КИПиА ЯМАЛ

Слесарь по контрольно-измерительным приборам и автоматике

5-й разряд · МДК.01.03

Тема 3.2

«Программируемые логические контроллеры»

Полный конспект лекций и лабораторно-практической подготовки

6 академических часов: 4 часа теории + 2 часа практики

Тренажёр: /plc.php

Редакция: 06.10.2026

СОДЕРЖАНИЕ

1. Занятие 1. Устройство ПЛК, сигналы и цикл работы — 2 часа
2. Занятие 2. LAD, память, таймеры, счётчики и диагностика — 2 часа
3. Лабораторно-практическое занятие — 2 часа
4. Контрольные вопросы и итоговая задача

Цель темы. научить читать структуру ПЛК, связывать полевой сигнал с адресом программы, понимать цикл сканирования и безопасно проверять простые LAD-алгоритмы с памятью, таймером и счётчиком.

1. Занятие 1. Устройство ПЛК, сигналы и цикл работы — 2 часа

ПЛК — промышленное вычислительное устройство для циклического выполнения алгоритмов управления. В учебной схеме важно видеть всю цепочку: физический датчик создаёт сигнал, входной модуль преобразует его в состояние программы, CPU выполняет алгоритм, а выходной модуль воздействует на исполнительное устройство.

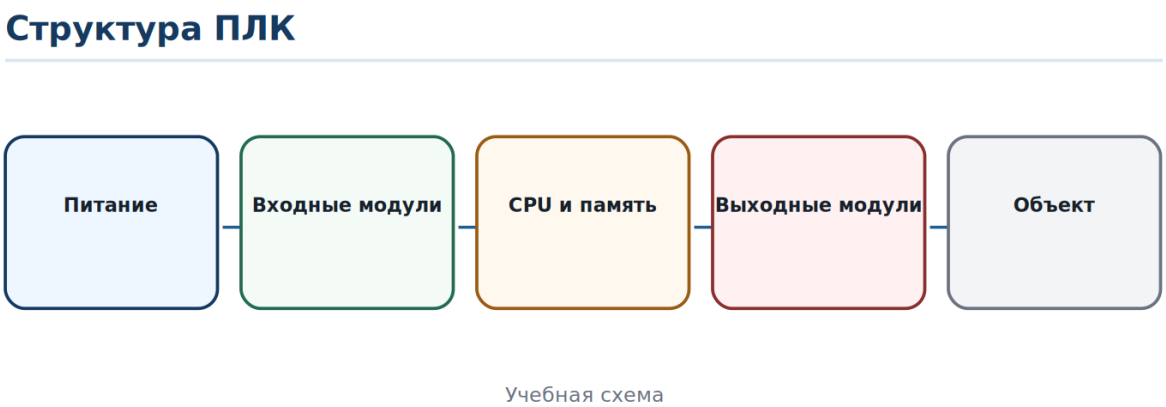


Рисунок 1 — Основные части ПЛК и связь с объектом

1.1. CPU, память и питание

CPU выполняет программу, хранит рабочие данные и обслуживает внутренние устройства. Питание контроллера и полевых цепей должно соответствовать проекту; нельзя автоматически считать, что все входы и выходы имеют одинаковое напряжение или общий провод.

• проверять номинальное напряжение и род тока;

• учитывать полярность DC-входов;

• разделять питание CPU и питание нагрузки, если это требуется схемой;

• проверять защиту выходной цепи.

1.2. Входы и выходы



Рисунок 3 — Основные типы сигналов автоматизации

Дискретный вход сообщает состояние 0/1. Аналоговый вход представляет непрерывную величину, например 4–20 мА. Импульсный вход используется для подсчёта фронтов или частоты. В реальной системе также применяются интерфейсы связи.



Рисунок 4 — От полевого контакта к логическому входу X

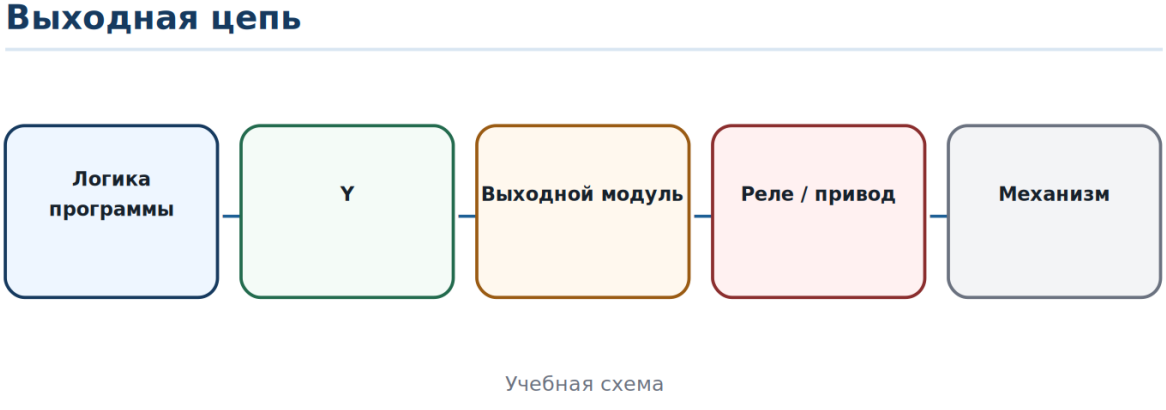


Рисунок 5 — От логического выхода Y к исполнительному устройству

Безопасность. состояние бита Y в программе ещё не доказывает, что механизм реально сработал. Для диагностики сравнивают команду, электрический выход и обратную связь объекта.

1.3. Цикл сканирования

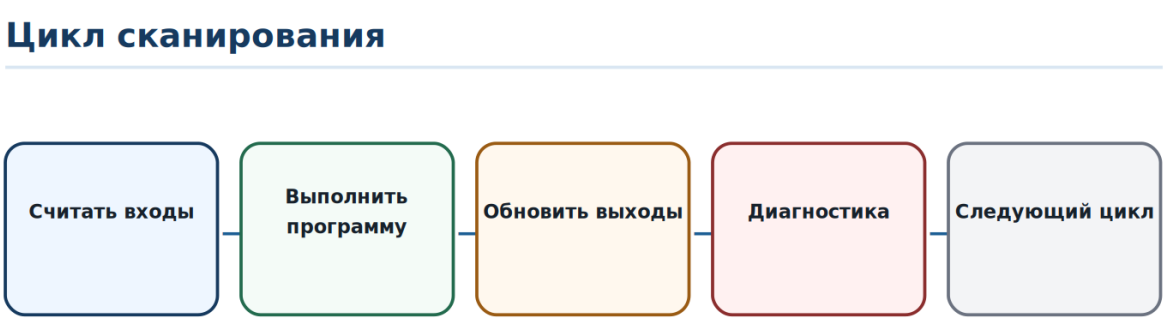


Рисунок 2 — Упрощённый scan-цикл PLC

В типовом циклическом исполнении контроллер считывает входы, выполняет программу на их основе, обновляет выходы и повторяет цикл. Поэтому изменение физического входа и реакция выхода происходят не абсолютно мгновенно: учитываются входные фильтры, scan-time, программа и задержки выходного канала.

В учебном /plc.php цикл выполняется примерно каждые 100 мс, чтобы изменение состояний можно было наблюдать визуально. В реальном ПЛК время цикла определяется аппаратурой, объёмом программы и настройками.

1.4. Карта I/O



Рисунок 6 — Что должно быть в карте входов и выходов

Тег	Адрес	Назначение	Нормальное состояние
START	X0	кнопка Пуск	0
STOP	X1	цепь Стоп	зависит от принятой логики
MOTOR	Y0	команда пускателя	0
RUN_MEM	M0	внутренний самоподхват	0

До программирования полезно отделить физическое обозначение на схеме от адреса PLC. Тег описывает смысл, адрес — конкретный ресурс контроллера.

1.5. Адреса X/Y/M/T/C



Рисунок 7 — Учебная адресация тренажёра

Тип	Смысл в тренажёре	Пример
X	дискретный вход	X0 — Пуск
Y	дискретный выход	Y0 — исполнительный выход
M	внутренний бит/реле	M0 — память самоподхвата
T	таймер	T0 — TON 5 s
C	счётчик	C0 — CTU 5 импульсов

1.6. Языки программирования

Действующий в РФ ГОСТ Р МЭК 61131-3-2016 устанавливает языки программирования ПЛК и основан на IEC 61131-3:2013. Международная IEC 61131-3 обновлена редакцией 2025 года. Для этой темы достаточно уверенно читать LD/LAD и понимать назначение FBD, ST и SFC.

Обозначение	Идея
LD/LAD	контактная логика
FBD	функциональные блоки
ST	структурированный текст
SFC	шаги и переходы

2. Занятие 2. LAD, память, таймеры, счётчики и диагностика — 2 часа

2.1. Контакты и катушки



Рисунок 8 — Структура одного rung

НО-контакт проводит логическую цепь, когда соответствующий бит равен 1. НЗ-контакт логически истинен, когда бит равен 0. Катушка записывает результат rung в выход или внутреннее устройство. Следует различать физически НЗ-кнопку и НЗ-инструкцию в программе: это связанные, но не одинаковые понятия.

2.2. Самоподхват



Рисунок 9 — Хранение команды через внутренний бит M0

Самоподхват позволяет кратковременной кнопке Пуск установить устойчивое состояние. В тренажёре M0 сохраняется после отпускания X0, пока цепь Стоп разрешает работу. Выход Y0 управляется уже от M0.

```
Логика тренажёра
stopOk = НЕ X1.
M0 = stopOk И (X0 ИЛИ M0).
Y0 = M0.
```

2.3. Взаимная блокировка

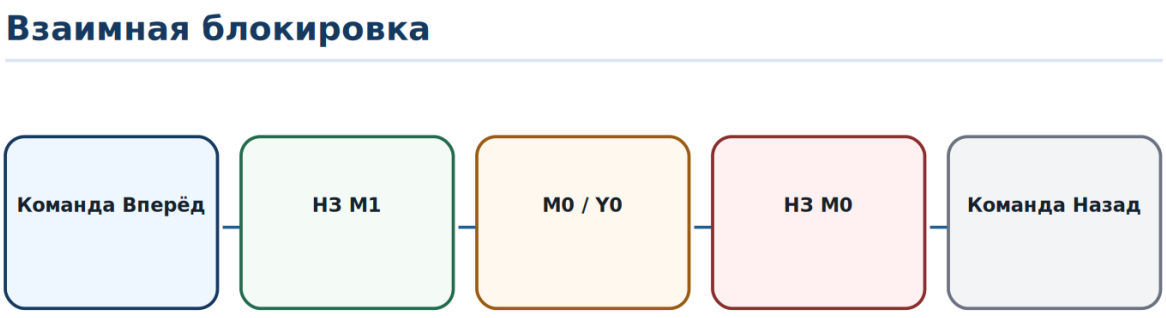


Рисунок 10 — Запрет несовместимых команд

Для реверса нельзя одновременно активировать направления Вперёд и Назад. Программная блокировка использует состояние противоположного внутреннего реле. В реальной силовой схеме программная блокировка обычно дополняется аппаратной.

2.4. Таймер TON



Рисунок 11 — Задержка включения

TON начинает отсчёт, когда вход IN становится истинным. До истечения PT выход Q остаётся 0; после выдержки Q становится 1. В учебном T0 уставка 5 секунд. Если вход исчезает раньше, накопленное время сбрасывается.

```
Проверка TON
Включить X0.
Через 2 с T0.Q должен оставаться 0.
После 5 с T0.Q=1 и Y0 включается.
Снять X0 — таймер и Y0 сбрасываются.
```

2.5. Счётчик CTU



Рисунок 12 — Подсчёт фронтов, а не времени удержания

CTU увеличивает значение при событии, обычно на положительном фронте. В тренажёре C0 считает переход X0 из 0 в 1. Если держать X0 постоянно равным 1, счётчик не должен бесконечно увеличиваться. X1 выполняет RESET.

После пятого фронта C0.done становится 1 и включает Y0.

2.6. Диагностика



Рисунок 13 — Локализация неисправности от команды к объекту

Удобно идти по цепочке: есть ли физическая команда → видит ли её X → проходит ли логика rung → активен ли Y → появилось ли напряжение/сигнал на выходе → сработал ли исполнительный механизм → есть ли обратная связь.

Ситуация Кнопка Пуск нажата, но Y0=0. Если X0=0 — искать проблему до входа PLC. Если X0=1, но условие rung ложное — проверять блокировки и программу. Если Y0=1, но механизм не работает — проверять выходную и силовую цепи.

2.7. Изменение программы



Рисунок 14 — Безопасный маршрут изменения

Изменение рабочей программы выполняют контролируемо: сохранить исходную версию, понять требование, внести минимальное изменение, проверить его офлайн/на стенде, согласовать загрузку, затем провести тесты нормального и аварийного поведения.

Не делать. не загружать неизвестную программу в действующий PLC без понимания безопасного состояния выходов и разрешённой процедуры предприятия.

3. Лабораторно-практическое занятие — 2 часа

ЛПЗ выполняется в /plc.php. Сначала режим «Наблюдение LAD», затем «Практика».

1. Сценарий X0→Y0: переключить X0 и объяснить связь входа, контакта и катушки.

2. Сценарий НЕ X0: сравнить НО и НЗ инструкцию.

3. Сценарии И и ИЛИ: проверить все четыре комбинации X0/X1.

4. Самоподхват: нажать Пуск, отпустить его, затем нажать Стоп; объяснить M0.

5. Взаимная блокировка: попытаться включить два направления и объяснить, почему одновременно Y0/Y1 запрещены.

6. TON: включить X0, записать состояние T0 до 5 с и после 5 с.

7. CTU: сформировать пять отдельных фронтов X0, затем выполнить RESET X1.

8. В режиме «Практика» собрать собственный rung с двумя контактами и Y0 или T0, проверить минимум четыре набора входов.

Проверка	Входы	Ожидаемо	Факт
И	X0=1, X1=1	Y0=1	
Самоподхват	Пуск отпущен	M0=1, Y0=1	
TON	t<5 с	Y0=0	
TON	t≥5 с	Y0=1	
CTU	5 фронтов	C0=5, Y0=1	

4. Контрольные вопросы и итоговая задача

1. Назовите основные части PLC.

2. Что такое цикл сканирования?

3. Чем X отличается от Y?

4. Для чего используются M?

5. Чем таймер отличается от счётчика?

6. Что означает НО контакт в LAD?

7. Что означает НЗ контакт в LAD?

8. Как работает самоподхват?

9. Зачем нужна взаимная блокировка?

10. Как проверить TON до истечения уставки?

11. Что считается одним импульсом CTU?

12. Почему Y=1 ещё не доказывает работу механизма?

13. Что должно быть в карте I/O?

14. Зачем перед изменением программы нужна резервная копия?

Итоговая задача
Требуется: двигатель включается кнопкой X0, удерживается через M0, останавливается X1 и включает Y0.
Дополнительно вентилятор Y1 должен включиться через 5 секунд после M0.
Сначала построить самоподхват M0, затем TON T0 от M0 и отдельный rung T0→Y1.
Проверить: пуск, удержание, останов до 5 с, включение после 5 с и сброс при Стоп.