

Език за програмиране PL7-2

1.1 Представяне

Софтуерът PL7-2 обхваща два графични програмни езика за програмиране на програмируеми контролери TSX 17-20, TSX 27, TSX 47-J, TSX 47-10 и TSX 47-20.

Предимства на графичните езици

- Лесно програмиране:
 - позволяват лесно провеждане на анализи;
 - голяма близост до процеса на проектиране;
- Лесна настройка на програмите:
 - визуализация на контактните мрежи и стъпките на Grafset в реално време;
- Възможност за създаване на документация, която е много точно отражение на проекта и управляващата програма.

Език Ladder

Програмите, написани на езика Ladder, се състоят от поредица от контактни мрежи.

Основните графични елементи са контакти (входове) и изходни елементи (бобини), допълнени с множество функционални блокове и операторни блокове.

Език Grafset

Езикът Grafset се състои от:

- Автоматен граф (Grafset), включващ стъпки, преходите между тях и директни връзки, които дефинират основната структура на управляващата програма;
- Контактни мрежи, написани на езика Ladder, които дефинират действията свързани със стъпките и условията, свързани с преходите между тях.

Приложни средства

Структурата на езика PL7-2 позволява интегрирането в потребителската програма на диагностични средства, както по отношение на състоянието на хардуера на контролера, така и по отношение на самата приложна програма.

Език за програмиране PL7-2

1.2 Адресация на входовете и изходите на TSX 17-20

Входно/изходната адресация на контролера TSX 17-20 е базирана на геометричен принцип:

Дискретни вх./изх.:

%Ix,i = вход
%Qx,i = изход

x = номер на входно изходния модул (0 - 3)

l = номер на входно/изходната точка:

входове - от 0 до 23;

изходи - от 0 до 15.

Максимални конфигурации:

Базов процесорен модул	Дискретни вх./изх. разширения	Интелигентни разширения
1	3	0
1	2	1
1	1	2
1	0	3

Входове-тригери:

%I0,24 и %I0,25 при съответно конфигуриране.

Правила за определяне на номера x на входно/изходния модул:

- Базовият процесорен модул винаги има номер 0;
- Първото входно/изходно разширение винаги има номер 1;
- Второто входно/изходно разширение винаги има номер 2;
- Третото входно/изходно разширение винаги има номер 3;

Забележка: Входно/изходните модули могат да бъдат свързвани в какъвто и да е ред.

Пример на връзки между входно/изходни модули:



Примери за вх./изходна адресация:

- %I0,5 = вход 5 на модул 0;
- %I2,3 = вход 3 на модул 2;
- %Q1,5 = изход 5 на модул 1.

Език за програмиране PL7-2

1.2 Адресация на входовете и изходите на TSX 17-20 (продължение)

Типове конфигурации на контролера TSX 17-20

Тип модул	Брой вх./изх.	Входове	Изходи	Входовете-тригери
Базов модул	20 (20 вх., 8 изх.)	от %I0,00 до %I0,11	от %Q0,00 до %Q0,07	%I0,00 %I0,25
	34 (22 вх., 12 изх.)	от %I0,00 до %I0,21	от %Q0,00 до %Q0,11	%I0,24 %I0,25
	40 (24 вх., 16 изх.)	от %I0,00 до %I0,23	от %Q0,00 до %Q0,15	%I0,24 %I0,25
Разширителен блок	34 (22 вх., 12 изх.)	от %Ix,00 до %Ix,21	от %Qx,00 до %Qx,11	- -
	40 (24 вх., 16 изх.)	от %Ix,00 до %Ix,23	от %Qx,00 до %Qx,15	- -
Разширителен модул	8 (8 входа)	от %Ix,00 до %Ix,07	- -	- -
	6 (6 изхода)	- -	от %Qx,00 до %Qx,05	- -

Вход %I0,00 и изход %Q0,00

Този вход и този изход могат да се използват по два различни начина:

Вх./изх. точка	Използуване по подразбиране	Използуване след конфигуриране
%I0,00	нормален вход	вход за управление на СТАРТ/СТОП на контролера
%Q0,00	нормален изход	изход, даващ възможност за използване във външни вериги за безопасност

Адресиране на вх./изх. регистри (думи) на интелигентните модули

%IWx,i = входна дума

%Qwx,i = изходна дума

Език за програмиране PL7-2

x = номер на модула (от 1 до 3)

i = номер на думата (от 1 до 7)

Забележка: Интелигентните модули също така съдържат и вх./изх. битове,

които се адресират по същия начин както при дискретните вх./изх. модули.

1.4 Адресируеми обекти в езика PL7-2

Адресируемите обекти в езика са: битове и 16-битови думи.

Списък на обектите битове:

Тип	Адрес	Максимален брой	Разрешен ли е запис в тях ?	Раздел
Входен бит	%Ix,i	96	не	1.2
Изходен бит	%Qx,i	64	да	1.2
Входен бит-тригер	%I0,24	1	не	1.2
	%I0,25	1	не	1.2
Вътрешен бит	%Mi	256	да	-
Системен бит	%SYi	24	да/не	1.4
Бит на функционален блок	%Mi,R (пример)	В зависимост от функционалния блок	не	2
Стъпка на Grafset	%Xi	96	да	6.3
Бит за грешки	%Si,j	4x8	не	7.5

Списък на обектите думи:

Тип	Адрес	Максимален бой	Разрешен ли е запис в тях ?	Раздел
Вътрешна дума	%MWi*	1024	да	-
Константна дума	%KW*	1024	не	-
Системна дума	%SWi	64	да/не	1.4
Com-дума на FIPWAY	COMi,j	16x4	ако I = номера на	-

Език за програмиране PL7-2

			станцията	
Дума на функционален блок	%M _{Ni,P} (пример)	В зависимост от функционалния блок	да/не	2
Активно време на стъпка на Grafscett	%X _{i,V}	96	не	6.3
Входна дума	%IW _{x,i}	4x8	не	1.2
Изходна дума	%QW _{x,i}	4x8	да	1.2

(*) TSX 17-20 позволява директно адресиране само на първите 128 думи.

Останалите думи до 1024 са достъпни с помощта на функцията за трансфер на таблици.

1.5 Графични елементи

Елементи за тест и активиране на битове:

Тип	Графичен символ	Обекти, свързани с него
Контакти (входове)		Всички битови обекти и битове на думи
Бобини (изходи)		Всички битови обекти, разрешени за запис
Изходи с памет *		%Q _{x,i} , и %M _i




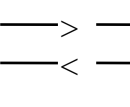



(*) Тези изходи запазват състоянието си в 1-вия цикъл след топъл рестарт.

Разклонителни елементи:

Тип	Графичен символ	Обекти, свързани с него
Изходи за преход		Номер на контактна мрежа
Хоризонтални и вертикални връзки		- -

Език за програмиране PL7-2

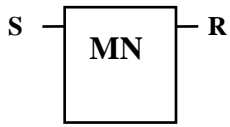
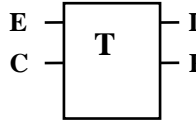
Специфични графични елементи в езика Grafset

Тип	Графичен символ	Максимален брой	Битове, свързани с него	Думи, свързани с него
Стъпка		96*	%Xi: бит стъпка	%Xi,V: време на стъпката
Начална стъпка		16	%Xi: бит стъпка	%Xi,V: време на стъпката
Преход		128	- -	- -
Начало и край на паралелни клонове		- - -	- - -	- - -
Начало и край на разклонения		- -	- -	- -
Отдалечена връзка		42	- -	- -
Директна връзка		- -	- -	- -
Преход от една стъпка към друга		128	- -	- -

(*) Включително и началните стъпки.

Графини елементи (продължение)

Функционални блокове:

Тип	Символ	Максимален брой	Характеристики
Моновибратор		8	Генерира програмируем импулс в диапазона 10 ms - 9999 min.
Таймер		32	Програмируем таймер в диапазона 10 ms - 9999 min.

Език за програмиране PL7-2



Брояч		31	Броене нагоре от 0 до 9999 и надолу от 9999 до 0.
Регистър		4	FIFO и LIFO структури от 1 до 16 думи
Стъпков избирач		8	Броене от 0 до 9999 или таймер в диапазона от 0.555 ms до 5.55 sec.
Бърз брояч/таймер		1	

Свързаните с блоковете обекти, зависят от типа на функционалния блок:

- битове за изходите
- думи за параметрите.

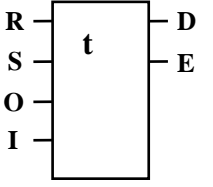
Графични елементи (продължение)

Операторни блокове: Неограничен брой. Операции над думи, индексирани думи, низове от битове или таблици от думи.

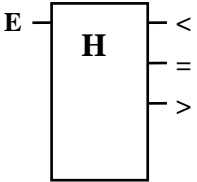
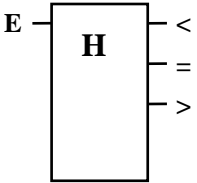
Операции	Символи	Типове
Сравнение		<>, <, >, >=, <=, =
Аритметични Логически Преобразувания Логическо изместване Трансфер		+, -, *, / AND, OR, XOR Двоичен код - BCD, Двоичен код - ASCII Циклично изместване наляво и надясно Дума - дума, низ от битове - дума или индексна дума, таблици от думи

Език за програмиране PL7-2

Текстови блокове: Комуникационни функционални блокове

Тип	Символ	Максимален брой	Характеристики
Пълен текстов блок		8	Обмен на таблици от думи с: <ul style="list-style-type: none"> • периферни устройства, свързани към паралелния порт; • друга станция; • друга UNI-TELWAY станция.

Час/дата блок (или "Н" блок): Неограничен брой

Тип	Символ	Характеристики
"Н" блок от седмичен тип		Използува се за установяване на деня от седмицата и времето(час и минути) във времезависими функции
"Н" блок от годишен тип		Използува се за установяване на датата (месец и ден) във времезависими функции

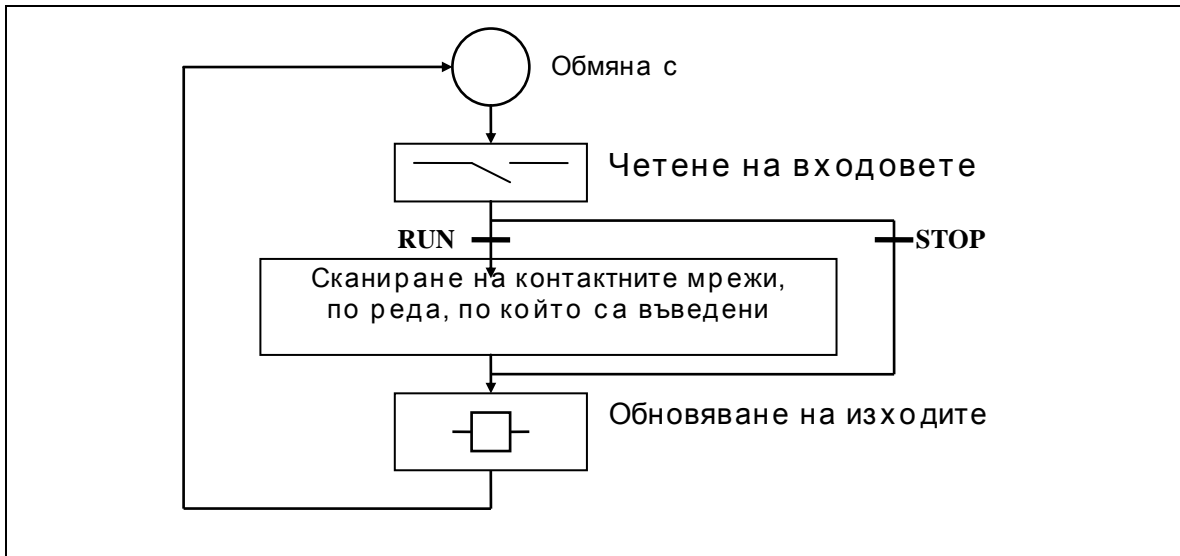
1.6 Еднозадачен цикъл на сканиране на TSX 17-20.

Еднозадачен цикъл: %SY19=1 (Изключена бърза задача)

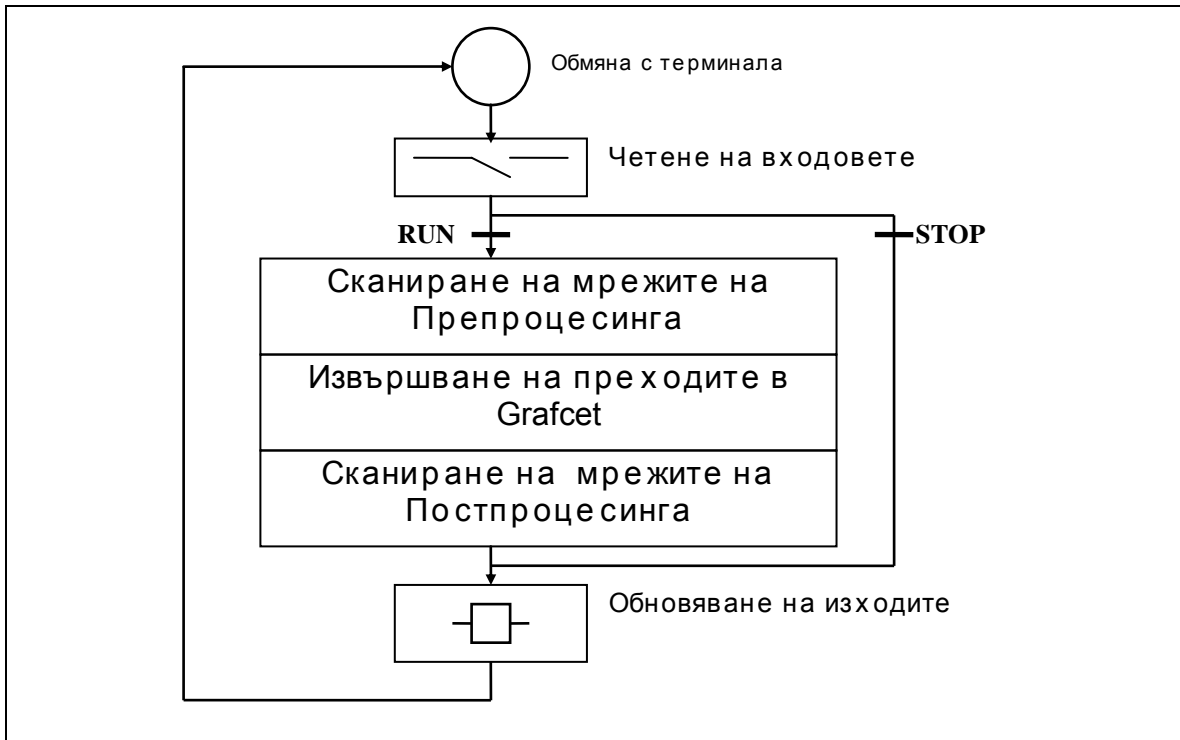
Време за изпълнение на един цикъл < 150ms (контролирано от "watchdog")

Език за програмиране PL7-2

Език Ladder



Език Grafcet



Забележка: В момента на спиране на сканирането:
- ако %SY8=1, изходите се установяват в 0;
- ако %SY8=0, изходите запазват състоянието си.

1.7 Двузадачен цикъл на сканиране на TSX 17-20

Двузадачен цикъл: %SY19=0 (Разрешене бърза задача)

Език за програмиране PL7-2

Изпълнение на задачите:

- Основна задача - цикъл < 150 ms (както при еднозадачния цикъл, вж. 1.6)
- Бърза задача - стартира се при възникване на събитие.

Предназначение на бързата задача: Да осигури бързо време на реакция на важни събития.

Принцип на действие: Бързата задача прекъсва изпълнението на основната задача в следните случаи:

- когато има физическа промяна на състоянието на някой от входовете тригери I0,24 и I0,25 (ако са конфигурирани като такива);
- когато бързият брояч FC достигне зададената си стойност.

Прекъсването на основната задача е възможно единствено след:

- приключване на сканирането на текущата контактна мрежа;
- приключване на действията в текущата стъпка или преход в Grafset.

Забележка: Ако постъпи нова заявка за изпълнение на бързата задача, преди още да е завършило предишното нейно изпълнение, то тя се изпълнява още веднъж, без да се предава управлението на основната задача.

Цикъл на изпълнение на бързата задача



Конфигуриране на входовете и изходите в бързата задача

Само един входен байт (%I0,00 - %I0,07; %I0,08 - %I0,15 или %I0,16 - %I0,23) и един изходен байт (%Q0,00 - %Q0,07 или %Q0,07 - %Q0,15) от базовия контролер, могат да бъдат конфигурирани в бързата задача.

Когато бързата задача е активна, тя обновява изходите, конфигурирани в нея.

Бързата задача не се активира ако състоянието на входовете тригери се промени от програмата или от програматора.

Език за програмиране PL7-2

2.1 Характеристики

Контактните мрежи в езика PL7-2

Контактните мрежи се състоят от:

- Етикет от 1 до 999 (задължителен);
- Коментар от максимум 15 символа (незадължителен);
- Мрежа от контакти и изходни бобини, свързани с хоризонтални и вертикални връзки.

Максимална конфигурация:

- 4 линии от 9 контакта в тестовата зона;
- 4 изходни бобини в зоната на действията.

Контактните мрежи също могат да съдържат:

В тестовата зона:

- функционални блокове;
- блокове за сравнение.

В зоната на действията:

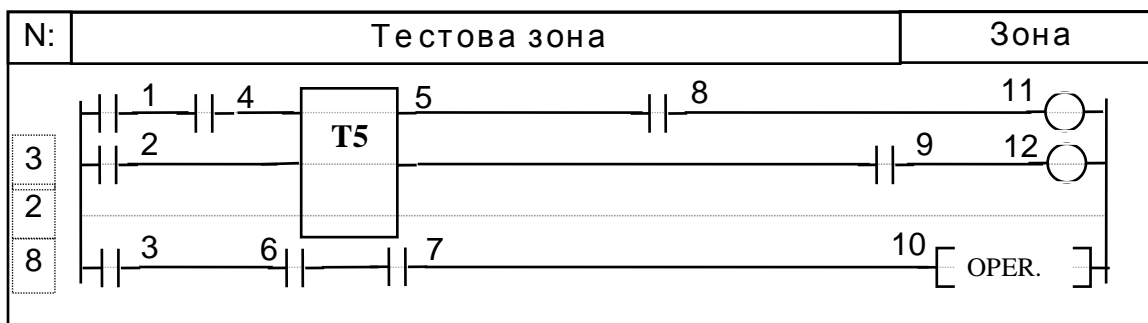
- операторни блокове.

Сканиране на контактните мрежи:

- Колона по колона, от ляво на дясно;
- Блоковете се изпълняват веднага щом сканирането достигне до левия горен ъгъл на блока.

Пример:

Числата, нанесени в контактната мрежа, показват реда, по който елементите ще бъдат сканирани и изпълнени



Съставяне на програмата:

Програмата на езика Ladder се състои от последователност от контактни мрежи. Те се сканират по реда, по който са въведени (а не по реда на техните етикети).

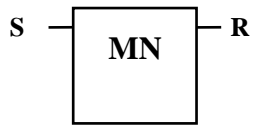
Език за програмиране PL7-2

2.2 Моновибратор

Характеристики

Брой: 8 (%MN0 - %MN7)

Период на импулса: мин. 10ms
макс: 9999ms

Входове	Графичен символ	Изходи	Асоциирани битове
S: Старт (импулс)		R: Работа	%MN _i ,R
2 колони, 3 линии			

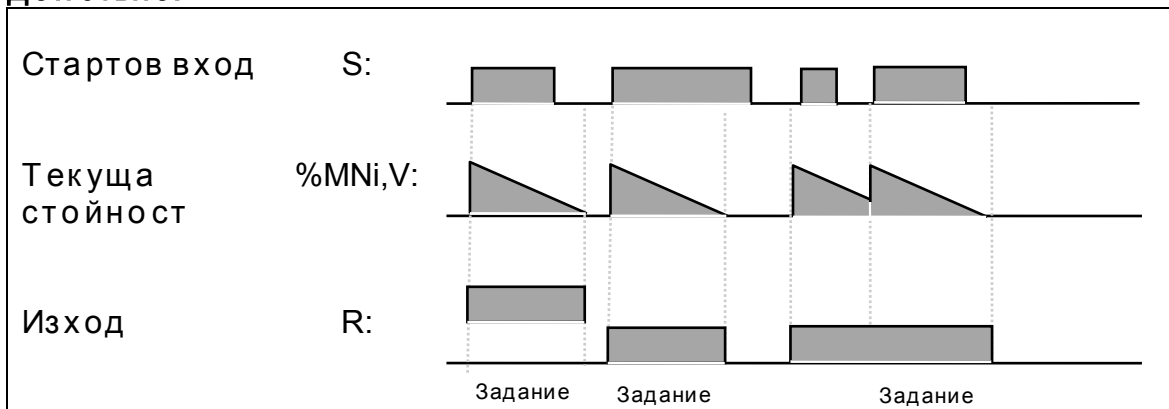
Параметри, свързани с блока:

Параметър	Дума	Стойности	Промяна от:
Базово време	-	<u>1ms</u> , 1s, 100ms, 10ms	-
Задание	%MN _i ,P	от 0 до <u>9999</u>	P, T(*)
Текуща стойност	%MN _i ,V	от %MN _i ,P до 0	T
Модификация	-	<u>YES</u> или NO	-

P - от програмата; T - от терминала; T(*) от терминала при зададено YES;

Подчертани са стойностите по подразбиране.

Действие:



Език за програмиране PL7-2

Специални случаи:

- STOP инструкцията и инструкцията за безусловен преход (JUMP) не оказват влияние на текущата стойност %MNi,V;
- При студен рестарт се нулира текущата стойност %MNi,V и заданието приема стойността, зададена в конфигурацията (ако по-късно е била променена);
- При топъл рестарт се запазват съществуващите стойности на %MNi,V и %MNi,P.

2.3 Таймер

Характеристики

Брой: 32 (%T0 - %T31)

Времезадръжка: мин. 10ms
макс: 9999mn

Входове	Графичен символ	Изходи	Асоциирани битове
Е: разрешение С: управление	 <p>2 колони, 3 линии</p>	D: Изтекло време R: Работа	%Ti,D %Ti,R

Параметри, свързани с блока:

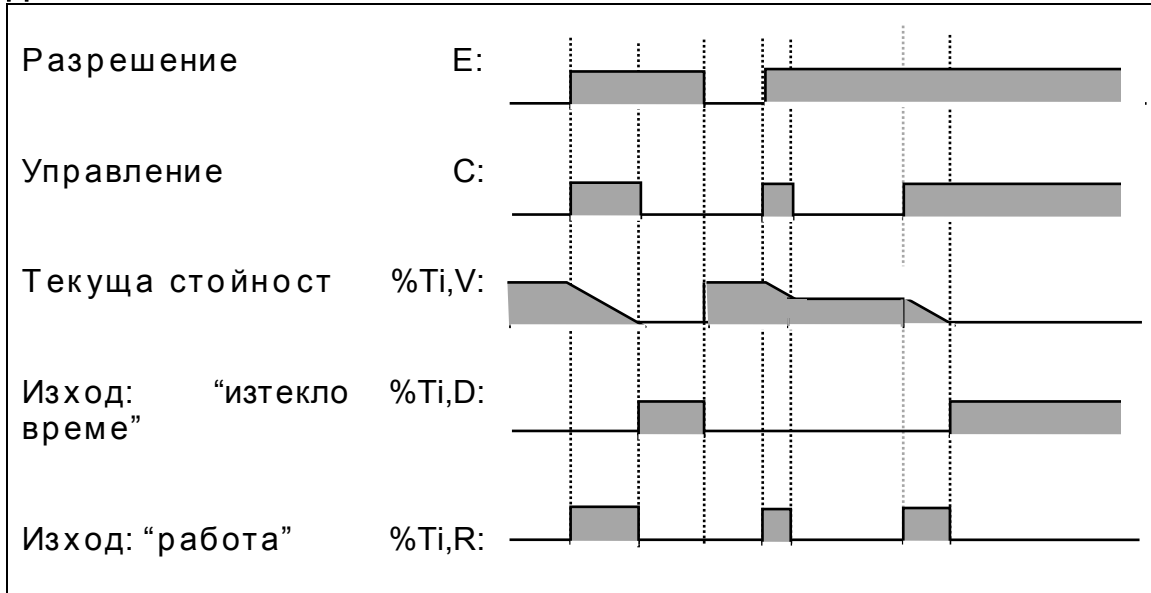
Параметър	Дума	Стойности	Промяна от:
Базово време	-	<u>1mn</u> , 1s, 100ms, 10ms	-
Задание	%Ti,P	от 0 до <u>9999</u>	P, T(*)
Текуща стойност	%Ti,V	от %Ti,P до 0	T
Модификация	-	<u>YES</u> или NO	-

P - от програмата; T - от терминала; T(*) от терминала при зададено YES;

Подчертани са стойностите по подразбиране.

Език за програмиране PL7-2

Действие:



Специални случаи:

- STOP инструкцията и инструкцията за безусловен преход (JUMP) не оказват влияние на текущата стойност M_i,V ;
- При студен рестарт се нулира текущата стойност M_i,V и заданието приема стойността, зададена в конфигурацията;
- При топъл рестарт се запазват съществуващите стойности на $\%M_i,V$ и $\%M_i,P$.

2.4 Регистър

Характеристики

Брой: 4 (R0 - R3)

Тип: FIFO: структура "опашка"
или LIFO: структура "стек"

Входове	Графичен символ	Изходи	Асоциирани битове
R: изчистване I: запис (импулс) O: четене (импулс)	<p>2 колони, 3 линии</p>	F: пълен E: празен	R_i,F R_i,E

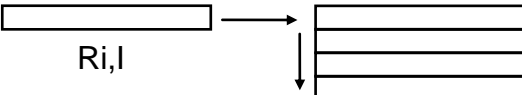
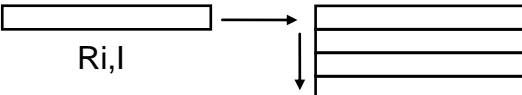
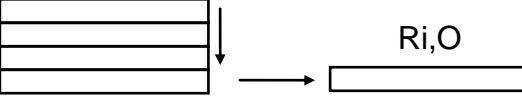
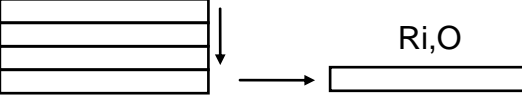
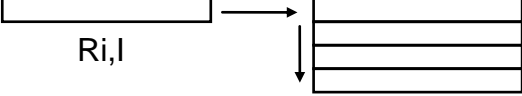
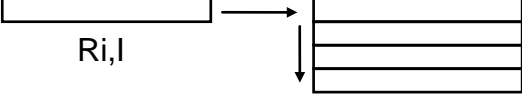
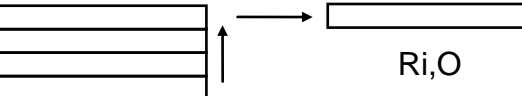
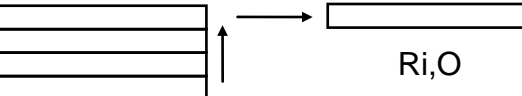
Параметри, свързани с блока:

Параметър	Дума	Стойности	Промяна от:
Тип	-	FIFO или LIFO	-
Дължина	-	от 1 до 16 думи	-
Входна дума	$\%R_i,I$	-	P,T
Изходна дума	$\%R_i,O$	-	P,T

Език за програмиране PL7-2

P - от програмата; T - от терминала; Подчертани са стойностите по подразбиране.

Действие:

Тип	Операция	Команда	Резултат
FIFO	Запис (ако блока не е пълен $R_i, F = 0$)	$R_i, I \uparrow$ 	
	Четене (ако блока не е пълен $R_i, E = 0$)	$R_i, O \uparrow$ 	
LIFO	Запис (ако блока не е пълен $R_i, F = 0$)	$R_i, I \uparrow$ 	
	Четене (ако блока не е пълен $R_i, E = 0$)	$R_i, O \uparrow$ 	

Специални случаи:

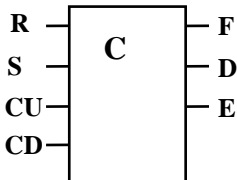
- При студен рестарт, блокът регистър се изчиства ($R_i, E = 0$).

2.5 Брояч

Характеристики

Брой: 31 (%C0 - %C30)

Броене нагоре: от 0 до 9999
или
Броене надолу: от 9999 до 0

Входове	Графичен символ	Изходи	Асоциирани битове
R: Нулиране P: Установяване на задание U: Увеличаване D: Намаляване		E: Празен D: Достигнато задание F: Пълен	%C _i ,E %C _i ,D %C _i ,F
2 колони, 4 линии			

Език за програмиране PL7-2

Параметри, свързани с блока:

Параметър	Дума	Стойности	Промяна от:
Задание	%Ci,P	от 0 до 9999	P, T(*)
Текуща стойност	%Ci,V	от 0 до 9999	T
Модификация	-	<u>YES</u> или NO	-

P - от програмата; T - от терминала; T(*) от терминала при зададено YES;

Подчертани са стойностите по подразбиране.

Действие:

Вход за нулиране (R)	в състояние 1, %Ci,V = 0
Вход за установяване (P)	в състояние 1, %Ci,V = %Ci,P
Вход за увеличаване (U)	при преден фронт увеличава Ci,V с 1
Вход за намаляване (D)	при преден фронт намалявава %Ci,V с 1
Изход "достигнато задание" (D)	=1, ако %Ci,V = %Ci,P
Изход "пълен" (F)	=1, ако %Ci,V се препълва от 9999 в 0
Изход "празен" (E)	=1, ако %Ci,V се препълва от 0 в 9999

Специални случаи:

- При студен рестарт %Ci,V се нулира, а %Ci,P приема стойността, зададена в конфигурацията на програмата;
- При топъл рестарт %Ci,V и %Ci,P запазват техните съществуващи стойности.

2.6 Бърз брояч/таймер

Представяне:

Контролерът TSX 17-20 разполага с бърз брояч "FC", който може да се конфигурира от потребителя като бърз брояч или като бърз таймер. Заданието се конфигурира от потребителя в програмен ре-жим (Program Mode) и може да се променя от програмата или от режима за настройка (Adjust Mode). Когато заданието се достигне, бързият брояч/таймер активира бързата задача.

Език за програмиране PL7-2

Характеристики:

- **Бърз брояч** (конфигуриран по подразбиране): Броячът има два физически входа (броене и нулиране), за свързване на 5 или 24VDC датчици. Максималната честота на броене е 2kHz и могат да бъдат преброени до 10 000 точки;
- **Бърз таймер**: Физическият вход за броене може да бъде заменен от вътрешни тактови импулси с база 0,555 ms. Постъпването на преден фронт на входа за нулиране, нулира таймера.

Входове	Графичен символ	Изходи	Асоциирани битове
R: Нулиране P: Установяване на задание V: Текуща стойност C: Управление	 <p>2 колони, 4 линии</p>	E: Празен D: Достигнато задание F: Пълен	%Ci,E %Ci,D %Ci,F

Параметри, свързани с блока:

Параметър	Дума	Стойности	Промяна от:
Задание	%FC,P	от 0 до 9999	P, T(*)
Текуща стойност	%FC,V	от 0 до 9999	T
Модификация	-	<u>YES</u> или NO	-

P - от програмата; T - от терминала; T(*) от терминала при зададено YES;

Действие:

Вход за нулиране (R)	в състояние 1, %FC,V=0
Вход за установяване (P)	в състояние 1, позволява зареждане на ново задание
Вход "текуща стойност" (V)	в състояние 1, позволява четене на текущата стойност от програмата
Вход за управление (C)	в състояние 1, позволява броенето
Изход "достигнато задание" (D)	=1, ако %FC,V = %FC,P
Изход "пълен" (F)	=1, ако %FC,V < %FC,P
Изход "празен" (E)	=1, ако физическият вход нулиране е 1

Бърз брояч/таймер (продължение)

Език за програмиране PL7-2

Системен бит %SY15

Когато бързият брояч/таймер достигне заданието:

- системният бит %SY15 се установява в 1;
- активира се бързата задача, ако предварително системен бит %SY19 е бил установен в 1 от основната задача;
- бързият брояч се нулира.

Забележка: Системният бит %SY15 трябва да бъде нулиран от бързата задача.

Физически връзки:

Свързването на бързия брояч/таймер се осъществява през 9-иглен куплунг, намиращ се в горния ляв ъгъл на базовия модул на TSX 17-20. Този куплунг притежава два напълно изолирани входа:

- **физически броячен вход:**
0V = канал 1 5V = канал 7 24V = канал 6
- **физически нулиращ вход:**
0V = канал 5 5V = канал 8 24V = канал 9

Специални случаи:

- Нулирането на брояча от нулиращия вход води до нулиране на текущата стойност %FC,V, ако предварително входа V е установен в 1. Стойността на динамичното задание се запазва;
- При студен рестарт, текущата стойност %FC,V се нулира, а заданието се установява в статичната си стойност, такава каквато е конфигурирана в програмния режим (Program Mode);
- При топъл рестарт текущата стойност %FC,V се запазва;
- Бързият брояч/таймер може да бъде четен и модифициран в режим на настройка (Adjust Mode).

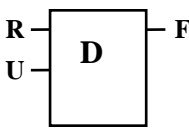
2.7 Стъпков избирач

Характеристики

Брой: 8 (D0 - D7)

Брой стъпки: 16

Брой командни битове: 16

Входове	Графичен символ	Изходи	Асоцииран и битове
R: изчистване U: преход (към нова стъпка -		F: пълнен (текуща стъпка = последна	%Di,F

Език за програмиране PL7-2

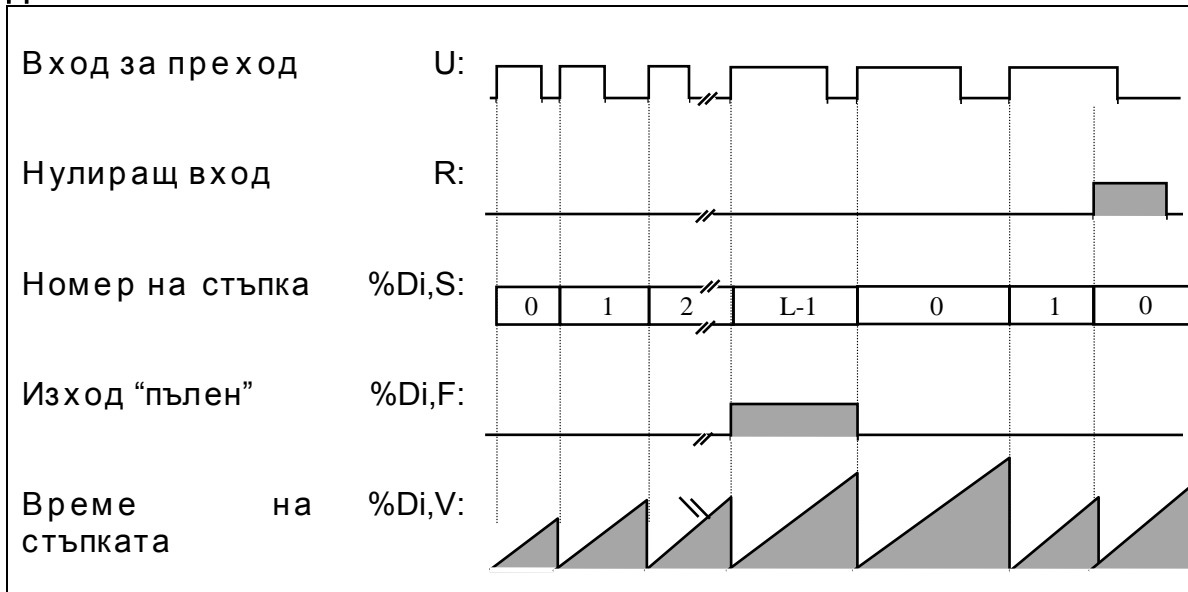
(преден фронт) 2 колони, 3 линии стъпка)

Параметри, свързани с блока:

Параметър	Дума	Стойности	Промяна от:
Брой на стъпките - L	-	от 1 до <u>16</u>	-
Номер на текущата стъпка	%Di,S	от 0 до 15	P,T
Състояния на стъпките	%Di,Wi	Бинарна стойност	-
Командни битове	-	Vi или Oi,i (максимум 16)	-
Време на стъпката	%Di,V	от 0 до <u>9999</u>	-
Базово време	-	<u>1m</u> , 1s, 100ms, 10ms	-

P - от програмата; T - от терминала; Подчертани са стойностите по подразбиране.

Действие :



Стъпков избирач (продължение)

Операции

Показаната таблица обобщава характеристиките на функционалния блок "стъпков избирач". Той работи на същите принципи, както и механичния стъпков избирач.

Език за програмиране PL7-2

Колона	F	E	D		2	1	0	
Команден бит	%Q 0,4	%Q 3,1	%Q 1,0		%M 5	%Q1 ,5	%Q3 ,8	
Стъпка 0	0	1	0		1	1	0	=%Di,W0
Стъпка 1	1	0	1		1	0	0	=%Di,W1
							
Стъпка 5	0	1	1		0	1	0	=%Di,W5
							
Стъпка 14	0	1	1		0	1	0	=%Di,W14
Стъпка 15	1	1	1		1	0	0	=%Di,W15

Текущият номер на стъпката се увеличава при всеки постъпващ преден фронт на входа U; този номер може да бъде модифициран от програмата.

Специални случаи:

- Инструкцията за безусловен преход (JUMP) не нулира командните битове;
- Командните битове се обновяват при смяна на стъпката или при топъл рестарт;
- При студен рестарт стъпковия избирач се установява в стъпка 0.

3.1 Блок за сравнение

Характеристики

Брой: Неограничен; програмират се в тестовата зона

Формат: OP1 | Знак | OP2



2 колони, 1 линия

OP1	Обект, който се сравнява	Всеки обект от тип дума
Знак	Тип на сравнението	>, <, =, >=, <=, <>
OP2	Стойност, с която се сравнява	Стойност или всеки обект от тип дума

Език за програмиране PL7-2

3.2 Блок за изчисление и трансфер

Характеристики

Брой: Неограничен, програмиране в зона на действията

-[OPER.]-

3 колони, 1 линия

Използвани обекти:

Думи: %MWi; %KWi; %SWi; %COMi,j; %Ti,P (напр.); %Xi,V; %IWx,i; %QWx,i

Стойности:

- Двоични L'101001001001101'
 - Шестнадесетични H'524D'
 - Текстови M'MR'
 - Десетични 21069
-

Низ от битове (блок от тип трансфер)

- Тип битове %Mi; %Qx,i; %Ix,i
 - Адресиране Адрес на 1-вия бит [брой битове]
брой битове <= 16
 - Пример B5[10] = низ от битове от B5 до B14
-

Индексно адресиране на думи

- Думи, които се индексират %MWi; %KWi; 0<= i или i<=127
 - Индекс %MWj 0<= стойност на индекса <= 127-i
 - Адресация Дума, която се индексира (индекс)
 - Пример %MW5(%MW10)
-

Таблицы от думи (блок от тип трансфер)

- Думи %MWi; %KWi
 - Адресация Първа дума от таблицата [брой думи]
брой думи < 128-i
 - Пример %KW5[10]
-

3.2 Блок за изчисление и трансфер (продължение)

Аритметични и логически операции

Формат: OP1 | OP | OP2 | □ □ Резултат

OP1	1-ви операнд	Всеки обект от тип дума
OP	операция	Код на операцията

Език за програмиране PL7-2

OP2	2-ри операнд	Стойност или всеки обект от тип дума
	Резултат	%MWi; %COMi,j; %Ti,P (напр.); %QWx,i

Тип	Код на операцията	Операнди и стойности
Аритметични	+, -	Стойности от -32768 до +32767
	*, /	Стойности от -32768 до +32767
	REM (1)	Стойности от -32768 до +32767
Логически	AND, OR, XOR	операции по битове
	CPL	не използва операнд 1

(1) REM = код на операции за извличане на остатъка от операция делене

Забележка: %SY18 регистрира пренос от операциите или делене на 0;
%SY17 регистрира пренос от операции върху числа без знак.

Преобразуване и циклично изместване

Формат: OP | OP2 | □ □ Резултат

OP	Тип на преобразуването	Код на операцията
OP2	Дума за преобразуване	Всеки обект от тип дума
	Резултат	%MWi; %COMi,j; %Ti,P (напр.); %QWx,i

%MWi и %COMi,j само за ASCII преобразуване

Тип	Код на операцията	Стойност на OP2	Резултат
Двоично/BCD преобразуване	BIN BCD	BCD Двоичен	Двоичен BCD
Двоично/ASCII преобразуване	BTA ATB	Двоичен ASCII (3 думи)	ASCII (3 думи) Двоичен
Циклично изместване	SLCn SRCn	Ляво Дясно	n = брой на битовете за изместване (<16)

Забележка: %SY18 контролира валидността на преобразуването; за BCD преобразуване стойността трябва да е < 9999.
BTA/ATB преобразуването се извършва върху 5 цифрени

Език за програмиране PL7-2

стойности, с ASCII кодове в 3 последователни думи
(задава се адрес само на първата дума).

3.2 Блок за изчисление и трансфер (продължение)

Операторен блок от тип трансфер

Формат: OP2 | Резултат

OP2 (източник)	<input type="checkbox"/> (приемник)	Пример
Низ от 16 бита	дума	%I1,0[16] <input type="checkbox"/> %MW4
Низ от 16 бита	индексирана дума	%I1,0[16] <input type="checkbox"/> %MW4(%MW50)
Низ от битове по-малък от 16 бита	низ от битове с дължина = OP2	%I1,0[8] <input type="checkbox"/> %M0[8]
Дума	низ от 16 или по-малко бита	%MW3 <input type="checkbox"/> %Q2,5[5]
Дума	дума	%MW30 <input type="checkbox"/> %M2,P
Дума	индексирана дума	%MW5 <input type="checkbox"/> %MW100(W50)
Индексирана дума	дума	%KW10(W0) <input type="checkbox"/> %MW30
Индексирана дума	индексирана дума	%KW50(W0) <input type="checkbox"/> %MW10(W1)
Таблица от думи	таблица от думи	%MW512[60] <input type="checkbox"/> %MW0[60]
Стойност	низ от 16 или по-малко бита	51 <input type="checkbox"/> %Q1,5[8]
Стойност	дума	127 <input type="checkbox"/> %T1,P
Стойност	индексирана дума	127 <input type="checkbox"/> %MW10(%MW50)

Адресиране на всички 1024 Wi и 1024 KWi

Wi и CWi са структури-
в 8 страници

000 - 127
127 - 255
256 - 383
384 - 511
512 - 639
640 - 767
768 - 895
896 - 1023

Директен достъп до стр. 1

Достъп до другите страници
чрез трансфер в страница
1

Език за програмиране PL7-2

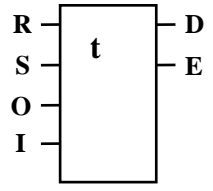
Пример: %MW512[128] □ %MW0[128]
 %KW640[60] □ %MW0[60]

4.1 Текстов блок

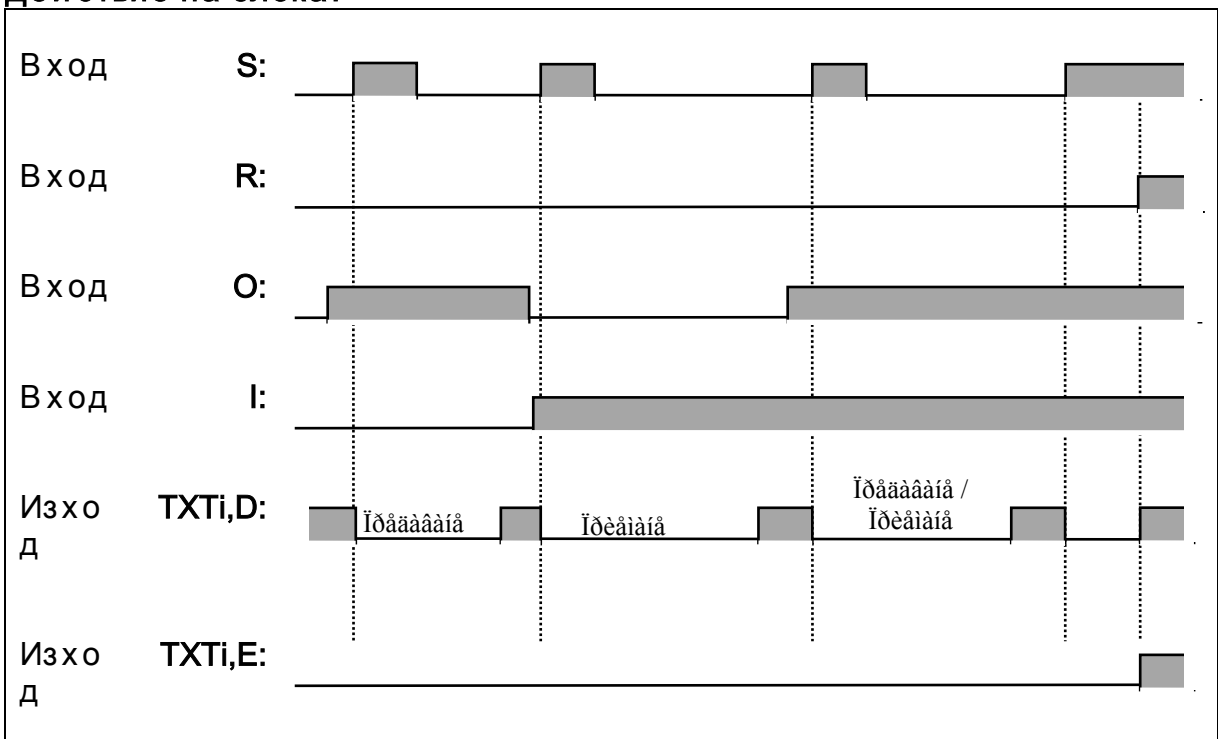
3 типа текстови блока

Тип	Обмен	Достъп
TER	Програма □□ терминален порт	LOCAL
TXT	Програма □□ друга програма	NETWORK
CPL	Програма □□ UNI-TELWAY магистрала	BUS

Общи характеристики Брой: 8 (от TXT0 до TXT7)

Входове	Графичен символ	Изходи	Асоциирани битове
R: Начално установяване S: Старт (импулс) O: Предаване I : Приемане	 <p>2 колони, 4 линии</p>	D: Край работа E: Грешка	%TXTi,D %TXTi,E

Действие на блока:



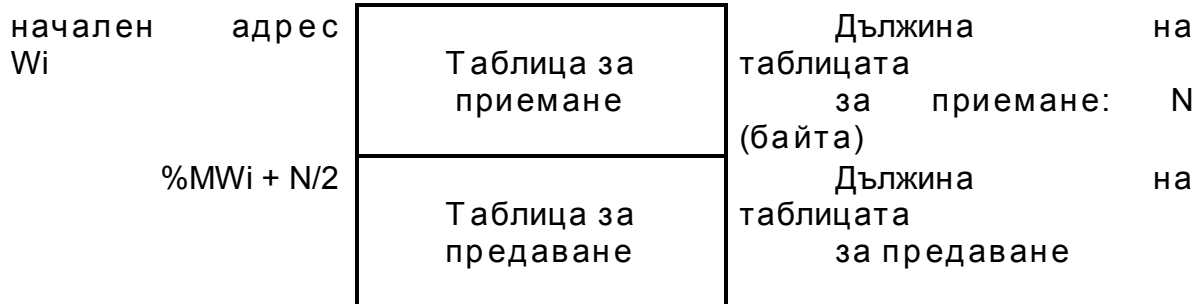
Език за програмиране PL7-2

4.1 Текстов блок (продължение)

Параметри на текстовия блок

Параметър	Думи	Стойности
Начален адрес на таблицата	%MWi или %KWi	$0 \leq i \leq 127$
Дължина на таблицата за приемане	[n]	от 0 до 30 байта
Дължина на таблицата за предаване	%THTi,L (1)	от 0 до 30 байта
Състояние на обмена (статус)	%THTi,S	брой приети или предадени битове, ако %THTi,E = 0; ако %THTi,E = 1: <ul style="list-style-type: none"> • 1 = прекъсване от RESET (вход R); • 2 = грешна дължина на таблицата; • 3 = отказано съобщение; • 7 = грешка при преобразуване; • 11 = несъвместим тип на текстов блок.

(1) - Възможна промяна от терминала и/или от програмата
Организация на таблицата за приемане / предаване



Специфични параметри за текстов блок тип ТХТ

Параметър	Думи	Стойности	Възможност за промяна
Номер на отдалечения текстов блок (в другия контролер)	%THTi,T	от 0 до 63 (2)	Р или Т

Език за програмиране PL7-2

Адрес на приемащата станция	%ТХТi,A	от 0 до F (шестнадесетично)	Р или Т
-----------------------------	---------	--------------------------------	---------

Р - от програмата; Т - от терминала;

(2) от 0 до 7, ако отдалечения контролер е TSX 17-20 или TSX 47-J/47-10/20.

4.1 Текстов блок (продължение)

Текстов блок тип CPL

Параметър	Думи	Стойности	Възможност за промяна
Адрес	ТХТi,M	Н'0х..' (шестдесетично) х = Слот, от 0 до 3 .. = Лог. адрес, от 00 до FF	Р или Т
Код на заявката	ТХТi,C	Дефиниране на обменните данни (Вж. съответните ръководства)	Р или Т
Отчет	ТХТi,R	Показва валидността на обмена	-

Р - от програмата; Т - от терминала;

Специални случаи

Студеният и топлият рестарт предизвикват:

- Установяване в 1 на ТХТi,D и нулиране на ТХТi,E;
- Установяване в зададените в конфигурацията стойности на параметрите: ТХТi,L; ТХТi,T; ТХТi,A; ТХТi,M; ТХТi,C.

6.1 Език Grafset - характеристики

Страница Grafset

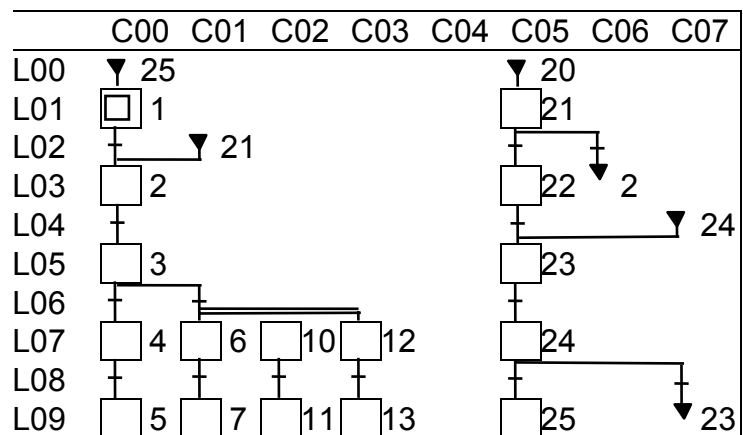
1 страница на Grafset =

8 колони от:

- 6 стъпки;
- 6 прехода;
- 1 входна връзка;
- 1 изходна връзка;

Възможни са максимум

8 страници - от P0 до P7



Език за програмиране PL7-2



Автоматен граф, в смисъла на Grafset

1 граф = последователност от стъпки и преходи, построена съгласно правилата на езика Grafset.

Забележка: Един граф може да бъде програмиран на няколко страници и една страница може да съдържа няколко графа.

Действия в стъпките и условия за преходите

Всяка стъпка **може** да бъде свързана с контактна мрежа, дефинираща действията свързани с дадена стъпка.

Всеки преход **трябва** да бъде свързан с контактна мрежа, дефинираща условията за извършване на прехода.

6.2 Основни елементи

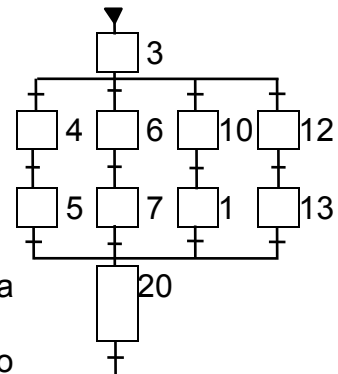
Всички базови елементи са систематизирани в раздел 1.5

Избор на последователност

Максимален брой преходи
в посока надолу: 4

Максимален брой преходи
в посока нагоре: 4

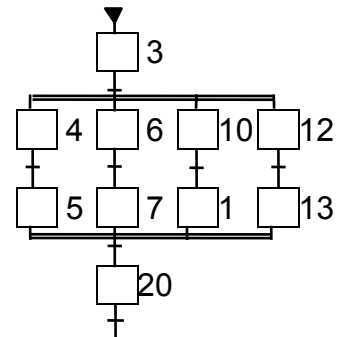
За да се избегне едновременно сработване на няколко прехода, условията в тях трябва да са взаимно изключващи се.



Език за програмиране PL7-2

Активиране и деактивиране на паралелни действия

Максимум: 4 стъпки в посока надолу
4 стъпки в посока нагоре



Входни и изходни връзки

Предназначение: Да се определи от къде и на къде се предава

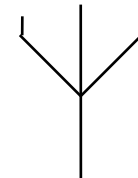
управлението по графа

Отдалечени връзки (изходна) =
= символ + N: на отдалечената стъпка, от
която се предава управлението.

Максимален брой: 42

Отдалечени връзки (входна) =
= символ + N: на отдалечената стъпка, към
която се предава управлението.

Максимален брой: 42



Всяка входна връзка задължително трябва да бъде свързана със съот-ветстваща и изходна връзка.

6.3 Стъпки и свързани с тях действия

Стъпки

Максимален брой: 96, включително 16 инициализиращи стъпки;

Битове, свързани със стъпките: X_i , i = номер на стъпката;

$X_i = 0$, ако стъпката не е активна;

$= 1$, ако стъпката е активна.

Максимум 16 стъпки могат да бъдат активни едновременно.

Свързани със стъпките действия: могат да се програмират по два начина:

1. Като контактни мрежи в стъпките на **Grafcet**, състоящи се от:
Етикет = номера на стъпката (генерира се автоматично)

+

Език за програмиране PL7-2

Коментар, съдържащ максимум 15 символа
(незадължителен)

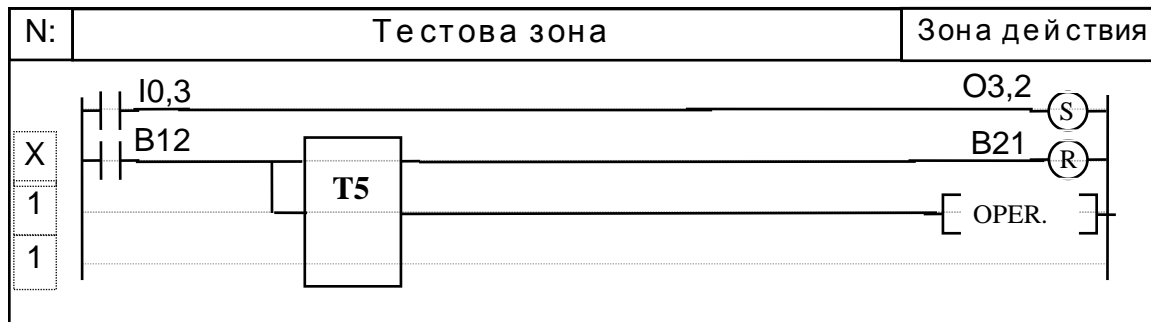
+

Контактна мрежа с максимум 4 действия, които могат да бъдат:

- безусловни;
- условни;
- с време задръжка.

Тези контактни мрежи се изпълняват, само когато съответстващата им стъпка е активна.

За стъпките, съдържащи функционални блокове, е необходимо да се осигури преден фронт за съответните входове.



2. В частта на Препроцесинга

С използване на битовете X_i , свързани със стъпките. В този случай броя на действията е неограничен



Стъпки и свързани с тях действия (продължение)

Активно време на стъпката

X_i, V = активно време на стъпката (от 0 до 9999 с базова стъпка 1s).

Език за програмиране PL7-2

Xi,V започва да се увеличава веднага щом се активира съответната стъпка и се нулира при настъпване на всяко от следните събития:

- при деактивиране на стъпката;
- при реинициализация на Grafcet (%SY21=1);
- при студен рестарт (%SY0=1);
- при нулиране (RESET) на Grafcet (%SY22=1);
- при установяване (PRESET) на Grafcet (%SY23=1).

Xi,V продължава да се увеличава когато изпълнението на програмата е спряло.

6.4 Преходи между стъпките и свързани с тях условия

Преходи

Максимален брой: 128 прехода максимум; 24 от тях могат да бъдат разрешавани едновременно.

Условия за преходите

Условията, свързани с преходите се състоят от :

Етикет, съдържащ номера на стъпката източник и номера на стъпката, към която се преминава (генерира се автоматично)

+

Коментар от максимум 15 символа (незадължителен)

+

Контактна мрежа, в която само зоната на теста може да се използва

Тази контактна мрежа се сканира само когато стъпката преди съответния преход е активна.

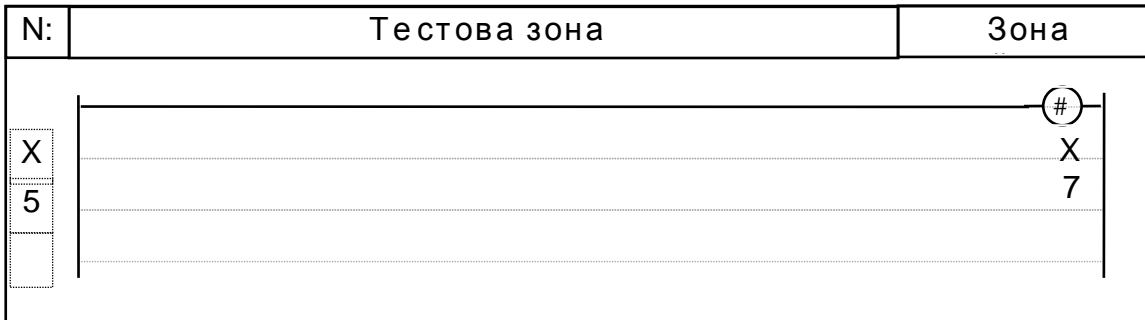
Забележка: Когато са налице няколко стъпки източници или няколко стъпки, към които се извършва прехода, номерът посочен в етикета на мрежата е този на най-лявата стъпката.

Език за програмиране PL7-2

Преходи между стъпките и свързани с тях условия (продължение)

Специални условия:

Условие за преход, който винаги се изпълнява:



Условие, което не е зададено = условие, което никога не е изпълнено

Използуване на активното време на стъпката в условието за преход:



Език за програмиране PL7-2

6.5 Структура на програмата

PL7-2 Grafset - програма

Програмата на PL7-2 Grafset е структурирана в три секции, всяка от които има специфични функции

Основна задача



Серия от контактни мрежи

Функции:

- връщане на захранването;
- повреди
- смяна на режима на управление:
 - ⇒ инициализация (%SY21);
 - ⇒ препозициониране (%SY21);
 - ⇒ замръзване (%SY21);
- входна логика.

Поредица от стъпки на Grafset и условия за преходи между тях, заедно с асоциираните им мрежи написани на езика Ladder. **Функции:** Програмиране на последователните стъпки от управлението на обекта

Серия от контактни мрежи

Функции:

- команди, свързани със състоянието на стъпките в Grafset;
- действия, свързани с изходите.

Дефиниция на цикъла на сканиране е дадена в секция 1.6

Забележка:

Език за програмиране PL7-2

- Бързата задача може да бъде програмирана само на езика Ladder, независимо от езика, използван в основната задача;
- Инструкциите за преход (JUMP) могат да бъдат използвани само в рамките на една програмна секция (препроцесинг или постпроцесинг);
- Контактните мрежи в препроцесинга и постпроцесинга могат да имат едни и същи етикети.

7.1 Отпадане/възстановяване на захранването

Установяване на отпадане на захранването

Прекъсване за по-голямо време от автономността на захранващия модул в контролера.

Последователност при отпадане на захранването

Продължителност на прекъсването	Поведение на TSX 17-20
повече от 10 ms	<ul style="list-style-type: none">• нормално изпълнение на програмата
повече от автономността на захранващия модул	<ul style="list-style-type: none">• спиране на сканирането;• запазване на информация за текущо-то състояние*;• спиране на процесора.

(*) съхраняване на текущото състояние е възможно само ако контроле-рът е снабден с литиева батерия. В противен случай, тази информация се губи след 1 час.

Топъл рестарт (запазени данни - програма, данни и вх./изх. състояния)

Възможни причини:

- Връщане на захранването след прекъсване за повече от 10 ms;
- Връщане на захранването след прекъсване за повече от времето на автономност на захранващия модул;
- Установяване в 1 на SY1 от програмата или терминала.

Последователност при топъл рестарт

- %SY1 се установява в 1;
- %QWx,i, изходите без памет %Qx,i и %Mi се нулират;

Език за програмиране PL7-2

- Програмният цикъл се рестартира от точката на прекъсване, но изходите не се обновяват преди края на следващия цикъл. Тогава се нулира и %SY1 (от системата).

Студен рестарт (загуба на текущото състояние на програмата)

Възможни причини:

- Липсваща или повредена батерия;
- сменена EEPROM или EPROM касета;
- Установяване в 1 на %SY0 от програмата или терминала
- Инициализация от програматора.

Последователност при студен рестарт

- %SY0 се установява в 1;
- Нулиране на всички вътрешни битове, вх./изх. битове, текущи стойности на функционални блокове;
- Загуба на стойностите на функционалните блокове, които са били допълнително променени от програмата или програматора;
- Премахване на форсираните битове и точките на прекъсване;
- Реинициализация на Графсета;
- Работата на контролера започва от началото на цикъла (входове, програма, изходи) и %SY0 се нулира от системата

Забележка: Бързата задача не може да бъде изпълнена докато %SY0 и %SY1 не бъдат нулирани, т.е. в първия цикъл след рестарт (топъл или студен) на контролера.