

Modicon TSX Nano PLCs

PL7 language

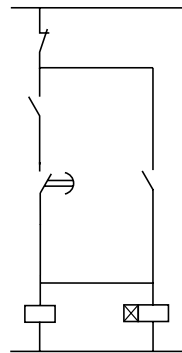
General

Caractéristiques :
page 40052/3
Références :
page 40053/3

PL7 language on Nano PLCs enables the programming of simple sequential applications such as those requiring numerical processing or specific functions such as schedule blocks, fast counting, etc. This programming is in List language (Instruction List) or in Ladder language.

These two languages are reversible provided a few simple programming rules are respected : any Nano PLC program which has been written in Instruction List (on an FTX 117 terminal or using PL7-07 software) can be read and modified in Ladder language (with PL7-07 software on an FT 2000/FTX 517 terminal or PC compatible) or vice versa.

List language



```
000 LD      %I0.0
001 AND (   %I0.1
002 ANDN   %TMO.Q
003 OR     %Q0.1
004 )
005 ST     %Q.1
006 IN    %TMO
007 ---
```

PL7 List language comprises a list of instructions from different families for direct translation into :

- Instructions on Ladder diagram bits, logic diagrams or Boolean equations
- Instructions on control system function blocks (timers, counters, etc)
- Grafcet instructions
- Instructions on words for numerical processing
- Instructions on the program for structuring programs

Ladder language

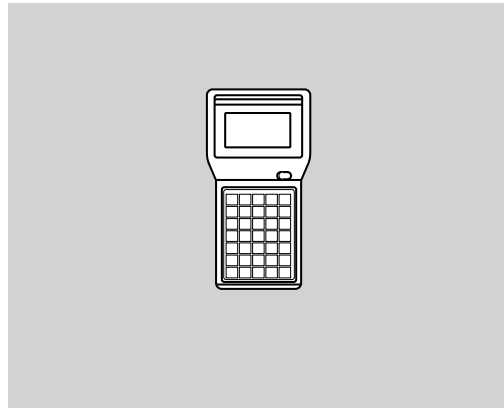
PL7 Ladder language is entirely graphic and thus offers the advantage of similarity with electromagnetic relay control systems. Its basic symbols are complemented by graphic elements allowing it to carry out control system functions, numerical processing and structuring of Nano PLC programs.

Ladder language provides additional assistance when debugging applications through the real-time display of graphic symbols (for example, the highlighting of closed contacts).

Programming terminals

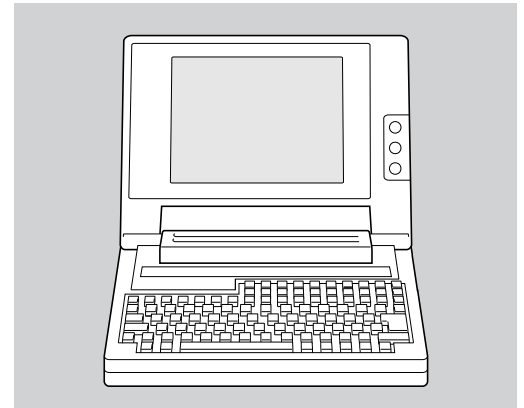
The development, transfer, debugging and archiving of programs for Nano PLCs can be carried out equally well on either of the two types of terminal :

FTX 117



Dedicated pocket terminal, for programming in List language with operation in offline or online mode.

FT2000/FTX 517 or PC compatible



Standardised design office and workshop terminal, with PL7-07 software for programming in Ladder and/or List language (Instruction List).

Modicon TSX Nano PLCs

PL7 language

Characteristics

References :
page 40053/3

Instructions	Combined List instructions	Grafcet List instructions #	
	<ul style="list-style-type: none"> ● LD, LDN, LDR, LDF : read the state of a bit (direct, inverse, rising and falling edge) ● ST, STN, S, R : write an output (direct, inverse, set, reset) ● AND, ANDN, ANDR, ANDF : logic AND with a bit (direct, inverse, rising and falling edge) ● OR, ORN, ORR, ORF : logic OR with a bit (direct, inverse, rising and falling edge) ● LD (, AND (, OR(,) : open and close brackets (8 possible levels) ● XOR, XORN, XORR, XORF : exclusive OR with a bit ● MPS, MRD, MPP : buffer memory management for divergence towards output bits ● N : negation 	<ul style="list-style-type: none"> ● -*i : step ($1 \leq i \leq 62$) ● =*i : initial step ($1 \leq i \leq 62$) ● #i : activate step i, after deactivation of current step ● # : deactivate current step ● #Di : deactivate step i after another step ● =*POST : start post-processing ● %Xi : bit associated with step i 	
	<p>List comments and title with PL7-07 software</p> <ul style="list-style-type: none"> ● Title : 122 characters before each instruction LD, LDN, LDR, LDF ● Comments : 4 lines of 122 characters before each instruction LD, LDN, LDR, LDF ● Possibility of associating a comment of 122 characters with each instruction 	<p>Instructions on program</p> <ul style="list-style-type: none"> ● MCS, MCR : master relay ● END, ENDC, ENDCN : end of program (conditional or unconditional) ● JMP, JMPC, JMPCN : jump to a label %L (conditional or unconditional) ● SRn : call subroutine n ($0 \leq n \leq 15$) ● RET : end of subroutine ● NOP : non-operative instruction 	
	<p>Ladder rungs</p> <ul style="list-style-type: none"> ● 10 contacts of 7 lines with 1 output per line ● Title : 122 characters per rung ● Comments : 4 lines of 122 characters 	<p>Ladder language graphic symbols</p> <ul style="list-style-type: none"> ● Normally open, normally closed and on edge contacts ● Direct, inverse, SET and RESET coils ● Program jump, subroutine call 	
	<p>Standard function blocks</p> <ul style="list-style-type: none"> ● 32 timers : %Tmi ($0 \leq i \leq 31$) 0 to 9999 (word) ● 16 up/down counters : %Ci ($0 \leq i \leq 15$) 0 to 9999 (word) ● 4 16-bit LIFO or FIFO registers : %Ri ($0 \leq i \leq 3$) ● 4 drum controllers : %DRi ($0 \leq i \leq 3$) 8 steps ● Real-time clock : %RTCi ($0 \leq i \leq 15$) month, day, hour, minute, with TSX Nano 16 and 24 I/O <p>Numerical instructions</p> <ul style="list-style-type: none"> ● Assignment in word, indexed word, bit strings word tables : := ● Arithmetic : +, -, x, /, REM, SQRT ● Logic : AND, OR, XOR, NOT, INC, DEC ● Shift operation : SHL, SHR, ROL, ROR (logic and rotate) ● Conversion : BTI, ITB (BCD <-> Binary) ● Comparison : >, <, <=, >=, =, <> 	<p>Specific function blocks</p> <ul style="list-style-type: none"> ● Transmission/reception of message of 64 words maximum (internal or constant) : EXCH ● Exchange control : %MSG available output, fault output ● 8 shift bit registers : %SBRI ($0 \leq i \leq 7$), shift one step to the left or right (max. 16 steps). ● 8 step counter blocks : %SCi ($0 \leq i \leq 7$), move forward or back one step (max. 256 steps) ● 1 fast counter (max.10 KHz), frequency meter (max. 10 KHz), up/down counter (max. 1 KHz) : %FC with 2 high speed outputs ● Pulse width modulated output : %PWM ● Pulse output : %PLS 	
	<p>Specific functions</p> <ul style="list-style-type: none"> ● 1 input for PLC RUN/STOP command ● 1 PLC status (security) output : PLC error ● 6 latching inputs : 100 μs minimum 	<ul style="list-style-type: none"> ● Real-time display of Grafcet steps used ● Symbol table management ● Porting of TSX Nano applications to TSX Micro (List or Ladder) 	
	<p>Addressable objects</p>	<p>Bit objects</p> <ul style="list-style-type: none"> ● %I/Qx.y : 28 inputs and 20 outputs max. ● %Mi : 128 internal bits ● %Si : 128 system bits ● %Xi : 62 Grafcet steps ● %●●i.j : function block bits ● %●●i:Xk : bits extracted from internal words, system words, constant words, input and output words <p>Bit string and word table objects</p> <ul style="list-style-type: none"> ● %●i:L : bit strings (I/O, internal, system and Grafcet bits) 	<p>Word objects</p> <ul style="list-style-type: none"> ● %MWi : 256 internal words ● %KWi : 64 constant words ● %SWi : 128 system words ● %IWi.j : 2 input words per PLC (exchange words for inter-PLC communication) ● %QWi.j : 2 output words per PLC (exchange words for inter-PLC communication) <ul style="list-style-type: none"> ● %●Wi:L : word tables (internal, constant and system words)

Modicon TSX Nano PLCs

PL7 language

Functions

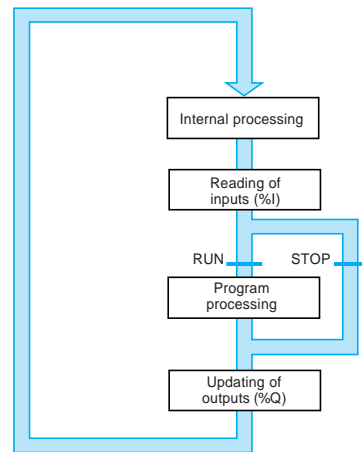
Characteristics :
page 40052/3
References :
page 40053/3

Software structure

There are two types of scan execution :

- Normal cyclic execution. This is the default setting.
- Periodic execution. This type of execution and the period of time are defined by the user during configuration.

Normal (cyclic) execution

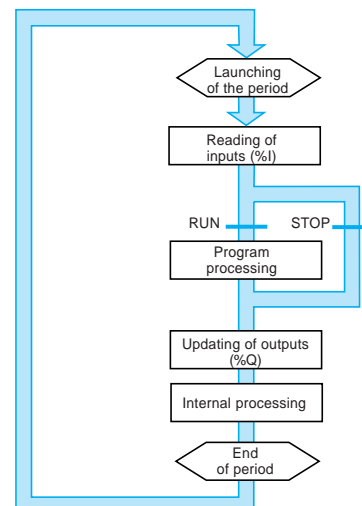


At the end of each scan the PLC system relaunches execution of a new scan. The execution time of each scan, which must not exceed 150 ms, is monitored by a software watchdog.

If this value is exceeded, a fault appears causing :

- Immediate stop of the scan (STOP)
- Display on the PLC front panel (RUN light flashing)
- Memorisation in a system bit (%S11)
- If an output is configured for the SECURITY function, it is reset to 0

Periodic execution



The execution of a scan is relaunched at the end of each period. The scan execution time must be less than that of the period defined (2 to 150 ms). If it exceeds this, it is memorised in a system bit (%S19) which should be tested and reset to 0 by the user (via the program or the terminal).

A software watchdog of 150 ms monitors the scan time. If it exceeds 150 ms, an execution fault is displayed (see normal execution).

PLC scan

In both types of execution, the system carries out :

- **Internal processing**
The system implicitly :
 - monitors and controls the PLC
 - processes requests from the terminal
- **Reading of inputs**
The state of each preactuator connected to the inputs (%I) is memorised. It is this memorised state which is taken into account during program processing.
- **Program processing**
The program is executed in the order in which the user has written it (except for program or subroutine jump instructions).
- **Updating of outputs**
The outputs (%Q) are activated or deactivated depending on the state (0 or 1) defined by the program.

Modicon TSX Nano PLCs

PL7 language

Functions

Characteristics :
page 40052/3
References :
page 40053/3

Instruction List language

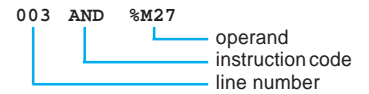
Program structure

A program in PL7 language comprises a list of instructions (up to 1000 instructions) from the following different families :

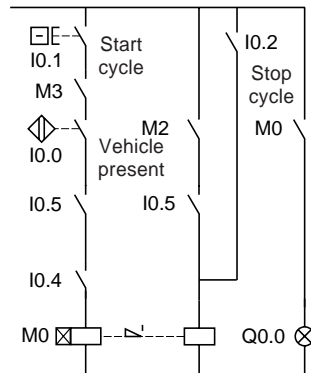
- Bit instructions : for example, read input n° 3 : LD %I0.3
- Function block instructions : for example, start timer n° 0 : IN %TM0
- Word instructions : for example, an addition [%MW10 := %MW50 + 100]
- Program instructions : for example, call subroutine n° 5 : SR5
- Grafcet instructions : for example, step n° 8 : -* - 8

Each program line has an automatically generated line number, an instruction code and a bit or word operand.

Example of a program line :



Simple application programming (Boolean processing)

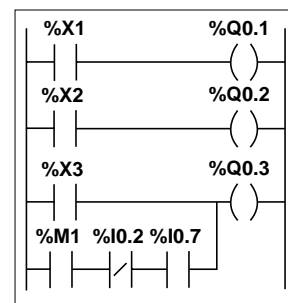
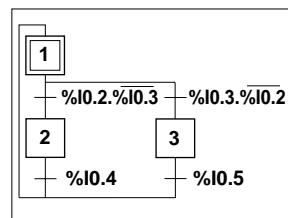
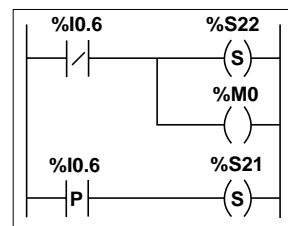


The translation of a Ladder diagram into an Instruction List program is immediate.

```

000  LD    %I0.1  Start cycle pushbutton
      AND  %I0.0  Vehicle present proximity sensor
      AND  %M3    Real-time clock authorisation bit
      AND  %I0.5  High roller limit switch
      AND  %I0.4  Rear gantry limit switch
005  S    %M0    Memo start cycle
      LD    %M2
      AND  %I0.5
      OR   %I0.2  Stop cycle pushbutton
      R    %M0
010  LD    %M0
      ST   %Q0.0  Scan indicator
    
```

Application programming with Grafcet



A Grafcet program is divided into 3 parts, each with a specific role.

```

000  LDN   %I0.6  Pre-processing
001  S     %S22  This is made up of a list of instructions
002  ST    %M0  for processing :
003  LDR   %I0.6  ● Power returns
004  S     %S21  ● Failures
                          ● Changes in mode
                          ● Input logic
It ends with the first =* or -* instruction encountered.
    
```

```

005  =* =  1  Sequential processing
006  LD    %I0.2  This is made up of the chart (instructions
007  ANDN  %I0.3  representing the chart) :
008  #     2  ● Steps
009  LD    %I0.3  ● Transitions
010  ANDN  %I0.2  ● Conditions
011  #     3  It ends with execution of the =* POS
012  -* -  2  instruction.
013  LD    %I0.4
014  #     1
015  -* -  03
016  LD    %I0.5
017  #     1
    
```

```

018  =* =  POST Post-processing
019  LD    %X1  This is made up of a list of instructions
020  ST    %Q0.1 for processing :
021  LD    %X2  ● Instructions from the sequential
022  ST    %Q0.2 processing part to control the outputs
023  LD    %X3  ● Safety interlocks specific to the
024  OR(   %M1  outputs
025  ANDN  %I0.2
026  AND   %I0.7
027  )
028  ST    %Q0.3
    
```

Modicon TSX Nano PLCs

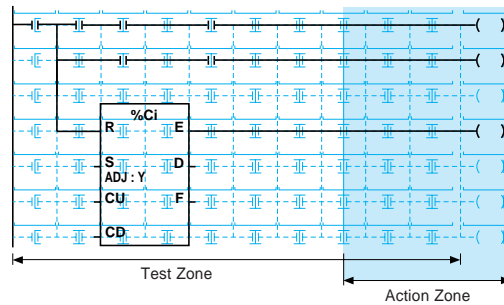
PL7 language

Functions (cont.)

Characteristics :
page 40052/3
References :
page 40053/3

Ladder language

Program structure



A program in Ladder language consists of a series of rungs. Each rung is labelled and can be :

- Described by a title of 122 characters maximum.
- Completed by a comment of 4 lines of 122 characters maximum

A rung consists of 7 lines of 11 columns with a maximum of 10 contacts and one coil per line.

The rung is divided into two different zones :

- Test Zone for receiving graphic elements; contacts, comparison blocks and function blocks (standard or specific).
- Action Zone for receiving coils (in column 11) and operation blocks (from column 8 onwards).

Within a rung, coils or operation blocks must be connected by at least one vertical link in order to form a single group.

Graphic elements

The graphic elements which make up a rung are :

• Contacts



These test the state of the bit associated with them.

4 types are available : normally open, normally closed, rising edge (P) and falling edge (N).

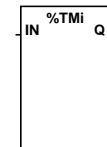
• Coils



These control the output bits or internal bits.

4 types are available : direct, inverse, set and reset.

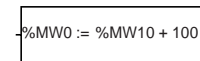
• Standard and specific function blocks



These correspond to the control system functions.

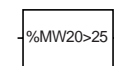
There are 10 of them (see next page). A single function block is authorised for each rung.

• Operation blocks



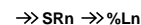
These enable numerical processing : assignment of words, arithmetic, logic, conversion, logic and rotate shift operations, incrementation/decrementation. They call up the List language numerical instructions.

• Comparison blocks



These enable comparison of two words of any type (>, >=, <, <=, =, <>).

• Program structure elements



These call up subroutine n and the program jump for rung n.

• Connecting elements



These elements, called horizontal Boolean logic and vertical Boolean logic are used to connect all the graphic elements described above.

Reversibility

The reversibility of List and Ladder languages enables the display of programs in whichever language is desired, regardless of the language used in their creation. For example, an application developed in the design office in Ladder language can be read, and even modified, in List language, and vice versa.

In order to be reversible, an application written in List language must respect a few rules of reversibility :

- Certain instructions such as XOR, JMPCN, etc must not be used.
- Function blocks such as BLK, OUT_BLK and END_BLK, etc must be used.

Each part of a non reversible program is represented in List language, the rest of the reversible program is presented in the form of rungs.

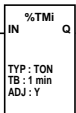
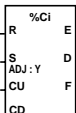
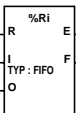
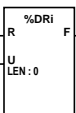
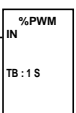
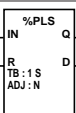
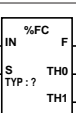
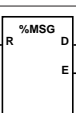
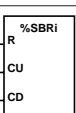
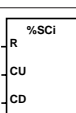
Modicon TSX Nano PLCs

PL7 language Standard function blocks

Characteristics :
page 40052/3
References :
page 40053/3

Functions

Function blocks are pre-programmed in the Nano PLC and allow control system functions to be integrated easily into application programs.

Description	Number	Chart	Function	Function	Function	
Standard function blocks						
Timer 1 ms minimum 9999 min maximum	32		E TYP TB	Enable input TON on-delay timer TOF off-delay timer TP Monostable Time base : 1 ms (TMO & TM1), 10 ms/100 ms/1 s or 1 min	C %Ti,P %Ti,V ADJ	Timer output Preset value word 0 to 9999 Current value word Adjustment permitted (Y) or prohibited (N)
Up/down counter	16		R P CU CD	Reset input Preset input Increment on edge input Decrement on edge input	E Ci,D F %Ci,P %Ci,V ADJ	Overflow output bit (0 to 9999) Preset done output bit Overflow output bit (9999 to 0) Preset value word 0 to 9999 Current value word Adjustment permitted (Y) or prohibited (N)
LIFO/FIFO register	4		R I O TYP	Reset input Storage on edge input Retrieval on edge input FIFO, stack LIFO, stack	%Ri,I %Ri,O Ri,E Ri,F	Register access word Register output word Register empty output bit Register full output bit
Drum controller	4		R U LEN	Return to step zero Forward step input Number of steps	%DRi.S F Command bits	Number of current step Last bit not currently defined 16 %Qi or %Mi bits
Specific function blocks						
Width modulated output	1		IN TB	Pulse input Time base 0.1 ms, 10 ms, 1 s	%PWM.P %PWM.R %Q0.0	Period preset ≤ 32767 Period ratio 0 to 100% Width modulated output
Pulse output	1		IN R TB ADJ N	Pulse input Reset number of pulses to 0 input Time base 0.1 ms, 10 ms, 1 s	%PLS.P %PLS.N Q D %Q0.0 ADJ	Period preset ≤ 32767 Pulse number ≤ 32767 Current pulse output bit Done pulse output bit Pulse output Adjustment permitted (Y) or prohibited (N)
Fast up/down counter Frequency meter	1		IN S TYP TH0 TH1	Enable input Preset input Threshold value S0 ≤ 65535 Threshold value S1 ≤ 65535	%FC.P %FC.V F %Q0.1 %Q0.2 TH0 TH1	Up/down preset value ≤ 65535 Current value Overflow output bit High-speed output 0 High-speed output 1 Current output bit value ≥ threshold TH0 Current output bit value ≥ threshold TH1
Message transmission/reception	–	EXCH	EXCHANGE	Transmission or reception (1) via (Uni-Telway or ASCII) terminal port or Modbus link	%MWi:L %KWl:L	Internal word table L ≤ 64 Constant word table L ≤ 64
Exchange control	–		R	Communication initialisation input	E D	Communication error output bit Available link output bit
Bit shift register	8		R CU CD	Reset 16 %SBRi.j bits to 0 Shift input left Shift input right	%SBRi.j	Bits 0 to 15 of register %SBRi
Step counter	8		R CU CD	Reset %SCi.j bits to 0 Increment input one step Decrement input one step	%SCi.j	Bits 0 to 255 of step counter %SCi
Schedule block (real-time clock)	16	RTC:i	Q : MTWTFSS hh:mm	Assignment of output %Mi or %Qj.k activated by schedule block Activation days of the week Hours (0 to 23) and minutes (0 to 59) of start and end of activation	DD-MMM	Validation start and end date DD : day 1 to 31 MMM : month Jan.-Dec.

(1) This function is specific to PL7-07 ≥ V3, compatible with Nano PLCs ≥ version 2.