# Конспект по дисциплине «Автоматизированные процессы в полиграфии»

# Лекция 1.

Одним из основных направлений деятельности человека является совершенствование процессов производства с целью облегчения тяжелого физического труда и повышения эффективности процесса в целом. Это направление может реализоваться через автоматизацию производственных процессов (АПП).

# Цель АПП:

1) повышение производительности

2) повышение качества изделия

3) улучшение условий труда

Тех процесс состоит из трех основных частей:

1) рабочий цикл (основной технологический процесс)

2) холостых ходов (вспомогательные операции)

3) транспортно-накопительный операции

АПП имеет комплексный подход и не решив одну задачу, может не достигнуть определенного эффекта.

Выделяют два вида: частичная и полная.

Частичная автоматизация – автоматизация части операций по управлению производственным процессом при условии, что остальная часть всех операций выполняется автоматически (пример: автоматические линии).

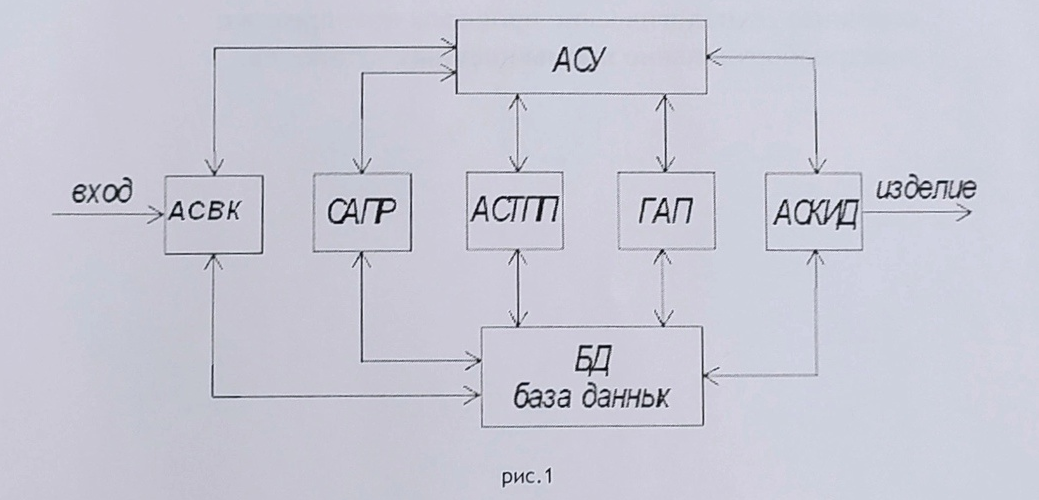
Полная автоматизация характеризуется автоматическим выполнением всех функций, для осуществления производственного процесса без непосредственного вмешательства человека в работу оборудования. В обязанности человека входит настройка машины или группы машин, включение контроль.

# Гибкие автоматизированные системы

ГАП предназначены для автоматизации тех. Процессов со сменным объектом производства, в том числе для единичного и мелкосерийного производства. Включает в себя основное и вспомогательное оборудование, работающее от одной СУ.

Использование ГАП в Японии и Европе даже в автоматизированном режиме и при недостаточном опыте работы приводит к увеличению отдачи станков на 80-200%, к сокращению продолжительности их обслуживания на 60-70%, что в свою очередь уменьшает время производственного процесса и стоимость живого труда примерно на 80%.

Создание ГАП является очень сложным и многоэтапным процессом, поэтому он по возможности автоматизируется, в результате образуется системное окружение ГАП.



Создание такой сложной системы (рис. 1) должно протекать поэтапно, тем более что эксплуатация ее обуславливает значительные изменения в структуре предприятия. Начальным этапом является разработка подсистемы АСВК (Входной контроль), САПР (NASTRAN, ADAMS), АСТПП(Технология), ГАП, АСКИД (Контроль и измерение детали). Основным элементом при создании такой подсистемы – унификация банка данных (БД) для информационного обеспечения функционирования подсистемы. Унификация БД предполагает, прежде всего применение групповой технологии, технологических процессов обработки деталей объединенных рядом технологических и конструктивных признаков

# Блок-схема алгоритма реализации групповой технологии в ГАП

1. Прогнозирование развития технологии оборудования, изделия

2. Анализ унификации изделий и их элементов

3. Анализ унификации ТП и их элементов

4. Группирование деталей

5. Проектирование ГТП (групповой технический процесс) на уровне компоновочных решений

6. Проектирование оснащения

7. Моделирование обработки детали

8. Проектирование производственных процессов

9. Моделирование работы производственных процессов по прогнозу

10. Анализ результатов моделирования, формирование плана корректировки деталей

В состав ГАП включены подсистемы: обработки, манипулирования, контроля и управления. Проектирование, разработка и внедрение осуществляется в установленном нормами и правилами порядке.

# САУ

Система автоматического управления представляет собой комплекс аппаратных и программных средств, предназначенный для обеспечения автоматического управления отдельными агрегатами или группой оборудования, связанных единым техническим процессом.

САУ выполняют функции автоматического управления, регулирования, контроля и защиты процессов, обеспечивающих безаварийную и длительную работу различного технологического оборудования.

# АСУ

Автоматизированная система управления (сокращённо АСУ) — комплекс аппаратных и программных средств, а также персонала, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия. АСУ применяются в различных отраслях промышленности, энергетике, транспорте и т. п. Термин «автоматизированная», в отличие от термина «автоматическая», подчёркивает сохранение за человеком-оператором некоторых функций, либо наиболее общего, целеполагающего характера, либо не поддающихся автоматизации. АСУ с Системой поддержки принятия решений (СППР) являются основным инструментом повышения обоснованности управленческих решений.

Важнейшая задача АСУ — повышение эффективности управления объектом на основе роста производительности труда и совершенствования методов планирования процесса управления. Различают автоматизированные системы управления объектами (технологическими процессами — АСУТП, предприятием— АСУП, отраслью — ОАСУ) и функциональные автоматизированные системы, например проектирование плановых расчётов, материально-технического снабжения и т. д.

Система автоматического управления средствами автоматизации

САУ обеспечивают работу по заранее заданной программе. Система управления автоматами может различаться по признакам:

1) по принципу синхронизации

2) по степени централизации управления

3) по методу воздействия

4) по числу управляемых координат

5) по виду программы-носителя

6) по наличию или отсутствию программной связи

СУ бывают централизованными и децентрализованными. У центр СУ технологический цикл управляется с центрального командного устройства (контроллер или ЭВМ). С центрального ком устройства независимо от действия и положения исполнительных органов.

# Особенности централизированных СУ:

1) Продолжительность рабочего цикла для каждого исполнительного органа является как правило величиной постоянной

2) Простота схем управления

3) Надежность в работе

4) Удобство обслуживания и наладки

5) Необходимость иметь дополнительные предохранительные устройства, то есть команды с центрального командного пункта подаются в независимости от действия и положения исполнительных рабочих органов.

Особенности децентрализованной системы управления

1) управление при помощи датчиков, выключаемых исполнительными рабочими органами

2) исполнительные органы связаны между собой так, что каждое последующее движение одного может происходить после окончания движения предыдущего

3) отсутствие сложной блокировки

4) датчики расположены в рабочей зоне оборудования нередко выходят из строя из-за попадания пыли и тд и выдает неправильную информацию об объекте

Наиболее важным и характерным признаком любой системы управления программного управления является способ задания программы обработки или движения. Наиболее высокой надежностью обладают следующие системы управления: централизованная, разомкнутая система управления без обратной связи, системы, которые обеспечивают надежную и точную синхронизацию всех движений рабочего цикла любой сложности.

Основой систем цифрового программного управления являются следящий привод, который состоит из устройства правления, преобразующего устройства, исполнительных двигателей, усилителя, устройства сравнения.

Процесс автоматического слежения заключается в сведении нулю ошибки согласования

Вопросы:

# Что такое непрерывные (контурные) системы управления?

Контурная система программного управления (КСПУ) - система управления, в которой командная информация содержит кроме признака звена, направления и величины перемещения еще и параметры траектории (контура), по которой осуществляется движение. Такие системы обозначают сокращенно СР (continuous path). Контурное управление обеспечивает перемещение звеньев манипулятора по непрерывной траектории, обладает высокой универсальностью и значительными технологическими возможностями. Промышленные роботы контурной СУ применяются для выполнения, как правило, основных, а не вспомогательных технологических операций

# Комбинированные системы управления

Комбинированные системы сочетают в себе достоинства систем управления по отклонению и по возмущению, что повышает точность управления. Действие неучтенных возмущений в комбинированных системах компенсируется или ослабляется управлением по отклонению.

В комбинированных системах компенсационная связь по основному возмущению (задающему воздействию) устраняет составляющую ошибки, вызываемую этим возмущением (изменением задающего воздействия), а в результате действия обратной связи уменьшаются ошибки, вызываемые второстепенными возмущающими воздействиями, по которым нет компенсационных связей. Если с помощью компенсационных связей не полностью устраняются ошибки, вызываемые основными возмущающими (задающими) воздействиями, то остаточные ошибки также уменьшаются с помощью обратной связи.

# Прямоугольные системы управления рабочими перемещениями

Управляют перемещением вдоль отрезков, параллельных направляющим станка. Прямоугольные системы предназначены для последовательного управления одной из двух взаимно-перпендикулярных координат. Такие системы применяются на токарных станках для управления обработкой деталей типа ступенчатых валиков, а на фрезерных — деталей с прямоугольным контуром.

# Лекция 2. Надежность программного обеспечения автоматизированных систем управления технологического процесса

Одной из основных частей СУ является ПО, представляющее собой совокупность взаимосвязанных и автономных программ, описаний, инструкций программистам и пользователя, тестов и тд.

Надежность ПО – св-во ПО своевременно выполнять заранее указанные условия эксплуатации, установленные ф-ии.

Надежность устанавливается по результатам работы ПО, те при динамической проверке всех программ на множестве входной информации. Некорректное ПО заведомо ненадежно, однако некорректное ПО может быть и надежным. Определение надежности базируется на понятии отказа программы.

Отказы ПО делятся на случайные и неслучайные. Неслучайные отказы обусловлены действием так называемых компьютерных вирусов. Случайные отказы ПО наблюдаются в случайные моменты времени работы ЭВМ. По своим последствиям эти отказы классифицируются на случайные сбои программ и устойчивые отказы ПО.

Сбой ПО – самоустраняющийся отказ программы, возникающий при некоторых возможных случайных состояний ЭВМ и информации, наблюдаемый пользователем случайный момент времени и исчезающий без вмешательства программиста.

Устойчивый отказ ПО наблюдается в случайный момент процессорного времени в форме результата в нормальном функционировании ЭВМ. Причиной отказа ПО служит некоторая систематическая ошибка программы, после устранения которой программистом данный отказ исчезает, те имеет место восстановление ПО.

Различают ошибки первичного и вторичного типа. Ошибки первичного типа связаны с неточностями в текстах программ и возникают при подготовке носителей и документаций ПО при записях кодов на алгоритмических языках и трансляции программ на машинный язык.

Ошибки вторичного типа во многом являются следствием первичных ошибок программ. К ним относят:

- вычислительные (неверная индексация и подсчет временных параметров, расхождение результата ручного и машинного счета, появление неустойчивых операций и тп)

- логические (пропуск логических условий, неверные краевые условия и др)

- сопряжения интерфейсов (межмодульных, программно-технических, информационных)

Ошибки первичного и вторичного типов порождаются на этапах разработки спецификаций на ПО; проектирования ПО; реализации программ.

Отказы ПО при его эксплуатации имеют ряд отличий от отказов технических элементов:

- Отказ ПО не приводит к разрушению или поломке программного элемента. Отказы ПО не связаны с физическим износом элемента (в частности носителя программ)

- Отказ ПО не коррелирован с процессорным и, тем более, астрономическим временем (с процессорным временем или числом прогонов ПО программ пользователем).

- При длительной эксплуатации ПО все его ошибки могут быть устранены и программы становятся абсолютно надежными. Если обозначить через N(t) число не выявленных ошибок ПО в произвольный момент времени t, то формально имеет место соотношение lim N(t) = 0, справедливое при условии, что в процессе восстановления программ в них не вносятся новые ошибки.

Опыт создания и эксплуатации ПО реального времени показывает, что при устранении одних ошибок вносятся другие.

Для описания надежности ПО используют такие же функциональные и числовые характеристики, как и при исследовании надежности технических элементов.

Основные показатели надежности ПО:

1. функция ненадежности или отказа ПО Q(t) = Вер (того, что отказ ПО появится до момента времени t).

2. функция надежности ПО Р(t) = Вер (того, что отказ ПО появится после момента времени t);

3. интенсивность отказов ПО (λ(t) = dQ/dt:

4. средняя наработка на отказ ПО: t = Ка =

Программное обеспечение АСУТП состоит из большого числа программ, подпрограмм и модулей, находящихся под управлением операционной системы реального времени или программы-диспетчера. Выполнение каждой из этих программ осуществляется последовательно во времени на одном и том же процессоре. Если эти программы имеют взаимные информационные связи или предназначены для получения одного результата у (вычисление одной функции), то в надежностном отношении такой программный комплекс представляет собой простую систему без избыточности и вероятность его безотказной работы равна произведению вероятностей безотказной работы каждой i-ой программы:

где m— общее число программ.

Надежность такого ПО определяется надежностью отказов самой «ненадежной» программы, имеющей наибольшее значение λ, i=1, m.

Для повышения надежности нерезервированного ПО следует в первую очередь улучшить характеристики самых «ненадежных» программ.

Резервирование: три вида. Временное, информационное и программное

Временное резервирование ПО заключается в многократном прогоне одних и тех же «ненадежных» программ и сравнении результатов расчета. Такое нагруженное резервирование позволяет устранять влияние случайных сбоев и выявлять случайные ошибки, требующие восстановления программ.

Информационное резервирование ПО основано на дублированных исходных и промежуточных данных. Эти данные могут проходить дополни тельную обработку, например, усреднение, до ввода в ПО, где они обрабатываются один раз; или обрабатываться одной и той же программой дважды, т.е. информационное резервирование подкрепляется временным.

Программное резервирование предусматривает наличие в ПО двух или больше разных программ для получения одного и того же результата у

или реализации одной функции. Здесь возможно нагруженное и ненагруженное резервирование.

Резервирование программного обеспечения распределенных АСУТП часто сопровождается аппаратурным резервированием.

# Лекция 3.

OPC в свете COM

Историческая справка

Сравнительно недавно, в1994 г., под эгидой Microsoft, была создана организация OPC Foundation. как определяет сама OPC Foundation, ее целью создания является разработка и поддержка открытых промышленных стандартов, регламентирующих методы обмена данными в реальном времени между клиентами на базе PC и ОС Microsoft. Сейчас эта организация насчитывает более 220 членов, включая почти всех ведущих поставщиков контрольно измерительного и управляющего оборудования для АСУ ТП. Достаточно назвать такие фирмы, как Siemens, Shneider Automation, Rockwell Software, Wonderware, Intellium, Ci Technologies, не говоря уже о самой Microsoft.

Технология

Как уже отмечалось выше, технология OPC реализована и продолжает реализовываться по типу разработка стандартов. OPC Foundation определяет направления, по которым ведутся разработки, и создает по этим направлениям комитетв. Комитеты делают следующее:

1) разрабатывают спецификации COM-интерфейсов и COM-объектов

2) присваивают им GUID

3) оформляют все в виде стандартов и опубликовывают

4) генерируют или создают вспомогательные файлы: idl-, h- и c- файлы для Custom-интерфейса; библиотеки типов для интерфейса автоматизации; заместители (proxy) и заглушки (stub) для поддержки межпроцессорного взаимодействия.

5) разрабатывают вспомогательные компоненты, например утилиту opeenum, позволяющую OPC-клиенту «увидеть» список всех OPC-серверов локальной сети

Спецификации

В настоящее время имеются следующие OPC-стандарты.

OPC Common Definitions and Interfaces общие для всех ОРС-спецификаций интерфейсов

Data Access Custom Interface Standard – спецификация COM-интерфейсов для обмена оперативными данными, программирование на C++

Data Access Automation Interface Standard – спецификация COM-интерфейсов для обмена оперативными данными, программирование на языках типа Visual Basic

OPC Batch Custom Interface Specification – спецификация COM-интерфейсов конфигурирования оборудования, программирование на C++

OPC Batch Automation Interface Specification – спецификация COM-интерфейс для конфигурирования оборудования, программирование на языках типа Visual Basic

OPC Alarms and Events Interface Specification – спецификации COM-интерфейсов для обслуживания событий (event) и нештатных ситуаций (alarm), программирование на C++

Historical Data Access Custom Interface Standard – спецификация COM-интерфейсов для работы с хранилищами данных, программирование на C++

OPC Security Custom Interface – спецификация COM-интерфейсов для обработки прав доступа к данным, программирование на C++

Как видим, перечень достаточно большой. Консорциум OPC Foundation пытается охватить все аспекты, связанные с взаимодействием с технологическим оборудованием. В разработке самих спецификаций принимают участие ведущие производители оборудования и систем автоматизации, которые стараются максимально учесть свой опыт и предоставить абсолютно «все необходимое» тому, кто будет использовать OPC. Далее мы проиллюстрируем это на примере спецификации Data Access (DA).

OPC-сервер (потребители «снизу»)

Кто же использует ОРС? Первая категория – производители оборудования автоматизации или ОЕМ (Original Equipment Manufacturer – поставщик комплексного оборудования). Предполагается, что тот, кто создает, например плату сбора данных, снабжает ее не только драйвером, но и реализует OPC-сервер, работающий с этой платой через драйвер или даже напрямую. Тем самым ОЕМ-производитель предоставляет стандартный доступ к своей плате.

Каждая переменная обладает свойствами. Различаются обязательные свойства, рекомендуемые и пользовательские. Обязательными свойствами, понятно, обязана обладать каждая переменная. Это, в-первых, текущее значение переменной, тип переменной и права доступа (чтение и/или запись). Во-вторых, очень важные свойства – качество переменной и метка времени. Технология OPC ориентирована на работу с оборудованием, а оборудование может давать сбои, так что корректное значение переменной не всегда известно ЩЗС-серверу, о чем и уведомляется клиент через качество (хорошее/плохое/неопределенное и дополнительная информация). Метка времени сообщает о том, когда переменная получила данное значение и/или качество. Еще одним обязательным свойством является частота опроса переменной OPC-сервером. Не совсем понятно, почему это свойство объявлено обязательным, так как не все ОРС-серверы работают в режиме опроса оборудования. Поэтому существуют серверы, не реализующие это свойство. Последним из обязательных свойств является описание переменной. Это строковое значение, содержащее информацию для пользователя о том, зачем нужна эта переменная.

Дополнительные свойства являются необязательными для реализации в ОРС-сервере. Это, например диапазон изменения (выход за границы диапазона должен специальным образом обрабатываться клиентом) и единица измерения. Есть перечень рекомендуемых свойств, но можно добавить и свои собственные.

Получение данных

Существует три основных способа получения ОРС-клиентом данных от ОРС-сервера: синхронное чтение, асинхронное чтение и подписка. При синхронном чтении клиент посылает серверу запрос со списком интересующих его переменных и ждет, когда сервер его выполнит. При асинхронном чтении клиент посылает серверу запрос, а сам продолжает работать. Когда сервер выполнил запрос, клиент получает уведомление (через интерфейс соответствующего COM-объекта, реализованного в клиенте!). И, наконец, в случае по подписки клиент передает серверу список интересующих его переменных, а сервер затем регулярно присылает клиент информацию об изменившихся переменных из этого списка (опять же, через интерфейс соответствующего COM-объекта клиента!). Эти списки в терминологии OPC называются группами. Каждый клиент может поддерживать одновременно много групп с разной скоростью обновления.

Что должен сделать производитель, если он задался целью обеспечить свой продукт стандартным интерфейсом? Он должен получить нужную спецификацию и прилагаемые программные компоненты. Затем он должен изучить COM-интерфейсы тех СОМ-объектов этой спецификации, которые относятся в ней к модели ОРС-сервера. И, наконец он должен посадить самого опытного программиста за Visual Studio, и тот с помощью ATL-библиотеки реализуют требуемые интерфейсы, а значит и OPC-сервер

ОРС-клиент (потребители «сверху»)

Правила игры заданы – ОРС-сервер поставляет данные, ОРС-клиент потребляет. Этим задается вторая категория пользовательских спецификаций ОРС. И к ней относятся в первую очередь те, кто реализует ПО

# Тема: Архитектура PROM

Все объекты системы управления в производственном процессе загружается желательно целевой ресурс

1. …

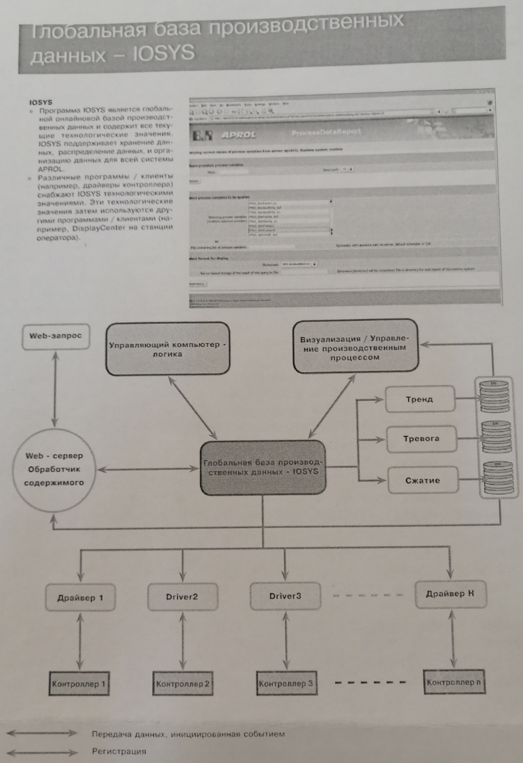
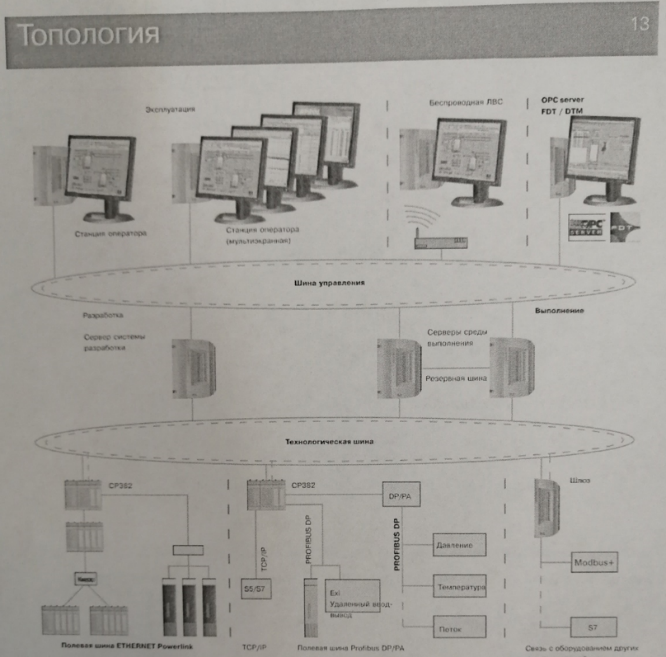
2 Сервер среды выполнения представляет базу данных системы управления производственным процессом, работающую в режиме реального времени, и содержит все технологические сообщения

3. Станция оператора используется для управления контролем производственного процесса на предприятии и отображает все данные для оператора

# Промышленная сеть Ethernet

С точки зрения общей системы управления производственными процессами ключевыми факторами являются надежность и готовность к передаче данных реального системы автоматизации производственных процессов (контроллеров) туда, где эти данные обрабатываются (сервер среды выполнения) и база данных, работающая в режиме реального времени и подстанция оператора, которые ответственна за контроль. В этом случае необходимо применять резервирование, чтобы обеспечить наивысшую возможную работоспособность сети.

# Глобальная база производственных данных



Изображение выглядит как текст

Автоматически созданное описание

# Определение и характеристики ETHERNET POWERLINK

ETHERNET Powerlink: Промышленная сеть Ethernet, работающая в режиме реального времени – это реальность

Ethernet имеет гарантированное будущее.

Базовые концепции Ethernet доступны уже более 30 лет. Длительные циклы амортизации, характерные для индустрии автоматизации. Требуют надежного основания.

Технология Ethernet хорошо известна.

Ethernet и соответствующие протоколы сегодня широко известны. Огромный объем имеющихся инструментов, программ и компонентов приведет к дальнейшему сокращению цен.

Ethernet – прозрачная технология.

Стандарты Ethernet объединяют различные протоколы передачи данных на основе IP. Интеграция информационных технологий и автоматизации при использовании Ethernet дает вам реальную функциональную совместимость с гибкостью Ethernet.

Ethernet является системой реального времени.

С Ethernet Powerlink, Ethernet также включает уровни датчиков и исполнительных механизмов, с временами цикла дом 200 мкс, и радикальное улучшение точности синхронизации, лучше одной микросекунды.

# Структура сети

В Ethernet Powerlink выделяются области реального времени и области, где режим реального времени не требуется. Это разделение соответствует типичной концепции машины и технологического процесса. Оно также удовлетворяет растущим требованиям к безопасности с целью предотвращения хакерских атак на уровне машин или ущерба от ошибочной передачи данных на верхних иерархических уровнях сети. Фактические требования режима реального времени. Менее критические с временной точки зрения данные прозрачным образом передаются между областью реального времени и стандартной областью с использованием стандартных IP кадров. А четкое разграничение между сетью машины и сетью предприятия с самого начала предотвращает потенциальные угрозы безопасности, сохраняя полную прозрачность данных.

Модель ISO. Стандартизация для связи между системами различных изготовителей. Нужно согласовывать типы коммуникационной среды. Процедуры доступа, протоколы. Основой для описания систем связи является общая стандартная модель передачи данных (соединение открытых систем), разработанная международной организация по стандартизации.

Можно выделить 7 уровней:

-физический

- канальный

- сетевой

- транспортный

- сеансовый

- представления

- прикладной

Уровни Ethernet Powerlink

# Уровень 1

Изначально протокол определялся для физического уровня на основе 100Base-TX (Fast Ethernet, 100 Мбит/с по витой паре). В конце 2006 года был разработан вариант Ethernet Powerlink для Gigabit Ethernet (1 Гбит/с).

В сетях Ethernet Powerlink (в сегменте сети, в котором требуется сеть реального времени) рекомендуется использовать концентраторы вместо коммутаторов для минимизации задержек и джиттера. Для проектирования сетей Ethernet Powerlink применяются руководства IAONA (*Industrial Ethernet Planning and Installation Guide*) в части прокладки кабелей. В качестве разъемов применяются как распространенные Ethernet-разъемы 8P8C (RJ45) так и М12.

# Уровень 2 – MAC, коммутатор…

Уровень 2 описывает механизмы, использованные для безопасной передачи данных. На этот уровень включены стандарты Ethernet для управления доступом к среде (MAC). На этом уровне работают мосты и коммутаторы. В отличие от концентраторов, эти устройства интеллектуальны и анализируют полученные кадры Ethernet. Коммутатор анализирует распределение станций на сети, используя адреса источника и адресата. Таким образом, он может посылать полученные пакеты.

# Уровень 3 – IP, маршрутизатор…

Вторая часть TCP/IP – межсетевой протокол IP. Он включен на уровень 3 и ответственен за реальную передачу данных по сети. На этом уровне работают маршрутизаторы. Они соединяют подсети, используя IP-адреса, которые уникальны в сети. Поэтому коммутаторы на уровне 3 – это маршрутизаторы.

# Уровень 4 – TCP, UDP…

Уровень 4 (транспортный уровень) включает первую часть термина TCP/IP. Протокол управления передачей. TCP можно сравнить с UDP (Протоколом пользовательских датаграмм). Оба реализуют обмен данными между тремя нижними уровнями (системой транспортировки) и тремя верхними уровнями (прикладной системой). Единственное отличие – метод, которым обрабатывается поток данных. TCP предлагает больше механизмов защиты, чем UDP, но медленнее из-за обширной обработки данных.

# Уровень 5, Уровень 6, Уровень 7 – Сеанс, презентация, приложение…

Уровни 5, 6 и 7 формируют прикладную систему. Они ответствененны за структурирование связи, представление данных. Все приложения, использующие TCP/IP, включены на эти уровни.

Уровень протокола

Сетевой уровень – уровень 3

На сетевом уровне устанавливается логическое соединение между передатчиком и приемником. Работа сетевого уровня важна для маршрутизации сети, когда данные передаются более, чем между двумя точками. Кроме маршрутизации сети, выполняется управление пакетами. Таким образом, транспортный протокол управляет работой всей сети.

Транспортный уровень – уровень 4.

Транспортный уровень отделяет уровни, технологически зависимые от верхних уровней. На этом уровне логические адреса преобразуются в физические адреса, определяется маршрутизация сети и распределение пакетов.

Сеансовый уровень – уровень 5.

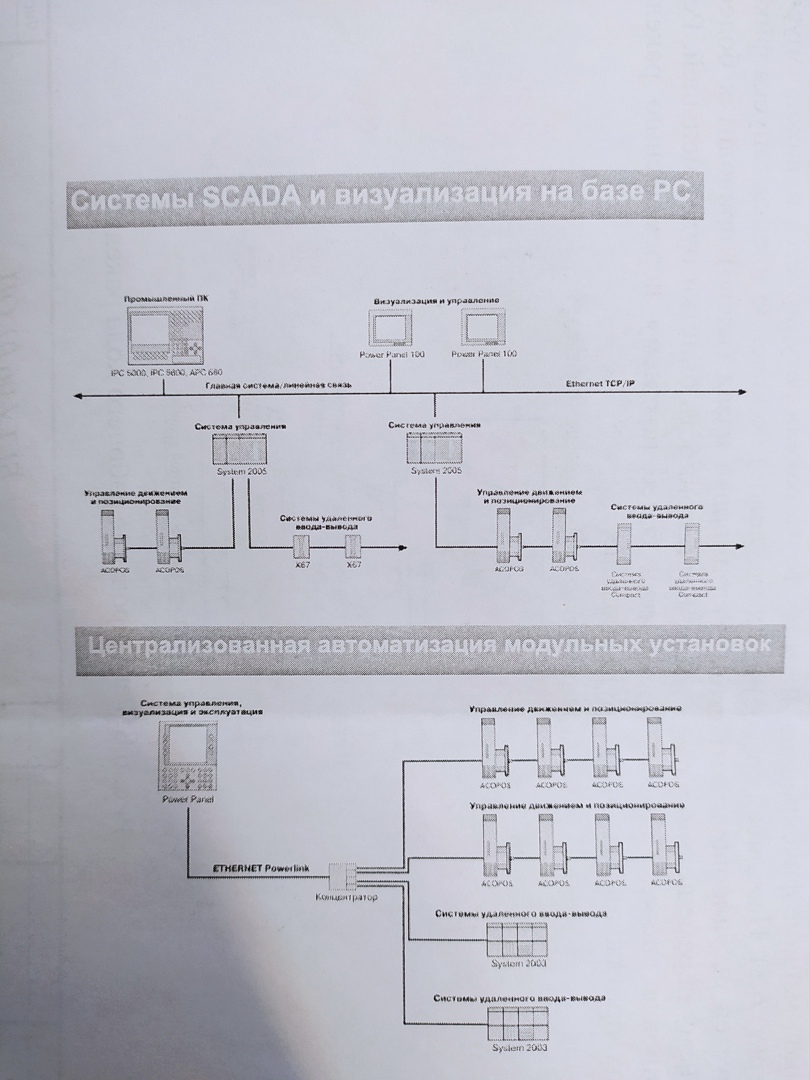
Этот уровень отвечает за синхронизацию задач на уровнях передачи и приема и содержит такие функции, как установка соединений, организация буферизации данных, проверка права доступа и контроль соединений.

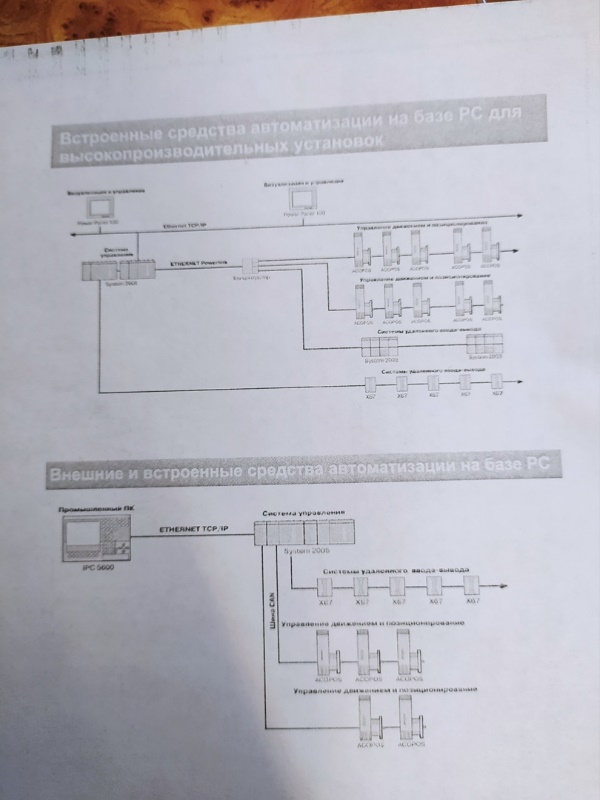
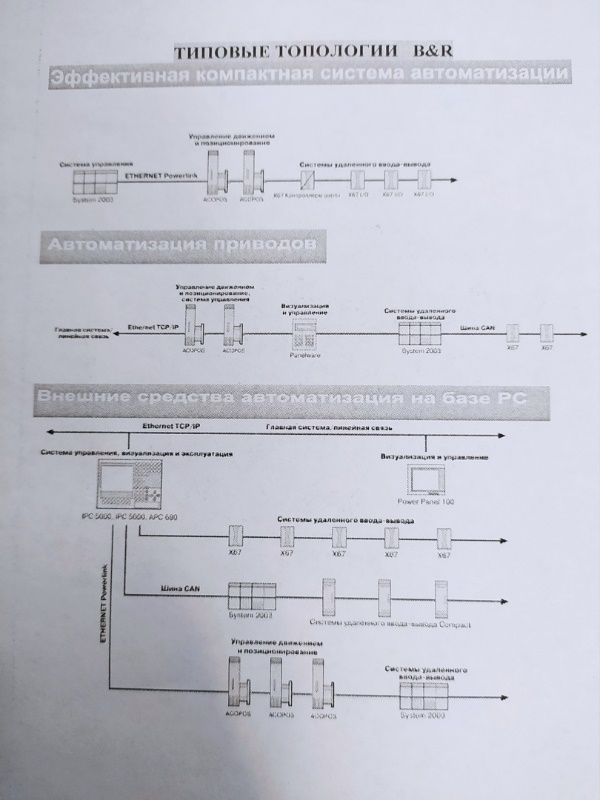
Уровень представления – уровень 6.

На этом уровне формат данных узла преобразуется с учетом синтаксиса, используемого в сети. Дополнительные задачи включают, например, кодирование или сжатие данных.

Прикладной уровень – уровень 7

Этот уровень обеспечивает интерфейс между пользователем и сетью для работы с сетевыми приложениями без дополнительных специальных знаний по функционированию сетей.





Режимы работы

Совместимые устройства может работать в следующих рабочих режимах:

- базовый. Устройство работающее в сетях Ethernet когда не требуется передача данных в режиме реального времени. Этот режим используется по умолчанию после включения устройства.

- в ходе запуска системы или после подсоединения устройства к существующей сети, данные конфигурации загружаются по каналу (асинхронно)

- режим Ethernet Powerlink. После завершения процедуры загрузки устройство работает в реальном времени. Управляющий узел проверяет синхронизацию. Время цикла зависит от объема данных, а также от количества узлов.

Базовый цикл состоит из следующих фаз:

- фаза запуска. Все сетевые узлы синхронизируются с часами управляющего узла.

- изохронная фаза. Управляющий узел присваивает каждому узлу фиксированное временное окно для передачи критических данных. На этй фазе все другие узлы всегда могут принимать все данные. Принцип издатель-подписчик

- асинхронная фаза. Управляющий узел представляет право передавать конкретные данные одному конкретному узлу. На этой фазе используются стандартные протоколы

Уровень передачи данных

Детерминированная синхронизация достигнута применением графика циклической синхронизации для всех соединенных узлов при обращениях к физическим уровням. График разделен на изохронную фазу и асинхронную фазу

На изохронной фазе передаются критические данные

Асинхронная фаза резервирует пропускную способность для некритических с временной точки зрения данных.

Что предотвращает столкновение. В системе Ethernet Powerlink используется уникальный мак-адреса для каждого устройства. Кроме того, в области реального времени присвоены идентификаторы устройств EPL.

# Profibus протокол обмена. Пример программы

Для обмена данными между узлами автоматизированной системы под управлением оборудования Siemens реализуется как по полевой шине, которая называется Profibus, так и в промышленной сети Profinet.

Существует несколько протоколов обмена по шине Profibus:

- PROFIBUS DP — реализация этого протокола подразумевает связь с удаленными подчиненными устройствами, в случае с PROFINET этому протоколу соответствует протокол PROFINET IO.

- PROFIBUS PA — является по сути тем же PROFIBUS DP, только используется для взрывобезопасных исполнений передачи данных и питания (аналог PROFIBUS DP с другими физическими свойствами). Для PROFINET взрывобезопасного протокола по аналогии с PROFIBUS пока не существует.

- PROFIBUS FMS — предназначен для обмена данными с системами других производителей, которые не могут использовать PROFIBUS DP. Аналогом PROFIBUS FMS в сети PROFINET является протокол PROFINET CBA.

Преимущества Profibus:

**Независимость от поставщика**

Устройства **PROFIBUS** предлагаются широким спектром квалифицированных поставщиков. Это позволяет пользователю выбирать наиболее подходящего поставщика и наилучшие изделия.

**Признанность**

**PROFIBUS** — это признанная технология, имеющая большое число приложений в автоматизации строительства, управлении производством, технологическими процессами и приводами. В настоящее время более 150 поставщиков уже открыли этот стремительно растущий рынок и предлагают большое количество **PROFIBUS**-продукции.

**Сертифицируемость**

Тестирование устройств **PROFIBUS** на соответствие стандарту и совместимость для совместной работы, осуществляемое уполномоченными тестовыми лабораториями, и сертифицирование PNO дают пользователям гарантию качественной работы в сетях, построенных на изделиях разных поставщиков.

**Ориентация на будущее**

Признанными на рынке автоматизации лидерами поддерживается целый спектр новейших изделий **PROFIBUS**. Целью PNO является поддержка стандарта **PROFIBUS** и координация дальнейшего его развития между пользователями и производителями. Ваш выбор **PROFIBUS** – это гарантия того, что вы работаете с промышленной шиной будущего. Сеть **PROFIBUS** построена в соответствии с многоуровневой моделью ISO 7498 — OSI.

**Поддерживаемые стандарты**

Открытость и независимость от производителя гарантирует стандарт EN 50 170, все остальное реализовано в соответствии со стандартом DIN 19245 (а именно: техника передачи данных, методы доступа, протоколы передачи, сервисные интерфейсы для уровня приложений, спецификация протоколов, кодирование, коммуникационная модель и т. д.). С помощью **PROFIBUS**, устройства разных производителей могут работать друг с другом без каких-либо специальных интерфейсов.

**Протоколы сети PROFIBUS**

Одни и те же каналы связи сети PROFIBUS допускают одновременное использование нескольких протоколов передачи данных:

• PROFIBUS DP (Decentralized Peripheral - Распределенная периферия) — протокол, ориентированный на обеспечение скоростного обмена данными между системами автоматизации (ведущими DP-устройствами) и устройствами распределённого ввода-вывода (ведомыми DP-устройствами). Протокол характеризуется минимальным временем реакции и высокой стойкостью к воздействию внешних электромагнитных полей. Оптимизирован для высокоскоростных и недорогих систем. Эта версия сети была спроектирована специально для связи между автоматизированными системами управления и распределенной периферией. Электрически близка к RS-485, но сетевые карты используют 2-х портовую рефлективную память, что позволяет устройствам обмениваться данными без загрузки процессора контроллера.

• PROFIBUS PA (Process Automation - Автоматизация процесса) — протокол обмена данными с оборудованием полевого уровня, расположенным в обычных или Ex-зонах (взрывоопасных зонах). Протокол отвечает требованиям международного стандарта IEC 61158-2. Позволяет подключать датчики и приводы на одну линейную шину или кольцевую шину.

• PROFIBUS FMS (Fieldbus Message Specification - Спецификация сообщений полевого уровня) — универсальный протокол для решения задач по обмену данными между интеллектуальными сетевыми устройствами (контроллерами, компьютерами/программаторами, системами человеко-машинного интерфейса) на полевом уровне. Некоторый аналог промышленного Ethernet, обычно используется для высокоскоростной связи между контроллерами и компьютерами верхнего уровня и используемыми диспетчерами. Скорость до 12 Мбит/с.

Все протоколы используют одинаковые технологии передачи данных и общий метод доступа к шине, поэтому они могут функционировать на одной шине.

Для программирования и проектирования Profibus-DP используется программный пакет STEP 7. Это базовое программное обеспечение для программирования и проектирования систем SIMATIC S7.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1– SIMATIC Manager

# Общие сведения о периферийных устройствах и их связях с микроэвм

# 1.1. Назначение и классификация

Для эффективного использования ПЭВМ большое значение имеет оборудование их периферийными устройствами. Функциональные возможности ПЭВМ в значительной мере зависят от применяемого набора и технических характеристик периферийных устройств.

Периферийные устройства ПЭВМ предназначены для ввода, вывода и хранения данных с последующим их вводом в ПЭВМ, являются средствами коммуникации ПЭВМ с внешними источниками и потребителями информации и обеспечивают согласование сигналов внешних объектов и сигналов, используемых в ПЭВМ. В процессе своей работы они лишь преобразуют данные из одной формы представления информации в другую, не меняя их содержания.

Периферийные устройства ПЭВМ различаются по назначению, виду обслуживаемых объектов, физической природе обрабатываемых сигналов и носителей информации, принципу действия, техническим и эксплуатационным характеристикам, конструктивному исполнению и другим признакам. Все периферийные устройства ПЭВМ обычно подразделяют на три большие группы; 1) устройства связи ПЭВМ с человеком-пользователем; 2) устройства связи с объектами контроля и управления; 3) устройства для длительного хранения информации большой емкости.

К первой группе относятся устройства ввода-вывода и передачи информации, необходимые для общения человека с машиной, и электронные функциональные модули, их обслуживающие. Это прежде всего клавиатуры, различные планшеты (графоповторители) для работы с чертежами, устройства управления курсором дисплея (манипуляторы типа «мышь», «джойстик» и т. п.), видеотерминалы (дисплеи), печатающие устройства, сканеры, графопостроители. В последние годы быстро развиваются средства ввода-вывода человеческой речи.

Во вторую группу периферийных устройств включают устройства связи ПЭВМ с объектами и интерфейсами приборных систем: различные датчики и исполнительные органы, цифроаналоговые (ЦАП) и аналогово-цифровые (АЦП) преобразователи — устройства, необходимые для преобразования непрерывных сигналов с датчиков в цифровые сигналы и обратного преобразования при выдаче информации на исполнительные органы. С помощью периферийных устройств данной группы ПЭВМ приспосабливаются для управления технологическими процессами и оборудованием, автоматизации контроля и измерений, сбора данных и т. п. Все больше ПЭВМ оборудуется также средствами телеобработки, позволяющими подключать ПЭВМ к локальным и распределенным сетям ЭВМ. Значительный интерес представляют модемы — периферийные устройства, позволяющие передавать информацию от одной ПЭВМ к другой по телефонному каналу.

Третью группу периферийных устройств ПЭВМ составляют внешние запоминающие устройства (ВЗУ): накопители на гибких и жестких магнитных дисках, на магнитных лентах, на микросборках ЦМД. Серьезными конкурентами накопителям на магнитных носителях информации в ПЭВМ являются накопители на магнитооптических и оптических дисках.

Характерна также классификация периферийных устройств в зависимости от выполняемых ими в ПЭВМ функций. В соответствии с ней периферийные устройства ПЭВМ можно разделить на две категории. К первой относятся периферийные устройства, без которых практически невозможно функционирование вычислительной системы: клавиатуры, накопители на магнитных дисках, дисплеи, печатающие устройства, т. е. периферийные устройства, входящие в базовый комплект ПЭВМ. Такие периферийные устройства часто называют системными (общего назначения). Ко второй категории относят устройства, которые предоставляют дополнительные возможности пользователю профессиональной ПЭВМ, позволяя создавать рациональную конфигурацию вычислительной системы в зависимости от ее конкретной профессиональной ориентации (накопители на магнитной ленте, манипуляторы, графические планшеты, графопостроители, сканеры, коммуникационные и интерфейсные адаптеры, средства мультимедиа и т.д.). Подобные периферийные устройства называют дополнительными.

Особенности подключения периферийных устройств к ПЭВМ. Подключение различных по функциональному назначению, принципу работы, интерфейсу и конструктивному исполнению периферийных устройств к центральному процессору ПЭВМ осуществляется через электронные функциональные модули. Эти модули выполняют обычно согласование интерфейсов периферийного устройства и системной шины ПЭВМ, буферизацию информации, а также некоторые локальные функции, часто сложные, освобождая от них центральный процессор.

Электронные функциональные модули, обслуживающие работу периферийных устройств в составе ПЭВМ, аналогично самим периферийным устройствам, могут быть разделены на две группы.

Модули первой группы управляют работой системных периферийных устройств,их называют контроллерами (управляющими устройствами). Модули второй группы приспосабливают дополнительные периферийные устройства для работы с ПЭВМ. Поскольку модули служат для адаптирования внешних сигналов периферийных устройств к сигналам системной шины ПЭВМ, их часто называют адаптерами. Такое деление электронных функциональных модулей ПЭВМ довольно условно, поэтому в литературе понятия «контроллер» и «адаптер» в большинстве случаев считаются синонимами.

Принцип подсоединения периферийных устройств к ПЭВМ (как системных, так и дополнительных) единый: через системную шину к центральному процессору подключается электронный функциональный модуль, управляющий работой данного периферийного устройства, этот же модуль электрически соединяется кабелем с самим периферийным устройством.

Для подключения к ПЭВМ периферийных устройств, производимых различными предприятиями (фирмами), наборы сигналов, передаваемых по кабелю и разъемным соединителям, техническое исполнение, а также правила обмена информацией между периферийным устройством и адаптером образуют систему, называемую интерфейсом периферийного устройства. Поэтому иногда электронные функциональные модули называют также интерфейсами, что, однако, не отражает сути последнего понятия. В профессиональной ПЭВМ обычно реализуются стандартные или унифицированные интерфейсы для подключения клавиатуры, дисплея, внешних запоминающих и других устройств.

Электронные функциональные модули, адаптирующие к ПЭВМ периферийные устройства, осуществляют не только управление периферийным устройством, но и обмен данными через системную шину между центральным процессором и периферийным устройством. Они представляют собой сложные устройства и часто содержат микропроцессорные схемы. Для управления процессом обмена информацией предварительно составляются и записываются в память специальные управляющие программы — драйверы. При необходимости выполнения операции ввода-вывода работа текущей программы прерывается и управление передается драйверу (через системную шину необходимая команда поступает к адаптеру, который через соответствующий интерфейс взаимодействует с подключенным периферийным устройством). По окончании работы периферийного устройства управление снова передается основной программе.

# 1.2. Критерии выбора периферийных устройств

При выборе периферийных устройств и дополнительных электронных функциональных модулей пользователь ПЭВМ должен принимать во внимание их совместимость с используемой ПЭВМ, технические параметры (скорость обмена информацией, мощность, надежность работы и т. д.), которые должны соответствовать характеристикам ПЭВМ, а также наличие служб технического обслуживания и ремонта, запасных частей и других принадлежностей (например, НГМД, ленты и бумаги для печати).

Важное значение для профессиональных ПЭВМ имеет возможность использования средств машинной графики — аппаратуры (графических дисплеев, графопостроителей, устройств ввода и обработки графических изображений), методов и приемов для преобразования данных в их графическое представление и наоборот.

Характеристики периферийных устройств (стоимость, масса и габаритные размеры, быстродействие, надежность и др.) в основном определяют характеристики всей вычислительной системы на базе ПЭВМ. Во многом это связано с тем, что традиционно периферийные устройства содержат большое количество громоздких и дорогостоящих, медленнодействующих и малонадежных электромеханических, а также механических узлов и деталей. Применение указанных узлов в периферийных устройствах сдерживает возможности комплексной микроминиатюризации ПЭВМ и обусловливает невысокие эксплуатационные характеристики периферийных устройств. В настоящее время в мире разрабатываются более совершенные малогабаритные периферийные устройства для ПЭВМ, в том числе основанные на нетрадиционных принципах преобразования и хранения информации.

# 1.3. Кодирование информации

Так как большинство внешних устройств (ВУ) позволяет человеку общаться с ЭВМ на привычном ему языке слов и десятичных чисел, а ЭВМ понимает лишь язык физических состояний, кодируемых двоичными числами, то в ВУ обычно производится кодирование (декодирование) информации, пересылаемой в (из) микроЭВМ. Для кодирования русских (33 буквы) и латинских (26 букв) прописных и строчных букв, а также различных графических (например, символа %) и служебных (например, код для перевода строки в пишущей машинке) символов необходимо иметь более чем 27=128 различных кодов. Поэтому кодирование обычно производится в 8-битовом (1-байтовом) коде обмена информацией.

Так как ВУ имеют достаточно много механических элементов, то они менее надежны, чем электронные схемы процессора и памяти. Причиной неправильного восприятия кода считываемого символа могут быть пыль на считывающих головках накопителей на магнитных лентах или дисках, царапины на самих лентах и дисках и т. п. Искажения из-за электрических помех возникают и при передаче данных между процессором (памятью) и ВУ (контроллером ВУ). Короче говоря, если надо ввести, вывести или передать символы, то могут появиться ошибки. Для обнаружения таких ошибок (а иногда и для их исправления) коды символов преобразуют в какую-либо стандартную форму (например, добавляют еще один бит, чтобы в новом коде было четное число единиц). При передаче такого стандартного кода производится определенного вида контроль (например, контроль четности), и по его результатам принимается решение об использовании полученного символа.

# 1.7. Устройства для связи микроЭВМ с объектами управления

МикроЭВМ управляют станками и игровыми автоматами, ходом научных экспериментов и доильными аппаратами, автомобилями и роботами. В этих (а также других) устройствах, приборах и системах используется много серийных электрических датчиков и исполнительных органов (например, электродвигателей), работающих с непрерывными (аналоговыми) электрическими сигналами. Для связи таких датчиков и исполнительных органов с микроЭВМ требуется преобразовать аналоговый сигнал в число, пропорциональное амплитуде этого сигнала, и наоборот.

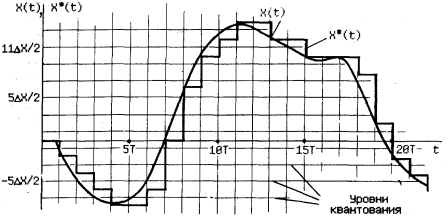


Рисунок 1.3 – Дискретизация и квантование непрерывного сигнала X

Процедура аналогоцифрового преобразования состоит из двух этапов: дискретизации по времени (выборки) и квантования по уровню. Процесс дискретизации состоит из измерения значений непрерывного сигнала *Х(t)*только в дискретные моменты времени 0, *Т, 2Т, ...,* отстоящие друг от друга на величину периода дискретизации *Т* (рис. 1.3).

Такой измерительный процесс, если он реализуется без необходимых мер предосторожности, может привести к возникновению значительных погрешностей. Важно иметь достаточное число отсчетов за единицу времени для правильного представления существенной информации, содержащейся в высокочастотной составляющей этого сигнала. В то же время частые измерения требуют увеличения быстродействия преобразования и обработки, а следовательно, сложности и стоимости преобразователя. Поэтому темп дискретизации необходимо поддерживать на минимально допустимом уровне, при котором составляющая общей погрешности, обусловленная процессом дискретизации, не превышает установленных пределов.

Для квантования диапазон изменения входного сигнала подразделяется на равные интервалы (уровни квантования): ...,-5DХ/2, -ЗDХ/2, - DХ/2, - DХ/2, 3DХ/2, 5DХ/2,**...,**гдеDХ—интервал (шаг квантования). Операция квантования сводится к определению того интервала, в который попало дискретизированное значение Х(t***),*** и к присваиванию выходному значению Х\*(t*)* цифрового кода, соответствующего значению центра найденного интервала. При такой замене может быть допущена ошибка, равная DХ/2. Для ускорения процесса преобразования, упрощения и удешевления преобразователя надо выбирать максимально допустимый шаг квантования, при котором еще не появляются большие погрешности.

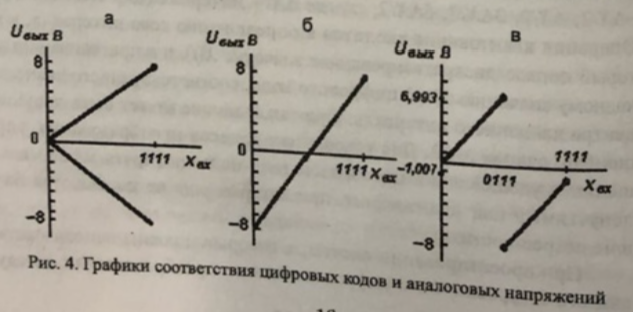
При проектировании систем, в которых должны использоваться аналогоцифровые или цифроаналоговые преобразователи, следует иметь хотя бы минимальное представление о принципах работы этих преобразователей.

Цифроаналоговый преобразователь (ЦАП) – функциональный узел, однозначно преобразующий кодовые комбинации цифрового сигнала в значения аналогового сигнала. Основой для нахождения однозначного соответствия может служить напряжение на выходе ЦАП:

где E0 – опорное напряжение; X{x1, x2, …, xn} – цифровой код; xi(1,2,…,n) принимают значения 0 или 1. При определенном E0 каждому xi на выходе устройства соответствует напряжение Uвыхi.

В цифроаналоговых преобразователях используют три основных двоичных кода: прямой, смещенный и дополнительный. Графики соответствия цифровых кодов Xвх и аналогового напряжения Uвых при прямом (а), смещенном (б) и дополнительном (в) кодах показаны на рис.4.

Наиболее просто определяется соответствие цифровых и аналоговых величин при прямом коде. Этот код удобен при преобразовании сигналов следящих систем, так как при переходе через нуль не меняются старшие разряды кода, что позволяет реализовывать линейный переход от малых положительных к малым отрицательным выходным напряжениям. Для преобразования как положительных, так и отрицательных кодов используют знаковый разряд, который управляет переключением выходного напряжения ЦАП.



Для исключения коммутирующих элементов из схемы ЦАП используют смещенный код, являющийся наиболее простым для реализации в схеме преобразователя.

При применении дополнительного кода положительные числа преобразуются так же, как для прямого кода, а отрицательные – двоичным дополнением соответствующего положительного числа (инверсия всех разрядов с последующим добавлением единицы в младший разряд).

Базовая схема ЦАП, реализующего выражение (1), показанная на рис.5, а, содержит источник опорного напряжения E0, матрицы двоично-весовых резисторов, набор ключей и дифференциальный операционный усилитель. Основные недостатки этой схемы определяются необходимостью применения резисторов с большим диапазоном номиналов, например 1R-1024R для 10-разрядного ЦАП.

Рациональным способом уменьшения количества номиналов резисторов является использование резистивной (лестничной) матрицы R-2R, изображенной на рис. 5, б. Выражение (1) реализуется схемой ЦАП (рис. 5, а) непосредственно, так как соотношение Rос/Ri равно весу соответствующего xi. Схема ЦАП на основе матрицы R-2R также реализует выражение (1). Так как потенциал суммирующей точки операционного усилителя равен нулю, то, анализируя эквивалентную схему, изображенную на рис. 6, а, можно записать для точки a

где Ki – коэффициент передачи E0 в точку a от разряда цифрового кода с соответствующим индексом.

# Проектирование программного обеспечения полиграфических машин и систем обработки информации

Новая аппаратная база, микроЭВМ контроллеры позволяет реализовать в системах автоматического управления оптимальные законы управления и этим обеспечивает их более высокие статические и динамические показатели. Типовыми режимами работы электропривода являются:

- разгон

- торможение

- реверс

- стабилизация определенных параметров при сильно изменяющихся внешних воздействиях

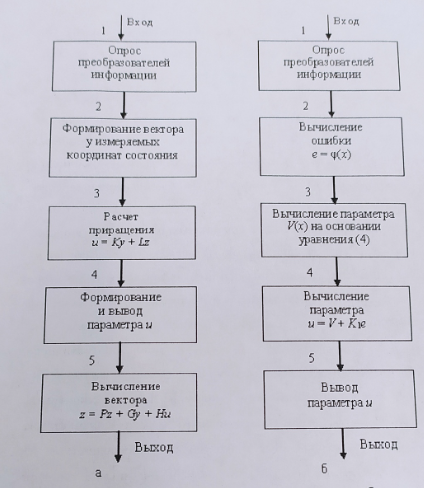
- слежение, в том числе программное при наличии внешних воздействий

- согласование конечных состояний, режимов стабилизации или слежения

Если в электроприводе последовательно реализуется хотя бы 2 из указанных режимов, то параметры стационарного регулятора с жесткой структурой следует выбирать с учетом обеспечения приемлемого качества процесса этих режимов. Наивысшее качество процессов может быть достигнуто в том случае, когда для каждого режима работы электропривода используется регулятор соответствующей структуры настройка которого зависит от изменяющихся параметров объекта управления и внешних воздействий. При использовании управляющих ЭВМ структура регулятора изменяется простым переходом от одной программы расчета управляющего воздействия к другой.

Рассмотрим возможные структуры и методы проектирования цифровых регуляторов для каждого режима работы электропривода.

Для режимов стабилизации и слежения при наличии возмущений может быть использован один и тот же типовой алгоритм управления, схема которого изображена на рис 1, а. В блоке 1 этого алгоритма опрашиваются преобразователи информации, используемые в системе управления и измеряющие состояние электропривода и внешних воздействий. Формирование вектора y в блоке 2 заключается в переходе от абсолютных значений выходных величин преобразователей информации к отклонениям относительно номинальных (или программных) значений; здесь же вектор y включается значение управляющего воздействия, рассчитанное в блоке 3 на предыдущем такте. Такое расширение вектора измеренных координат позволяет компенсировать запаздывание, обуславливаемое управляющей ЭВМ. Это запаздывание равно времени выполнения расчетов в блоках 1-4.



В блоке 3 рассчитывается приращение u управляющего воздействия относительно номинального управления, обеспечивающего номинальный (или программный) режим стабилизации. Исходными данными для этого являются вектор y измеряемых координат и вектор z, являющийся оценкой неизмеряемых координат состояния. Матричные коэффициенты K и L являются матрицами-строками, если электродвигатель управляется по одному входу (по цепи якоря или по цепи возбуждения), и имеют две строки, если электродвигатель управляется по двум входам. Выбором матриц K и L обеспечиваются заданные свойства собственного движения системы управления электроприводом, а также компенсируются доминирующие возмущающие воздействия. Для расчета матриц K L используются методы комбинированного, модального и оптимального управления.

В блоке 4 форрмируется полное управляющее воздействие как сумма номинального и рассчитанного отклонений управления; это полное управляющее воздействие выводится на цифроаналоговоый преобразователь.

В блоке 5 вычисляется вектор оценки неизмеряемых координат состояния. Эта оценка используется для расчета управляющего воздействия в следующем такте. Выбором матриц P, G H можно обеспечить требуемую скорость сходимости оценки, фильтрацию ошибок измерения, экстраполяцию отсчетов, снимаемых с преобразователей информации, а также требуемый порядок астатизма системы.

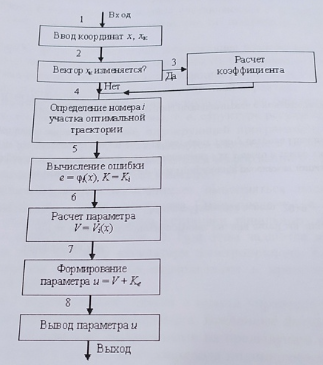
Одновременно требуется обеспечить выполнение некоторых соотношений координат вектора состояния (фазовых координат) x вида φ(x)=0, определяющих, например, условия постоянства частоты вращения двигателя, мощности и т.п. В пространстве состояний рассматриваемое уравнение задает некотороые многообразие, в частности поверхность, отклонение от которого характеризуется ошибкой.

Алгоритм управления, обеспечивающий минимизацию этой ошибки, имеет вид:

Аналитическое выражение для составляющей V(x) может быть получено из условия e=0. Матрица K1 обеспечивает желаемое качество отработки ошибки и выбирается с помощью математического обеспечения, используемого при синтезе регуляторов стабилизации или слежения.

На основании сказанного блок схема алгоритма управления согласованеим конечного и начального состояний режимов разгона и стабилизации имеет вид, изображенный на рисунке 1, б. В блоке 1 опращиваются преобразователи информации и в ЭВМ вводятся координаты вектора состояния x электропривода.

Схема типового алгоритма такого оптимаьного управления приведена на рис.2.



В блоке 1 осуществляется ввод координат вектора состояния x с преобразователей информации, а также ввод с пульта управления координат вектора конечного состояния xк, которое должно быть достигнуто в результате разгона, торможения или реверса. Если число преобразователей информации меньше числа координат вектора х, то после блока l вводится блок оценки координат вектора x. Если с пульта управления поступает новая уставка xk (блок 2), то в блоке 3 рассчитываются новые значения коэффициентов аппроксимирующих функций φi(x)=0.

В блоке 4 путем проверки логических условий, полученных на этапе синтеза, определяется номер I участка оптимальной траектории.

Затем рассчитывается отклонение e изображающего многообразия и выбирается коэффициент усилиения стабилизирующей составляющей управляющего воздействия (блок 5).

В блоке 6 вычисляется основная составляющая V управляющего воздействия, обеспечивающая движение по выбранному многообразию.

В блоке 7 формируется полное управляющее воздействие, которое затем выводится на объект управления (блок 8). Составляющая управления Ke обеспечивает ликвидацию скользящих режимов в том случае, когда основная составляющая V имеет релейный характер.

Все рассмотренные программы выполняются на одной и той же управляющей ЭВМ. Тип программы, т.е. структура регулятора выбирается с помощью специальной планирующей программы (рис.3), исходными данными для которой являются команды с пульта управления, а также показания преобразователей информации о состоянии объекта управления и внешних воздействий.

Планирующая программа начинает выполняться после пуска ЭВМ и прекращает – после ее остановка. Сразу после пуска ЭВМ выполняется специальная программа установки начального состояния регистров и областей ОЗУ (блок 2). При этом задаются начальные условия для всех структур регулятора электропривода. Остальные блоки программы выполняются периодически в процессе работы управляющей ЭВМ.

В блоке 3 опрашиваются уставки с пульта управления, а также показания преобразователей информации. Введенные данные в блоке 4 сравниваются с данными, полученными на предыдущем цикле. Изменяющиеся условия работы электропривода индицируются присвоением логической переменной L единичного значения (блок 6). При этом блоке 5 запоминаются новые условия работы. Если же условия работы не изменяются, то в блоке 8 логической переменной L присваивается нулевое значение.

В блоке 9 в результате проверки логических условий, определяющих последовательность переключения структуры регулятора, находится номер I программы расчета управляющего воздействия, которая выполняется в данное время. При изменившихся условиях работы (L=1 в блоке 10) в блоке 11 рассчитываются параметры программы Пi (элементы матричных коэффициентов алгоритма стабилизации, коэффициенты уравнения многообразия алгоритма согласования и т.п.). Затем выполняется одна из описанных программ Пi расчета управляющего воздействия.

# Этапы проектирования и внедрения АСУ ТП

Стандарт [ГОСТ](https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2) 34.601-90 предусматривает следующие стадии и этапы создания [автоматизированной системы](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0):

1. Формирование требований к АС

* Обследование объекта и обоснование необходимости создания АС
* Формирование требований пользователя к АС
* Оформление отчета о выполнении работ и заявки на разработку АС

2. Разработка концепции АС

* Изучение объекта
* Проведение необходимых научно-исследовательских работ
* Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователей
* Оформление отчета о проделанной работе

3. Техническое задание

* Разработка и утверждение технического задания на создание АС

4. Эскизный проект

* Разработка предварительных проектных решений по системе и её частям
* Разработка документации на АС и её части

5. Технический проект

* Разработка проектных решений по системе и её частям
* Разработка документации на АС и её части
* Разработка и оформление документации на поставку комплектующих изделий
* Разработка заданий на проектирование в смежных частях проекта

6. Рабочая документация

* Разработка рабочей документации на АС и её части
* Разработка и адаптация программ

7. Ввод в действие

* Подготовка объекта автоматизации
* Подготовка персонала
* Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)
* Строительно-монтажные работы
* Пусконаладочные работы
* Проведение предварительных испытаний
* Проведение опытной эксплуатации
* Проведение [приёмочных испытаний](https://en.wikipedia.org/wiki/Acceptance_testing)

8. Сопровождение АС.

* Выполнение работ в соответствии с гарантийными обязательствами
* Послегарантийное обслуживание

Эскизный, технический проекты и рабочая документация — это последовательное построение все более точных проектных решений. Допускается исключать стадию «Эскизный проект» и отдельные этапы работ на всех стадиях, объединять стадии «Технический проект» и «Рабочая документация» в «Технорабочий проект», параллельно выполнять различные этапы и работы, включать дополнительные.

Данный стандарт не вполне подходит для проведения разработок в настоящее время: многие процессы отражены недостаточно, а некоторые положения устарели.

# Проектирование микропроцессорных систем управления полиграфическим оборудованием

# 1. Технология разработки и отладки микропроцессорной системы

Микропроцессорная система может быть описана на различных уровнях абстрактного представления.

Существующую микропроцессорную систему можно описать на любом известном уровне представления, но в начальной стадии проектирования ее можно описать только на концептуальном уровне. В процессе разработки системы происходит переход от одного уровня ее представления к другому, более детальному. Каждая абстракция несет в себе только информацию, которая соответствует данному уровню, и не содержит каких-либо сведений относительно более низких уровней. Микропроцессорная система может быть описана, например, на одном из следующих уровней абстрактного представления:

1) "черный ящик";

2) структурный;

3) программный;

4) логический;

5) схемный.

На уровне "черного ящика" микропроцессорная система описывается внешними спецификациями; перечисляются внешние характеристики.

Структурный уровень создается компонентами микропроцессорной системы: микропроцессорами, запоминающими устройствами, устройствами ввода/вывода, внешними запоминающими устройствами, каналами связи. Микропроцессорная система описывается функциями отдельных устройств и их взаимосвязью, информационными потоками.

Программный уровень разделяется на два подуровня: команд процессора и языковой. Микропроцессорная система интерпретируется как последовательность операторов или команд, вызывающих то или иное действие над некоторой структурой данных.

Логический уровень присущ исключительно дискретным системам. На этом уровне выделяются два подуровня: переключательных схем и регистровых пересылок. Подуровень переключательных схем образуется вентилями и построенными на их основе операторами обработки данных. Переключательные схемы подразделяются на комбинационные и последовательностные; первые в отличие от последних не содержат запоминающих элементов. Поведение системы на этом уровне описывается алгеброй логики, моделью конечного автомата, входными/выходными последовательностями 1 и 0. Комбинационные схемы представляются таблицей истинности, в которой каждому входному набору значений сигналов ставится в соответствие набор значений сигналов на выходах. Последовательностные схемы могут описываться диаграммами или таблицами входов/выходов, в которых определены взаимно однозначные соответствия между входами схемы, внутренними состояниями (комбинациями значений элементов памяти) и выходами. Подуровень регистровых пересылок характеризуется более высокой степенью абстрагирования и представляет собой описание регистров и передачу данных между ними. Он включает в себя две части: информационную и управляющую. Информационная часть образуется регистрами, операторами и путями передачи данных. Управляющая часть определяет зависящие от времени сигналы, инициирующие пересылку данных между регистрами.

Сxемный уровень образуется резисторами и конденсаторами. Показателями поведения системы на этом уровне служат напряжение и ток, представляемые в функции времени или частоты. Этот уровень описания дискретной системы широко используется в описаниях аналоговых систем и не является ни наинизшим из возможных, ни достаточным для полной характеристики системы.

Этапы проектирования микропроцессорных систем

Микропроцессорные системы по своей сложности, требованиям и функциям могут значительно отличаться надежностными параметрами, объемом программных средств, быть однопроцессорными и многопроцессорными, построенными на одном типе микропроцессорного набора или нескольких, и т.д. В связи с этим процесс проектирования может видоизменяться в зависимости от требований, предъявляемых к системам. Например, процесс проектирования МПС, отличающихся одна от другой содержанием ПЗУ, будет состоять из разработки программ и изготовления ПЗУ.

При проектировании многопроцессорных микропроцессорных систем, содержащих несколько типов микропроцессорных наборов, необходимо решать вопросы организации памяти, взаимодействия с процессорами, организации обмена между устройствами системы и внешней средой, согласования функционирования устройств, имеющих различную скорость работы, и т. д. Ниже приведена примерная последовательность этапов, типичных для создания микропроцессорной системы:

1. Формализация требований к системе.

2. Разработка структуры и архитектуры системы.

3. Разработка и изготовление аппаратных средств и программного обеспечения системы.

4. Комплексная отладка и приемосдаточные испытания.

Этап 1. На этом этапе составляются внешние спецификации, перечисляются функции системы, формализуется техническое задание (ТЗ) на систему, формально излагаются замыслы разработчика в официальной документации.

Этап 2. На данном этапе определяются функции отдельных устройств и программных средств, выбираются микропроцессорные наборы, на базе которых будет реализована система, определяются взаимодействие между аппаратными и программными средствами, временные характеристики отдельных устройств и программ.

Этап 3. После определения функций, реализуемых аппаратурой, и функций, реализуемых программами, схемотехники и программисты одновременно приступают к разработке и изготовлению соответственно опытного образца и программных средств. Разработка и изготовление аппаратуры состоят из разработки структурных и принципиальных схем, изготовления прототипа, автономной отладки.  
Разработка программ состоит из разработки алгоритмов; написания текста исходных программ; трансляции исходных программ в объектные программы; автономной отладки.

Этап 4. см. Комплексная отладка.

На каждом этапе проектирования МПС людьми могут быть внесены неисправности и приняты неверные проектные решения. Кроме того, в аппаратуре могут возникнуть дефекты.

Источники ошибок

Рассмотрим источники ошибок на первых трех этапах проектирования.

Этап 1. На этом этапе источниками ошибок могут быть: логическая несогласованность требований, упущения, неточности алгоритма.

Этап 2. На данном этапе источниками ошибок могут быть: упущения функций, несогласованность протокола взаимодействия аппаратуры и программ, неверный выбор микропроцессорных наборов, неточности алгоритмов, неверная интерпретация технических требований, упущение некоторых информационных потоков.

Этап 3. На этом этапе источниками ошибок могут быть: при разработке аппаратуры - упущения некоторых функций, неверная интерпретация технических требований, недоработка в схемах синхронизации, нарушение правил проектирования; при изготовлении прототипа - неисправности комплектующих изделий, неисправности монтажа и сборки; при разработке программных средств - упущения некоторых функций технического задания, неточности в алгоритмах, неточности кодирования.

Субъективные неисправности отличаются от физических тем, что после обнаружения, локализации и коррекции больше не возникают. Однако, как следует из перечня источников ошибок, субъективные неисправности могут быть внесены на этапе разработки спецификации системы, а это означает, что даже после самых тщательных испытаний системы на соответствие ее внешним спецификациям в системе могут находиться субъективные неисправности.

Процесс проектирования - итерационный процесс. Неисправности, обнаруженные на этапе приемосдаточных испытаний, могут привести к коррекции спецификаций, а следовательно, к началу проектирования всей системы. Обнаруживать неисправности необходимо как можно раньше, для этого надо контролировать корректность проекта на каждом этапе разработки.

Основные методы контроля правильности проектирования следующие: верификация - формальные методы доказательства корректности проекта; моделирование; тестирование.

Для контроля корректности проекта на каждом этапе проектирования необходимо проводить моделирование на различных уровнях абстрактного представления системы и проверку правильности реализации заданной модели путем тестирования. На этапе формализации требований контроль корректности особо необходим, поскольку многие цели проектирования не формализуются или не могут быть формализованы в принципе. В идеальном случае разрабатываются тесты, целиком основанные на этой спецификации и дающие возможность проверки любой реализации системы, которая объявляется способной выполнять функции, оговоренные в спецификации. Этот способ - полная противоположность другим, где тесты строятся применительно к конкретным реализациям. Независимая от реализации функциональная проверка обычно заманчива лишь в теоретическом плане, но практического значения не имеет из-за высокой степени общности.

После обнаружения ошибки должен быть локализован ее источник, чтобы провести коррекцию на соответствующем уровне абстрактного представления системы и в соответствующем месте. Ложное определение источника ошибки или проведение коррекций на другом уровне абстрактного представления системы приводит к тому, что информация о системе на верхних уровнях становится ошибочной и не может быть использована для дальнейшей отладки при производстве и эксплуатации системы.

Отладка микропроцессорных систем. О правильности функционирования микропроцессорной системы на уровне "черного ящика" с полностью неизвестной внутренней структурой можно говорить лишь тогда, когда произведены ее испытания, в ходе которых реализованы все возможные комбинации входных воздействий, и в каждом случае проверена корректность ответных реакций. Однако исчерпывающее тестирование имеет практический смысл лишь для простейших элементов систем. Следствием этого является тот факт, что ошибки проектирования встречаются при эксплуатации, и для достаточно сложных систем нельзя утверждать об их отсутствии на любой стадии жизни системы. В основе почти всех методов испытаний лежит та или иная гипотетическая модель неисправностей, первоисточником которой служат неисправности, встречающиеся в практике. В соответствии с моделью в рамках каждого метода предпринимаются попытки создания тестовых наборов, которые могли бы обеспечить удовлетворительное выявление моделируемых неисправностей. Любой метод тестирования хорош ровно настолько, насколько правильна лежащая в его основе модель неисправности.

Диагностика неисправности - процесс определения причины появления ошибки по результатам тестирования. Отладка - процесс обнаружения ошибок и определение источников их появления по результатам тестирования при проектировании микропроцессорных систем. Средствами отладки являются приборы, комплексы и программы.

Точность, с которой тот или иной тест локализует неисправности, называется его разрешающей способностью. Требуемая разрешающая способность определяется конкретными целями испытаний.

Так как процесс проектирования микропроцессорной системы содержит неформализуемые этапы, то отладка системы предполагает участие человека.

Свойство контролепригодности системы.

Успех отладки зависит от того, как спроектирована система, предусмотрены ли свойства, делающие ее удобной для отладки, а также от средств, используемых при отладке. Для проведения отладки проектируемая микропроцессорная система должна обладать свойствами управляемости, наблюдаемости, предсказуемости.

Управляемость - свойство системы, при котором ее поведение поддается управлению, т. е. имеется возможность остановить функционирование системы в определенном состоянии, и затем сновва ее запустить. Наблюдаемость - свойство системы, позволяющее проследить за поведением системы, сменой ее внутренних состояний. Предсказуемость - свойство системы, позволяющее установить систему в состояние, из которого все последующие состояния могут быть предсказаны.

Средства отладки должны:

1) управлять поведением системы или/и ее модели на различных уровнях абстрактного представления;

2) собирать информацию о поведении системы или/и ее модели, обрабатывать и представлять на различных уровнях абстракции;

3) преобразовывать системы, придавать им свойства контролепригодности;

4) моделировать поведение внешней среды проектируемой системы.

Под управлением поведением системы или ее модели понимаются определение и подача входных воздействий для запуска или останова системы или ее модели, для перевода в конкретное состояние последних. Чтобы определить место субъективной неисправности, которая может быть внесена на любой стадии проектирования, необходимо уметь собирать информацию о поведении системы и представлять ее в тех формах, которые приняты для данного проекта. Например, это могут быть временные диаграммы, принципиальные электрические схемы, язык регистровых передач, ассемблер и др.

Процесс отладки прототипа проектируемой системы должен начинаться с отладки аппаратуры и отладки программ.

Отладка аппаратуры предполагает тестирование отдельных устройств микропроцессорной системы - процессора, ОЗУ, контроллеров, блока питания, генератора тактовых импульсов путем подачи тестовых входных воздействий и приема ответных реакций. Тестовые входные воздействия и ответные реакции определяются, исходя из спецификаций на устройства, а также структурных схем устройств. При этом проверяются реальная аппаратура прототипа, спецификации, структурные схемы и отлаживаются тесты. После отладки отдельных устройств проверяется их взаимодействие. Процессор системы работает с шинами адресов, данных и управления. Анализируя их сигналы, можно проконтролировать выполнение программы в процессоре.

Поскольку ША и ШД синхронные, их работу лучше всего проверить с помощью методов логических состояний. Перед анализом последовательностей данных на этих шинах необходимо удостовериться в том, что сигналы, управляющие взаимодействием процессора с другими устройствами, выдаются в соответствующем порядке. Поскольку ШУ состоит из линий, работающих асинхронно, необходимо просматривать сигналы многих линий в течение одного и того же промежутка времени. Для анализа асинхронной работы линий управления необходимо также наблюдать за сигналами на них при возникновении определенного события, чтобы можно было четко разделить и идентифицировать различные состояния линий. Например, среди сигналов ШУ могут быть сигналы длительностью всего несколько наносекунд, но могут также возникать кратковременные ложные узкие импульсы, вызванные перекрестными помехами или шумами.

После того как доказана работоспособность ШУ, проводится дальнейшая проверка работы аппаратуры при различных режимах адресации процессора и кодах выбираемых данных. Для проверки выполнения процессором инструкций разрабатывается тестовая программа, которая помещается в ОЗУ или ППЗУ. При этом проверяется временная диаграмма сигналов и прохождения данных в системе (как осуществляется передача информации по отношению к строб-сигналам). Если тестовая программа - системный проверяющий тест пройдет успешно, можно утверждать, что автономно аппаратура отлажена.

При автономной отладке аппаратуры могут потребоваться приборы, умеющие: а) выполнять функции аналогового прибора, т. е. измерять напряжение и ток; воспроизводить форму сигнала, подавать импульсы определенной формы и т. д.; б) подавать последовательность сигналов одновременно на несколько входов в соответствии с заданной временной диаграммой или заданным алгоритмом функционирования аппаратуры, представленным в спецификации на языке высокого уровня, или другим способом; собирать значения сигналов многих линий в течение одного и того же промежутка времени, который определяется задаваемыми, программируемыми событиями - комбинацией или последовательностью сигналов на линиях, например, ложным сигналом на линии; обрабатывать и представлять собранную информацию либо в виде временной диаграммы, либо в виде диаграммы или таблицы логических состояний, либо на языке высокого уровня, например, языке регистровых передач.

Для автономной отладки аппаратуры широко используются осциллографы, вольтметры, амперметры, частотомеры, генераторы импульсов, позволяющие отлаживать аппаратуру на схемном уровне. Чтобы автономно отладить аппаратуру МПС на более высоком уровне, применяют логические анализаторы, генераторы слов, пульты, комплексы диагностирования.

Отладка программ микропроцессорной системы проводится, как правило,

Проверка корректности программ, т.е. проверка соответствия их внешним спецификациям, осуществляется тестированием. Программы проверяются на функционирование с различными исходными данными. Результаты функционирования программ сравниваются с эталонными значениями.

Отладка программ подразделяется на следующие этапы: планирование отладки; составление тестов и задания на отладку; исполнение программ; информирование о результатах исполнения программ по заданным исходным данным; анализ результатов, обнаружение ошибок и локализация неисправностей.

Существует два способа начального тестирования программ: пошаговый режим и трассировка программ.

Исполнение программ осуществляется по шагам последовательно во времени и в соответствии с заданиями, содержащимися в операторах. При этом производится переработка значений переменных и определение оператора приемника. Если в ходе исполнения программы регистрируется последовательность операторов, реализуемых на каждом шаге процесса, то получается трасса или маршрут исполнения программы, который для конкретной программы зависит только от значений исходных данных.

Отдельные участки программы после проверки, используя пошаговый режим или трассировку, можно объединить и проверить с помощью установки контрольных точек, вводимых в программу и прерывающих ее исполнение, для передачи управления программе-отладчику. По контрольным точкам можно по желанию выполнить избранные участки программы и проанализировать результаты. Контрольные точки устанавливаются обычно для конкретной команды, но в некоторых системах предусматриваются прерывания программы при чтении или записи данных в определенные ячейки памяти. Возможны и более сложные условия прерывания программы.

Расстановка контрольных точек предполагает, что программист связывает с ней точный адрес памяти. Для некоторых отладчиков программист задает абсолютный шестнадцатеричный адрес. Последние отладчики допускают символьные значения адресов, которые программист определяет в исходной программе; это позволяет значительно экономить время, распечатывая после каждого редактирования и транслирования программы новую копию листинга.

При тестировании можно планировать проверку всех возможных маршрутов исполнения программы для разных исходных переменных. Однако это реализуемо только для очень простых программ небольшого объема при малых диапазонах изменения исходных данных. Поэтому при планировании отладки программ применяют критерии полноты тестирования, которые, однако, не гарантируют полной проверки программ. Выбор критерия зависит от наличия ресурсов для тестирования и структурной сложности отлаживаемой программы. Критерии характеризуются глубиной контроля программ и объемом проверок.

Средства отладки программ должны:  
а) управлять исполнением программ (останавливать, изменять порядок, запускать и т. д.);

б) собирать информацию о ходе выполнения программы;

в) обеспечивать обмен информацией (диалог) между программистом и ЭВМ на уровне языка программирования;

г) моделировать работу отсутствующих аппаратных средств микропроцессорной системы.

# Тема: Приводы

# Двигатель

Общий обзор

Двигатели преобразуют электрическую энергию в кинетическую или вращательную энергию. Они являются важной частью системы привода.

Изображение выглядит как текст, доска

Автоматически созданное описание

# Типы двигателей

С начала исследований электромеханических энергетических установок в начале 19-ого столетия, были разработаны 3 различных типа двигателей, которые отличаются по структуре и функциональным возможностям:

- двигатель постоянного тока (сокращение ДПТ, DCM)

- Синхронный двигатель (сокращение СД, SM)

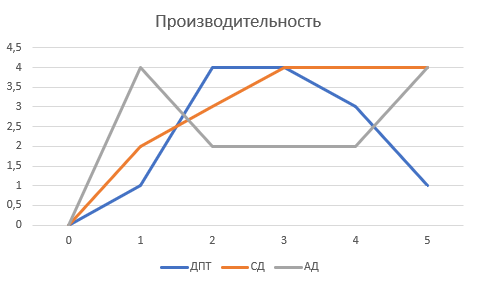
- Асинхронный двигатель (сокращение АД, ASM)

Изображение выглядит как текст, счетчик, устройство, датчик

Автоматически созданное описание

Имеются много разновидностей этих базовых типов, например: шаговые двигатели, реактивные синхронные электродвигатели, линейные двигатели.

# Сравнение типов двигателей



1 – прочность

2 – сложность управляющей электроники

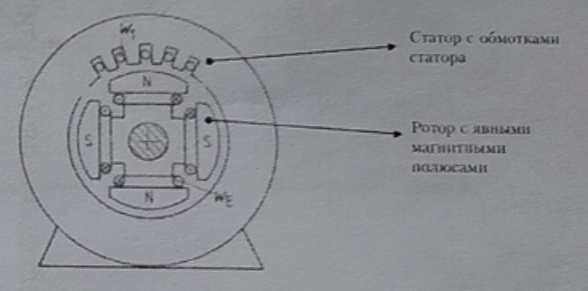
3 – динамика

4 – удельная мощность (отношение мощность/размер)

5 – эксплуатационные требования

# Конструкция синхронного двигателя

Основные элементы этого двигателя с вращающимся полем, в форме двигателя с внутренним полюсом (статор + ротор), показаны на следующей схеме:



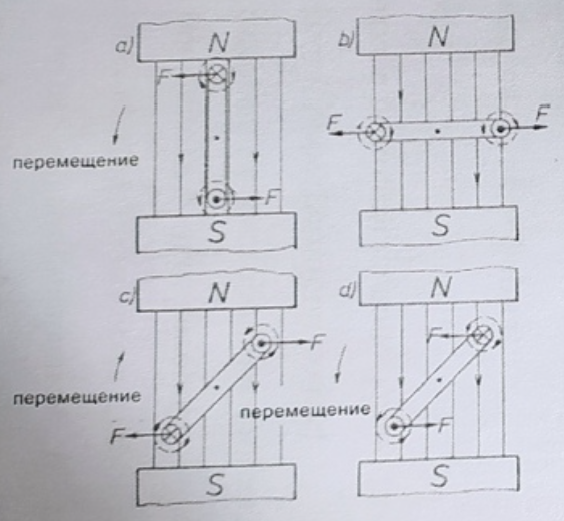
Обмотки статора соединены трехфазной звездой (L1, L2, L3).

При подаче трехфазного электропитания обмотка статора вырабатывает вращающее поле.

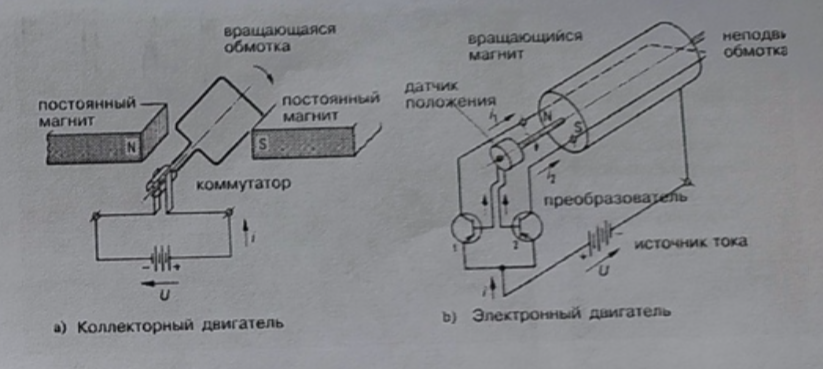
Для двигателей средней производительности в роторе используются, главным образом, высококачественные постоянные магниты. Они производят поле ротора, соответствующее его позиции.

Используя следующую формулу, можно показать, что сила, действующая на проводник на расстоянии r от геометрического центра (вала двигателя), создает вращающий момент M между вектором силы и ротором, зависящий от угла альфа:

Для получения максимального момента угол должен равняться 90о, иллюстрируется на следующей схеме:



От механической коммутации к электрической

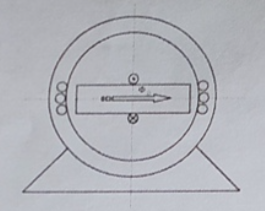


а) Коллекторный двигатель – двигатель постоянного тока (ДПТ)

b) Электронный двигатель – синхронный двигатель (СД)

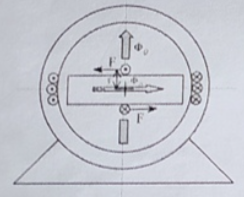
# Вращение СД

Чтобы показать последовательность вращения, предположим, что поле возбудителя создается одной катушкой ротора, через которую протекает постоянный ток. При этом поле возбудителя постоянно, а его направление зависит от позиции ротора.



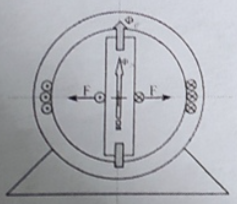
Если ток протекает через некоторые из обмоток статора, то создается новое поле статора.

Оно проходит через ротор, и, согласно правилу левой руки, на проводник с током действует сила. Ротор перемещается возникающим вращающим моментом.



Это перемещение заканчивается, когда магнитное поле ротора направлено так же, как магнитное поле статора

В этом случае результирующий вращающий момент = 0, потому что силы взаимно компенсируются.



Если к ротору приложена нагрузка, то он отклоняется из установленного положения, пока двигатель не создаст достаточный вращающий момент. Вращающий момент двигателя зависит от угла между фиQ и фиD.

Однако, мы хотим устранить отклонения между установленной и реальной позициями. Для этого можно запитать обмотки статора так, чтобы поле фиQ было перпендикулярно к полю фиD, используя ток IQ, создающий вращающий момент двигателя, который работает против нагрузки.



Эта процедура (создание угла 90o между фиQ и фиD), позволяющая получить максимальный вращающий момент, называется управлением по ориентации поля или векторным управлением.

# 1. АЦП и ЦАП

# ЦАП

ЦАП служат для преобразования информации из цифровой формы в аналоговый сигнал. ЦАП широко применяется в различных устройствах автоматики для связи контроллеров, вырабатывающих сигналы управления в виде цифрового кода, с аналоговыми элементами системы.

Принцип работы ЦАП состоит в суммировании аналоговых сигналов, пропорциональных весам разрядов входного цифрового кода, с коэффициентами, равными нулю или единице в зависимости от значения соответствующего разряда кода.

ЦАП преобразует цифровой двоичный код а0, а1, а2, .. ап-1 в аналоговую величину, обычно напряжение Uвых.. Каждый разряд двоичного кода имеет определенный вес i-го разряда вдвое больше, чем вес (i-1)-го. Работу ЦАП можно описать следующей формулой:

Изображение выглядит как антенна

Автоматически созданное описание -

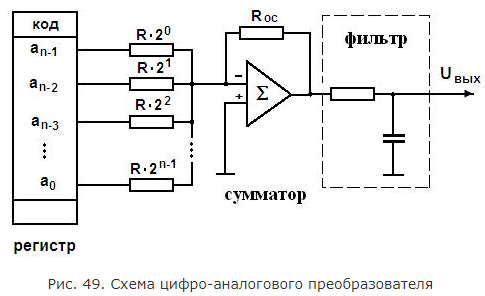
 напряжение, соответствующее весу младшего разряда, аi- значение i -го разряда двоичного кода (0 или 1).

Точность преобразования:

Изображение выглядит как текст

Автоматически созданное описание

Выбором е можно установить требуемый масштаб аналоговой величины. На рисунке 49 приведена схема цифро-аналогового преобразователя.



В регистр записывается двоичный код выходного сигнала, на выходе сумматора формируется аналоговый эквивалентный сигнал. Этот сигнал имеет ступенчатую форму (рис. 50) и для его сглаживания нужен фильтр низкой частоты.



# АЦП

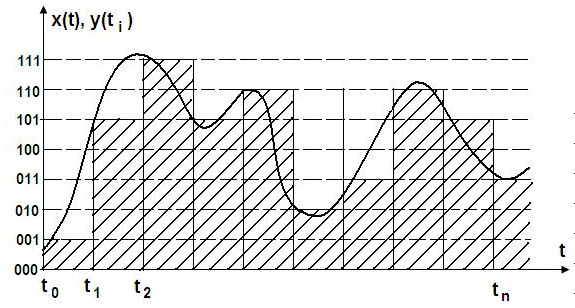
В настоящее время в большинстве случаев датчики являются элементами цифровых систем управления, что требует преобразования сигнала датчиков в цифровую форму, при вводе в контроллер и иногда обратного преобразования цифрового управляющего сигнала контроллера в аналоговую форму при выводе на исполнительное устройство. Аналого-цифровое преобразование (АЦП, ADC) содержит 3 фазы: дискретизацию по времени, квантование по уровню, цифровое, двоичное кодирование.

**а)** При дискретизации по времени из непрерывного сигнала x(t) формируется последовательность отсчетов **y(Ti).** В случае равномерной дискретизации . Возможность восстановления исходного сигнала по отсчетам определена теоремой Котельникова, по которой между частотой квантования и максимальной частотой спектра сигнала, которую надо учитывать, должно выполняться соотношение.

Изображение выглядит как текст

Автоматически созданное описание

Для упрощения построения АЦП обычно выбирают . Рис. 48.



**b)** Квантование по уровню состоит в округлении значения отсчета до ближайшего уровня квантования. Весь диапазон значений измеряемой величины от  до  разбивается на равные интервалы. Действительные значения воспроизводятся с помощью дискретных, отличающихся на . Процесс перехода от непрерывных значений в дискретные называется квантованием.  - шаг квантования. Шаг выбирается в пределах допускаемой погрешности измерения. Времени  соответствует значение , и т.д. Величина  - единицы измерения. Это равномерная шкала квантования, реже используются другие, например, логарифмические или специальные: A, l.. При квантовании возникает задержка нулевого уровня, которую надо учитывать при оценке динамических характеристик системы: .

**c)** Отсчеты кодируются в двоичном или двоично-десятичном коде. Для цифрового кодирования необходимо в каждый дискретный момент времени воспроизвести в цифровой форме значения, заменившее непрерывную измеряемую величину. Для этого дискретные значения  представляют в виде последовательности цифровых кодов. Например, Х0 – 001, Х1 – 101, Х2 – 111 и т. д.

# 2. Датчики системы управления перемещения, углового перемещения, угловых поворотов, угловой скорости, световых сигналов

Датчик линейных перемещений — это устройство, предназначенное для определения изменения местоположения объекта по одной координате, а также расстояния до объекта. При этом объект может находиться в твердой, жидкой или сыпучей форме.

Такие датчики преобразуют данные о перемещении объекта в выходной сигнал. Являются одним из важных измерительных элементов систем управления и контроля. Они широко применяются в различных областях, поэтому выделяют несколько разновидностей, отличающихся по принципу действия, точности, цене.

Основные функции:

показывают положение объекта управления (ОУ) или рабочего органа оборудования;

отслеживают линейные перемещения ОУ или рабочего органа;

фиксируют окончание этапа в системах цикловой автоматики;

определяют размеры ОУ (например, заготовок);

измеряют уровень жидкости;

характеризуют состояние оборудования в части его загрузки.

Датчик угла поворота (сокр. ДУП), также энкодер (от англ. encoder — кодирующее устройство) — измерительный преобразователь, предназначенный для преобразования угла поворота вращающегося объекта (например, вала) в цифровые или аналоговые сигналы, позволяющие определить угол его поворота.

Датчики угла поворота имеют множество применений. Они широко применяются в промышленности (в частности в сервоприводах), в робототехнике, в автомобилестроении (например, для определения угла поворота рулевого колеса), в компьютерной технике (для определения угла поворота колеса компьютерной мыши) и т. п.

Да́тчик углово́й ско́рости (ДУС) — устройство, первичный прибор (датчик) для измерения угловой скорости поворота корпуса летательных аппаратов относительно невращающейся инерциальной системы координат. Используется в системах управления различных летательных аппаратов: ракет, самолётов, вертолётов и др. Выходной сигнал устройства обычно электрический, пропорциональный угловой скорости и используется в пилотажных системах летательных аппаратов, в частности, автопилоте, системах стабилизации траектории полёта ракет. Обработка сигнала производится во внешней системе.

Для измерения угловых скоростей по трём перпендикулярным координатным осям почти всегда применяют три по-разному ориентированных датчика — датчики угловой скорости крена, тангажа и рысканья.

# 3. Программное обеспечение для идентификации объектов

Среди множества задач, связанных с первичным сбором информации, можно выделить автоматическую идентификацию (распознавание и различение) разнородных предметов (товары, инвентарь, багаж, документы и т. п.). Ее решение предполагает выполнение следующих шагов: присвоение каждому предмету определенного идентификатора (номера или кода), нанесение на предмет специализированной метки (кодирование), содержащей идентификатор, считывание данных с метки цифровым устройством и перевод данных метки в электронный вид.

В качестве идентификатора можно использовать графические, магнитные, радиочастотные и электронные метки. Все они предназначены для автоматической идентификации и находят применение в различных сферах. Наибольшей популярностью пользуется графическая метка в виде штрих-кода.

Из доступных технологий автоматической идентификации наибольшую популярность приобрела технология штрихового кодирования. Прежде всего это связано с простотой данной технологии и низкой стоимостью расходных материалов: нанесение штрих-кода на ярлык или упаковку обходится значительно дешевле нанесения магнитных, радиочастотных и электронных меток. Если штрих-код наносится типографским способом, то на стоимости упаковки это не отражается, если же штрих-код печатается на самоклеящейся этикетке, то стоимость упаковки возрастает незначительно. Таким образом, маркировка упаковки благодаря методу штрихового кодирования является простой и доступной.

Штрих-код (или от англ, *bar code -* баркод) - это графическая метка, в которой по определенным правилам закодирована информация. Штриховой код - это информация, закодированная в виде, удобном для считывания техническими средствами, как правило, это алфавитно-цифровой код-идентификатор.

Помимо изображения штрихового кода на упаковке или этикетке может присутствовать поле с алфавитно-цифровым эквивалентом штрих-кода и дополнительная текстовая и графическая информация, предназначенная для прочтения человеком.



Для создания изображения штрихового кода необходимо специальное программное обеспечение, которое поставляется вместе с принтерами штрих-кода. Примером такого программного пакета является «АТОЛ: ВагСобеМакег». Приобрести его можно как во время покупки оборудования, так и отдельно. При установке «АТОЛ: ВагСобеМакег» на ПК вместе с программным обеспечением устанавливаются специальные шрифты, которые могут использоваться для создания штрих-кода в любых приложениях.

Созданное изображение штрих-кода можно напечатать типографским способом на специализированном принтере для этикеток или на обычном офисном принтере.

Распознавание штрих-кода осуществляется специальным устройством - сканером штрих-кода. Сканер распознает и считывает штрих-код, переводит данные в электронный вид и передает для дальнейшей обработки в кассовый аппарат или ПК по тому или иному интерфейсу.

По принципу действия сканирующие устройства делятся на контактные и дистанционные. Рабочий инструмент контактного устройства представляет собой световое перо, перемещаемое по поверхности с нанесенным штрих-кодом. Такие сканеры весьма популярны вследствие их низкой стоимости. Они очень просты в использовании, однако требуют от оператора сохранения постоянной скорости движения через поле штрих-кода и плоской поверхности позади него, для обеспечения постоянного давления, прилагаемого оператором к сканеру во время считывания.

В дистанционных сканерах используется считывание лазерного или инфракрасного луча, отраженного от поверхности с нанесенным штриховым кодом.

Плохо отпечатанные штрих-коды, штрих коды разных размеров с неровных поверхностей лазерные сканеры обычно считывают с расстояния примерно в 25-30 см.

Стационарные сканеры могут производить считывание штрихового кода с движущихся объектов. Причем перемещаться может объект сканирования относительно сканера либо сам сканер может быть расположен на платформе движущегося оборудования.