

ГЕНЕРАЦИЯ КОДА НА ОСНОВЕ ДЕТЕРМИНИРОВАННОГО КОНЕЧНОГО АВТОМАТА

*В.С. Гительман, студент гр.8Т8Б
И.А. Тутов, старший преподаватель ОАР ИШИТР
Томский политехнический университет
E-mail: vsg16@tpu.ru*

Введение

Для программирования программируемых логических контроллеров (ПЛК) применяются языки стандарта МЭК-61131-3-2016. Данные языки программирования (ЯП) обладают как достоинствами, так и недостатками. При разработке программного обеспечения (ПО) для систем автоматизации на языках упомянутого стандарта программист может столкнуться с рядом трудностей, в особенности когда задача не является тривиальной. При этом он может выполнять несколько проектов, в рамках которых будет вынужден переключать своё внимание и забывать суть задания и идеи, реализованные в программе. Для упрощения процесса программирования предлагается использование разрабатываемого программного обеспечения, построенного на основе использования идей автоматного программирования и концепции детерминированного конечного автомата (ДКА).

Решение

С целью упрощения написания программного кода предполагается использование инструмента для создания направленных графов на основе ДКА. ДКА – конечный автомат (КА), в котором следующее состояние однозначно определяется текущим состоянием и выход зависит только от текущего состояния и текущего входа. В ДКА переход из любого состояния по любому условию возможен не более чем в одно состояние [1]. Конечным автоматом называют дискретное устройство, которое имеет один вход и один выход и в каждый момент времени находится в одном состоянии из множества возможных. На основе данных понятий разрабатывается программный инструмент, который позволит программисту не только более эффективно возобновлять в памяти программируемые раннее процессы, но и в рамках локальных тривиальных задач генерировать код на одном из выбранных текстовых языках программирования. Кодогенератор существенно сократит время на написание программного обеспечения для автоматизированных систем управления (АСУ), а в ряде случаев и вовсе может заменить программиста.

Языки МЭК 61131-3-2016 довольно распространены и программирование с их использованием стандартизировано, хотя некоторые возможности языков данного стандарта ограничены, в отличие от компьютерных языков высокого уровня. Это позволяет повысить безопасность разрабатываемых систем автоматического управления. При этом стоит отметить, что разработка кода на данных языках обладает рядом недостатков. Например, программирование вручную на языке ST является трудоемким и монотонным занятием, не обладает наглядностью для заказчика разрабатываемой системы. В свою очередь, язык FBD имеет недостатки, заключающиеся в громоздкости создаваемых схем, а также сложностью внесения изменений в последовательную логику работы алгоритма. Язык LD при возрастании сложности алгоритма значительно затрудняет понимание программы. Если рассматривать язык IL, то данный ЯП устарел и был исключен в третьей редакции (IEC 61131-3:2013). Такой язык, как SFC не является самостоятельным ЯП, а служит лишь для организации кода, созданного на языках стандарта МЭК 61131-3-2016. При этом каждый из упомянутых языков не исключает возможность допущения ошибок при написании ПО для автоматизированных систем управления.

Таким образом, на основе анализа возможностей языков программирования ПЛК, можно сделать вывод о необходимости применения другого подхода при организации программ для АСУ. Для упрощения процессов разработки ПО для систем автоматизации возможно применение автоматной парадигмы. Использование данной методологии позволит отобразить работу оборудования графически, что сделает более эффективным поддержание кода. Стоит отметить, что без использования диаграмм описание систем со сложным поведением может привести к трудностям на всех этапах жизненного цикла [2]. Графическое отображение процессов позволит программисту на время забыть особенности разрабатываемого ПО для конкретной системы и при необходимости быстрее восстановить их в памяти.

Ещё одним преимуществом генерации программ для систем автоматики на основе автоматной парадигмы заключается в упрощении взаимодействия технолога (заказчика) и программиста на этапе согласования технического задания (ТЗ). Диаграммы позволяют более точно отразить тот или иной технологический процесс предприятия так, чтобы программист, не вникая в особенности предприятия, смог понять принципы работы оборудования на уровне графической схемы и генерируемого на её основе программного кода. Это значительно ускорит процесс согласования технического задания. Более того, технолог (заказчик), обладая минимальными навыками программирования, сможет на уровне диаграмм, а не на уровне кода, объяснить программисту сущность требуемого для разработки программного обеспечения.

С использованием идей, представленных ранее, разрабатывается инструмент для упрощения разработки программного обеспечения. В данном ПО есть такие объекты, как состояние и переходы между ними. На их основе можно графически отобразить работу оборудования на предприятии или алгоритм контроля того или иного технологического параметра. Для примера демонстрации работы ПО выбран алгоритм программы для управления параметрами трубчатой печи, представленный на рисунке 1. Данный алгоритм подробно описан в источнике [3]. и реализован с помощью разработанного программного инструмента в виде направленного графа: автомата Мура (Рис. 1)

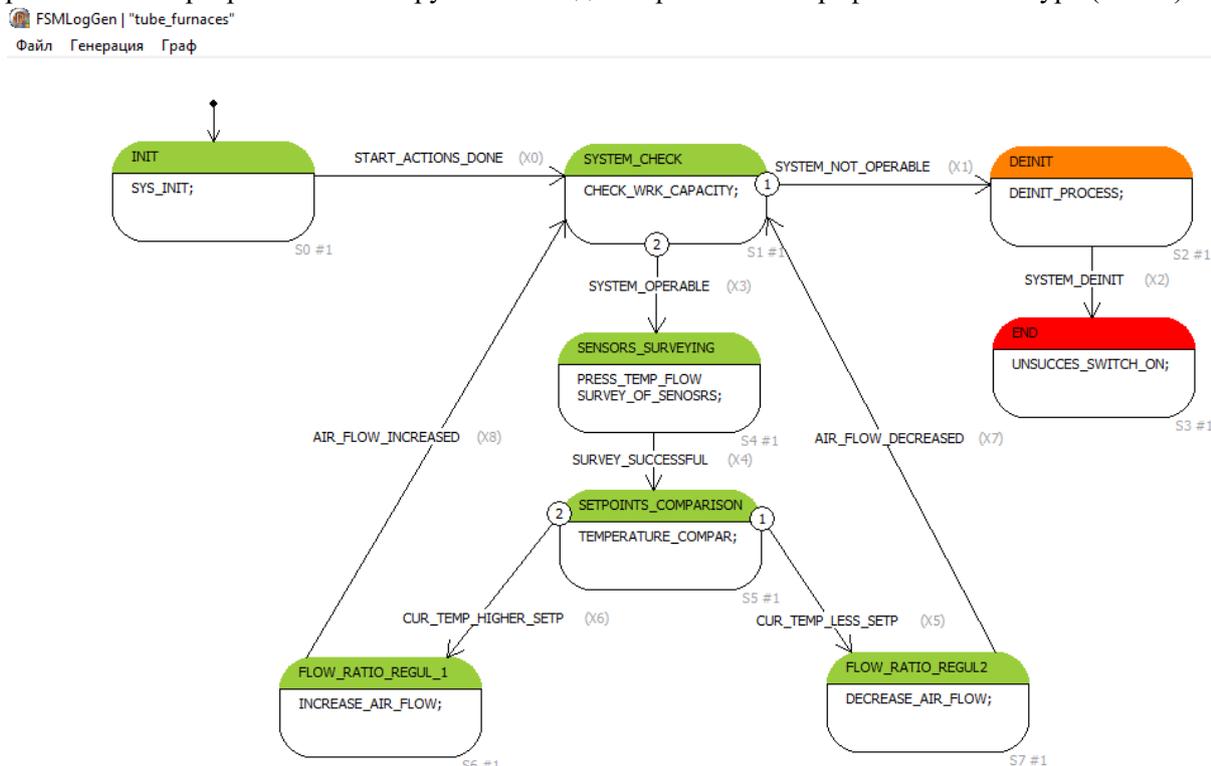


Рис. 1. Граф, созданный с использованием автоматной парадигмы в разрабатываемом ПО.

На основе графа из рисунка 1 можно сгенерировать код на языке ST. Для генерации кода выбран именно данный язык, поскольку он имеет возможность переноса кода из одной среды программирования ПЛК в другую и возможность создания программ ST в любом текстовом редакторе. При этом ST является языком программирования стандарта МЭК 61131-3-2016.

Заключение

Таким образом, в разработанном программном обеспечении реализованы функции построения диаграмм, отображающих в наглядном виде алгоритмы работы оборудования. При этом можно использовать генерацию кода на основе заданной графической схемы и получить готовый для загрузки в ПЛК программный код на языке ST.

Список использованных источников

1. Тутов, И. А. Кодогенератор графического представления детерминированного конечного автомата для систем автоматики / И. А. Тутов, В. С. Гительман, О. Б. Воскобойникова // Известия Тульского государственного университета. Технические науки. – 2021. – № 12. – С. 474-480.
2. Шальто А.А. Парадигма автоматного программирования // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2008. №53. С. 6-7.
3. Масленникова, С. В. ДонНТУ / С. В. Масленникова. — Текст : электронный // Исследование и разработка САУ температуры трубчатой печи типа ПТБ -10Э в условиях Лисичанского НПЗ : [сайт]. — URL: <http://masters.donntu.org/2015/fkita/maslennikova/diss/index.htm> (дата обращения: 25.02.2022).