

Обзор специфических функций языка FBD на примере программируемых реле LOGO!

А.Н. Долидзе

Государственный университет аэрокосмического приборостроения, Санкт-Петербург

Аннотация: В статье обозначена проблема, возникающая в результате обобщённого обучения специалистов по стандарту международной электротехнической комиссии 61131-3. Стандарт регламентирует языки, на которых производится разработка программного обеспечения для программируемых логических контроллеров. Обучение по стандарту позволяет специалисту работать с контроллерами любых производителей, обратной стороной чего является недостаточно глубокое изучение дополнительных возможностей контроллеров, характерных для моделей конкретного производителя. Рассматривается ряд специфических специальных функций языка FBD на примере программируемых реле LOGO! фирмы Siemens, не характерных для других устройств, работающих с этим языком. Приводятся примеры использования этих функций в различных технологических процессах. Разбираются возможности работы с аналоговыми сигналами, в том числе аналоговые вычисления и вариант нестандартного использования функции поиска минимума/максимума в качестве запоминающего элемента. Многие разработчики игнорируют подобные функции, так как они не рассматриваются в учебных курсах, в то время как их использование упрощает разработку программ и предоставляет дополнительные возможности программисту. Таким образом, показана необходимость более глубокого изучения возможностей конкретного контроллера.

Ключевые слова: программируемые логические контроллеры, функциональные блок-диаграммы, подготовка специалистов, обработка аналоговых сигналов, редко используемые функции.

Введение

Язык Function Block Diagram (FBD) или язык функциональных блок-диаграмм, один из пяти языков, предлагаемых международной электротехнической комиссией (МЭК) в стандарте МЭК 61131-3 для реализации в программируемых логических контроллерах [1]. Производители различных контроллеров могут вводить собственные языки программирования для своей продукции, но в подавляющем большинстве случаев используется хотя бы один из языков стандарта. Так, рассматриваемое в данной статье программируемое реле Siemens LOGO! (упрощённая версия программируемого логического контроллера [2]),

поддерживает два из пяти языков стандарта – язык функциональных блочных диаграмм и язык релейной логики.

В основу стандарта вкладывалась идея создания единой школы подготовки специалистов. Человек, прошедший обучение по программе, включающей стандарт МЭК 61131, смог бы работать с контроллером любой фирмы [3].

Тем не менее, сам стандарт не регламентирует конечный набор функций реализуемых в том или ином языке. Производители вольны добавлять специфические функции для упрощения реализации конкретных задач. В результате, специалист прошедший базовую подготовку, как правило, не использует эти функции, попросту не зная об их назначении. Так многие функции остаются незаслуженно забытыми, а вместо них появляются сложные логические конструкции.

Базовые и специальные функции на примерах

В LOGO! все функции делятся на две категории: базовые и специальные [4]. Базовые функции включают в себя стандартный набор двоичной логики [5] и не представляют особого интереса. К специальным функциям относятся таймеры, счётчики, обработка аналоговых сигналов, триггеры и прочее. Именно среди специальных функций находятся специфические, характерные только для серии контроллеров LOGO!. Далее будут рассмотрены некоторые из них.

Одна из оригинальных специальных функций LOGO! – интервальное реле с запуском по фронту [6]. По переднему фронту сигнала на входе, она генерирует заданное число импульсов (от 1 до 9). Продолжительность импульсов и интервалов настраивается независимо и может составлять от миллисекунды, до ста часов. С точки зрения программирования, функция представляет собой комбинацию асинхронного генератора, таймера задержки

включения (или счётчика) и логической функции «И», т.е. применение каждой такой функции позволяет сэкономить два блока.

Пример 1. Есть система с баком, в котором перемешивается некая смесь. Смесь нужно перемешивать полчаса, чередуя направление вращения двигателя: 5 минут в одну сторону, 5 минут – в другую. После завершения цикла, смесь перемешивается в одном направлении в течение получаса.

На рис. 1 показан вариант решения данной задачи с использованием интервального реле с запуском по фронту (B001). По нажатию кнопки «пуск» начинается цикл перемешивания – выходы «вращение влево» и «вращение вправо» поочередно активируются. Когда функция обрабатывает, выход «вращение вправо» остаётся активным. Таймер задержки включения (B004), отключит «вращение вправо» через полчаса после завершения цикла.

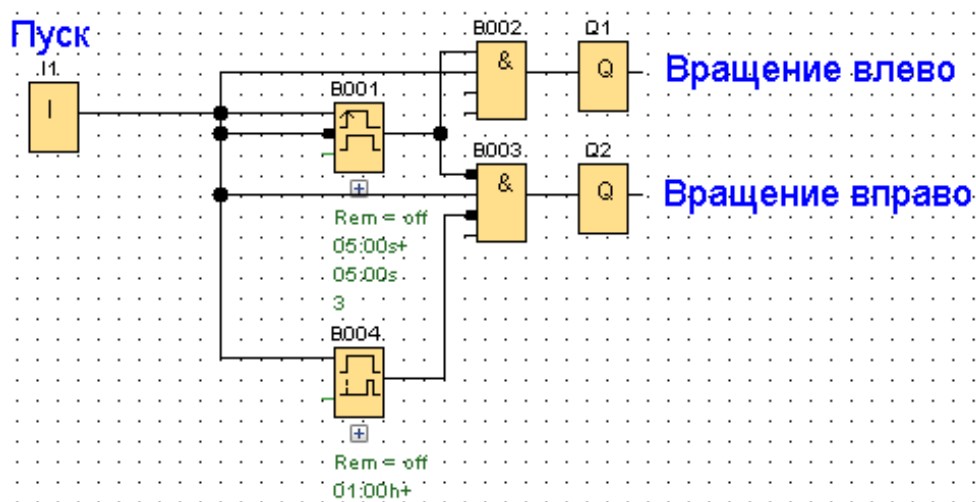


Рис. 1. – Применение интервального реле с запуском по фронту для изменения направления вращения

Пример 2. Реализация работы светофора. Прежде, чем зелёный свет погаснет, он трижды мигает, в течение трёх секунд, вариант решения этой задачи представлен на рис. 2. RS-триггер (B001) устанавливается, когда нужно зажечь зелёный свет, при его сбросе будет активировано интервальное реле с запуском по фронту (B003), которое обеспечит необходимое мигание.

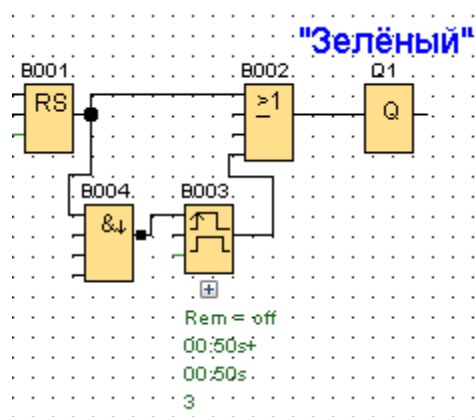


Рис. 2. – Фрагмент программы, для управления светофором

При разработке первых версий программируемых реле, LOGO! работе с аналоговыми сигналами было уделено очень мало внимания. Аналоговые сигналы можно было анализировать и сравнивать, но программист не мог проводить какие-либо вычисления на основе этих сигналов, а также у него не было доступа к таким параметрам, как текущее значение таймеров/счётчиков.

С каждой новой версией LOGO! список специальных функций расширялся, в том числе добавлялись новые функции для работы с аналоговыми сигналами [7]. Начиная с четвертого поколения LOGO! появилась возможность использовать в качестве аналогового параметра одного блока (задержки в таймерах, установки в счётчиках и т.п.) указание на значение другого блока, что значительно расширило возможности LOGO! Новая возможность получила название «Ссылка» или «Reference». Изначально это была буквально ссылка на параметр какого-то блока, в 8-й версии среды разработки для увеличения наглядности эту концепцию сменили на линию, визуальную соединяющую параметры блоков.

Пример 3. В качестве примера использования ссылки рассмотрим задачу регулирования времени выдержки в печи с помощью потенциометра. Значение с аналогового входа контроллера через пороговый элемент передаётся в качестве времени задержки на таймер (рис. 3). Альтернативой

этому решению была бы огромная схема на пороговых элементах, а регулировка параметра получилась бы ступенчатой [8].

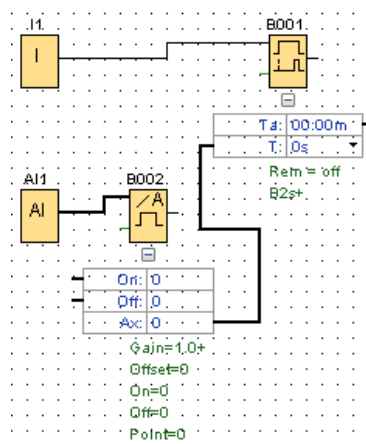


Рис. 3. – Пример использования ссылки, фрагмент программы на FBD

Одним из существенных недостатков LOGO!, считается невозможность сохранять аналоговые значения для последующего сравнения, но на самом деле можно использовать функцию поиска минимума/максимума (Min/Max) как ячейку памяти для аналогового значения. Функция Min/Max запоминает минимальное и/или максимальное значение, поступающее на её аналоговый вход в течение интервала времени пока активен её разрешающий вход (enable). Если разрешающий вход будет активирован на один программный цикл (активация по фронту), функция Min/Max запомнит текущее значение аналогового входа как минимальное/максимальное и будет хранить его до следующей активации разрешающего входа.

Пример 4. Рассмотрим производственную линию, расфасовывающую сыпучий продукт по упаковкам. Порция продукта отмеряется по весу. В конце рабочего дня нужно знать – сколько было расфасовано продукта.

Вариант решения задачи представлен на рис. 4, функция Min/Max (B001) сохраняет массу уже расфасованного продукта по переднему фронту (B002) сигнала с входа П1. Масса вычисляется с помощью функции «математическая инструкция» (B003), которая будет описана позже.

Вычисление продолжается, пока активен сигнал I2, чтобы сбросить сумму, нужно отключить I2 и подать импульс на I1.

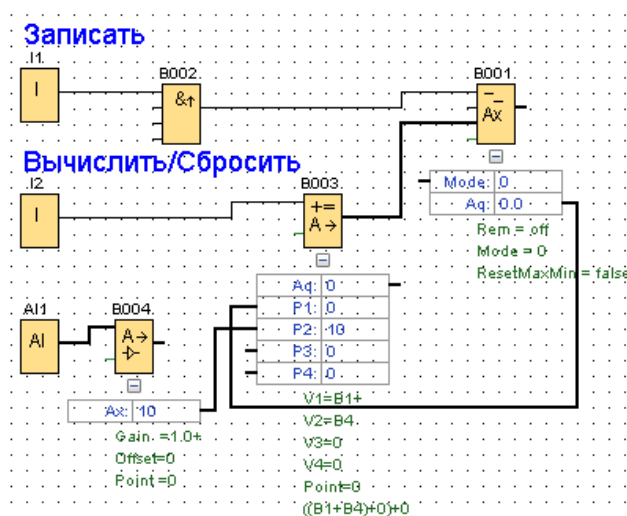


Рис. 4. – Вычисление массы расфасованного продукта

Начиная с шестой версии LOGO!, появляется функция, позволяющая выполнять математические операции с аналоговыми значениями. Доступны целочисленные операции сложения, вычитания, умножения и деления. На вход функции подаётся до четырёх аргументов (только с помощью ссылки), между которыми в произвольном порядке можно применять указанные выше операции. Приоритет операций устанавливается вручную, что означает то, что порядок выполнения может нарушать математические правила.

Пример 5. На проходной установлено четыре турникета. Необходимо подсчитать количество вошедших посетителей. Поскольку несколько турникетов могут сработать одновременно, невозможно подключить их датчики на один счетчик – несколько посетителей будут засчитаны за одного. Используем для решения этой проблемы функцию математическая инструкция как показано на рис. 5.

Функция широтно-импульсной модуляции (ШИМ, англ. pulse-width modulation (PWM)) — процесс управления мощностью методом пульсирующего включения и выключения потребителя энергии [9].

Наиболее распространённое применение ШИМ – управление яркостью освещения [10], нецелесообразно в отношении LOGO!, он не обладает достаточной скоростью переключения контактов. Максимальная частота переключения 10 Гц вне зависимости от типа выхода, сама же функция ШИМ на практике не способна поддерживать даже такую частоту. С помощью ШИМ можно управлять двигателями, но только на модулях LOGO! с транзисторными выходами. Релейные выходы быстро изнашиваются при такой нагрузке. Наиболее эффективно функция ШИМ проявит себя в управлении более медленными, инерциальными устройствами, например, нагревательными элементами, когда период работы функции может измеряться минутами.

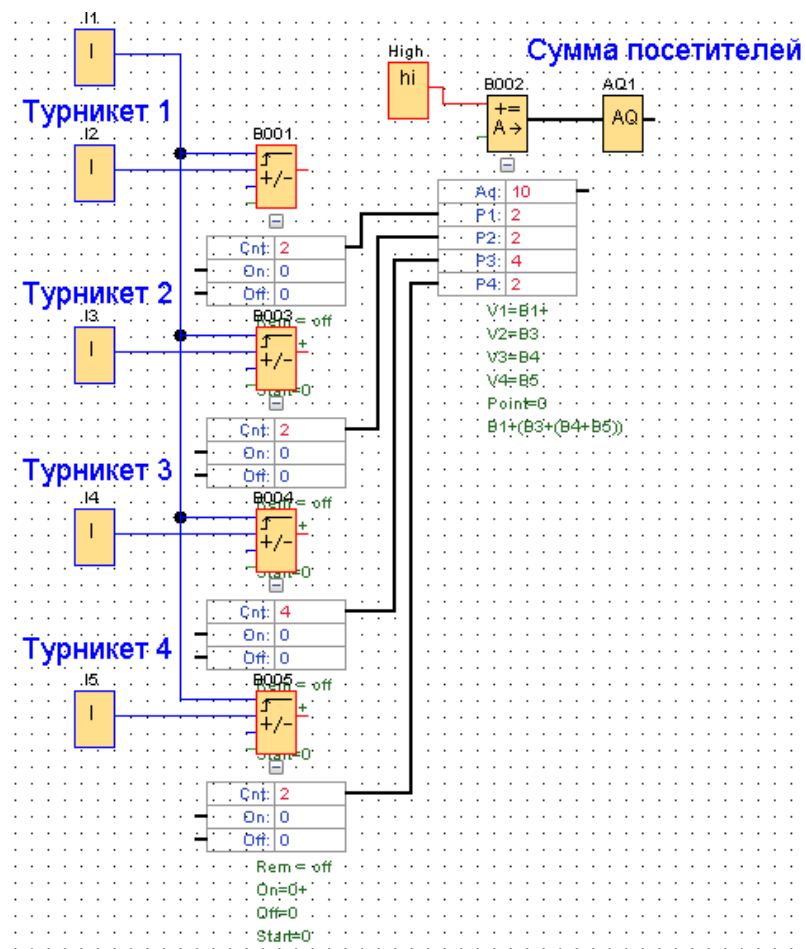


Рис. 5. – Пример использования математической инструкции: счётчик посетителей

Заключение

На примере рассмотренных функций показана возможность более эффективного использования контроллеров в управлении технологическими процессами.

Общая подготовка по стандарту IEC 61131 позволяет программисту работать с любыми контроллерами и успешно решать стоящие перед ним задачи автоматизации. Однако при этом упускаются возможности, которые могут существенно облегчить ему работу. Примеры, приведенные в статье, могут быть использованы при обучении специалистов в области автоматизации.

Литература

1. Programmable controllers – Part 3: Programming languages / International Electrotechnical Commission, International Electrotechnical Commission, 2013. 438 с.
2. Hugh Jack Automating Manufacturing Systems with PLCs. Lulu.com, 2010. 644 с.
3. Петров И.В. Програмируемые контроллеры. Стандартные языки и приёмы прикладного программирования. Москва: СОЛОН-Пресс, 2004. 256 с.
4. Siemens AG LOGO! - System Manual. Nuremberg: Siemens AG, 2022. 371 с.
5. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Теория автоматов. 2 изд. Москва: Юрайт, 2022. 321 с.
6. Миленина С. А., Миленин Н. К. Электротехника, электроника и схемотехника. 2 изд. Москва: Юрайт, 2022. 407 с.
7. Ицкович Э.Л. Тенденции развития промышленных контроллеров и их перспективные свойства // Автоматизация в промышленности. 2017. №6. С. 3-7.

8. Джеймс А. Рег, Глеин Дж. Сартори. Промышленная электроника. Москва: ДМК Пресс, 2016. 1136 с.

9. Introduction to Pulse Width Modulation (PWM) // Barr Group URL: barrgroup.com/embedded-systems/how-to/pwm-pulse-width-modulation (дата обращения: 10.11.2022).

10. Роузен Рич. Методы регулировки яркости для импульсных драйверов светодиодов // Время электроники URL: russianelectronics.ru/files/55533/EK3_27-29%20Роузен.pdf (дата обращения: 10.11.2022).

References

1. Programmable controllers – Part 3: Programming languages. International Electrotechnical Commission, International Electrotechnical Commission, 2013. 438 p.

2. Hugh Jack Automating Manufacturing Systems with PLCs. Lulu.com, 2010. 644 p.

3. Petrov I.V. Programmiruemye kontrollery. Standartnye yazyki i priyomy prikladnogo programmirovaniya [Programmable controllers. Standard languages and techniques of application programming]. Moskva: SOLON-Press, 2004. 256 p.

4. Siemens AG LOGO! - System Manual. Nuremberg: Siemens AG, 2022. 371 p.

5. Kudryavcev V. B., Aleshin S. V., Podkolzin A. S. Teoriya avtomatov [Automata theory]. 2nd ed. Moskva: Yurayt, 2022. 321 p.

6. Milenina S. A., Milenin N. K. Elektrotehnika, elektronika i skhemotekhnika [Electrical engineering, electronics and circuit engineering]. 2nd ed. Moskva: Yurayt, 2022. 407 p.

7. Ickovich E.L. Avtomatizaciya v promyshlennosti [Automation in industry]. 2017. №6. P. 3-7.



8. James A. Reg, Glenn J. Sartori. Promyshlennaya elektronika [Industrial electronics]. Moskva: DMK Press, 2016. 1136 p.

9. Introduction to Pulse Width Modulation (PWM). Barr Group. URL: barrgroup.com/embedded-systems/how-to/pwm-pulse-width-modulation (accessed: 10.11.2022).

10. Rosen Rich. Metody regulirovki yarkosti dlya impul'snyh drajverov svetodiodov. Vremya elektroniki [Methods of brightness adjustment for pulsed LED drivers. Time Electronics]. URL: russianelectronics.ru/files/55533/EK3_27-29%20Роузен.pdf (accessed: 10.11.2022).