

# uPLC Structured Text Compiler

## Introduction to the ST to IL Converter

The **ST to IL Converter** is a specialized utility designed to bridge the gap between high-level programming and drive-level execution. This tool allows developers to write control logic using **Structured Text (ST)**—a high-level, Pascal-like language—and automatically convert it into the **Instruction List (IL)** format required by our integrated drives.

This tool is ideal for users who need to implement complex mathematical algorithms, data handling, or conditional loops that are often more efficiently managed via text-based coding than graphical environments.

You can use this tool here: [https://www.ni-lab.online/new\\_websmart/nilab\\_st\\_converter.php](https://www.ni-lab.online/new_websmart/nilab_st_converter.php)

## How it Works with NiLAB Starter

The converter is fully integrated into the NiLAB development workflow, providing a flexible alternative to the Ladder Editor:

```
ST TO IL CONVERTER
uplc_program
CONVERT
DOWNLOAD IL
SAVE ST
LOAD ST
EXAMPLE
CLEAR
SIMULATOR
SHOW REFERENCE
37/44

STRUCTURED TEXT INPUT
(* NiLAB uPLC - 4-Step Motion Sequence *)
(* *)
(* Constants: R675=0 R676=1 R677=10 R678=100 R679=1000 *)
(* Set Parameter 1792 to 5 Raps, 2 positions per rap *)
(* Configure 10 position tanks in the motion controller *)

VAR_ALIAS
  DI0 := R620.0;
  DI1 := R620.1;
END_VAR_ALIAS

R683 := R675 + R676;
R684 := R683 + R683;
R685 := R684 + R682;

IF DI0 AND DI1 THEN
  R622 := R685;
END_IF;

IF DI0 AND NOT DI1 THEN
  R622 := R684;
END_IF;

IF NOT DI0 AND DI1 THEN
  R622 := R683;
END_IF;

IF NOT DI0 AND NOT DI1 THEN
  R622 := 0;
END_IF;

Inet: 37 steps compiled
a Optimizer: 24 -> 21 steps (-3, -12%). SPT-1: 3.

INSTRUCTION LIST
LD  R676,0
ADD R676,R676,R683
ADD R683,R683,R684
ADD R684,R683,R685
LD  R620,0
AND R620,I
MOV R685,R622
LD  R620,0
ANDI R620,I
MOV R684,R622
LDI R620,0
AND R620,I
MOV R685,R622
LDI R620,1
MOV R675,R622
LD  R676,0
ENI R622,R686
LDI R673,3
MOV R676,R622
MOV R622,R686
END
```

- **Code Development:** Write your control algorithms in the ST editor using standard syntax (IF-THEN-ELSE, CASE, FOR/WHILE loops, etc.).
- **Conversion:** The tool parses the Structured Text and generates an optimized **Instruction List (IL)** file.
- **Integration:** The resulting IL file is loaded into **NiLAB Starter**, which compiles the code specifically for the integrated drive's architecture.
- **Deployment:** Using NiLAB Starter, the compiled program is downloaded directly into the

motor, ready for execution.

## Key Advantages

- **Flexible Programming:** Choose the best language for the task—use ST for complex logic and Ladder for sequential control.
- **Seamless Translation:** Automated conversion eliminates manual coding errors when moving from ST to IL.
- **Optimized Performance:** The converter generates clean Instruction List code to ensure efficient execution within the drive’s internal uPLC.
- **Unified Workflow:** Both the ST Converter and the Ladder Editor share the same deployment path via NiLAB Starter, providing a consistent user experience.

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

Permanent link:  
[https://www.nilab.at/dokuwiki/doku.php?id=integrated\\_drive\\_motors:st\\_converter](https://www.nilab.at/dokuwiki/doku.php?id=integrated_drive_motors:st_converter)

Last update: **2026/04/21 18:08**

