

АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

ЛЕКЦИЯ 15

4. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РСУ

21 Системы программирования ПЛК на стандартных языках [4]

Упростить создание программ для ПЛК и обеспечить соответствие ПЛК идеологии открытых систем позволяет стандарт МЭК 61131-3. Он, по мнению разработчиков этого стандарта, позволяет учесть все особенности задач для ПЛК путем настройки нескольких универсальных программ на выполнение конкретной задачи, тем самым ориентируя языки программирования на технологов. Основными целями стандарта МЭК 61131-3 являются:

- 1) повышение скорости и качества разработки программ для ПЛК,
- 2) создание языков программирования, ориентированных на технологов,
- 3) обеспечение соответствия ПЛК идеологии открытых систем,
- 4) исключение этапа дополнительного обучения при смене типа ПЛК.

Системы программирования, основанные на МЭК 61131-3 характеризуются:

- 1) надежностью создаваемого программного обеспечения (ПО). Это обеспечивается средой разработки, содержащей средства для написания, тестирования и отладки программ с помощью эмуляторов и реальных ПЛК, а также множества готовых фрагментов программного кода;
- 2) возможностью простой модификации программы и наращивания ее функциональности;
- 3) переносимостью проекта с одного ПЛК на другой;
- 4) возможностью повторного использования отработанных фрагментов программы;
- 5) простотой языка и ограничением количества его элементов.

Стандарт МЭК 61131-3 устанавливает пять языков программирования: три графических и два текстовых со следующими названиями:

- 1) ST (Structured Text) – структурированный текст;
- 2) SFC (Sequential Function Chart) – последовательные функциональные схемы;
- 3) FBD (Function Blok Diagram) – диаграммы функциональных блоков;
- 4) LD (Ladder Diagram) – релейно-контакторные схемы;
- 5) IL (Instruction List) – список инструкций.

Выбор одного из пяти языков определяется не только предпочтениями пользователя, но и смыслом решаемой задачи. Если исходная задача формулируется в терминах последовательной обработки и передачи сигналов, то для нее проще и нагляднее использовать язык FBD (диаграммы функциональных блоков). Если задача описывается как последовательность срабатываний некоторых ключей и реле, то для нее нагляднее всего будет язык LD (релейно-контакторные схемы). Для задач, которые изначально формулируются в виде сложного разветвленного алгоритма, удобнее будет язык ST (структурированный текст).

Языки МЭК 61131-3 базируются на следующих принципах:

- 1) вся программа разбивается на множество функциональных элементов (Program Organization Units (POU)), каждый из которых может состоять из функций, функциональных блоков и программ;
- 2) стандарт требует строгой типизации данных;
- 3) имеются средства для исполнения разных фрагментов программы в разное время, с разной скоростью, а также параллельно. Например, можно сканировать один датчик с частотой 10 раз в секунду, другой – один раз в секунду;
- 4) для выполнения определенной последовательности операций в заданный момент времени или наступления события используется язык SFC (последовательные функциональные схемы);
- 5) стандарт поддерживает структуры для описания разнородных данных. Например, аналоговые и дискретные сигналы данного технологического агрегата можно описать с помощью единой структуры с названием и передавать ее внутри программы как единый элемент;
- 6) в данной программе можно использовать все пять языков, поэтому для каждого фрагмента программы может быть выбран любой язык;
- 7) программа, написанная для одного контроллера, может быть перенесена на любой другой контроллер, совместимый со стандартом МЭК 61131-3.

Пример программы на языке LD (релейно-контакторные схемы) показан на рисунке 18.1.

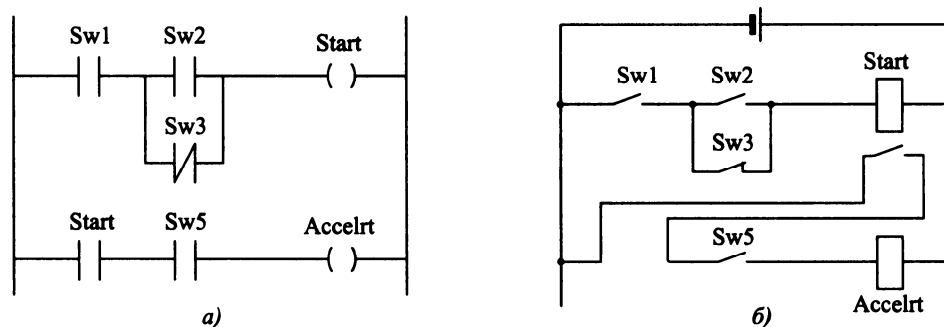


Рисунок 18.1 – Пример программы на языке LD (релейно-контакторные схемы) (а) и ее эквивалент в виде электрической схемы (б)

Пример программы на языке FBD (диаграммы функциональных блоков) показан на рисунке 18.2.

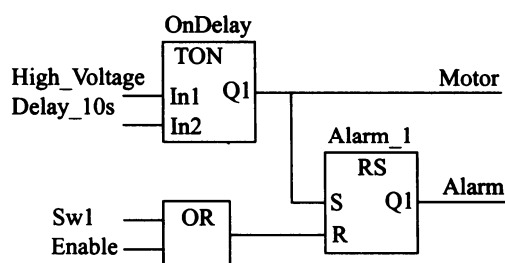


Рисунок 18.2 – Пример программы на языке FBD (диаграммы функциональных блоков)

Описанию и применению функциональных блоков посвящены, помимо МЭК 61131-3 еще и стандарты МЭК 61499 и МЭК 61804.

Стандарт МЭК 61499 устанавливает обобщенную архитектуру функциональных блоков в PCSU, где ПО распределено между несколькими физическими устройствами (ПЛК) и

несколькими функциональными блоками (ФБ), причем промышленная сеть рассматривается как составная часть системы.

Стандарт МЭК 61804 содержит детализацию требований к РСУ, построенной на основе ФБ. Стандарт определяет минимальный набор ФБ. Набор состоит из сложных ФБ (ПИД-регулятор, селектор, таймер, интегратор) и простых ФБ (тригонометрические функции, модуль, суммирование, усреднение и др.).

Более подробное описание языков программирования ПЛК можно найти в литературе [Денисенко] и др.

22 SCADA-пакеты

SCADA (Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных). В любой АСУТП важную роль играет выполнение информационных функций и ручное управление. ПО для выполнения этих функций составляют значительную часть прикладного ПО. Как раз SCADA-пакеты предназначены прежде всего для упрощения программирования задач сбора, обработки, архивирования и документирования информации, а также представления оператору информации и возможности управления в удобном виде. На практике SCADA-пакеты включают в себя широкий набор функциональных возможностей, не ограниченный только диспетчерским управлением и сбором данных. Они часто охватывают автоматическое управление и другие функции. Эти функции можно разделить на несколько групп:

- 1) настройка SCADA на конкретную задачу, т.е. разработка программой части системы автоматизации;
- 2) диспетчерское управление;
- 3) автоматическое управление;
- 4) хранение истории процессов;
- 5) выполнение функций безопасности;
- 6) выполнение общесистемных функций.

Основным отличием SCADA является наличие интерфейса с пользователем в удобном виде, уменьшающем ошибки оператора: комфорт рабочего места, понятность интерфейса, наличие подсказок, блокировка явно ошибочных действий оператора и др.

В SCADA используют понятия события и аларма. *Событие* – это изменение некоторых состояний в системе. Примеры: включение двигателя привода конвейера, достижение уровня в бункере заданного значения, появление нового оператора и т.п. События информируют оператора о состоянии системы и не требуют от него каких-либо действий.

Аларм (сигнал тревоги) – предупреждение о важном событии, в ответ на которое нужно предпринять некоторые действия. Примеры – достижение температуры опасного уровня, отказ одного из узлов поточно-транспортной системы, превышение загазованности рабочего места допустимого предела, несанкционированное открывание дверей и т.п.

Алармы делятся на подтвержденные и неподтвержденные. Аларм подтвержденный, если в ответ на его появление оператор ввел команду подтверждения, до этого момента аларм неподтвержденный. Алармы делятся на аналоговые и дискретные. Дискретный аларм появляется при изменении значения дискретной переменной, аналоговый – при вхождении аналоговой переменной в заданный интервал своих значений. Идея формирования алармов для аналоговой переменной и пример назначения интервалов показаны на рисунке 19.1.

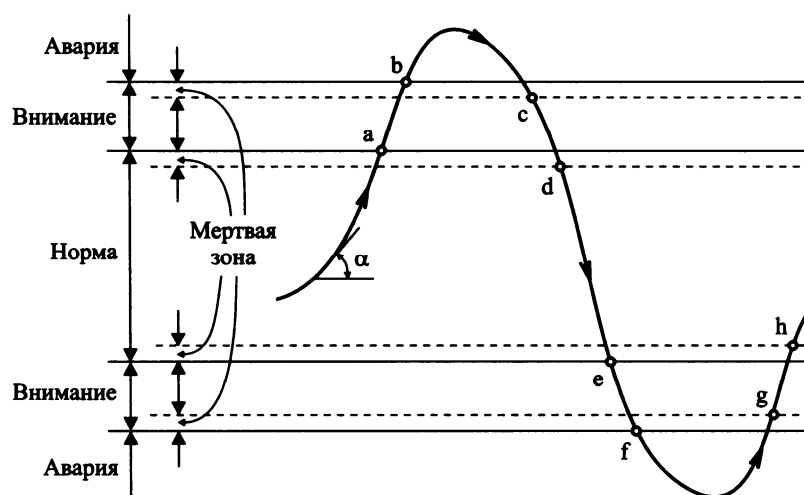


Рисунок 19.1 – Пример назначения интервалов аналоговым алармам

Как видно, здесь используются алармы двух видов: предупреждение (внимание) и авария. Методика выдачи алармов должна быть надежной. Окна с сообщениями должны появляться поверх остальных окон, алармы могут дублироваться звуком и светом. Множеству алармов назначают разные приоритеты, отличающиеся видом окна, уровнем звука и света.

Настройка SCADA на конкретную задачу. Одной из основных функций SCADA является разработка человеко-машинного интерфейса (ЧМИ). SCADA является одновременно ЧМИ и инструментом для его создания. Быстрота разработки ЧМИ является основным показателем качества SCADA с точки зрения системного интегратора. В процесс разработки входят следующие операции:

- 1) создание графического интерфейса: мнемосхем, графиков, таблиц, всплывающих окон, элементов для ввода команд оператора и т.д.;
- 2) программирование и отладка алгоритмов работы систем автоматизации в режиме эмуляции (имитационного моделирования) и на реальном оборудовании;
- 3) настройка системы коммуникации (сетей, модемов, коммуникационных контроллеров и т.д.), создание базы данных и подключение к ним SCADA.

Диспетчерское управление. Как система диспетчерского управления SCADA может выполнять следующие задачи:

- 1) взаимодействие с оператором: выдача визуальной и слуховой информации, передаче в систему команд оператора;
- 2) помощь оператору в принятии решений (функции экспертной системы);
- 3) сигнализация об авариях и критических ситуациях;
- 4) ведение журнала событий в системе;
- 5) извлечение информации из архива и представление ее оператору в удобном виде;
- 6) подготовка отчетов;
- 7) учет наработки технологического оборудования.

Автоматическое управление. Задачи автоматического регулирования обычно выполняется ПЛК, но более сложные задачи регулирования и управления могут возлагаться на SCADA. ПЛК в небольших системах могут отсутствовать, тогда компьютер со SCADA является единственным средством управления. SCADA обычно выполняет следующие функции управления:

- 1) автоматическое регулирование;
- 2) управление последовательностью операций в системе автоматизации;
- 3) адаптивное и оптимальное управление процессами;
- 4) автоматическая блокировка ИУ при выполнении заданных условий.

Хранение истории процессов. Знание предыстории позволяет улучшить будущее поведение системы, проанализировать причины аварийных ситуаций, брака и ошибок оператора. Для создания истории система выполняет следующие функции:

- 1) сбор данных и их обработка: цифровая фильтрация, интерполяция, сжатие, нормализация, масштабирование и т.д.;
- 2) архивирование данных: действий оператора, собранных и обработанных данных, алармов, графиков, экранных форм и т.п.;
- 3) управление базами данных реального времени и архивных.

Выполнение функций безопасности. Сейчас PCY используются для управления ответственными процессами и должна быть обеспечена их защита от действия враждебных лиц. Для повышения безопасности SCADA используют следующие методы:

- 1) разграничение доступа к системе между разными категориями пользователей;
- 2) защита информации (шифрование, секретность протоколов связи);
- 3) другие методы.

Выполнение общесистемных функций. На SCADA могут возлагаться некоторые общесистемные функции:

- 1) взаимодействие между несколькими SCADA, между SCADA и другими программами;
- 2) диагностика аппаратуры, каналов связи и ПО.

Разработка SCADA-системы. Программные продукты класса SCADA широко представлены на мировом рынке. Это несколько десятков SCADA - систем, многие из которых нашли свое применение и в странах СНГ. Наиболее популярные из них приведены ниже:

SCADA	Фирма-разработчик	Страна
Cimplicity	GE Fanuc Automation	США
Citect	CI Technology	Австралия
iFIX	Intellution	США
Genesis	Iconics	США
InTouch	Wonderware	США
MasterSCADA	InSAT	Россия
TraceMode	AdAstra	Россия
WinCC	Siemens	Германия
КРУГ2000	НПО "Круг"	Россия

Выбор SCADA осуществляется на основе технических, экономических и эксплуатационных характеристик.

После выбора SCADA - системы, начинается разработка АСУТП для конкретного объекта, включающая следующие этапы:

- 1) Разработка архитектуры АСУТП в целом. На этом этапе определяется функциональное назначение каждого узла системы.
- 2) Решение вопросов, связанных с возможной поддержкой распределенной архитектуры.
- 3) Создание прикладной программы для каждого узла, т.е. написание алгоритмов, совокупность которых позволяет решать задачи автоматизации.
- 4) Связь прикладной программы устройствами нижнего уровня (ПЛК, датчики, исполнительные устройства и др.)
- 5) Отладка созданной прикладной программы в режиме эмуляции.