

УДК 621.583

ПРИНЦИП РАБОТЫ И ОСНОВЫ ПРОГРАММИРОВАНИЯ ПЛК

Леганьков Д.А.

Научный руководитель – старший преподаватель Жуковская Т.Е.

Программируемые логические контроллеры (ПЛК)

До появления твердотельных логических схем разработка систем логического управления основывались на электромеханических реле. По сей день реле не устарели в своем предназначении, но все же в некоторых своих прежних функциях они заменены контроллером.



В современной промышленности существует большое количество различных систем и процессов, требующих автоматизации, но теперь такие системы редко проектируются из реле. Современные производственные процессы нуждаются в устройстве, которое запрограммировано на выполнение различных логических функций. В конце 1960-х годов американская компания «Bedford Associates» разработала компьютерное устройство, названное MODICON (Modular Digital Controller). Позже название устройства стало названием подразделения компании, спроектировавшей, сделавшей и продавшей его.

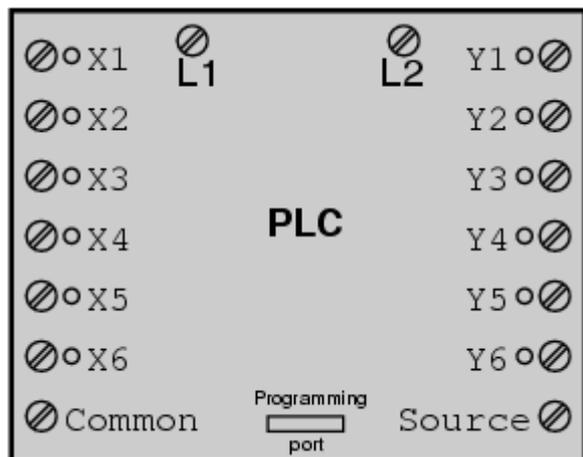
Другие компании разработали собственные версии этого устройства, и, в конце концов, оно стало известно, как ПЛК, или программируемый логический контроллер. Целью программируемого контроллера, способного имитировать работу большого количества реле, была замена электромеханических реле на логические элементы.

ПЛК имеет набор входных клемм, с помощью которых можно контролировать состояние датчиков и выключателей. Также имеются выходные клеммы, которые сообщают «высокий» или «низкий» сигнал индикаторам питания, электромагнитным клапанам, контакторам, небольшим двигателям и другим самоконтролируемым устройствам.

ПЛК легки в программировании, так как их программный язык напоминает логику работы реле. Так обычный промышленный электрик или инженер-

электрик, привыкший читать схемы релейной логики, будет чувствовать себя комфортно и при программировании ПЛК на выполнение тех же функций.

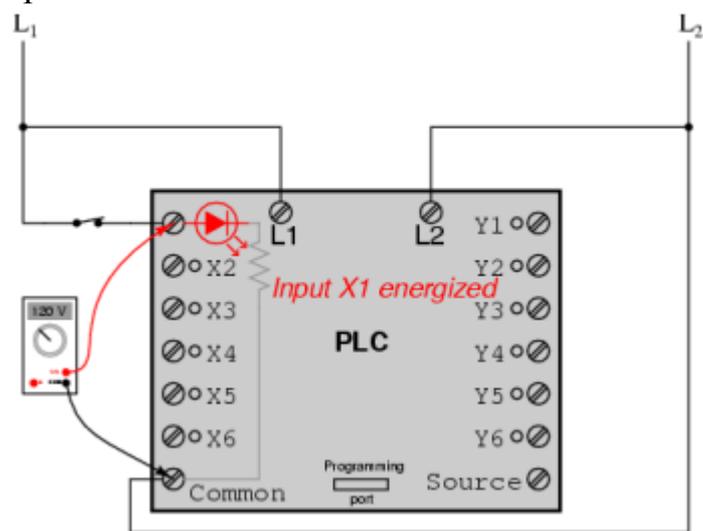
Подключение сигналов и стандартное программирование несколько отличаются у разных моделей ПЛК, но они достаточно схожи, что позволяет разместить здесь «общее» введение в программирование этого устройства.



Следующая иллюстрация показывает простой ПЛК, а точнее то, как он может выглядеть спереди. Две винтовые клеммы, обеспечивающие подключение для внутренних цепей ПЛК напряжением до 120 В переменного тока, помечены *L1* и *L2*.

Шесть винтовых клемм, расположенных с левой стороны, обеспечивают подключение для входных устройств. Каждая клемма представляет свой входной канал (X).

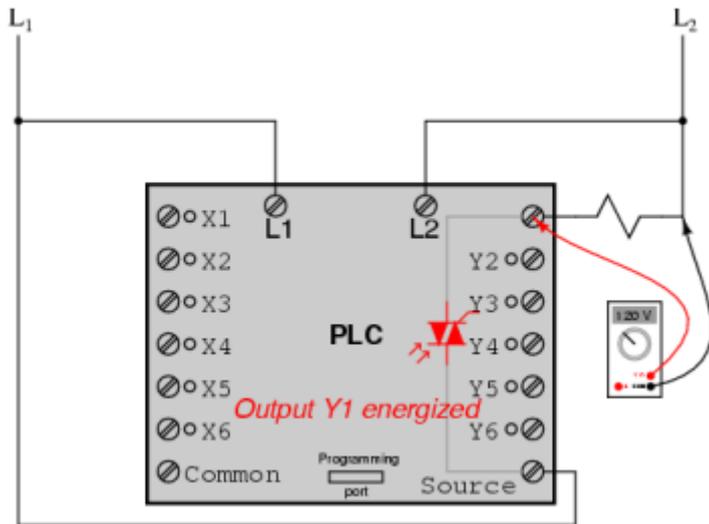
Винтовая клемма («общее» подключение) расположенная в левом нижнем углу обычно подключается к L2 (нейтральная) источника тока напряжением 120 В переменного тока.



Внутри корпуса ПЛК, связывающего каждую входную клемму с общей клеммой, находится оптоизолятор устройства (светодиод), который обеспечивает электрически изолированный «высокий» сигнал для схемы компьютера (фототранзистор интерпретирует свет светодиода), когда 120-тивольтовый переменный ток устанавливается между соответствующей входной

клеммой и общей клеммой. Светодиод на передней панели ПЛК дает возможность понять, какой вход находится под напряжением:

Выходные сигналы генерируются компьютерной схмотехникой ПЛК, активируя переключающее устройство (транзистор, тиристор или даже электромеханическое реле) и связывая клемму «Источник» (правый нижний угол) с любым помеченным буквой Y выходом. Клемма «Источник» обычно связывается с L1. Так же, как и каждый вход, каждый выход, находящийся под напряжением, отмечается с помощью светодиода:

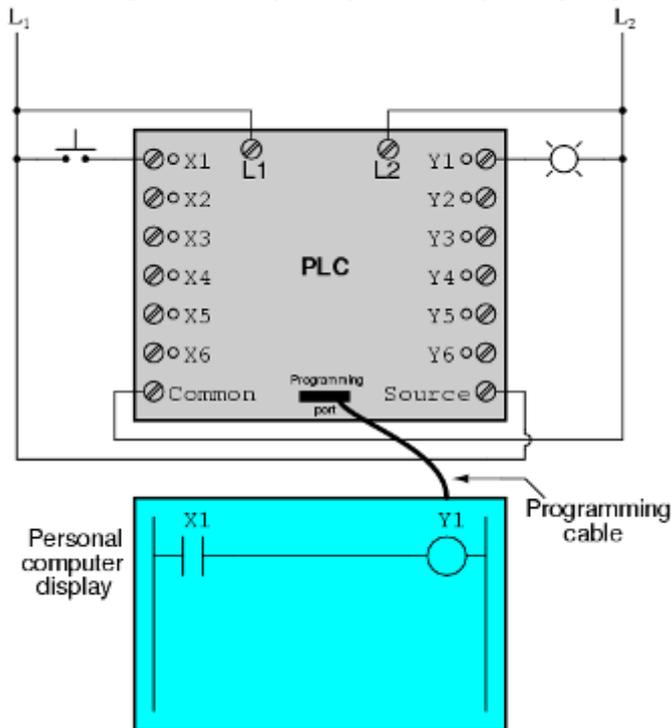


Таким образом, ПЛК может подключаться к любым устройствам, таким как переключатели и электромагниты.

Основы программирования ПЛК

Современная логика системы управления установлена в ПЛК посредством компьютерной программы. Эта программа определяет, какие выходы находятся под напряжением и при каких входных условиях. Хотя сама программа напоминает схему логики реле, в ней не существует никаких контактов переключателя или катушек реле, действующих внутри ПЛК для создания связей между входом и выходом. Эти контакты и катушки мнимые. Программа пишется и просматривается с помощью персонального компьютера, подключенного к порту программирования ПЛК.

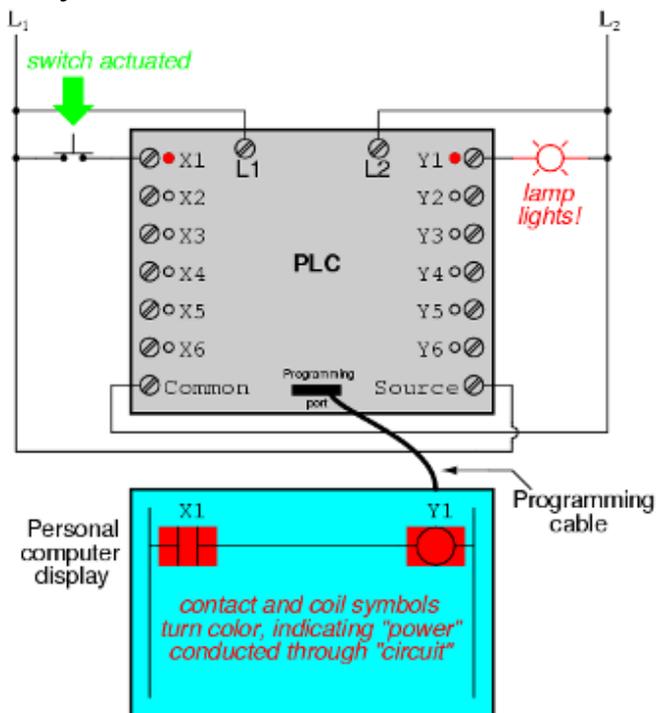
Рассмотрим следующую схему и программу ПЛК:



Когда кнопочный переключатель не задействован (находится в не нажатом состоянии), сигнал не посылается на вход X1. В соответствии с программой,

которая показывает «открытый» вход X1, сигнал не будет посылаться и на выход Y1. Таким образом, выход Y1 останется обесточенным, а индикатор, подключенный к нему, погасшим.

Если кнопочный переключатель нажат, сигнал будет отправлен к входу X1. Все контакты X1 в программе примут активированное состояние, как будто они являются контактами реле, активированными посредством подачи напряжения катушке реле, названной X1. В этом случае открытый контакт X1 будет «закрыт» и отправит сигнал к катушке Y1. Когда катушка Y1 будет находиться под напряжением, выход Y1 осветится лампочкой, подключенной к нему.



Следует понимать, что контакт X1 и катушка Y1 соединены с помощью проводов, а «сигнал», появляющийся на мониторе компьютера, виртуальный. Они не существуют как реальные электрические компоненты. Они присутствуют только в компьютерной программе - часть программного обеспечения - и всего лишь напоминают то, что происходит в схеме реле.

Не менее важно понять, что компьютер, используемый для написания и редактирования программы, не нужен для дальнейшего использования ПЛК. После того, как программа была

загружена в программируемый контроллер, компьютер можно отключить, и ПЛК самостоятельно будет выполнять программные команды. Мы включаем монитор персонального компьютера в иллюстрации для того, чтобы вы поняли связь между реальными условиями (замыкание переключателя и статусы лампы) и статусы программы (сигналы через виртуальные контакты и виртуальные катушки).

Истинная мощь и универсальность ПЛК раскрывается, когда мы хотим изменить поведение системы управления. Поскольку ПЛК является программируемым устройством, мы можем изменить, команды, которые мы задали, без перенастройки компонентов, подключенных к нему. Предположим, что мы решили функцию «переключатель – лампочка» перепрограммировать наоборот: нажать кнопку, чтобы выключить лампочку, и отпустить ее, чтобы включить.