DOCUMENT
Structured Text standard library reference
LADV-IR-ST

PROJECT
Wartens LADV virtual PLCs
Rev 1.0

Structured Text standard library reference

Every standard function and function block the LADV controllers provide.

Document	Structured Text standard library reference
Code	LADV-IR-ST
Revision	1.0, 1 November 2026
Firmware	V1.0.0
Issued by	Wartens, makers of LADX

1. Functions

Functions take inputs by position or by name and return one value. Those marked with ... take two or more inputs.

Numeric

Function	Call	Result
ABS	ABS(IN)	Absolute value.
SQRT	SQRT(IN)	Square root of a REAL or LREAL.
LN	LN(IN)	Natural logarithm.
LOG	LOG(IN)	Logarithm to base 10.
EXP	EXP(IN)	e raised to IN.
SIN	SIN(IN)	Sine, IN in radians.
COS	COS(IN)	Cosine, IN in radians.
TAN	TAN(IN)	Tangent, IN in radians.
ASIN	ASIN(IN)	Arc sine, in radians.
ACOS	ACOS(IN)	Arc cosine, in radians.
ATAN	ATAN(IN)	Arc tangent, in radians.
ATAN2	ATAN2(Y, X)	Angle of the point (X, Y) from the X axis, in radians.
EXPT	EXPT(IN1, IN2)	IN1 raised to the power IN2.

Arithmetic

Function	Call	Result
ADD	ADD(IN1, IN2, ...)	Sum. Same as +.
MUL	MUL(IN1, IN2, ...)	Product. Same as *.
SUB	SUB(IN1, IN2)	Difference. Same as -.
DIV	DIV(IN1, IN2)	Quotient. Same as /. Integer division truncates toward zero; by zero faults with F310.

Function	Call	Result
MOD	MOD(IN1, IN2)	Remainder, with the sign of IN1. By zero faults with F310.
MOVE	MOVE(IN)	Returns IN. Same as assignment.

Bit shift

Function	Call	Result
SHL	SHL(IN, N)	Shift left N bits at the width of IN, zeros in.
SHR	SHR(IN, N)	Shift right N bits at the width of IN, zeros in.
ROL	ROL(IN, N)	Rotate left N bits at the width of IN.
ROR	ROR(IN, N)	Rotate right N bits at the width of IN.

Selection

Function	Call	Result
SEL	SEL(G, IN0, IN1)	IN1 when G is TRUE, otherwise IN0.
MAX	MAX(IN1, IN2, ...)	The largest input.
MIN	MIN(IN1, IN2, ...)	The smallest input.
LIMIT	LIMIT(MN, IN, MX)	IN held between MN and MX.
MUX	MUX(K, IN0, IN1, ...)	Input number K, counting from 0. K outside the inputs faults with F312.

Comparison

Function	Call	Result
GT	GT(IN1, IN2)	IN1 > IN2.
GE	GE(IN1, IN2)	IN1 >= IN2.
EQ	EQ(IN1, IN2)	IN1 = IN2.
LE	LE(IN1, IN2)	IN1 <= IN2.
LT	LT(IN1, IN2)	IN1 < IN2.
NE	NE(IN1, IN2)	IN1 <> IN2.

String

Function	Call	Result
LEN	LEN(IN)	Number of characters.
LEFT	LEFT(IN, L)	The first L characters.
RIGHT	RIGHT(IN, L)	The last L characters.
MID	MID(IN, L, P)	L characters starting at position P, counting from 1.
CONCAT	CONCAT(IN1, IN2, ...)	The inputs joined end to end.
INSERT	INSERT(IN1, IN2, P)	IN2 inserted into IN1 after position P.
DELETE	DELETE(IN, L, P)	IN with L characters removed from position P.
REPLACE	REPLACE(IN1, IN2, L, P)	L characters of IN1 from position P replaced by IN2.
FIND	FIND(IN1, IN2)	Position of the first IN2 in IN1, or 0 when it is not there.

Conversion

Function	Call	Result
TRUNC	TRUNC(IN)	REAL to integer, toward zero.
BCD_TO_INT	BCD_TO_INT(IN)	Binary-coded decimal to integer. A digit above 9 faults with F313.
INT_TO_BCD	INT_TO_BCD(IN)	Integer to binary-coded decimal.

Time and date

Function	Call	Result
ADD_TIME	ADD_TIME(IN1, IN2)	TIME + TIME.
SUB_TIME	SUB_TIME(IN1, IN2)	TIME - TIME.
MUL_TIME	MUL_TIME(IN1, IN2)	TIME multiplied by a number.
DIV_TIME	DIV_TIME(IN1, IN2)	TIME divided by a number.
ADD_TOD_TIME	ADD_TOD_TIME(IN1, IN2)	Time of day + TIME.
ADD_DT_TIME	ADD_DT_TIME(IN1, IN2)	Date and time + TIME.
SUB_TOD_TIME	SUB_TOD_TIME(IN1, IN2)	Time of day - TIME.
SUB_DT_TIME	SUB_DT_TIME(IN1, IN2)	Date and time - TIME.
SUB_DATE_DATE	SUB_DATE_DATE(IN1, IN2)	DATE - DATE, as TIME.
SUB_TOD_TOD	SUB_TOD_TOD(IN1, IN2)	Time of day - time of day, as TIME.
SUB_DT_DT	SUB_DT_DT(IN1, IN2)	Date and time - date and time, as TIME.
CONCAT_DATE_TOD	CONCAT_DATE_TOD(IN1, IN2)	A DATE and a time of day as one DT.

Type conversion

Any elementary type converts to another with X_TO_Y, for example INT_TO_REAL, REAL_TO_INT, DINT_TO_INT, BOOL_TO_INT, INT_TO_STRING, STRING_TO_INT, TIME_TO_DINT. TO_Y converts from whatever type the argument has. REAL to integer rounds half to even; TRUNC and REAL_TRUNC_INT truncate. An integer conversion that does not fit wraps at the width of the target, as hardware does.

2. Function blocks

Declare an instance, call it, and read its outputs as instance.name.

Block	Inputs	Outputs	Behaviour
TON	IN : BOOL, PT : TIME	Q : BOOL, ET : TIME	On delay. Q goes TRUE when IN has been TRUE for PT. ET counts up to PT. IN FALSE resets both.
TOF	IN : BOOL, PT : TIME	Q : BOOL, ET : TIME	Off delay. Q follows IN up at once, and stays TRUE for PT after IN goes FALSE.
TP	IN : BOOL, PT : TIME	Q : BOOL, ET : TIME	Pulse. A rising edge on IN gives Q TRUE for exactly PT; IN cannot cut it short or extend it.
LTON	IN : BOOL, PT : LTIME	Q : BOOL, ET : LTIME	TON with LTIME.
LTOF	IN : BOOL, PT : LTIME	Q : BOOL, ET : LTIME	TOF with LTIME.
LTP	IN : BOOL, PT : LTIME	Q : BOOL, ET : LTIME	TP with LTIME.
CTU	CU : BOOL, R : BOOL, PV : INT	Q : BOOL, CV : INT	Count up. Each rising edge on CU adds one to CV. Q is CV >= PV. R sets CV to 0.

CTD	CD : BOOL, LD : BOOL, PV : INT	Q : BOOL, CV : INT	Count down. Each rising edge on CD takes one from CV. Q is CV <= 0. LD loads PV into CV.
CTUD	CU : BOOL, CD : BOOL, R : BOOL, LD : BOOL, PV : INT	QU : BOOL, QD : BOOL, CV : INT	Count up and down. QU is CV >= PV, QD is CV <= 0. R resets, LD loads PV.
CTU_DINT	CU : BOOL, R : BOOL, PV : DINT	Q : BOOL, CV : DINT	CTU with DINT counts.
CTD_DINT	CD : BOOL, LD : BOOL, PV : DINT	Q : BOOL, CV : DINT	CTD with DINT counts.
CTUD_DINT	CU : BOOL, CD : BOOL, R : BOOL, LD : BOOL, PV : DINT	QU : BOOL, QD : BOOL, CV : DINT	CTUD with DINT counts.
CTU_LINT	CU : BOOL, R : BOOL, PV : LINT	Q : BOOL, CV : LINT	CTU with LINT counts.
CTD_LINT	CD : BOOL, LD : BOOL, PV : LINT	Q : BOOL, CV : LINT	CTD with LINT counts.
CTUD_LINT	CU : BOOL, CD : BOOL, R : BOOL, LD : BOOL, PV : LINT	QU : BOOL, QD : BOOL, CV : LINT	CTUD with LINT counts.
CTU_UDINT	CU : BOOL, R : BOOL, PV : UDINT	Q : BOOL, CV : UDINT	CTU with UDINT counts.
CTD_UDINT	CD : BOOL, LD : BOOL, PV : UDINT	Q : BOOL, CV : UDINT	CTD with UDINT counts.
CTUD_UDINT	CU : BOOL, CD : BOOL, R : BOOL, LD : BOOL, PV : UDINT	QU : BOOL, QD : BOOL, CV : UDINT	CTUD with UDINT counts.
CTU_ULINT	CU : BOOL, R : BOOL, PV : ULINT	Q : BOOL, CV : ULINT	CTU with ULINT counts.
CTD_ULINT	CD : BOOL, LD : BOOL, PV : ULINT	Q : BOOL, CV : ULINT	CTD with ULINT counts.
CTUD_ULINT	CU : BOOL, CD : BOOL, R : BOOL, LD : BOOL, PV : ULINT	QU : BOOL, QD : BOOL, CV : ULINT	CTUD with ULINT counts.
R_TRIG	CLK : BOOL	Q : BOOL	Rising edge. Q is TRUE for one call when CLK goes from FALSE to TRUE.
F_TRIG	CLK : BOOL	Q : BOOL	Falling edge. Q is TRUE for one call when CLK goes from TRUE to FALSE. Not on the first call.
SR	S1 : BOOL, R : BOOL	Q1 : BOOL	Set dominant bistable. Q1 is TRUE after S1, FALSE after R, and TRUE when both are.
RS	S : BOOL, R1 : BOOL	Q1 : BOOL	Reset dominant bistable. Q1 is TRUE after S, FALSE after R1, and FALSE when both are.

Timers measure time between calls, so a timer called every 10 ms has 10 ms resolution. Counters count edges, not scans: a CU held TRUE counts once.