uPLC-Beispiele - Instruction List (IL)

Bitte verwenden Sie die Online-Grafikoberfläche für die Ladder-Programmierung: https://www.ni-lab.online/new_websmart/nilab_ladder_editor.php

Um das Programm zu schreiben oder zu bearbeiten, verwenden Sie bitte den uPLC-Editor in der NILAB-Starter-Software: https://www.nilab.at/dokuwiki/doku.php?id=nilab_starter:uplc_editor

Beispiel 1 - Digitalen Eingang auf digitalen Ausgang kopieren

; IN0 (Bit0 von Reg 4620) → OUT0 (Bit0 von Reg 4621)

LD 4620, 0 ; Bit0 des digitaleingang-Registers laden
OUT 4621, 0 ; Ergebnis auf Bit0 des digitalausgang-Registers schreiben
END



Beispiel 2 — Invertierter Eingang

; OUT1 ist EIN, wenn IN0 AUS ist (NC-Kontaktlogik)

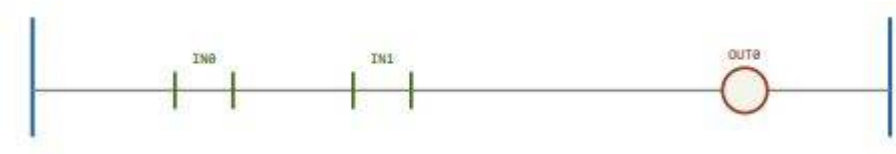
LDN 4620, 0 ; NICHT(Bit0 des digitaleingangs) laden
OUT 4621, 1 ; Ergebnis auf Bit1 des digitalausgang-Registers schreiben
END



Beispiel 3 — UND-Logik zwischen zwei Eingängen

; OUT0 = IN0 UND IN1

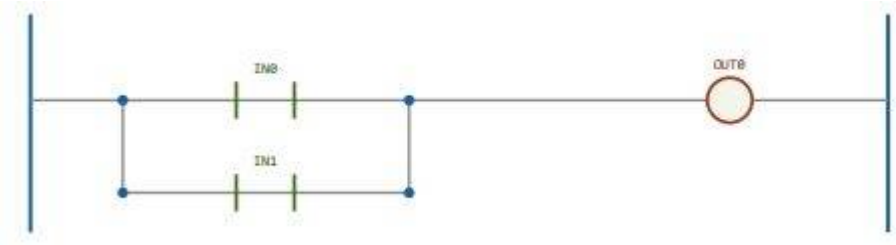
LD 4620, 0 ; IN0 laden
AND 4620, 1 ; UND mit IN1
OUT 4621, 0 ; Ergebnis auf OUT0 schreiben
END



Beispiel 4 — ODER-Logik zwischen zwei Eingängen

; OUT0 = IN0 ODER IN1

LD 4620, 0 ; IN0 laden
OR 4620, 1 ; ODER mit IN1
OUT 4621, 0 ; Ergebnis auf OUT0 schreiben
END

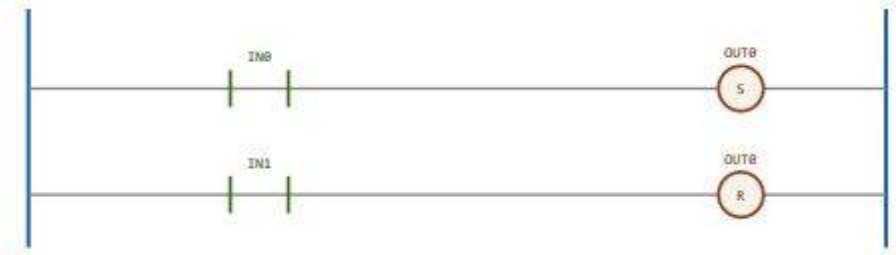


Beispiel 5 — SET/RESET-Latch

; SET: IN0 setzt OUT0
; RESET: IN1 setzt OUT0 zurück
; Ausgang hält seinen Zustand, wenn beide Eingänge AUS sind

LD 4620, 0 ; IN0 laden
SET 4621, 0 ; wenn Stack=1, OUT0 setzen (verriegelt EIN)

LD 4620, 1 ; IN1 laden
RES 4621, 0 ; wenn Stack=1, OUT0 zurücksetzen (verriegelt AUS)
END

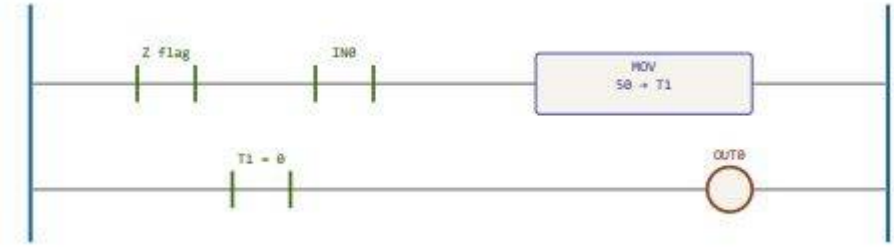


Beispiel 6 — Ansprechverzögerungs-Timer

; Wenn IN0 hoch geht, Timer1 mit 50 Scan-Zyklen (~300ms) laden
; OUT0 schaltet EIN, wenn der Timer abläuft

; Timer nur neu laden, wenn IN0 hoch und Timer abgelaufen ist
LD 8671, 3 ; Z-Flag laden: gesetzt, wenn Timer1 == 0 (abgelaufen)
AND 4620, 0 ; UND mit IN0
MOV 8678, 8028 ; Timer1 mit Konstante 10 neu laden (Reg 8678 = 10) ; 8678 durch ein Benutzerregister für längere Verzögerungen ersetzen

LD 8671, 0 ; Bit0 des Statusregisters laden: Timer1 abgelaufen
OUT 4621, 0 ; OUT0 EIN, wenn der Timer abgelaufen ist
END

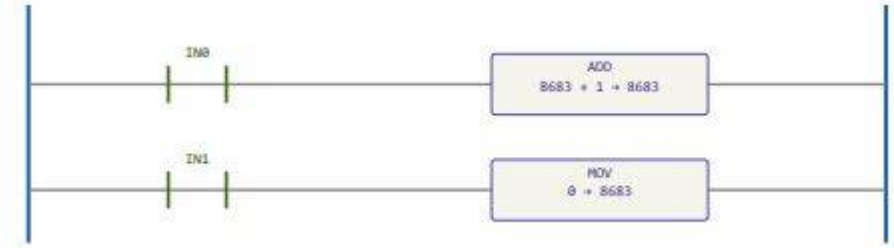


Beispiel 7 — Zykluszähler

; Benutzerregister 8683 bei jedem Scan-Zyklus erhöhen, wenn IN0 aktiv ist

LD 4620, 0 ; Bedingung: IN0 aktiv
ADD 8683, 8676, 8683 ; 8683 = 8683 + 1 (Konstante 1 in Reg 8676 gespeichert)

; Zähler zurücksetzen, wenn IN1 aktiv ist
LD 4620, 1 ; IN1 laden
MOV 8675, 8683 ; 8683 = 0 (Konstante 0 in Reg 8675 gespeichert)
END

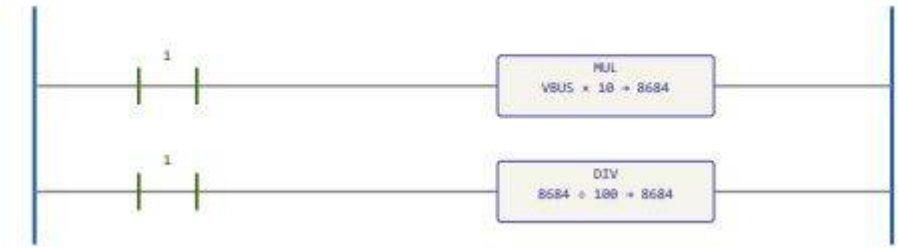


Beispiel 8 — Analogwert-Skalierung

; VBUS-Spannung (Reg 4631) um Faktor 1/10 skalieren
; Formel: 8684 = 4631 * 10 / 100

LD 8676, 0 ; Stack = 1 (immer wahr, Konstante 1 in Reg 8676)
MUL 4631, 8678, 8684 ; 8684 = VBUS * 10 (Konstante 10 in Reg 8678)

```
DIV 8684, 8679, 8684 ; 8684 = (VBUS * 10) / 100 (Konstante 100 in Reg 8679)
END
```



Beispiel 9 — CMP: Ausgang, wenn Position den Schwellwert überschreitet

```
; OUT0 schaltet EIN, wenn die tatsächliche Motorposition (Reg 4625, low word)
; den in Benutzerregister 8685 gespeicherten Schwellwert überschreitet

LD 8676, 0 ; Stack = 1 (immer wahr)
CMP 4625, 8685 ; Position mit Schwellwert vergleichen, setzt Bit4 (G-Flag) in Reg 8671, wenn 4625 > 8685

LD 8671, 4 ; G-Flag laden (Bit4): 1, wenn Position > Schwellwert
OUT 4621, 0 ; OUT0 EIN, wenn Position den Schwellwert überschreitet
END
```



Beispiel 10 — Zweiphasen-Sequencer mit Timern

```
; Automatischer Zweiphasen-Zyklus:
; Phase 1: OUT0 EIN für 50 Scan-Zyklen (~300ms)
; Phase 2: OUT1 EIN für 100 Scan-Zyklen (~600ms)
; Sequenz wiederholt sich kontinuierlich
;
; Verwendete Variablen:
; Reg 8683 Bit0 = aktuelles Phasen-Flag (0=Phase1, 1=Phase2)
; Timer1 (Reg 8028) steuert beide Phasendauern

; — Timer neu laden, wenn abgelaufen und wir in Phase 1 sind —
LD 8671, 3 ; Z-Flag: Timer1 == 0 (abgelaufen)
```

ANDN 8683, 0 ; UND NICHT Phasen-Flag (wir sind in Phase 1)
MOV 8678, 8028 ; Timer1 mit 10 Zyklen neu laden ; 8678 durch ein Benutzerregister ersetzen, das in Produktion auf 50 gesetzt ist

; — Zu Phase 2 wechseln, wenn Timer1 während Phase 1 abläuft —
LD 8671, 0 ; Timer1 abgelaufen (Bit0 des Status-Registers)
ANDN 8683, 0 ; wir sind derzeit in Phase 1
SET 8683, 0 ; Phasen-Flag setzen → Phase 2 betreten

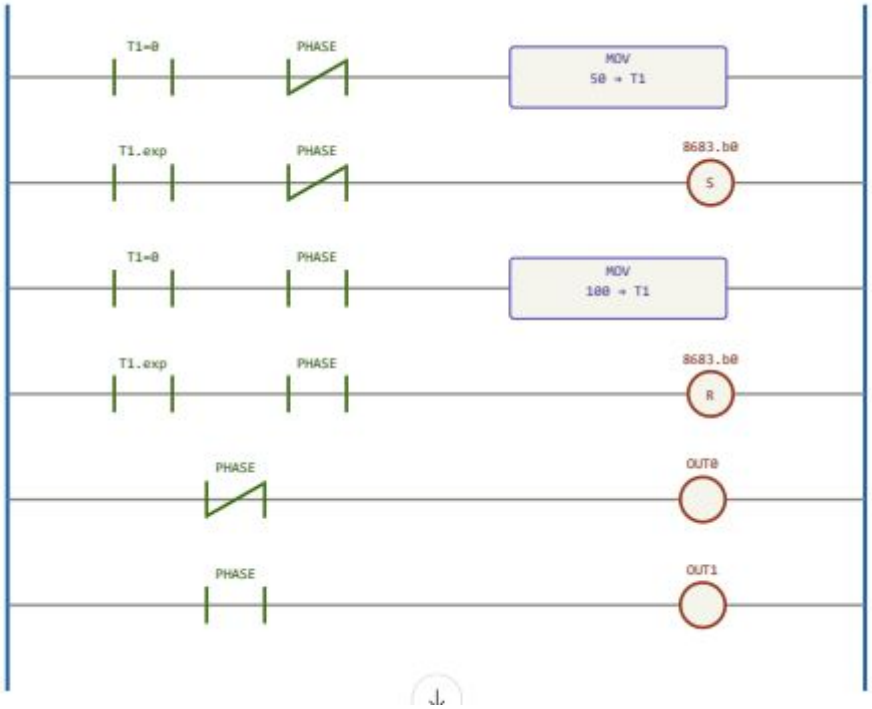
; — Timer für Phase-2-Dauer neu laden —
LD 8671, 3 ; Timer1 abgelaufen
AND 8683, 0 ; wir sind derzeit in Phase 2
MOV 8679, 8028 ; Timer1 mit 100 Zyklen neu laden (Konstante 100 in Reg 8679)

; — Zu Phase 1 zurückwechseln, wenn Timer1 während Phase 2 abläuft —
LD 8671, 0 ; Timer1 abgelaufen
AND 8683, 0 ; wir sind derzeit in Phase 2
RES 8683, 0 ; Phasen-Flag löschen → zurück zu Phase 1

; — Antriebsausgänge basierend auf aktueller Phase —
LDN 8683, 0 ; NICHT Phasen-Flag = wir sind in Phase 1
OUT 4621, 0 ; OUT0 EIN während Phase 1

LD 8683, 0 ; Phasen-Flag gesetzt = wir sind in Phase 2
OUT 4621, 1 ; OUT1 EIN während Phase 2

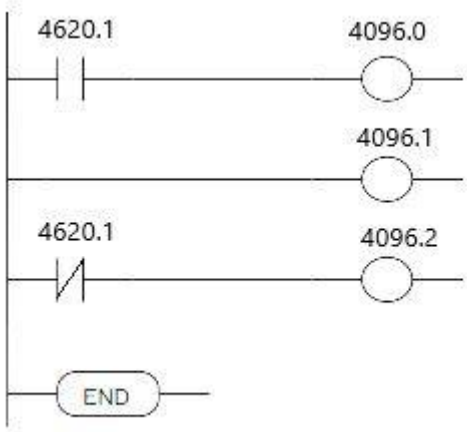
END



Beispiel 11 - Motor mit digitaleingang 1 aktivieren

Dies ist ein einfaches Programm, um den Motor mit digitaleingang 1 zu aktivieren. Parameter 4620 ist der Status der digitaleingänge, Parameter 4096 ist das command word des Antriebs (3⇒Motor aktivieren, 4⇒Motor deaktivieren)

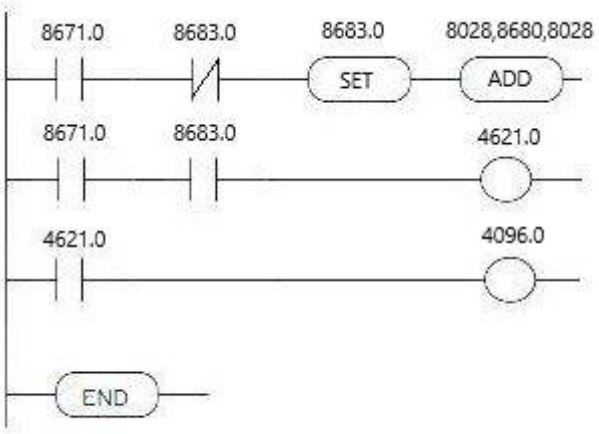
```
LD 4620.1
OUT 4096.0
OUT 4096.1
LDn 4620.1
OUT 4096.2
END
```



Beispiel 12 - Motor 6 Sekunden nach SPS-Start aktivieren

Dieses Programm setzt Timer 1 auf 1000 (mit Konstante Parameter 8680) und ein Flag-Bit (8683.0), um 6 × 1000 = 6 Sekunden zu zählen.
Nach 6 Sekunden wird Digitalausgang 0 auf 1 gesetzt und der Motor wird mit dem control word (Parameter 4096) aktiviert.

```
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=de:integrated_drive_motors:uplc_example

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