**1. Иерархический принцип построения интегрированных систем управления производством**

Объединение оборудования многономенклатурных производственных систем в единую совокупность, автома­тизация подготовки производства и реализации управле­ния в современных условиях осуществляются человеко-компьютерным комплексом, включающим программные и аппаратные средства вычислительной техники, образую­щие *автоматизированную систему управления*(АСУ). Современные взгляды на построение подобных систем предполагают автоматизацию как сферы материального производства, так и сферы информационной технологии в их взаимосвязи на всех уровнях и стадиях, что приводит к концепции *интегрированных систем управления*(ИАСУ). ИАСУ осуществляет автоматизацию как материальной, так и информационной компонент производственного процесса в их взаимосвязи, от формирования портфеля заказов и до сбыта и отгрузки готовой продукции.

Сложившийся опыт создания и эксплуатации гибких производственных систем в различных областях производства привел к *иерархическому принципу их построения,*предусматривающему распределение как функций, так и реализующих их устройств по уровням иерархии (стратам). Взаимосвязи в этом случае могут существовать только меж­ду рядом расположенными уровнями, а взаимодействие с технологическим оборудованием осуществляется только нижним уровнем ИАСУ. При прекращении работы верх­них уровней процесс производства продолжается за счет продолжения работы нижних уровней в рамках их компе­тентности и в пределах срока их автономности. Работоспо­собность системы в течение определенного времени сохра­няется, хотя и с меньшим уровнем автоматизации. Этим обеспечивается живучесть системы.

Система управления производством обычно делится на 4 уровня (считая сверху):

· 1-й уровень — заводской (уровень предприятия);

· 2-й уровень — межцеховой;

· 3-й уровень — цеховой;

· 4-й уровень — производственные участки, транспорт­ные системы, низовые службы и др.

**2. Основные функции SCADA-систем**

Системы SCADA, от простых конфигураций до больших и сложных установок, составляют основу современного производства, выполняя важные повседневные функции, такие как:

* управление производственными процессами;
* оповещение операторов о проблемах, которые могут привести к простоям;
* мониторинг, сбор и обработка данных;
* непосредственное взаимодействие с устройствами через программное обеспечение и записи событий.

**3. Основные требования, предъявляемые к SCADA-системам**

* надёжность системы (технологическая и функциональная);
* безопасность управления;
* точность обработки и представления данных;
* простота расширения системы.

**4. Основные структурные компоненты SCADA-систем**

Любая SCADA-система включает три компонента: удалённый терминал (RTU – Remote Terminal Unit), диспетчерский пункт управления (MTU – Master Terminal Unit) и коммуникационную систему (CS – Communication System).

Удаленный терминал подключается непосредственно к контролируемому объекту и осуществляет управление в режиме реального времени. Таким терминалом может служить как примитивный датчик, осуществляющий съем информации с объекта, так и специализированный многопроцессорный отказоустойчивый вычислительный комплекс, осуществляющий обработку информации и управление в режиме реального времени.

Диспетчерский пункт управления осуществляет обработку данных и управление высокого уровня, как правило, в режиме квазиреального времени. Он обеспечивает человеко-машинный интерфейс. MTU может быть как одиночным компьютером с дополнительными устройствами подключения к каналам связи, так и большой вычислительной системой или [локальной сетью](https://www.tadviser.ru/index.php?title=%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C&action=edit&redlink=1) рабочих станций и серверов.

Коммуникационная система необходима для передачи данных с RTU на MTU и обратно. В качестве коммуникационной системы могут использоваться следующие каналы передачи данных: выделенные линии, радиосети, аналоговые телефонные линии, [ISDN](https://www.tadviser.ru/index.php?title=ISDN&action=edit&redlink=1) сети, сотовые сети [GSM](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:GSM) ([GPRS](https://www.tadviser.ru/index.php?title=GPRS&action=edit&redlink=1)). Зачастую устройства подключаются к нескольким сетям для обеспечения надёжности передачи данных.

**5. Основные подсистемы SCADA-систем**

SCADA-система обычно содержит следующие подсистемы:

• драйверы или серверы ввода-вывода (СВВ) данных, обеспечивающие связь SCADA с ПЛК, АЦП и другими устройствами ввода/вывода информации;

• система реального времени, позволяющая выполнять приложения с распределёнными приоритетами;

• HMI – человеко-машинный интерфейс, позволяющий визуализировать данные о ходе процесса оператору, и позволяет оператору контролировать процесс и дистанционно управлять им;

• база данных реального времени, обеспечивающая хранение информации о процессе в режиме реального времени;

• система логического управления, обеспечивающая исполнение пользовательских программ в SCADA-системе;

• система управления тревогами, обеспечивающая автоматический контроль технологических событий, а также обработку событий оператором или компьютером;

• генератор отчётов, обеспечивающий создание пользовательских отчётов о технологических событиях;

• редактор для разработки человеко-машинного интерфейса;

• редактор для разработки пользовательских программ;

• внешние интерфейсы обмена данными между SCADA и другими приложениями (OPC, DDE, ODBC, DLL и т.д.).

**6. Механизмы для подсоединения драйверов ввода/вывода к SCADA-системе**

Для подсоединения драйверов ввода/вывода к SCADA-системе в настоящее время используются следующие механизмы:

• стандартные протоколы динамического обмена данными (DDE), открытый механизм взаимодействия с базами данных ODBC;

• собственные протоколы фирм-производителей SCADA-систем, реально обеспечивающие самый скоростной обмен данными;

• OPC (OLE for Process Control ) – протокол, который является стандартным и поддерживается большинством SCADA-систем.

**7. Использование протокола OPC для реализации механизма доступа к данным**

Суть OPC проста — предоставить разработчикам промышленных программ универсальный фиксированный интерфейс (то есть набор функций) обмена данными с любыми устройствами. В то же время разработчики устройств предоставляют программу, реализующую этот интерфейс (набор функций).

Цель OPC-стандарта заключается в определении механизма доступа к данным с любого устройства из приложений. OPC позволяет производителям оборудования поставлять программные компоненты, которые стандартным способом обеспечат клиентов данными с ПЛК. Преимущество использования OPC:

• OPC позволяет определять на уровне объектов различные системы управления и контроля, работающие в распределённой гетерогенной (неоднородной) среде;

• OPC устраняет необходимость использования различного нестандартного оборудования и соответствующих коммуникационных программных драйверов.

**8. Операционные системы для реализации SCADA-системы**

В настоящий момент многие производители, выпускающие SCADA-системы, выпускают версии своих программных продуктов под различные операционные системы (ОС) (Windows, Unix, QNX, VMS и другие операционные системы реального времени (ОСРВ)). Подавляющее большинство SCADA-систем реализовано на платформах MS Windows. Большим недостатком SCADA-систем на платформах Windows по сравнению со SCADA-системами на платформах (ОСРВ) является отсутствие поддержки жёсткого реального времени. Ряд фирм разработали расширения реального времени для семейства платформ Windows NT, превращающие их в ОС жёсткого реального времени (например, подсистеме реального времени RTX (Real Time Extension) фирмы Ventur Com).

**9. Реализация человеко-машинного интерфейса в SCADA-системах**

Одной из основных задач, решаемых SCADA-системами, является обеспечение HMI при представлении информации о процессе. Составной частью графического интерфейса являются формы (экраны) с изображением мнемосхем технологического процесса, анимационных изображений, гистограмм, таблиц, и т.д. Для ограничения доступа к изменяемым параметрам системы предусмотрена система защиты от несанкционированного доступа.

**10. Средства разработки приложений пользователей в SCADA-системах**

Большинство SCADA-систем содержат встроенные средства разработки приложений пользователей. При этом выделяются два подхода реализации средства разработки: • использование языков высокого уровня (Basic-подобные языки); • использование языков международного стандарта МЭК 6-1131/3 (ST (Structured Text), FBD (Function Block Diagram), SFC (Sequential Function Chart), LD (Ladder Diagram), IL (Instruction List)).

**11. Инструментарий среды разработки CodeSys**

Основой комплекса CODESYS является среда разработки прикладных программ для [программируемых логических контроллеров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80) (ПЛК). Она распространяется бесплатно и может быть без ограничений установлена на нескольких рабочих местах. Для программирования контроллера в среде CODESYS в него должна быть встроена система исполнения (Control Runtime System). Она устанавливается в контроллер в процессе его изготовления. Существует специальный инструмент([Software development kit](https://ru.wikipedia.org/wiki/Software_development_kit)), позволяющий адаптировать её к различным аппаратным и программным платформам.

**12. Стандартные элементы конструктора среды разработки CodeSys**

Встроенные компиляторы CODESYS генерируют [машинный код](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) (двоичный код), который загружается в контроллер. Поддерживаются основные 16- и 32-разрядные [процессоры](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80): [Infineon C166, TriCore](https://ru.wikipedia.org/wiki/Infineon_Technologies), [80x86](https://ru.wikipedia.org/wiki/X86), [ARM (архитектура)](https://ru.wikipedia.org/wiki/ARM_(%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0)), [PowerPC](https://ru.wikipedia.org/wiki/PowerPC), [SH](https://ru.wikipedia.org/wiki/SuperH), [MIPS (архитектура)](https://ru.wikipedia.org/wiki/MIPS_(%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0)), [Analog Devices Blackfin](https://ru.wikipedia.org/wiki/Analog_Devices), [TI C2000/28x](https://ru.wikipedia.org/wiki/Texas_Instruments) и другие.

При подключении к контроллеру среда программирования переходит в [режим отладки](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B). В нем доступен мониторинг/изменение/фиксация значений переменных, [точки останова](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D1%87%D0%BA%D0%B0_%D0%BE%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%B0), [контроль потока выполнения](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C_%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%B0_%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F&action=edit&redlink=1), горячее обновление кода, графическая трассировка в [реальном времени](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%B2%D1%80%D0%B5%D0%BC%D1%8F) и другие отладочные инструменты.

CODESYS версии V3 построен на базе так называемой платформы автоматизации: **CODESYS Automation Platform**. Она позволяет изготовителям оборудования развивать комплекс путём подключения собственных [плагинов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD).

Расширенная профессиональная версия среды разработки носит название **CODESYS Professional Developer Edition**. Она включает поддержу [UML](https://ru.wikipedia.org/wiki/UML)-диаграмм классов и состояний, подключение [системы контроля версий](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9) [Subversion](https://ru.wikipedia.org/wiki/Subversion), [статический анализатор](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80_%D0%BA%D0%BE%D0%B4%D0%B0&action=edit&redlink=1) и [профилировщик](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%84%D0%B8%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) кода. Распространяется по лицензии.

Инструмент **CODESYS Application Composer** позволяет перейти от программирования практических приложений к их быстрому составлению. Пользователь составляет собственную базу объектов, соответствующих определенным приборам, механическим узлам машины и т. п. Каждый объект включает программную реализацию и визуальное представление. Законченное приложение составляется из необходимых объектов, конфигурируется и автоматически генерируется программа на языках МЭК 61131-3.

**CODESYS Automation Server**  - это облачная платформа автоматизации для контроллеров с CODESYS. Обеспечивает: удаленный мониторинг данных ПЛК, контроль исправности ПЛК, обновление ПО ПЛК по расписанию, резервное копирование проектов и параметров, контроль версий, удаленное формирование нарядов для обслуживания на местах.

С 20 марта до конца 2020 года платформа удаленной работы **CODESYS Automation Server** доступна бесплатно для всех пользователей.

**13. Типы данных среды разработки CodeSys**

1. **Целочисленные переменные** отличаются различным диапазоном сохраняемых данных и, естественно, различными требованиями к памяти. Подробно данные характеристики представлены в таблице ниже.

2. **Логические переменные** объявляются ключевым словом **BOOL**. Они могут принимать только значение логического нуля («0») *FALSE*(ЛОЖЬ) или логической единицы («1») *TRUE*(ИСТИНА). При начальной инициализации логическое значение по умолчанию — ЛОЖЬ. Занимает 8 бит памяти, если не задан прямой битовый адрес.

3. **Переменные действительного типа** (**REAL** и **LREAL**) представляют действительные числа в формате с плавающей точкой. Для типа REAL необходимо 32 бита памяти и 64 – для LREAL.  
Диапазон значений REAL от: 1.175494351e-38F до 3.402823466e+38F  
Диапазон значений LREAL от: 2.2250738585072014e-308 до 1.7976931348623158e+308

4. **Время суток и дата** типы переменных, выражающие время дня или дату, представляются в соответствии с [ISO 8601](https://docs.cntd.ru/document/1200029038).

5. **Интервал времени** – переменные типа **TIME**. В отличие от времени суток (TIME\_OF\_DAY) временной интервал не ограничен максимальным значением в 24 часа. Числа, выражающие временной интервал, должны начинаться с ключевого слова **TIME#** (в сокращенной форме **Т#**). Максимальное значение для типа TIME: 49d17h2m47s295ms (4194967295 ms).

6. **Тип строковых переменных** (**STRING**) определяет переменные, содержащие текстовую информацию. Размер строки задается при объявлении. Если размер не указан, принимается размер по умолчанию – 80 символов. Размер задается в круглых или квадратных скобках.

**Пользовательские типы данных**

**Массивы, Структуры, Перечисления, Ограничение диапазона значений**

**14. Определение входных и выходных данных систем управления объекта в среде разработки CodeSys**

Входные переменные доступны только для чтения внутри того программного объекта, в котором они объявлены, и доступны для чтения и записи со стороны других программных объектов, вызывающих этот объект. Входные переменные позволяют получить программному объекту данные от других программных объектов.

Выходные переменные доступны для чтения и записи внутри того программного объекта, в котором они объявлены, и доступны для чтения со стороны других программных объектов, вызывающих этот объект. Выходные переменные позволяют программному объекту передать данные другим программным объектам.

Переменные области «вход-выход» доступны для чтения и записи как внутри того программного объекта, в котором они объявлены, так и со стороны других программных объектов, вызывающих этот объект. В большинстве случаев использование этой области не приносит существенной пользы, но затрудняет понимание логики передачи данных внутри приложения. Поэтому мы рекомендуем не использовать эту область в программных объектах (но, например, допустимо использовать ее в объектах визуализации – мы рассмотрим пример такого подхода в п. 6.6.1)

**15. Разработка информационных окон в среде разработки CodeSys**

На программируемых устройствах под управлением CODESYS v3 могут одновременно запускаться несколько визуализаций. Все они могут работать независимо друг от друга. При создании пользовательского интерфейса контроллеров и панелей операторов в среде программирования CODESYS v3 сначала добавляются компоненты визуализации. Для этого в дереве проекта правой кнопкой мыши выбирается приложение Application – Add Object – Visualization и добавляются компоненты Visualization Manager и Visualization. Для добавления в дерево проекта дополнительной страницы визуализации нужно выбрать приложение Application-Add ObjectVisualization.

**16-17. Визуализация аппаратных средств в среде разработки CodeSys и Разработка человеко-машинного интерфейса (HMI) в среде разработки CodeSys**

Одним из важных компонентов среды CODESYS является визуализация. Редактор визуализации позволяет создать человеко-машинный интерфейс (HMI), с помощью которого оператор сможет получать информацию о технологическом процессе и управлять им. Визуализация может отображаться на одном или нескольких клиентах визуализации. В роли такого клиента может выступать:

• экран панельного контроллера или дисплей, подключенный к контроллеру с видеовыходом (эта технология называется TargetVisu); • web-браузер (WebVisu); при этом каждая открытая вкладка считается отдельным клиентом;

• приложение, запускаемое на ПК и представляющее собой специфический вариант виртуального контроллера, предназначенный для работы с визуализацией (CODESYS HMI). Является разновидностью таргет-визуализации;

• приложение, запускаемое на ПК и отображающее таргет-визуализацию контроллера (RemoteTargetVisu). В этом случае обработка визуализации происходит именно на стороне контроллера, а приложение просто подключается к нему как к удаленному рабочему столу (по принципам технологии VNC).

Набор поддерживаемых конкретным контроллером клиентов визуализации определяется его производителем.

**18. Программирование трендов как системы прогнозирования состояния системы управления объекта в среде разработки CodeSys**

Редактор визуализации CODESYS предоставляет разработчику широкий набор графических элементов – начиная от простых (прямоугольники, индикаторы, кнопки, аналоговые дисплеи, ползунки и т. д.) и заканчивая продвинутыми (таблица тревог, тренд, XY-график, текстовый редактор и т. д.).

g\_VisuTrendWithTimeSelector UpdateBehaviour – UINT - Режим применения настроек тренда. Подробнее см. по ссылке (VIS-627)

TimeRangePickerControlVariables – изменение значения одной из управляющих переменных элемента выбора временного диапазона тренда.

Также в библиотеке есть список глобальных переменных Visu\_DateTime, который содержит единственный элемент DisplayUTC типа BOOL. Если он имеет значение TRUE – при отображении времени в элементах визуализации отображается всемирное координированное время (UTC), если FALSE – то отображается локальное время. Это влияет на элементы:

• отображающие системное время (т.е. у которых в параметре Тексты указаны спецификаторы формата времени, но не привязано текстовой переменной);

• тренд;

• таблицу тревог и баннер тревог;

• аналоговые часы и элемент выбора даты и времени (если к нему не привязана переменная).

**19. Стандартные элементы конструктора среды разработки Trace Mode**

TRACE MODE 6 – это программный комплекс, предназначенный для разработки и запуска в реальном времени ПО распределенных автоматизированных систем управления технологическими процессами (АСУТП) и решения ряда задач управления предприятием (АСУП). Для решения задач АСУП в TRACE MODE 6 интегрирован пакет TFACTORY. Комплекс программ TRACE MODE 6 можно разделить на 3 части.

• Интегрированная среда разработки проекта (ИС) – единая программная оболочка, содержащая все необходимые средства для разработки проекта. Итогом разработки проекта в ИС является создание файлов, содержащих необходимую информацию об алгоритмах работы АСУ. Эти файлы затем размещаются на аппаратных средствах (компьютерах и контроллерах) и выполняются под управлением исполнительных модулей TRACE MODE.

• Исполнительные модули (мониторы реального времени, МРВ) – программные модули различного назначения, под управлением которых в реальном времени выполняются составные части проекта, размещаемые на отдельных компьютерах или в 26 контроллерах.

• Драйверы обмена – драйверы, используемые мониторами TRACE MODE для взаимодействия с устройствами, протоколы обмена с которыми не встроены в мониторы.

**20. Определение входных и выходных данных систем управления объекта в среде разработки Trace Mode**

В столбце тип указывают тип аргумента: IN— для передачи в программу или вывода на экран, OUT— для передачи из программы или ввода данных через интерфейс человек-машина, IN/OUT— передача данных как в программу, так и из нее, вывод на экран и ввод данных. В столбце привязка производится привязка к определенному значению канала.

**21. Программирование трендов как системы прогнозирования состояния системы управления в среде разработки Trace Mode**

SCADA TRACE MODE 6 располагает мощными средствами отображения и анализа истории технологического процесса при помощи графиков-трендов. На трендах SCADA TRACE MODE можно выводить информацию за неограниченный временной интервал (определяется глубиной СПАД-архива), при помощи неограниченного числа кривых. На одном тренде есть поля для представления аналоговых и дискретных переменных. Дискретность представления информации на тренде SCADA TRACE MODE составляет 1 мс.

В реальном времени, тренды можно модифицировать, меняя источники данных и параметры кривых, а также распечатывать по команде оператора или временному условию через механизм отчетов.

В SCADA TRACE MODE есть три типа трендов:

* Универсальный тренд;
* Архивный тренд;
* Тренд XY.

Универсальный тренд предназначен для отображения данных, поступающих в реальном времени, а также, при необходимости, данных, извлекаемых из архивов TRACE MODE.  
Архивный тренд - обеспечивает графическое отображение только архивных данных.  
Наконец, Тренд XY отображает значения двух переменных по осям абсцисс и ординат, со временем, учитываемым как параметр.

**22. Разработка информационных окон в среде разработки Trace Mode**

В SCADA системе TRACE MODE® 6 все данные между каналами, экранами, программами и прочими компонентами передаются через *аргументы*. Это позволяет использовать один и тот же компонент многократно. Например, если в проекте автоматизируется 40 однотипных котельных, то нет необходимости редактировать по отдельности 40 мнемосхем. Достаточно создать один экран и 40 *вызовов* этого экрана. Каждый вызов привязан к конкретным каналам через набор *аргументов шаблона экрана*. Чтобы избежать утомительной ручной привязки каналов к аргументам каждого из 40 вызовов, разработчик АСУТП может воспользоваться процедурой автопостроения и автопривязки каналов из редактора аргументов вызова экрана. При ее выполнении для каждого аргумента в выбранном узле TRACE MODE будет создан канал соответствующего типа с именем, совпадающим с именем аргумента.

**23. Типы данных среды разработки Trace Mode**

1. **Целочисленные переменные** отличаются различным диапазоном сохраняемых данных и, естественно, различными требованиями к памяти. Подробно данные характеристики представлены в таблице ниже.

2. **Логические переменные** объявляются ключевым словом **BOOL**. Они могут принимать только значение логического нуля («0») *FALSE*(ЛОЖЬ) или логической единицы («1») *TRUE*(ИСТИНА). При начальной инициализации логическое значение по умолчанию — ЛОЖЬ. Занимает 8 бит памяти, если не задан прямой битовый адрес.

3. **Переменные действительного типа** (**REAL** и **LREAL**) представляют действительные числа в формате с плавающей точкой. Для типа REAL необходимо 32 бита памяти и 64 – для LREAL.  
Диапазон значений REAL от: 1.175494351e-38F до 3.402823466e+38F  
Диапазон значений LREAL от: 2.2250738585072014e-308 до 1.7976931348623158e+308

4. **Время суток и дата** типы переменных, выражающие время дня или дату, представляются в соответствии с [ISO 8601](https://docs.cntd.ru/document/1200029038).

5. **Интервал времени** – переменные типа **TIME**. В отличие от времени суток (TIME\_OF\_DAY) временной интервал не ограничен максимальным значением в 24 часа. Числа, выражающие временной интервал, должны начинаться с ключевого слова **TIME#** (в сокращенной форме **Т#**). Максимальное значение для типа TIME: 49d17h2m47s295ms (4194967295 ms).

6. **Тип строковых переменных** (**STRING**) определяет переменные, содержащие текстовую информацию. Размер строки задается при объявлении. Если размер не указан, принимается размер по умолчанию – 80 символов. Размер задается в круглых или квадратных скобках.

**Пользовательские типы данных**

**Массивы, Структуры, Перечисления, Ограничение диапазона значений**

**24. Визуализация аппаратных средств в среде разработки Trace Mode**

В SCADA системе TRACE MODE® 6 все данные между каналами, экранами, программами и прочими компонентами передаются через *аргументы*. Это позволяет использовать один и тот же компонент многократно. Например, если в проекте автоматизируется 40 однотипных котельных, то нет необходимости редактировать по отдельности 40 мнемосхем. Достаточно создать один экран и 40 *вызовов* этого экрана. Каждый вызов привязан к конкретным каналам через набор *аргументов шаблона экрана*. Чтобы избежать утомительной ручной привязки каналов к аргументам каждого из 40 вызовов, разработчик АСУТП может воспользоваться процедурой автопостроения и автопривязки каналов из редактора аргументов вызова экрана. При ее выполнении для каждого аргумента в выбранном узле TRACE MODE будет создан канал соответствующего типа с именем, совпадающим с именем аргумента.

**25. Разработка человеко-машинного интерфейса (HMI) в среде разработки Trace Mode**

**Автопостроение каналов TRACE MODE по источникам данных** обычно используется сразу после автопостроения самих источников. Этот вид автопостроения реализуется простым перетаскиванием (методом **Drag-n-Drop**) иконки источника данных в узел связанного с ним [монитора реального времени](http://www.adastra.ru/products/runtime/scada/rtm/) (основного сервера TRACE MODE) или SOFTLOGIC контроллера под управлением Микро МРВ. Автопостроенные по источникам данных каналы готовы к использованию.

Фактически, для создания простого человеко-машинного интерфейса (HMI)информационной системы остается только настроить коммуникационные порты узла и создать мнемосхему. Есть и другой способ автопостроения проекта для разработки "от графики". Если разработчик хочет сначала нарисовать мнемосхемы HMI, а уже затем подобрать необходимое оборудование, то ему пригодится **автопостроение каналов из редактора аргументов**.

**26. Инструментарий среды разработки Trace Mode**

SCADA TRACE MODE® 6 состоит из инструментальной системы -[**Интегрированной среды разработки**](http://www.adastra.ru/products/overview/integrated/)и из набора [**исполнительных модулей**](http://www.adastra.ru/products/runtime/)**.**

В Интегрированную среду разработки TRACE MODE® 6 **встроены более десяти редакторов**, автоматически открывающихся, при вызове, того или иного компонента проекта. Среди них:

* Редактор графических мнемосхем и экранных форм;
* Редактор программ на визуальном языке **FBD**;
* Редактор программ на визуальном языке **SFC**;
* Редактор программ на визуальном языке **LD**;
* Редактор программ на процедурном языке **ST**;
* Редактор программ на процедурном языке **IL**;
* Редактор шаблонов документов;
* Построитель связей с СУБД;
* Редактор паспортов оборудования (**EAM**);
* Редактор персонала (**HRM**);
* Редактор материальных ресурсов (**MES**);

**27. Понятие объекта в SCADA-системах**

Под Объектом в SCADA-системах понимается предмет (предприятие или система), который будет автоматизирован.

Объект – это основная единица разрабатываемой системы, соответствующая реальному технологическому объекту (цеху, участку, аппарату, насосу, задвижке, датчику и т.п.), управляемому разрабатываемой с помощью SCADA системой. С другой стороны, это и традиционный с точки зрения программирования объект, обладающий следующими стандартными для программных объектов качествами.

**28. Качества объекта в SCADA-системе с точки зрения программирования**

С другой стороны, это и традиционный с точки зрения программирования объект, обладающий следующими стандартными для программных объектов качествами.

Объект имеет набор свойств и документов, которые жёстко связаны с ним.

Ограничение области видимости.

Наследование.

Типизация и тиражирование.

**29. Понятия свойств объекта и документов объекта в SCADA-системах**

Свойства объекта – это, например, период опроса и способ обработки сигналов от его датчиков. Документы объекта – его изображение, описание, чертёж, перечень сообщений и т.п

**30 - 31. Возможности наследования объектов в SCADA-системах и Понятие наследования и связи объектов в SCADA-системах**

Наследование. По умолчанию все настройки наследуются от «родительского» объекта. Каждый объект имеет множество настроек. Такое обилие могло бы потребовать от разработчика системы выполнения огромного количества действий. Но так как для разных объектов их список в основном одинаков, то все настройки можно сделать только один раз, все подчинённые объекты автоматически воспримут настройки родительского элемента, т.е. «унаследуют» их. Исключение будут составлять только те настройки и только у тех элементов, которые разработчик изменил сам.

Связь объектов позволяет избежать излишнего написания кода. Например, можно разместит на мнемосхеме элемент датчика, а затем связать этот элемент с тегом БД.

**32-33. Понятия типизации и тиражирования объектов в SCADA-системах и Возможности типизации и тиражирования объектов в SCADA-системах**

Типизация и тиражирование. Допускается многократное использование одного и того же объекта со всеми созданными для него документами, в том числе при разработке различных систем. При копировании объекта или сохранении его в библиотеке все его настройки и документы, настройки документов и внутренние связи будут сохранены. Внешние связи с источниками данных будут восстановлены при наличии источников с такими именами, внешние связи с приёмниками данных будут восстановлены, если эти приёмники данных свободны, остальные будут показаны в общем списке. Благодаря этому управление и контроль типовым технологическим объектом (насосом, задвижкой, реактором, фильтром и т.п.) создаются один раз для всех проектов. Это позволяет создавать объекты для одной системы параллельно независимыми разработчиками.

**34. Возможности SCADA-систем для связи с внешними системами (контроллерами, базами данных и др.)**

Система поддерживает популярные программные интерфейсы: OPC, DDE, ODBC. Взаимодействие с внешними базами осуществляется с помощью встроенного редактора SQL-запросов. Кроме того, существует возможность подключения компонентов ActiveX.

**35. Специальные средства повышения надёжности и живучести разрабатываемых систем, предусмотренные в SCADA-системах**

В MasterSCADA предусмотрены специальные средства повышения надежности и живучести создаваемых систем. Прежде всего, это средства резервирования разработчик проекта может резервировать компьютеры, контроллеры, сети. Переключение на резерв производится по команде средств диагностики, вычисляемому в пользовательской программе событию или по команде оператора. Для обеспечения безударного восстановления после сбоя и в компьютерах, и в контроллерах предусмотрен горячий рестарт с восстановлением значений параметров и внутренних накопительных ячеек расчетных алгоритмов. В случае перерывов в работе каналов связи с контроллерами они могут накапливать собственные архивы, которые будут целиком переданы в архивы верхнего уровня после восстановления связи.  
  
Подсистема контроля действий операторов предусматривает ведение журналов, а также предоставление доступа к просмотру информации или управлению в соответствии с должностью зарегистрированного в данный момент на компьютере оператора. Настройка ограничений может быть сделана как для всех действий одного типа (например, управления исполнительными механизмами), так и индивидуально (для конкретной команды).

**36. Способы создания мнемосхем в SCADA-системах**

Как известно, мнемосхема – это наглядное графическое отображение функциональной схемы управляемого и (или) контролируемого объекта. Мнемосхема может быть представлена разными способами – и набором аналоговых индикаторов, и сенсорной панелью с цифровым изображением мнемосхемы, и многими другими.

**37. Основные компоненты среды разработки Trace Mode**

SCADA TRACE MODE® 6 состоит из инструментальной системы -[**Интегрированной среды разработки**](http://www.adastra.ru/products/overview/integrated/)и из набора [**исполнительных модулей**](http://www.adastra.ru/products/runtime/)**.**

**38. Понятие, назначение и возможности режима эмуляции объекта**

Эмуляция объекта – проверка работоспособности объекта.

Для отладки проекта в редакторе представления данных предусмотрено два режима эмуляции работы монитора реального времени:

* непрерывный режим;
* пошаговый режим.

**39. Основные термины и определения: SCADA, интегрированная среда разработки проекта, исполнительный модуль, узел проекта, канал**

Термин «SCADA» имеет двоякое толкование. Наиболее широко распространено понимание SCADA как приложения[[1]](https://ru.wikipedia.org/wiki/SCADA#cite_note-1), то есть программного комплекса, обеспечивающего выполнение указанных функций, а также инструментальных средств для разработки этого программного обеспечения. Однако часто под SCADA-системой подразумевают программно-аппаратный комплекс. Подобное понимание термина SCADA более характерно для раздела [телеметрия](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F).

Значение термина SCADA претерпело изменения вместе с развитием технологий автоматизации и управления технологическими процессами. В 80-е годы под SCADA-системами чаще понимали программно-аппаратные комплексы сбора данных в реальном времени. С 90-х годов термин SCADA больше используется для обозначения только программной части [человеко-машинного интерфейса](https://ru.wikipedia.org/wiki/HMI) АСУ ТП.

Исполнительный модуль интегрированной **SCADA-системы TRACE MODE**, предназначен для исполнения проектов, созданных в [инструментальной системе TRACE MODE](https://www.tracemode.ru/products/articles/3673).

Узел проекта – корневые группы слоя Система. Предопределенное название узла указывает на семейство мониторов, для которых данный узел предназначен. Узел может содержать только те компоненты, которые поддерживаются мониторами соответствующего семейства.

Канал – поток данных, содержащий информацию от или для устрйоств.