



LOGO!

программирование

# Что такое LOGO! ?

LOGO! – это универсальный логический модуль фирмы Siemens.

LOGO! включает в себя

- функции управления
- блок управления и отображения
- блок питания
- интерфейс для программных модулей и кабеля PC
- готовые к использованию функции, часто требуемые в повседневной работе, например, функции задержки включения и выключения и импульсное реле
- часовой выключатель
- двоичные маркеры

ВХОДЫ И ВЫХОДЫ В СООТВЕТСТВИИ С ТИПОМ УСТРОЙСТВА

# Что может делать LOGO!?

LOGO! можно использовать для решения различных технических задач:

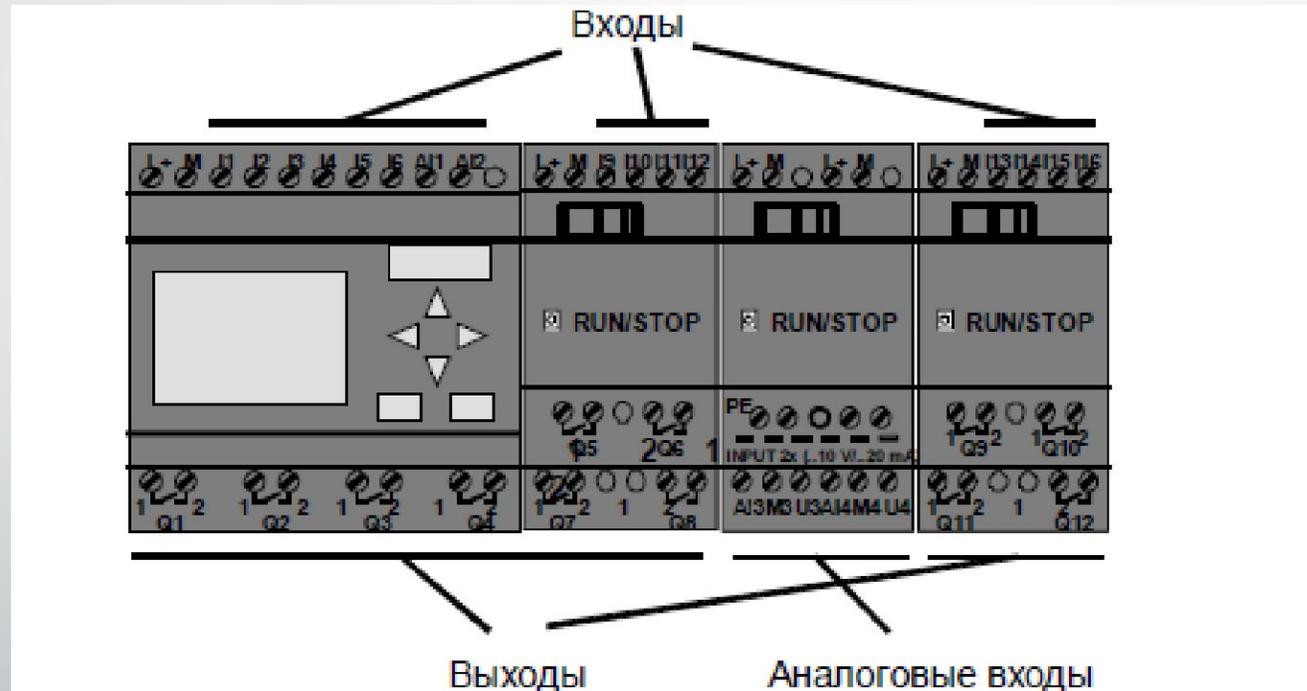
- в электрооборудовании жилых помещений (например, освещение лестничных клеток, внешнее освещение, тенты, жалюзи, освещение витрин магазинов и т.д.),
- в коммутационных шкафах,
- в управлении машинами и аппаратами (например, системы управления воротами, вентиляционные системы или насосы для откачки дождевой воды и т.д.).

# Программирование LOGO!

- Под программированием мы подразумеваем ввод схемы.
- Программа LOGO! является на самом деле ничем иным, как коммутационной схемой, представленной другим способом.

# Соединительные элементы

У LOGO! есть входы и выходы. Каждый вход обозначается буквой I и номером. Когда вы смотрите на LOGO! спереди, то сверху вы видите клеммы для входов. Каждый выход обозначается буквой Q и номером. Клеммы для выходов видны на рисунке снизу.



Термин «соединительный элемент» относится ко всем соединениям и состояниям в LOGO!.

Входы и выходы могут иметь состояние .0. или .1..

0. означает, что на входе нет напряжения;

.1. означает, что оно есть.

Чтобы облегчить создание коммутационной программы, создатели LOGO! Soft Comfort ввели соединительные элементы hi, lo:

.hi. (high = высокий) имеет фиксированное состояние .1.,

а .lo. (low = низкий) имеет фиксированное состояние .0..

# LOGO! распознает следующие соединительные элементы

Соединительные элементы	LOGO! Basic / Pure 		DM 	AM 
Входы	LOGO! 230RC/RCo, LOGO! 24 RC/RCo	Две группы: I1... I4 и I5 ... I8	I9 ... I24	AI1...AI8
	LOGO! 12/24RC/RCo, LOGO! 24/24o	I1... I6, I7, I8	I9 ... I24	AI3...AI8
Выходы	Q1...Q4		Q5 ... Q16	нет
Io	Сигнал с уровнем '0' (выключено)			
Ii	Сигнал с уровнем '1' (включено)			
x	Существующий соединительный элемент, который не используется			

DM: Цифровой модуль.

AM: Аналоговый модуль.

# Блоки и номера блоков

- Блок в LOGO! - это функция, которая преобразует входную информацию в выходную информацию.

Раньше мы должны были подключать отдельные элементы в шкафу управления или в клеммной коробке.

При создании коммутационной программы мы связываем соединительные элементы с блоками. Для этого мы просто выбираем желаемое соединение из меню **Co**.

Обозначение меню имя **Co**, является сокращением английского термина «Connector», т.е. соединительный элемент.

# Логические операции

Простейшими блоками являются логические операции:

- AND [И]
- OR [ИЛИ]

Значительно более мощными являются специальные функции:

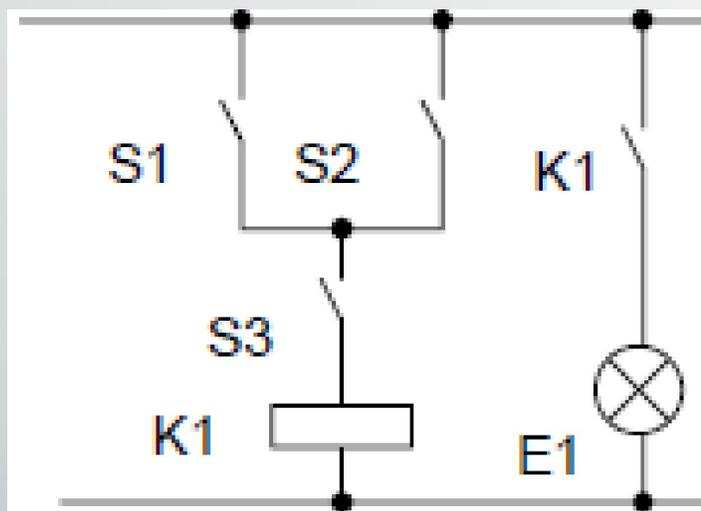
- Импульсное реле
- Реверсивный счетчик
- Задержка включения

Программный выключатель

# От коммутационной схемы к LOGO!

Конечно, вы знаете, что собой представляет коммутационная схема.

Пример:



Потребитель E1 включается и выключается с помощью выключателей (S1 **OR** S2) **AND** S3. (OR=ИЛИ; AND=И).

Реле K1 срабатывает, когда замкнуты S1 или S2, а также S3.

В LOGO! схема создается соединением друг с другом блоков и соединительных элементов

Коммутационная программа автоматически присваивает неиспользуемым соединительным элементам состояние, обеспечивающее надлежащее функционирование соответствующего блока. Можно обозначить неиспользуемые соединительные элементы символом .x..



# Пример создания программы

[Ссылка на ролик в Ютубе по программированию Logo!](#)

- [https://www.youtube.com/watch?time\\_continue=54&v=Moq6kBxY2\\_U](https://www.youtube.com/watch?time_continue=54&v=Moq6kBxY2_U)

## Выходы и входы

- . Коммутационная программа всегда вводится от выхода к входу.
  - . Можно соединить выход с несколькими входами, но не несколько выходов с одним входом.
  - . Нельзя соединять выход с предшествующим входом в пределах одного программного пути.
- Для образования таких внутренних обратных связей включайте промежуточные флаги или выходы.

# Функции LOGO!

LOGO! в режиме программирования предоставляет в ваше распоряжение различные элементы.

Чтобы при этом не потерять общего представления, разработчики разделили

эти элементы на списки. Этими списками являются:

↓ Co: список соединительных элементов (Connector [Соединительный элемент]).

↓ GF: список основных функций AND [И], OR [ИЛИ], ...

↓ SF: список специальных функций

↓ BN: список готовых к использованию в коммутационной программе блоков

# Константы и соединительные элементы . Co

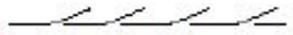
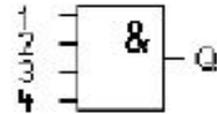
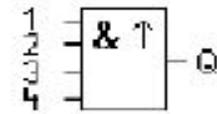
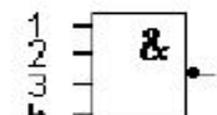
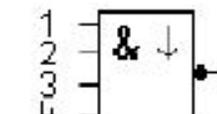
- Константы и соединительные элементы (= Co) – это входы, выходы, биты памяти и фиксированные уровни напряжения (константы).

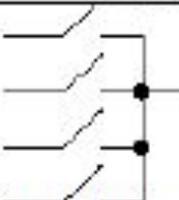
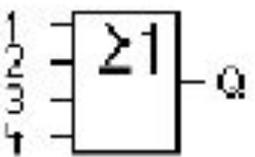
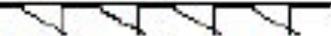
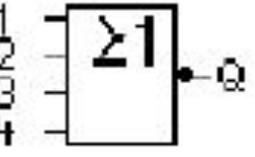
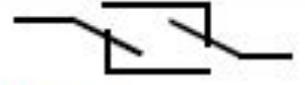
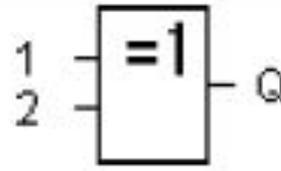
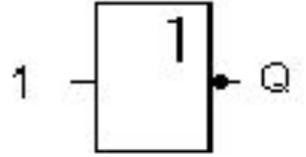
# Список основных функций - GF

Основные функции - это простые логические элементы булевой алгебры.

Можно инвертировать входы отдельных основных функций, т.е. коммутационная программа инвертирует логическую «1» на соответствующем входе в логический «0»; если же на входе установлен «0», то программа устанавливает логическую «1».

Имеются следующие основные функции:

Представление на коммутационной схеме	Представление в LOGO!	Наименование основной функции
 Последовательное соединение замыкающих контактов		AND (И)
		AND с анализом фронта
 Параллельное соединение размыкающих контактов		NAND (И-НЕ)
		NAND с анализом фронта

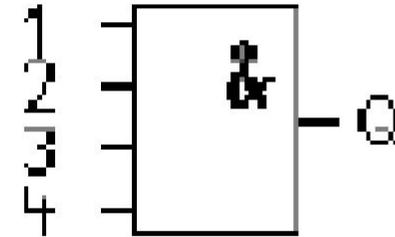
Представление на коммутационной схеме	Представление в LOGO!	Наименование основной функции
 <p>Параллельное соединение замыкающих контактов</p>		OR (ИЛИ)
 <p>Последовательное соединение размыкающих контактов</p>		NOR (ИЛИ-НЕ)
 <p>Двойной перекидной контакт</p>		XOR (исключающее ИЛИ)
 <p>Размыкающий контакт</p>		NOT (отрицание, инверсия)

# AND (И)

Последовательное соединение  
нескольких замыкающих контактов  
на коммутационной схеме:



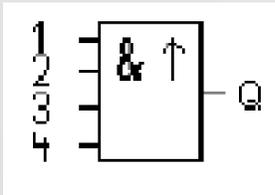
Символ в LОGО!:



Выход И принимает состояние 1 только тогда, когда **все** входы имеют состояние 1 (т.е. все контакты замкнуты).

Если какой-либо вход этого блока не подключен (х), то для этого входа  $x = 1$ .

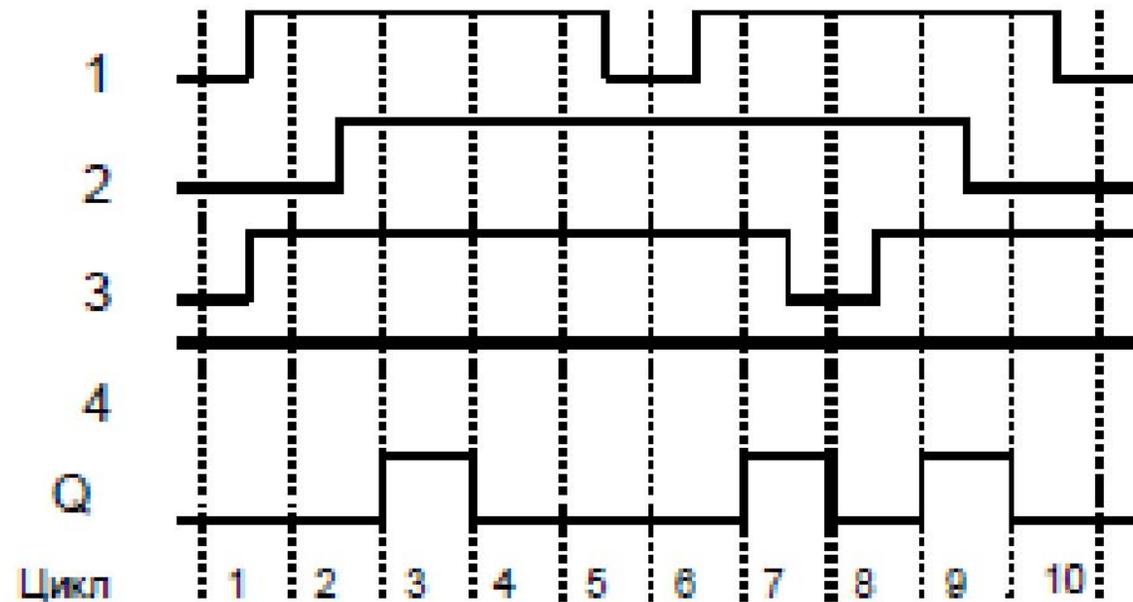
# AND с анализом фронта



Выход функции И с анализом фронта принимает состояние 1 только тогда, когда **все** входы имеют состояние 1 и **хотя бы один** вход в предыдущем цикле имел состояние 0.

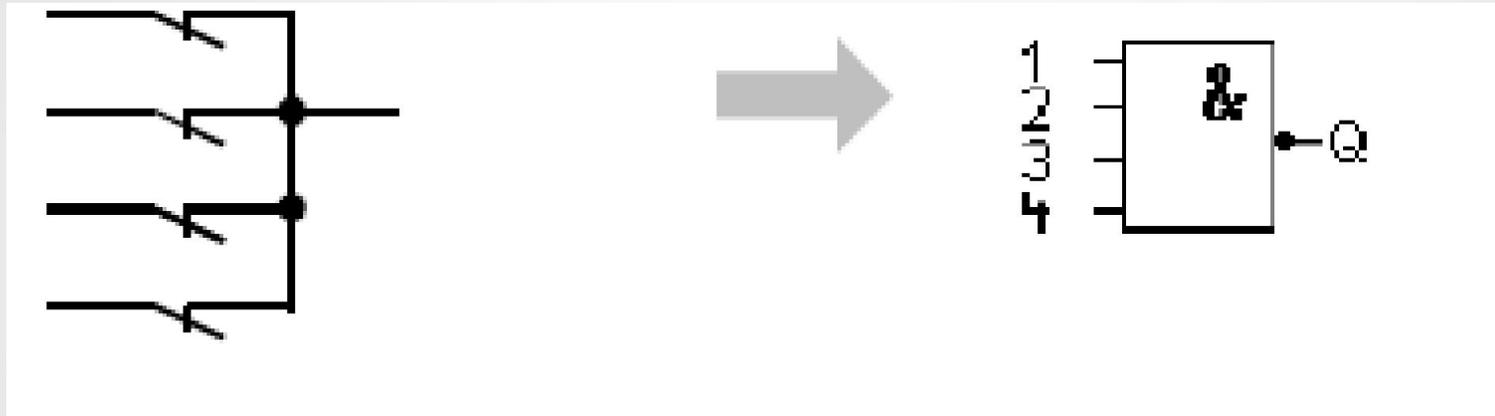
Если какой-либо вход этого блока не используется (x), то для этого входа  $x = 1$ .

## Временная диаграмма для функции И с анализом фронта



# NAND (И-НЕ)

Параллельное соединение нескольких размыкающих контактов на коммутационной схеме:



Символ в LOGO!:

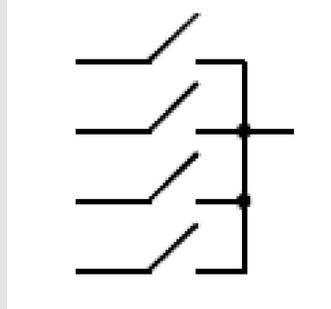
Выход функции NAND принимает состояние 0 только тогда, когда на **все** входы подан сигнал 1 (в коммутационной схеме все контакты разомкнуты).

Если какой-либо вход этого блока не подключен (x), то для этого входа  $x = 1$ .

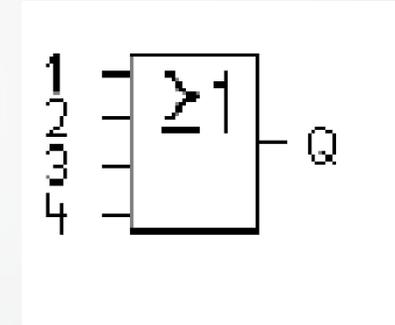


# OR (ИЛИ)

Параллельное соединение нескольких замыкающих контактов на коммутационной схеме:



Символ в LOGO!:



Выход функции ИЛИ принимает состояние 1, если **хотя бы один** вход имеет состояние 1 (т.е. замкнут).

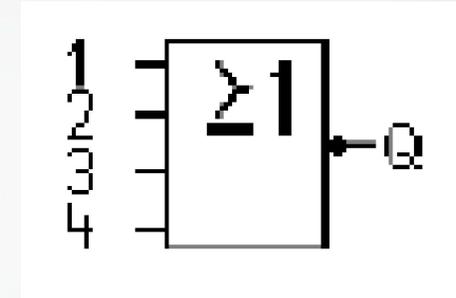
Если какой-либо вход этого блока не используется (х), то для этого входа  $x = 0$ .

# NOR (ИЛИ-НЕ)

Последовательное соединение нескольких размыкающих контактов на коммутационной схеме:



Символ в LOGO!:



Выход функции ИЛИ-НЕ принимает состояние 1 только тогда, когда **все** входы имеют состояние 0 (т.е. они выключены). Как только любой из входов включается (состояние 1), выход И-НЕ устанавливается в 0.

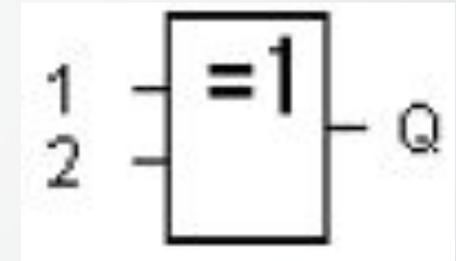
Если какой-либо вход этого блока не используется (x), то для этого входа  $x = 0$ .

# XOR (исключающее ИЛИ)

Исключающее ИЛИ на коммутационной схеме представляется последовательным соединением двух перекидных контактов:



Символ в LOGO!:

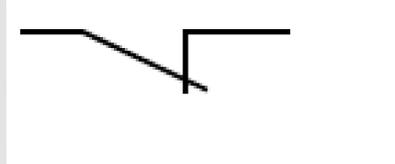


Выход исключающего ИЛИ принимает состояние 1, если входы имеют **разные** состояния.

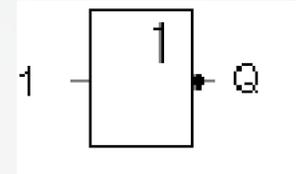
Если какой-либо вход этого блока не используется (x), то для этого входа  $x = 0$ .

# NOT (НЕ, отрицание, инверсия )

Размыкающий контакт на коммутационной схеме:



Символ в LOGO!:



Выход принимает состояние 1, если вход имеет состояние 0.  
Иными словами, функция НЕ инвертирует состояние входа.

Преимущество функции НЕ может быть проиллюстрировано следующим примером: вам больше не нужны размыкающие контакты для LOGO! Вы просто используете замыкающий контакт и преобразуете его в размыкающий контакт с помощью блока НЕ.

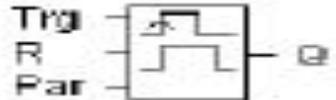
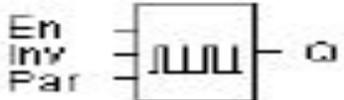
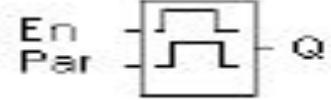
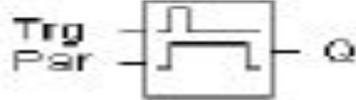
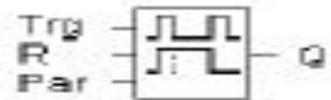
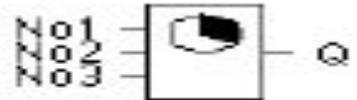
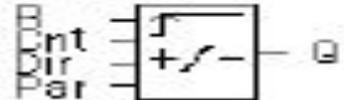
# Основные сведения о специальных функциях

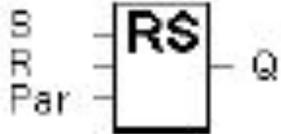
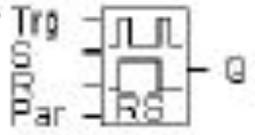
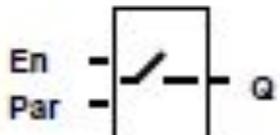
Специальные функции отличаются от основных функций, на первый взгляд, из-за различий в обозначении их входов.

Специальные функции включают в свой состав функции времени, обладают свойством сохраняемости и различными возможностями параметризации, чтобы приспособить программу к вашим индивидуальным потребностям.

# Список специальных функций - SF

Представление в LOGO!	Название специальной функции	Rem
<b>Таймеры</b>		
	Задержка включения	REM
	Задержка выключения	REM
	Задержка включения/ выключения	REM
	Задержка включения с запоминанием	REM
	Интервальное реле времени (вывод импульса)	REM

Представление в LOGO!	Название специальной функции	Rem
	Интервальное реле времени, запускаемое фронтом	REM
	Асинхронный генератор импульсов	REM
	Генератор случайных импульсов	
	Выключатель света на лестничной клетке	REM
	Двухфункциональный выключатель	REM
	Семидневный часовой выключатель	
	Двенадцатимесячный часовой выключатель	
<b>Счетчики</b>		
	Реверсивный счетчик	REM

Представление в LOGO!	Название специальной функции	Rem
<b>Разное</b>		
	Самоблокирующееся реле	REM
	Импульсное реле	REM
	Тексты сообщений	
	Программный выключатель	REM
	Регистр сдвига	REM

## Логические входы

- . **S (Set = установить)**: Сигнал на входе S устанавливает на выходе логическую «1».
- . **R (Reset = сбросить)**: Вход сброса R имеет приоритет над всеми остальными входами и переключает выходы в «0».
- . **Trg (Trigger = запустить)**: Этот вход используется для запуска функции на выполнение.
- . **Cnt (Count = считать)**: Этот вход используется для счета импульсов.
- . **Fre (Frequency = частота)**: К входу с этим описанием прикладываются частотные сигналы, подлежащие анализу.
- . **Dir (Direction = направление)**: Этот вход используется, например, для установки направления, в котором должен считать счетчик.
- . **En (Enable = разрешить)**: Этот вход разблокирует функцию, выполняемую блоком. Если на этом входе «0», то другие сигналы блоком игнорируются.
- . **Inv (Invert = инвертировать)**: Выходной сигнал блока инвертируется, когда этот вход активизирован.
- . **Ral (Reset all = сбросить все)**: Сбрасываются все внутренние значения.

**Соединительный элемент X на входах специальных функций** Если вы подключаете входы специальных функций к соединительному элементу x, то этим входам будет присвоено значение 0, т.е. к ним прикладывается сигнал низкого уровня.

## Параметрические входы

Имеется несколько входов, к которым вы не прикладываете сигналы, а параметризуете соответствующий блок определенными значениями.

Примеры:

- **Par (Parameter = параметр):**

Этот вход не подключается. Здесь для блока устанавливаются параметры (времена, пороги включения и выключения и т.д.).

- **No (Nocken = шаблон):**

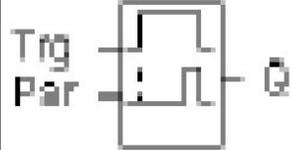
Этот вход не подключается. Здесь устанавливается шаблон времени.

- **P (Priority = приоритет):**

Этот вход не подключается. Здесь устанавливаются приоритеты.

# Задержка включения

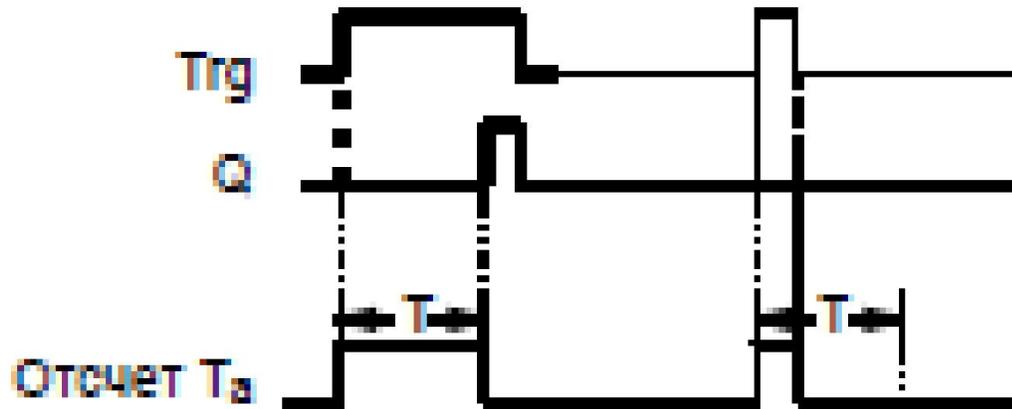
При задержке включения выход включается только по истечении параметризуемого интервала времени.

Символ в LOGO!	Подключение	Описание
	Вход Trg	Через вход Trg (trigger = запустить) производится запуск отсчета времени задержки включения.
	Параметр	T представляет время, по истечении которого включается выход (выходной сигнал переключается с 0 на 1). Сохраняемость: / = сохраняемость отсутствует R = состояние сохраняется.
	Выход Q	Q включается по истечении заданного времени T, если Trg все еще установлен.

Допустимые диапазоны для базы времени, если T является параметром

База времени	макс. величина	мин. разрешение	Точность
s (секунды)	99:99	10 мс	+ 10 мс
m (минуты)	99:59	1 с	+ 1 с
h (часы)	99:59	1 мин	+ 1 мин

### Временная диаграмма



Когда состояние входа Trg меняется с 0 на 1, начинается отсчет времени  $T_a$  ( $T_a$  - это текущее время в LOGO!).

Если состояние сигнала на входе Trg остается равным 1, по крайней мере, в течение заданного времени T, то выход устанавливается в 1 по истечении времени T (имеет место задержка между включением входа и появлением сигнала на выходе).

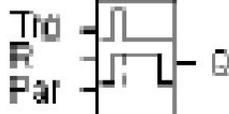
Если состояние сигнала на входе Trg снова становится равным 0 до истечения времени T, то время сбрасывается.

Выход сбрасывается в 0, если состояние сигнала на входе Trg равно 0.

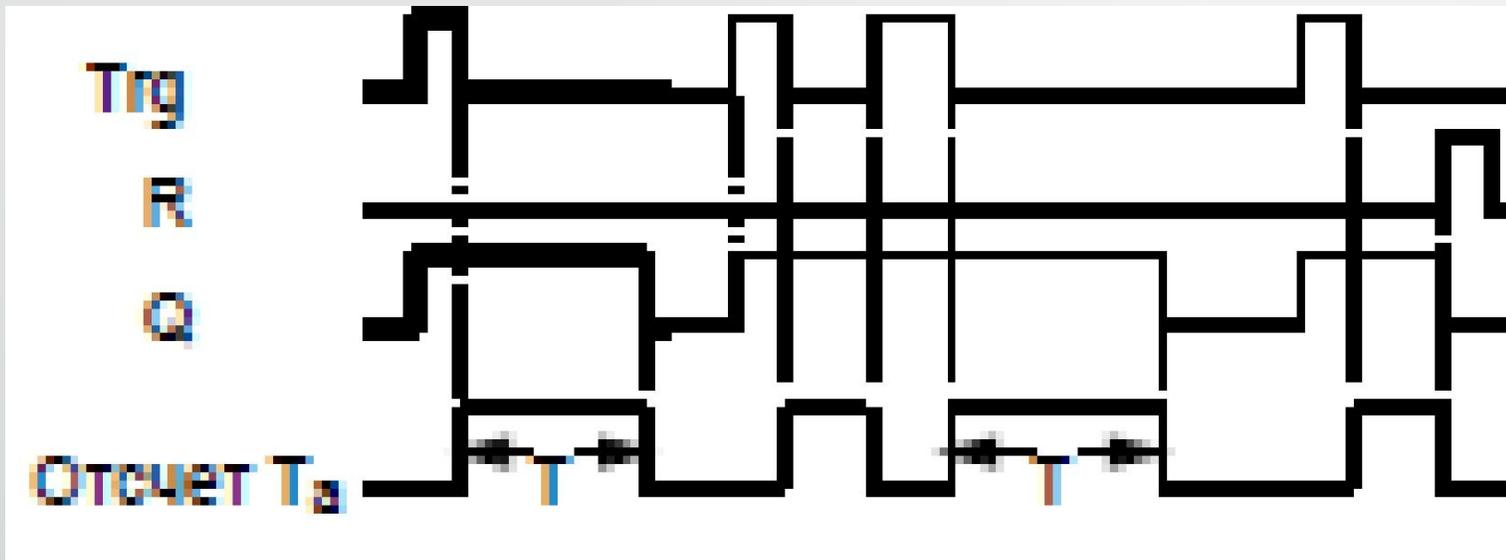
Если сохраняемость не активизирована, то выход Q и истекшее время при исчезновении питания сбрасываются.

# Задержка выключения

При задержке выключения выход сбрасывается только по истечении заданного интервала времени.

Символ в LOGO!	Подключение	Описание
	Вход Trg	Задержка выключения запускается отрицательным фронтом сигнала (изменением с 1 на 0) на входе Trg (trigger = запустить)
	Вход R	Сигнал на входе R сбрасывает время задержки выключения и устанавливает выход в 0.
	Параметр	T – это время, через которое выключается выход (выходной сигнал переключается с 1 на 0). Сохраняемость: / = Сохраняемость отсутствует R = Состояние сохраняется.
	Выход Q	Q устанавливается сигналом на входе Trg. Он сохраняет это состояние, пока не истечет время T.

## Временная диаграмма



Когда состояние сигнала на входе Trg меняется на 1, выход Q переключается на 1 немедленно.

Если состояние сигнала на входе Trg изменяется с 1 на 0, то в LOGO! снова запускается текущее время  $T_a$ , а выход остается установленным.

Если  $T_a$  достигает значения, указанного через T ( $T_a=T$ ), то выход Q сбрасывается в 0 (задержка выключения).

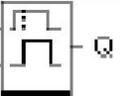
Если вход Trg включается и выключается снова, то время  $T_a$  снова запускается.

Вход R (сброс) сбрасывает время  $T_a$  и выход до того, как истечет установленная задержка времени  $T_a$ .

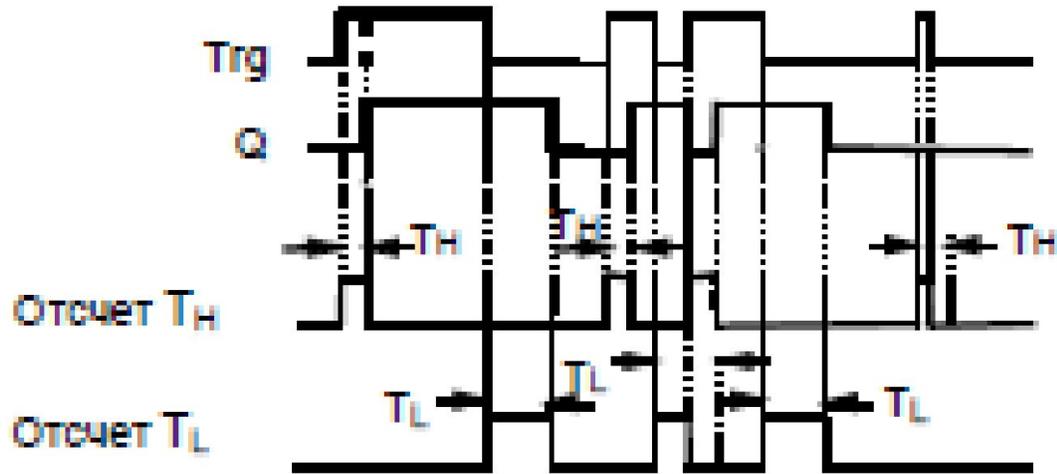
Если сохраняемость не активизирована, то выход Q и уже истекшее время при исчезновении питания сбрасываются .

# Задержка включения и выключения

При задержке включения и выключения выход устанавливается по истечении заданной задержки включения и сбрасывается по истечении заданной задержки выключения.

Символ в LOGO!	Подключение	Описание
	Вход Trg	Нарастающий фронт (изменение с 0 на 1) на входе Trg (trigger = запустить) запускает время $T_n$ для задержки включения. Падающий фронт (изменение с 1 на 0) запускает время $T_l$ для задержки выключения.
	Параметр	$T_n$ – это время, по истечении которого выход включается (выходной сигнал переключается с 0 на 1). $T_l$ – это время, по истечении которого выход выключается (выходной сигнал переключается с 1 на 0). Сохраняемость: I = Сохраняемость отсутствует R = Состояние сохраняется.
	Выход Q	Q включается по истечении заданного времени $T_n$ , если Trg еще установлен, и выключается по истечении времени $T_l$ , если Trg не будет тем временем снова установлен.

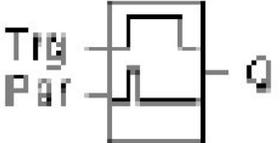
## Временная диаграмма

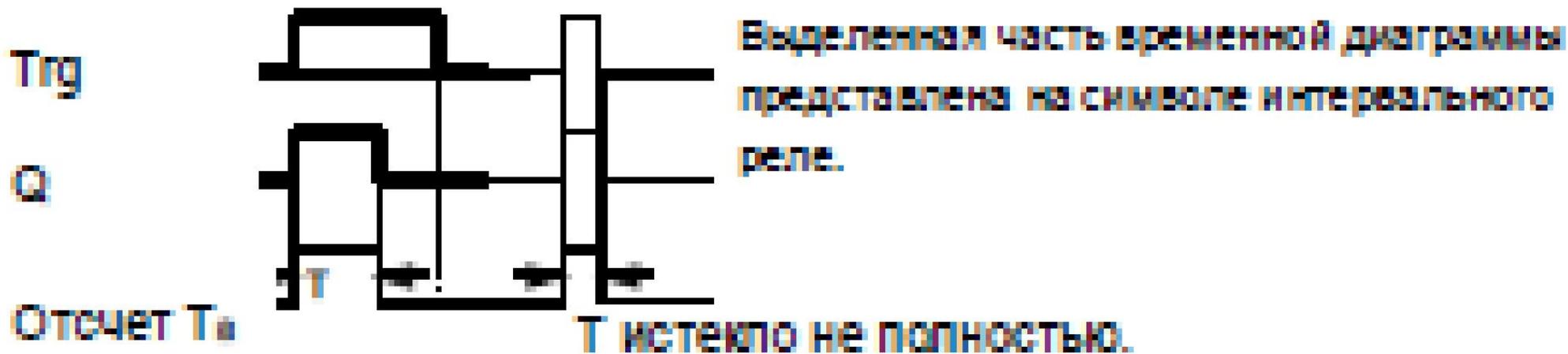


Когда состояние сигнала на входе Trg меняется с 0 на 1, начинается отсчет времени  $T_n$ . Если состояние сигнала на входе Trg остается равным 1, по крайней мере, в течение параметризованного интервала времени  $T_n$ , то по истечении времени  $T_n$  выход устанавливается в 1 (имеет место задержка между включением входа и появлением сигнала на выходе). Если состояние сигнала на входе Trg снова становится равным 0 до истечения времени  $T_n$ , то время сбрасывается. Когда состояние сигнала на входе Trg снова меняется на 0, начинается отсчет времени  $T_l$ . Если состояние сигнала на входе Trg остается равным 0, по крайней мере, в течение заданного интервала времени  $T_l$ , то по истечении времени  $T_l$  выход устанавливается в 0 (имеет место задержка между выключением входа и исчезновением сигнала на выходе). Если состояние сигнала на входе Trg меняется обратно на 1 до истечения времени  $T_l$ , то время сбрасывается. Если сохраняемость не активизирована, то выход Q и уже истекшее время при исчезновении питания сбрасываются.

# Интервальное реле (вывод импульса)

Входной импульс вызывает появление сигнала заданной длительности на выходе.

Символ в LOGO!	Подключение	Описание
	Вход Trg	Сигнал на входе Trg (trigger = запустить) запускает отсчет времени для интервального реле.
	Параметр	T – это время, через которое выключается выход (выходной сигнал переключается с 1 на 0). Сохраняемость: / = Сохраняемость отсутствует R = Состояние сохраняется.
	Выход Q	Q включается одновременно с Trg и остается включенным в течение времени Ta, если входной сигнал остается равным 1.



Когда вход Trg принимает состояние 1, то и выход Q переключается в состояние 1. Одновременно начинается отсчет времени  $T_a$ , в течение которого выход остается установленным.

Когда  $T_a$  достигает значения, заданного посредством T ( $T_a=T$ ), выход Q устанавливается в 0 (вывод импульса).

Если состояние сигнала на входе Trg изменяется обратно с 1 на 0 до истечения заданного времени, то выход тоже немедленно переключается обратно с 1 на 0.

Если сохраняемость не активизирована, то выход Q и уже истекшее время при исчезновении питания сбрасываются.

# Асинхронный генератор импульсов

Форма импульсов на выходе может быть изменена путем задания отношения длительности импульса к длительности паузы.

Символ в LOGO!

Подключение

Описание



Вход En

Вход En используется для включения и выключения асинхронного генератора импульсов.

Вход INV

Вход INV используется для инвертирования выходного сигнала активного асинхронного генератора импульсов.

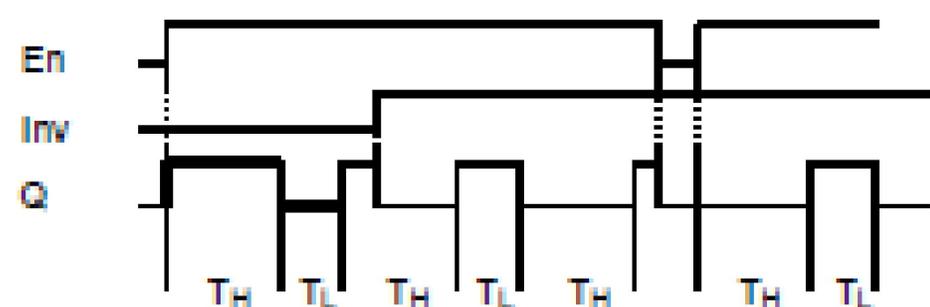
Параметр

Вы можете установить длительность импульса  $T_H$  и длительность паузы  $T_L$ .  
Сохраняемость:  
/ = Сохраняемость отсутствует  
R = Состояние сохраняется.

Выход Q

Q включается и выключается циклически в соответствии с соотношением времен  $T_H$  и  $T_L$ .

Временная диаграмма



Вы можете установить длительность импульса и паузы с помощью параметров  $T_H$  (Time High=длительность высокого уровня сигнала) и  $T_L$  (Time Low=длительность низкого уровня сигнала).

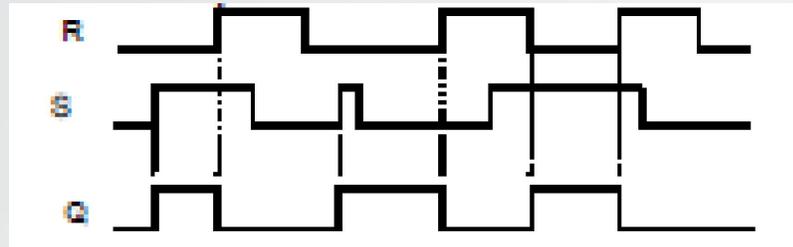
Вход INV дает возможность инвертировать выход. Вход INV инвертирует выход только тогда, когда Блок активизирован сигналом на входе EN. Если сохраняемость не активизирована, то выход Q и уже истекшее время при исчезновении питания сбрасываются.

# Самоблокирующееся реле

Вход S устанавливает выход Q, вход R снова сбрасывает выход Q.

Символ в LOGO!	Подключение	Описание
	Вход S	Выход Q устанавливается в 1 сигналом на входе S.
	Вход R	Вход R сбрасывает выход Q в 0. Если S и R равны 1 одновременно, то выход сбрасывается.
	Параметр	Сохраняемость: / = Сохраняемость отсутствует R = Состояние сохраняется.
	Выход Q	Q включается сигналом на входе S и остается включенным, пока не будет установлен вход R.

## Временная диаграмма



Самоблокирующееся реле . это простая двоичная ячейка памяти. Сигнал на выходе зависит от состояний входов и предыдущего состояния сигнала на выходе.

Следующая таблица еще раз иллюстрирует эту логику:

$S_n$	$R_n$	$Q$	Примечание
0	0	x	Состояние сохраняется
0	1	0	Сброс
1	0	1	Установка
1	1	0	Сброс (сброс имеет преимущество перед установкой)

Если свойство сохраняемости было активизировано, то текущее состояние выходного сигнала после потери питания сохраняется.