
; — Ricarica il timer per la durata della fase 2 —
LD 8671, 3 ; Timer1 scaduto
AND 8683, 0 ; siamo attualmente in fase 2
MOV 8679, 8028 ; ricarica Timer1 con 100 cicli (costante 100 in reg 8679)

; — Torna alla fase 1 quando Timer1 scade durante la fase 2 —
LD 8671, 0 ; Timer1 scaduto
AND 8683, 0 ; siamo attualmente in fase 2
RES 8683, 0 ; cancella flag fase → torna alla fase 1

; — Uscite drive basate sulla fase corrente —
LDN 8683, 0 ; NOT flag fase = siamo in fase 1
OUT 4621, 0 ; OUT0 ON durante la fase 1

LD 8683, 0 ; flag fase impostato = siamo in fase 2
OUT 4621, 1 ; OUT1 ON durante la fase 2

END
```

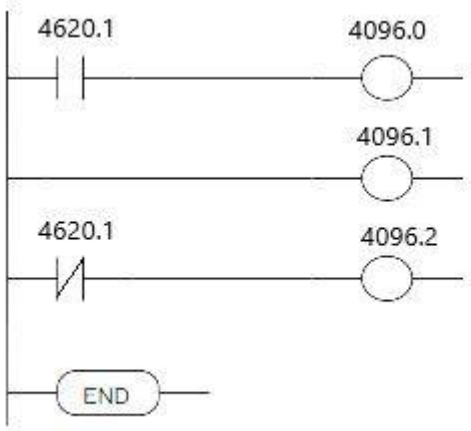


Esempio 11 - Abilitare il motore usando l'ingresso digitale 1

Questo è un semplice programma per abilitare il motore con l'ingresso digitale 1. Il parametro 4620 è lo stato degli ingressi digitali, il parametro 4096 è il command word del drive (3⇒abilitazione motore,

4⇒ disabilitazione motore)

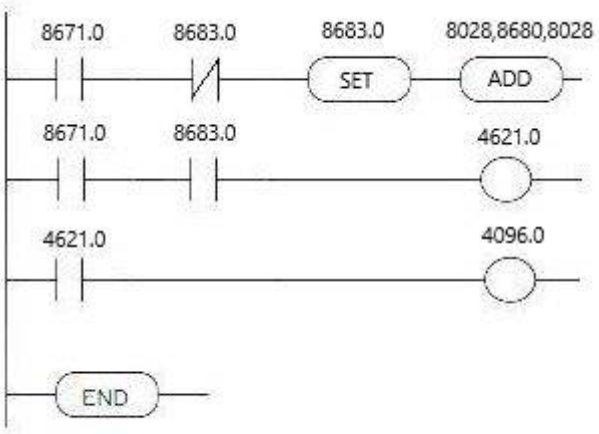
```
LD 4620.1
OUT 4096.0
OUT 4096.1
LDn 4620.1
OUT 4096.2
END
```



Esempio 12 - Abilitare il motore 6 secondi dopo l'avvio del PLC

Questo programma imposta il Timer 1 a 1000 (usando il parametro costante 8680) e un bit di flag (8683.0) per iniziare a contare 6 × 1000 = 6 sec.
Dopo 6 secondi l'uscita digitale 0 è impostata a 1 e il motore è abilitato usando il control word (Parametro 4096).

```
LD 8671.0
ANDn 8683.0
SET 8683.0
ADD 8028 8680 8028
LD 8671.0
AND 8683.0
OUT 4621.0
LD 4621.0
OUT 4096.0
END
```



From:
https://www.nilab.at/dokuwiki/ - **NiLAB GmbH Knowledgebase**

Permanent link:
https://www.nilab.at/dokuwiki/doku.php?id=it:integrated_drive_motors:uplc_example

Last update: **2026/09/13 19:56**