

Глава 5

Быстрые выходы

В качестве быстрых импульсных выходов в ПЛК используются цифровые выходы с адресами Q0.0 и Q0.1, работающие независимо друг от друга. В соответствии с байтом управления (таблица П1-4) ПЛК генерирует на этих выходах:

- либо бесконечную последовательность импульсов с программируемой скважностью (*pulse width modulation* – функция PWM - ШИМ);
- либо конечную последовательность импульсов со скважностью 50% (*pulse train output* – функция РТО).

Инициация выдачи последовательности импульсов осуществляется с помощью блока PLS (рис. 5-1), размещенного в папке счетных операций Counter (рис. 4-1). Значение параметра Q=0 иницирует работу импульсного выхода Q0.0, а Q=1 – Q0.1.

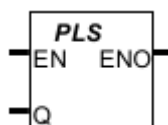


Рис. 5-1. Блок выдачи последовательности импульсов

Перед запуском блока PLS нужно задать параметры последовательности импульсов в области специальных маркеров в формате, представленном в таблице П1-4. Таких параметров три:

- 1) период следования импульсов T (рис. 5-2) - актуален для РТО и PWM;
- 2) длительность импульсов ΔT (рис. 5-2) - актуальна только для PWM;
- 3) количество импульсов (актуально только для РТО).

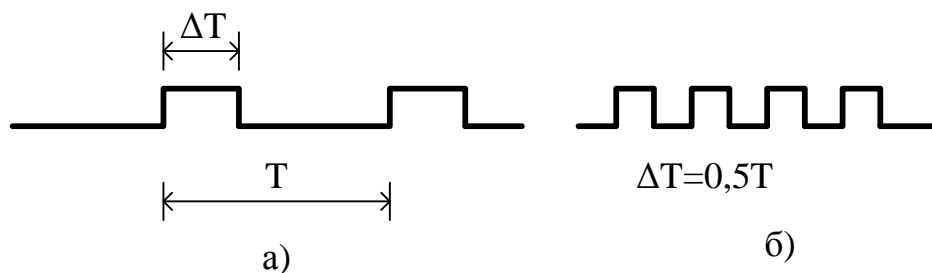


Рис. 5-2. Соотношение временных параметров последовательности импульсов: а - для функции PWM; б - для функции РТО

Функция PWM (ШИМ)

На рис. 5-3 представлен пример программирования функции PWM для выдачи последовательности импульсов с периодом следования 1 секунда и относительной длительностью включения 30%: в течение 1 секунды выход Q0.0 должен находиться 0,3 секунды в состоянии логической 1, а 0,7 секунды – в состоянии логического 0. Запуск последовательности осуществляется по переднему фронту на физическом входе I0.1, блокировка – по заднему фронту на физическом входе I0.5.

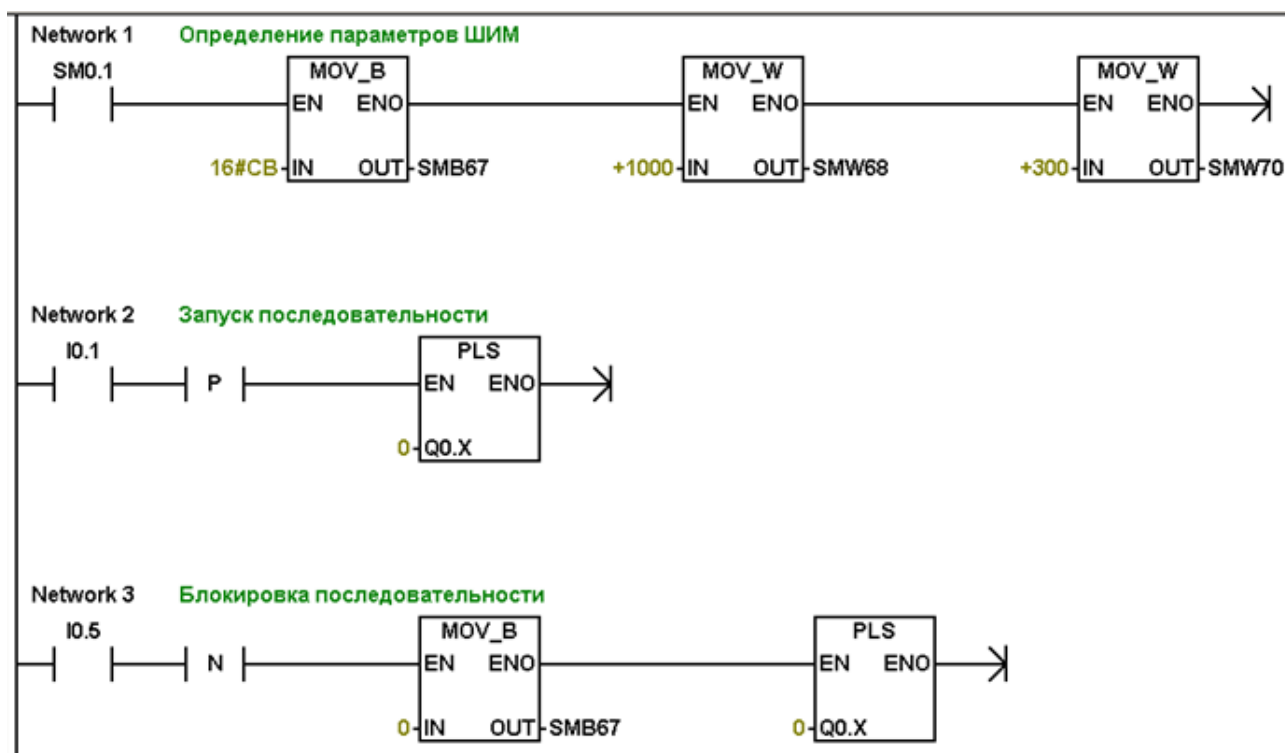


Рис. 5-3. Пример программирования функции PWM

Параметрирование данной функции включает в себя занесение в область специальных маркеров (см. формат в таблице П1-4) следующей информации:

- управляющий байт (рис. 5-4) – в SMB67;
- период следования импульсов 1000 миллисекунд - в SMW68;
- время включения 300 миллисекунд - в SMW70.

Они реализуются с помощью блоков пересылки (Network 1 на рис. 5-3).

2 # 1 1 0 0 1 0 1 1 = 16 # СВ

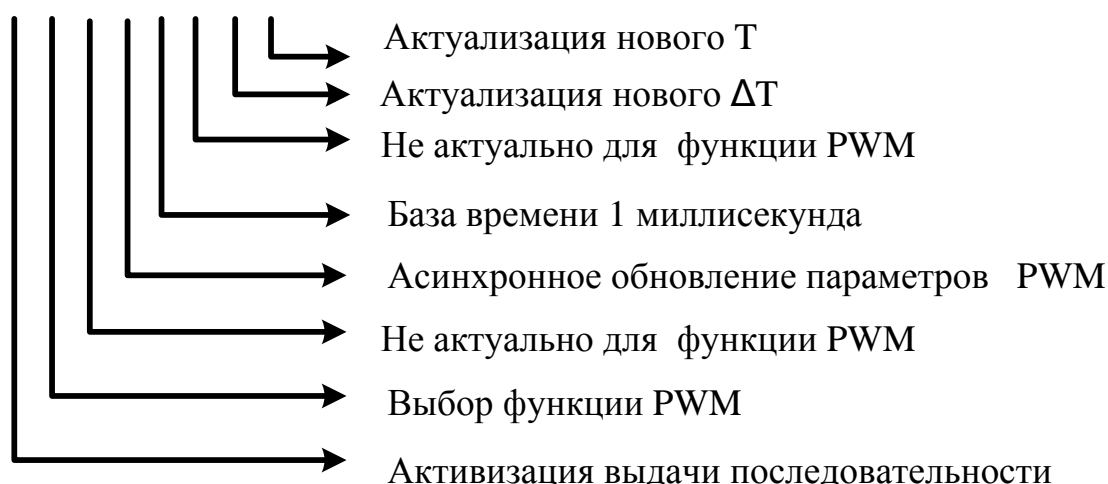


Рис. 5-4. Управляющий байт для функции PWM рассматриваемого примера.

Запуск последовательности (Network 2 на рис. 5-3) осуществляется блоком PLS, после успешного выполнения которого CPU начинает выдавать на своём физическом выходе Q0.0 бесконечную последовательность импульсов с указанными параметрами. Контакт выделения переднего фронта \uparrow^P необходим для однократного выполнения блока PLS в этой строке программы.

Блокировка выдачи последовательности (Network 3 на рис. 5-3) осуществляется также блоком PLS, но перед его вызовом осуществляется обнуление управляющего байта пересылкой нуля в область SMB67. Контакт \uparrow^N необходим для однократного выполнения блока PLS в строке программы.

Изменением параметров последовательности в области SM и перезапуском посредством блока PLS можно управлять выдачей импульсов на соответствующем физическом выходе. При этом переход от одной последовательности импульсов к другой осуществляется одним из двух способов обновления:

- синхронным обновлением (бит 4 в управляющем байте равен 1) - изменение параметров происходит на границе периода следования после команды PLS, передающей новые параметры, записанные в специальных маркерах;
- асинхронным обновлением (бит 4 в управляющем байте сброшен в 0), производящимся с кратковременной блокировкой работы импульсного выхода.

Функция РТО

Функция РТО позволяет организовать конечные последовательности импульсов с заданным периодом и половинным временем включения (относительная длительность импульсов составляет 50%, показано на рис. 5-2, б). При этом возможна реализация:

1) односегментная выдача заданного количества импульсов с неизменными параметрами. Пример программирования приведен на рис. 5-5, управляющий байт – на рис. 5-6. Здесь показан фрагмент программы в редакторе LAD, реализующий выдачу 100 импульсов с $T=1$ секунда на выход Q0.0. Запуск последовательности осуществляется по заднему фронту на I0.5. Как и для ШИМ, здесь в области специальных маркеров задан управляющий байт и параметры последовательности (количество и период следования импульсов). Инициацию выдачи осуществляет блок PLS.

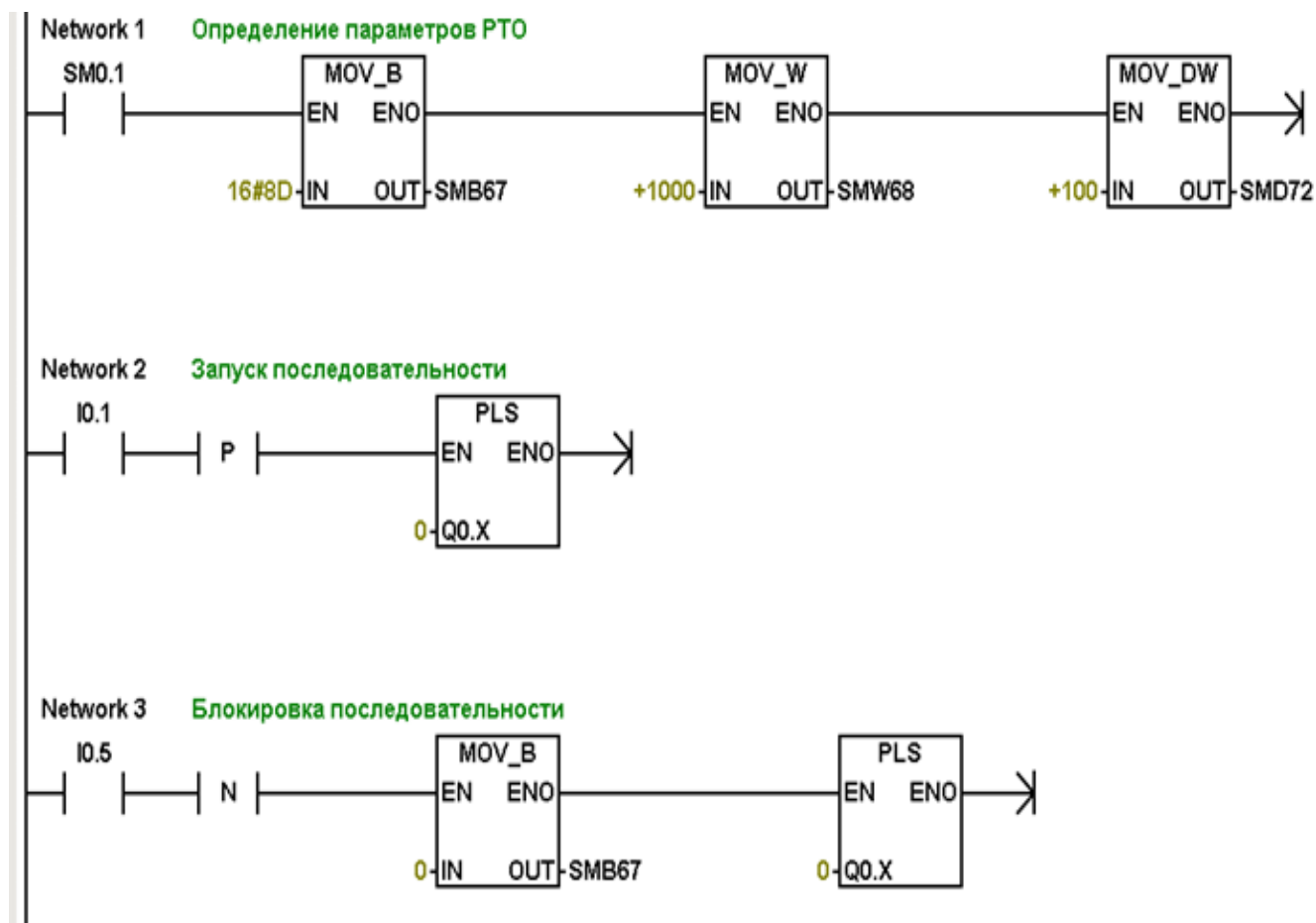


Рис. 5-5. Пример программирования функции односегментного РТО

2 # 1 0 0 0 1 1 0 1 = 16 # 8D

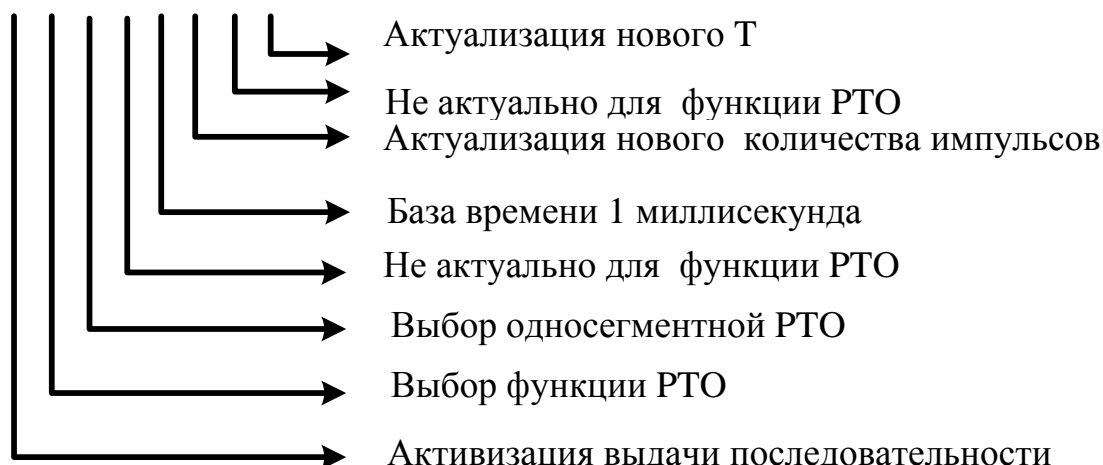


Рис. 5-6. Управляющий байт односегментной PTO.

- многосегментная выдача , реализуемая с помощью таблицы профиля, расположенной в области памяти переменных с форматом, представленным в таблице 5-1. Особенности данного способа являются (таблица П1-4):
 - база времени выбирается одной и той же для всех сегментов;
 - области SMW68 и SMW78, задающие период следования импульсов, и области SMD72 и SMD82, задающие количество импульсов, не используются;
 - в специальные маркеры SMW168 или SMW178 необходимо загрузить начальное смещение (в байтах) таблицы профиля относительно адреса VB0 (таблица П1-4);
 - возможно изменение периода следования импульсов на заданную величину, выполняемое на каждом импульсе;
 - во время выдачи импульсов можно считать количество активных в данный момент сегментов из области SMB166 или SMB176.

Подготовленная в специальных маркерах информация инициирует запрограммированную последовательность импульсов посредством функции PLS.

Таблица 5-1. Формат таблицы профиля РТО

Смещение в байтах от начала таблицы	Номер сегмента	Описание записей таблицы
0		Количество сегментов в формате байта. Значение 0 генерирует нефатальную ошибку, выход РТО не генерируется
1	1	Начальный период следования импульсов в формате слова
3		Приращение периода следования импульсов (слово)
5		Количество импульсов в формате двойного слова
9	2	Начальный период следования импульсов (слово)
11		Приращение периода следования импульсов в формате слова со знаком
13		Количество импульсов в формате двойного слова
17	3	Начальный период следования импульсов
19		И т.д.

В качестве примера рассмотрена последовательность из трёх сегментов:

- 10 импульсов с постоянным периодом 1 сек;
- 7 импульсов с нарастающим периодом (шаг 0,25 сек) от 1 секунды;
- 5 импульсов с убывающим периодом (шаг 0,5 сек) от 5 секунд.

Таблица этого профиля для приведена в таблице 5-2 для начального адреса таблицы VB0, база времени – миллисекунды, программа – на рис. 5-7, управляющий байт – на рис. 5-8. Поскольку таблица профиля располагается в области памяти переменных, её удобно задавать в блоке данных DB0 (рис. 5-9).

Таблица 5-2. Таблица профиля РТО

Адрес таблицы	Номер сегмента	Описание записей таблицы
VB0		3
VW1	1	1000
VW3		0
VD5		10
VW9	2	1000
VW11		250
VD13		7
VW17	3	5000
VW19		-500
VD21		5

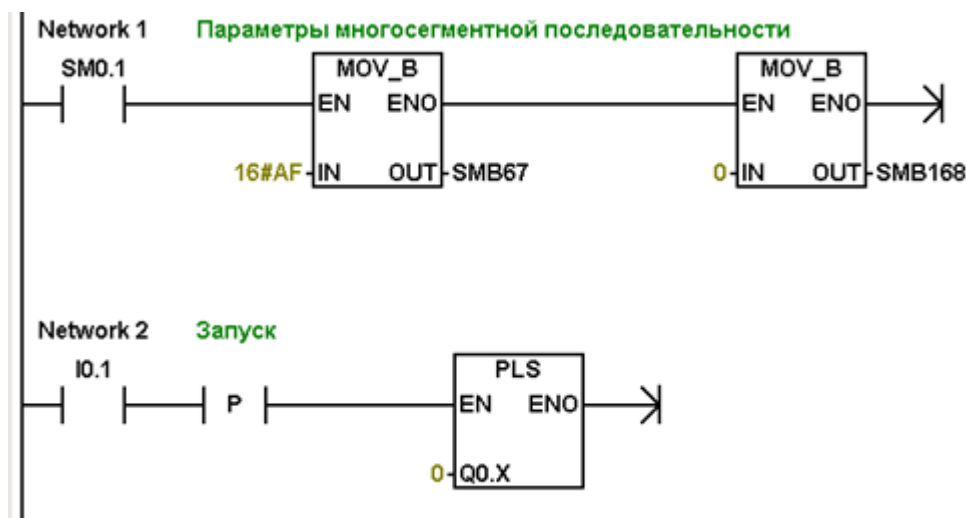
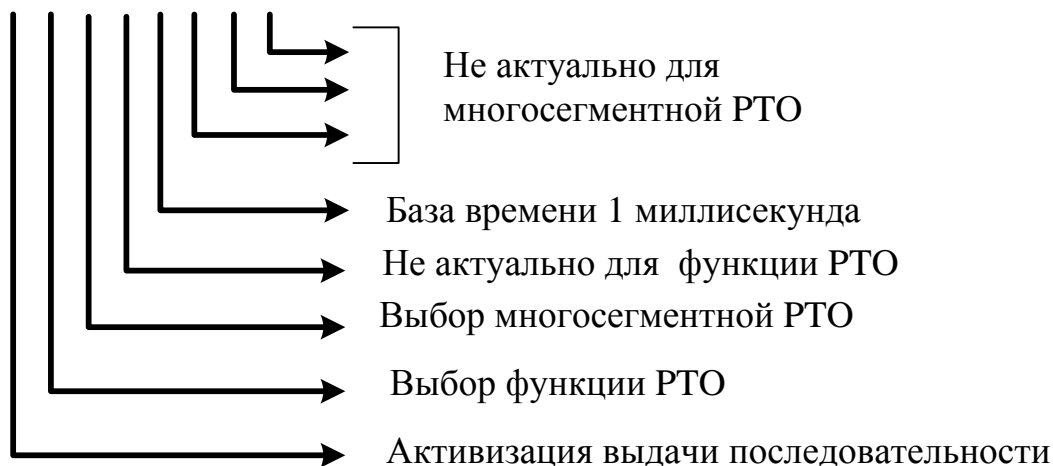


Рис. 5-7. Пример программирования функции многосегментной PTO

$$2 \# 10101000 = 16 \# A8$$



P

ис. 5-8. Управляющий байт для функции многосегментной PTO

```

Data Block
  2 3 4 5 6
  //
  //Параметры трёх последовательностей импульсов
  //
  VB0    3           // Количество сегментов
  VW1    1000, 0     //Первая последовательность
  VD5    10
  VW9    1000, 250   //Вторая последовательность
  VD13   7
  VW17   5000, -500  //Третья последовательность
  VD21   5
    
```

Рис. 5-9. Блок данных для многосегментной PTO для примера

Вопросы для самоконтроля

1. Сравните временные диаграммы выполнения функций РТО и PWM [3]: почему для РТО выбранный выход предварительно устанавливается в 0, а для PWM – в 1?
2. Почему рекомендуется программировать период следования импульсов четным числом?
3. Что такое относительная длительность включения?
4. Каково состояние быстрого выхода при 100%-ной относительной длительности включения?
5. Напишите алгоритм изменения параметров ШИМ при работающем быстром выходе.
6. Напишите алгоритм изменения периода следования импульсов для функции РТО, для работающего быстрого выхода.
7. Напишите алгоритм изменения периода следования импульсов для функции РТО, для неработающего быстрого выхода.
8. Напишите алгоритм блокировки быстрого выхода.
9. В какое состояние переключится быстрый выход Q0.0, если будет иметь место его принудительная установка в основной программе?