

Компьютерные технологии в области автоматизации и управления

Раздел 1.

Роль компьютерных технологий в автоматизации и управлении



Что такое ПТК?

Средства автоматизации выпускаются в виде наборов программно-технических комплексов (ПТК). Основными признаками таких комплексов для построения АСУ ТП являются совместимость их отдельных компонентов, способность функционировать в единой системе, а также стандартизация интерфейсов, функциональная полнота, позволяющая строить целиком АСУ ТП из средств только данного комплекса.

Функциональные элементы, присущих большинству ПТК:

- программируемые логические контроллеры или контроллеры на базе РС,
- интеллектуальные устройства связи с объектом;
- устройства связи с оператором (дисплеи, панели операторов и т.д.)
- промышленные сети;
- рабочие станции и серверы различного назначения;
- прикладное программное обеспечение.



Структура ПТК

Структура ПТК определяется средствами и характеристиками взаимосвязи отдельных компонентов комплекса (контроллеров, пультов оператора, удаленных блоков ввода/вывода), т.е. его сетевыми возможностями.

Гибкость и разнообразие возможных структур ПТК зависит от

- числа имеющихся сетевых уровней,
- возможных типов связи на каждом уровне сети (шина, звезда, кольцо),
- параметров сети каждого уровня:
 - возможных типов кабеля,
 - максимально возможных расстояний,
 - максимального числа узлов (компонентов комплекса), подключаемых к каждой сети,
 - скорости передачи информации при разных типах кабеля,
 - методе доступа компонентов к сети (случайный по времени доставки сообщений или гарантирующий время их доставки).


Указанные свойства ПТК характеризуют: возможность распределения аппаратуры в производственных цехах; объем производства, который может быть охвачен системой автоматизации, реализованной на данном ПТК; предельную динамику передачи оперативной информации через любую из имеющихся сетей, возможность переноса блоков ввода/вывода непосредственно к датчикам и исполнительным механизмам, что позволяет существенно сэкономить затраты на кабель и уменьшить помехи из-за передачи низковольтных аналоговых сигналов на большие расстояния. Для компенсации аварийных ситуаций, требующих согласованной во времени работы ряда контроллеров, важно обеспечение требуемого времени передачи приоритетных сигналов по сети. Наличие информационной сети для передачи больших массивов информации между пультами операторов и между ними и сервером корпоративной сети предприятия, а также характеристики этой сети (включая ее протоколы) позволяют судить о возможностях связей рассматриваемой системы автоматизации с другими более высокими уровнями управления производства.



Стандартизация, типизация и открытость ПТК

Наиболее распространенные типовые решения:


- *магистрально-модульная архитектура связей плат контроллеров*, которая позволяет использовать в одном приборе платы разных фирм, собранные по одному стандарту;
- *конструктивное оформление* контроллеров и выносных блоков ввода/вывода, корпусов, стоек и шкафов;
- *операционные системы реального времени для контроллеров*, имеющие малое время реакции на внешние сигналы, создающие открытую среду для разработчиков прикладных программ и облегчающие типизацию и перенос прикладных программ на контроллеры разных фирм;
- *промышленные сети*, используемые в качестве полевых и системных сетей, имеющие гарантированное время передачи сигналов по сети и позволяющие связывать контроллеры и приборы разных фирм, имеющие интерфейс к этим сетям;
- *операционные системы для пультов оператора типа Windows*, дающие возможности использовать обширное поле прикладных программ, работающих под данными операционными системами;
- *информационная сеть Ethernet*, имеющая наибольшее распространение в корпоративных сетях предприятий и позволяющая непосредственно обмениваться данными с производственными отделами предприятия;
- *пакеты визуализации технологической информации на дисплейных пультах операторов*, имеющие драйверы к контроллерам разных производителей и стандартные межпрограммные интерфейсы для использования приложений создаваемых различным программным обеспечением.



Программное обеспечение ПТК

Неотъемлемой частью ПТК является программное обеспечение, которое подразделяется на следующие части:

- *системное программное обеспечение контроллеров;*
- *системы подготовки программ для контроллеров;*
- *программное обеспечение для визуализации информации на пультах операторов;*
- *сервисные программы параметризации отдельных модулей;*
- *прилагаемые к ПТК программы САПР и прикладные пакеты.*



ПТК на базе ПК или PC-совместимого контроллера (PC based Control)

Основные сферы использования контроллеров на базе ПК – специализированные системы автоматизации в медицине, научных лабораториях, средствах коммуникации, для небольших замкнутых объектов в промышленности. Общее число входов/выходов такого контроллера обычно не превосходит нескольких десятков, а функции выполняют достаточно сложную обработку измерительной информации с расчетом управляющих воздействий.

Рациональную область применения контроллеров на базе ПК можно очертить следующими условиями:


- при нескольких входах и выходах объекта надо производить большой объем вычислений за достаточно малый интервал времени (необходима большая вычислительная мощность);
- средства автоматизации работают в окружающей среде, не слишком отличающейся от условий работы обычных ПК;
- реализуемые контроллером функции целесообразно (в силу их нестандартности) программировать не на одном из специальных технологических языков, а на обычном языке программирования высокого уровня типа C++, PASCAL;
- мощная поддержка работы операторов.



ПТК на базе локальных ПЛК (Local PLC)

Единичный локальный контроллер выполняет все необходимые функции системы автоматизации на достаточно изолированном небольшом производственном узле, при этом не требуется его связь с другими средствами автоматизации. Он либо является автономным конструктивом, подсоединяемым к автоматизируемому объекту, либо встраивается в оборудование и является его неотъемлемой частью.

Контроллеры обычно рассчитаны на десятки входов/выходов от датчиков и исполнительных механизмов; их вычислительная мощность невелика; они реализуют простейшие типовые функции обработки измерительной информации, логического управления, регулирования.




ПТК для создания сетевого комплекса контроллеров (PLC NetWork)

Этот класс микропроцессорных средств наиболее широко используется во всех отраслях промышленности.

Обычный состав сетевого комплекса контроллеров:

- несколько одно- или разнотипных контроллеров, обычно, одной серии;
- одна или несколько дисплейных рабочих станций операторов;
- промышленная сеть, соединяющая контроллеры и рабочие станции между собой;
- полевая сеть, позволяющая выносить блоки ввода/вывода контроллера к отдельным датчикам и исполнительным механизмам.



ПТК для создания распределенных систем управления (distributed control systems – DCS)

Такие ПТК имеют несколько уровней промышленных сетей, соединяющих контроллеры между собой и с рабочими станциями операторов (например, нижний уровень, используемый для связи контроллеров и рабочей станции отдельного компактно расположенного технологического узла, и высший уровень, реализующий связи средств управления отдельных узлов друг с другом и с рабочими станциями операторов всего автоматизируемого участка производства).

Свойства, присущие данному классу контроллерных средств:

- наличие мощных по вычислительным возможностям модификаций контроллеров, что позволяет реализовать в них многие современные высокоэффективные, но сложные и объемные алгоритмы контроля, диагностики, управления;
- использование контроллеров различных серий;
- использование протяженных промышленных сетей, позволяющих подсоединять к одной шине сотни узлов (контроллеров и рабочих станций) и распределять эти узлы на значительные расстояния (десятки километров);
- работа взаимодействующих рабочих станций в клиент/серверном режиме и в структуре Интранет;
- достаточно проработанное включение в систему информационных сетей для связи рабочих станций операторов друг с другом, для их связи с серверами баз данных, для взаимодействия данной системы с корпоративной сетью предприятия, для возможности построения необходимой иерархии управляющих центров планирования, диспетчеризации и оперативного управления, для вывода нужной информации за пределы данного предприятия с помощью глобальной сети Internet;
- наличие в составе системы ряда прикладных пакетов программ, реализующих функции эффективного управления отдельными агрегатами (многосвязное регулирование, оптимизация и т. д.), функции диспетчерского управления участками производства (компьютерная поддержка принятия управленческих решений), функции технического и экономического учета и оперативного планирования производства в целом.

Уровни управления



Основной тенденцией развития современных автоматизированных систем управления предприятием является интеграция исторически уже сложившихся уровней, создание единого информационного пространства. При этом выделяют вертикальную и горизонтальную интеграцию.

Так как основу автоматизированных средств составляют элементы цифровой вычислительной техники то взаимодействие различных уровней автоматизации предполагает взаимодействие между собой различных уровней программного обеспечения

Программное обеспечение различных уровней автоматизации



- Вершину иерархии задач управления составляют задачи относящиеся к классу ERP (Enterprise Resource Planning – планирование ресурсов предприятия) или MRP (Manufacturing Resource Planning – планирование ресурсов производства). Системы ERP ориентированы на предприятие в целом, а MRP на его технологические подразделения.
- Группа задач уровня MES (Manufacturing Execution Systems). Это граничные задачи уровней АСУП и АСУТП, не относящиеся ни к одному из этих уровней в чистом виде.
- Следующий уровень задач в иерархии управления производством относят к системам типа SCADA (Supervisory Control and Data Acquisition) и DCS (Distributed Control Systems). Они решают задачи управления процессом с операторных станций и поддержки распределенных систем управления. Кроме прочего системы SCADA и DCS включают в себя задачи класса MMI (Man-Machine Interface) или HMI (Human-Machine Interface) – обеспечения человеко-машинного интерфейса, двусторонней связи "оператор – технологическое оборудование".
- Нижний уровень образуют задачи, решаемые на уровне программируемых логических контроллеров. Они четко относятся к сфере АСУТП.

Современные технологии САПР для предприятий представлены системами CAD/CAM/CAE/PDM (Computer Aided Design, Manufacturing, Engineering, Product Data Management). САПР могут использоваться на всех перечисленных выше уровнях управления предприятием.

Для осуществления перечисленных задач с помощью средств вычислительной техники должно существовать соответствующее программное обеспечение, а также средства для создания такого программного обеспечения.

Место и назначение элементов уровней АСУ ТП





Условия интеграции подсистем АСУ

Для того чтобы различным уровням АСУ обмениваться данными, необходимо установить общий язык и иметь общее информационное пространство для общения. Для обеспечения этого необходимо:

- Использование баз данных на каждом уровне не только в качестве средства для хранения функциональных данных, но и как информационных буферов между различными подсистемами.
- Применение класса продуктов, главным назначением которых является импортирование объектов из одной подсистемы и экспортирование их в другую подсистему.



Базы данных реального времени

Базы данных становятся основным средством интеграции подсистем АСУ. Однако этому препятствует несколько основных ограничений:

- производственные процессы генерируют данные очень быстро и обычные БД не всегда могут выдержать подобную нагрузку;
- объёмы производственной информации настолько велики, что она просто может не вписаться в традиционную БД.

Результатом преодоления этих ограничений стало появление класса продуктов, называемых базами данных реального времени (БДРВ). В настоящее время БДРВ являются продуктами, ориентированными на хранение технологической информации, на высокие скоростные характеристики регистрации, на сжатие данных, на обеспечение связи с управленческими данными, на использование уже ставших стандартными в подсистемах АСУП интерфейсов.

Используемая в БДРВ архитектура клиент-сервер позволяет заполнить промежуток между промышленными системами контроля и управления реального времени, для которых характерны большие объёмы информации, и открытыми гибкими управленческими информационными системами. Благодаря наличию мощного и гибкого процессора запросов пользователи имеют возможность осуществлять поиск любой степени сложности для выявления зависимостей и связей между физическими характеристиками, оперативными условиями и технологическими событиями.

Стандартным механизмом поиска информации на серверах БДРВ является SQL, что гарантирует доступность данных самому широкому кругу приложений. В подмножество языка SQL входят расширения, служащие для получения динамических производственных данных и позволяющие строить запросы на базе временных отметок.

Примером Программных пакетов создания СУБД РВ является IndustrialSQL Server, предлагаемый фирмой Wonderware. Он позволяет регистрировать данные в реальном времени. Построен IndustrialSQL Server на базе Microsoft SQL Server. Это существенно расширяет возможности всего производственного персонала в смысле возможности доступа к полной информации о любом этапе производства.



Развитие связующих технологий

Применение продуктов, назначением которых является импортирование объектов из одной подсистемы и экспортирование их в другую подсистему невозможно без широкой стандартизации интерфейсов. Очевидно, что верхние уровни управления «привыкли» работать с офисными приложениями, поэтому работая с технологической информацией необходимо уметь передавать и принимать данные из офисных приложений. Рынок развития программного обеспечения определил ведущую роль в этой области решений предлагаемых компанией Microsoft.

Первоначально Microsoft для интеграции между различными офисными приложениями в Windows (для использования объектов одного приложения, например, таблицы Excel, в другом приложении, например, в редакторе Word) ввела технологию OLE (Object Linking and Embedding – Связывание и Встраивание Объектов). Развитие этой технологии привело к созданию Модели Составных Объектов (Component Object Model – COM) и её сетевого расширения DCOM (Distributed COM – Распределённая COM). Начиная с версии OLE 2.0, в её основу была положена модель COM.

Постепенно COM пронизала все варианты Windows 9.x/NT/CE. Достаточно упомянуть такие её производные, как ActiveX (OLE-Автоматизация) или OLE DB (OLE for Data Base OLE для Баз Данных), не говоря уже о самой офисной OLE. В Windows2000 она преобразовывается в COM+ и объявляется основной склеивающей технологией программирования в архитектуре DNA (Distributed interNet Application – Распределённые Приложения Internet), а связанные с этим технологии объединяются под общим названием Component Services (Сервисы Компонентов).

Модель COM оперирует объектами, очень похожими на объекты в объектно-ориентированных языках программирования типа C++. Но сама технология COM не является языком программирования. Она только регламентирует поведение своих объектов.

Объекты COM предоставляют свою функциональность через интерфейсы. Именно интерфейс, вернее указатель на него, является тем, с чем работает вызывающий процесс (программист). Объект может предоставлять несколько интерфейсов.

Объект COM, предоставляющий через интерфейсы свои функции, называют COM-сервером. Запрашивающая программа, соответственно, называется COM-клиент. Существует DCOM расширение COM, позволяющее добираться до объектов на других компьютерах. С точки зрения программирования DCOM это системный сервис, делающий COM прозрачным в локальных сетях.



OPC


В 1994 г. под эгидой Microsoft, была создана организация OPC Foundation. Её целью 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, Schneider Automation, Rockwell Software, Wonderware, Intellution, Ci Technologies.

OPC это аббревиатура от OLE for Process Control (OLE для Управления Процессами), т.е. это технология OLE на основе COM для промышленных применений.

Технология OPC реализована и продолжает реализовываться по схеме предоставления стандартизированных склеивающих интерфейсов.

Комитеты OPC Foundation делают следующее:

- разрабатывают спецификации COM-интерфейсов и COM-объектов;
- присваивают им идентификаторы (GUID);
- оформляют всё в виде стандартов и публикуют;
- генерируют или создают вспомогательные файлы, библиотеки для поддержки межпроцессного взаимодействия;
- разрабатывают вспомогательные компоненты, например, утилиты.



В настоящее время имеются следующие OPC-стандарты:

- OPC Common Definitions and Interfaces общие для всех OPC-спецификаций интерфейсы.
- 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-интерфейсов для конфигурирования оборудования, программирование на языках типа Visual Basic.
- OPC Alarms and Events Interface Specification спецификация 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 данными, программирование на C++.
- OPC Security Custom Interface спецификация COM-интерфейсов для обработки прав доступа к данным, программирование на C++.



OPC-интерфейс допускает различные варианты обмена: получение данных с физических устройств, из распределенной системы управления или из любого приложения. OPC-сервером можно снабдить контроллер, плату ввода/вывода, адаптер полевой шины, программу пересчёта, генератор случайных чисел, что угодно, лишь бы это могло поставлять или принимать данные.

Например предполагается, что тот, кто создаёт плату сбора данных, снабжает её не только драйвером, но и реализует OPC-сервер, работающий с этой с платой через драйвер или даже напрямую. Тем самым производитель предоставляет стандартный доступ к своей плате. Что должен сделать производитель, если он задался целью обеспечить свой продукт стандартным OPC-интерфейсом? Он должен получить нужную спецификацию и прилагаемые программные компоненты. Затем он должен изучить COM-интерфейсы тех COM-объектов этой спецификации, которые относятся в ней к модели OPC-сервера. И, наконец, реализовать с помощью нужных средств требуемые интерфейсы, а значит и OPC-сервер.

Раздел 2.

Интеграция в компьютерных технологиях автоматизации и управления



Определение промышленной сети

Промышленная сеть – это набор стандартных протоколов обмена данными, позволяющая связать воедино оборудование различных производителей, а также обеспечить взаимодействие нижнего и верхнего уровней АСУ.

Основные требования, предъявляемые к «идеальной» промышленной сети:

- высокая производительность;
- предсказуемость времени доставки информации;
- помехоустойчивость;
- доступность и простота организации физического канала передачи данных;
- максимальный сервис для приложений верхнего уровня;
- минимальная стоимость устройств аппаратной реализации, особенно на уровне контроллеров;
- возможность получения «распределенного интеллекта», путем представления максимального доступа к каналу нескольким ведущим узлам;
- управляемость и самовосстановление в случае возникновения нестандартных ситуаций.

В состав средств организации сети входят следующие компоненты:

- коммуникационная сеть, состоящая из среды передачи, средств для подключения к сети и передачи данных, а также соответствующие технологии передачи;
- протоколы и службы, используемые для обмена данными между перечисленными выше устройствами;
- модули, предназначенные для подключения программируемого контроллера или компьютера к ЛС (коммутационные процессоры)

Модель сети Международной организации по стандартизации

Модель сети Международной организации по стандартизации (ISO), призвана разграничить и формализовать функции, выполняемые различными уровнями аппаратного и программного обеспечения сетевой структуры. Данная модель определяет семь уровней сервиса, предоставляемого сетью.

7	Application	Прикладной уровень
6	Presentation	Уровень представления
5	Session	Уровень сессий
4	Transport	Транспортный уровень
3	Network	Сетевой уровень
2	Data Link	Канальный уровень
1	Physical	Физический уровень

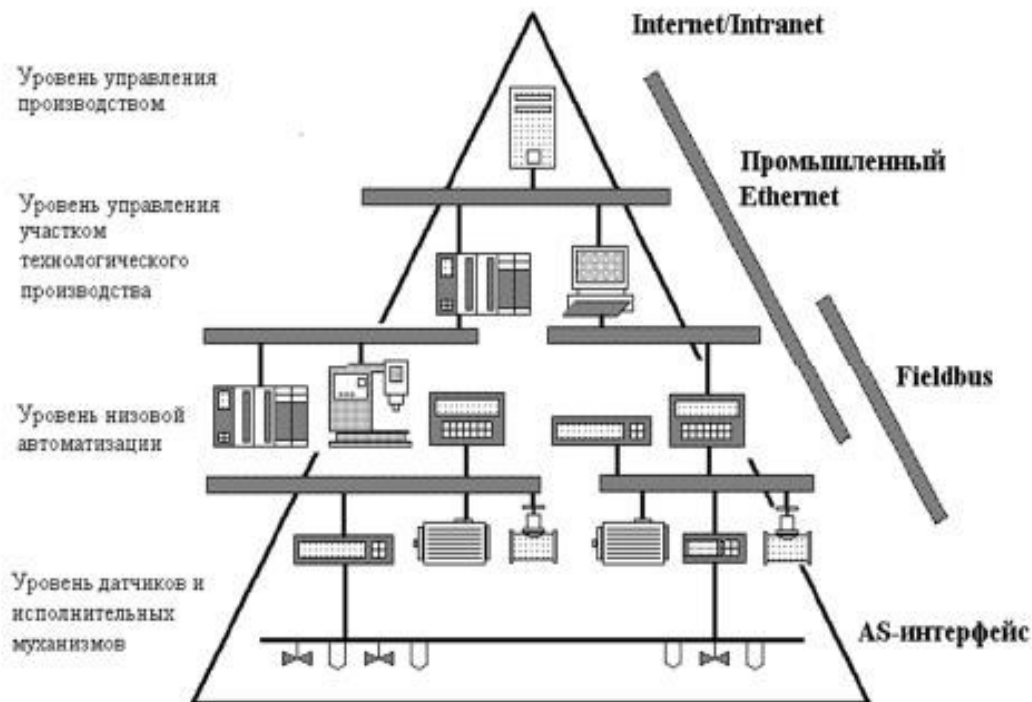
- На физическом уровне определяются физические характеристики канала связи и параметры сигналов.
- Канальный уровень формирует основную единицу передаваемых данных - пакет и отвечает за дисциплину доступа устройства к каналу связи (Medium Access Control) и установление логического соединения (Logical Link Control).
- Сетевой уровень отвечает за адресацию и доставку пакета по оптимальному маршруту.
- Транспортный уровень разбирается с содержимым пакетов, формирует ответы на запросы или организует запросы, необходимые для уровня сессий.
- Уровень сессий оперирует сообщениями и координирует взаимодействие между участниками сети.
- Уровень представления занимается преобразованием форматов данных, если это необходимо.
- Прикладной уровень – это набор интерфейсов, доступных программе пользователя.

На практике большинство промышленных сетей ограничивается только тремя из них, а именно физическим, канальным и прикладным. Наиболее «продвинутые» сети решают основную часть аппаратно, оставляя программную прослойку только на седьмом уровне.

Основные типы современных цифровых промышленных сетей

Семейство промышленных сетей представляет собой открытую систему связи, предназначенную для использования на всех уровнях иерархии систем автоматизированного управления в условиях промышленного производства.

- AS-интерфейс
- CAN протокол (сети SDS, CANOpen, DeviceNet)
- Interbus
- PROFIBUS
- Foundation Fieldbus
- Internet/Intranet



Основные характеристики AS интерфейса

AS-Interface - это открытый международный стандарт EN 50 295. AS-интерфейс (Actuator/Sensor Interface) был представлен в виде концепции в 1993 году. Поддерживается консорциумом ведущих производителей средств АСУТП, в числе которых фирмы Siemens, Pepperl+Fuchs и другие. Относится к классу ЦПС оконечных устройств, осуществляя непосредственную интеграцию датчиков и исполнительных механизмов в систему автоматизации. Позволяет полностью исключить из АСУ ТП аналоговые линии связи, кроссировочные шкафы и другое вспомогательное оборудование.



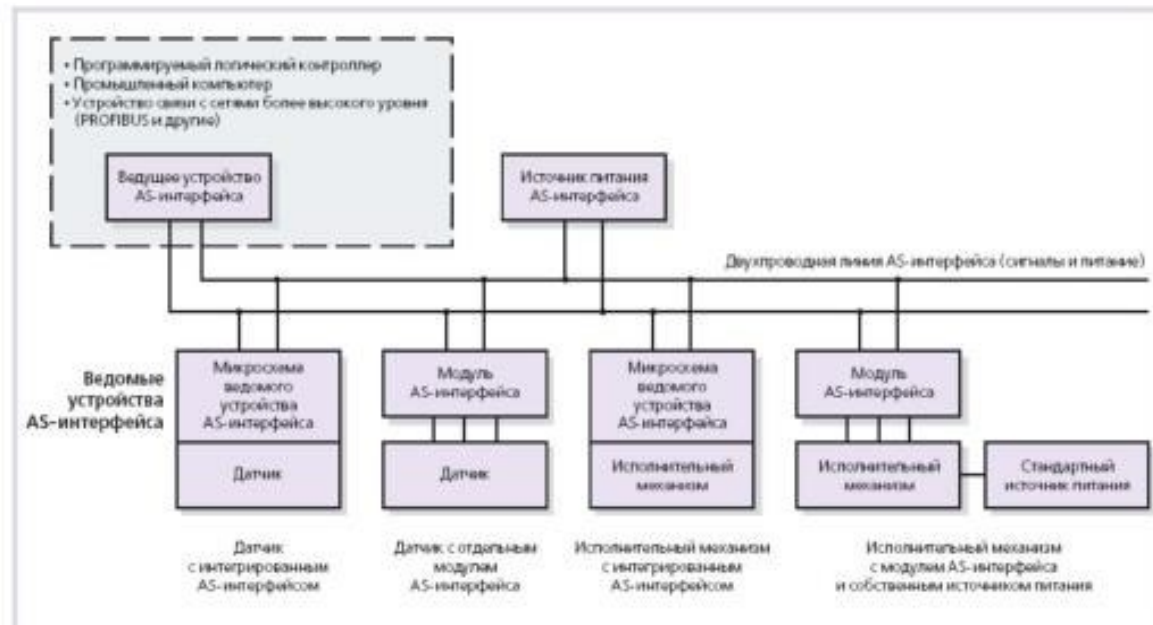
Отличительными чертами AS-интерфейса являются следующие основные характеристики:

- топология – произвольная («шина», «звезда», «дерево», «кольцо»);
- число ведущих устройств (master) – 1;
- число ведомых устройств (slave) – до 31;
- метод доступа - последовательный опрос;
- AS-интерфейс оптимален для подключения бинарных датчиков и исполнительных механизмов. Кабель AS-i используется как для обмена данными, так и для подачи напряжения питания на датчики/исполнительные механизмы.
- простой и экономичный монтаж соединений.
- малое время реакции: ведущему устройству требуется не более 5 мс для циклического обмена данными с 31 узлом сети.
- при использовании стандартных AS-i модулей на кабеле может находиться до 124 исполнительных механизмов/датчиков.
- если используются AS-i модули с расширенным режимом адресации, с одним ведущим устройством могут работать до 186 исполнительных механизмов и 248 датчиков.
- установка адресов устройств - автоматическая или ручным сервисным прибором;
- кабель - неэкранированный двух проводной с сечением 2x1,5 мм2 или специальный плоский;
- максимальная суммарная протяженность линий связи сегмента сети, обслуживаемого одним ведущим устройством, - 300м (с использованием повторителей);
- электропитание напряжением 30 В постоянного тока.

Конвейерная линия с элементами AS-интерфейса



Подключение устройств к AS интерфейсу

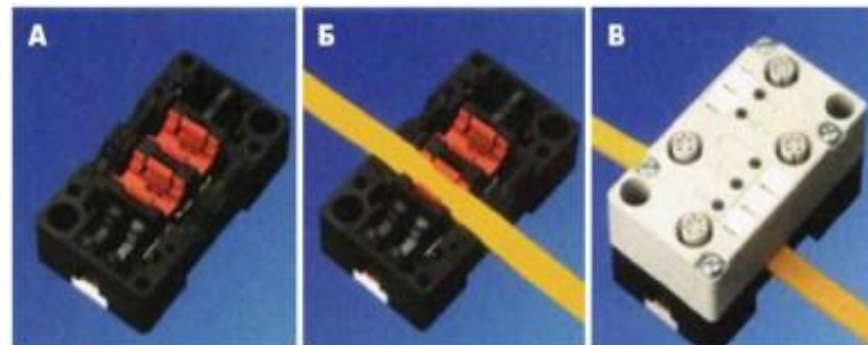
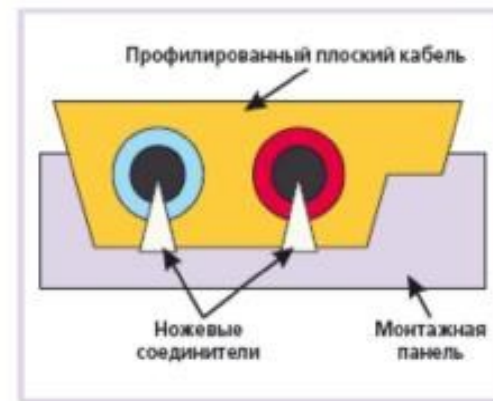


Для развертывания сегмента AS-сети необходимо следующее оборудование:

- кабель - специальный профилированный или обычный двухпроводный сечением 2x1,5 мм²;
- источник электропитания;
- ведомые устройства для подключения датчиков и исполнительных механизмов;
- ведущее устройство
- повторители (при необходимости);
- сервисный прибор для адресации и диагностирования.

Основные компоненты промышленной сети на базе AS интерфейса

Кабель AS - интерфейса имеет профилированную форму, исключая его неправильный монтаж. Быстрое и надежное подключение узлов к кабелю обеспечивается с помощью специальной конструкции. В нижней части корпуса монтируемого устройства находятся ножевые контакты, прорезающие кабель и обеспечивающие непосредственный контакт с токоведущими жилами. Несимметричная форма кабеля гарантирует точное попадание контактов в сердечник проводников и абсолютно надежное соединение в течение всего срока эксплуатации.

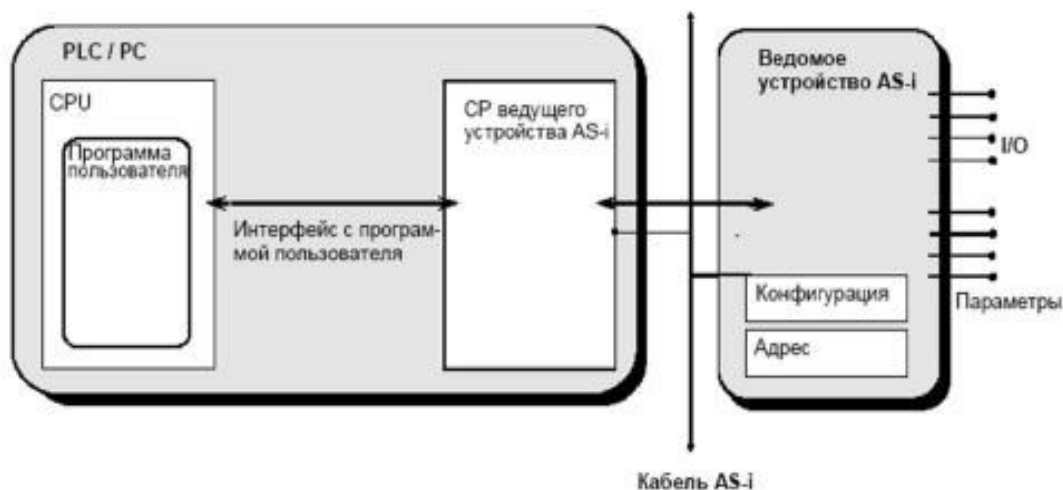


Принцип организации обмена

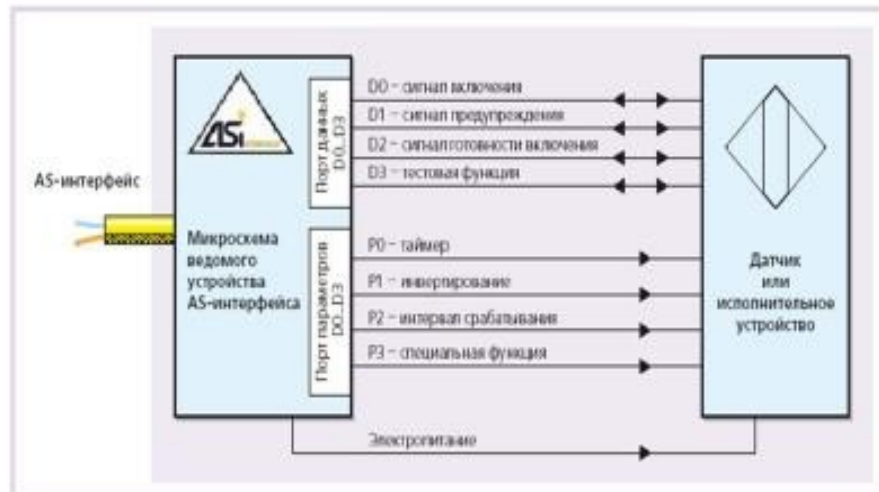
AS-интерфейс является системой с одним ведущим устройством и работает по принципу "ведущий – ведомый" (master-slave). Это означает, что ведущее устройство AS-интерфейса, подключенное к кабелю AS-i, управляет процедурой обмена данными с ведомыми устройствами, также подключенными к этому кабелю. На рисунке ниже показано два интерфейса, которыми обладает коммуникационный процессор, являющийся ведущим устройством AS-i.

- Через интерфейс, объединяющий CPU и CP ведущего устройства, передаются данные процесса и команды задания параметров. В программах пользователя предусматриваются необходимые обращения к функциям, имеются механизмы для чтения и записи данных через этот интерфейс.

- Обмен данными с ведомыми устройствами AS-i осуществляется через интерфейс между коммуникационным процессором ведущего устройства и кабелем AS-интерфейса.



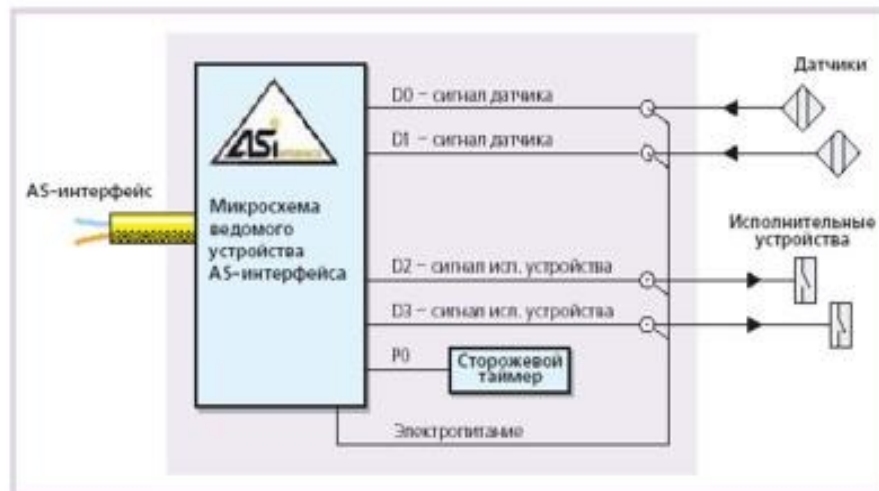
Схемы модулей ввода-вывода



В качестве ведомых узлов могут выступать либо датчики/исполнительные механизмы со встроенным AS-i интерфейсом, либо модули AS-i, к которым можно подключить до 4 обычных бинарных датчиков/исполнительных механизмов.

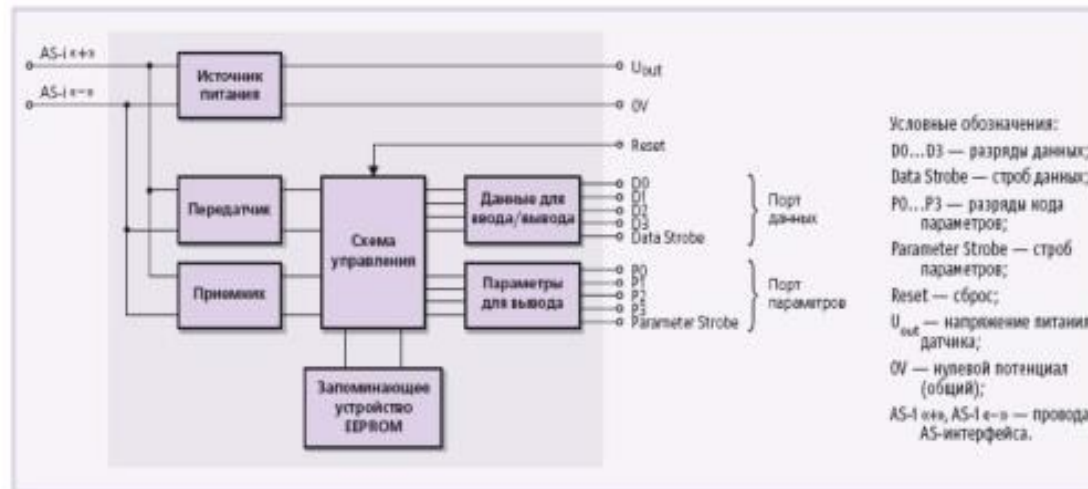
Электронная часть ведомого устройства, если её необходимо встраивать непосредственно в датчик или исполнительное устройство, должна быть компактной и, кроме того, дешевой. Это возможно только при использовании специализированных микросхем. Возможны два способа использования специализированных интерфейсных микросхем :

- чип ведомого устройства AS-интерфейса может быть встроен прямо в датчик или исполнительное устройство, в результате чего получается устройство с интегрированным AS-интерфейсом



- чип ведомого устройства AS-интерфейса может быть встроен в модуль таким образом, что к модулю можно подключать обыкновенный датчик или исполнительное устройство, которые характеризуются как устройства с внешним AS-интерфейсом

Структура специализированной интерфейсной БИС



Специализированная микросхема (ASIC) обеспечивает датчик или исполнительное устройство электропитанием от сети, распознает переданную от ведущего устройства информацию и посылает в ответ собственные данные.

В каждом цикле передаются 4 бита данных от ведущего устройства последовательно к каждому ведомому и обратно. Необходимые для этого порты данных каждой микросхемы можно конфигурировать отдельно как входные, выходные или двунаправленные порты. Конфигурация портов ведомых устройств устанавливается в соответствии с так называемой конфигурацией ввода-вывода.

По команде «*Write Parameter*» ведомое устройство получает от ведущего 4 бита данных, соответствующих значению параметра. С их помощью можно управлять особыми функциями ведомого устройства. Установка кодов параметров производится ациклично, причем в одном цикле AS-интерфейса она может быть выполнена только для одного ведомого устройства.



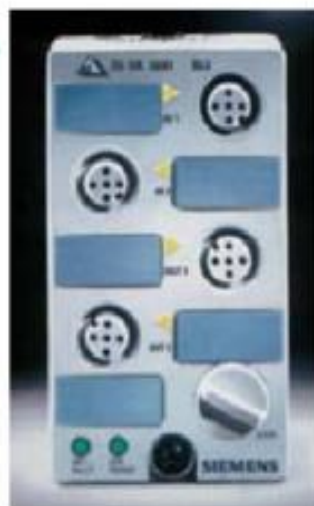
Разновидности модулей ввода-вывода

В качестве оконечных устройств сети AS-интерфейса могут выступать как датчики и исполнительные механизмы со встроенным AS-интерфейсом (интеллектуальные устройства), так и обычные оконечные устройства. Первые подключаются к сети напрямую, вторые - посредством модулей ввода-вывода (МВВ).

Разновидности МВВ по типу подключаемых устройств (все входные цепи гальванически развязаны относительно AS-интерфейса):

- 4 входа;
- 4 выхода;
- 2 входа + 2 выхода;
- 4 входа + 2 выхода;
- 4 входа + 3 выхода (согласно спецификации 2.1);
- 4 входа + 4 выхода (согласно спецификации 2.0);
- 2 аналоговых входа (ток 0_20/4_20 мА или напряжение 0...10 В);
- 2 аналоговых выхода (ток 0_20/ 4_20 мА или напряжение 0...10 В);
- 4 аналоговых входа от термодатчиков типа Pt 100.

Модули AS интерфейса с разъемом M12



Интеллектуальные оконечные устройства, такие как датчики приближения, фотоэлектрические датчики и др., как правило, имеют стандартный интерфейсный разъем M12.



Модули для монтажа в электрических шкафах



Внешний вид модуля VAA-4E

Внешний вид модуля серии KF

Монтаж в электрических
шкафах ведется на DIN-рейку

Модули для монтажа в распределительных коробках, модули с кабельными выводами





ПРОТОКОЛ CAN (Controller Area Network)

Данный протокол определяет только первые два уровня ISO/OSI – физический и уровень доступа к среде передачи данных. Разработчик – компания Bosch. На основе этого протокола реализовано большое количество полнофункциональных сетей, таких как SDS, CANOpen, DeviceNet и др.

Основными достоинствами, определившими высокую популярность этого протокола у разработчиков встраиваемых и промышленных систем, являются высокая скорость (до 1 Мбит/с), метод доступа CSMA/CA, возможность иметь в сети несколько ведущих устройств, надежная система обнаружения и исправления ошибок. CSMA/CA сочетает минимальную задержку передачи информации с эффективным арбитражем ситуаций, когда несколько узлов начинают передавать данные одновременно. Благодаря этому гарантируется доставка сообщения, то есть система является детерминированной. Технические характеристики для DeviceNet: максимальное расстояние 500 м, максимальное количество узлов 64, длина информационной посылки 8 байт, используемый кабель Belden 3082A.

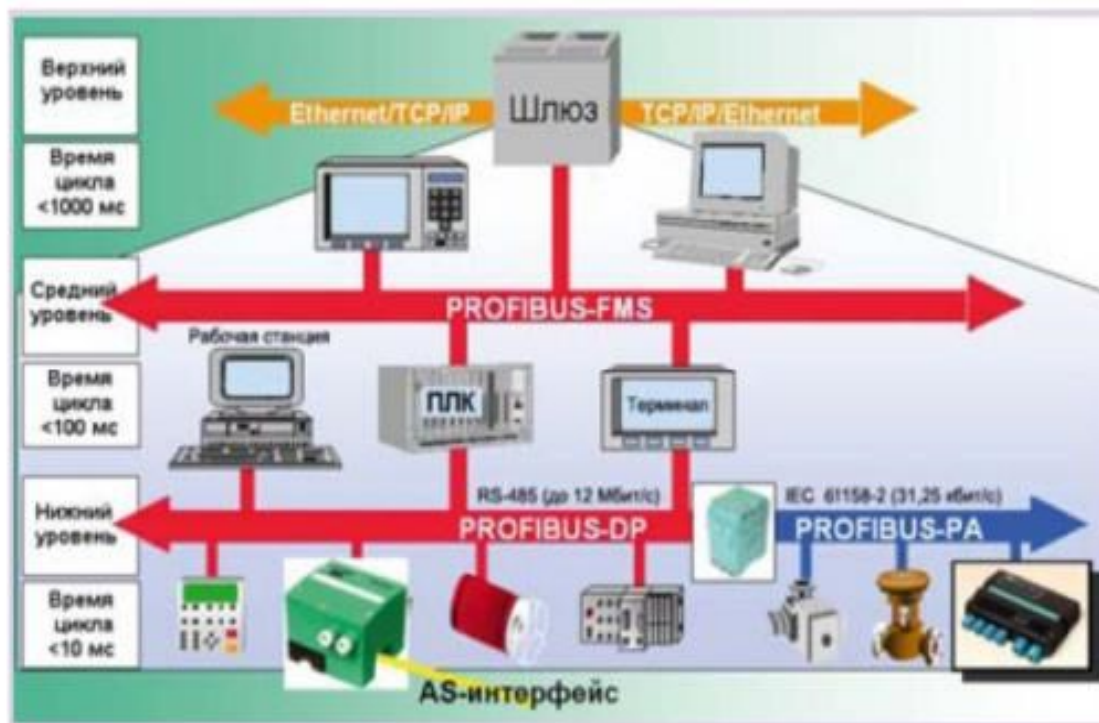


Спецификация Interbus

Спецификация Interbus была разработана фирмой Phoenix Contact в 1984 г. Физический уровень основан на стандарте RS-485. Максимальное расстояние, которое может охватывать эта ЦПС – до 13 километров обеспечивается благодаря ретрансляции сигнала в каждом узле. Максимальное количество узлов 512, расстояние между узлами до 400 метров, используемый кабель Belden 3119A.

Узлы-ретрансляторы образуют основу топологии Interbus, оконечные же устройства подключаются к дополнительным кольцевым сегментам, в которых питающее напряжение передается вместе с данными. Длина дополнительных сегментов может составлять до 200 метров, для их прокладки используется обычная неэкранированная витая пара. Доступ к среде передачи данных в Interbus организован по принципу суммирующего фрейма и обеспечивает гарантированное время передачи информации.

Сети PROFIBUS



Это целое семейство ЦПС, обеспечивающее комплексное решение коммуникационных проблем предприятия, было разработано фирмой Siemens в начале 90-х годов прошлого века. Это коммуникационная сеть полевого уровня и уровня отдельных производственных участков, базирующаяся на стандарте PROFIBUS EN 50170.1.2 или IEC 61158.2, в которой используется гибридный метод доступа к шине (маркерное кольцо между активными узлами и ведущий-ведомый, между активными и пассивными узлами). Средой передачи может являться витая пара, волоконно-оптический кабель или беспроводная среда.



Три модификации PROFIBUS

- **PROFIBUS-DP** сеть нижнего уровня, обеспечивающая высокоскоростной обмен данными с оконечными устройствами. Применяет уровни 1 и 2 модели OSI, а также пользовательский интерфейс. Протокол физического уровня соответствует стандарту RS-485. Скорость обмена прямо зависит от длины сетевого сегмента и варьируется от 100 кбит/с на расстоянии 1200 метров до 12 Мбит/с на дистанции до 100 метров. Взаимодействие узлов в сети PROFIBUS определяется моделью «Master-slave». Master сегмента последовательно опрашивает подключенные узлы и выдает команды в соответствии с заложенной в него технологической программой. Этот профиль протокола PROFIBUS оптимизирован для быстрого обмена данными специально для коммуникаций между системами автоматизации и децентрализованной периферией на полевом уровне.
- **PROFIBUS-FMS** сеть более высокого уровня, ориентированная на обеспечение информационного обмена одноранговых устройств. В PROFIBUS-FMS применяются уровни 1,2 и 7. Пользовательский уровень состоит из FMS (Fieldbus Message Specification) и LLI (Lower Layer Interface). FMS содержит пользовательский протокол и предоставляет в распоряжение коммуникационные службы. LLI реализует различные коммуникационные связи и создает для FMS аппаратно- независимый доступ к уровню 2. FMS применяется для обмена данными на уровне ячеек (PLC и PC). Мощные FMS- сервисы открывают широкие области использования и большую гибкость при передаче больших объемов данных.
- **PROFIBUS-PA** – применяет расширенный PROFIBUS-DP-протокол передачи данных, Интерфейс, соответствующий требованиям стандарта IEC 61158-2 обеспечивает надежность и питание полевых приборов через шину. Может применяться для построения сети, соединяющей исполнительные устройства, датчики и контроллеры, расположенные непосредственно во взрывоопасной зоне. Сегмент PROFIBUS-PA может иметь длину до 1900 метров со скоростью обмена между узлами 31,25 кбит/с. На прикладном и канальном уровнях PROFIBUS-PA использует весь сервис, доступный в PROFIBUS-FMS.

PROFIBUS-DP и PROFIBUS-FMS применяют одинаковую технику передачи и единый протокол доступа к шине и поэтому могут работать через общий кабель.

Если требуется объединить в детерминированную сеть несколько контроллеров, оптимальным вариантом будет PROFIBUS-FMS. Для создания сети с централизованным интеллектом и распределенным вводом-выводом полевого уровня лучше всего подойдет PROFIBUS-DP.




FOUNDATION FIELDBUS

Один из самых распространенных стандартов ЦПС, появившийся на свет в 1995 г. По многим параметрам эта система схожа с PROFIBUS-PA: возможность установки во взрывоопасных зонах, передача информационного сигнала вместе с питающим напряжением по одной паре проводов, двухуровневая иерархия и т.д.

Среди других ЦПС Foundation Fieldbus выделяют две особенности:

- Разработан специальный язык описания оконечных устройств (Devise Description Language), использование которого позволяет подключать новые узлы к сети по широко применяемой в обычных IBM PC совместимых компьютерах технологии plug-and-play. Пользователям доступны как типовые DD для стандартных устройств (клапанов, датчиков температуры и т.д.), так и возможность описания нестандартных изделий.
- Foundation Fieldbus ориентирована на обеспечение одноранговой связи между узлами без центрального ведущего устройства. Этот подход ориентирует на реализацию системы управления, распределенную не только физически, но и логически, что во многих случаях позволяет повысить надежность и живучесть АСУ ТП.





Industrial Ethernet


Решения на базе Ethernet практически вытеснили все остальные из офисных распределенных приложений. Поэтому следует считаться с желанием пользователей распространить сферу применения Ethernet в промышленные цеха на уровень низовой автоматике.

Требования, предъявляемые к системам связи, работающим в условиях промышленного производства, существенно отличаются от требований к традиционным офисным системам связи. Это затрагивает практически все составляющие системы связи, в частности, активные и пассивные компоненты сети, подключаемое оконечное оборудование данных (ООД), концепцию построения сети и сетевые топологии, требования к надёжности, интенсивность потока данных и требования к среде окружения. Также имеются протоколы, оптимизированные специально для систем промышленной связи, хотя в последнее время в системах управления производством и технологическими процессами всё больше укрепляются позиции протокола TCP/IP, классического протокола, распространённого в офисных системах связи.

Концепция Industrial Ethernet базируется на использовании существующих стандартов (стандарт для сетей Ethernet IEEE 802.3), которые дополнены необходимыми и существенными нюансами, характерными для систем промышленной связи.

Первый стандарт Ethernet 10BASE5 был введён международным Институтом инженеров по электротехнике и электронике (IEEE) в 1985 году. Стандарт базировался на использовании в качестве среды передачи коаксиального кабеля. С момента своего появления и стандарт IEEE непрерывно улучшался в качестве. Примером является использование в качестве канала связи волоконно-оптических кабелей и витых пар, а также появление стандарта Fast Ethernet, увеличившего скорость передачи данных в 10 раз.

Общим свойством всех версий сети Ethernet является немодулированная передача данных и метод доступа CSMA/CD.



Немодулированная передача данных

Согласно IEEE 802.3, в Ethernet используется немодулированная передача данных. Это означает, что в канале связи передаются немодулированные импульсные сигналы. Среда распространения сигнала формирует единый канал связи, ресурсы которого должны использоваться одновременно всеми подключенными терминальными устройствами (ООД). Все подключенные терминальные устройства принимают передаваемую информацию одновременно. В любое время правом на передачу данных обладает лишь одно терминальное устройство. Если несколько терминальных устройств передают данные одновременно, в канале связи возникает коллизия. Сигналы терминальных устройств, пытающихся передавать данные одновременно, подавляют друг друга. Совершенно очевидно, что возникает необходимость в координировании доступа к среде передачи, используемой совместно. В стандарте IEEE 802.3 для решения этой проблемы используется протокол CSMA/CD.



Управление доступом с помощью протокола CSMA/CD

Протокол CSMA/CD (множественный доступ с опросом несущей и обнаружением коллизий) также известен как протокол LWT (Listen While Talk .буквально, «говори, слушая»). Такой метод доступа является распределённым, поскольку все терминальные устройства, подключенные к сети, обладают равными правами.

Если терминальное устройство собирается передавать данные, оно сначала «слушает», не передаются ли по каналу связи данные другими терминальными устройствами. Если другие терминальные устройства не передают данные, оно может начать передачу. Если же терминальное устройство обнаружило, что среда передачи уже используется другим ООД, оно должно дождаться освобождения канала связи. Все терминальные устройства «слышат» передаваемые данные. Информация об адресе назначения, содержащаяся в данных, позволяет терминальному устройству распознать, должно оно принимать данные или нет. Если несколько терминальных устройств собираются передавать данные одновременно, и оба они обнаружили, что канал связи свободен, они начинают передачу. Спустя короткое время произойдёт «столкновение» передаваемых данных.

Терминальные устройства снабжены механизмом, который позволяет им обнаруживать такие коллизии. Все терминалы, оказавшиеся участниками коллизии, прекращают передачу и в течение некоторого времени, величина которого случайна и рассчитывается для каждого отдельного терминала поразному, вновь предпринимают попытку передачи данных. Так повторяется до тех пор, пока один из терминалов не добьётся успешной передачи в отсутствие коллизий. Другие терминалы ожидают освобождения канала связи.

Техника доступа CSMA/CD функционирует без ошибок в сети Ethernet, протяжённость которой ограничена максимальным допустимым временем распространения пакета данных. Расстояние, в пределах которого протокол CSMA/CD работает корректно, называют доменом обнаружения коллизий или "коллизийным доменом". В классических сетях Ethernet (10 Мбит/с) домен обнаружения коллизий имеет протяжённость 4520 м.



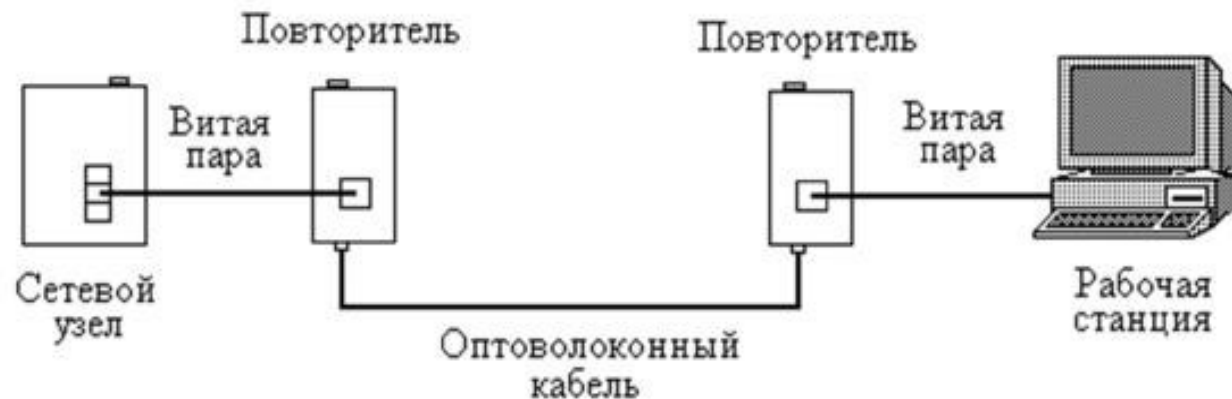
Основные принципы коммутируемых сетей

Технология коммутирования заключается в направлении пакетов данных от входного порта напрямую к выходному порту на основе информации об адресе, передаваемой с пакетом данных. Переключение позволяет установить прямое соединение. Технология коммутирования выполняет следующие основные функции:

- **Объединение доменов обнаружения коллизий / подсетей.** Поскольку функционирование репитеров (повторителей) и хабов (звездообразных разветвителей) протекает на физическом уровне, их использование ограничено протяжённостью домена обнаружения коллизий. Технология коммутирования позволяет объединить домены обнаружения коллизий. Таким образом, использование коммутаторов не ограничено максимальной протяжённостью сетей с повторителями. Коммутаторы позволяют реализовать очень протяжённые сети, охватывающие расстояния до 150 км.
- **Защита от перегрузки.** За счёт фильтрации данных, в основе которой лежат адреса Ethernet (MAC), локальные данные остаются локальными. В отличие от повторителей или хабов, которые транслируют данные на все порты / узлы сети, не выполняя фильтрации, коммутаторы осуществляют селекцию данных. Данные, поступившие на входной порт, фильтруются, и на соответствующий выходной порт коммутатора поступают только те данные, которые предназначены узлам в других подсетях, подключенных к данному порту. Это становится возможным в результате создания таблицы назначения адресов Ethernet (MAC) выходным портам. Таблица создаётся коммутатором в режиме обучения.
- **Нераспространение ошибок за пределы поврежденного сегмента сети.** Выполняя проверку достоверности пакета данных, базирующуюся на контрольной сумме, содержащейся в каждом пакете данных, коммутатор предотвращает дальнейшую передачу повреждённых пакетов данных. Коллизии, имеющие место в одном сегменте сети, не распространяются в другие сегменты.
- **Параллельный обмен данными.** Одним из свойств коммутаторов является их способность манипулировать несколькими пакетами данных, передаваемыми между различными сегментами сети или узла, одновременно. В зависимости от количества портов, имеющихся у коммутатора, он устанавливает несколько временных и динамических соединений между различными парами сегментов / терминальных устройств сети. В результате наблюдается существенное увеличение пропускной способности сети и значительное повышение её производительности.

Повторитель

Повторитель предназначен для соединения разнородных сегментов сети Ethernet и преодоления проблем, связанных с ограничениями длины сегмента кабеля. Повторители представляют собой устройства с двумя портами. В соответствии со стандартом IEEE 802.3 повторители осуществляют постоянный контроль состояния подключенных сегментов путем передачи в линию специальных служебных сигналов (при отсутствии информационных пакетов); если повторитель не получает по какому либо порту таких сигналов, то фиксируется обрыв линии.



Концентраторы и коммутаторы

Концентратор – это много портовый повторитель сетевого интерфейса с равноправными портами. Получив сигнал от одной из подключенных к нему станций, концентратор транслирует его на все свои активные порты. Концентраторы можно использовать как автономные устройства или соединять друг с другом, увеличивая тем самым размер сети и создавая более сложные топологии. Их основное назначение – объединение отдельных рабочих мест в рабочую группу в составе локальной сети. Концентраторы работают на физическом уровне (Уровень 1 базовой эталонной модели OSI) и не чувствительны к протоколам верхних уровней. К их задачам относятся: фазовая и временная синхронизация получаемых пакетов данных; удаление пакетов некорректной длины, которые могут появляться в результате коллизий; обработка коллизий в соответствии со стандартом IEEE 802.3.



Коммутаторы являются более интеллектуальными, чем концентраторы, устройствами. Коммутатор Ethernet поддерживает внутреннюю таблицу соответствия портов адресам подключенных к ним сетевых узлов. Эту таблицу администратор сети может создать самостоятельно или задать режим её автоматического формирования встроенными средствами устройства. Используя таблицу адресов и содержащийся в передаваемом пакете адрес получателя, коммутатор направляет полученный пакет только в тот порт, где находится адресат. Исключение делается только в случае широковещательных рассылок или при передаче пакетов с неизвестным адресом получателя, которые рассылаются по всем подключенным соединениям. На основе описанной процедуры коммутатор фактически выполняет важнейшую функцию сегментирования сети Ethernet, что в конечном счёте значительно расширяет её суммарную пропускную способность.

Коммутируемые ЛС / ЛС совместного доступа



Коммутируемая ЛС

- Каждый отдельный сегмент полностью функционален и обладает максимальной скоростью передачи данных.
- Одновременный поток данных в нескольких сегментах; передача нескольких фреймов
- Фильтрация: Локальные данные остаются локальными; за пределы сегмента передаются только выбранные пакеты

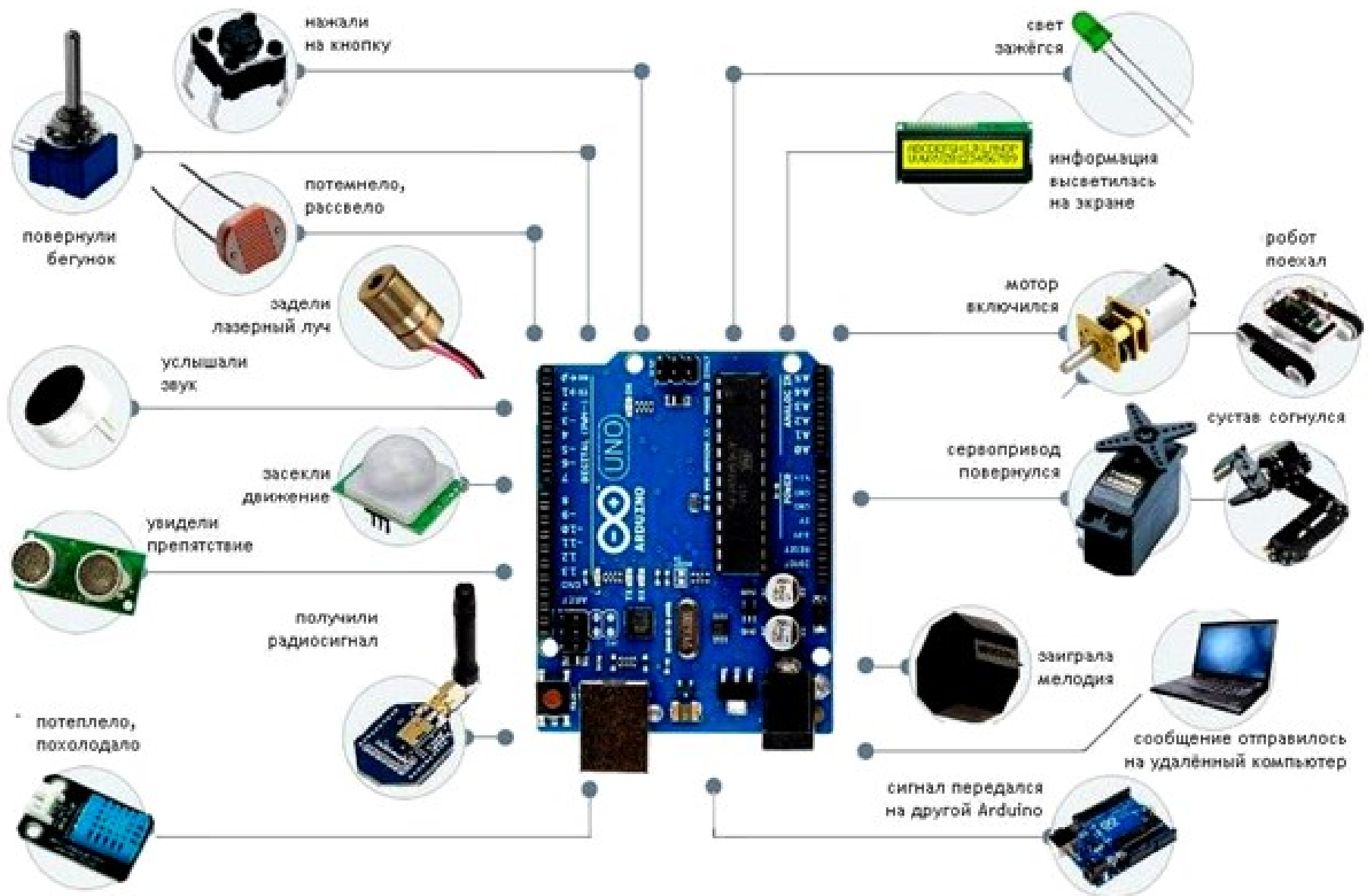
ЛС совместного использования

- Все узлы сети одновременно используют ресурсы канала связи / скорость передачи
- Все пакеты данных проходят через все сегменты
- В любое время по сети может передаваться лишь один фрейм
- Наличие коллизий снижает производительность сети, приблиз. на 40 %

Рисунок 2–1 Коммутируемые ЛС / ЛС совместного доступа

Еще на уровне офисных приложений была решена проблема недетерминированности Ethernet. Переход от концентраторов (hub) к коммутаторам (switch) и от полудуплексных каналов связи к дуплексным позволил снять вопрос о возможности блокировки обмена по сетевому каналу из-за многочисленных коллизий информационных кадров. Благодаря своим «интеллектуальным» возможностям коммутатор направляет полученный информационный кадр только на то подключение, где реально находится абонент, а не широкопередает во всю сеть. В результате общий объем трафика в сети многократно сокращается. Фактически топология «общая шина» на логическом уровне трансформируется в топологию «каждый с каждым», обеспечивая гарантированную доставку данных.

Кроме того, одним из основных препятствий к применению Ethernet в АСУ ТП было несоответствие между исполнением аппаратных средств и условиями их применения в промышленности. Сейчас появился целый ряд концентраторов и коммутаторов, выполненных в соответствии с требованиями промышленных условий эксплуатации.



повернули бегунок

нажали на кнопку

потемнело, рассвело

засекли лазерный луч

услышали звук

засекли движение

увидели препятствие

получили радиосигнал

потеплело, похолодало

свет зажёгся

информация высветилась на экране

мотор включился

робот поехал

сервопривод повернулся

сустав согнулся

заиграла мелодия

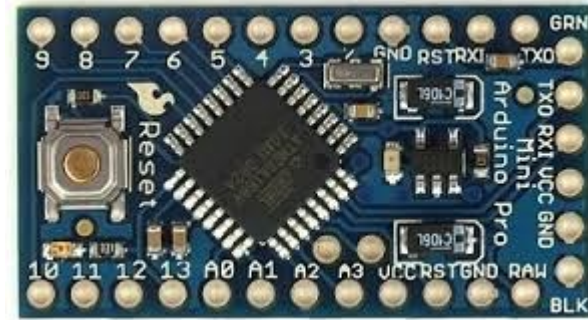
сообщение отправилось на удалённый компьютер

сигнал передан на другой Arduino

Виды Arduino



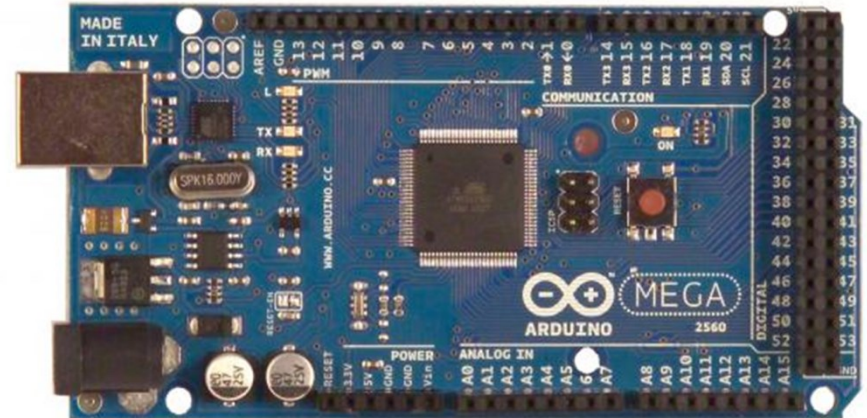
Arduino "Diecimila"



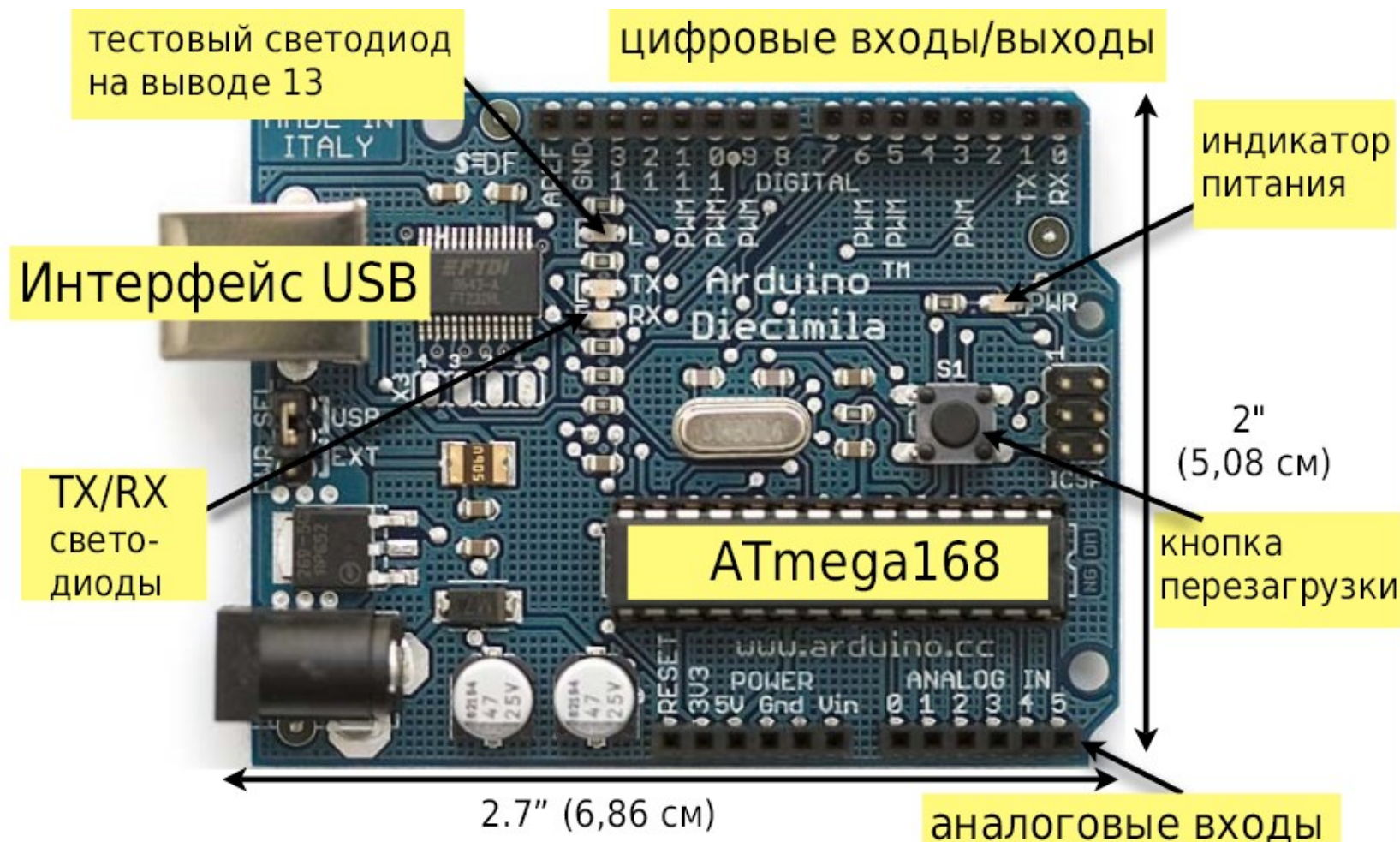
Arduino "Nano"



Arduino "Uno"



Arduino "Mega"



Arduino поддерживает последовательный интерфейс, осуществляемый выводами 0 (RX) и 1 (TX).

Светодиоды RX и TX на платформе будут мигать при передаче данных через USB подключение .

Pin 0 и 1 в программах лучше не использовать.

MADE
IN ITALY

AREF GND 13 12 11 10 9 8 7 6 5 4 3 2
TXD → 1 RXD ← 0
TX3 14 RX3 15 TX2 16 RX2 17 TX1 18 RX1 19 SDA 20 SCL 21

COMMUNICATION

22
24
26
28
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
DIGITAL
GND

WWW.ARDUINO.CC



MEGA

2560

ARDUINO

RESET-EN

RESET

ANALOG IN
A0 A1 A2 A3 A4 A5 A6 A7

POWER
3.3V 5V GND GND Vin

RESET

100 47 25V
100 47 25V

SPK16.000Y

TX RX



В настоящее время в мире выпускается очень много клонов Ардуино, абсолютно совместимых с Ардуино. Мы будем работать с «Arduino UNO» и с «Arduino MEGA». «Arduino UNO» имеет 14 контактов цифрового ввода/вывода, шесть из которых могут быть использованы для вывода сигналов ШИМ, и шесть аналоговых входов АЦП с дискретностью в 1024 значений. Она также содержит: кварцевый резонатор на 16 МГц, разъем USB , разъем питания типа Mini-Jack, разъем для подключения внешнего программатора ICSP и кнопку сброса. Микроконтроллер ATmega168 имеет 16 К программной памяти (из которых 2 К используются для хранения загрузчика). Кроме того, микросхема имеет 1 Кбайт ОЗУ (SRAM), 512 байт долговременной памяти данных (EEPROM).

Возможна замена управляющего микроконтроллера на ATmega328, что увеличивает объем доступной памяти в 2 раза: память программ - 32 К (2 К заняты загрузчиком), ОЗУ - 2 Кбайт, EEPROM - 1 Кбайт.

«Arduino MEGA» имеет 54 контакта цифрового ввода/вывода, четырнадцать из которых могут быть использованы для вывода сигналов ШИМ, и шестнадцать аналоговых входов АЦП с дискретностью в 1024 значений.

Каждый из цифровых выводов модуля можно использовать как ввод или как вывод данных. Эти выводы работают с сигналами уровнем 0...5В. Каждый из выводов рассчитан на входной (или выходной) ток до 20 мА. Выводы 3, 5, 6, 9, 10 и 11 могут работать в режиме выходов сигналов широтно-импульсного модулирования ШИМ. Это позволяет управлять яркостью горения светодиодов или скоростью вращения электромоторов.

Модуль также имеет аналоговые входы, подключенных к аналого-цифровому преобразователю (АЦП), который обеспечивает 10 разрядное аналого-цифровое преобразование (т.е. различает 1024 значений уровня сигнала). Входы АЦП настроены на диапазон входного напряжения от 0 до 5В.

Приобретенный в готовом виде модуль содержит все необходимое для разработки микроконтроллерных устройств. Нужно просто подсоединить модуль к компьютеру при помощи стандартного USB кабеля.

В качестве периферии могут быть подключены любые устройства - от светодиодов и простейших датчиков до механизмов радиуправляемых моделей и роботов.

1. Программы для Ардуино пишутся на языке Wiring, условно можно считать что это урезанный C и некоторые функции специфичные для платы Ардуино, подача/чтения сигнала с выходов, перевод контроллера в определенный режим работы и другие.

Для удобства работы с Arduino существует бесплатная интегрированная среда разработки «Arduino IDE» —включающее в себя редактор кода, компилятор и модуль передачи введенной программы (прошивки) в плату. Она является кроссплатформенной и поэтому работает и операционных системах Windows, Mac OS и Linux.

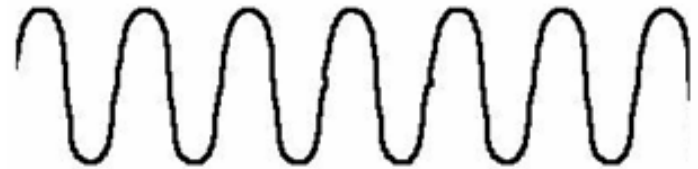
Для получения ответной информации с борта «Arduino» в этой программе предусмотрен простой Serial монитор, который выводит на свой экран «Serial Monitor» информацию в режиме реального времени

2. Для вывода информации, полученной с борта «Arduino» на экран монитора компьютера или видеопроектора в учебных или иных целях можно использовать различные программы – PascalABC, Delphi, C++, мультимедийную среду программирования Processing и т.д.

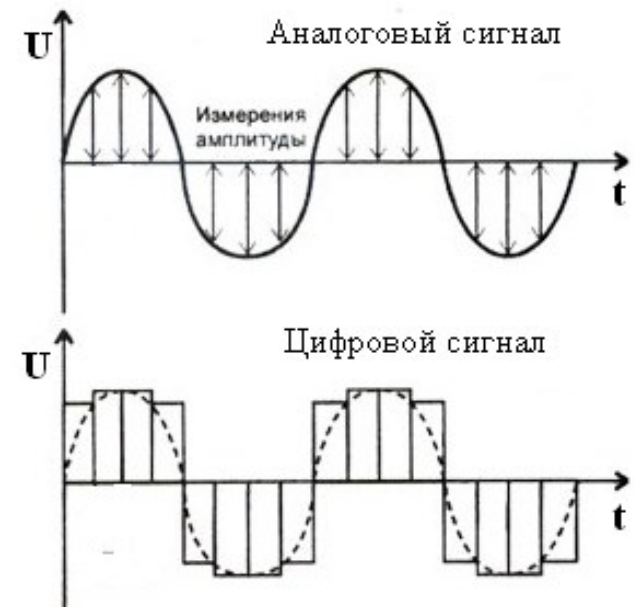
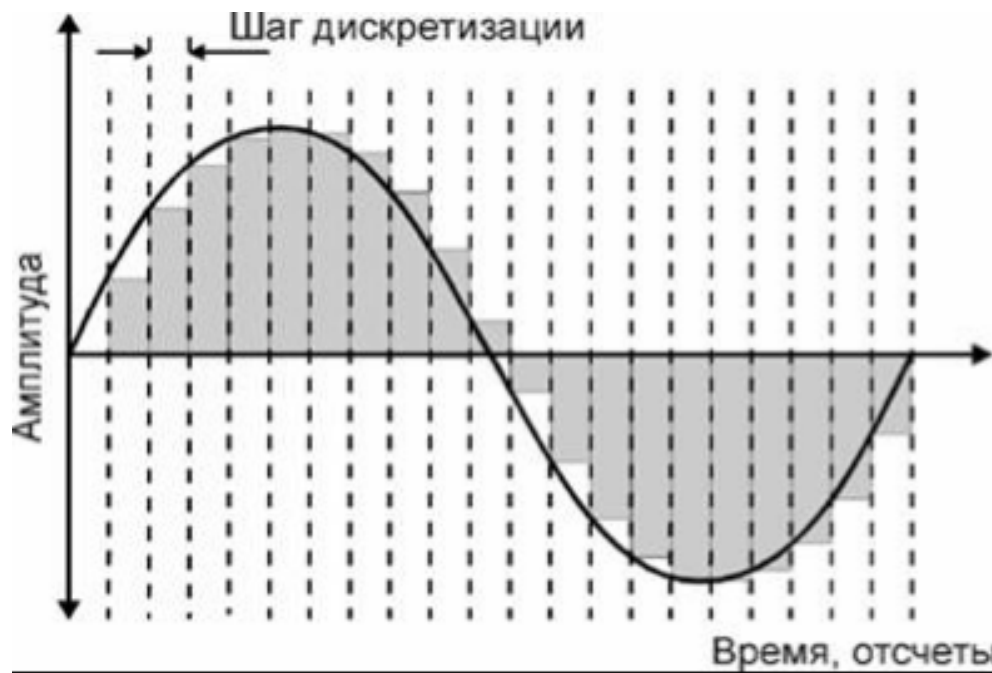
Цифровой и аналоговый сигналы.



Цифровой сигнал

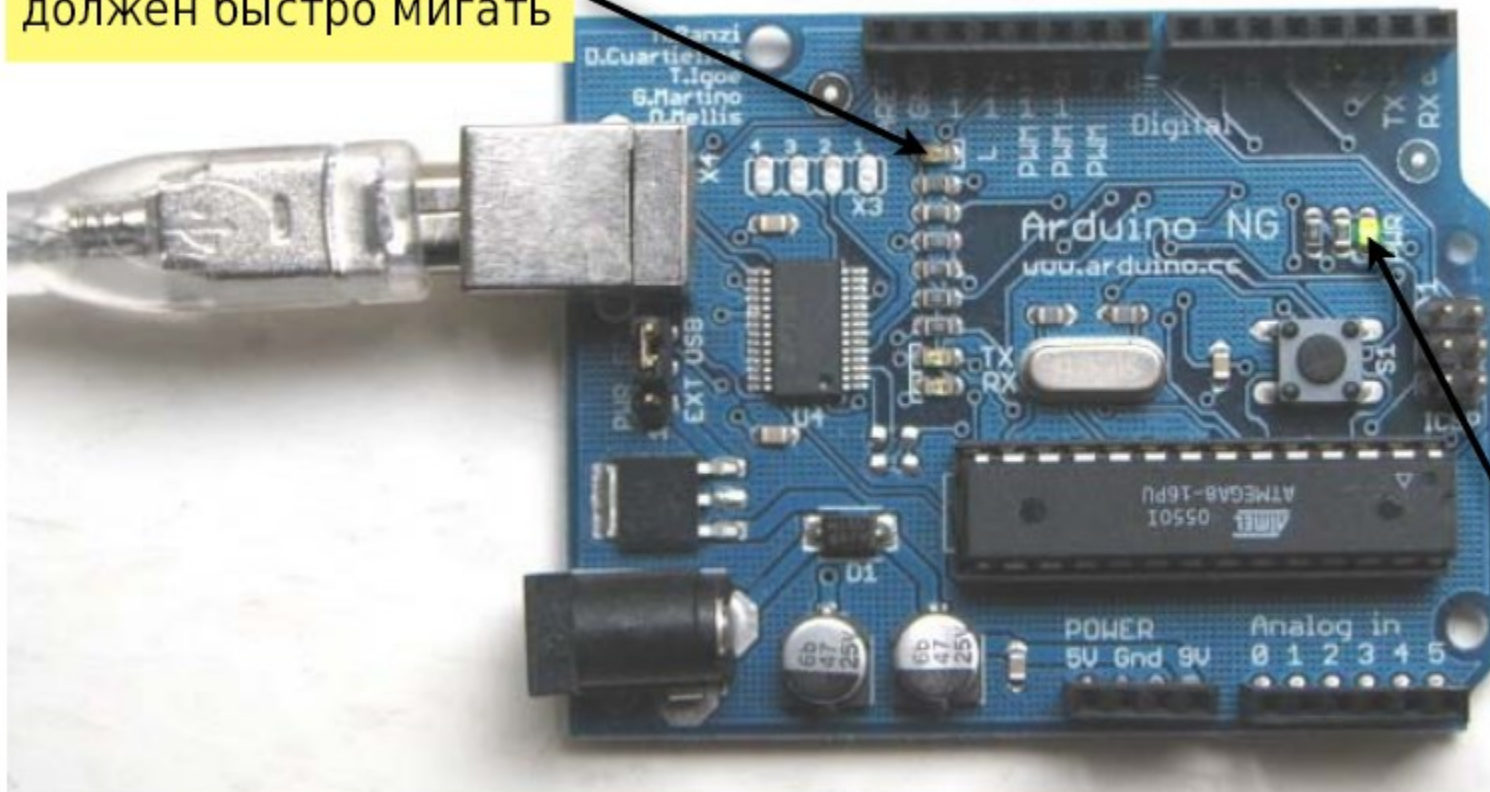


Аналоговый сигнал



Подключение платы Arduino

тестовый светодиод
должен быстро мигать



Индикатор питания должен светиться

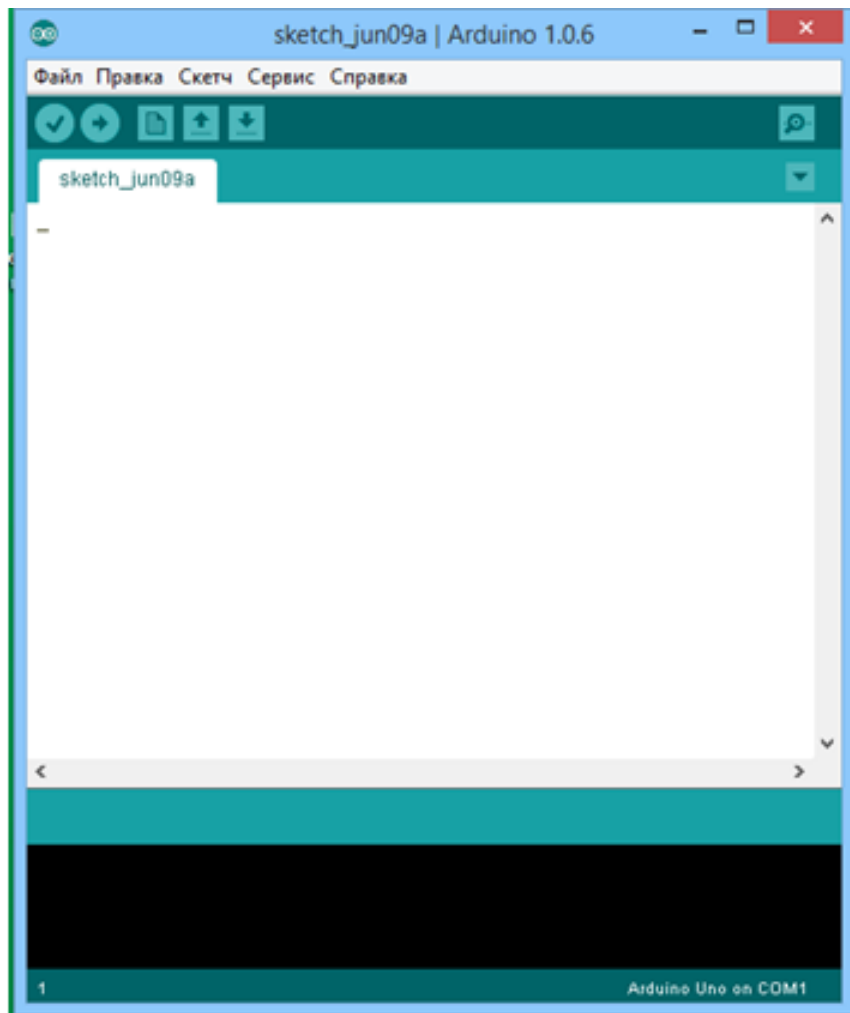
Плата готова к работе.

Установка драйвера Arduino . Проверка установки и определение номера виртуального COM порта для Arduino.

Чтобы начать работать с «Arduino» сначала необходимо установить в компьютер драйвер этого устройства. Драйвер это небольшая программ, которая позволяет компьютеру определить какое устройство к нему подключено. Кроме того в драйвере находится инструкция для компьютера как работать с данным устройством.

Теперь вопрос – а где взять драйвер ? А вот где – в программе, которая называется, так же как и устройство – «Arduino». Зайдите в любую поисковую систему и введите : «Скачать программу «Arduino». В окне поисковика Вам будет предложено много сайтов с которых можно скачать эту программу. Скачайте последнюю версию «Arduino 1.0.6». В разделе «Загрузка» появится файл архива «Arduino-1.0.6-windows.zip», распакуйте и нажмите левой кнопкой мышки. Начнется инсталляция «Arduino». Во время инсталляции соглашайтесь со всем, что Вам предлагает программа инсталляции. По окончании инсталляции перезагрузите компьютер – на рабочем столе монитора компьютера появится иконка программной среды «Arduino IDE» (рис. 2.1).





Для загрузки среды «Arduino IDE» кликаем левой кнопкой мышки по



иконке «Arduino» и на экране монитора появляется среда «Arduino IDE»

Вы видите, что в этой среде есть несколько меню : Файл, Правка, Скетч, Сервис и Правка. Все меню Вам знакомы, за исключением меню «Скетч».

Не пугайтесь – «Скетч» – это другое название программы для этой среды.



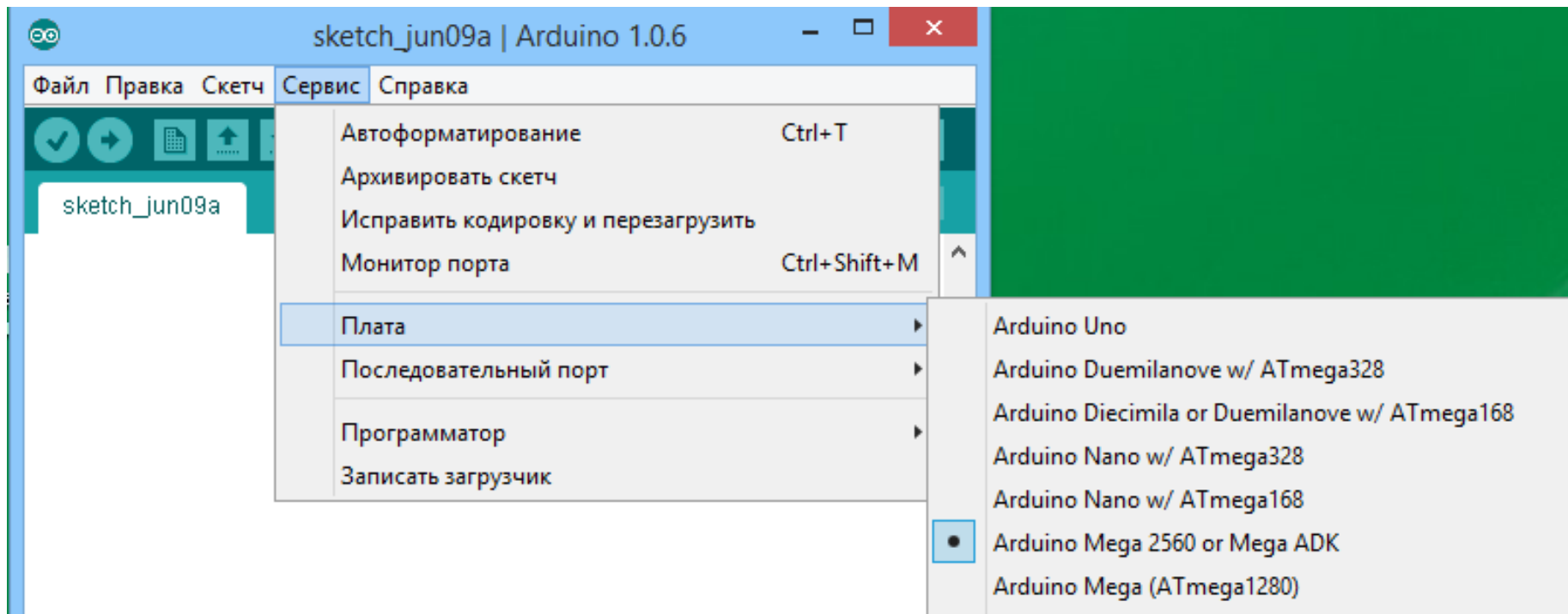
1 2 3 4 5

6

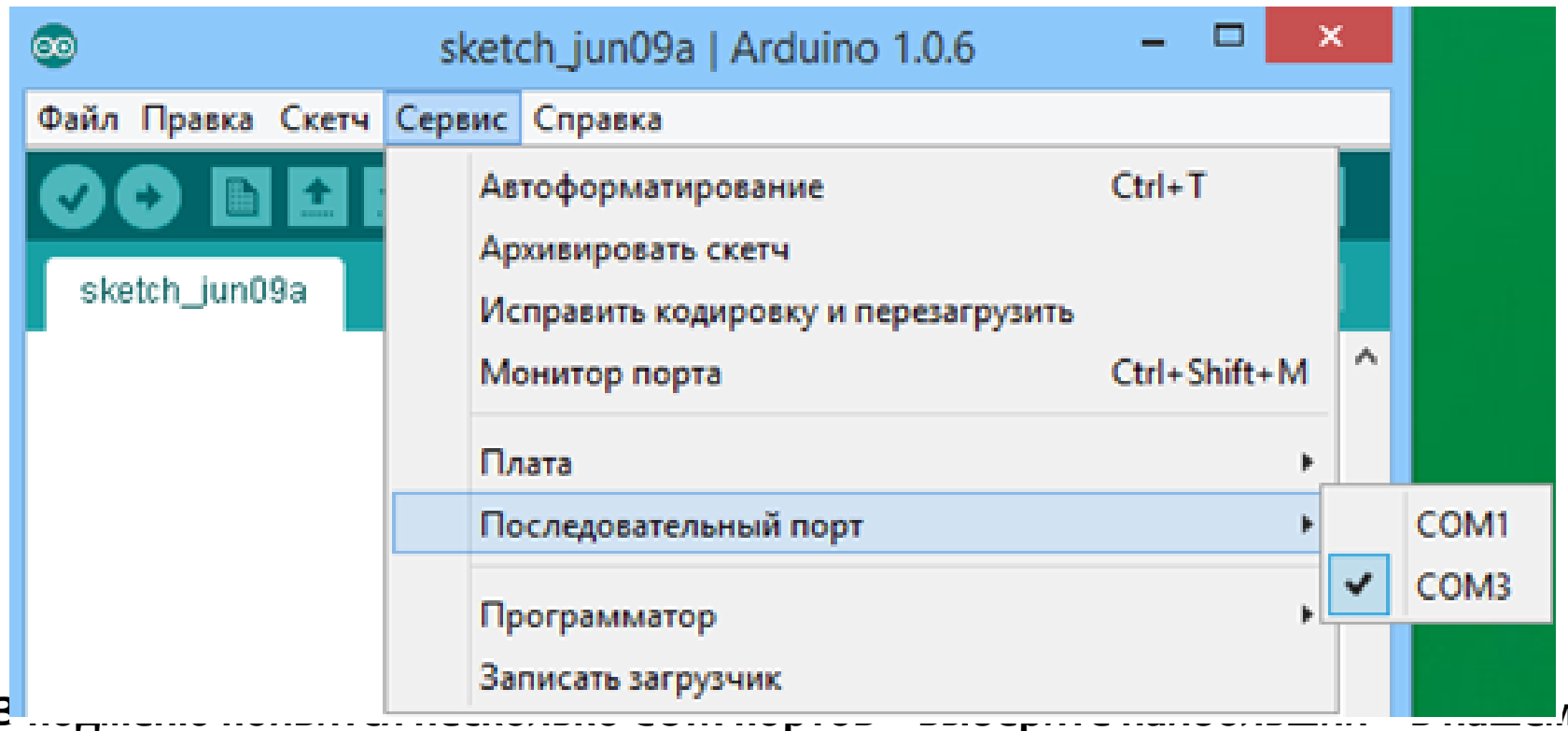
1. Наведите курсор на кнопку 1 – появится надпись – проверить.
2. Наведите курсор на кнопку 2 – появится надпись – загрузить.
3. Наведите курсор на кнопку 2 – появится надпись – создать.
4. Наведите курсор на кнопку 2 – появится надпись – открыть.
5. Наведите курсор на кнопку 2 – появится надпись – сохранить.
6. Наведите курсор на кнопку 2 – появится надпись – монитор порта.

Всеми этим кнопками мы будем пользоваться в процессе работы с «Arduino».

Продолжаем работать дальше. Программа «Arduino IDE» может работать с разными устройствами «Arduino». Установим в программу нашу плату «Arduino MEGA». Для этого войдем в меню «Сервис» и откроем подменю «Плата».



Вам уже известно, что «Arduino» будет работать с компьютером с помощью виртуального COM порта (реальный COM порт у современных компьютеров отсутствует). Для того чтобы подключить нашу «Arduino» к установленному COM порту в меню «Сервис» откроем подменю «Последовательный порт».

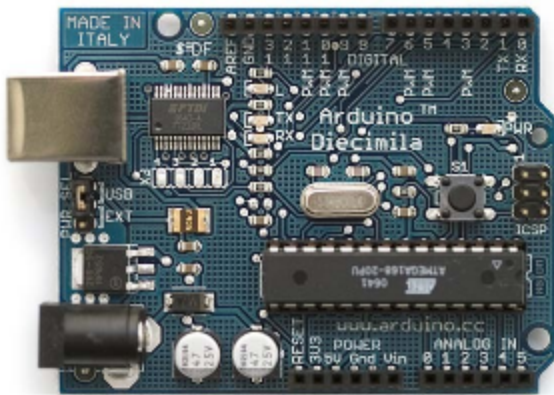


В случае COM 3.

Вот и прошла вся установка. Во время следующего сеанса работы с «Arduino» надо будет только щелкнуть по иконке и проверить соответствие платы и COM порта.

У слова "Arduino" 3 значения:

Физическое устройство



Среда разработки

Сообщество и философия



Язык Arduino

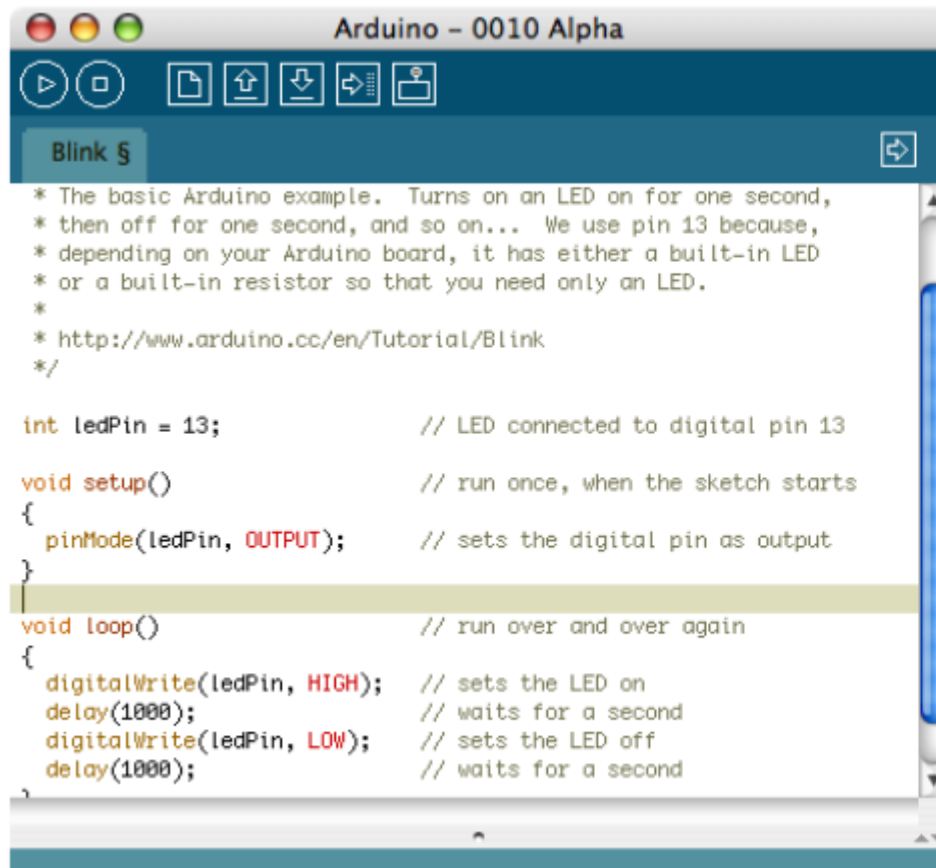
«*скетч*» - программа, написанная для запуска на Arduino

«*пин*» - вход или выход, подключённый к чему-либо. Пример: светодиод на выходе или кнопка на входе

«*цифровой*» - значение HIGH или LOW (как вкл/выкл или один/ноль)
Пример: состояние выключателя

«*аналоговый*» - значение обычно в промежутке 0-255. Пример: яркость светодиода, скорость мотора.

Среда программирования Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Arduino - 0010 Alpha". The menu bar includes icons for Run, Stop, New, Open, Save, Upload, and Download. The sketch name "Blink 5" is visible in the top left. The code is as follows:

```
* The basic Arduino example. Turns on an LED on for one second,  
* then off for one second, and so on... We use pin 13 because,  
* depending on your Arduino board, it has either a built-in LED  
* or a built-in resistor so that you need only an LED.  
*  
* http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
int ledPin = 13;           // LED connected to digital pin 13  
  
void setup()              // run once, when the sketch starts  
{  
  pinMode(ledPin, OUTPUT); // sets the digital pin as output  
}  
  
void loop()               // run over and over again  
{  
  digitalWrite(ledPin, HIGH); // sets the LED on  
  delay(1000);                // waits for a second  
  digitalWrite(ledPin, LOW);  // sets the LED off  
  delay(1000);                // waits for a second  
}
```

- Как текстовый редактор
- Можно смотреть/писать/редактировать скетчи
- Но затем Вы загружаете их в микроконтроллер

Познакомимся с «мини-шаблоном» для любого скетча.

```
int ... = ... ;
```

```
void setup()  
{  
  pinMode(... , OUTPUT);  
  pinMode(... , INPUT);  
}
```

```
void loop()  
{  
  .....  
}
```

1. Раздел описаний (может и не отсутствовать. Назначения «ножек» и описания переменных.

2. Раздел загрузки. Настройка портов ввода и вывода в фигурных скобках { } в функции «void setup()».

3. Бесконечный цикл. Написание последовательности команд в фигурных скобках { } в функции «void loop()».

Внимание !!! В дальнейшем, все Ваши скетчи (программы) будут состоять из этих трех блоков.

Не забудьте после каждой строки ставить точку с запятой.

Работа со светодионом.

Если разбить слово "светодиод" на составляющие, то мы получим "свето" и "диод".

Вот так светодиод выглядит в жизни :

светодиод

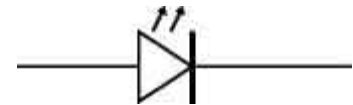
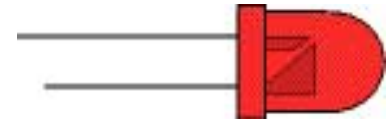
А так обозначается на схеме :

светодиод на схеме

Для чего служит светодиод?

Светодиоды излучают свет, когда через них проходит электрический ток.

Были изобретены в 70-е года прошлого века для смены электрических лампочек, которые часто перегорали и потребляли много энергии.



Уровни сигналов порта HIGH и LOW

При чтении или записи к цифровому порту применимо только два возможных значения – порт может быть установлен как HIGH (высокий уровень) или LOW (низкий уровень).

Это делается с помощью функции - `digitalWrite()` - она устанавливает на цифровом пине (контакте) высокий (5 вольт) или низкий (0 вольт) уровень напряжения.

`digitalWrite(13, HIGH);` // Уровень HIGH соответствует 5 вольтам на выходе порта.

`digitalWrite(13, LOW);` // Уровень LOW соответствует 0 вольтам на выходе порта.

Настройка цифровых портов на ввод (INPUT) и вывод (OUTPUT) сигналов

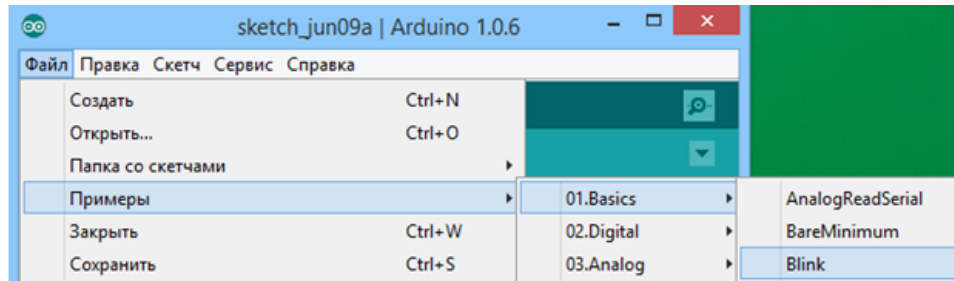
Цифровые порты могут использоваться на ввод или вывод сигналов. Изменение порта с ввода на вывод производится при помощи функции `pinMode()`.

Пример:

```
pinMode(13, OUTPUT); //13й вывод будет выходом  
pinMode(12, INPUT); //а 12й – входом
```

Первый скетч.

Первый скетч позволит управлять зажиганием и гашением светодиода. В загруженной среде «Arduino» есть довольно много примеров, поэтому первую программу мы возьмем готовую. Называется она «Blink» (мигать). Она создана для тестирования «Arduino». К пину (ножке) цифрового входа/выхода № 13 на плате «Arduino» подключен светодиод.



Для загрузки примера скетча для примера откроем файл «Blink». Для этого откроем путь к этому файлу : Файл > Примеры > 01.Basics > Blink . Наведем курсор на Blink, щелкаем левой кнопкой мышки и скетч будет загружен в среду программирования «Arduino IDE».

В этом скетче очень много комментариев. Их можно не писать. Программа их не выполняет. Мои комментарии написаны обычным текстом.

Раздел описаний – отсутствует.

Раздел загрузки.

```
void setup()  
{  
  pinMode(13, OUTPUT);  
}
```

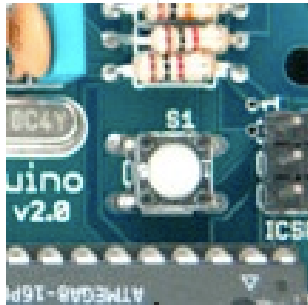
Раздел - бесконечный цикл.

```
void loop()  
{  
  digitalWrite(13, HIGH); // цифровому пину № 13 присваивается высокий  
                          // уровень  
  delay(1000);           // задержка 1000 мс  
  digitalWrite(13, LOW); // цифровому пину № 13 присваивается низкий  
                          // уровень  
  delay(1000);           // задержка 1000 мс  
}
```

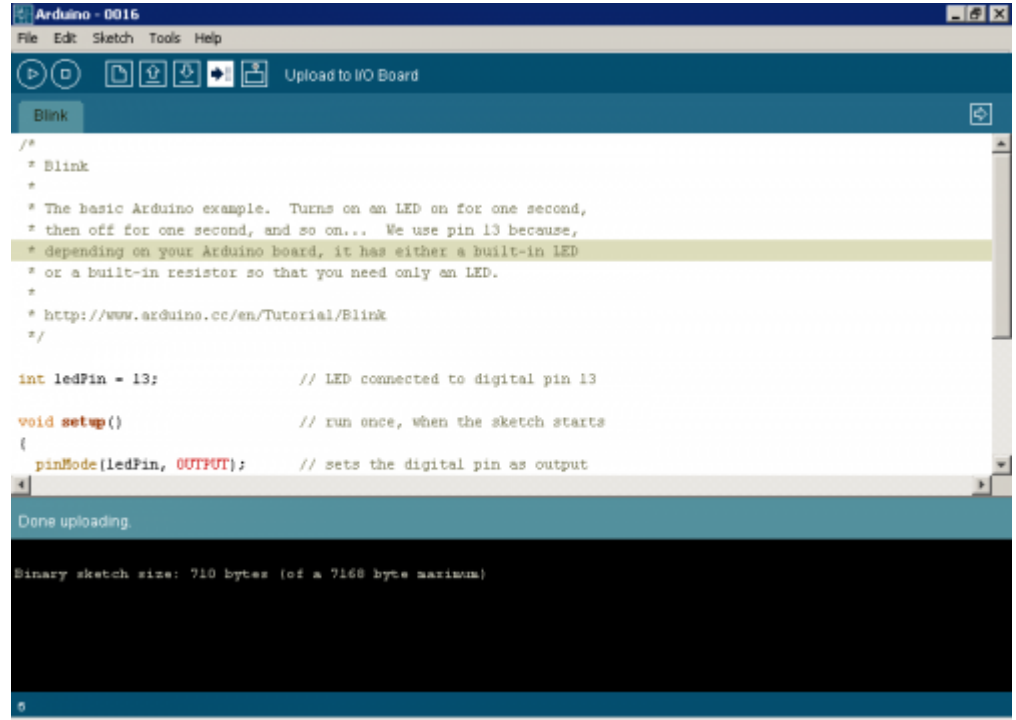
Скетч



Done compiling.



Done compiling.



Компилируем и загружаем скетч в МК.

Вмонтированный Светодиод мигает.

Скетч для светодиода с выводом на

```
int ledPin = 13;
```

Serial Monitor

```
void setup()
```

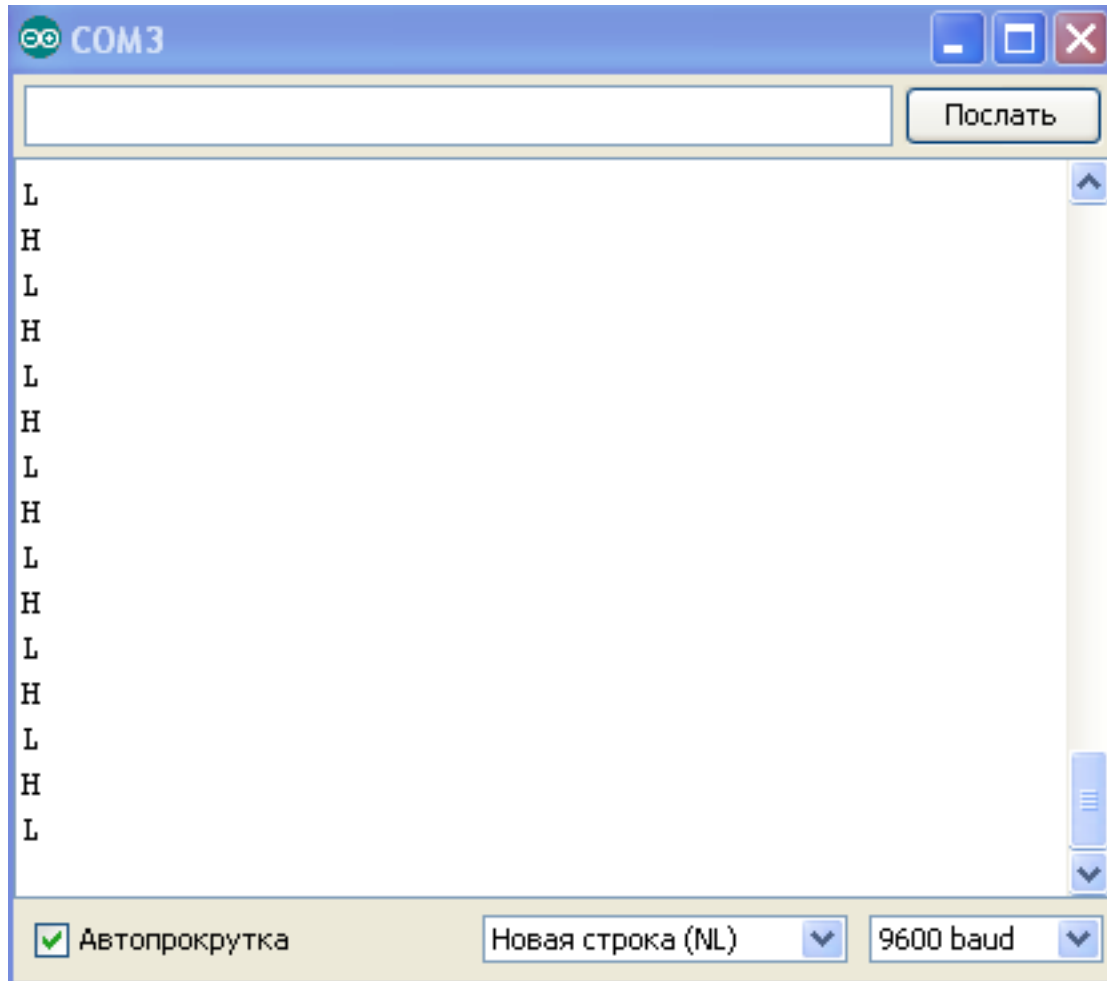
```
{  
  pinMode(ledPin, OUTPUT);  
  Serial.begin(9600); // инициализация работы с COM-  
портом  
}
```

```
void loop() {
```

```
  digitalWrite(ledPin, HIGH);  
  Serial.print("H"); // светодиод горит – пишем H  
  delay(1000);  
  digitalWrite(ledPin, LOW);  
  Serial.println("L"); // светодиод погасил – пишем L  
  delay(1000);  
}
```

Вывод на Serial Monitor

Запускаем скетч и нажимаем кнопку **6** - монитор порта.

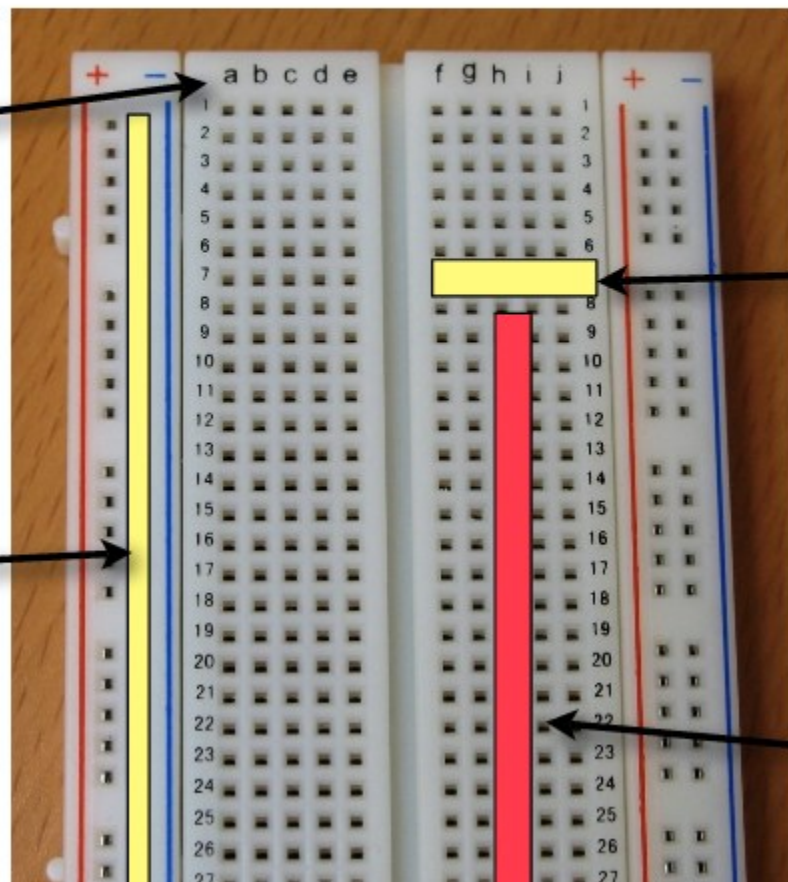


Работа с внешними цифровыми устройствами.

Светодиоды и резисторы



Беспаячные макетные платы

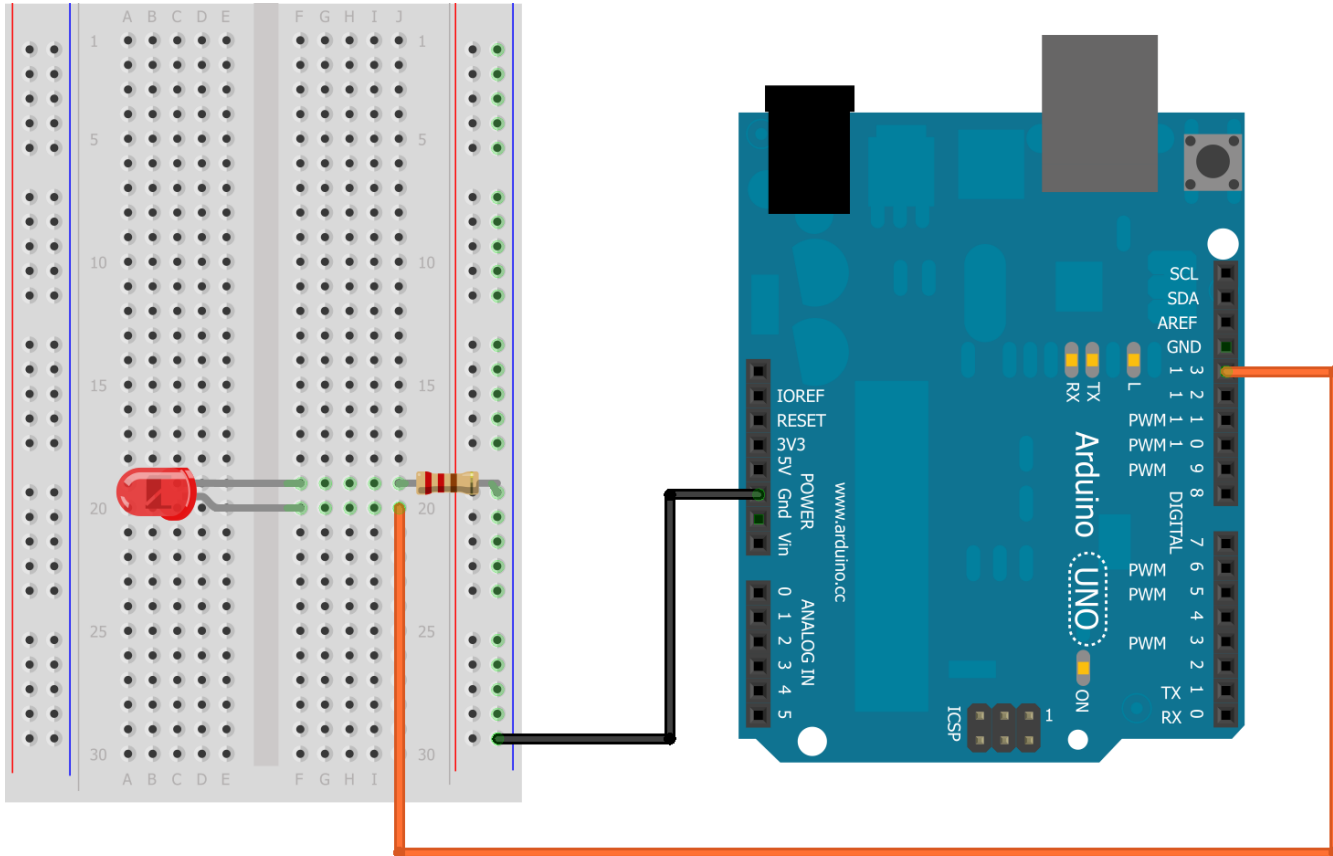
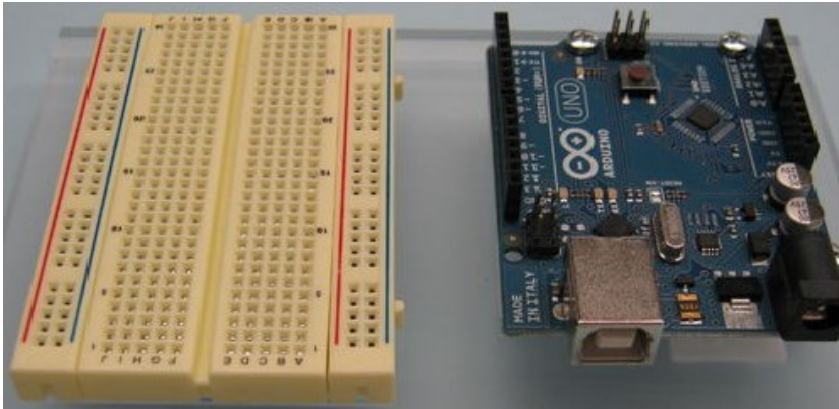


цифры и
буквы — как
координаты

соединены
по 5 вместе

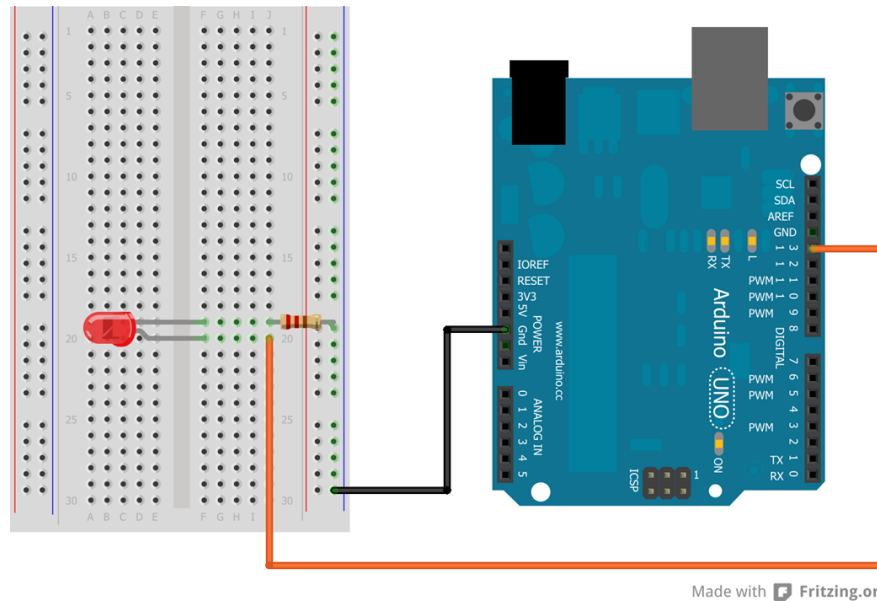
Все соединены
«общая шина»

не
соединены



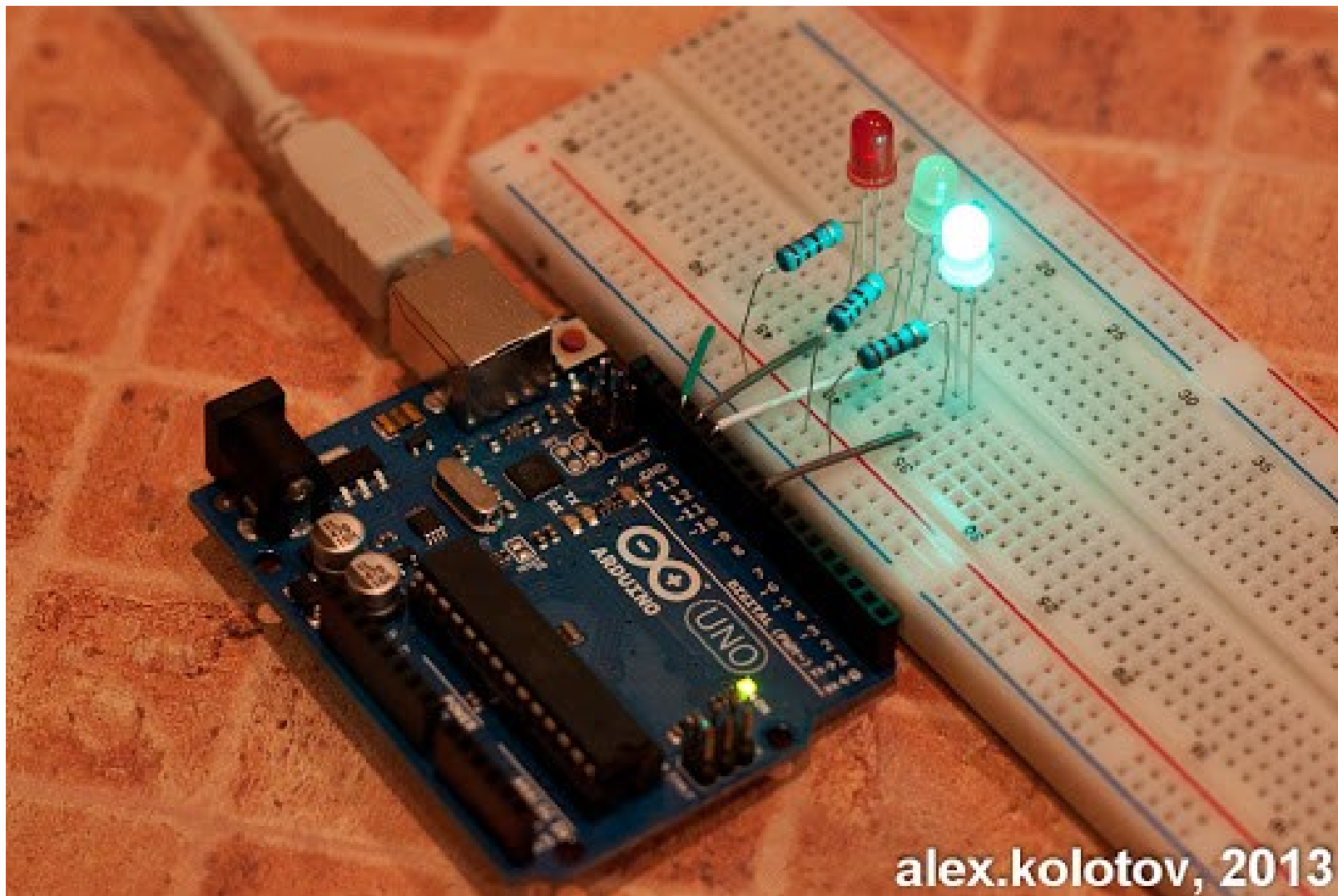
Практическая работа 1.

Мигание светодиода.



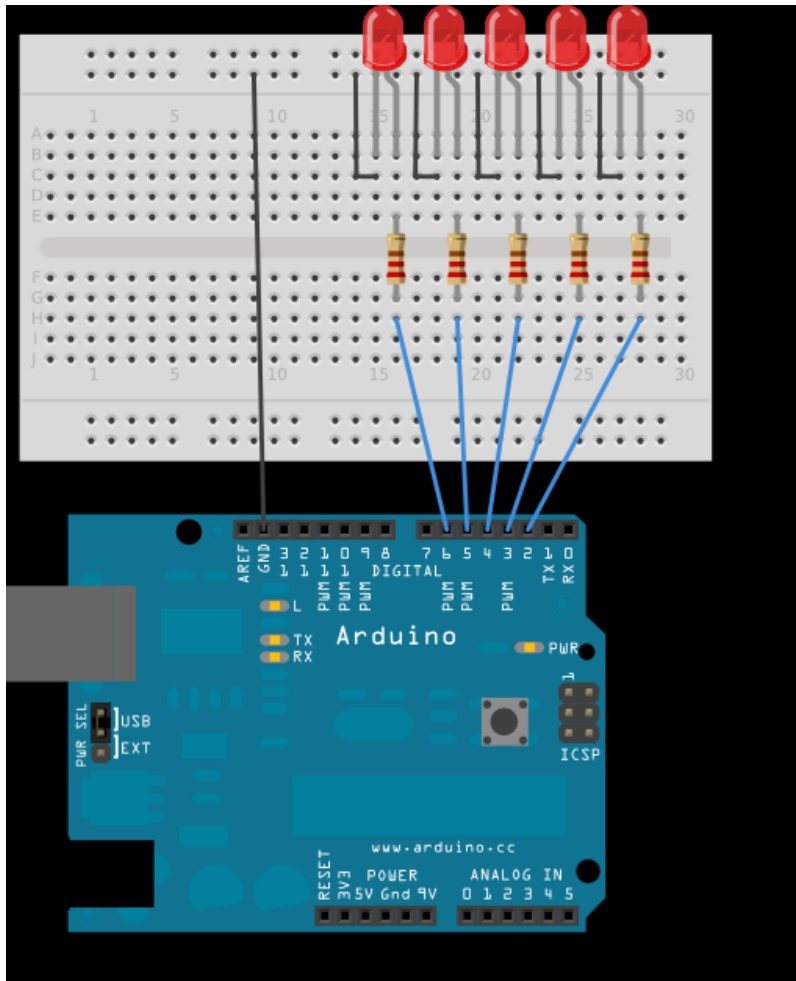
1. Оборудование : микроЭВМ «Arduino», плата прототипирования, светодиод, Резистор 300 – 720 Ом, соединительные провода.
2. Собрать схему. Компилировать и загрузить скетч. Светодиод начнет мигать. Измените время горения и время и время не горения светодиода (изменять Delay).
3. Подключите светодиод к другому пину, например 3 – 12, и изменив скетч, сделайте то же самое.

Практическая работа 2. Светофор на Ардуино.



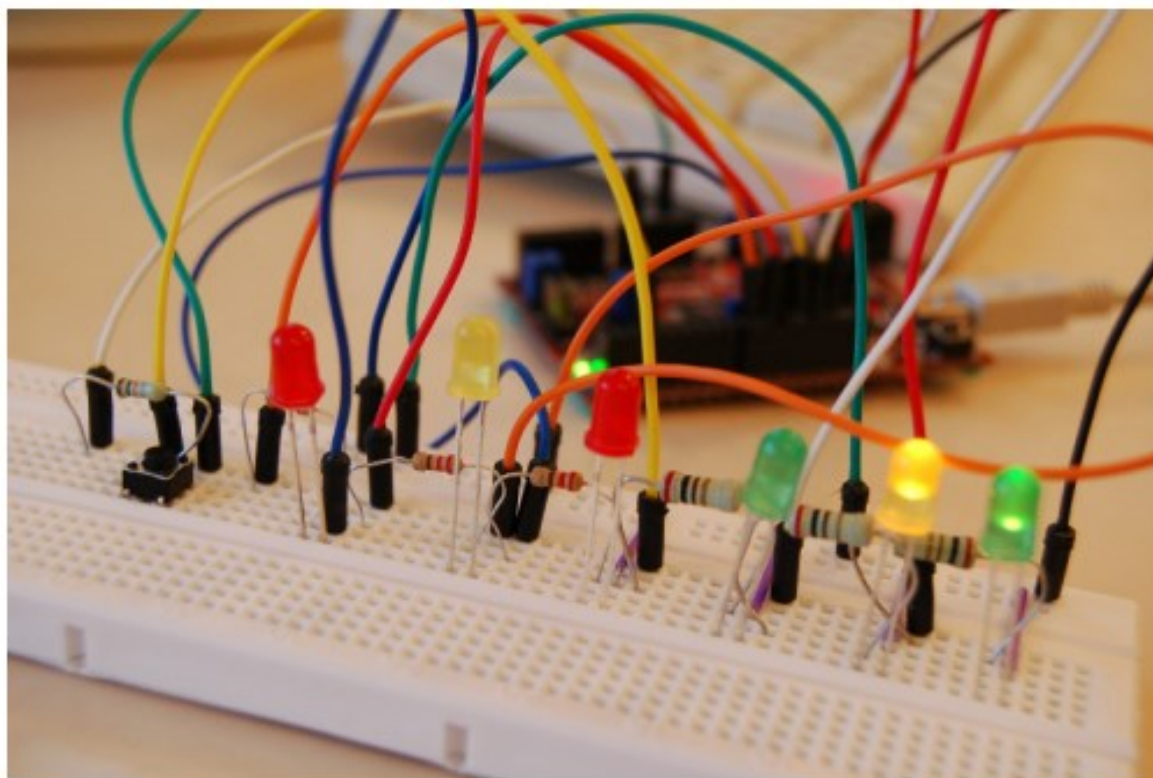
Практическая работа 3.

Бегущие огни.



1. Оборудование : микроЭВМ «Arduino», плата прототипирования, светодиоды - 5, Резисторы 300 – 720 Ом - 5 шт., соединительные провода.
2. Собрать схему. Написать скетч – светодиоды последовательно зажигаются и гаснут. Компилировать и загрузить скетч. Светодиоды начнут зажигаться и гаснуть согласно скетчу..
3. Измените скетч, чтобы получился бегущий огонь слева направо и справа налево.
4. Измените скетч, чтобы огни доходили до середины и расходились к краям.

Практическая работа № 4 Азбука Морзе на Ардуино.



Бегущие огни Жени

При отпущенной кнопке огонёк бежит в одну сторону, при нажатой — в другую.

Практическая работа № 5

Азбука Морзе на Ардуино.

А • —	Л • — • •	Ц — • — •
Б — • • •	М — —	Ч — — — •
В • — —	Н — •	Ш — — — —
Г — — •	О — — —	Щ — — • —
Д — • •	П • — — •	Ъ • — — • — •
Е •	Р • — •	Ы — • — —
Ж • • • —	С • • •	Ь — • • —
З — — • •	Т —	Э • • — • •
И • •	У • • —	Ю • • — —
Й • — — —	Ф • • — •	Я • — • —
К — • —	Х • • • •	

Внешние цифровые устройства. Работа на выход.

RGB-светодиоды

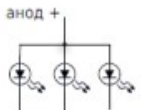
Обычный светодиод



анод +
катод -



RGB-светодиод

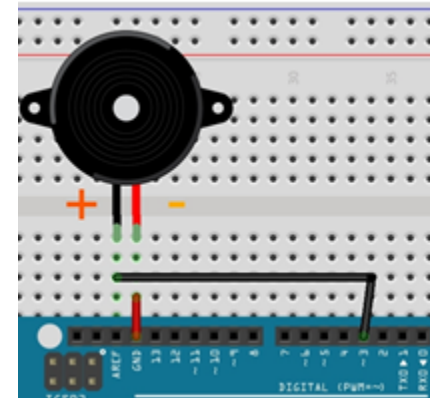


красный синий зелёный

красный катод -
анод +
синий катод -
зелёный катод -

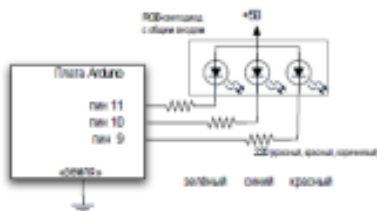


в действительности - 3 светодиода в одном корпусе



Смешивание цветов

Всего 3 светодиодами можно сделать любой* цвет



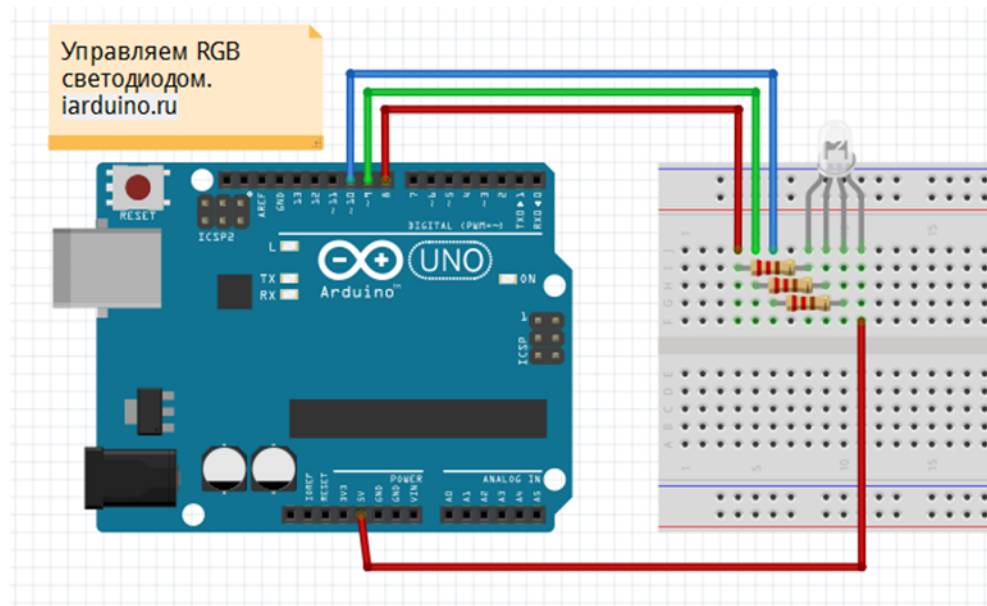
С RGB можно сделать любой цвет (кроме чёрного)

Смешение цветов — аддитивная цветовая модель (в печати используется субтрактивная, в ней результатом смешения является тёмно-коричневый)

**Пьезоизлучатель.
Электродвигатели.
Электромагниты.
Электромагнитные
реле.**

Практическая работа 5.

Свечение RGB светодиода.



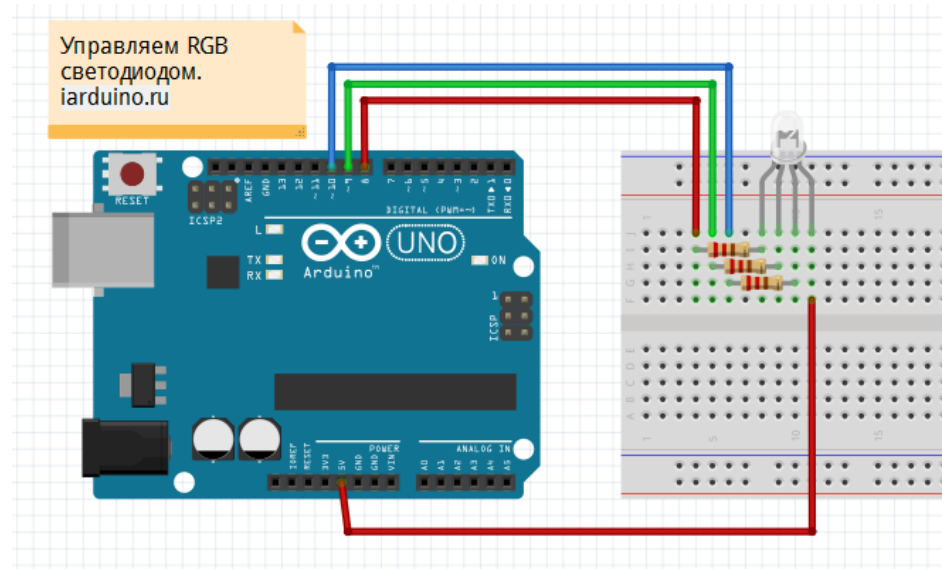
1. Оборудование : микроЭВМ «Arduino», плата прототипирования, светодиод RGB, резисторы 300 – 720 Ом, соединительные провода.
2. Собрать схему. Написать скетч. Компилировать и загрузить скетч. Светодиод начнет светиться.,
3. Измените время задержки в скетче для каждого светодиода, получите палитру различных цветов..

Три светодиода цветомузыка.

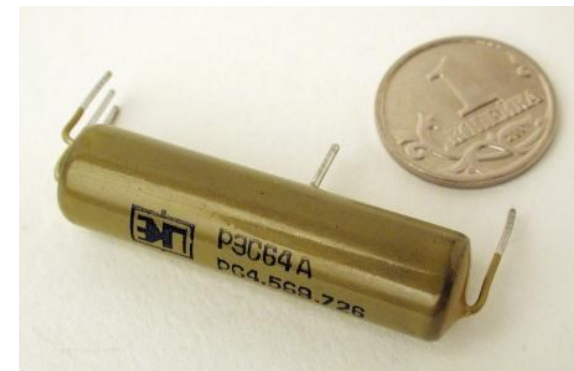
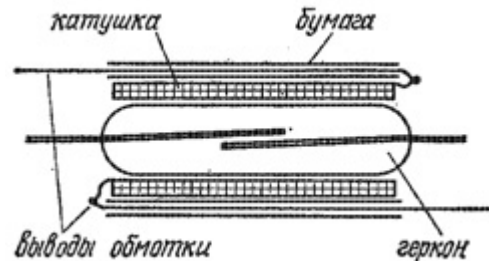
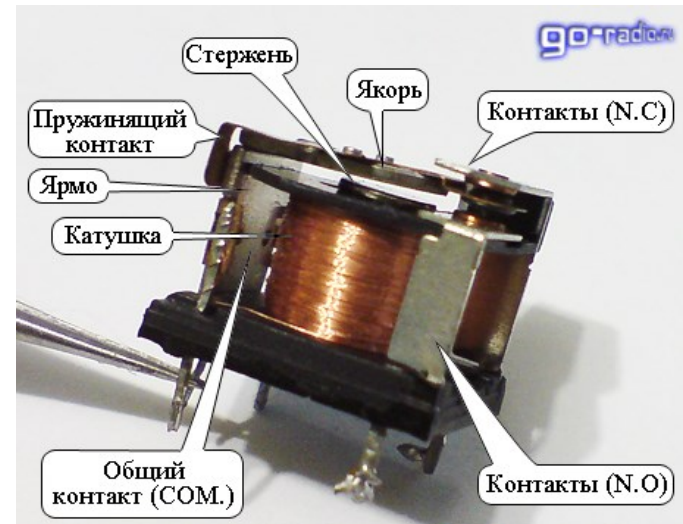
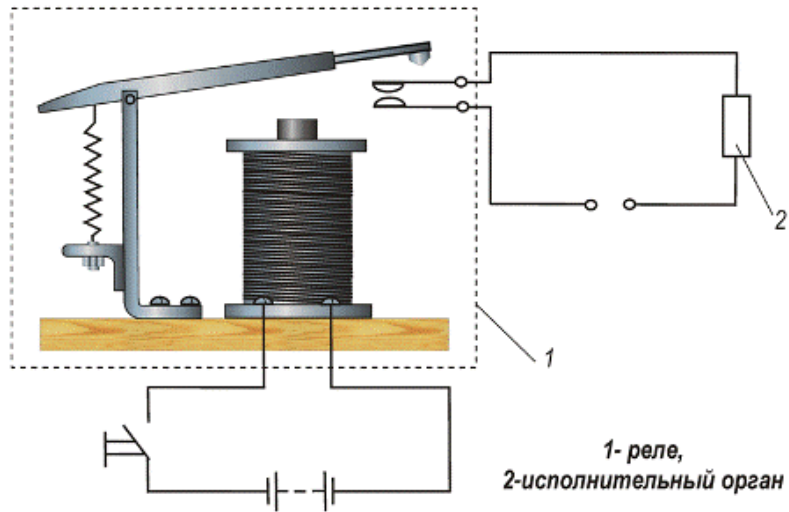
Добавим еще пару разноцветных светодиодов.

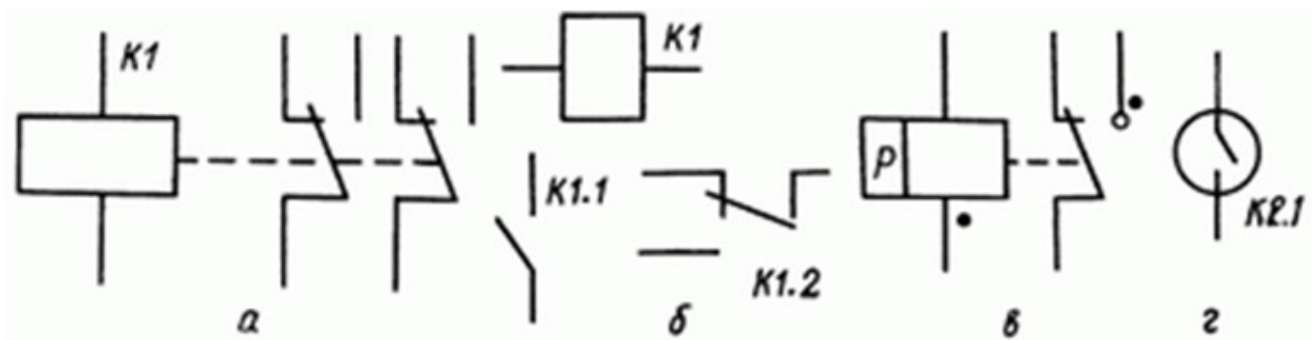
У нас будут три светодиода: красный, синий и зеленый. Схема будет выглядеть так.

Запустите пример и убедитесь, что все работает правильно. Для улучшения результата склейте бумажный кубик и наденьте его на светодиоды. У вас получится полупрозрачная конструкция, которая будет напоминать цветомузыку.

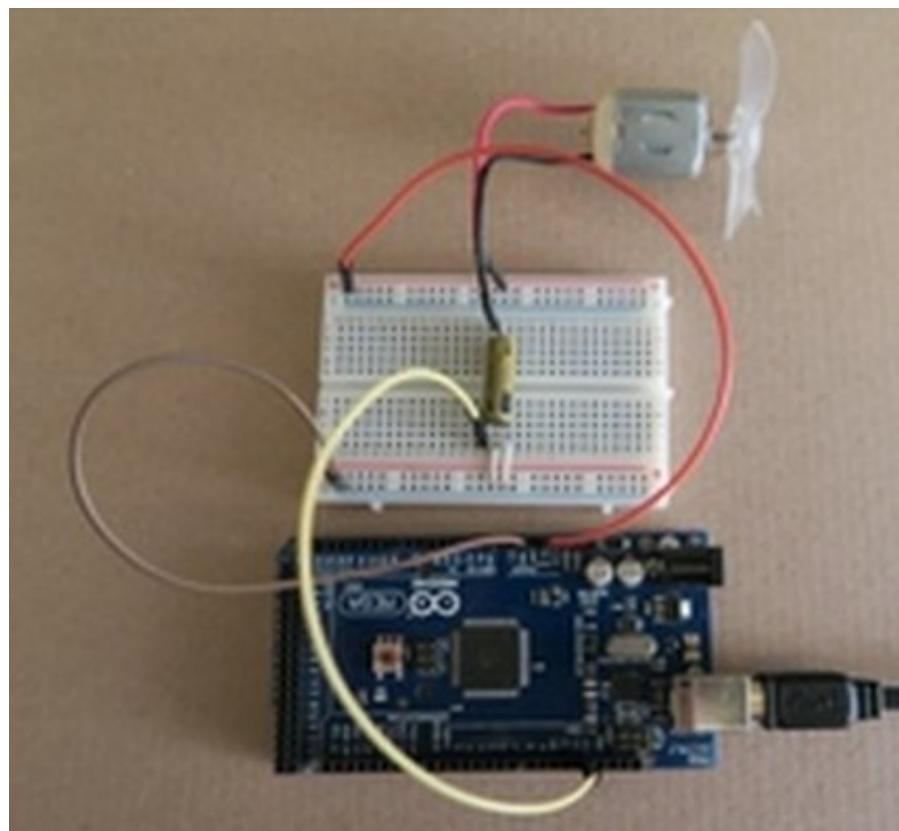
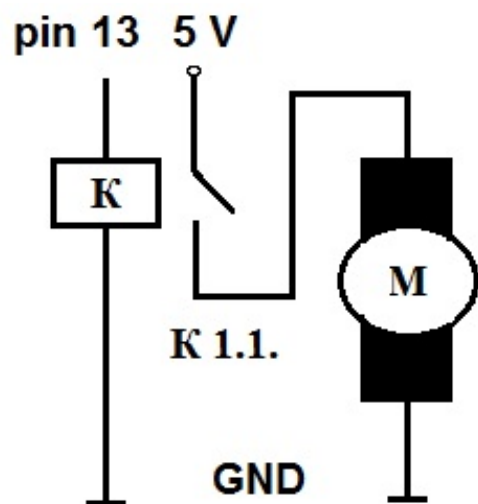


Подключение мощных нагрузок. Электромагнитное реле.

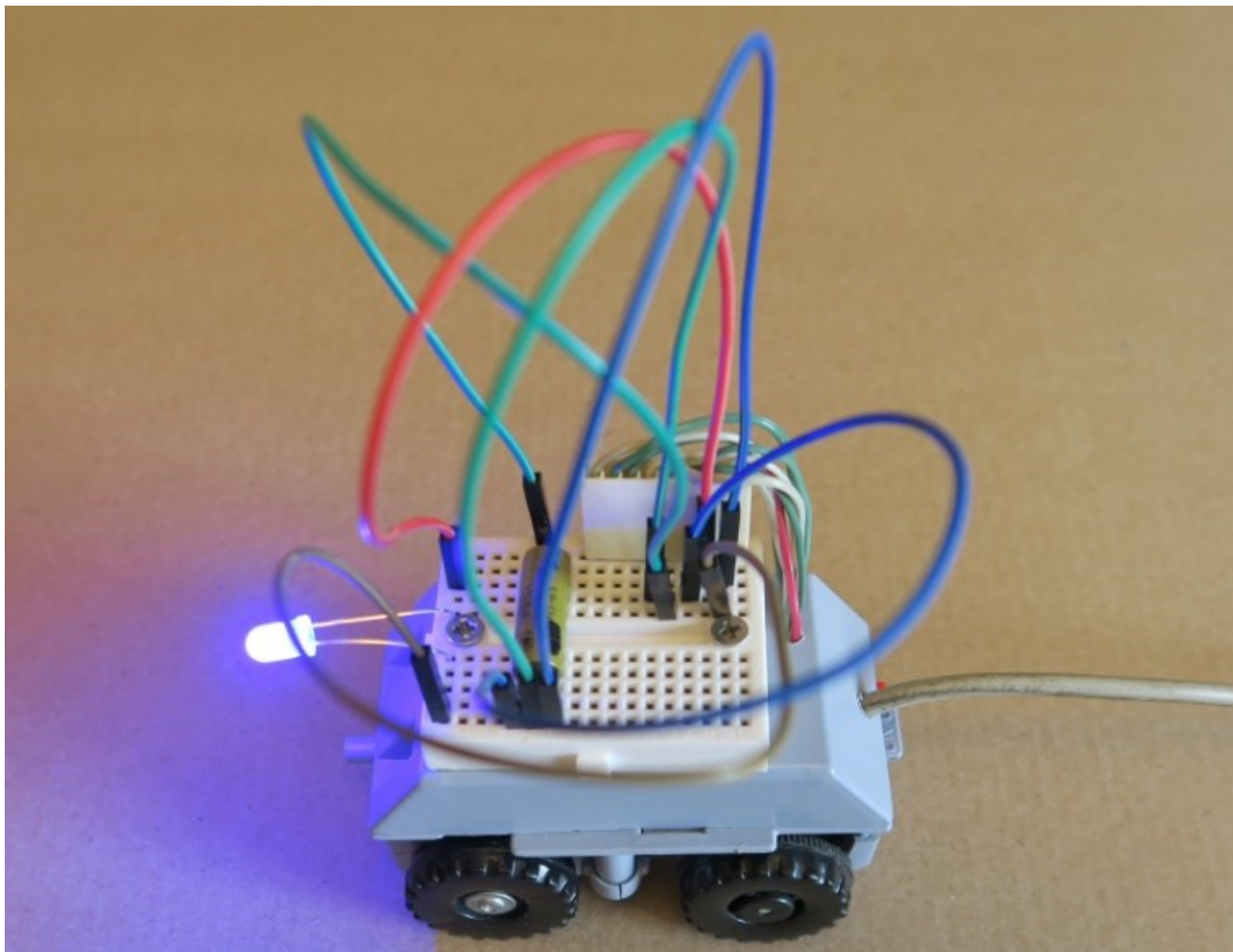




Условные графические обозначения электромагнитных реле



МАШИНА + ГЕРКОНОВОЕ РЕЛЕ + СВЕТОДИОД



Подключение мощных нагрузок. Усилитель на транзисторе.

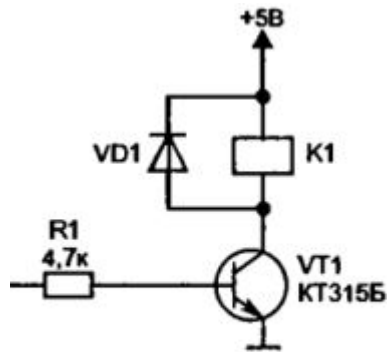
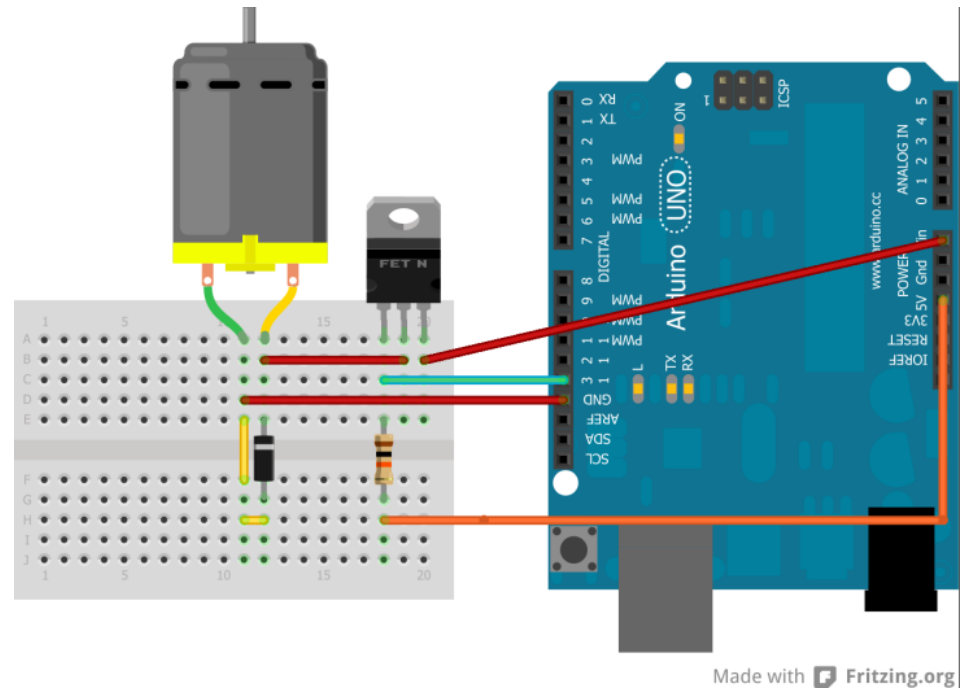
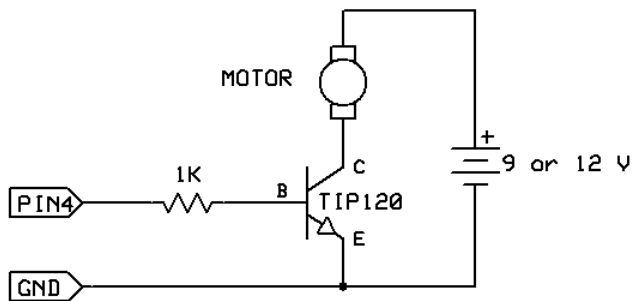


Рисунок 1



Внешние цифровые устройства.

Работа на вход - кнопка.

Практическая работа № 6.

Цифровой ввод — кнопка

Соберем простую схему:

как видно – это дополненная
схема Blink-а :)

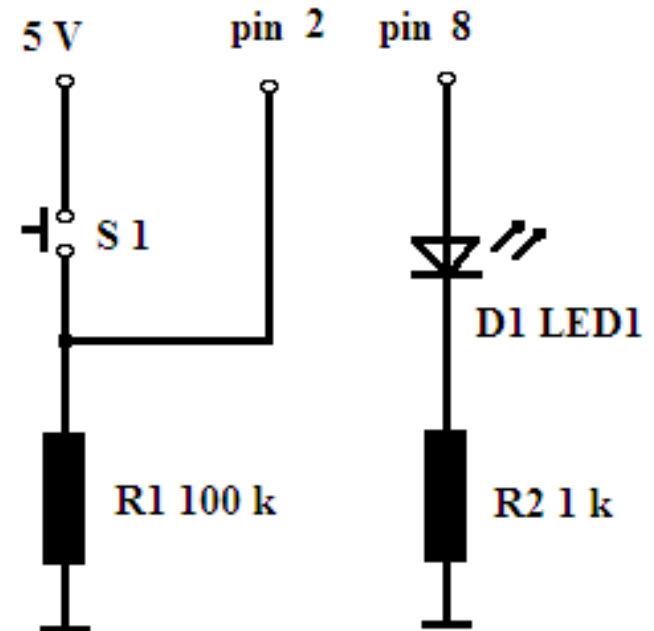
Что же происходит на 2-м цифровом
порту?

Если к порту ничего не подключено –
его состояние меняется между 0V и 5V
(LOW & HIGH).

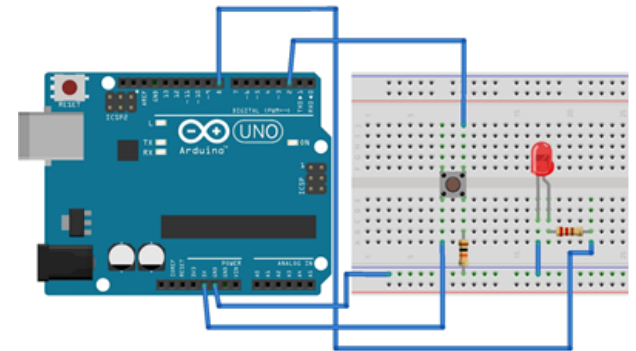
Подключая резистор между портом и землёй
— мы тем самым «прижимаем» порт к 0 (LOW),

а при нажатии кнопки происходит соединение с напряжением питания и тем
самым порт «подтягивается» к 5V (HIGH).

Тем самым мы узнаём – нажата кнопка или нет – просто считывая значение на
входе 12 цифрового порта и проверяя – HIGH это или LOW.

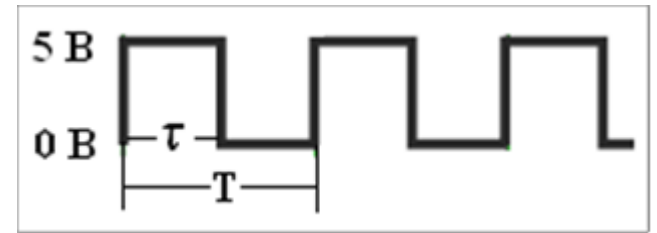


```
int button = 2;           // Кнопку
                          //подключаем к pin 2
int led = 6;              // Светодиод подключаем
к pin 6
void setup() {
pinMode(led, OUTPUT);    // pin 6 настраиваем на
выход
pinMode(button, INPUT);  // pin 2 настраиваем на
вход
}
void loop(){
if (digitalRead(button) == HIGH) // Если кнопка
нажата
{
digitalWrite(led, HIGH);      // то светодиод
горит
}
else {
digitalWrite(led, LOW);       // иначе светодиод
не горит
}
}
```



Управление напряжением на цифровых выходах «Arduino» ШИМ (PWM)

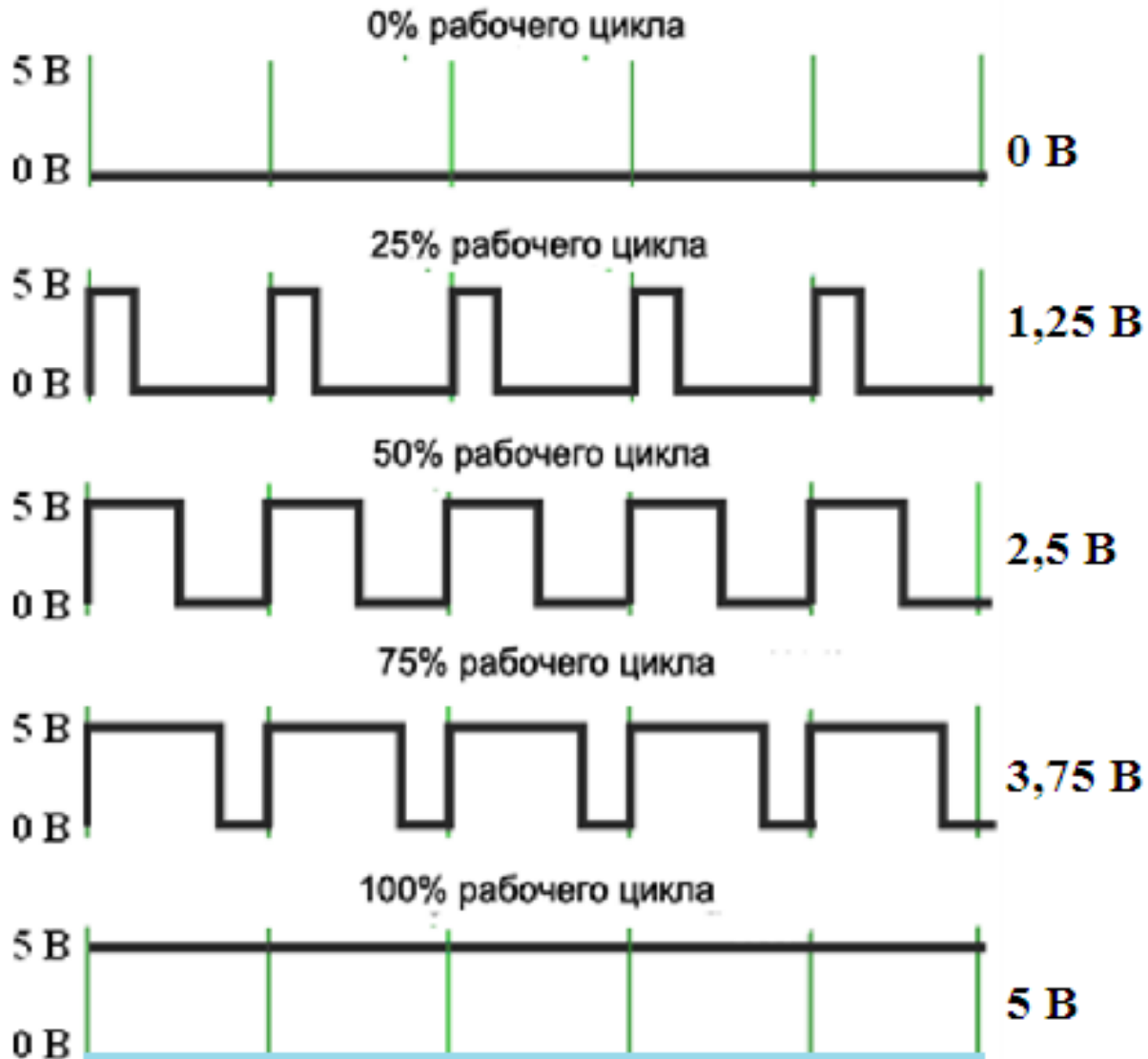
Вам уже известно, что на цифровых выходах «Arduino» могут быть два уровня напряжения 0В и 5В. Но разработчики «Arduino» предусмотрели и такой вариант: если на цифровой выход подать прямоугольные импульсы с переменной скважностью, то тогда эффективное напряжение на цифровом выходе будет изменяться. Давайте выясним, что такое скважность. Рассмотрим симметричный прямоугольный импульс (рис.4). Скважностью называют безразмерную физическую величину, характеризующую любую импульсную систему.



Она обозначается буквой S и определяется отношением периода импульсных сигналов T к их длительности. $S = T/\tau$. Если длина импульса равна половине периода, то скважность равна двум, такой сигнал называется меандром. Длительность такого импульса и паузы между импульсами в периоде равны. Скважность может выражаться в процентах и является важной характеристикой импульсной системы. У нас она выражается в вольтах.

Так вот, если в «Arduino» программно изменять скважность, то на цифровых выходах (3, 5, 6, 9, 10, 11) можно изменять напряжение от 0 В до 5 В. На научном языке это называется ШИМ (англ. PWM) Широтно-Импульсная Модуляция.

Широтно-Импульсная модуляция



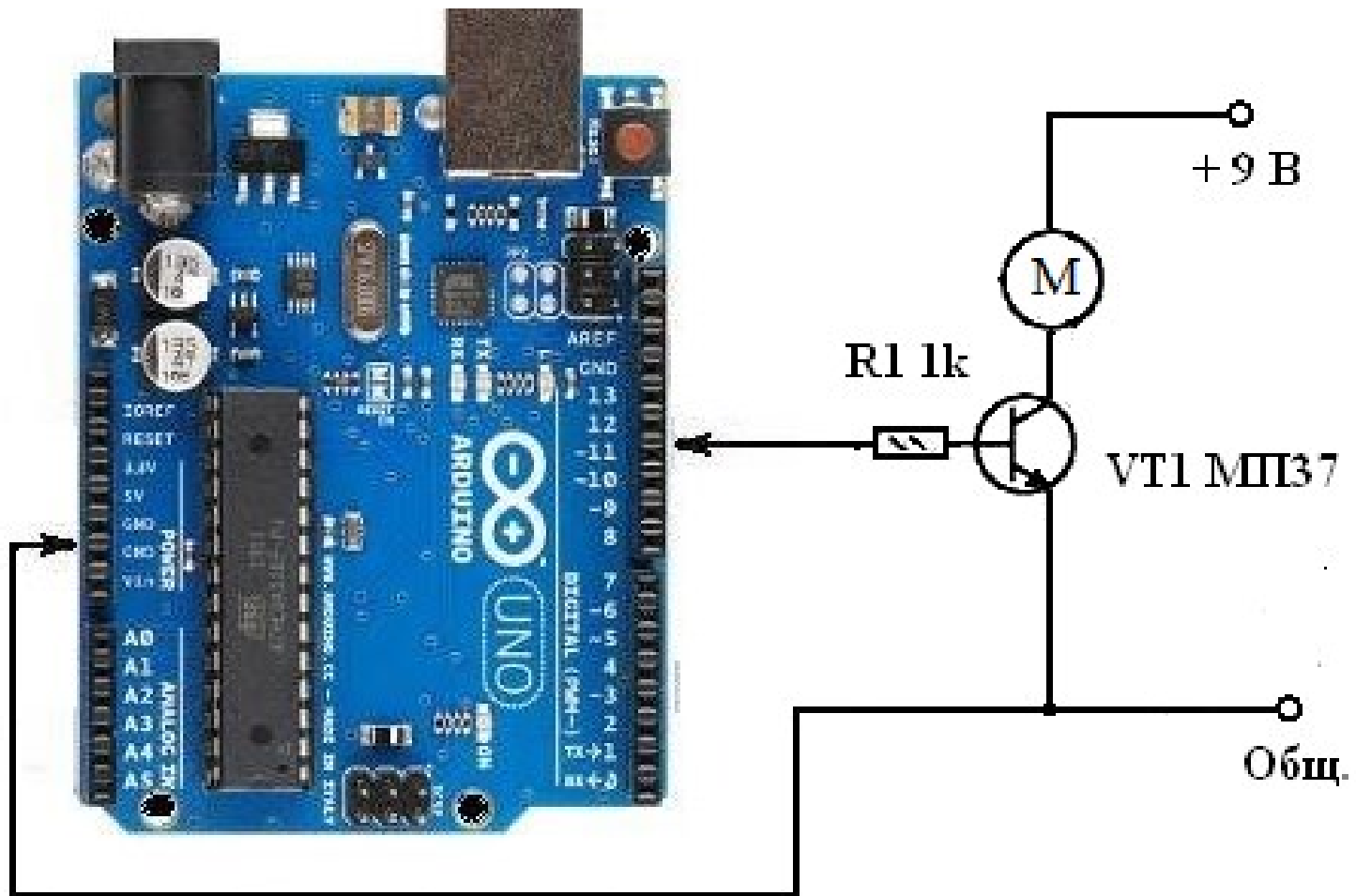
Скважность может выражаться в процентах и является важной характеристикой импульсной системы. У нас она выражается в вольтах.

Так вот, если в «Arduino» программно изменять скважность, то на цифровых выходах (3, 5, 6, 9, 10, 11) можно изменять напряжение от 0 В до 5 В. Контакты 3, 5, 6, 9, 10, 11 имеют двойное назначение: вход/выход и ШИМ (PWM).

Скетч для ШИМ.

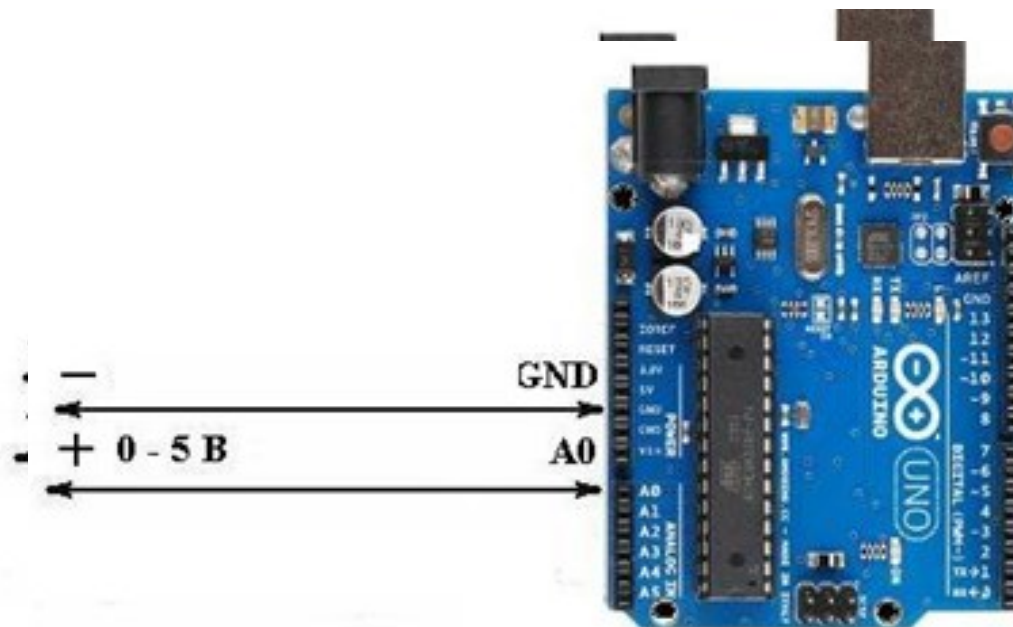
```
int value = 0;           //здесь храним значение яркости
int ledpin = 9;         //светодиод подключим к 9 контакту
void setup()
{
  pinMode(ledpin, OUTPUT);    //устанавливаем 9 контакт в режим вывода
}
void loop()
{
  for(value = 0 ; value <= 255; value+=5) //плавно зажигаем светодиод
  {
    analogWrite(ledpin, value);    //выставляем значение
    delay(30);                    //небольшая пауза
  }
  for(value = 255; value >=0; value -=5) //плавно гасим
  {
    analogWrite(ledpin, value);
    delay(30);
  }
}
```

Подключение электродвигателя к выходу ШИМ



Внешние аналоговые устройства. Работа только на вход.

Мы уже говорили о том, что на аналоговый вход можно подавать напряжение от 0 до 5 В. Таким образом, при десяти разрядном АЦП, мы имеем практически готовый вольтметр для измерения напряжения от 0 до 5 В с точностью 4,9 мВ. А работа с потенциометром только подтвердила это. Схема такого вольтметра достаточно проста. Необходимо подключить к аналоговому входу A0 (или A1 – A5) и к GND (земле) напряжение от 0 до 5 В.

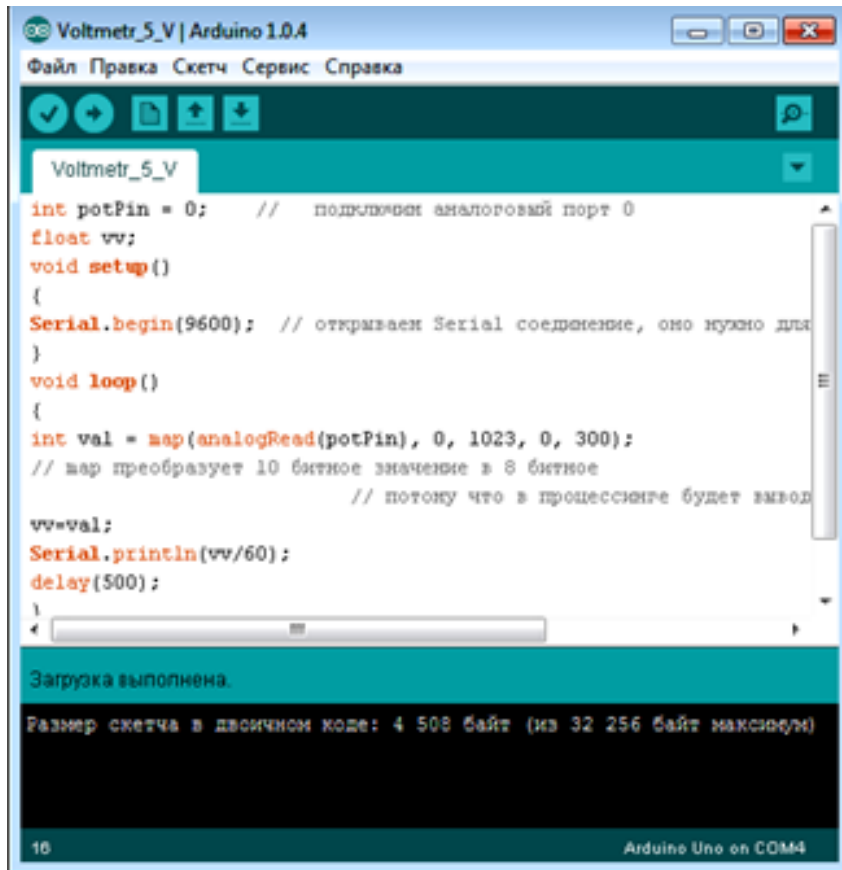


Скетч Voltmetr 5 V;

```
int potPin = 0; // подключим аналоговый порт 0
float vv;      // новая строка вводим новую вещественную переменную vv
void setup()
{
  Serial.begin(9600); // открываем Serial соединение, оно нужно для приема и
  передачи
}
void loop()
{
  int val = map(analogRead(potPin), 0, 1023, 0, 300);
                // map преобразует 10 битное значение в 8 битное потому что в
будущем
                // в языках программирования будет выводиться яркость цвета
для
                // некоторых величин от 0 до 255
  vv=val;      // преобразуем целочисленную переменную val в вещественную
vv

  Serial.println(vv/60); // чтобы вывести напряжение в пределах 0 - 5 В делим на 60
  delay(50);
}
```

Работа вольтметра.

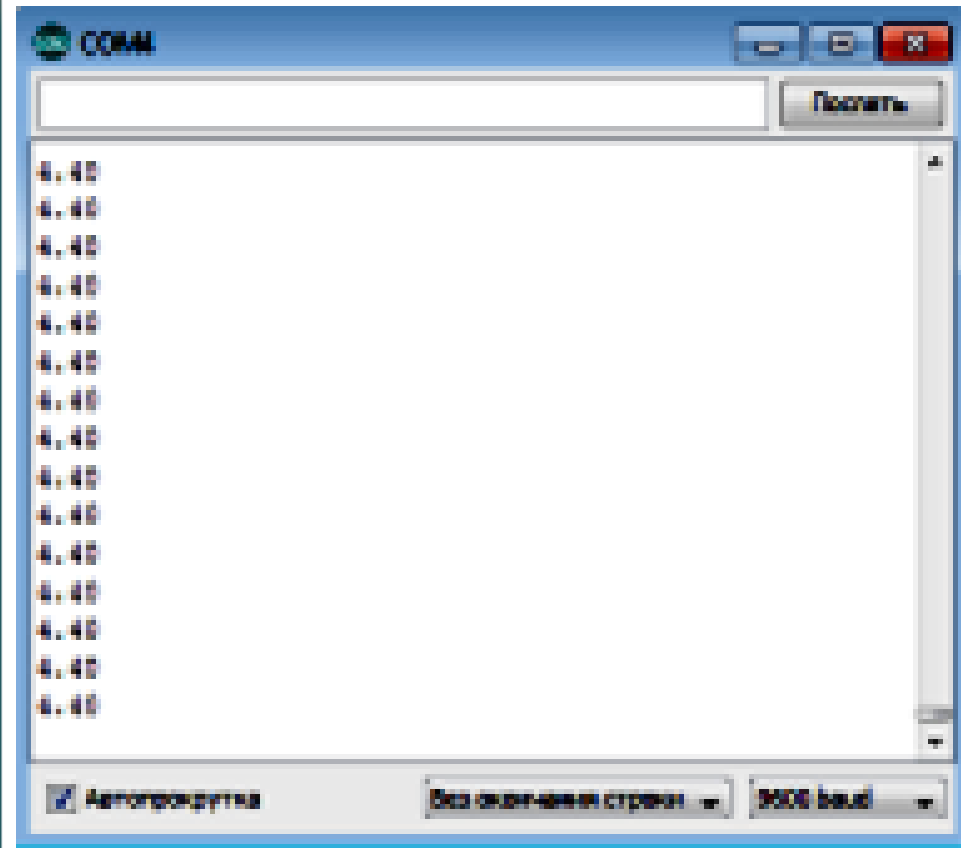


The screenshot shows the Arduino IDE interface with a sketch named "Voltmetr_5_V". The code is as follows:

```
int potPin = 0; // подключает аналоговый порт 0
float vv;
void setup()
{
  Serial.begin(9600); // открывает Serial соединение, оно нужно для
}
void loop()
{
  int val = map(analogRead(potPin), 0, 1023, 0, 300);
  // map преобразует 10 битное значение в 8 битное
  // потому что в процессинге будет вывод
  vv=val;
  Serial.println(vv/60);
  delay(500);
}
```

Below the code, the IDE shows the status "Загрузка выполнена." (Upload completed) and the sketch size: "Размер скетча в двоичном коде: 4 508 байт (из 32 256 байт максимум)". The status bar at the bottom indicates "Arduino Uno on COM4".

Скетч.



The screenshot shows the Serial Monitor window for COM4. The "Послать" (Send) button is visible. The monitor displays a continuous stream of the value "0.40". At the bottom, the settings are: "Автоскроллинг" (checked), "Вся оконечная строка" (All line endings), and "3008 baud".

Монитор Порты.

Вывод информации на экран компьютера в программе PascalABC



Программа на языке PascalABC

```
program ComRead;
uses System.IO.Ports,GraphABC;
var sp : SerialPort;
vv : integer;
begin
vv:=1;
  sp := new SerialPort('COM4');
  sp.Open();
  SetWindowSize(1280,1100);
  SetBrushColor(RGB(0,0,250));
  Rectangle(0,0,1280,1000);
```

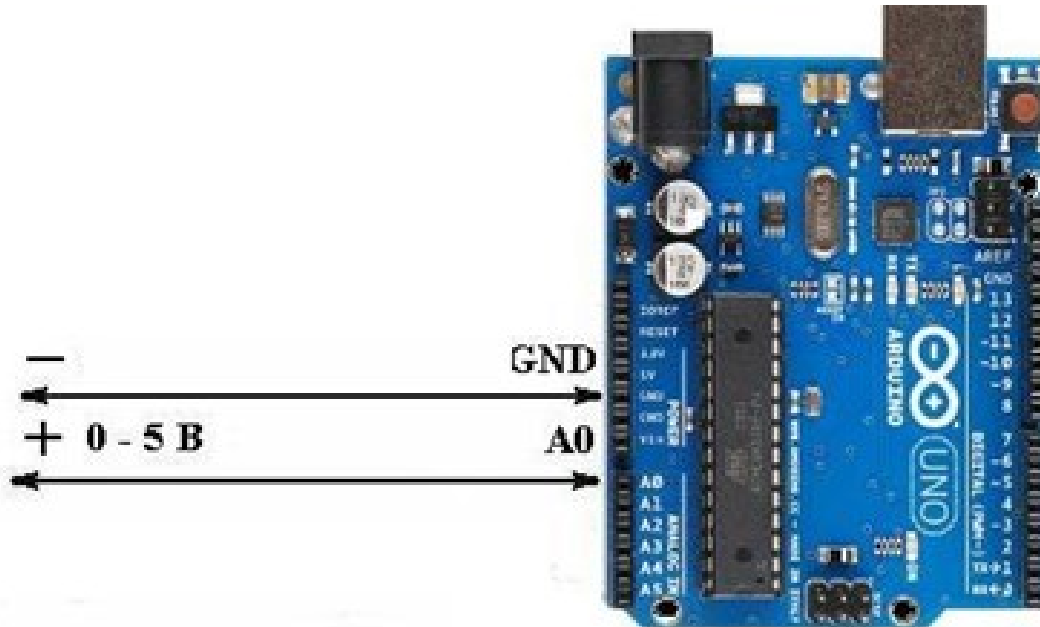
```
repeat
  begin
  LockDrawing;
  SetFontName('Times');
  SetFontStyle(fsBold);
  SetFontSize(300);
  SetFontColor(RGB(255,255,255));
  textOut(250,400,sp.ReadLine());
  textOut(755,400,' B ');
  SetFontSize(90);
  textOut(95,60,'Н А П Р Я Ж Е Н И Е');
  SetFontSize(50);
  textOut(500,250,'( 0 - 30 В )');
  Redraw;
  end;
until vv>1;
sp.Close();
end.
```

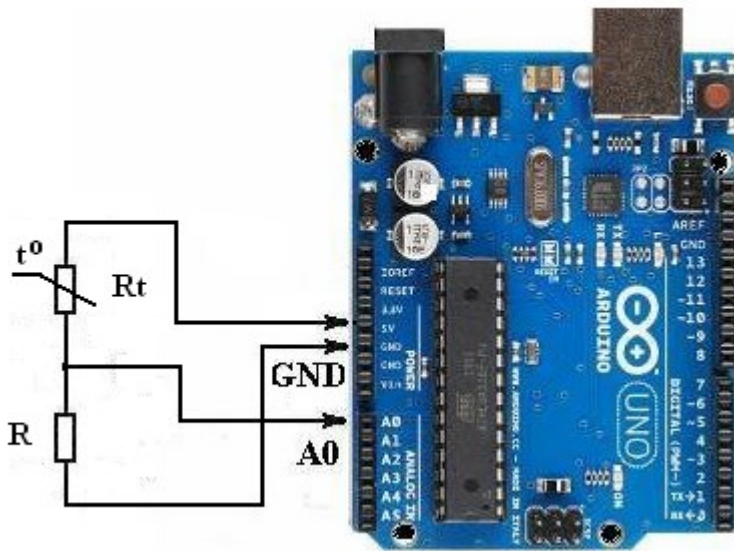
Практическая работа № 1.

Вольтметр : 0 – 5 В.

Оборудование :

1. Оборудование : микроЭВМ «Arduino», провод USB, компьютер с установленным скетчем.
2. Собрать схему. Компилировать и загрузить скетч.
3. Включить Serial Monitor.
4. Подключив к входу A0 измерить напряжение источника тока.
5. Проверить точность измерений с помощью вольтметра.





```
int potPin = 0; // подключим аналоговый
порт 0
float vv;      // вводим вещественную
               // переменную vv

void setup()
{
  Serial.begin(9600);}
void loop()
{
  int val = map(analogRead(potPin), 0, 1023, 0,
  500);

  vv=(( 0.018281606*val+0.258168026)*15);
  // преобразуем целочисленную
  // переменную
  // val в вещественную vv ;

  Serial.println(vv);

  delay(500);
}
```

ТЕМПЕРАТУРА

(0 - 100 град. С)

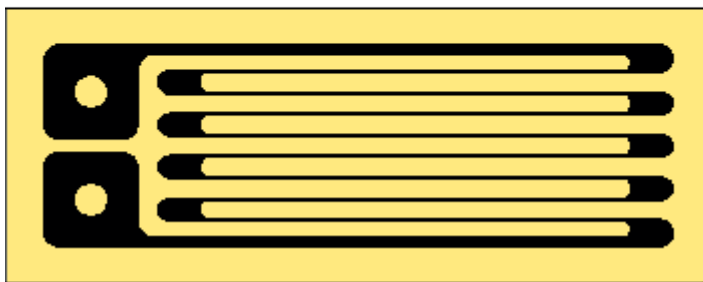
21.15 град. С

```
uses System.IO.Ports,GraphABC;
var sp : SerialPort;
vv : integer;
begin
vv:=1;
  sp := new SerialPort('COM9');
  sp.Open();
  SetWindowSize(1280,1100);      SetBrushColor(RGB(0,0,250));
  Rectangle(0,0,1280,1000);
  repeat
  begin
  LockDrawing;          SetFontName('Times');
  SetFontStyle(fsBold); SetFontSize(200);
  SetFontColor(RGB(255,255,255));
  textOut(150,400,sp.ReadLine());
  SetFontSize(50);      textOut(750,575,' град. C');
  SetFontSize(60);
  textOut(100,60,'Т Е М П Е Р А Т У Р А '); // Измененная строка
  SetFontSize(50);      textOut(300,250,'( 0 - 100 град. C )');
  Redraw;      end;
until vv>1;
  sp.Close();
end.
```

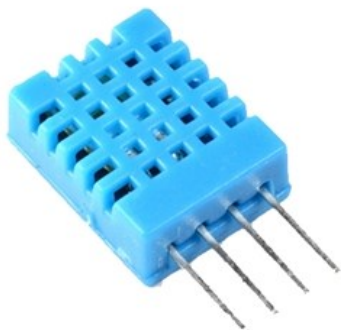
Применяя различные датчики аналогично измеряем другие физические величины

zakatayrukava.ru

$$R = 143 \text{ Ом}$$



Тензорезистор.
Измеряем силу.



Датчик влажности.



Фоторезистор.
Измеряем освещенность.