DOCUMENT
LADV system manual
LADV-SM

PROJECT
Wartens LADV virtual PLCs
Rev 1.0

LADV system manual

How a Wartens LADV controller behaves: modes, the scan, I/O, memory, restarts, online functions and diagnostics.

Document	LADV system manual
Code	LADV-SM
Revision	1.0, 1 November 2026
Firmware	V1.0.0
Issued by	Wartens, makers of LADX

> Wartens LADV controllers are virtual PLCs that run in a web browser, for learning, for testing logic and for preparing for commissioning. They are not real time, not safety rated, and they cannot control equipment.

1. Intended use

LADV controllers are for learning PLC programming, for testing control logic against a simulated process, and for preparing a program before it meets real hardware. A program that passes in the lab still has to be commissioned on the real controller, with the real I/O and the real safety system. Results are behavioural: timing and edge cases on a particular vendor's controller can differ.

2. The LADV line

CPU	Series	DI	DO	AI	AO	Slots	Fastest scan	Retentive
CPU 110	LADV100	8	6	0	0	0	20 ms	No
CPU 120	LADV100	12	8	2	1	0	20 ms	Yes
CPU 210	LADV200	16	16	4	2	0	10 ms	Yes
CPU 230	LADV200	24	16	8	4	0	10 ms	Yes
CPU 310	LADV300	8	8	0	0	8	5 ms	Yes
CPU 350	LADV300	8	8	0	0	16	5 ms	Yes

3. Operating modes

A controller is always in one of three modes. Outputs are off unless it is in RUN.

Mode	What happens	How it is left
STOP	The program does not execute. Outputs are off. Inputs are still read and shown.	RUN, if a program is loaded.
RUN	The program executes every scan period. Outputs follow the program.	STOP, a download, or a fault.
FAULT	A runtime error or the watchdog stopped execution. Outputs are off.	Reset the fault, which goes to STOP, or download again.

A download always puts the controller in STOP. RUN is refused while faulted, and refused with no program loaded; the diagnostic buffer says which.

4. The scan

Each scan has three phases, as on hardware:

1. Input: the field values, or the forced values, are copied into the input image. A change halfway through a scan is not seen until the next one.
2. Program: the program executes once, top to bottom for ladder, every task that is due for Structured Text.
3. Output: the output image is presented to the field, with forces applied.

Scans run on simulated time at the scan period. If the browser falls behind, for example a tab left in the background, the controller runs up to 25 scans to catch up and then drops the rest, counting them as skipped scans and logging W310, rather than freezing the page to replay them.

The scan period is the program's scan time, 100 ms unless the program sets it, and never faster than the CPU's fastest scan. A Structured Text program with tasks scans at the interval of its fastest task.

5. I/O and addressing

Channels are numbered from 0 within each kind: DI0, DO0, AI0, AO0. On a modular controller, module channels continue from the CPU's own, in slot order.

Kind	Structured Text address	Ladder
Digital input n	%IXb.c where n = 8b + c, so DI10 is %IX1.2	The n-th BOOL input tag
Digital output n	%QXb.c, as above	The n-th BOOL output tag
Analog input n	%IWn	The n-th INT input tag
Analog output n	%QWn	The n-th INT output tag

Analog channels carry INT values from 0 to 1000, meaning 0 to 100 percent of span, across the whole line. A program that needs a channel the controller does not have is refused at download with the channel named.

6. Program memory

Memory is counted in units a programmer can count. Structured Text is counted in the same table as ladder, so one capacity serves both:

- a variable counts as a tag;
- 8 statements count as a rung;
- 2 program organisation units (PROGRAM, FUNCTION_BLOCK, FUNCTION) count as a routine.

CPU	Tags or variables	Rungs	ST statements	Routines	ST POUs
CPU 110	64	32	256	2	4
CPU 120	128	64	512	4	8
CPU 210	512	256	2048	16	32
CPU 230	1024	512	4096	32	64
CPU 310	4096	2048	16384	64	128
CPU 350	8192	4096	32768	128	256

7. Restarts and retentive memory

- Warm restart, STOP to RUN on a retentive CPU: ladder memory is kept, and in Structured Text the RETAIN variables keep their values while every other variable takes its initial value, as IEC 61131-3 defines.
- Cold restart: every variable, RETAIN included, takes its initial value. A non-retentive CPU always starts cold. The Cold restart

command does it on any CPU.

- Download: memory starts cold.

8. Watchdog

Each scan is timed on the real clock. A scan that takes longer than the CPU's watchdog time faults the controller with F300. Structured Text is also checked every 1,024 statements during the scan, so a loop that never ends faults rather than hanging the page.

9. Online functions

Watch. Every variable of a Structured Text program is shown live, with its type, its address and whether it is RETAIN. A ladder program shows power flow on its rungs.

Modify. Writes a value into memory once. The program may overwrite it on the next scan, and an input is overwritten by the field at the next input phase.

Force. Holds an I/O point at a value until the force is removed. A forced input is what the program reads, whatever the field does. A forced output is what the field gets in RUN, whatever the program writes. While anything is forced the MAINT lamp is lit and the display shows FORCE. Forces are logged (W400 on, I401 and I402 off), kept when a controller is saved and restored, and dropped at download for any point the new program does not have. Only I/O can be forced.

Forcing is how a real machine is commissioned and how real machines surprise people. A forgotten force is one of the most common causes of a machine doing something its program does not say.

10. Diagnostic buffer

The last 64 events, newest first, each with the time since power on, a level and a code: mode changes, downloads, warnings, forces and faults. The codes are listed in LADV-DC.

11. Saving and restoring

A saved controller comes back in STOP, whatever mode it was saved in: nothing should start moving because a page was reopened. A retentive CPU comes back with its memory; a non-retentive one comes back cold, as hardware does after power is removed.

12. Digital twins

In the lab a controller is wired to a LADX Twin, a simulated process with sensors, actuators and objectives that judge the program as it runs. The twin's signals are wired by name: a ladder tag or a Structured Text variable with the twin's signal name is connected to it. Structured Text names are not case sensitive, so MOTOR reaches the twin's Motor.