

12+

Ленар Ильдарович

Садыков

АСУТП с нуля практическое руководство

Ленар Садыков
АСУТП с нуля
практическое руководство

<https://litres.ru/74590188>

ISBN 9785007142649

Аннотация

Книга «АСУТП с нуля: практическое руководство инженера» — сборник из 50 практических задач по программированию промышленных логических контроллеров (ПЛК), рассчитанный на тех, кто хочет освоить автоматизацию не через теорию, а через реальную инженерную практику.

Содержание

АСУТП с нуля	5
Предисловие	6
Глава 0. Минимум теории перед стартом	8
Часть I. Основы (задачи 1—15)	10
IF SA1 THEN	17
Конец ознакомительного фрагмента.	20

АСУТП с нуля практическое руководство

**Ленар Ильдарович
Садыков**

© Ленар Ильдарович Садыков, 2026

ISBN 978-5-0071-4264-9

Создано в интеллектуальной издательской системе Ridero

АСУТП с нуля практическое руководство инженера

50 практических задач по программированию ПЛК

Предисловие

Эта книга — не теоретический курс, а сборник практических задач для тех, кто хочет научиться программировать промышленные логические контроллеры (ПЛК) через реальную практику.

Каждая задача построена по одной и той же схеме:

— **Условие** — что должна делать система.

— **Входы/выходы** — таблица сигналов.

— **Решение** — программа на языке Ladder Diagram (релейно-контактная логика, LD), как наиболее универсальном и наглядном языке стандарта IEC 61131—3. Для части задач дополнительно приведён вариант на Structured Text (ST).

— **Пояснение** — разбор логики и типичных ошибок.

Задачи не привязаны к одному бренду. Логика, представленная здесь, переносится на любую платформу — Siemens (TIA Portal / STEP 7, язык LAD), Allen-Bradley (Studio 5000 / RSLogix, язык Ladder), российские контроллеры ОВЕН (программируются в среде CODESYS или ISaGRAF, в зависимости от серии) и любые другие среды, поддерживающие IEC 61131—3. Там, где есть значимые отличия в реализации между платформами, дана отдельная врезка «Особенности платформ».

Книга разделена на три части по нарастанию сложности:

— **Часть I. Основы** (задачи 1—15) — базовые элемен-

ты: контакты, катушки, таймеры, счётчики, фиксация состояния.

— **Часть II. Типовые промышленные алгоритмы** (задачи 16—35) — конвейеры, насосные станции, сигнализация, взаимные блокировки, ПИД-регулирование на уровне логики.

— **Часть III. Комплексные задачи** (задачи 36—50) — многошаговые последовательности (Sequential Function Chart), аварийные сценарии, диагностика, интеграция нескольких подсистем.

Как пользоваться книгой

Рекомендуется решать задачи по порядку, даже если какие-то темы кажутся знакомыми: каждая следующая задача использует приёмы из предыдущих. Перед тем как смотреть решение, попробуйте написать программу самостоятельно по условию и таблице сигналов — это даёт основной обучающий эффект.

Для проверки решений на реальном или виртуальном ПЛК подойдут:

— Siemens: TIA Portal (есть версия PLCSIM для симуляции без железа);

— Allen-Bradley: Studio 5000 + эмулятор RSLogix Emulate;

— Универсально: CODESYS с виртуальным контроллером CODESYS Control Win.

Глава 0. Минимум теории перед стартом

ПЛК (программируемый логический контроллер)

— промышленный компьютер, циклически выполняющий программу: читает входы → выполняет логику → обновляет выходы. Этот цикл называется **сканированием** (scan cycle).

Языки программирования по IEC 61131—3:

— **LD (Ladder Diagram)** — релейная логика, графический язык, похожий на электрическую схему с «контактами» и «катушками». Самый распространён на практике.

— **ST (Structured Text)** — текстовый язык, похожий на Pascal, удобен для сложных вычислений.

— **FBD (Function Block Diagram)** — блоки функций, соединённые линиями.

— **SFC (Sequential Function Chart)** — язык шагов и переходов для последовательных процессов.

Базовые элементы Ladder Diagram:

— — || — нормально разомкнутый контакт (NO) — пропускает «ток», если сигнал = 1 (TRUE).

— — |/| — нормально замкнутый контакт (NC) — пропускает «ток», если сигнал = 0 (FALSE).

— — () — катушка — устанавливает выход в 1, пока по цепи «течёт ток».

— — (S) — / — (R) — катушки Set/Reset — фиксируют выход в 1 или 0 без удержания условия.

— Таймеры **TON** (задержка на включение), **TOF** (задержка на выключение), **TP** (импульс).

— Счётчики **CTU** (вверх), **CTD** (вниз), **CTUD** (реверсивный).

Этого достаточно, чтобы приступить к задачам.

Часть I. Основы (задачи 1—15)

Задача 1. Управление лампой одной кнопкой (без фиксации)

Условие. Лампа горит, пока нажата кнопка. Отпустили кнопку — лампа гаснет.

Входы/выходы:

Сигнал	Тип	Описание
SB1	Вход (BOOL)	Кнопка "Пуск"
HL1	Выход (BOOL)	Лампа

Решение (LD):

SB1 HL1

—

|

|

()

Пояснение. Простейшая цепь: катушка HL1 активна, пока замкнут контакт SB1. Это базовый принцип — «выход = функция входов в данный момент сканирования», без памяти.

Задача 2. Фиксация состояния (самоподхват / Seal-in)

таксис контактов идентичен показанному. В ST этот же алгоритм записывается так:

$KM1 := (SB1 \text{ OR } KM1) \text{ AND NOT } SB2;$

Задача 3. Реверс кнопки: Set/Reset вместо самоподхвата

Условие. То же самое, что и в задаче 2, но реализовать через инструкции Set/Reset.

Решение (LD):

SB1 KM1

— | |

(S)

SB2 KM1

— | |

(R)

Пояснение. Set (S) устанавливает KM1 в 1 и держит его так независимо от состояния SB1. Reset (R) сбрасывает в 0. Важное правило: если в одном скане активны и Set, и Reset, приоритет имеет тот блок, который расположен в программе **ниже** (выполняется последним) — в данном случае Reset. Это нужно всегда проверять в документации конкрет-

ной платформы, так как порядок приоритета иногда настраивается.

Задача 4. Задержка включения лампы (таймер TON)

Условие. После нажатия кнопки лампа должна загореться через 5 секунд и гореть, пока кнопка удерживается.

Входы/выходы:

Сигнал	Тип	Описание
SB1	Вход	Кнопка
HL1	Выход	Лампа
T1	Таймер TON	Задержка 5 с

Решение (LD):

SB1 T1 (TON, PT=5s)

— |

()

T1.Q HL1

— |

()

Пояснение. Таймер TON начинает отсчёт, пока вход IN=1 (в нашем случае — пока нажата SB1). Через заданное

время РТ выход Q таймера становится 1. Если SB1 отпустить раньше 5 секунд, таймер сбрасывается.

Задача 5. Задержка выключения лампы (таймер TOF)

Условие. Лампа в коридоре должна загораться сразу при срабатывании датчика движения и гаснуть через 30 секунд после того, как движение прекратилось.

Входы/выходы:

Сигнал	Тип	Описание
SQ1	Вход	Датчик движения
HL1	Выход	Лампа
T1	Таймер TOF	Задержка 30 с

Решение (LD):

SQ1 T1 (TOF, PT=30s)

—

|

|

()

T1.Q HL1

—

|

|

()

Пояснение. TOF работает наоборот: выход Q становится

1 сразу, как только $IN=1$, но при переходе IN обратно в 0 выход остаётся включённым ещё в течение времени PT , а затем гаснет.

Задача 6. Мигающая лампа (генератор импульсов на двух таймерах)

Условие. Реализовать лампу-мигалку: 1 секунда горит, 1 секунда не горит, пока включен общий тумблер SA1.

Решение (LD, вариант «перекрёстные таймеры»):

SA1 T2.Q T1 (TON, PT=1s)

— | | — ———|/|

()

T1.Q T2 (TON, PT=1s)

— | |

()

T1.Q HL1

— | |

()

Пояснение. T1 запускается, пока SA1 включен и T2 ещё не сработал. Через 1 секунду $T1.Q=1$, что запускает T2. Ещё через 1 секунду $T2.Q=1$, что через контакт NC размыкает

цепь T1 — таймер T1 сбрасывается, T1.Q падает в 0, T2 тоже сбрасывается, и цикл начинается заново. HL1 повторяет состояние T1.Q, мигая с периодом 2 секунды.

Альтернатива на ST (проще и надёжнее для реальных проектов):

IF SA1 THEN

Blink_Timer (IN:= NOT *Blink_Timer*. *Q*, PT:=
T#1s);

$$HL1 := Blink_Timer.Q;$$

ELSE

$$HL1 := FALSE;$$

END IF;

Задача 7. Счётчик деталей на конвейере (CTU)

Условие. Фотодатчик SQ1 фиксирует прохождение детали (импульс). Нужно считать количество деталей и включать сигнал HL1 «Ящик полон» после 10 штук. Кнопка SB1 сбрасывает счётчик (например, при смене ящика).

Входы/выходы:

Сигнал	Тип	Описание
SQ1	Вход	Датчик прохождения детали
SB1	Вход	Кнопка сброса
HL1	Выход	"Ящик полон"
C1	Счётчик СТУ	PV=10

Решение (LD):

SQ1 C1 (CTU, PV=10)

— | | — — — — —CU

SB1 R (сброс C1)

C1.Q HL1

()

Пояснение. Каждый передний фронт сигнала SQ1 увеличивает текущее значение счётчика на 1 (вход CU считает именно фронты, а не уровень сигнала). Когда текущее значение достигает $PV=10$, выход Q становится 1. Вход R сбрасывает счётчик в 0 независимо от CU.

Задача 8. Определение фронта сигнала (R_TRIG)

Условие. По каждому нажатию кнопки SB1 (именно по нажатию, не по удержанию) выход HL1 должен на один скан программы переключать своё состояние на противоположное (триггер-переключатель, toggle).

Решение (LD):

SB1 R_TRIG XOR-логика:

— || — — (CLK) — — — — Q — — — — ⊣
|

HL1 (текущее) — — — — — XOR

— — — — — (→ HL1)

Пояснение на ST (нагляднее для этой задачи):

R_TRIG_1 (CLK:= SB1);

IF R_TRIG_1.Q THEN

HL1:= NOT HL1;

END_IF;

Блок R_TRIG формирует импульс длительностью один скан именно в момент перехода входа CLK из 0 в 1 (передний фронт). Это ключевой приём для задач «по нажатию», а не «пока нажато» — без него кнопка, удерживаемая дольше одного скана (а это почти всегда так, скан длится миллисекунды), заставила бы HL1 переключаться десятки раз за одно нажатие.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.