

Программирование логических контроллеров

Олег Вальпа

В статье приведено описание среды разработки программ для программируемых логических контроллеров на основе конкретного практического примера, разработанного автором статьи.



Дополнительные материалы к этой статье можно скачать, перейдя по ссылке в QR-коде

Введение

В настоящее время огромную роль в автоматизации технологических процессов выполняют программируемые логические контроллеры (ПЛК), которые представляют собой процессорное устройство с памятью, портами ввода-вывода, интерфейсами и органами управления. Они могут работать автономно или в составе системы управления.

С целью облегчения разработки программ для ПЛК была создана международная система стандартизации МЭК 61131-3, описывающая языки программирования для контроллеров. ПЛК, поддерживающие данный стандарт с помощью специальной среды разработки, можно программировать без изучения логической архитектуры и электрической схемы самого контрол-



Рис. 1. Внешний вид ПЛК Pixel

лера. Стандарт МЭК 61131-3 устанавливает пять языков программирования.

- Список инструкций (IL) (Instruction List). Текстовый язык, похожий на ассемблер.
- Структурированный текст (ST) (Structured Text). Текстовый язык, похожий на язык высокого уровня Паскаль.
- Релейно-контактные схемы, или релейные диаграммы (LD) (Ladder Diagram). Графический язык, программная реализация электрических схем на базе электромагнитных реле.
- Диаграммы функциональных блоков (FBD) (Function Block Diagram). Графический язык, программа создаётся путём соединения множества функциональных блоков.
- Последовательные функциональные схемы (SFC) (Sequential Function Chart). Графический высокоуровне-

вый язык создан на базе математического аппарата сетей Петри, описывает последовательность состояний и условий переходов.

Основной целью стандарта является повышение скорости и качества разработки программ для ПЛК, а также создание языков программирования, ориентированных на технологов. Поддержка средой разработки даже одного из этих языков программирования для ПЛК относит его к классу свободно программируемых.

Программируемый логический контроллер

Рассмотрим один из широко распространённых и недорогих ПЛК под названием Pixel отечественной компании Segnetics [1]. Его внешний вид представлен на рис. 1.



Рис. 2. Расположение органов ПЛК Pixel

Contra Property Charles Contra Charles	nacipolitar una	Par Lionou	10															
D· 🗃 🖬 👌 🚵 🤊 🤊 🛤	a 🐨 😐 🛱	HI	5 ?	A	118	8 0	1	:01	E	*	3	•						
FBD	? # ×	Проект	1		-				_									4.0
в 📛 Логика																		
🖂 — Мультиплексоры, Дешифраторы и Ден	ультиплексоры	1.1																
🖂 Триггеры, Регистры и Счетчики																		
Математика																		
Операции отношения																		
CMP (int)																		
Z CMP (long)																		
CMP (real)																		
B MIN MAX (int)		1.1																
A MIN MAX (long)																		
MIN MAX (real)																		
Преобразования типов																		
Управление																		
С Генераторы, Тайнеры, Задержки		1.1																
🔁 Вреня и Дата																		
о опоки вывода на экран																		
опокон сохранения																		
диагностика																		
- Broad		1.1																
- Caberrold		1.1																
🖬 UI 🖉 Проводник 🔝 Устройства 📆	Macros FBD																	
Hasuraton	7 B X																	
		1.1																
		1.1																
		4																
		-	1									4						-

Рис. 3. Главное окно разработки





Рис. 4. Окно выбора типа ПЛК

Данный ПЛК имеет графический монохромный дисплей, 5 кнопок управления, два светодиодных индикатора, 5 входов DI, 2 релейных выхода DO, 5 аналоговых входов AI, 2 аналоговых выхода AO и интерфейс System Bus. Опционально в него может быть установлен сетевой модуль интерфейса Ethernet, модуль энергонезависимой памяти и литий-ионный элемент питания для внутренних часов. Расположение органов управления и индикации контроллера показано на рис. 2.

Интерфейс System Bus позволяет подключить к ПЛК несколько модулей расширения с целью увеличения количества портов ввода-вывода.

Среда разработки

Для программирования ПЛК его производителем создана свободно предоставляемая среда разработки SMLogix [2]. Данная среда позволяет разрабаты-

Рис. 5. Окно свойств контроллера

вать программы на языке FBD, т.е. путём применения различных логических и математических блоков, соединяемых между собой связями.

Рассмотрим конкретный пример разработки такой программы, которая предназначена для автономной первичной проверки ПЛК и может быть многократно использована для ручной отладки оборудования автоматики. Программа должна предоставить оператору возможность визуального контроля состояния входных портов и ручного управления выходными портами.

Процесс разработки начинается с обычной установки среды разработки на персональный компьютер. После установки и запуска среды разработки на экране монитора компьютера появится главное окно разработки, показанное на рис. 3.

Большое свободное поле справа предназначено для размещения блочных



Рис. 6. Окно состава контроллера

элементов из библиотеки, расположенной в левом поле. Поле навигатора предназначено для перемещения окна просмотра создаваемой схемы. В верхней области среды разработки находятся меню команд и программные кнопки инструментов среды разработки.

Для начала разработки необходимо выбрать закладку «Устройства» под полем блочных элементов библиотеки и затем дважды кликнуть по строке «Устройства».

При этом откроется окно выбора типа ПЛК, показанное на рис. 4.

Выбрав тип Pixel25xx, нужно нажать кнопку «Готово». После чего откроется окно свойств контроллера, показанное на рис. 5.



Рис. 7. Окно схемы проекта



Рис. 8. Окно инструмента SMArt

Здесь задаются общие и сетевые параметры, коды защиты и модули расширения. В нашем случае можно пропустить все настройки и нажать кнопку «ОК», после чего в левом поле появится список портов контроллера, его клавиатура и индикаторы, как показано на рис. 6.

Теперь можно разместить на главном поле схемы кнопки F1 и F2 клавиатуры, два индикатора и по одному типу портов, как показано на рис. 7. Логические выводы имеют красный цвет, а целочисленные выводы – зелёный. Для соединения разнотипных выводов применён преобразователь типов «bool –> int».

Недостающие элементы схемы для отображения значения портов на дисплее контроллера создаются с помощью встроенного программного инструмента SMArt на панели инструментов. После нажатия этой кнопки откроется окно, показанное на рис. 8.

Здесь с помощью элементов «Текст» вводятся названия портов на экране 0. Размещаемый после порта DI и DO элемент «Текст» должен иметь две записи в списке строк: 0 и 1. Для портов AI0 и AI05 добавляется элемент вывода чис-



Рис. 9. Окно списка элементов экрана 0

ла в целочисленном формате с количеством 6 знакомест. Наконец, для порта AO0 добавляется элемент ввода числа в целочисленном формате с количеством 4 знакомест. Данные элементы будут отображаться на дисплее контроллера и позволят вводить с клавиатуры значение сигнала для выходного порта AO0.

Теперь окно инструмента SMArt можно закрыть или оставить открытым и переключиться на главное окно среды разработки.

После проделанных операций в закладке UI появится окно списка элементов экрана 0 дисплея контроллера, показанное на рис. 9.

Элементы данного списка можно поместить на поле схемы и подключить к ним связи, показанные на рис. 7. Связи бывают двух типов: обычная и именованная. Выбор типа связи произволен и осуществляется соответствующей кнопкой на панели инструментов. Удобство именованной связи состоит в том, что с её помощью можно связать множество элементов в разных участках схемы без загромождения линиями связи.

Загрузка проекта

Перед загрузкой программы в контроллер необходимо подключить его порт RS-485 к компьютеру. Поскольку компьютер не имеет порта RS-485, используется широко распространённый, недорогой конвертер USB–RS-485, например, показанный на рис. 10.

Кроме того, контроллер необходимо подключить к источнику питания постоянного тока 24 В. Теперь программу проекта можно загрузить в контроллер и приступить к его проверке. Для этого необходимо нажать кнопку «Подключиться с отладкой» на панели инструментов или F5 на клавиатуре компьютера. При этом проект автоматически транслируется в код загрузочной программы, записывается в контроллер и стартует. На экране компьютера и дисплее контроллера будет отображаться состояние портов и значения сигналов.

Замыкая токопроводящей перемычкой вход DI0 контроллера с цепью INT, можно визуально проверить его работу по изменению сигнала на экранах монитора и контроллера. Поскольку сигнал с порта DI0 подключён в схеме проекта к выходному порту DO0, последний также будет переключать внутреннее реле контроллера. Эти переключения сопровождаются характерными звуковыми щелчками.

Замыкание и размыкание контактов реле можно проверить с помощью омметра на выводах порта DO1 контроллера.

Для проверки входа AI0 можно использовать резистивный потенциометр на 3 кОм, который будет имитировать датчик температуры РТ1000. Сопротивление в 1 кОм будет соответствовать температуре датчика 0°С.

Вход AI5 проверяется таким же потенциометром путём подключения его крайних выводов к контактам 10v и 0v, а среднего вывода – к входу AI5. Изменение положения движка потенциометра будет приводить к изменению измеряемой величины напряжения на входе порта.

Числовые значения для аналоговых входов AI внутри контроллера вычисляются по формуле: КОД_AI = (Uai/10 B)×



Рис. 10. Внешний вид конвертера USB-RS-485

×1023×30. Эти значения будут изменяться от 0 до 30 690.

Выходной порт AO1 проверяется путём ввода числового значения от 0 до 1023 в поле ввода на дисплее контроллера с помощью его клавиатуры. При этом выходной сигнал AO будет изменяться от 0 до 10 В и вычисляется по формуле: *U*ao = (KOД_AO/1023)×10 В.

Готовый файл проекта размещён на сайте журнала [3] и может быть использован для дальнейшего развития.

Кроме контроллера Pixel описанная здесь среда разработки позволяет программировать контроллеры: Matrix, Trim, SMH и др. с цветными графическими дисплеями. Благодаря такому замечательному инструменту процесс разработки программ для ПЛК превращается в интересное, творческое занятие и позволяет создавать самые разнообразные программы автоматизации.

Дополнительно к SMLogix можно установить программу SMConstructor [4], которая позволяет быстро создавать программы для управления тепловыми пунктами, вентиляционными системами, насосными станциями и др. системами. В программе SMConstructor выбираются необходимые агрегаты с датчиками, и конструктор автоматически генерирует FBD программу. При необходимости полученный код можно изменить или дополнить в SMLogix.

Литература

- 1. URL: https://segnetics.com/ru/pixel.
- URL: https://dl.segnetics.com/PRODUCTS/SM-Logix/Version_3.35/Lite_Pack/.
- 3. URL: https://www.cta.ru.
- 4. URL: https://dl.segnetics.com/PRODUCTS/.

ХХ Интернет-семинар

по управлению проектами профессора Gerald Cockrell

2 октября 2024 г. в демонстрационном зале НИТ УЦР возобновил работу очередной поток слушателей интернет-семинара по управлению проектами, который уже в 20-й раз проводит для студентов, аспирантов и преподавателей ГУАП Почётный доктор ГУАП Gerald Cockrell (президент ISA 2008 года). За эти годы более 520 слушателей приняли участие в работе семинара. Занятия проводятся на английском языке и продлятся до мая 2025 года.

Слушатели, успешно завершившие курс, получат сертификаты. ●