# ПЛК

## СОДЕРЖАНИЕ

# ПЛК

## ВВЕДЕНИЕ

Программируемый логический контроллер / Programmable logic controller (***ПЛК / PLC***) - микропроцессорное устройство, предназначенное для сбора информации с объекта управления (опрос датчиков), обработки информации, выполнения алгоритма управления (управляющая программа) и выдачи управляющих команд на исполнительные устройства.

ПЛК широко используются как в небольших локальных системах автоматики (например, управление станками, конвейерами), так и в больших распределенных системах управления. В распределенных системах управления (***РСУ***) ПЛК может выполнять как базовые функции управления, так и роль только системы ввода/вывода, коммутатора и/или хранилища данных.

Вне зависимости от исполнения, ПЛК включает в себя:

- Блок питания;
- Блок центрального процессора (ЦПУ);
- Блок памяти (ОЗУ и память для хранения управляющей программы);
- Блок цифрового интерфейса связи.
- Блок индикации состояния работы ПЛК.
- Кнопка перазагрузки.
- Переключатель состояния «Работа / Программирование».

ПЛК могут питаться как от сети переменного напряжения 220 В AC, так и от сети постоянного — 24 В DC.

Цифровые интерфейсы связи ПЛК предназначены для связи с различными устройствами: интеллектуальными датчиками и исполнительными механизмами, устройствами пуска приводов, модулями удаленного (распределенного) В/В, другими ПЛК, системами диспетчеризации (SCADA) и пр.

Выделяют следующие конструктивные исполнения ПЛК:

- ***Модульная (крейтовая / слотовая) архитектура***
- ***Единое устройство***

Управляющая программа ПЛК является свободно программируемой - реализуется в специальной среде разработки (Integrated Development Environment / ***IDE***). В большинстве ПЛК для написания управляющийх программ используются специальные языки программирования стандарта ***МЭК 61131-3***. Помимо программирования, среда разработки предлагает средства настройки ПЛК — например, настройка сетевых интерфесов, конфигурирование каналов В/В.

Готовая к исполнению управляющая программа загружается в ПЛК через цифровой интерфейс связи с помощью специального программного обеспечения (ПО) или непосредственно из среды разработки.

Некоторые ПЛК и, соответственно, их среды разработки поддерживают режимы отладки и симуляции в реальном времени (on-line mode).

# ПЛК

## АРХИТЕКТУРА

**Модульная (крейтовая) архитектура**

Представляет собой базу со специальными разъемами расширения - слотами, в которые устанавливаются модули в соответствии с решаемой задачей (модули расширяют базовый функционал ПЛК).

Модули — съемные и разного функционала, предназначенные для установки в конкретную модель (или серию) ПЛК. В один слот расширения можно установить только один модуль. Как правило, слот является универсальным — можно устанавливать модули различного функционала.

Максимальное количество слотов расширения зависит от конкретной модели ПЛК.

Некоторые модели ПЛК поддерживают «горячую замену» модулей — т. е. замену модулей без выключения питания.

Выделяют следующие модули по своему функционалу:

- Модуль ЦПУ
  включая: ОЗУ, память программ, индикацию, системные кнопки, сетевые порты
  опционально включая: часы реального времени, энергонезависимую память
- Модуль резервного ЦПУ
- Модуль блока питания
- Модуль резервного блока питания
- Модули Ввода/Вывода (В/В)
  унифицированные аналоговые и цифровые двоичные сигналы
- Модули сетевых портов (интерфейсов)
  RS-485 / PROFIBUS, ETHERNET / PROFINET, оптические

Блок ЦПУ и питания могут быть встроенными в базу (в случае неисправности одного из этих компонентов, замене подлежит вся база).

В случае выхода из строя модуля (или одного из каналов В/В модуля) замене подлежит только вышедший из строя модуль, а не весь ПЛК.

*Промышленный ПЛК ICP DAS iP-8847*

*Промышленный ПЛК SIEMENS S7-410H*

# ПЛК

## АРХИТЕКТУРА

**Единое устройство**

Архитектура ПЛК, когда блоки ЦПУ и питания, каналы В/В и сетевые интерфейсы объединены в одном корпусе.

Расширение функционала в данном случае выполняется штатными средствами (которые доступны и ПЛК с модульной архитектурой) — через сетевые интерфейсы с использованием какого-либо промышленного протокола (универсальный способ).

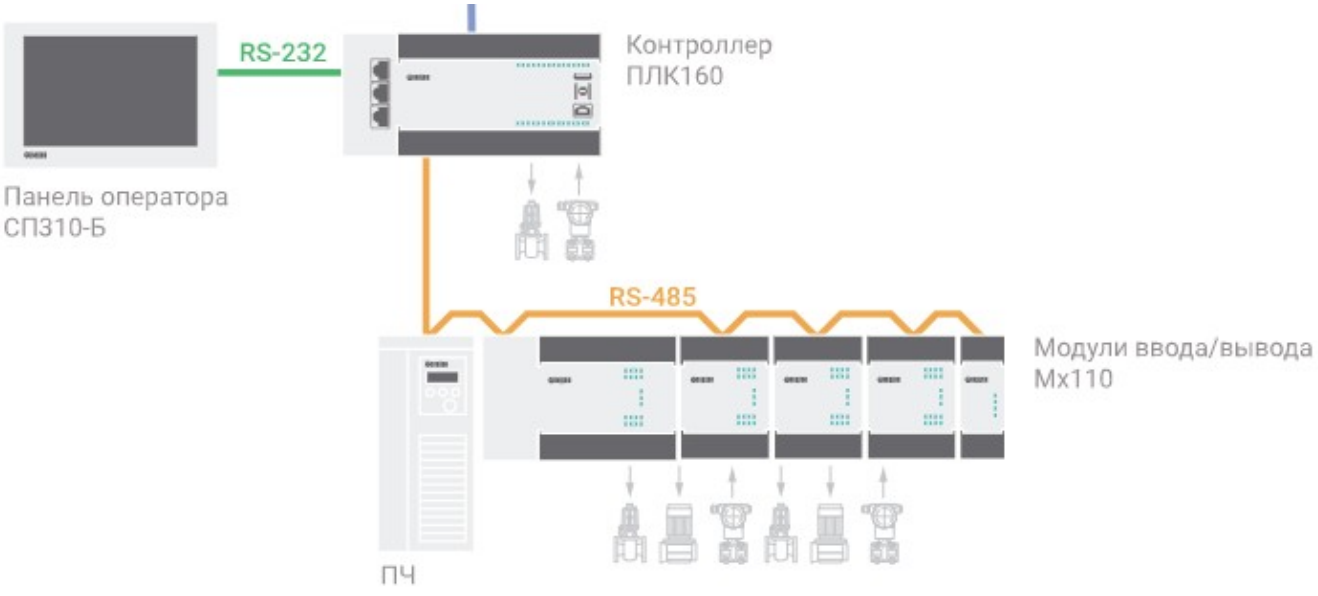
Обмениваться данными в данном случае можно с любым устройством, поддерживающим аналогичный способ и протокол связи — это, например, удаленные станции В/В, устройства пуска приводов (плавный пуск, ПЧ), другие ПЛК, станции оператора и пр.

Для некоторых моделей ПЛК данной архитектуры выпускаются специальные модули расширения, связь с которыми осуществляется по специальным интерфейсным линиям (общей интерфейсной шине).

В случае выхода из строя одного из компонентов ПЛК данной архитектуры (например, блок питания, сетевой интерфейс, канал В/В) — замене подлежит весь ПЛК, если, конечно повреждение носит критический характер или без этого компонента (например, канала В/В или сетевого интерфейса) невозможно выполнять поставленную задачу.

*ПЛК Овен 100 (малые системы автоматизации)*


*ПЛК Segnetics Pixel + специальные Модули В/В MR (средние системы автоматизации)*




*ПЛК Овен 160 + Панель оператора + универсальные Модули В/В + ПЧ (средние системы автоматизации)*

# ПЛК

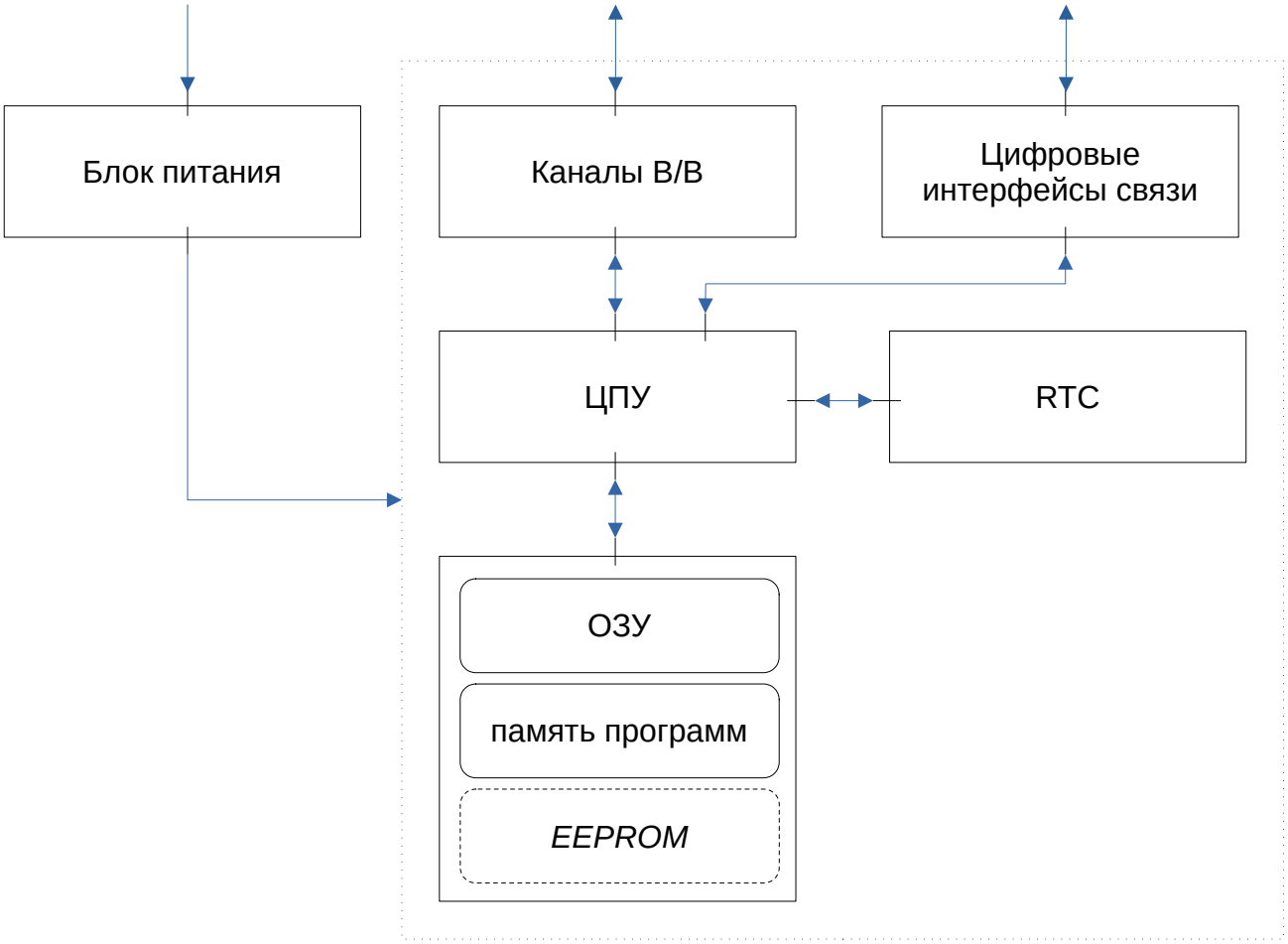
## БАЗОВЫЕ ФУНКЦИОНАЛЬНЫЕ БЛОКИ

Вне зависимости от исполнения, ПЛК включает в себя функциональные блоки:

- Блок питания
- Блок центрального процессора (ЦПУ)
- Блок памяти:
  - ОЗУ
  - память программ
  - энергонезависимая память (EEPROM)*
- Блок часов реального времени (RTC)*
- Блок каналов В/В
- Блок цифровых интерфейсов связи

* - *опциональные позиции*

Все, кроме блока питания, может быть реализовано в виде единого устройства — например, на базе одного кристалла микроконтроллера.



# ПЛК

## ПАМЯТЬ

### ОЗУ

ОЗУ (оперативная память) — перезаписываемая энергозависимая память, в которой во время работы ПЛК хранится выполняеамый машинный код (программы), а также входные, выходные и промежуточные данные.

ОЗУ, обычно, представляет собой память с произвольным доступом (***RAM*** — Random Access Memory) — доступ к данным осуществляется по адресу. Данный вид памяти является наиболее быстродействующим. Для увеличения скорости общения процессора с ОЗУ иногда применяют сверхбыструю память промежуточного уровня или аппаратный кэш (может быть встроен в процессор). Количество операций записи неограничено.

Содержащиеся в ОЗУ данные доступны и сохраняются только тогда, когда на модуль памяти подано питающее напряжение. Выключение напряжения, даже кратковременное, приводит к потере информации.

### Память программ

Память программ ПЛК представляет собой флэш-память (***Flash Memory***) — разновидность перезаписываемой энергонезависимой памяти.

Является памятью с произвольным доступом (по адресу). Но, данный вид памяти менее быстродействующий, чем ОЗУ, и имеет конечное количество циклов записи (при превышении заявленного количества цикла процедура записи может сохраниться, но производитель памяти уже не гарантирует качество и сохранность данных).

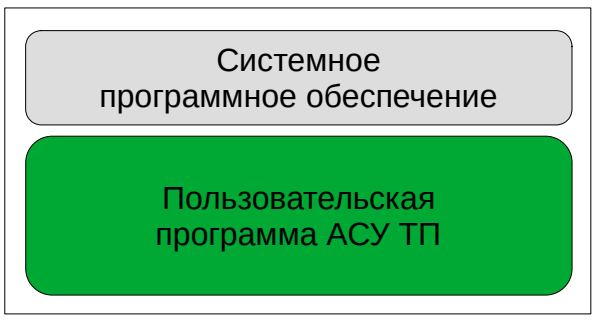
Память программ ПЛК условно делится на два раздела:

- ***системный раздел*** для системного программного обеспечения
- ***пользовательский раздел*** для пользовательской программы АСУ ТП

В каждый раздел памяти программ ПЛК с помощью специальных устройств-программаторов и специального программного обеспечения (загрузчиков) выполняется загрузка соответствующего программного обеспечения (встраиваемое ПО или прошивка).

Системное программное обеспечение (ПО) загружается на заводе-изготовителе ПЛК и, обычно, в дальнейшем может быть обновлено конечным пользователем с помощью специальных аппаратно-программных средств.

Пользовательская программа разрабатывается конечным пользователем в специальной среде разработки и загружается с помощью спецальных средств в соответствующий раздел памяти ПЛК.



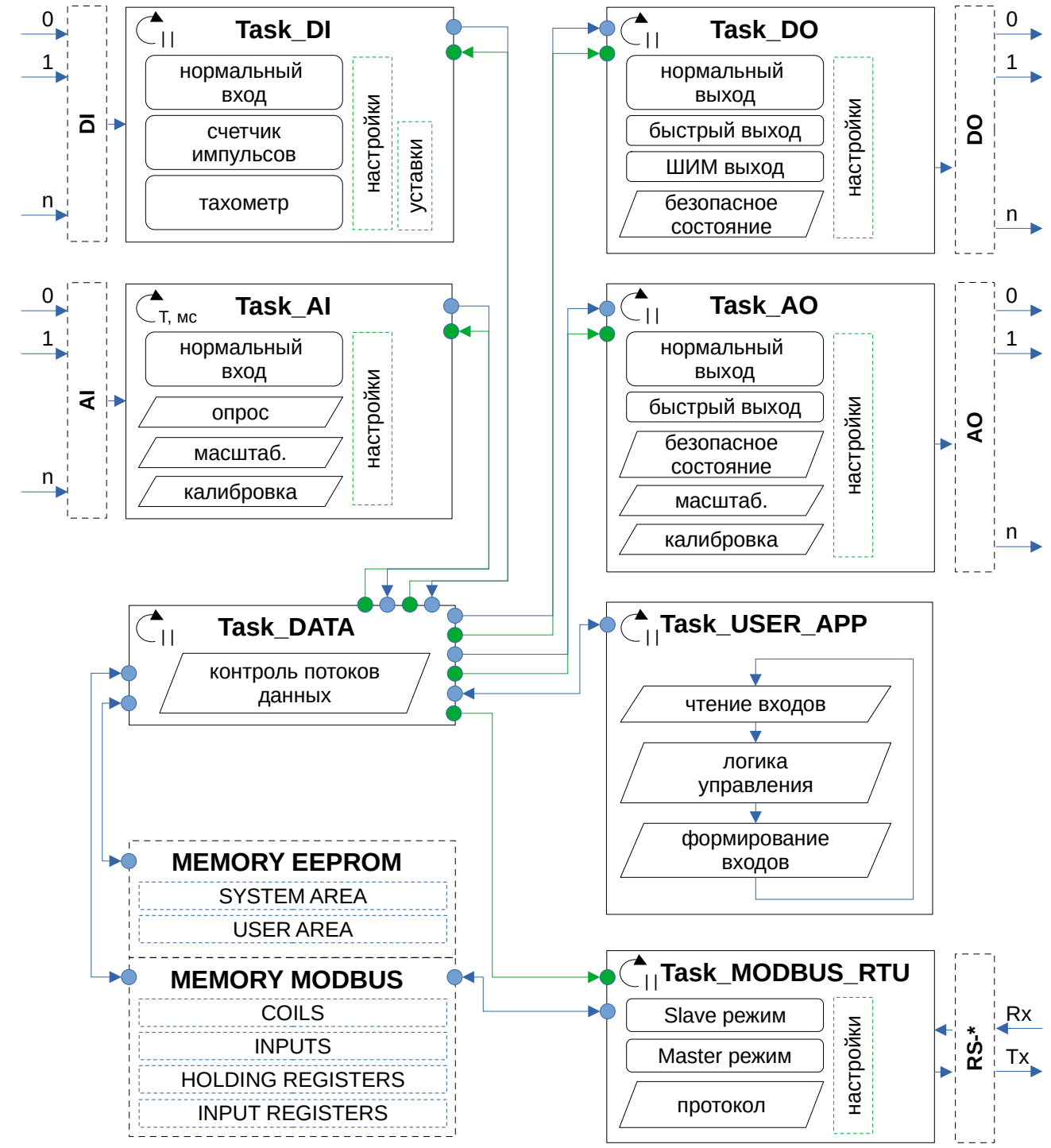
# ПЛК

## СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системное программное обеспечение включает в себя:

- Операционную систему реального времени (ОСРВ)
- Комплекс программ (задач) для управления ПЛК

Базовая структурная схема (потоковая диаграмма) комплекса управления ПЛК:




T, мс - периодический режим работы (задержка на время T в миллисекундах)

\| \| - работа в режиме ожидания (ожидание получение данных)

# ПЛК

## СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (продолжение)

Задача «**Task_DI»**

- Обслуживает каналы цифрового (дискретного) ввода
- Работает в режиме ожидания смены фронта каналов ввода (по прерыванию)
- Функции:
  - прием уровней каналов ввода в режиме «Нормальный»
  - подсчет импульсов для каналов ввода в одном из счетных режимах: «Счетный», «Тахометр», «Энкодер»
  - проверка достижения уставки для счетных каналов ввода
  - отправка данных в задачу «Task_DATA»
  - прием команд, настроек и уставок от задачи «Task_DATA»

Задача «**Task_AI»**

- Обслуживает каналы аналогового ввода
- Работает в режиме периодического опроса каналов ввода (цикл опроса = 50 мс)
- Функции:
  - чтение уровней каналов ввода (с АЦП) в режиме «Нормальный»
  - масштабирование кода АЦП в значение типа REAL (В, мА и т.п.)
  - отправка данных в задачу «Task_DATA»
  - прием команд, настроек и уставок от задачи «Task_DATA»
  - выполнение калибровки

Задача «**Task_DATA»**

- Обеспечивает безопасное взаимодействие между задачами и периферией ПЛК
- Работает в режиме ожидания данных
- Функции:
  - прием данных от задач
  - работа с памятью ПЛК (таблицы ModBus, EEPROM) в режимах: чтение, запись
  - контроль изменения команд, настроек и уставок в памяти ПЛК
  - отправка команд, настроек и уставок задачам

Задача «**Task_DO»**

- Обслуживает каналы цифрового (дискретного) вывода
- Работает в режиме ожидания данных
- Функции:
  - прием выходных уровней для режима «Нормальный»
  - прием данных от обработчика прерываний таймера ШИМ (режим «ШИМ»)
  - прием команд, настроек и уставок от задачи «Task_DATA»
  - выдача уровней на каналы вывода
  - перевод выводов в безопасное состояние
  - отправка признаков и кодов состояния в задачу «Task_DATA»

# ПЛК

## СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (продолжение)

Задача «**Task_AO»**

- Обслуживает каналы аналогового вывода
- Работает в режиме ожидания данных
- Функции:
  - прием выходных уровней для режима «Нормальный»
  - масштабирование значения типа REAL (В, мА и т. п.) в код ЦАП
  - прием команд, настроек и уставок от задачи «Task_DATA»
  - выдача уровней на каналы вывода
  - перевод выводов в безопасное состояние
  - калибровка
  - отправка признаков и кодов состояния в задачу «Task_DATA»

Задача «**Task_USER_APP»**

- Обслуживает пользовательскую программу АСУ ТП
- Работает в режиме периодического исполнения программы пользователя (время цикла программы задается в настройках проекта среды разработки)
- Функции:
  - выполняет пользовательскую программу
  - обмен данными с задачей «Task_DATA»
    (команды, настройки, уставки, уровни для каналов В/В)
    (данные для передачи по сети)

Задача «**Task_MODBUS_RTU»**

- Обеспечивает работу протокола ModBus по сети RS-*
- Работает в режиме ожидания данных (на прием и передачу)
- Функции:
  - поддержка протокола ModBus RTU в режимах: Slave, Master
  - работа с памятью ПЛК (таблицы ModBus) в режимах: чтение, запись
  - взаимодействие с сетевым интерфейсом RS-*
  - обмен данными с задачей «Task_DATA»
    (настройки сетевого интерфейса)

# ПЛК

## ПОЛЬЗОВАТЕЛЬСКАЯ ПРОГРАММА

**Среда разработки**

Разработка пользовательской программы выполняется в специальной среде разработки с использованием специальных языков программирования.

В качестве языков описания алгоритмов и логики работы программы выступают языки программирования стандарта IEC 61131-3.

Среда разработки предоставляет следующие средства:

- редактор проекта
- выбор целевой системы (ПЛК)
- настройки связи с целевым ПЛК
- настройки описания проекта
- настройка глобальных переменных проекта
- редактор языков программирования стандарта IEC 61131-3
- компилятор языков программирования стандарта IEC 61131-3
- кросс-компилятор GCC для целевой системы (ПЛК)
- средства отладки программы в режиме реального времени

Среда разработки позволяет работать в следующих режимах:

- конфигурирование
  - создание прикладной программы
  - реализация алгоритмов и логики в виде программных модулей
  - увязывание программных модулей с каналами ввода/вывода (в том числе интерфейсами связи
- компиляция
  - преобразование кода на языке IEC 61131-3 в машинный код целевого ПЛК
- передача программы на целевой ПЛК (прошивка)
- запуск/останов программы на целевом ПЛК
- отладка программы в режиме реального времени

Этапы компиляции пользовательской программы:

1) исходный код на языке IEC 61131-3 компилируется в единый код на языке ST
2) единый код на языке ST компилируется в код на языке ANSI C (включая дополнительный Си-код для целевого ПЛК)
3) код на языке ANSI C компилируется в машинный код целевого ПЛК

Машинный код (исполняемый файл) может быть загружен в память программ целевой системы через один из сетевых интерфейсов ПЛК (например, загрузка через RS-232, RS-485, ETHERNET).

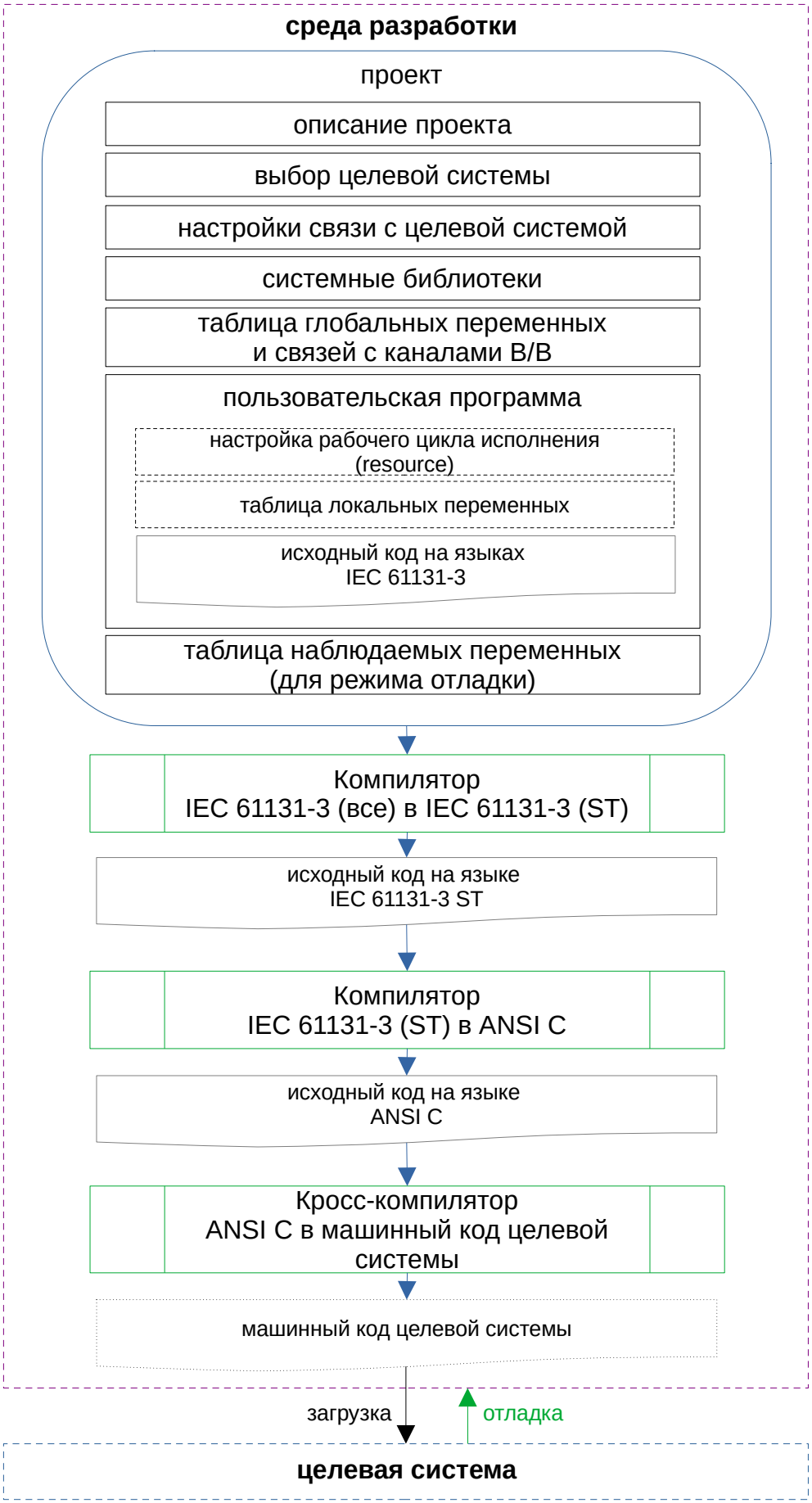
На целевом устройтве исполняемый файл запускается и в процессе работы выполняет следующие действия:

- С помощью драйверов каналов В/В обменивается данными с периферией
- Исполняет алгоритмы и логику управления
- Предоставляет данные для трансляции в системы верхнего уровня (SCADA)
- Сохраняет и транслирует информацию для отладки прикладной программы

# ПЛК

## ПОЛЬЗОВАТЕЛЬСКАЯ ПРОГРАММА (продолжение)

Общая схема по созданию прикладной программы в среде разработки:



# ПЛК

## ПОЛЬЗОВАТЕЛЬСКАЯ ПРОГРАММА (продолжение)

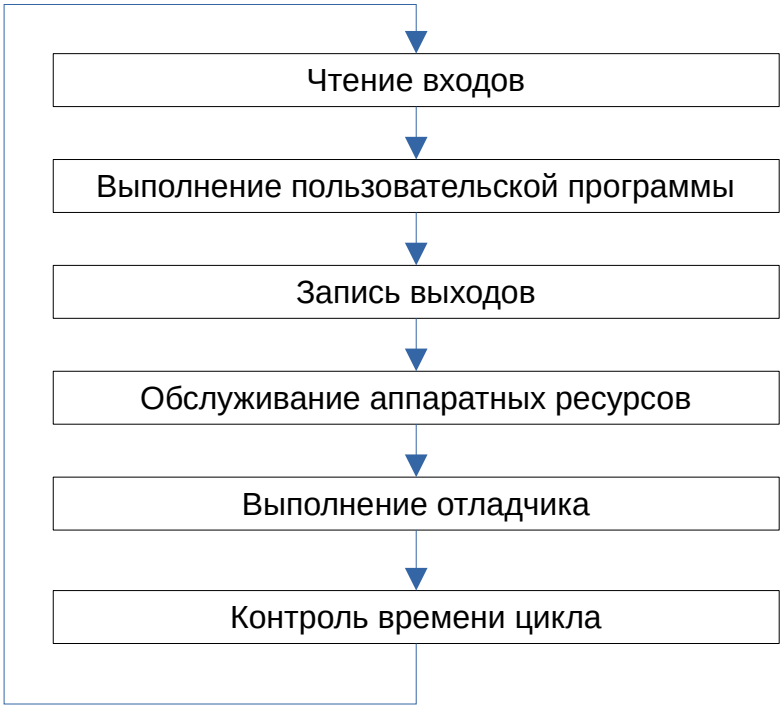
**Рабочий цикл управления**

Задачи управления требуют непрерывного циклического контроля. В любых цифровых устройствах непрерывность достигается за счет применения дискретных алгоритмов, повторяющихся через достаточно малые промежутки времени. Таким образом, вычисления в ПЛК всегда повторяются циклически. Одна итерация,включающая замер, обсчет и выработку воздействия, называется рабочим циклом ПЛК. Выполняемые действия зависят от значения входов контроллера, предыдущего состояния и определяются пользовательской программой (задается в программе).

При включении питания ПЛК выполняет самотестирование и настройку аппаратных ресурсов, очистку оперативной памяти данных (ОЗУ), контроль целостности прикладной программы пользователя. Если прикладная программа сохранена в памяти, ПЛК переходит к основной работе, которая состоит из постоянного повторения последовательности действий, входящих в рабочий цикл.

Рабочий цикл

- Начало цикла
- Чтение состояния физических входов и сохранение их в связанных переменных
  - *создается полная одномоментная зеркальная копия значений входов в памяти*
- Выполнение кода пользовательской программы
  - *программа работает с копией значений входов и выходов из памяти*
  - *если программа не загружена или остановлена, то данная фаза не выполняется*
- Приведение состояния физических выходов в соответствие с расчетными значениями
- Обслуживание аппаратных ресурсов ПЛК
  - *исполнение аппаратно-зависимых и иных системных задач*
- Монитор системы исполнения (отладчик пользовательской программы)
- Контроль времени цикла
- Переход на начало цикла



# ПЛК

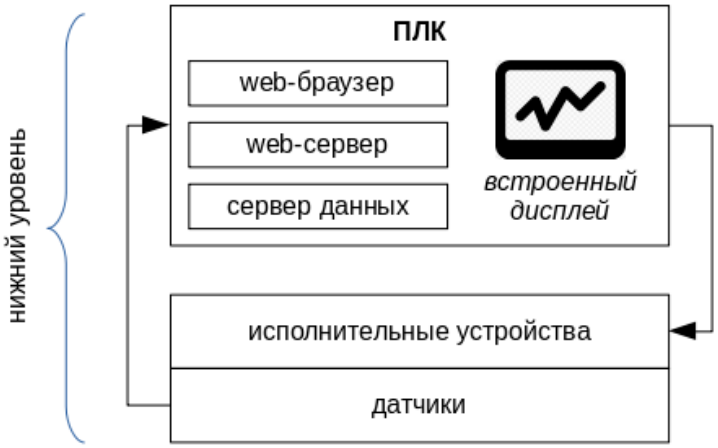
## СОВРЕМЕННЫЕ ТЕХНОЛОГИИ

Мы сегодня живём на рубеже смены эпох в области IT технологий. Эпоха персональных компьютеров, безраздельно главенствующая на протяжении последних трёх десятилетий, неминуемо сдаёт позиции. Вместе с этим, мы вступаем в эпоху «***после ПК***». Мобильные решения и ***облачные технологии*** – далеко не полный список нового поколения информационных технологий. «***Интернет вещей***», ***web-доступ*** к самым разным системам от бытовых приборов до сервисов государственных услуг - неотъемлемые атрибуты современной повседневной жизни. Что принесла новая эпоха и новые IT технологии в системы автоматизации?

Системы визуализации, основанные на web-стандарте ***HTML5***, позволяют отображать информацию не только на привычных мониторах, но и на планшетах, мобильных телефонах и других устройствах.

Современные технологии позволили совершить прорыв сквозь ещё недавно кажущуюся незыблемой жесткую линию разграничения между верхним и нижним уровнями АСУ ТП. Кроссплатформенность всех составляющих компонентов системы позволяет «отвязать» программную реализацию различных её частей от уровня в целом и от конкретных устройств в частности. Другими словами, программная часть становится переносимой не только по горизонтали (между устройствами одного уровня, работающими на различных платформах), но и по вертикали – между уровнями.

**ПЛК как устройство «Все в одном»**



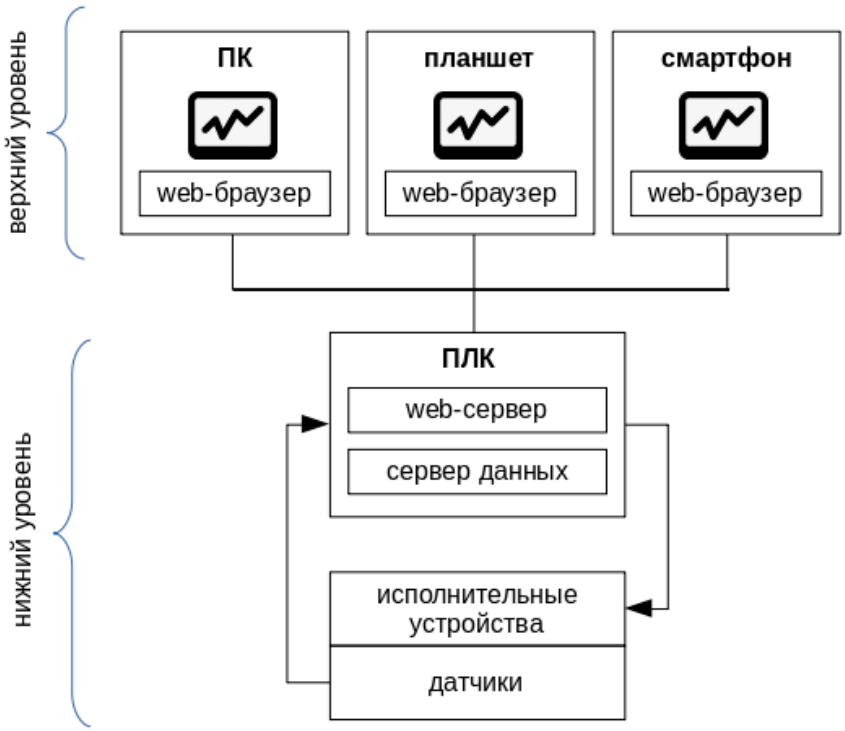

При таком решении все программные компоненты, включая систему визуализации, расположены в контроллере.

Компоненты, являющиеся ранее непременной составляющей верхнего уровня, теперь находятся на тех устройствах, которые ещё недавно считались принадлежностью сугубо нижнего уровня.

## СОВРЕМЕННЫЕ ТЕХНОЛОГИИ

**ПЛК как «Web-сервер и сервер данных»**

Если система визуализации построена на основе web-сервера, то достаточно подключения ПЛК к сети, чтобы часть системы отображения, представляющая собой браузер, могла свободно переместиться за пределы «нижнего» уровня АСУ ТП и перейти на уровень выше.



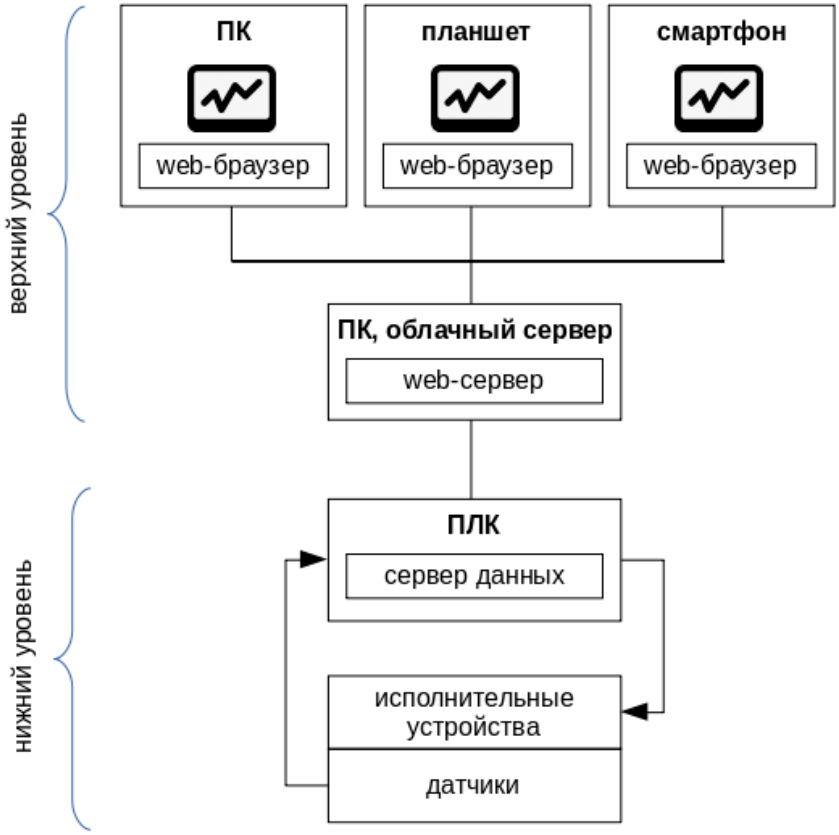

В таком случае на ПЛК располагается сервер данных и web-сервер.

Особенность такого решения заключается в том, что верхней частью системы отображения необязательно должен являться конкретный ПК, а любое устройство, на котором можно запустить веб-браузер, имеющее доступ по сети к ПЛК (например, ноутбук, планшет, смартфон, мобильный телефон).

## СОВРЕМЕННЫЕ ТЕХНОЛОГИИ

**ПЛК как «Сервер данных»**

Конфигурации системы, когда сервер данных остаётся в ПЛК, а система отображения, включая web-сервер, «переезжает» на уровень выше.



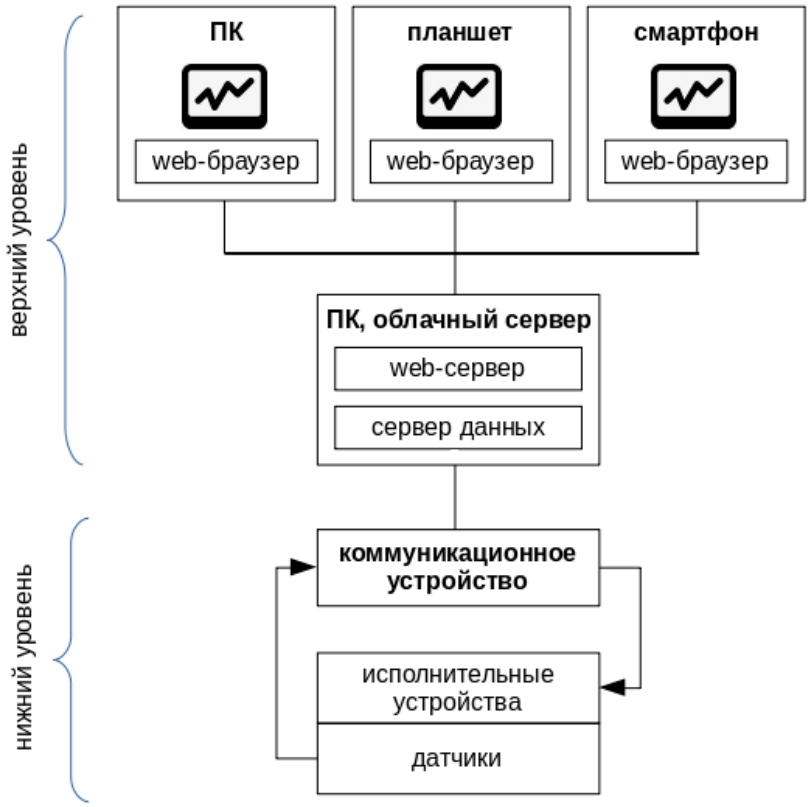

Как таковой «верхний» уровень не привязан к конкретному физическому устройству и может располагаться, в том числе, и на облачном сервере — например, вычислительное пространство в виде виртуального сервера, расположенного на каком-либо сервере (собственном или арендуемом) в сети Internet.

Данное решение позволяет построить систему визуализации не только для нескольких локальных контроллеров, но и для ПЛК и иных интеллектуальных устройств, расположенных на значительном удалении друг от друга.

## СОВРЕМЕННЫЕ ТЕХНОЛОГИИ

**ПЛК как «Коммуникационное устройство»**

Завершающий этап «миграции» системы управления наверх – перемещение туда же и сервера данных со стороны ПЛК.




В этом случае вся логика управления располагается на верхнем уровне, а необходимость в ПЛК, в его привычном представлении – утрачивается. Для связи с исполнительными устройствами и датчиками обратной связи достаточно применения недорогого коммуникационного устройства.

Разумеется, в рамках одного проекта никто не будет хаотично перемещать различные компоненты системы, и такая гибкость может показаться избыточной. Для чего же нужны все эти возможности? Во-первых, достаточно одной программной среды для создания систем управления самой разной конфигурации, топологии и сложности. Во-вторых, в системах управления, как правило, решается большое количество типовых задач. И самое ценное в работе инженера – это его наработки и готовые решения. Среда программирования, обладающая вышеперечисленными свойствами переносимости позволяет использовать одни и те же типовые наработки независимо от места их применения и конфигурации конкретной системы (кроссплатформенность), что сокращает время на разработку и удешевляет её стоимость.