

ФОНД НОВЫХ ТЕХНОЛОГИЙ В ОБРАЗОВАНИИ «БАЙТИК»

УМНЫЙ ДОМ РУКАМИ ДЕТЕЙ

Учебно-методический комплект

Москва-Троицк-2018

Учебно-методический комплект «Умный дом руками детей» предназначен для дополнительного образования школьников 7–8 классов. УМК будет интересен руководителям центров дополнительного образования, учителям информатики и физики, преподавателям программирования и другим заинтересованным представителям основного и дополнительного образования.

Авторы: **Мединский** Владимир Вячеславович
Панарин Алексей Сергеевич
Тихонов Николай Анатольевич

«Умный дом руками детей». Учебно-методический комплект:/ В.В. Мединский, А.С. Панарин, В.В. Тихонов. - Москва, Троицк, - 2018. – 93 с.: ил.

Редакционная группа: Шумкова Е.М., Григоренко М.М., Киревнина Е.И.

Макет обложки: Васяткин В.М.

Фонд новых технологий в образовании «БАЙТИК»,
108840, г. Москва, г. Троицк, Сиреневый б-р., д. 11, тел. (495) 851-03-67, www.bytic.ru.

Отпечатано с готового оригинал-макета в типографии ИП Комаров В.В.,
108840, г. Москва, г. Троицк, ул. Нагорная, д. 6, тел.: (917) 549-87-97, www.rosstiker.ru.

Формат 60x84/8. Гарнитура «Таймс». Печать офсетная. Тираж 50 экз.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
ЗАНЯТИЕ № 1	
Тема: Вводное занятие.....	6
ЗАНЯТИЕ № 2	
Тема: Электричество. Электрические схемы.....	7
ЗАНЯТИЕ № 3	
Тема: Среда Arduino. Первая схема включения светодиода.....	15
ЗАНЯТИЕ № 4	
Тема: Управление Цифровым входом-выходом. Типы данных в среде Arduino. Оператор «if...else».....	19
ЗАНЯТИЕ № 5	
Тема: Моделирование электрических схем. Функция analogWrite.....	24
ЗАНЯТИЕ № 6	
Тема: Аналоговые выводы Arduino. Последовательный порт. Делитель напряжения.....	27
ЗАНЯТИЕ № 7	
Тема: Потенциометр и фоторезистор, делитель напряжения.....	33
Занятие № 8	
Тема: Проверочная работа.....	37
ЗАНЯТИЕ № 9	
Тема: Повторение. Термодатчик ТМР3».....	38
Занятие № 10	
Тема: Комнатный термометр и сервопривод.....	40
Занятие № 11	
Тема: Подведение итогов. Работа с сервоприводом и термодатчиком. Функция Map().....	44
ЗАНЯТИЕ № 12	
Тема: Подведение итогов.....	46



ЗАНЯТИЕ № 13

Тема: Проектная работа. Обсуждение и выбор проектов для реализации. Формирование проектных групп 47

ЗАНЯТИЕ № 14

Тема: Изучение основ 3D моделирования (Inventor) 48

ЗАНЯТИЕ № 15

Тема: Продолжаем моделировать колесо 53

ЗАНЯТИЯ № 16 – № 29

Тема: Проектная работа..... 57

ЗАНЯТИЕ № 30

Тема: Интеграция устройств в единый центр управления.
Протоколы, используемые в системах «умный дом».
Центр управления «умным домом»..... 58

ЗАНЯТИЕ № 31

Тема: Интеграция устройств в единый центр управления.
Установка openHAB на raspberry pi. Что такое MQTT?
Установка Mosquit..... 67

ЗАНЯТИЕ № 32

Тема: Интеграция устройств в единый центр управления.
Что такое ESP826?..... 78

СПИСОК ИСТОЧНИКОВ 91

ПРИЛОЖЕНИЕ 92



ВВЕДЕНИЕ

На базе Фонда «БАЙТИК» в 2017–2018 учебном году реализована программа дополнительного образования «Умный дом руками детей» с использованием гранта Президента Российской Федерации на развитие гражданского общества, предоставленного Фондом президентских грантов.

Цель программы – повышение качества предпрофессиональной подготовки учащихся посредством реализации проектной деятельности.

В рамках программы под руководством преподавателей Фонда и с привлечением сотрудников научно-исследовательских институтов наукограда Троицк подростки создавали демонстрационные работающие образцы составных элементов умного дома, управляемые с мобильных устройств:

- контроль доступа в
- помещение;
- управление освещением;
- система автоматического ухода за домашними животными;
- система полива и освещения растений;
- погодный анализатор;
- газоанализатор;
- голосовое управление электронными устройствами.

Параллельно с процессом обучения создавался учебно-методический комплект для преподавателей, отражающий ход занятий, методику преподавания, дидактические материалы и практический результат обучения.

Итогом проекта стали бумажная и электронная версии учебно-методического комплекта (УМК) «Умный дом руками детей», состоящего из данной книги, поурочного плана и рабочей программы.

В июне 2018 года в рамках программы Фондом проведена Всероссийская конференция «Умный дом руками детей», где, были представлены проектные работы школьников из регионов России.

Методика была представлена и распространена среди руководителей центров дополнительного образования, учителей информатики и физики, преподавателей программирования и других заинтересованных представителей основного и дополнительного образования на Международной конференции «Современные информационные технологии в образовании», проводимой нашей организацией в течение 29 лет.

Электронную версию УМК получают специалисты дополнительного образования из различных регионов России в соответствии с накопленной базой электронных адресов международной конференции.

Методика также будет находиться в свободном доступе на сайте Фонда «БАЙТИК».



ЗАНЯТИЕ № 1

Тема: Вводное занятие

Протяженность занятия: 90 минут.

Цель: Введение в «Умный дом».

Тип занятия: Презентация.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	Вводное занятие. Вопросы не предусмотрены.		
10 мин.	Изложение нового материала	Подготовка материала, что такое умный дом (в общих словах). Электричество, что это такое?	Подготовка презентации	Презентация на тему «Умный дом». Презентация на тему «Электричество. Техника безопасности»
50 мин.	Практическая работа	Презентация элементов умного дома: метеостанция, датчики движения, умные выключатели, система контроля доступа		
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: – В чем заключается система «Умный дом»? – Основные элементы техники безопасности при работе с электричеством. – Основные элементы и правила работы в лаборатории «Автоматизированных систем управления»		

ЗАНЯТИЕ № 2

Тема: Электричество. Электрические схемы

Протяженность занятия: 90 минут.

Цель: Изучение характеристик электричества, основных законов, основных радиоэлементов, видов электрических схем

Задачи:

Обучающие:

- Разобрать характеристики электричества
- Изучить закон Ома
- Изучить мультиметр.
- Изучить основные радиоэлементы.

Тип занятия: Комбинированный урок (лекция + практическое занятие)

Оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- безопасная макетная плата,
- плата Arduino Nano.

Программное обеспечение: Arduino IDE

План занятия

1. Организационный момент;
2. Изложение материала;
3. Практическая работа;
4. Подведение итогов.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
50 мин.	Изложение нового материала	– Электрический ток, единицы измерения; – Электрическое напряжение (разность		Приложение. Материалы к занятию



Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
		<p>потенциалов), единицы напряжения;</p> <ul style="list-style-type: none"> – Закон Ома для участка цепи; – Некоторые составляющие электрической схемы: резистор, диод, светодиод; – Мультиметр 		
15 мин.	Практическая работа	Практическая работа, по измерению напряжения источника питания, сопротивления резистора, исправности светодиода		
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	<p>Вопросы для закрепления темы:</p> <ol style="list-style-type: none"> 1. Виды электрических схем 2. Основные характеристики электричества 3. Единицы измерения электрических величин 4. Сборка схемы со светодиодом на безопасных макетных платах 		

Ход занятия

Организационный момент

Отметить отсутствующих, объявить тему.

Объяснение темы и инструктаж

Электричество – совокупность явлений, обусловленных существованием, взаимодействием и движением электрических зарядов.

Электрический ток – это упорядоченное (направленное) движение заряженных частиц под действием электрического поля.

Сила тока – это количество заряженных частиц, протекающих через поперечное сечение проводника.

Сила электрического тока измеряется прибором, называемым **Амперметром**. Величина электрического тока (количество переносимого заряда) измеряется в **амперах (А)**.

Для увеличения диапазона обозначений единицы изменения существуют такие приставки кратности как микро - микроампер (мкА), милли – миллиампер (мА).

Электрическое напряжение – физическая величина, которая равна работе электрического поля по перемещению единичного заряда из одной точки в другую.

Электрическое напряжение измеряется прибором называемым Вольтметром.

Напряжение, т. е. разность потенциалов, которая заставляет течь ток, измеряется в **Вольтах (В)**.

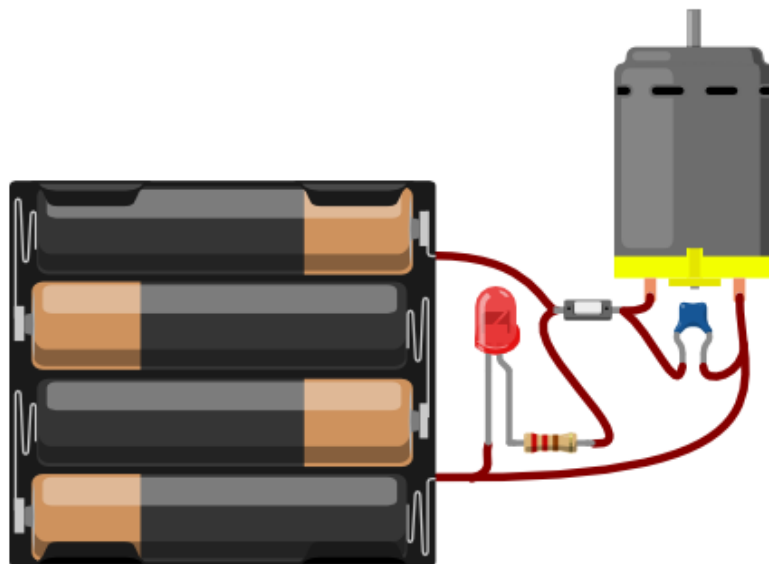
Так же, как для тока, для увеличения диапазона обозначений, существуют кратные приставки: (микро – микровольт (мкВ), милли – милливольт (мВ), кило – киловольт (кВ), мега – мегавольт (МВ)).

Напряжение ещё называют ЭДС – электродвижущей силой.

Виды электрических схем:

- Рисованная схема

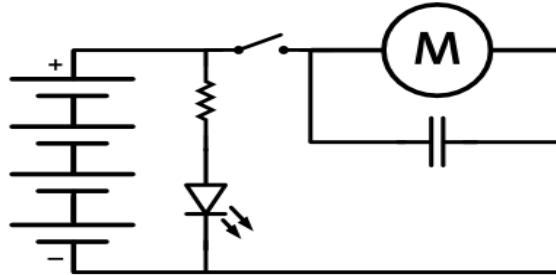
Красиво, но громоздко и непрактично:





– Принципиальная схема:

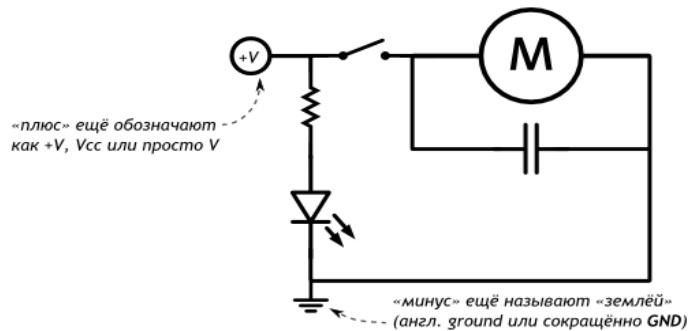
Компактно и наглядно:



- То, что соединено линией, в реальности должно быть соединено проводником
- то, что не соединено линией, в реальности должно быть электрически изолировано

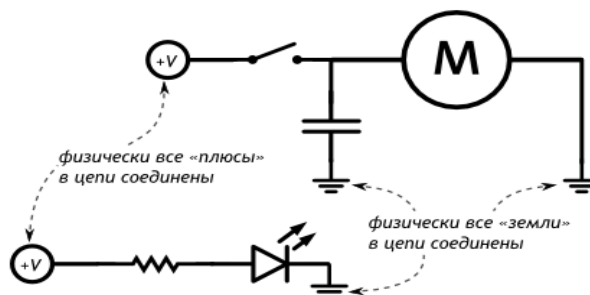
– Принципиальная схема без явного источника питания:

Источник питания зачастую не рисуют в явном виде, а используют отдельные символы для плюса и минуса. Такая схема ещё более компактна.



– Принципиальная схема с отдельными контурами:

Часто для удобства одну цепь на схемах разбивают на отдельные части. В сложных проектах так добавляются наглядности и делят зоны ответственности между несколькими инженерами-разработчиками.



Основные законы электричества

Закон Ома

Закон Ома, основанный на опытах, представляет собой в электротехнике основной закон, который устанавливает связь силы электрического тока с сопротивлением и напряжением.

Появление смартфонов, гаджетов, бытовых приборов и прочей электротехники коренным образом изменило облик современного человека. Приложены огромные усилия, направленные на исследование физических закономерностей для улучшения старой и создания новой техники. Одной из таких зависимостей является закон Ома.

Закон Ома - полученный экспериментальным путём (эмпирический) закон, который устанавливает связь силы тока в проводнике с напряжением на концах проводника и его сопротивлением, был открыт в 1826 году немецким физиком-экспериментатором Георгом Омом.

Строгая формулировка закона Ома может быть записана так: сила тока в проводнике прямо пропорциональна напряжению на его концах (разности потенциалов) и обратно пропорциональна сопротивлению этого проводника.

I – сила тока в проводнике, единица измерения силы тока - ампер [А];

U – электрическое напряжение (разность потенциалов), единица измерения напряжения - вольт [В];

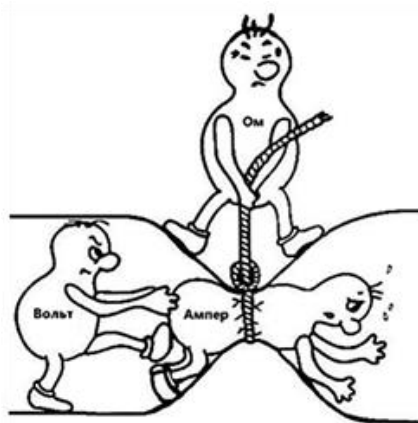
R – электрическое сопротивление проводника, единица измерения электрического сопротивления - Ом [Ом].

Закон Ома

Закон Ома – главный закон электричества

$$I = \frac{U}{R}$$

Амперы
Вольты
Омы





Закон Ома определяет силу тока в электрической цепи при заданном напряжении и известном сопротивлении.

Он позволяет рассчитать тепловые, химические и магнитные действия тока, так как они зависят от силы тока.

Закон Ома является чрезвычайно полезным в технике (электронной/электрической), поскольку он касается трех основных электрических величин: тока, напряжения и сопротивления. Он показывает, как эти три величины являются взаимозависимыми на макроскопическом уровне.

Мультиметр – это универсальный комбинированный измерительный прибор, который сочетает в себе функции нескольких измерительных приборов, то есть может измерять целый диапазон электрических величин.

Режимы работы мультиметра:



Подключение щупов мультиметра:





Резистор — искусственное «препятствие» для тока. Сопротивление в чистом виде. Резистор ограничивает силу тока, переводит часть электроэнергии в тепло.



Основные характеристики

Сопротивление (номинал)	R	Ом
Точность (допуск)	\pm	%
Мощность	P	Ватт

Диод — это электрический «ниппель». У него есть 2 полюса: анод и катод. Ток пропускается только от анода к катоду.

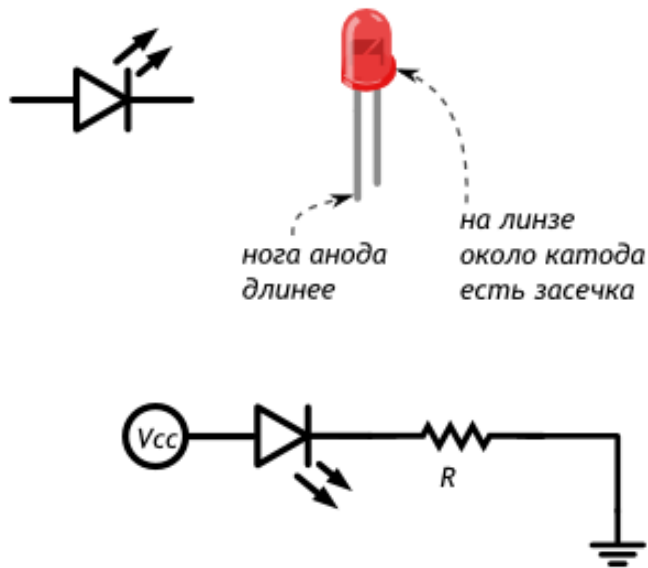
Основные характеристики

Падение прямого напряжения	V_F	Вольт
Максимальное сдерживаемое обратное напряжение	V_{DC}	Вольт
Максимальный прямой ток	I_F	Ампер

Светодиод (англ. Light Emitting Diode или просто LED) — энергоэффективная, надёжная, долговечная «лампочка». Светодиод — вид диода, который светится, когда через него проходит ток от анода (+) к катоду (-).

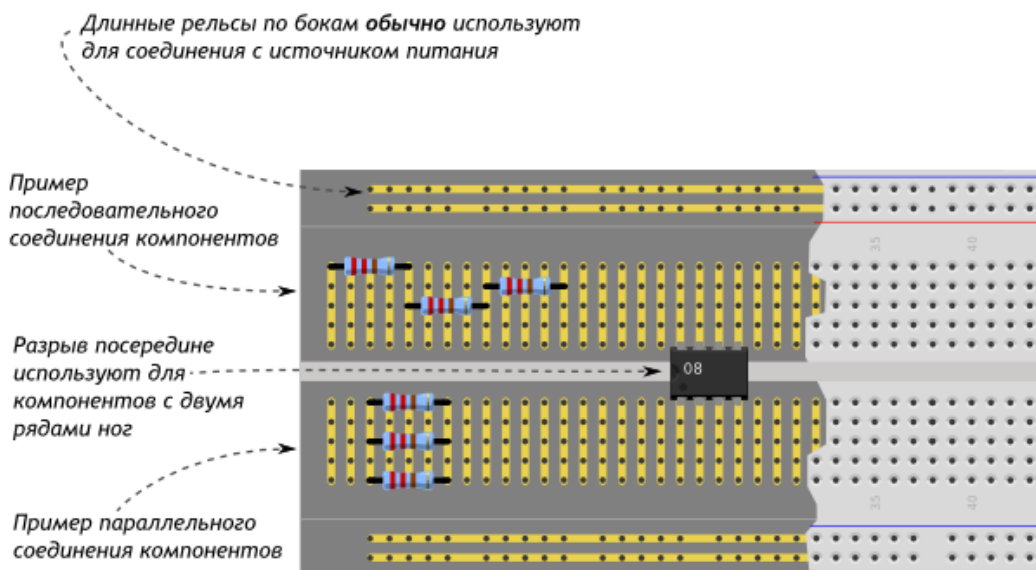
Основные характеристики

Падение напряжения	V_F	Вольт
Номинальный ток	I	Ампер
Интенсивность (яркость)	I_V	Кандела
Длина волны (цвет)	λ	Нанометр



Для питания 1 светодиода на 20 мА от 5 В, используйте резистор от 150 до 360 Ом.

Беспаячная макетная плата:



Устройство макетной платы такого типа простое. Ее основой является пластиковый корпус с большим количеством отверстий на верхней плоскости. В отверстиях расположены контактные разъемы для установки деталей. Разъемы допускают установку контактов и проводов диаметром до 0,7 мм, расстояние между ними – стандартное 2,54 мм, что позволяет устанавливать транзисторы и микросхемы в DIP-корпусах.

ЗАНЯТИЕ № 3

Тема: Среда Arduino. Первая схема включения светодиода

Протяженность занятия: 90 минут.

Цель: Изучить основные приемы работы в среде Arduino. Научиться управлять светодиодом с помощью «Arduino».

Задачи:

Обучающие:

- Сборка схем на плате Arduino;
- Программирование простых схем со светодиодом.

Тип занятия: Комбинированный урок (лекция, практическое занятие).

Оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1 – 10 кОм (для подтяжки питания) на 0,25 Вт,
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1) или провода «папа-папа»,
- безопасная макетная плата.

Программное обеспечение: Arduino IDE

План занятия

1. Организационный момент;
2. Актуализация знаний (обобщение и повторение материала с выходом на тему занятия);
3. Изложение нового знания/новой темы/материала;
4. Практическая работа;
5. Подведение итогов.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	<ul style="list-style-type: none"> – Электрический ток, единицы измерения. – Электрическое напряжение (разность потенциалов), единицы напряжения. 		



Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
		<ul style="list-style-type: none"> — Основной закон электричества (закон Ома для участка цепи). — Некоторые составляющие электрической схемы: резистор, диод, светодиод. — Мультиметр. — Микроконтроллер Arduino. 		
10 мин.	Изложение нового материала	Схема включения диода. Среда Arduino. Минимальная программа в Arduino Blink. Команды: pinMode, digitalWrite, delay.	Подготовка презентации	
50 мин.	Практическая работа	Практическая работа, сборка схемы со светодиодом, добавляем кнопку Изменение и коррекция программы		Приложение схемы, доп. материалы
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	<p>Вопросы для закрепления темы:</p> <ul style="list-style-type: none"> — Схема включения диода — Коррекция схемы — Ограничивающий резистор — Среда Arduino. Минимальный синтаксис Arduino. — Стандартная программа Blink. Команды pinmode, digitalWrite, delay. — Чтение порта, запись в порт 		

Ход занятия

Организационный момент

Отметить отсутствующих, объявить тему.

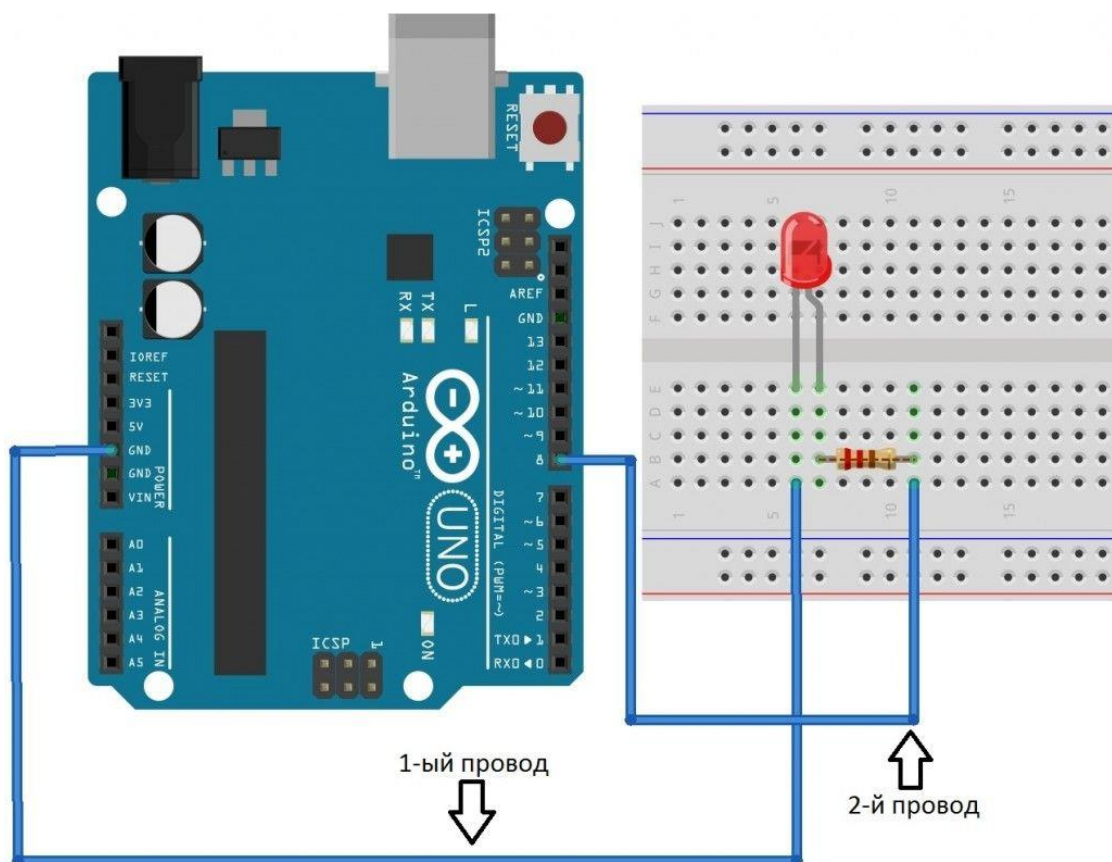
Актуализация деятельности учащихся

Проверка знаний ранее изученного материала, коррекция опорных знаний.

Объяснение темы и инструктаж

Краткие сведения: **Arduino** (Arduino) — популярная аппаратная вычислительная платформа, основными компонентами которой являются плата ввода-вывода и среда разработки. Arduino может использоваться как для создания автономных интерактивных объектов, роботов, так и подключаться к программному обеспечению, выполняемому на компьютере. Платы имеют аналоговые и цифровые порты, к которым можно подключить практически любое простое устройство: кнопка, датчик, мотор, экран.

Практическая работа



1. Объяснение схемы подключения светодиода к Arduino.
2. Для подключения мигающего светодиода на Arduino и управления им понадобятся:
 - a) плата Arduino;
 - b) breadboard;
 - c) провода «папа-папа»;
 - d) светодиод;
 - e) резистор.



3. Программирование:

```
int led = 8;

void setup()
{
    pinMode(led, OUTPUT);
}

void loop()
{
    digitalWrite(led, HIGH);
    delay(1000);
    digitalWrite(led, LOW);
    delay(1000);
}
```

Разобрать программу, объяснить назначение каждой команды.

Задания:

1. Изменять скорость моргания светодиода.
2. Сообщить азбукой Морзе какое-то слово.

Учащиеся самостоятельно должны найти необходимые буквы, и запрограммировать это.

ЗАНЯТИЕ № 4

Тема: Управление Цифровым входом-выходом.
Типы данных в среде Arduino. Оператор «if...else»

Протяженность занятия: 90 минут

Цель: Научиться управлять цифровыми входами-выходами микросхемы Arduino-Nano (в зависимости от имеющегося оборудования). Разобраться с типами данных и для чего они нужны. Использовать структуру if...else в написании программ управления светодиодами.

Задачи:

Обучающие:

- Собрать схему использования кнопки
- Собрать схему использования управления светодиодом

Тип занятия: Комбинированный урок (лекция + практическое занятие)

Оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1 – 10 кОм (для подтяжки питания) на 0,25 Вт,
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- беспаячная макетная плата.

Программное обеспечение:

- Arduino IDE

План занятия

1. Организационный момент;
2. Актуализация знаний (обобщение и повторение материала с выходом на тему занятия);
3. Изложение нового знания/новой темы/материала;
4. Практическая работа;
5. Подведение итогов.



Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	1. Схема ошибочного включения диода 2. Коррекция схемы 3. Ограничивающий резистор 4. Среда Arduino. Минимальный синтаксис Arduino. 5. Стандартная программа Blink. Команды pinMode, digitalWrite, delay. 6. Чтение порта		
10 мин.	Изложение нового материала	Типы данных в среде Arduino. Цифровые входы – выходы. Оператор if...else	Подготовка презентации	
50 мин.	Практическая работа	Практическая работа, по сборке схемы с использованием светодиода и управлением им по нажатию клавиши. Сборка схемы типа светофор и Азбука Морзе – SOS.		Приложение схемы.
10 мин.	Уборка рабочих мест			

Ход занятия

Переменные и типы данных

Переменная — поименованная область памяти, адрес которой можно использовать для осуществления доступа к данным и изменять значение в ходе выполнения программы.

Типы переменных — это описания множества значений, которые переменная может принимать, а также размер памяти, который она занимает.

Например:

int (16 бит — 2 байта) — число в диапазоне от -32768 до 32767

unsigned int (16 бит — 2 байта) — число в диапазоне от -0 до 65535



Название	Вес	Диапазон	Особенность
boolean	1 байт	0 или 1	Логическая переменная, может принимать значения true (1) и false (0)
char	1 байт	-128... 127	Хранит номер символа из таблицы символов ASCII
byte	1 байт	0... 255	
int	2 байта	-32 768... 32 767	
unsigned int	2 байта	0... 65 535	
word	2 байта	0... 65 535	То же самое, что unsigned int
long	4 байта	-2 147 483 648... 2 147 483 647	- 2 миллиарда... 2 миллиарда
unsigned long	4 байта	0... 4 294 967 295	0... 4 миллиарда...
float	4 байта	-3.4028235E+38... 3.4028235E+38	Хранит числа с плавающей точкой (десятичные дроби). Точность: 6-7 знаков
double	4 байта		То же самое, что float

Примеры объявления переменных

`int my_val;` объявление переменной с именем `my_val`

`int my_val = 10;` объявление переменной с именем `my_val` и присвоение ей значения `10`

Оператор условия

`if` (Условие)

```
{
// выполнять действия если условие выполняется
}
```

`if` (Условие)

```
{
// выполнять действия если условие выполняется
}
else {
// выполнять другие действия если условие не выполняется
}
```

Пример

```
if (buttonState==HIGH)
```



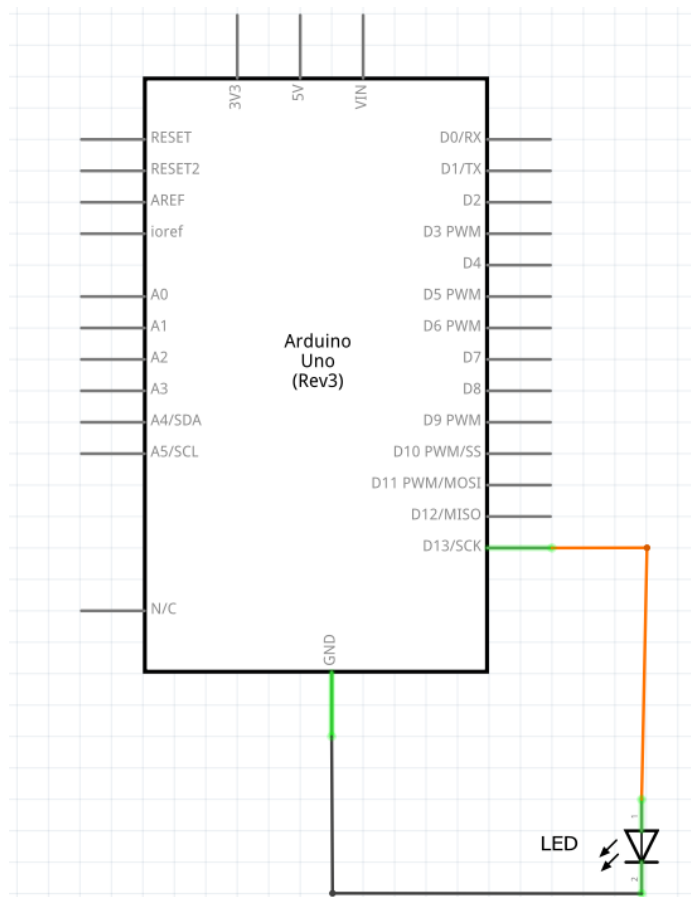
```
{  
// выполнять действия если условие выполняется  
digitalWrite(13, HIGH);  
}  
else {  
// выполнять другие действия если условие не выполняется  
digitalWrite(13, LOW);  
}
```

Операторы сравнения

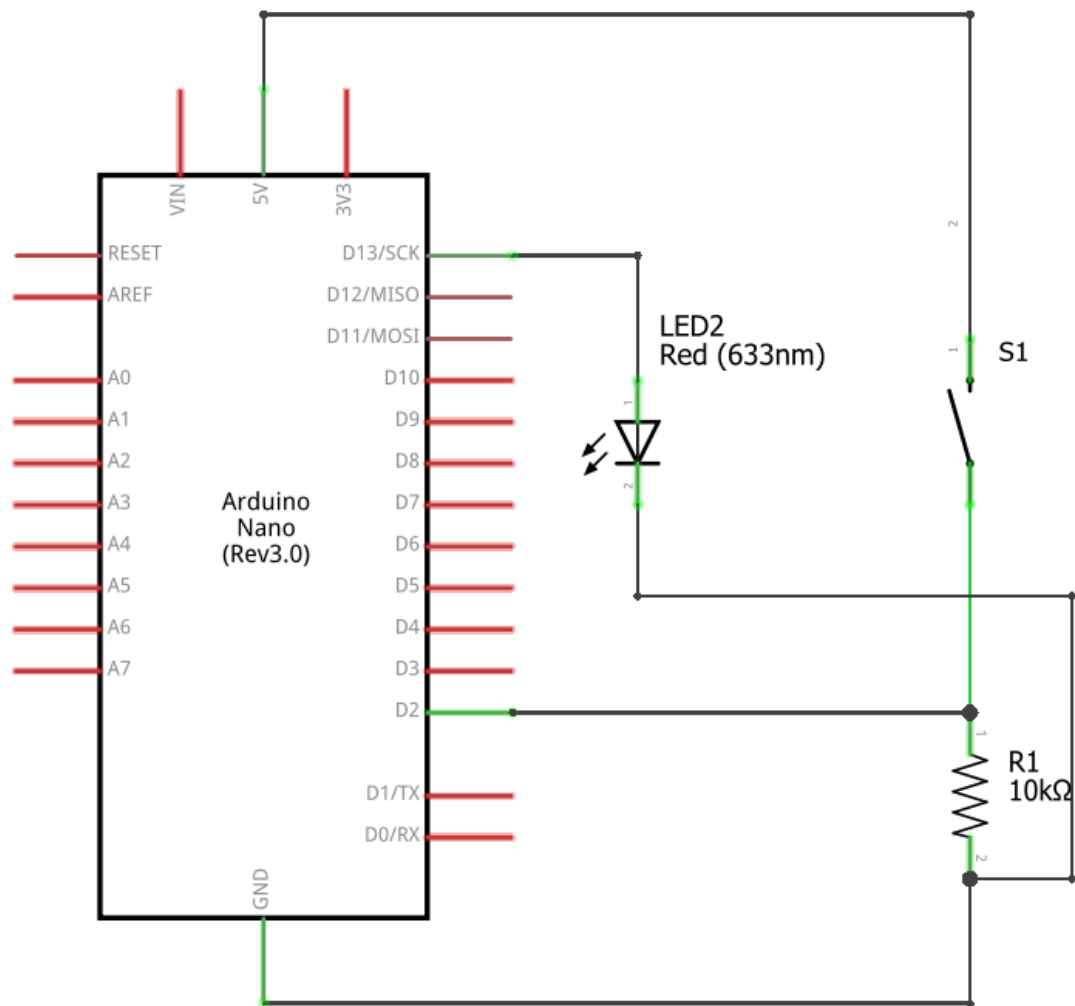
```
x == y // (x равно y)  
x != y // (x не равно y)  
x < y // (x меньше чем y)  
x > y // (x больше чем y)  
x <= y // (x меньше чем или равно y)  
x >= y // (x больше чем или равно y)
```

Задания на повторение изученного

1. Собрать на Arduino схему с одним светодиодом, моргать с задержкой полсекунды.



2. Та же схема, моргать сигнал SOS азбукой Морзе (три точки три тире три точки)
3. Подключить кнопку к цифровому входу Arduino и выполнить следующие задачи



- Написать программу для реализации следующей логики: Если кнопка нажата, светодиод горит, если кнопка не нажата, светодиод не горит.
- Написать программу для реализации следующей логики: При первом нажатии на кнопку светодиод горит, при втором нажатии на кнопку светодиод гаснет и т. д.
- В предыдущем задании нажатие на кнопку не каждый раз срабатывает. Почему? Как это исправить?



ЗАНЯТИЕ № 5

Тема: Моделирование электрических схем. Функция analogWrite

Протяженность занятия: 90 минут

Цель: Научиться моделировать электрические схемы в среде Tinkercad. Разобраться с принципом работы и возможностями применения команды analogWrite.

Задачи:

Обучающие:

- Научиться работать в системе Tinkercad
- Собрать схему с использованием светодиода и кнопки.
- Написать программу в среде Arduino по управлению включением светодиода при помощи кнопки.

Тип занятия: Комбинированный урок (лекция + практическое занятие)

Оборудование:

- резистор (200-300 Ом (для ограничения тока через светодиод)) на 0,25 Вт,
- подтягивающий резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт),
- светодиод (желтый, зеленый, красный),
- фоторезистор
- потенциометр или два резистора разного номинала
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- безопасная макетная плата.

Программное обеспечение:

- Arduino IDE

План занятия

1. Организационный момент;
2. Актуализация знаний (обобщение и повторение материала с выходом на тему занятия);
3. Изложение нового знания/новой темы/материала;
4. Практическая работа;
5. Подведение итогов.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	1. Схема ошибочного включения диода. 2. Коррекция схемы. 3. Ограничивающий резистор. 4. Среда Arduino. Минимальный синтаксис Arduino. 5. Стандартная программа Blink. Команды pinMode, digitalWrite, delay. 6. Чтение порта.		
10 мин.	Изложение нового материала	Регистрация на портале tinkercad.com. Демонстрация основных элементов интерфейса. Демонстрация принципов сборки простейших электрических схем в новой среде.	Подготовка презентации	
50 мин.	Практическая работа	Практическая работа, по сборке схемы с использованием светодиода и управлением им по нажатию клавиши.		
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: — Для чего используется виртуальная среда tinkercad, каковы ее возможности и ограничения. — Типовая схема включения светодиода. — Ограничивающий резистор для чего? — Резистор подтягивания для чего? — Чем отличаются digitalWrite и analogWrite		

Ход занятия



В век повсеместной компьютеризации и виртуализации уже во многих областях используются виртуальные математические модели различных систем, не обошло это и электронику. На данный момент существует большое количество программных продуктов, позволяющих собрать электрическую схему на экране компьютера и смоделировать ее поведения в различных условиях.

Такой подход стал очень популярен, т.к. позволяет значительно сократить сроки разработки устройств, скорость корректировки ошибок. Гораздо проще и быстрее нажать мышкой на элемент в схеме и изменить его номинал, нежели взять паяльник и заменить элемент в реальной схеме.

Виртуальная схема дает возможность ошибаться без материальных затрат. До недавнего времени все продукты требовали установки на компьютер и сложной настройки, но с появлением Tinkercad все стало проще и быстрее.

Tinkercad позволяет (прямо в браузере) собирать и моделировать электрические схемы, программировать контроллеры Arduino.

Практическая работа

Повторить все схемы и задания предыдущих уроков в виртуальной среде Tinkercad.

ЗАНЯТИЕ № 6

Тема: Аналоговые выходы Arduino. Последовательный порт.
Делитель напряжения

Протяженность занятия: 90 минут

Цель: Разобраться с понятием Аналоговые входы и научиться их использовать. Научиться использовать монитор последовательного порта Arduino IDE.

Задачи:

Обучающие:

- Собрать минимальную программу для инициализации последовательного порта. Считываем значение с цифрового порта и выводим значение на монитор Arduino IDE.
- Сбор схемы с кнопкой и частичным ее управлением.
- Изменение яркости свечения светодиода и управление через кнопку. Учимся управлять аналоговым выходом Arduino.

Тип занятия: Комбинированный урок (лекция + практическое занятие)

Оборудование:

- резистор (200-300 Ом (для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт),
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»,
- беспаячная макетная плата.

Программное обеспечение:

- Arduino IDE

План занятия

1. Организационный момент;
2. Актуализация знаний (обобщение и повторение материала с выходом на тему занятия);
3. Изложение нового знания/новой темы/материала;
4. Практическая работа;
5. Подведение итогов.



Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	<ol style="list-style-type: none"> 1. Как надо включать диод (ответ через резистор) 2. Зачем необходим резистор в этой схеме (токоограничивающий, чтобы не сгорел диод и порт на Arduino) 3. Режимы работы цифровых портов Arduino (вход/выход) 4. Как выбираются эти режимы работы (pinMode) 5. Как подают сигналы на выход Arduino (digitalWrite) 6. Как считывают сигналы со входа Arduino (digitalRead) 		
10 мин.	Изложение нового материала	Аналоговые выводы Arduino. Последовательный порт. Делитель напряжения	Подготовка презентации	
50 мин.	Практическая работа	Схема с кнопкой и управление ею. Управление кнопкой и светодиодом, изменение яркости свечения светодиода. Зарисовка и сборка схемы делителя напряжения		Приложение Доп. материалы
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: <ol style="list-style-type: none"> 1. Делитель напряжения 2. Считывание аналогового напряжения 3. Запись аналогового напряжения 		

Ход занятия

```

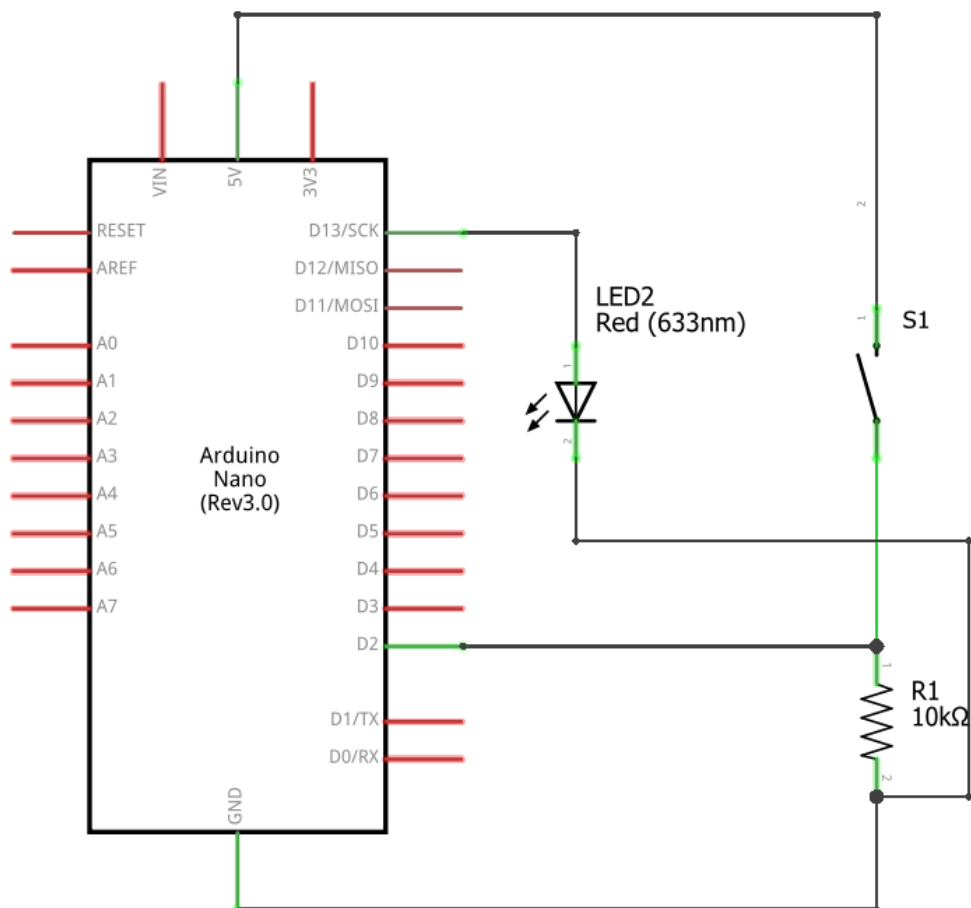
void setup() {
  Serial.begin(9600); // откроем последовательное соединение
  // отправим сообщение с переводом строки
}

void loop() {
  //посмотрим, что будет, если поместить сообщение в loop() без перевода строки
  Serial.println("Privet vsem");
}

```

1. Работа с монитором последовательного порта. Объяснение примера программы. Практика — выводим в последовательный порт различные сообщения.

2. Собрать схему:





Выводить в последовательный порт состояние кнопки Нажата/Отжата.

```
/*
  Пример для считывания нажатия кнопки и
  отображения на экране компьютера считанного значения
*/

// на пин D2 подключаем кнопку
int pushButton = 2;

void setup() {
  // Инициализируем последовательный порт для вывода информации
  Serial.begin(9600);
  // настраиваем пин с кнопкой на вход
  pinMode(pushButton, INPUT);
}

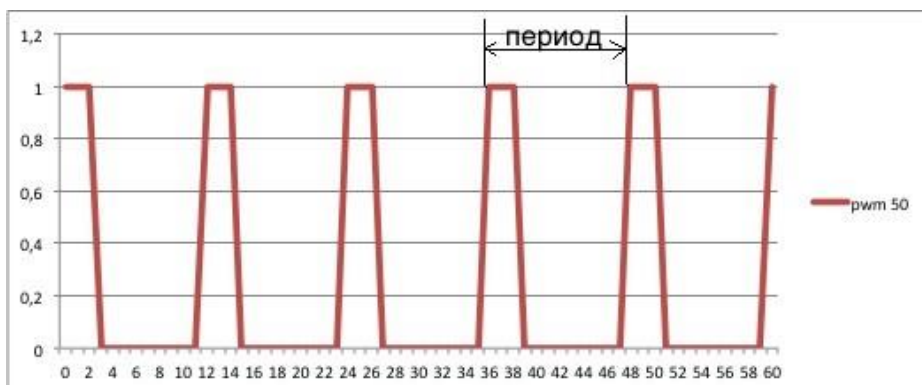
void loop() {
  // Считываем значение с кнопки
  int buttonState = digitalRead(pushButton);
  // выводим на экран считанное значение
  Serial.println(buttonState);
  delay(1);
}
```

Широтно-импульсная модуляция (аналоговый сигнал на выходе Arduino)

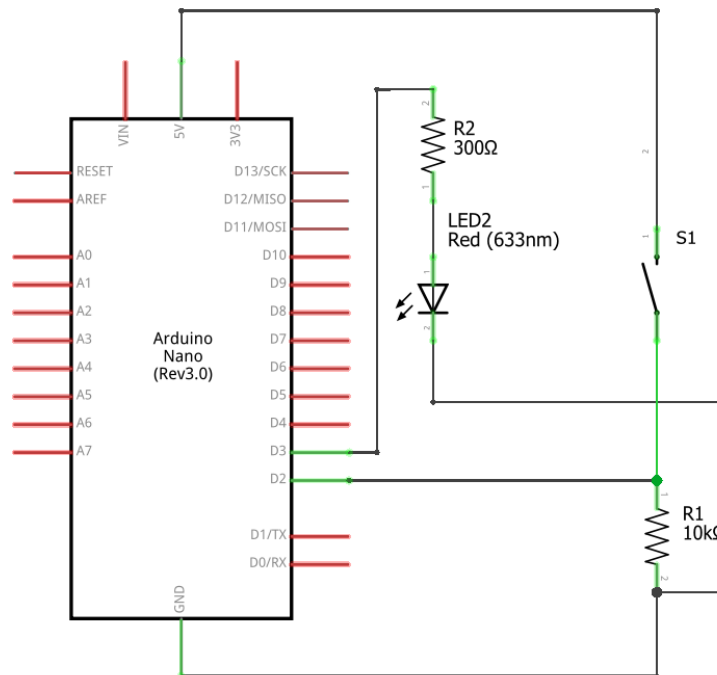
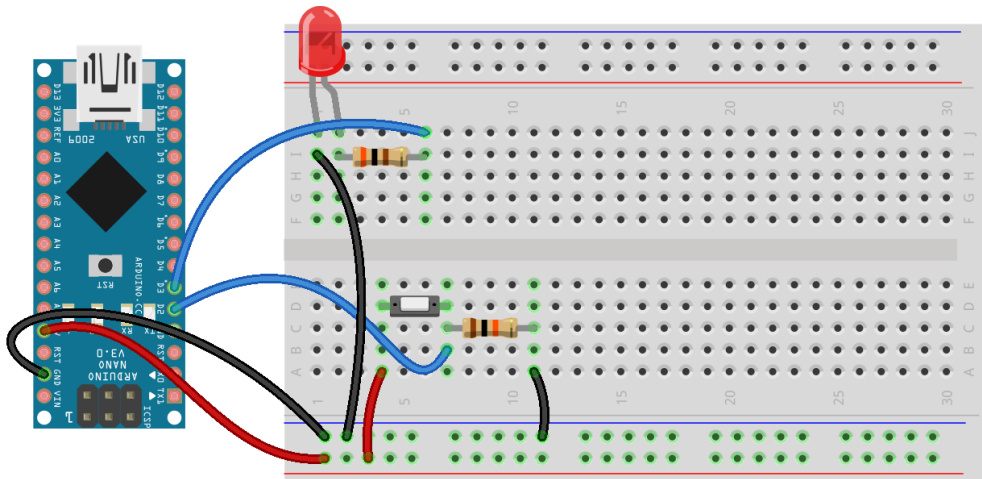
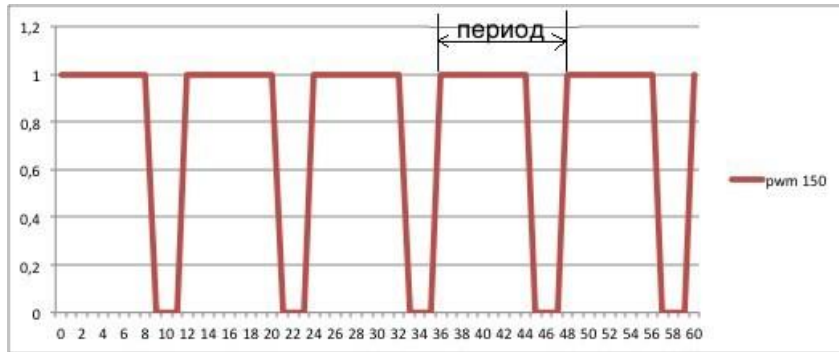
Функция `analogWrite()` формирует заданное аналоговое напряжение на выводе в виде ШИМ-сигнала. Может использоваться для варьирования яркости свечения светодиода или управления скоростью вращения двигателя.

После вызова `analogWrite()`, на выводе будет непрерывно генерироваться ШИМ-сигнал с заданным коэффициентом заполнения до следующего вызова функции `analogWrite()` (либо до момента вызова `digitalRead()` или `digitalWrite()`, взаимодействующих с этим же выводом).

```
analogWrite(2, 50);
```



`analogWrite(2, 150);`





Изменение яркости свечения светодиода

Изменять яркость свечения можно только на ножках, обозначенных *. Например, D3, D5.

Внесите соответствующее изменение в программу, изменив номер ножки.

Вместо **digitalWrite(ledpin, High)** теперь пишем так **analogWrite(ledpin, ledlevel)**.

ledlevel — переменная, в которой будем запоминать яркость свечения светодиода, она изменяется от 0 до 255.

Сделайте программу так, что изначально яркость свечения диода равна **0**, при каждом нажатии на кнопку она увеличивается на **51**.

Как только значение яркости становится больше **255**, оно сразу обнуляется.

ЗАНЯТИЕ № 7

Тема: Потенциометр и фоторезистор, делитель напряжения

Актуальность темы: Использование фоторезистора для управления светом (наличие освещенности).

Фоторезисторы — это резисторы, у которых меняется сопротивление в зависимости от действия света на светочувствительную поверхность. Сопротивление не зависит от величины напряжения, в отличие от обычного резистора.

В дальнейшем планируется полученные знания использовать в следующих возможных проектах:

- Системы безопасности (датчики движения, датчики присутствия)
- Управление освещением
 - умные выключатели, диммеры;
 - Модули управления шторами, жалюзи-рольставни;
 - RGB- и RGBW-контроллеры для управления светодиодными светильниками, прежде всего светодиодными лентами;
 - Датчики движения и присутствия;
 - Датчики освещенности;

Такие устройства позволяют автоматизировать управление светом и чаще всего используются, чтобы:

- автоматически включать свет, когда люди входят в помещение, и выключать, когда выходят;
- автоматически поддерживать освещенность на постоянном уровне, регулируя яркость светильников и положение жалюзи или штор;
- автоматически регулировать освещенность в зависимости от сезона и времени суток или по другим заранее заданным правилам.

— Система освещения (Lighting control systems, LCS) контролирует уровень освещенности в помещении, в том числе для экономии электроэнергии, за счет рационального использования естественного освещения.

Некоторые подсистемы:

- автоматика для включения/выключения света в заданное время суток;
- датчики движения для включения света только тогда, когда в помещении кто-то находится;
- автоматика для открытия/закрытия ставней, жалюзи, для регулировки прозрачности специальных оконных стекол.



- Система безопасности и мониторинга
 - система видеонаблюдения;
 - система контроля доступа в помещения;
 - имитация присутствия

Использование **потенциометра** возможно практически в любых схемах элементов «Умного дома». **Потенциометры** - это регулируемые делители напряжения, которые предназначены для регулирования напряжения при неизменной величине тока, и выполнены по типу переменного резистора.

В дальнейшем планируется полученные знания использовать в следующих возможных проектах:

- Изменение подачи значения U на выходе;
- Регулирование различных показателей мультимедийной аппаратуры (усилители, эквалайзеры и регуляторы тембра);
- В старых моделях телевизоров имелся специальный рычаг для регулировки яркости изображения, контрастности и цветопередачи;
- Масштабирование промежуточных результатов или задание начальных условий для расчета в аналоговых компьютерах;
- Тонкая настройка внутренних спецификаций в электрических цепях;
- Различные типы контроллеров;
- Подстройка мощности

Делитель напряжения (см. потенциометр). При проектировании электрических цепей возникают случаи, когда необходимо уменьшить величину напряжения (разделить его на несколько частей) и только часть подавать на нагрузку. Для этих целей используют делители напряжения. Они основаны на втором законе Кирхгофа.

Примеры применения делителя напряжения:

- Как делитель напряжения. Представьте, что у Вас есть лампочка, которая может работать только от 6 вольт и есть батарейка на 9 вольт. В этом случае при подключении лампочки к батарейке, лампочка сгорит. Для того, чтобы лампочка работала в номинальном режиме, напряжение 9 В необходимо разделить на 6 и 3 вольта. Данную задачу выполняют простейшие делители напряжения на резисторах.
- Датчик параметр - напряжение. Сопротивление резистивных элементов зависит от многих параметров, например, температура. Помещаем одно из сопротивлений в среду с изменяющейся температурой. В результате при изменении температуры будет изменяться сопротивление одного из делителей напряжения. Изменяется ток через делитель. Согласно закону Ома, входное напряжение перераспределяется между двумя сопротивлениями.



— Усилитель напряжения. Делитель напряжения может использоваться для усиления входного напряжения. Это возможно, если динамическое сопротивление одного из элементов делителя отрицательное, например, на участке вольтамперной характеристики туннельного диода.

Ограничения при использовании резистивных делителей напряжения

— Номинал сопротивлений делителя напряжения на резисторах должен быть в 100 - 1000 раз меньше, чем номинальное сопротивление нагрузки, подключаемой к делителю. В противном случае сопротивление нагрузки уменьшит величину разделенного делителем напряжения.

— Малые значения сопротивлений, являющихся делителем напряжения, приводят к большим потерям активной мощности. Через делитель протекают большие токи. Необходимо подбирать сопротивления, чтобы они не перегорали и могли рассеять такую величину отдаваемой энергии в окружающую среду.

— Резистивный делитель напряжения нельзя использовать для подключения мощных электрических приборов: электрические машины, нагревательные элементы, индукционные печи.

— Снижение КПД схемы за счет потерь на активных элементах делителя напряжения.

— Для получения точных результатов в делителе напряжения необходимо использовать прецизионные (высокоточные) сопротивления.

Цель занятия: Разобраться с некоторыми особенностями работы фоторезистора, потенциометра и делителя напряжения.

Задачи занятия:

— Собрать схему использования фоторезистора;

— Собрать схему использования потенциометра или делителя напряжения, снять данные;

— Написать программу в среде Arduino, по снятию данных с фоторезистора и делителя напряжения.

Техническое оборудование:

— резистор (200-300 Ом (для ограничения тока через светодиод) на 0,25 Вт,

— подтягивающий резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт),

— светодиод (желтый, зеленый, красный),

— штыревые кабели (с разъемами BLS1 или провода «папа-папа»),

— беспаячная макетная плата.



Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	<ul style="list-style-type: none"> — Виды выводов Arduino — Аналоговые выводы — AnalogWrite, AnalogRead — Вывод информации в последовательный порт 		
10 мин.	Изложение нового материала	Потенциометр, Фоторезистор	Подготовка презентации	
50 мин.	Практическая работа	Практическая работа, по сборке схемы с использованием фоторезистора и потенциометр		Приложение: схемы, доп. материалы
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: <ul style="list-style-type: none"> — Потенциометр — Фоторезистор — Для чего используется потенциометр — Для чего используется фоторезистор — Что такое делитель напряжения 		

ЗАНЯТИЕ № 8

Тема: Проверочная работа

Тестовое задание приведено в Приложении.

Практическое задание: «Светофор с кнопкой»: Необходимо собрать светофор из трех светодиодов и тактовой кнопки.

Работать он должен по следующей схеме:

- Светится зеленый светодиод.
- По нажатию кнопки *выключается зеленый и включается желтый светодиод.*
- Через 2 секунды желтый светодиод гаснет и загорается красный.
- Красный светодиод горит 10 секунд, после чего гаснет, и включается желтый на 2 секунды.
- Выключается желтый светодиод и загорается зеленый до следующего нажатия кнопки.

Цель занятия: Определить степень освоенности материала предыдущих занятий. Анализ готовности учащихся к самостоятельному выполнению практических заданий.

Задачи занятия: Ответить на вопросы тестового задания. Выполнить практическую задачу.

Техническое оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт),
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- безопасная макетная плата.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
30 мин.	Выполнение тестового задания			
50 мин.	Выполнение практической части			



ЗАНЯТИЕ № 9

Тема: Повторение. Термодатчик TMP36»

Актуальность темы:

Температурный датчик

Датчики температуры нужны для того, чтобы проконтролировать температуру в помещении, жидкости, твердого объекта или расплавленного металла.

Основой действия температурных датчиков в автоматизированном управлении является изменение температуры в электрический сигнал. Это обуславливает преимущества электрических измерений: результаты легко передавать по сети, скорость передачи может быть достаточно высокой. Величины могут преобразовываться друг в друга и обратно. Цифровой код создает повышенную точность замера, скорость и чувствительность.

Параметры выбора датчика температуры:

- Диапазон рабочей температуры.
- Возможность погружения датчика в объект измерения или среду. Если это невозможно, то лучше выбрать пирометр или термометр.
- Условия проведения замеров. Если нужно измерять в агрессивной среде, то надо выбирать датчик в коррозионностойком корпусе, или бесконтактного типа. Также следует определить наличие давления, влажности и т.д.
- Время работы датчика до калибровки или замены. Многие датчики не могут долго и стабильно работать (термисторы).
- Величина сигнала выхода. Существуют датчики, выдающие сигнал по току, или в градусах.

Технические данные: погрешность, разрешение, напряжение, время срабатывания. Для полупроводников важен тип корпуса.

Цель занятия: Разобраться с термодатчиком TMP36

Задачи занятия:

- Разобраться – что такое термодатчик TMP36.
- Освоение математических методов вычисления температуры, на основании снятых данных с датчика температуры.
- Создание на основе датчика комнатного термометра вы вода его в последовательный монитор порта Arduino IDE.
- Повторить схему с использованием реальных компонентов.

Техническое оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1кОм – 10 кОм на 0,25 Вт),
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- беспаячная макетная плата.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		
5 мин.	Повторение пройденного материала	Разбор проверочной работы. Повторение пройденного материала		
10 мин.	Изложение нового материала	Температурный датчик TMP36. Tinkercad	Подготовка презентации	
50 мин.	Практическая работа	Подключить аналоговый термометр к Arduino, написать программу для вывода на экран компьютера температуры. Продолжение: Подключить три светодиода желтый, красный и зеленый. При изменении температуры больше чем на полградуса красный светодиод начинает моргать. Если температура не меняется, то красный диод не моргает. Продолжение: Если температура увеличилась от начальной на один градус, то загорается желтый диод, если температура уменьшилась — загорается зеленый. В один момент времени может гореть только один из этих двух светодиодов. Поведение красного светодиода не меняется.		Приложение схемы
10 мин.	Уборка рабочих мест			



ЗАНЯТИЕ № 10

Тема: Комнатный термометр и сервопривод

Актуальность темы. Компьютерное моделирование является в настоящее время важнейшей составной частью процесса проектирования электронных устройств. Это объясняется следующими причинами:

- Необходимостью сокращения сроков разработки новых электронных устройств.
- Наличием эффективных алгоритмов и программ компьютерного моделирования электрических цепей.
- Развитой теорией математического моделирования электронных компонентов.

Современные программы моделирования электронных цепей представляют виртуальные лаборатории, включающие обширные библиотеки электронных компонентов. Они дают возможность инженеру проверить, удовлетворяет ли спроектированное устройство требованиям технического задания, когда используются реальные компоненты с характеристиками, отличающимися от идеальных.

Многие программы позволяют автоматизировать все стадии проектирования электронных устройств, включая подготовку принципиальных схем, моделирование процессов в аналоговых и цифровых цепях, компоновку и трассировку печатных плат, редактирование и расширение библиотек компонентов.

Программы схемотехнического моделирования широко используются для автоматизации проектирования интегральных схем, поскольку физическое моделирование ИС связано с большими материальными затратами. Для устройств энергетической электроники проблема компьютерного моделирования является не менее актуальной. Это объясняется тем, что импульсные преобразователи содержат элементы с резко нелинейными характеристиками. Аналитический расчет таких устройств является весьма трудоемким.

Симулятором называют устройство или сервис, имитирующие определенные функции другой системы, но не претендующим на создание точной копии. Это некоторая виртуальная среда, в которой мы просто моделируем другую систему. Эмулятор — это полноценный аналог, способный заменить оригинал. Tinkercad симулирует работу электронных схем и контроллера, но при этом он является эмулятором Arduino, реализуя практически все базовые функции Arduino IDE — от среды редактирования и компилятора, до монитора порта и подключения библиотек.

Некоторые оффлайн-программы: Pspice, Micro-CAP, Multisim, Circuit, PSIM и др.

Некоторые онлайн-программы: Tinkercad, EasyEDA, Circuits Sims, DcAcLab, EveryCircuit, DoCircuits, PartSim, 123DCircuits, TinaCloud, Spicy schematics, Gecko simulations

Tinkercad – это онлайн сервис, который сейчас принадлежит мастодонту мира CAD-систем — компании Autodesk. Tinkercad уже давно известен многим как простая и бесплатная среда для обучения 3D-моделированию. С ее помощью можно достаточно легко создавать свои модели и отправлять их на 3D-печать.

Tinkercad получил возможность создания электронных схем и подключения их к симулятору виртуальной платы Arduino. Эти крайне важные и мощные инструменты способны существенно облегчить начинающим разработчикам Arduino процессы обучения, проектирования и программирования новых схем.

Температурный датчик

Датчики температуры нужны для того, чтобы проконтролировать температуру в помещении, жидкости, твердого объекта или расплавленного металла.

Основой действия температурных датчиков в автоматизированном управлении является изменение температуры в электрический сигнал. Это обуславливает преимущества электрических измерений: результаты легко передавать по сети, скорость передачи может быть достаточно высокой. Величины могут преобразовываться друг в друга и обратно. Цифровой код создает повышенную точность замера, скорость и чувствительность.

Параметры выбора датчика температуры:

- Диапазон рабочей температуры.
- Возможность погружения датчика в объект измерения или среду. Если это невозможно, то лучше выбрать пирометр или термометр.
- Условия проведения замеров. Если нужно измерять в агрессивной среде, то надо выбирать датчик в коррозионностойком корпусе, или бесконтактного типа. Также следует определить наличие давления, влажности и т.д.
- Время работы датчика до калибровки или замены. Многие датчики не могут долго и стабильно работать (термисторы).
- Величина сигнала выхода. Существуют датчики, выдающие сигнал по току, или в градусах.
- Технические данные: погрешность, разрешение, напряжение, время срабатывания. Для полупроводников важен тип корпуса.

Сервопривод

Подобные устройства и механизмы оснащены датчиком, который отслеживает определенный параметр. Например, усилие, положение или скорость, а также управляющий блок в виде электронного устройства. Задачей этого устройства является поддержание необходимых параметров в автоматическом режиме во время функционирования устройства, в



зависимости от вида поступающего сигнала от датчика в определенные периоды времени.

Сервоприводы в настоящее время используются достаточно широко. Так, например, они применяются в различных точных приборах, промышленных роботах, автоматах по производству печатных плат, станках с программным управлением, различные клапаны и задвижки.

Наиболее популярными стали быстродействующие приводы в авиамодельном деле. Серводвигатели имеют достоинство в эффективности расхода электрической энергии, а также равномерного движения.

В начале появления серводвигателей использовались коллекторные трехполюсные моторы с обмотками на роторе, и с постоянными магнитами на статоре. Кроме этого, в конструкции двигателя был узел с коллектором и щетками. Далее, по мере технического прогресса число обмоток двигателя увеличилось до пяти, а момент вращения возрос, также, как и скорость разгона.

Следующим этапом развития серводвигателей было расположение обмоток снаружи магнитов. Этим снизили массу ротора, уменьшили время разгона. При этом стоимость двигателя увеличилась. В результате дальнейшего проектирования серводвигателей было решено отказаться от наличия коллектора в устройстве двигателя. Стали применяться двигатели с постоянными магнитами ротора. Мотор стал без щеток, эффективность его возросла вследствие увеличения крутящего момента, скорости и ускорения.

В последнее время наиболее популярными стали сервомоторы, работающие от программируемого контроллера (Arduino). Вследствие этого открылись большие возможности для проектирования точных станков, роботостроения, авиастроения (квадрокоптеры).

Так как приводы с моторами без коллекторов обладают высокими функциональными характеристиками, точным управлением, повышенной эффективностью, они часто применяются в промышленном оборудовании, бытовой технике (мощные пылесосы с фильтрами), и даже в детских игрушках.

Цель занятия: Разобраться с методами виртуализации схем и изучении online-комплекса Tinkercad

Задачи занятия.

- Собрать схему с использованием Tinkercad
- Освоение математических методов вычисления температуры, на основании снятых данных с датчика температуры
- Повторить схему с использованием реальных компонентов

Техническое оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт),

- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- безопасная макетная плата.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	Температурный датчик TMP036. Делитель напряжения		
10 мин.	Изложение нового материала	Температурный датчик Tinkercad. Сервопривод	Подготовка презентации	
50 мин.	Практическая работа	Сбор схемы с использованием Tinkercad и повторение через реальные компоненты		Приложение схемы. См. так же предыдущее занятие
10 мин.	Уборка рабочих мест			



ЗАНЯТИЕ № 11

Тема: Подведение итогов. Работа с сервоприводом и термодатчиком.
Функция `Map()`

Актуальность темы: Встроенная функция `Map()` преобразовывает значение переменной из одного диапазона в другой.

Т.е. значение переменной `value`, равное `fromLow`, будет преобразовано в число `toLow`, а значение `fromHigh` - в `toHigh`. Все промежуточные значения `value` масштабируются относительно нового диапазона [`toLow`; `toHigh`].

Функция не ограничивает значение переменной заданными пределами, поскольку ее значения вне указанного диапазона иногда несут полезную информацию.

Для ограничения диапазона необходимо использовать функцию `constrain()` либо до, либо после функции `map()`.

Обратите внимание, что нижние пределы указываемых диапазонов (`fromLow`, `toLow`) численно могут быть больше верхних пределов (`fromHigh`, `toHigh`).

Цель:

- Повторение и закрепление пройденного материала.
- Разобраться с математическими методами.
- Использование температурного датчика, сервопривода.
- Встроенная функция `Map()`.

Задачи: Использование сервопривода и температурного датчика в практических схемах.

Техническое оборудование:

- резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт,
- подтягивающий резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт),
- светодиод (желтый, зеленый, красный),
- штыревые кабели (с разъемами BLS1 или провода «папа-папа»),
- беспаячная макетная плата.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	Повторение		
10 мин.	Изложение нового материала	Новый материал не предусмотрен. Уточнение и расширение знаний по материалам предыдущих занятий	Подготовка презентации	
50 мин.	Практическая работа	Сбор схемы с использованием Tinkercad и повторение через реальные компоненты Подключить к схеме термометра из предыдущего занятия сервопривод. Нарисовать температурную шкалу от 15 до 30 градусов Цельсия. Написать программу, чтобы сервопривод указывать измеренную температуру. Для успешного решения задачи желательно освоить встроенную функцию Arduino Map() .		Приложение схемы к занятиям 10 и 9
10 мин.	Уборка рабочих мест			



ЗАНЯТИЕ № 12

Тема: Подведение итогов

Цель занятия: Повторение и закрепление пройденного материала. Использование температурного датчика, сервопривода, светодиодов, потенциометров и фоторезисторов.

Задачи занятия: Использование сервопривода и температурного датчика в практических схемах.

Техническое оборудование: резистор (200-300 Ом, для ограничения тока через светодиод) на 0,25 Вт, резистор 1кОм – 10 кОм (для подтяжки питания) на 0,25 Вт), светодиод (желтый, зеленый, красный), — штыревые кабели, безопасная макетная плата.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		-
5 мин.	Повторение пройденного материала	Повторение		
10 мин.	Изложение нового материала	Новый материал не предусмотрен. Уточнение и расширение знаний по материалам предыдущих занятий	Подготовка презентации	
50 мин.	Практическая работа	1. Сбор схемы с использованием TinkerCad, и повторение через реальные компоненты 2. Подключить к Arduino сервопривод и потенциометр. Управлять углом поворота сервопривода с помощью потенциометра. 3. Добавить в схему три светодиода. Если сервопривод находится в секторе от 0 до 45 градусов, горит первый светодиод, если от 45 до 135, то 1-й и 2-й, если от 135 до 180, третий. 4. Собрать комнатный термометр предыдущем занятии. Добавить три светодиода, как в задании 3.		Приложение схемы
10 мин.	Уборка рабочих мест			

ЗАНЯТИЕ № 13

Тема: Проектная работа. Обсуждение и выбор проектов для реализации. Формирование проектных групп

На этом занятии формируются проектные команды, выбираются проекты, которые будут реализованы для «умного дома».

Примерный план занятия:

1. Сообщить о цели занятия, 5- минут;
2. Рассказать о том, какие проекты могут быть реализованы учащимися до конца учебного года. В качестве примера, продемонстрировать ранее выполненные проекты (15-20 минут).

Лампа с голосовым и дистанционным управлением	http://www.instructables.com/id/Bluetooth-Voice-Controlled-Moving-Lamp/
Автоматизированный полив растений	https://www.coursera.org/learn/roboty-arduino?authMode=login
Дистанционно управляемый дверной замок	http://www.instructables.com/id/Arduino-Android-Based-Bluetooth-Control-Password-P/

3. Учащиеся разбиваются на проектные команды по 3-4 человека. Выбирают/придумывают проект. Готовят краткую презентацию своей идеи. (30 минут).

4. Каждая проектная команда представляет проект перед всей группой. После представления проекта возможно его обсуждение, вопросы-ответы. (40-50 минут).

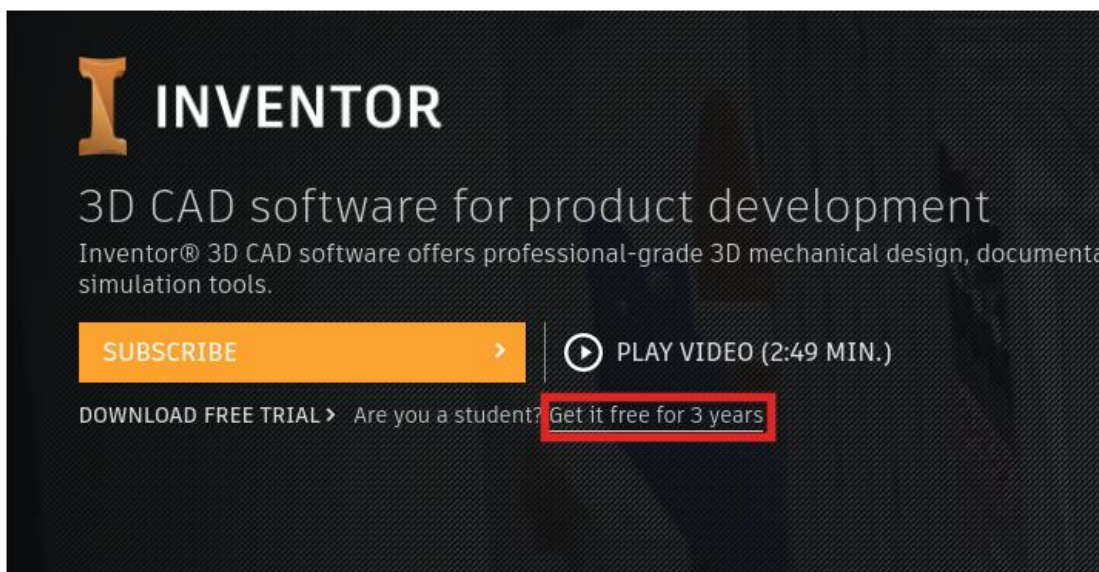


ЗАНЯТИЕ № 14

Тема: Изучение основ 3D моделирования (Inventor)

Цель: познакомить учащихся с основами 3D моделирования. Обучить основным навыкам и приемам работы в Autodesk Inventor. Предполагается, что полученные навыки учащиеся смогут применить в создании своих проектов.

Программный продукт для учащихся и преподавателя можно использовать, получив на сайте разработчика бесплатную лицензию на 3 года (<https://www.autodesk.com/products/inventor/>).



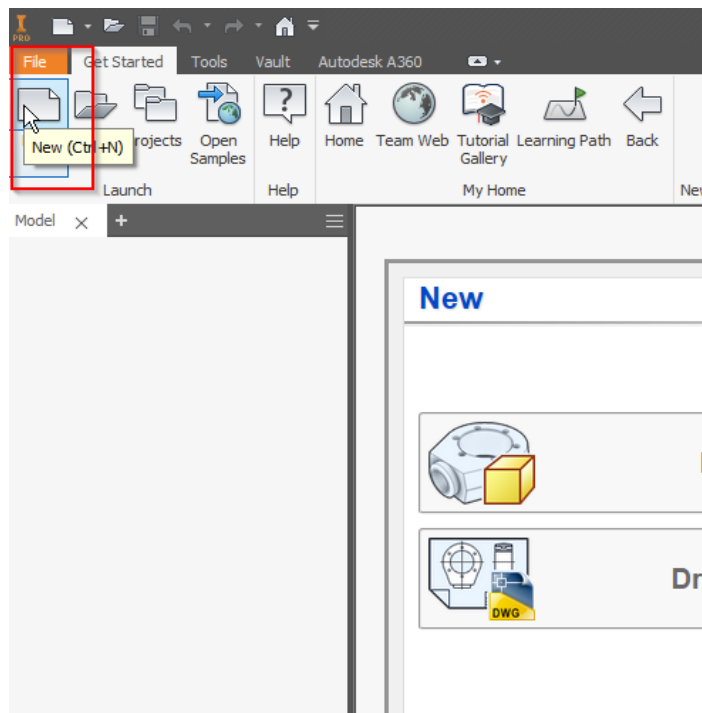
Теоретическая часть:

1. Что такое проекция. Основные проекции: Вид спереди, вид сверху и слева. Получить проекции цилиндра, шара, параллелепипеда и т.п.
2. Два основных способа получения объемных тел из плоских фигур. Вращение, выдавливание (параллельный перенос).

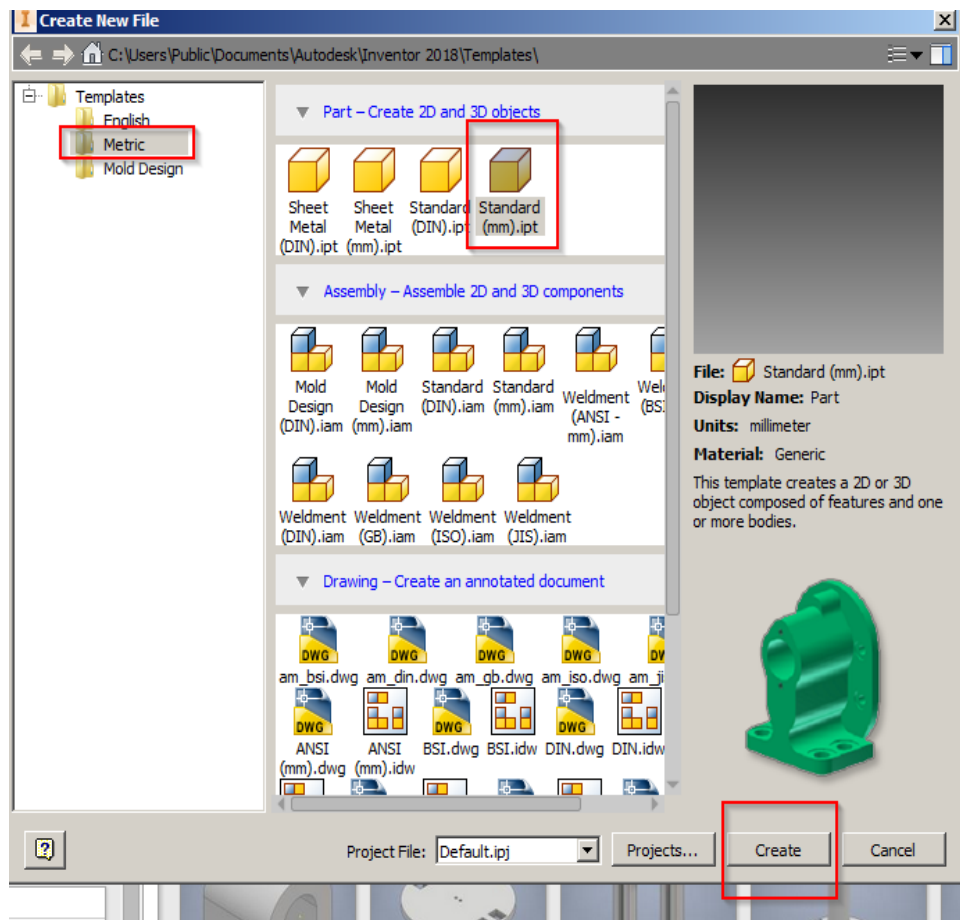
Практическая работа в Autodesk Inventor.

Создать простую модель колеса.

Создание новой детали. New.

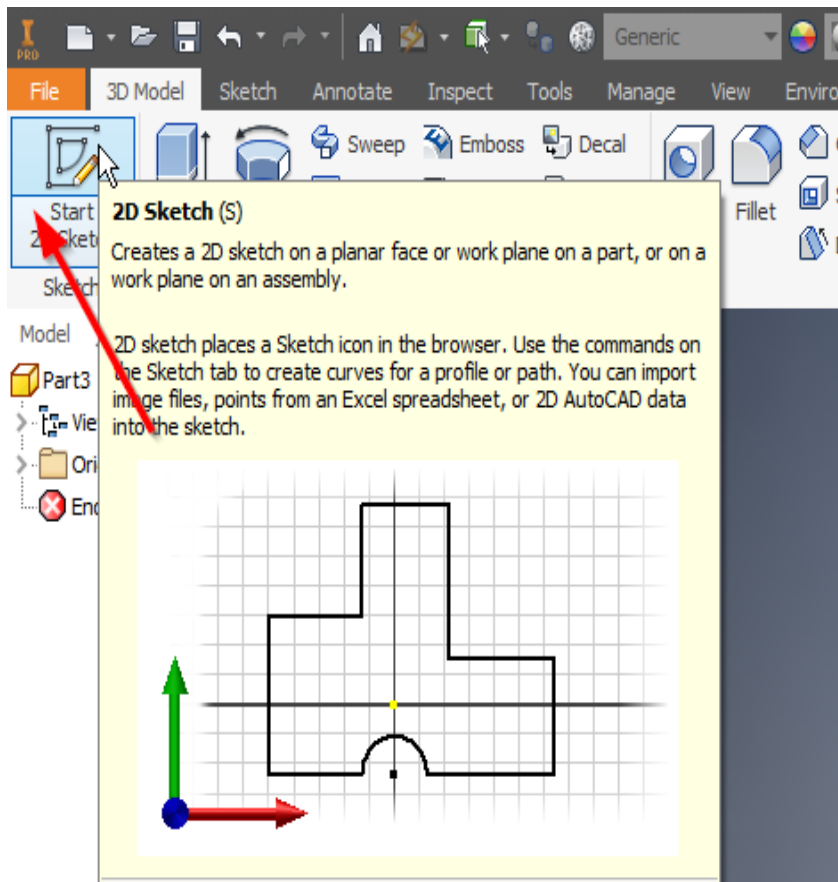


Выбрать Metric и Standard.

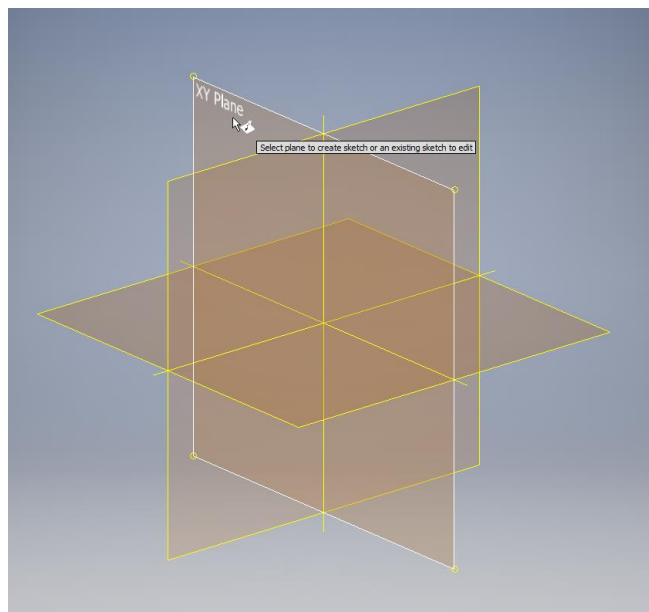




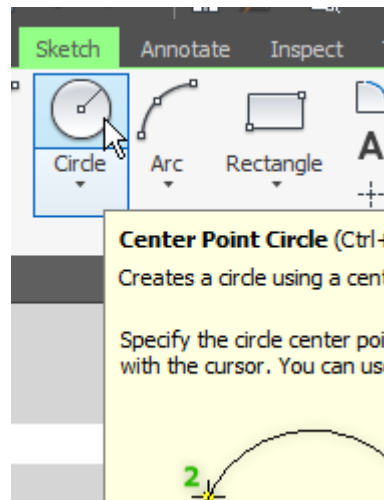
Создать 2d эскиз.



Выбрать плоскость XY

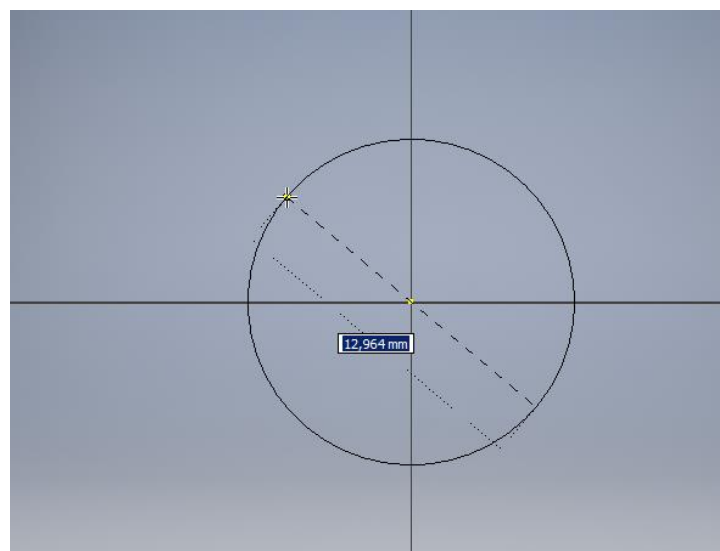
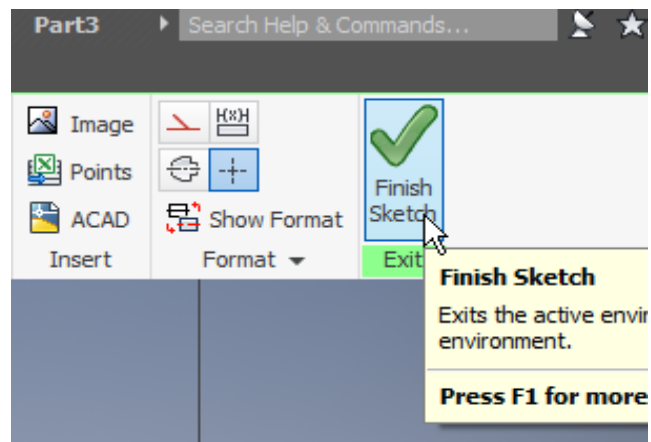


Выбрать инструмент окружность



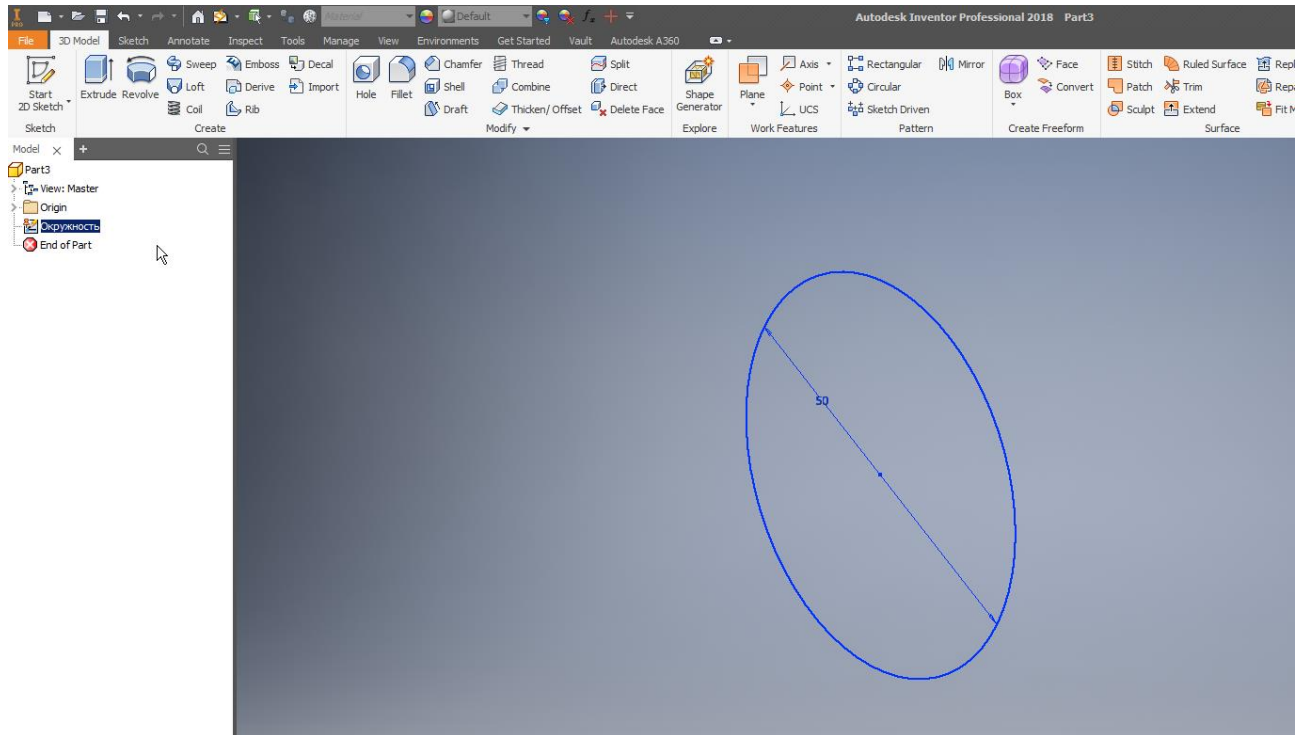
Нажать левую кнопку мыши в центре окружности и протянуть, затем отпустить и ввести диаметр окружности в мм (для этого примера 50мм).

Завершить 2D скетч

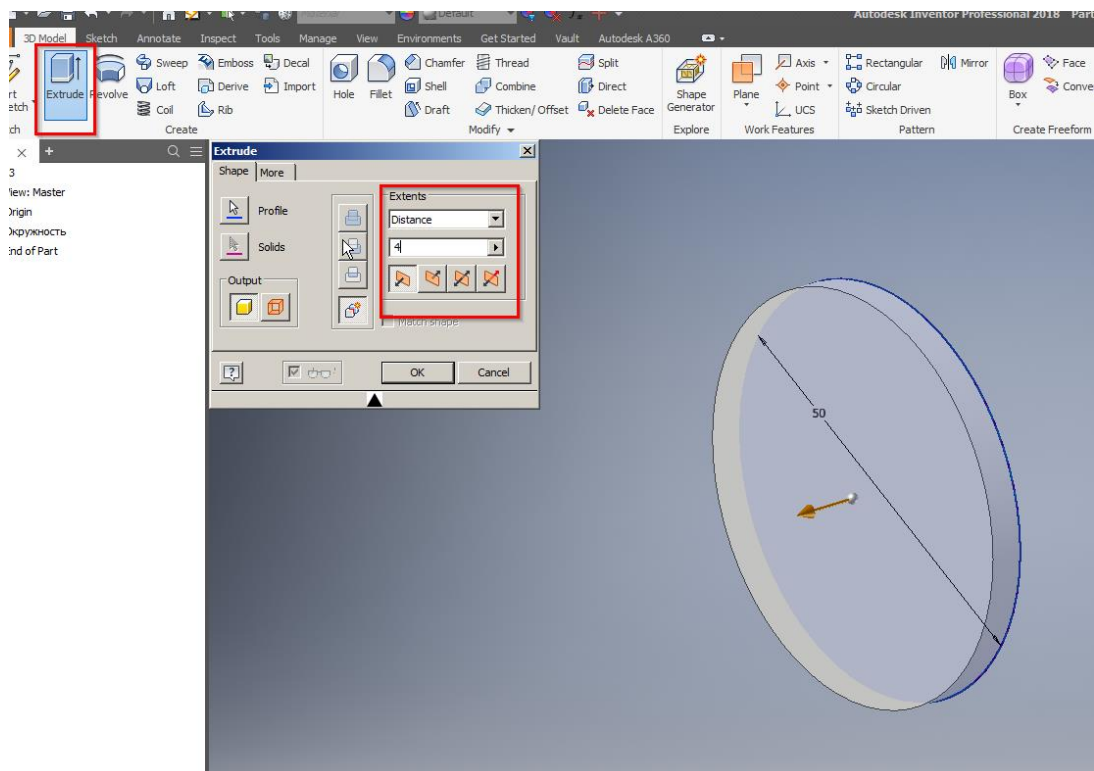




Скетч готов. Слева формируется дерево построения.



Сделать выдавливание (extrude) на 4 мм.

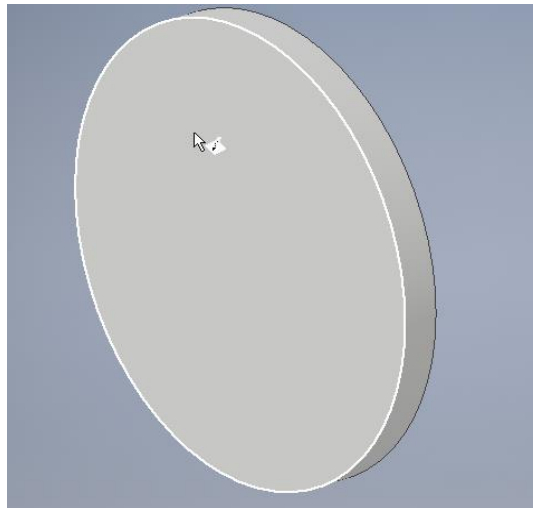


Заготовка для колеса готова.

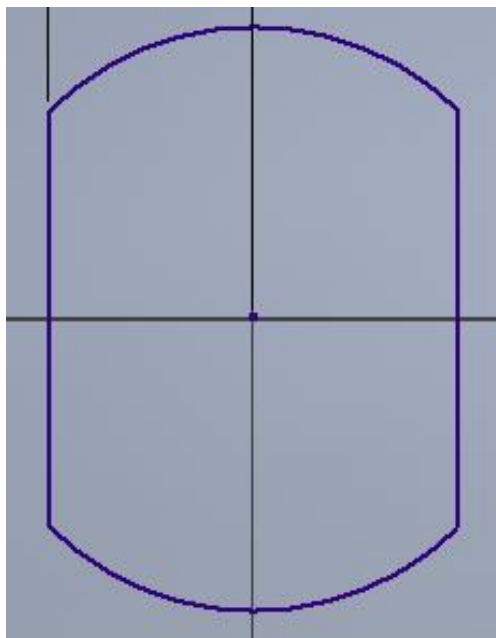
ЗАНЯТИЕ № 15

Тема: Продолжаем моделировать колесо

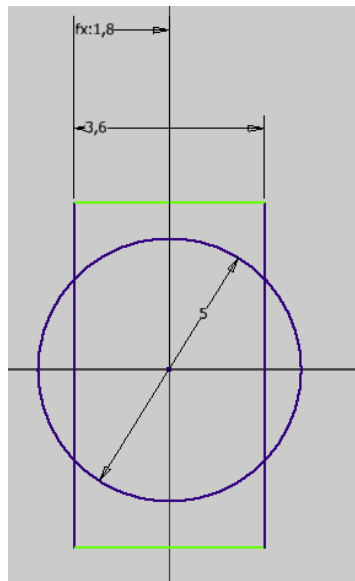
Далее необходимо сделать отверстие под вал. Сделать еще один 2D скетч. В качестве плоскости для построения выбрать плоскость колеса.



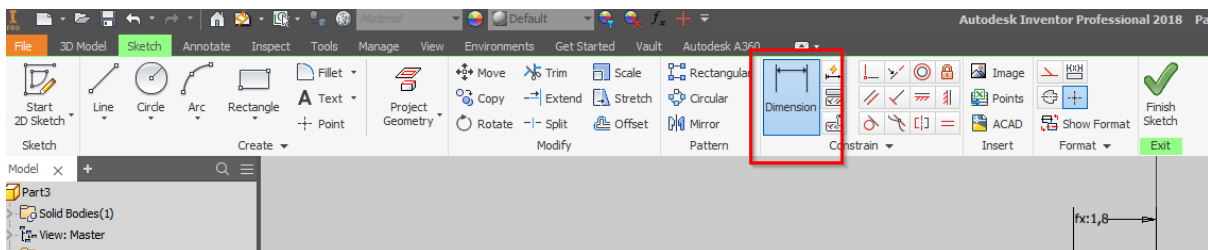
Форма вала двигателя



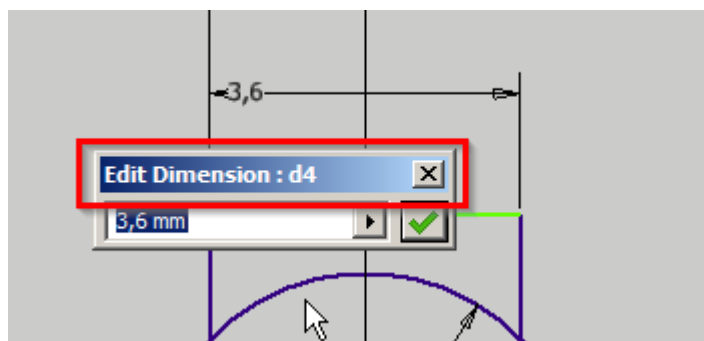
Сначала нужно начертить окружность диаметром 5 мм, затем прямоугольник с одной из сторон 3,6 мм.



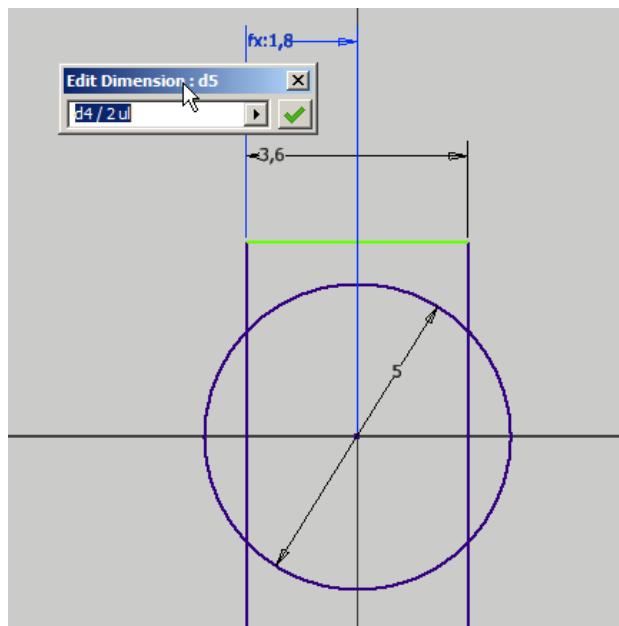
Размеры задаются с помощью инструмента Dimensions.



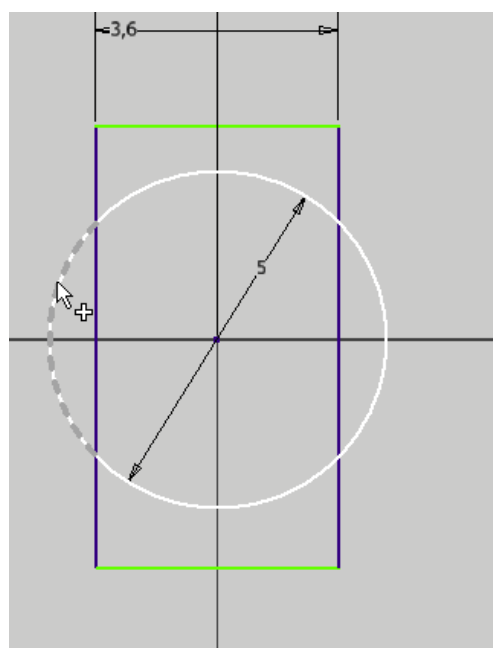
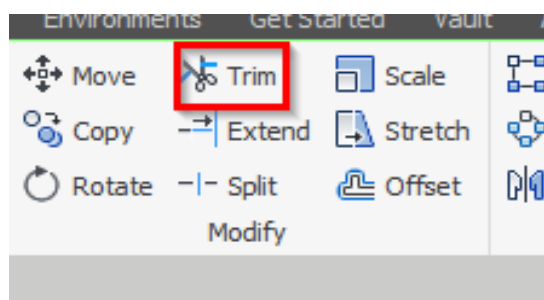
Каждый из заданных размеров имеет свое буквенное обозначение



Чтобы отцентрировать прямоугольник вертикально необходимо задать размер от центра координат до одной из сторон прямоугольника как половину длины все стороны. Т.е. в нашем случае $d4/2$.

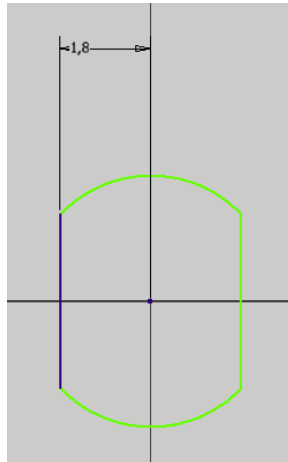


После удаляем ненужные отрезки с помощью инструмента Trim

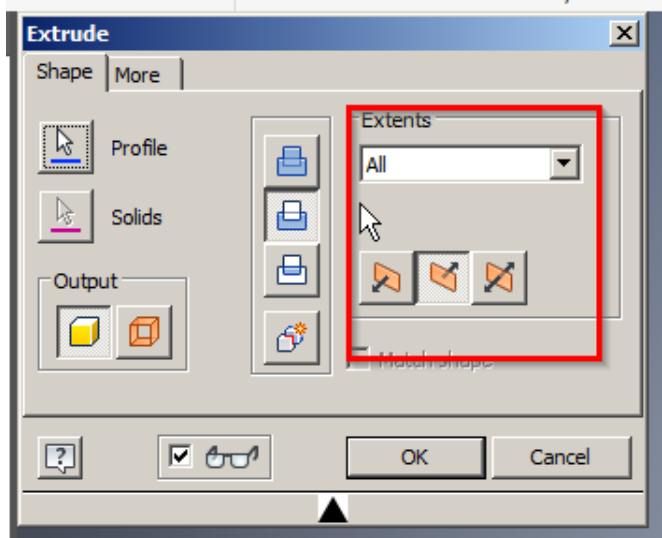




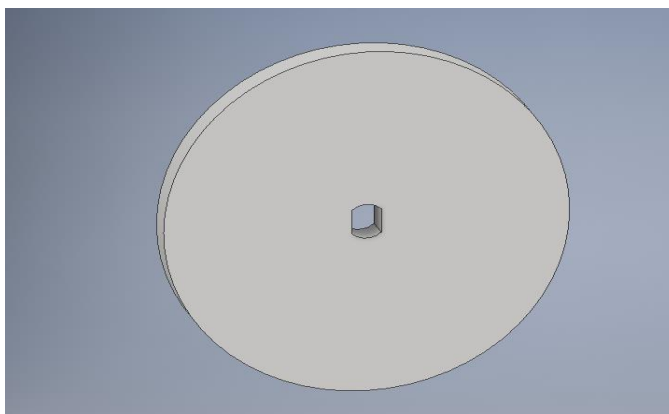
Получилась необходимая форма



Завершить 2D скетч и сделать Extrude на всю толщину изделия, выбрав соответствующее направление выдавливания.



Результат:



ЗАНЯТИЯ № 16 – № 29

Тема: Проектная работа

На этих занятиях учащиеся работают в составе проектных команд. Каждая команда работает по своему плану над своим проектом. Преподаватель консультирует их по возникающим вопросам и направляет деятельность.

Обобщенный план работ:

- Описание основной функции (назначения) разрабатываемого устройства.
- Составление функциональной схемы устройства. Отдельные относительно самостоятельные блоки устройства и их взаимосвязи.
- Определение полного перечня компонентов (в том числе, электронных) устройства.
- Распределение работ по участникам проекта.
- Создание чертежа устройства (общего вида и отдельных частей).
- Составление принципиальной схемы устройства.
- Изучение отдельных компонентов устройства (при необходимости).
- Сборка и отладка самостоятельных блоков устройства.
- Интеграционное тестирование блоков устройства.
- Сборка и отладка устройства.



ЗАНЯТИЕ № 30

Тема: Интеграция устройств в единый центр управления. Протоколы, используемые в системах «умный дом». Центр управления «умным домом»

Цель занятия:

- Разобраться с протоколами, используемыми в системах «умный дом»;
- Научиться устанавливать центра управления openHAB.

Задачи занятия:

- Подключение микрокомпьютера raspberry pi;
- Установка операционной системы на raspberry pi.

Техническое оборудование:

- Микрокомпьютер raspberry pi;
- Источник питания для raspberry pi;
- Кабель HDMI-HDMI;
- Монитор с видеовходом HDMI;
- Клавиатура и мышь с интерфейсом USB;
- Карта памяти формата SD;
- Компьютер с ОС Windows.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		
5 мин.	Повторение пройденного материала	1. Что такое «Умный дом»? 2. Состав «Умного дома». 3. Что такое Arduino?		
20 мин.	Изложение нового материала	Основные протоколы, используемые в системах и их особенности. Центры управления «Умным домом». Преимущества системы openHAB. Raspberry pi. Установка Raspbian	Подготовка презентации	
40 мин.	Практическая работа	Подключение микрокомпьютера raspberry pi.		

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
		Установка операционной системы. Изучение состава ОС Raspbian		
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: 1. Распространенные протоколы, используемые в системах «умный дом»; 2. Основные плюсы системы openHAB» 3. Особенности raspberry pi		

Для объединения устройств в систему, необходимо выбрать язык общения и научить каждое устройство этому языку. Для устройств такой язык называется – протокол. Выбор протокола и способ связи выбирается исходя из особенностей размещения элементов системы, объемов передаваемой информации и требований к надежности системы.

Протокол связи – это то, как сигналы передаются от одного устройства к другому, чтобы вызвать действие – включить свет или разблокировать умный дверной замок. Понимание различий между протоколами имеет важное значение при выборе устройств для создания экосистемы умного дома, поскольку каждый протокол имеет свои преимущества и недостатки в зависимости от применения.

Коммуникационные протоколы разрабатываются консорциумами, организациями или ассоциациями, где участники обсуждают и согласовывают детали «языка», описывают и создают образцы аппаратных средств для демонстрации возможностей.

Ниже представлены наиболее популярные протоколы связи для общения интеллектуальных устройств в умном доме.

Infrared Один из самых простых и надежных протоколов, как правило, предлагает одностороннюю связь с коротким диапазоном действия. Не самый популярный протокол для создания умных устройств, но некоторые находят интересные решения.

Ethernet Быстрый и надежный проводной протокол связи с высокой пропускной способностью и низкой чувствительностью к электромагнитным помехам. В большинстве своем используется для полной домашней автоматизации.



Wi-Fi - быстрая и надежная беспроводная связь с диапазоном покрытия около 25 метров и высокой пропускной способностью. Протокол широко применяется производителями для умных устройств, которые передают большое количество данных (видео, аудио, фото), например, системы домашней безопасности или умные колонки.

Bluetooth - протокол беспроводной связи с коротким диапазоном около 10 метров. Часто используется в телефонах, наушниках и колонках. Адаптивная система скачкообразной перестройки частоты обнаруживает существующие сигналы, такие как Wi-Fi, и согласовывает карту каналов для Bluetooth устройств, с тем, чтобы свести помехи к минимуму. Это выбор номер один для умных бытовых устройств – мультиварки, пылесосы, кофемашины и подобные.

Thread - беспроводной протокол связи, разработанный группой компаний, включая Nest, Samsung, QUALCOMM и OSRAM. Разработан, чтобы устройства продолжали общаться в случае отсутствия Wi-Fi. Один из самых перспективных протоколов для экосистемы умного дома.

ZigBee - представляет собой беспроводной протокол, который работает в ячеистой сети. В такой сети каждое устройство усиливает и расширяет сеть, передавая данные другому устройству. На “языке” протокола ZigBee общаются умные лампочки Philips Hue, умные розетки и переключатели WeMo.

Z-Wave - подобно ZigBee, Z-Wave представляет собой протокол ячеистой сети с открытым исходным кодом. С технической точки зрения, главное различие между ними состоит в пропускной способности данных – ZigBee примерно в 6 раз быстрее, чем Z-Wave. Однако, последний требует меньше энергии для покрытия такого же диапазона, как ZigBee.

В рамках курса для построения систем используется сеть Wi-Fi.

Основными плюсами этой сети является:

- Почти в каждой квартире и доме уже есть Wi-Fi;
- Низкая цена оборудования;
- Большое количество материалов по созданию и настройке систем в Сети.

В качестве центра управления умным домом мы будем использовать систему **openHAB**.

openHAB переводится как «*Open Home Automation Bus*». Это значит, что он нацелен на создание универсальной платформы для объединения всей домашней «умной» техники в единую систему управления.

Что такое «единая платформа домашней автоматизации»?

Направление умных домов развивается уже давно. За это время появилось множество различных поставщиков «умного» оборудования (освещение, RGB подсветки, термостаты, приводы для штор/жалюзи, разнообразные датчики и т.п.) Все они работают на разных протоколах (проводных и беспроводных).

В результате, пользователь, желающий автоматизировать свое жилище, должен выбирать между продуктами от различных производителей. В конечном итоге ему приходится пользоваться разными системами (пульты, приложения) для контроля разных устройств. Вдобавок, задача создания полноценного интернета вещей (когда устройства могут быть связаны в единую цепь и управляться все вместе в результате каких-либо условий и событий) становится очень сложной.

Простейший пример:

Необходимо, чтобы при включении верхнего света в комнате, когда за окном уже темно, шторы автоматически закрывались. И, наоборот, при открытии штор (когда они уже наполовину открыты) выключался верхний свет, и включалась подсветка.

При условии, что привод штор — это ZWave, верхний свет — KNX, а подсветка — несколько цветных лампочек Phillips Hue, надо как-то научить их «разговаривать» между собой. Не забывайте также про условие относительно темного времени суток...

Чем поможет openHAB?

openHAB реализует единую шину, т.е. позволяет объединить все устройства с разными протоколами в единую сеть, абстрагируя пользователя от каждого конкретного протокола. Таким образом, можно пользоваться единственным средством управления (скажем, приложением на смартфоне) и реализовать сколь угодно сложную логику взаимосвязи между устройствами.

Что отличает проект openHAB от аналогов?

Во-первых, он развивается уже довольно давно и уже сейчас готов к полноценному использованию. По своему личному опыту (впервые я начал работать с этой системой около года назад) могу сказать, что его освоение занимает совсем немного времени. И после этого он работает действительно так, как заявлено в описании.

Сейчас разработчики трудятся уже над вторым поколением платформы, основанной на специализированном фреймворке Eclipse SmartHome.

Во-вторых, это проект с открытым исходным кодом. А это значит, что весь код создается программистами со всего мира, которые заинтересованы в теме интернета вещей и создания единой системы умного дома. Этот код доступен в репозитории GitHub.

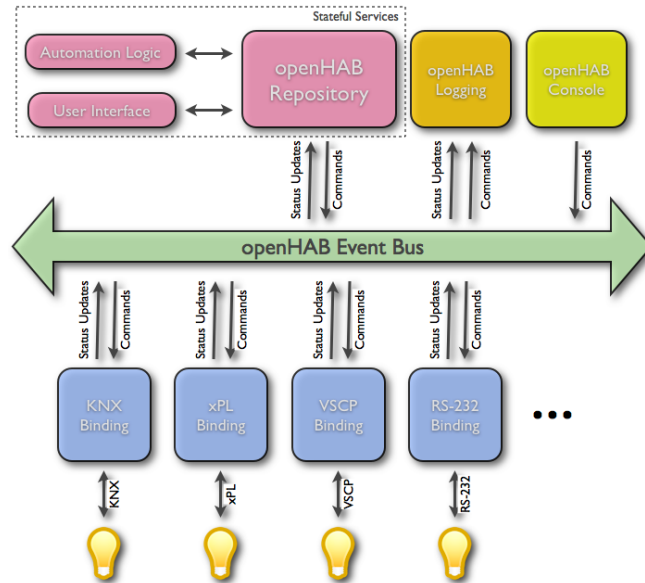
В-третьих, в результате открытости уже сейчас openHAB поддерживает около 50 (*пятидесяти!*) различных протоколов «умных» устройств. Среди них ZWave, KNX, EnOcean, системы мультимедиа типа Sonos, кинотеатр XBMC, Samsung SmartTV и многие-многие другие. Это, в свою очередь, дает возможность пользователю выбирать устройства для своего дома по самым разнообразным параметрам, практически не ограничиваясь возможностями всей платформы.

Как это работает?



openHAB — это специальный сервер, который может работать на любом компьютере под управлением любой ОС. Вся установка заключается в распаковывании дистрибутива сервера и установке *Java машины*. Далее начинается процесс настройки и созидания.

openHAB — это очень гибкий конструктор. Основанный на технологии *OSGi*, он позволяет конфигурировать каждый отдельный плагин (*Binding*) «налету», не перезагружая весь сервер.



Плагин (биндинг) в openHAB — это то самое связующее звено между конкретным протоколом устройства и единой системой openHAB. Пользователь может добавить в свою инсталляцию openHAB только те плагины, которые нужны именно ему. Например, если у вас есть только ZWave и Phillips Hue, то вам понадобятся только два биндинга.

В качестве сервера для openHAB мы будем использовать одноплатный компьютер Raspberry pi.

Что такое raspberry pi?

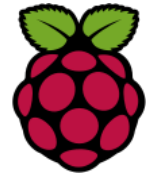


Raspberry Pi — одноплатный компьютер, то есть различные части компьютера, которые обычно располагаются на отдельных платах, здесь



представлены на одной. К тому же эта плата имеет относительно небольшой размер — примерно 8,5*5,5 см.

В названии продукта объединены **Raspberry** – малина и **Pi** – число Пи. Изображение малины стало логотипом проекта.



Характеристики:

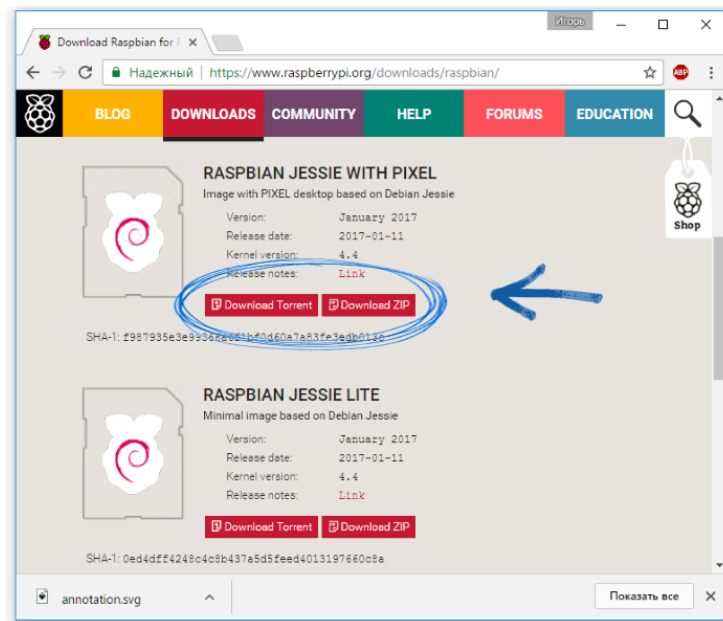
- Процессор: 64-битный 4-ядерный ARM Cortex-A53 с тактовой частотой 1,2 ГГц на однокристальном чипе Broadcom BCM2837;
- оперативная память: 1ГБ LPDDR2 SDRAM;
- цифровой видеовыход: HDMI;
- композитный выход: 3,5 мм (4 pin);
- USB порты: USB 2.0×4;
- сеть: Wi-Fi 802.11n, 10/100 Мб RJ45 Ethernet;
- Bluetooth: Bluetooth 4.1, Bluetooth Low Energy;
- разъем дисплея: Display Serial Interface (DSI);
- разъем видеочамеры: MIPI Camera Serial Interface (CSI-2);
- карта памяти: SD;
- порты ввода-вывода: 40;
- габариты: 85x56x17 мм.

Производитель рекомендует 6 операционных систем:

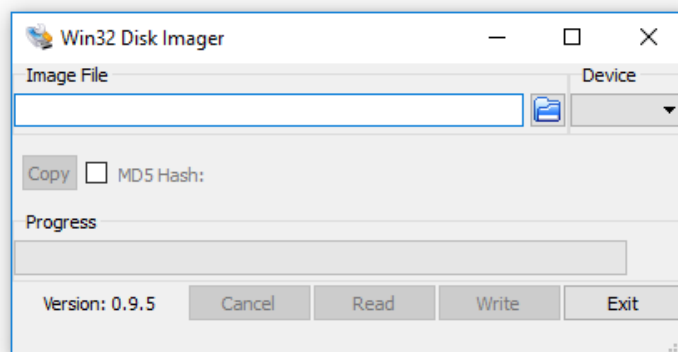
- Raspbian, основанная на Debian Wheezy;
- Pidora, основанная на Fedora Remix;
- Arch Linux;
- Risc OS;
- ARM-версия OpenELEC, XMBC медиа-центра;
- RaspBMC, XMBC медиа-центр.

Запись образа Raspbian на карту microSD:

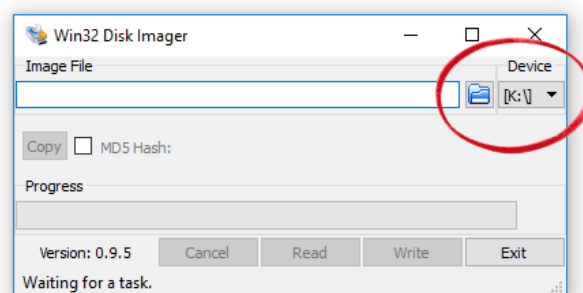
1. Скачайте образ на официальном сайте Raspberry Pi.
(<https://www.raspberrypi.org/downloads/>)



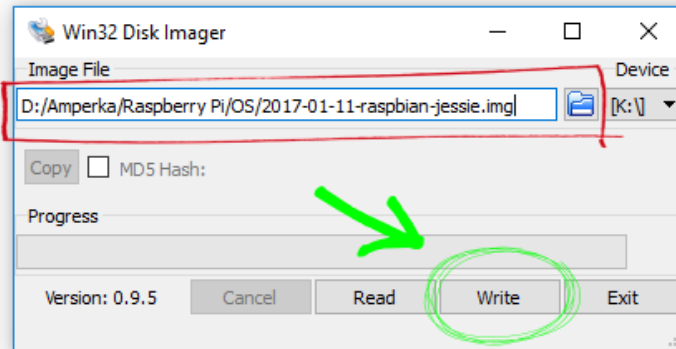
2. Распакуйте скачанный ZIP-архив.
3. Скачайте и установите на компьютер утилиту Win32DiskImager для записи образов на SD. (http://wiki.amperka.ru/_media/rpi:installation:win32diskimager-0.9.5-install.zip)
4. Запустите программу Win32DiskImager.



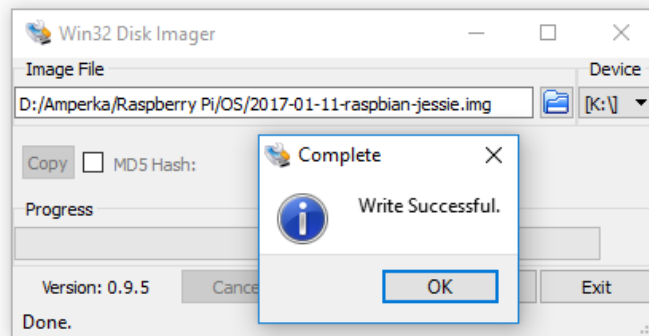
5. Подключите карту microSD с помощью картридера к компьютеру и откройте её во вкладке «Device»



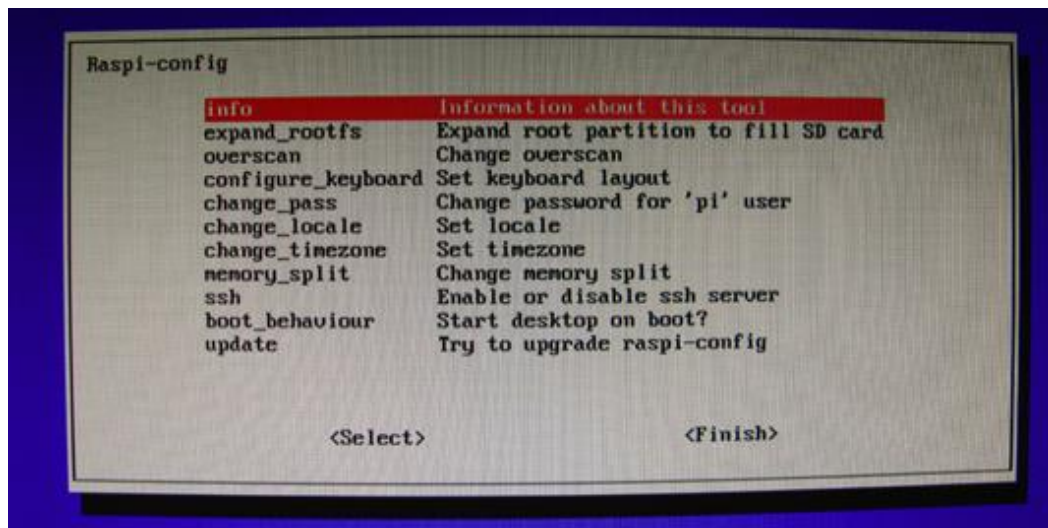
6. Укажите в поле «Image File» путь к скаченному образу «Raspbian» и нажмите кнопку «write»



7. Начнётся процесс загрузки образа. По завершению загрузки увидите сообщение Write Successful

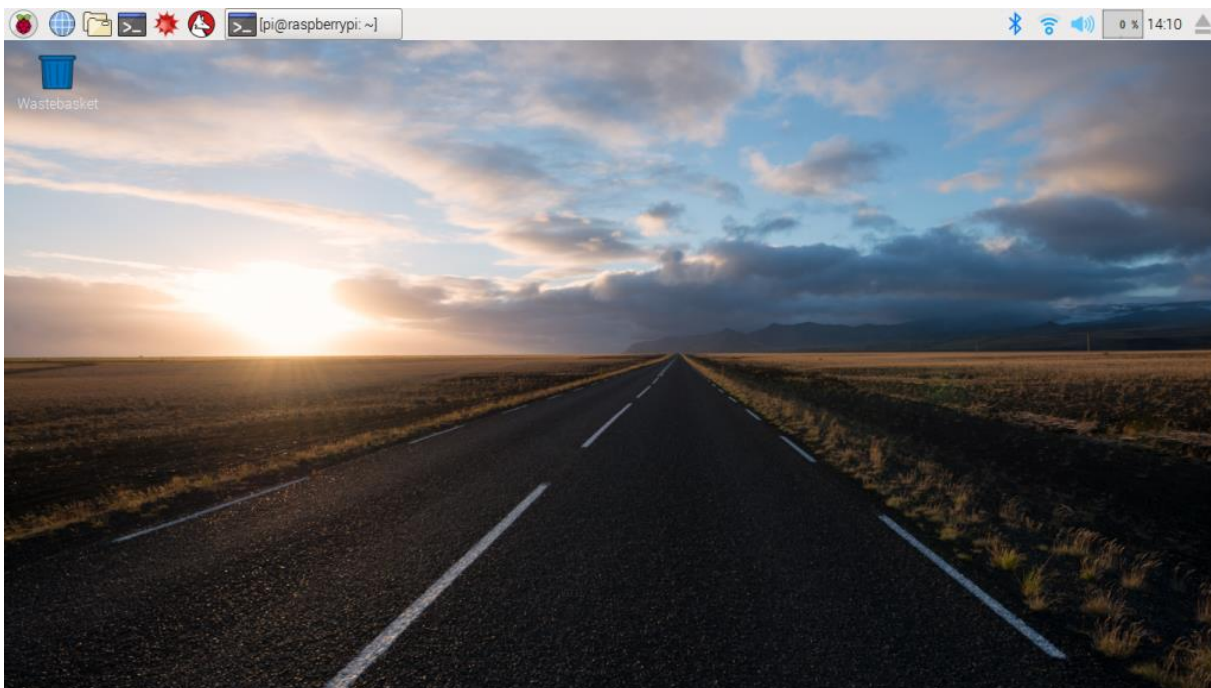


8. При первой загрузке автоматически будет запущена программа настройки системы raspi_config:





- a. Выполняем команду **expand_rootfs**, которая расширит root раздел на всю SD-карту.
- b. Входим в раздел **configure_keyboard** и устанавливаем наиболее подходящий тип клавиатуры. Logitech Cordless Desktop
- c. Входим в раздел **change_pass** устанавливаем новый пароль для пользователя pi (обратите внимание, что вводимые символы не отображаются вообще, даже в виде звездочек!).
- d. Устанавливаем часовой пояс (**set_timezone**). Например, если Вы живете в Москве, необходимо найти пункт Europe, а в нем — Moscow.
- e. **Memory_split** устанавливаем в соответствии с собственным желанием, рекомендуется отвести под video – 32Mb, если планируется пользоваться графическим интерфейсом.
- f. Обязательно активируем **ssh**!
- g. Если хотим, чтобы при загрузке Raspbian автоматически запускалась графическая среда – активируем опцию **boot_behaviour**.
9. Наконец, открывается рабочий стол системы.



ЗАНЯТИЕ № 31

Тема: Интеграция устройств в единый центр управления. Установка openHAB на raspberry pi. Что такое MQTT? Установка Mosquitt

Актуальность темы: Для работы серверной части системы OpenHAB2 необходимо наличие Java на вашем микрокомпьютере.

Установка Java

Для Raspberry Pi в консоли набираем следующую команду и выполняем дальнейшие инструкции по установке:

```
sudo apt-get install oracle-java8-jdk
```

Для завершения необходимо будет нажать Enter. Теперь необходимо заново обновить установочные пакеты в системе:

```
sudo apt-get update
```

И уже после этого запускаем установку Java с помощью команды:

```
sudo apt-get install oracle-java8-installer
```

Установка займет 10-15 минут в зависимости от скорости интернета. Во время установки необходимо будет нажать Y, потом появится розовое окно, в котором необходимо будет нажать OK и YES. Отлично, вы установили Java на Raspberry Pi.

После установки вводим следующую команду чтобы проверить версию Java на вашем Pi:

```
pi@orangepi-zero:~$ java -version
```

Если Java успешно установилась, то в ответ вы получите следующее сообщение:

```
Java version "1.8.0_151"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
```

```
Java HotSpot(TM) Client VM (build 25.151-b12, mixed mode)
```

Установка OpenHAB2

Все необходимые приготовления готовы, и мы можем начать установку OpenHAB2. Нам необходимо добавить ключ репозитория openHAB2 в систему управления установочными пакетами:

```
wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' | sudo apt-key add
```

Вводим пароль пользователя pi, в ответ получаем "OK" - ключ добавлен. Теперь необходимо разрешить программе Apt использовать протокол HTTPS:

```
sudo apt-get install apt-transport-https
```

Вводим следующий код в консоли, чтобы добавить в систему репозиторий с последней версией OpenHAB2:

```
echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' | sudo tee /etc/apt/sources.list.d/openhab2.list
```

Теперь обязательно еще раз необходимо обновить список доступных установочных пакетов:

```
sudo apt-get update
```



Наконец-то запускаем установку самого OpenHAB:

```
sudo apt-get install openhab2=2.1.0-1
```

Система достаточно долго будет скачивать и распаковывать установочные файлы OpenHAB2. После установки вы увидите сообщение о том как запускать OpenHAB, проверять его статус и добавить автозагрузку. Теперь добавим к системе дополнительные пакеты - вводим в консоли следующие команды:

```
sudo apt-get install openhab2-addons=2.1.0-1
```

```
sudo apt-get install openhab2-addons-legacy=2.1.0-1
```

Настроим автозагрузку OpenHAB2 при старте операционной системы:

```
sudo /bin/systemctl daemon-reload
```

```
sudo /bin/systemctl enable openhab2.service
```

Еще необходимо разрешить доступ пользователю `openhab` к паре системных папок, вводим в терминале:

```
sudo chown -hR openhab:openhab /etc/init.d/openhab2
```

```
sudo chown -hR openhab:openhab /usr/share/openhab2
```

Также с помощью следующих команд в консоли вы можете запустить, остановить и перезагрузить OpenHAB, а также посмотреть его статус.

```
sudo /bin/systemctl start openhab2.service
```

```
sudo /bin/systemctl stop openhab2.service
```

```
sudo /bin/systemctl restart openhab2.service
```

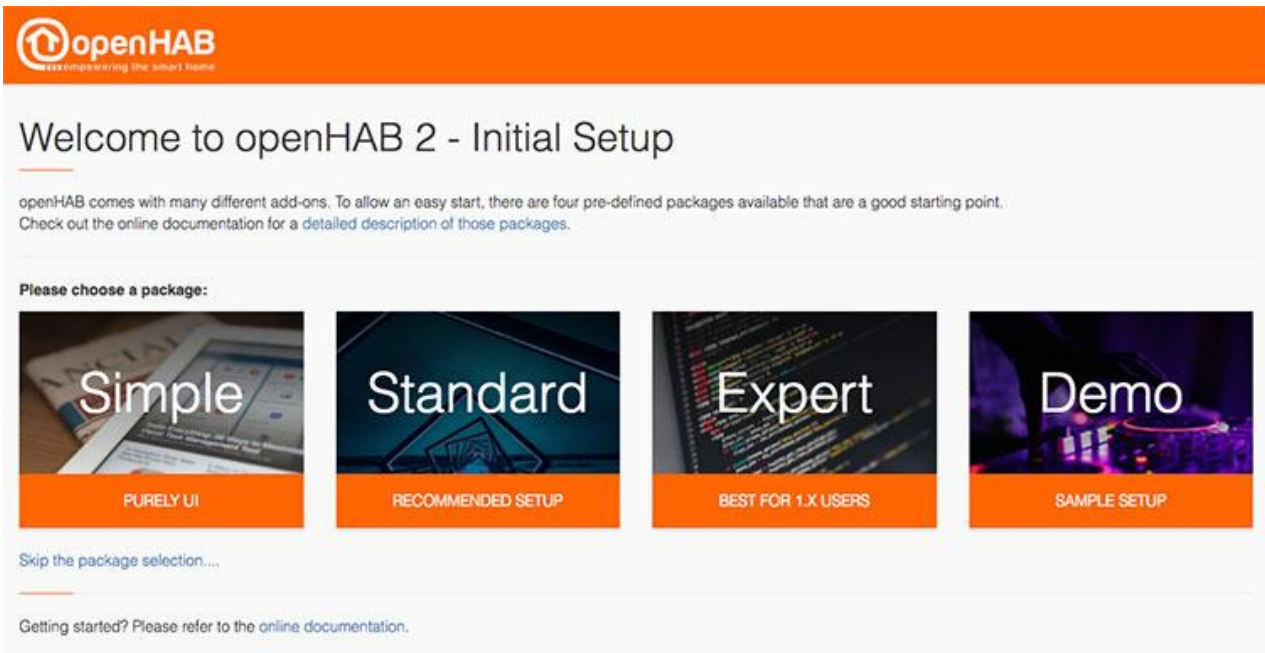
```
sudo /bin/systemctl status openhab2.service
```

После успешной установки и запуска команды `status` в консоли вы увидите примерно вот такое сообщение:

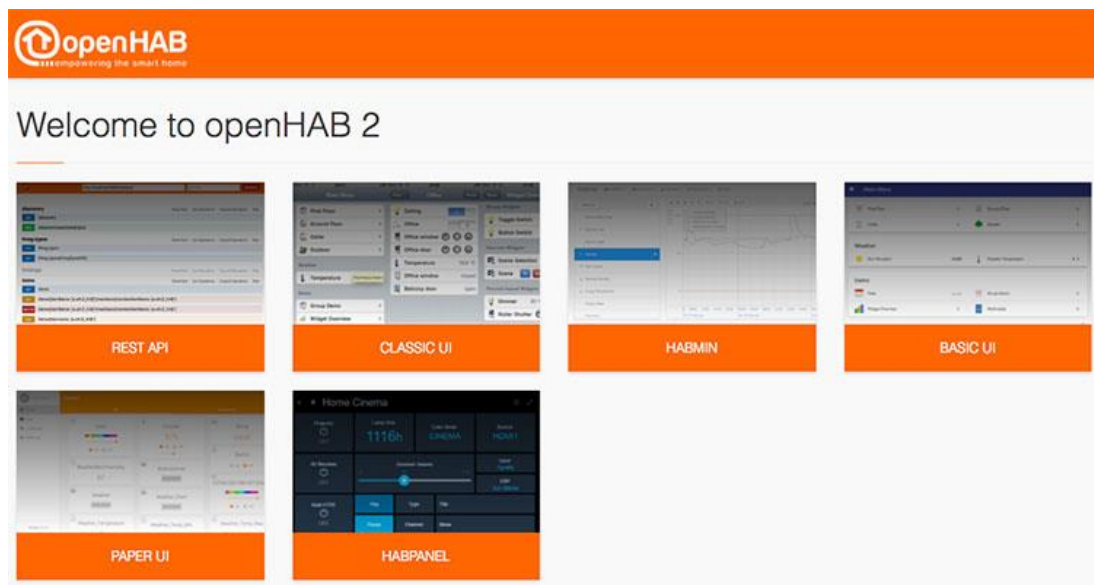
```
● openhab2.service - openHAB 2 - empowering the smart home
   Loaded: loaded (/usr/lib/systemd/system/openhab2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-12-12 22:47:54 MSK; 10s ago
     Docs: http://docs.openhab.org
           https://community.openhab.org
   Main PID: 15132 (karaf)
   CGroup: /system.slice/openhab2.service
           └─15132 /bin/bash /usr/share/openhab2/runtime/bin/karaf server
           └─15300 /usr/bin/java -Dopenhab.home=/usr/share/openhab2 -Dopen
```

Обратите внимание, что, несмотря на то, что мы видим статус "active", серверу OpenHAB требуется примерно 5-10 минут для полной загрузки и подготовки системы к работе. Такая долгая загрузка происходит практически каждый раз при перезагрузке вашего микрокомпьютера, но зато система очень надёжна.

Для завершения настройки вводим в адресной строке браузера вашего компьютера `http://ipдресвашегоPi:8080`. Вы увидите экран с возможными установками системы:



Выбираем вариант «Expert» - он даст нам самые широкие варианты настроек и максимальный функционал. Если на этой странице у вас ничего не отображается, надо подождать еще несколько минут и обновить страницу. После того как вы выбрали вариант установки, начнется сама установка, необходимо будет подождать ещё 5-10 минут. В результате вы увидите следующий экран:



Что такое MQTT?

MQTT или Message Queue Telemetry Transport – это легкий, компактный и открытый протокол обмена данными созданный для передачи данных на удалённых локациях, где требуется небольшой размер кода и есть ограничения



по пропускной способности канала. Вышеперечисленные достоинства позволяют применять его в системах M2M (Машинно-Машинное взаимодействие) и IoT (Промышленный Интернет вещей).

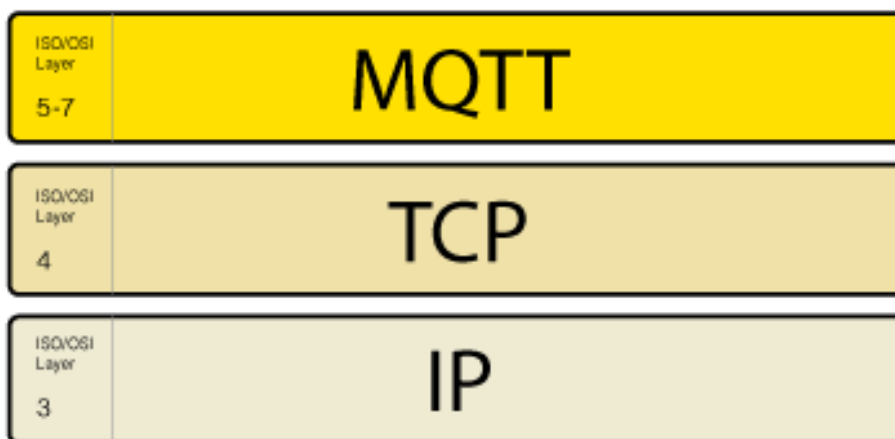
Также существует версия протокола **MQTT-SN** (MQTT for Sensor Networks), ранее известная как MQTT-S, которая предназначена для встраиваемых беспроводных устройств без поддержки TCP/IP сетей, например, Zigbee.

Особенности протокола MQTT

Основные особенности протокола MQTT:

- Асинхронный протокол
- Компактные сообщения
- Работа в условиях нестабильной связи на линии передачи данных
- Поддержка нескольких уровней качества обслуживания (QoS)
- Легкая интеграция новых устройств

Протокол MQTT работает на прикладном уровне поверх TCP/IP и использует по умолчанию 1883 порт (8883 при подключении через SSL).



Обмен сообщениями в протоколе MQTT осуществляется между клиентом (client), который может быть издателем или подписчиком (publisher/subscriber) сообщений, и брокером (broker) сообщений (например, Mosquitto MQTT).

Издатель отправляет данные на MQTT брокер, указывая в сообщении определенную тему, топик (topic). Подписчики могут получать разные данные от множества издателей в зависимости от подписки на соответствующие топики.

Устройства MQTT используют определенные типы сообщений для взаимодействия с брокером, ниже представлены основные:

- Connect – установить соединение с брокером;
- Disconnect – разорвать соединение с брокером;
- Publish – опубликовать данные в топик на брокере;
- Subscribe – подписаться на топик на брокере;
- Unsubscribe – отписаться от топика.



Схема простого взаимодействия между подписчиком, издателем и брокером



Семантика топиков

Топики представляют собой символы с кодировкой UTF-8. Иерархическая структура топиков имеет формат «дерева», что упрощает их организацию и доступ к данным. Топики состоят из одного или нескольких уровней, которые разделены между собой символом «/».

Пример топика, в который датчик температуры, расположенный в спальном комнате, публикует данные брокеру:

/home/living-space/living-room1/temperature

Подписчик может так же получать данные сразу с нескольких топиков, для этого существуют **wildcard**. Они бывают двух типов: одноуровневые и многоуровневые. Для более простого понимания рассмотрим в примерах каждый из них:

— Одноуровневый wildcard. Для его использования применяется символ «+»

К примеру, нам необходимо получить данные о температуры во всех спальнях комнатах:

/home/living-space+/temperature

В результате получаем данные с топиков:

/home/living-space/living-room1/temperature

/home/living-space/living-room2/temperature

/home/living-space/living-room3/temperature

— Многоуровневый wildcard. Для его использования применяется символ «#»

К примеру, чтобы получить данные с различных датчиков всех спален в доме:

/home/living-space/#

В результате получаем данные с топиков:

/home/living-space/living-room1/temperature

/home/living-space/living-room1/light1

/home/living-space/living-room1/light2

/home/living-space/living-room1/humidity

/home/living-space/living-room2/temperature

/home/living-space/living-room2/light1

...



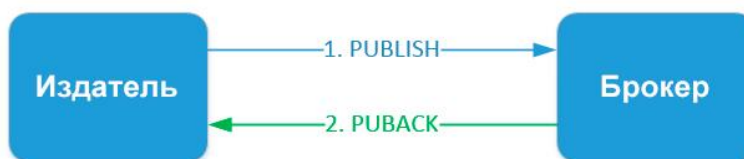
Качество обслуживания в протоколе MQTT (QoS)

MQTT поддерживает три уровня качества обслуживания (QoS) при передаче сообщений.

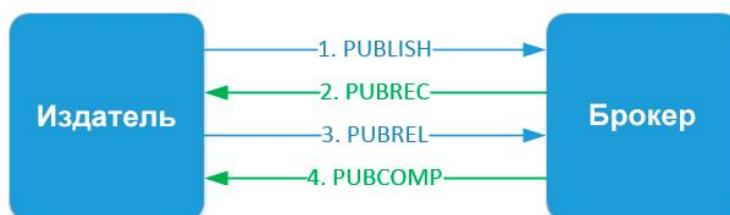
QoS 0 At most once. На этом уровне издатель один раз отправляет сообщение брокеру и не ждет подтверждения от него, то есть отправил и забыл.



QoS 1 At least once. Этот уровень гарантирует, что сообщение точно будет доставлено брокеру, но есть вероятность дублирования сообщений от издателя. После получения дубликата сообщения, брокер снова рассылает это сообщение подписчикам, а издателю снова отправляет подтверждение о получении сообщения. Если издатель не получил PUBACK сообщения от брокера, он повторно отправляет этот пакет, при этом в DUP устанавливается «1».



QoS 2 Exactly once. На этом уровне гарантируется доставка сообщений подписчику и исключается возможное дублирование отправленных сообщений.



Издатель отправляет сообщение брокеру. В этом сообщении указывается уникальный Packet ID, QoS=2 и DUP=0. Издатель хранит сообщение не подтвержденным, пока не получит от брокера ответ PUBREC. Брокер отвечает сообщением PUBREC, в котором содержится тот же Packet ID. После его получения издатель отправляет PUBREL с тем же Packet ID. До того, как брокер получит PUBREL, он должен хранить копию сообщения у себя. После получения PUBREL он удаляет копию сообщения и отправляет издателю сообщение PUBCOMP о том, что транзакция завершена.

Установка MQTT брокера Mosquitto на Raspberry Pi

Для одноплатных компьютеров на Linux существует несколько MQTT брокеров. Одним из самых популярных является Mosquitto. Сегодня мы с вами

разберемся, как поставить серверную и клиентскую, а также как проверить работу протокола MQTT.

Обновляем систему

Первым делом перед началом установки нам необходимо обновить систему. Подключаемся к нашему Pi по SSH и вводим:

```
sudo apt-get update
sudo apt-get upgrade
```

Подробнее посмотреть, как подключаться к Pi по SSH, и что потом можно делать вы можете в нашей статье.

Установка брокера и клиента “mosquitto”

К сожалению, обычные библиотеки “apt-get” Raspberry Pi (Orange Pi) не содержат последнюю версию Mosquitto software. Поэтому перед установкой нам нужно будет обновить библиотеки. Если на сервере будет установлена не последняя версия брокера, то могут появляться ошибки в процессе работы.

Добавляем ключ и обновляем репозиторий. Вводим эти строчки по очереди. Иногда надо будет вводить пароль от вашего пользователя, и нажимать Y для согласия с установкой.

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
cd /etc/apt/sources.list.d/
sudo wget http://repo.mosquitto.org/debian/mosquitto-jessie.list
sudo apt-get update
```

Устанавливаем MQTT брокер (сервер):

```
sudo apt-get install mosquitto
```

Устанавливаем MQTT клиент:

```
sudo apt-get install mosquitto mosquitto-clients
```

После установки mosquitto сервер будет автоматически запускаться при загрузке системы.

Останавливаем сервер для настройки

После установки сервер MQTT сразу запускается. Нам необходимо остановить его, чтобы произвести последующую настройку.

```
sudo /etc/init.d/mosquitto stop
```

Настройка

Открываем файл настроек

```
sudo nano /etc/mosquitto/mosquitto.conf
```

и заменяем его содержимое на следующее:

```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example
pid_file /var/run/mosquitto.pid
```

```
persistence true
```



```
persistence_location /var/lib/mosquitto/
```

```
log_dest topic
```

```
log_type error
```

```
log_type warning
```

```
log_type notice
```

```
log_type information
```

```
connection_messages true
```

```
log_timestamp true
```

```
include_dir /etc/mosquitto/conf.d
```

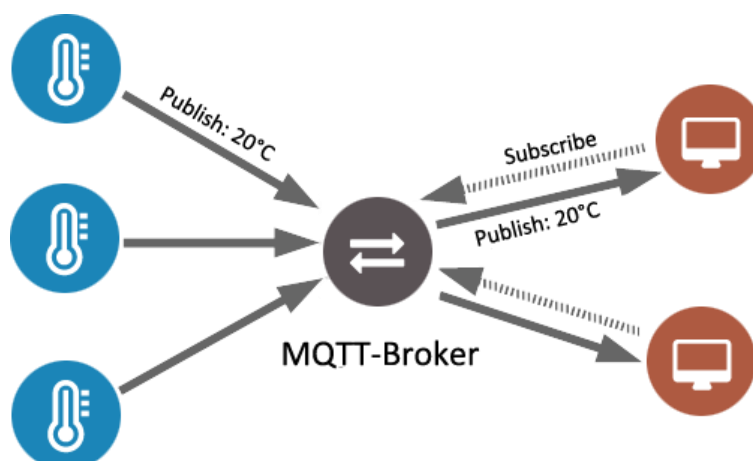
Сохраняем, закрываем программу nano и выходим обратно в терминал (ctrl+o, Enter, ctrl+x).

Запускаем сервер MQTT

```
sudo /etc/init.d/mosquitto start
```

Проверяем работу MQTT брокера

Для проверки работы нашего MQTT брокера нам понадобится MQTT клиента. Это могут быть смартфоны с MQTT клиентами, которые подключены к той же Wi-Fi сети. Но мы воспользуемся более простым способом - подключимся к серверу 2 терминалами параллельно.



Нам необходимо, чтобы первый терминал подписался на топик hello/world, второй терминал отправил в этот топик сообщение и первый терминал его получил.

Итак, в первом терминале, в котором вы делали все настройки, подписываемся на канал hello/world:



```
mosquitto_sub -d -t hello/world
```

Открываем второе окно терминала, по SSH подключаемся к нашему серверу и отправляем сообщение в канал hello/world:

```
mosquitto_pub -d -t hello/world -m "Hello from Terminal window 2!"
```

В результате в первом окне появится сообщение:

```
Hello from Terminal window 2!
```

```
Oct 03 14:28:43 orangepizero systemd[1]: Starting LSB: mosquitto MQTT v3.1 message broker...
Oct 03 14:28:45 orangepizero mosquitto[479]: * Starting network daemon: mosquitto
Oct 03 14:28:45 orangepizero mosquitto[479]:   ...done.
Oct 03 14:28:45 orangepizero systemd[1]: Started LSB: mosquitto MQTT v3.1 message broker.
pi@orangepizero:~$ mosquitto_sub -d -t hello/world
Client mosqsub/9144-orangepize sending CONNECT
Client mosqsub/9144-orangepize received CONNACK
Client mosqsub/9144-orangepize sending SUBSCRIBE (Mid: 1, Topic: hello/world, QoS: 0)
Client mosqsub/9144-orangepize received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/9144-orangepize received PUBLISH (d0, q0, r0, m0, 'hello/world', ... (29 bytes))
Hello from Terminal window 2!
```

Наш MQTT брокер работает!

Статус сервера MQTT

Проверить статус вашего MQTT брокера вы всегда можете через терминал:

```
sudo /etc/init.d/mosquitto status
```

После этой команды в терминале будет показана подробная информация о состоянии вашего сервера.

```
pi@orangepizero:~$ sudo /etc/init.d/mosquitto status
[sudo] password for pi:
● mosquitto.service - LSB: mosquitto MQTT v3.1 message broker
   Loaded: loaded (/etc/init.d/mosquitto; bad; vendor preset: enabled)
   Active: active (running) since Tue 2017-10-03 14:28:45 MSK; 1min 12s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 479 ExecStart=/etc/init.d/mosquitto start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/mosquitto.service
           └─517 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Oct 03 14:28:43 orangepizero systemd[1]: Starting LSB: mosquitto MQTT v3.1 message broker...
Oct 03 14:28:45 orangepizero mosquitto[479]: * Starting network daemon: mosquitto
Oct 03 14:28:45 orangepizero mosquitto[479]:   ...done.
Oct 03 14:28:45 orangepizero systemd[1]: Started LSB: mosquitto MQTT v3.1 message broker.
```

Настройка пароля для MQTT

Для тестов и маленьких проектов вы можете использовать подключение к MQTT серверу без пароля, но в больших системах для повышения уровня безопасности необходимо добавить логин и пароль. Это делается достаточно просто.



Для того чтобы клиенты могли подключиться к вашему брокеру только по логину и паролю, необходимо создать конфигурационный файл, который будет содержать имя пользователя и зашифрованный пароль.

Вводим в консоли:

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd <username>
```

Вместо `<username>` необходимо указать имя пользователя, которому будет разрешен доступ, например, `mqtusr`. После выполнения этой команды пользователь `mqtusr` будет добавлен в систему.

Дальше необходимо будет ввести ваш пароль пользователя `pi`, а потом 2 раза пароль для доступа пользователя `mqtusr` к `mqtt` брокеру. В качестве второго пароля не вводите пароль от пользователя `pi`, лучше придумайте новый.

Теперь необходимо поменять настройки доступа к системе, для этого необходимо отредактировать файл `/etc/mosquitto/conf.d/default.conf` на вашем `pi`. Можно использовать для этого любой текстовый редактор, например, `nano`, в консоли вводим:

```
sudo nano /etc/mosquitto/conf.d/default.conf
```

В открывшейся файл добавляем 2 строчки:

```
allow_anonymous false  
password_file /etc/mosquitto/passwd
```

Сохраняем и закрываем файл. Для этого в `nano` нажимаем `ctrl+x`, выбираем `Y` и нажимает `Enter`. Как вы видите, в этом конфиге мы прописали, в каком файле хранятся пользователи и пароли для подключения к нашему MQTT брокеру, а также мы запретили подключение к серверу всем пользователям без логина и пароля.

Теперь необходимо перезапустить MQTT сервер, для этого вводим в консоли:

```
sudo systemctl restart mosquitto.
```

Цель занятия:

- Разобраться с установкой `openHAB`;
- Изучить протокол MQTT;
- Научиться устанавливать MQTT брокера `Mosquitto`.

Задачи занятия:

- Установка центра управления `openHAB`;
- Установка и проверка MQTT брокера `Mosquitto`.

Техническое оборудование:

- Микрокомпьютер `raspberrypi`;

- Источник питания для raspberry pi;
- Кабель HDMI-HDMI;
- Монитор со видеовходом HDMI;
- Клавиатура и мышь с интерфейсом US;
- Карта памяти формата SD;
- Компьютер с ОС Windows.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		
5 мин.	Повторение пройденного материала	1. Что такое openHAB? 2. Распространенные протоколы, используемые в системах “Умный дом”? 3. Что такое raspberry pi?		
15 мин.	Изложение нового материала	Основные особенности установки системы openHAB. Протокол MQTT. Особенности, области применения. Особенности установки брокера Mosquitto.	Подготовка презентации	
45 мин.	Практическая работа	Установка системы openHAB. Установка MQTT брокера Mosquitto.		
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: 1. Какие пакеты необходимо установить для работы openHAB. 2. Особенности протокола MQTT. 3. Есть ли контроль доставки сообщений в MQTT?		



ЗАНЯТИЕ № 32

Тема: Интеграция устройств в единый центр управления. Что такое ESP8266?

Цель занятия:

- Изучить микроконтроллер ESP8266;
- Научиться писать программы для ESP8266.

Задачи занятия:

- Подключить микроконтроллер ESP8266 ESP-01;
- Создать проект на основе ESP-01 для управления светодиодом.

Техническое оборудование:

- Макетная плата;
- Регулятор напряжения LM1117;
- Конвертор USB-to-TTL;
- Светодиод;
- ESP8266 ESP-01.

Технологическая карта занятия

Время	Этап занятия	Содержание этапа	Методические рекомендации для преподавателя	Дидактические материалы
5 мин.	Организационная часть	Проверка численности группы, оценка состояния рабочих мест, проведение инструктажа по технике безопасности		
5 мин.	Повторение пройденного материала	1. Какие пакеты необходимо установить для работы openHAB? 2. Особенности протокола MQTT. 3. Есть ли контроль доставки сообщений в MQTT?		
15 мин.	Изложение нового материала	Новый микроконтроллер ESP8266. Сравнение ESP8266 и Arduino	Подготовка презентации	
45 мин.	Практическая работа	Подключение ESP8266 ESP-01 через USB-to-TTL. Создание проекта управления светодиодом через браузер.		
10 мин.	Уборка рабочих мест			
10 мин.	Закрепление пройденного материала	Вопросы для закрепления темы: 1. Различие между ESP8266 и Arduino 2. Особенности программирования ESP-01		

Описание ESP8266

ESP8266 – микроконтроллер с интерфейсом Wi-Fi, который имеет возможность исполнять программы из флеш-памяти. Устройство было выпущено в 2014 году китайской фирмой Espressif и практически сразу же стало популярным.



Контроллер недорогой, обладает небольшим количеством внешних элементов и имеет следующие технические параметры:

- Поддерживает Wi-Fi протоколы 802.11 b/g/n с WEP, WPA, WPA2;
- Обладает 14 портами ввода и вывода, SPI, I2C, UART, 10-бит АЦП;
- Поддерживает внешнюю память до 16 МБ;
- Необходимое питание от 2,2 до 3,6 В, потребляемый ток до 300 мА в зависимости от выбранного режима.

Важной особенностью является отсутствие пользовательской энергонезависимой памяти на кристалле. Программа выполняется от внешней SPI ПЗУ при помощи динамической загрузки необходимых элементов программы. Доступ к внутренней периферии можно получить не из документации, а из API набора библиотек. Производителем указывается приблизительное количество ОЗУ – 50 кБ.

Особенности платы ESP8266:

- Удобное подключение к компьютеру – через USB кабель, питание от него же;
- Наличие встроенного преобразователя напряжения 3,3В;
- Наличие 4 Мб флеш-памяти;
- Встроенные кнопки для перезагрузки и перепрошивки;
- Все порты выведены на плату на две гребенки с шагом 2,5 мм.

Сферы применения модуля ESP8266

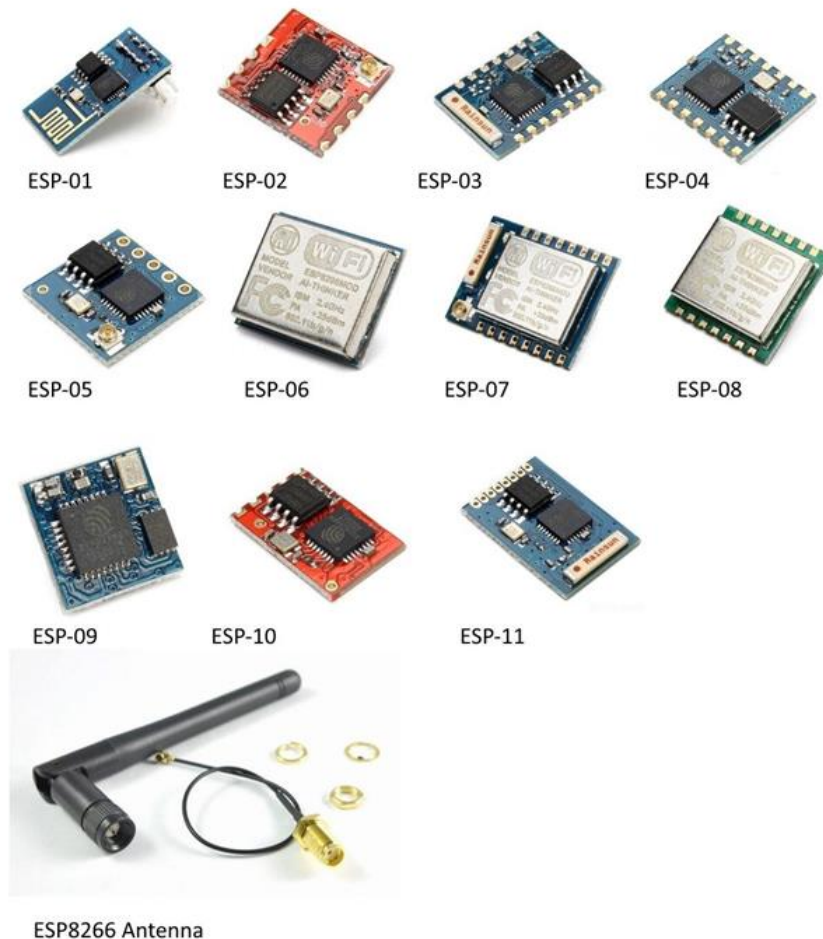
- Автоматизация;



- Различные системы для умного дома: беспроводное управление, беспроводные розетки, управление температурой, дополнение к сигнализационным системам;
- Мобильная электроника;
- ID метки;
- Детские игрушки;
- Mesh-сети.

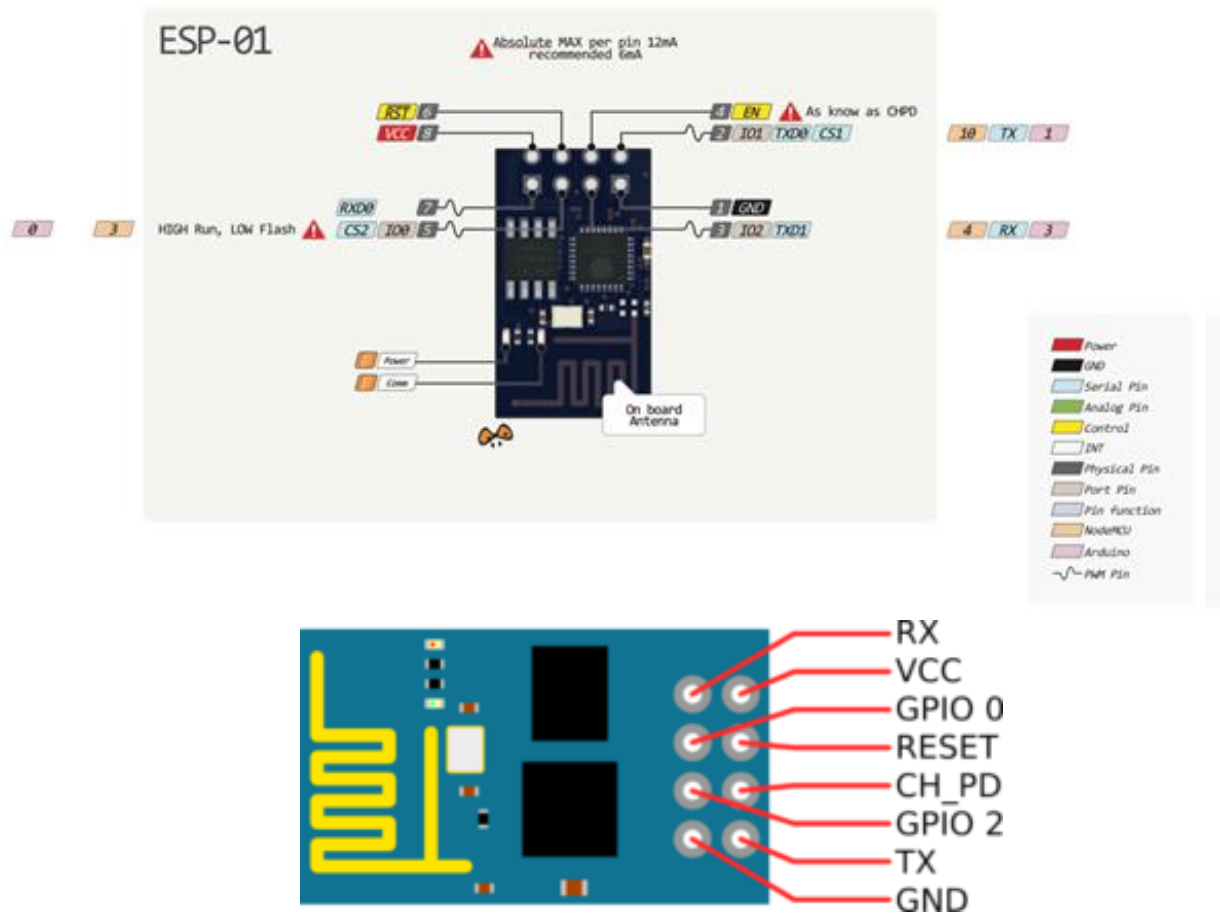
Распиновка esp8266

Существует огромное количество разновидностей модуля ESP8266. На рисунке представлены некоторые из них. Наиболее популярным вариантом является ESP 01.



Исполнение программы требуется задавать состоянием портов GPIO0, GPIO2 и GPIO15, когда заканчивается подача питания. Можно выделить 2 важных режима – когда код выполняется из UART (GPIO0 = 0, GPIO2 = 1 и GPIO15 = 0) для перепрошивки флеш-карты и когда выполняется из внешней ПЗУ (GPIO0 = 1, GPIO2 = 1 и GPIO15 = 0) в штатном режиме.

Распиновка для ESP01 изображена на картинке.



Описание контактов:

- 1 – земля, 8 – питание. По документации напряжение подается до 3,6 В. Это важно учесть при работе с Arduino, на которую обычно подают 5 В.
- 6 – RST, нужна для перезагрузки микроконтроллера при подаче на него низкого логического уровня.
- 4 — CP_PD, также используется для перевода устройства в энергосберегающий режим.
- 7 и 0 — RXD0 и TXD0, это аппаратный UART, необходимый для перепрошивки модуля.
- 2 — TXD0, к этому контакту подключается светодиод, который загорается при низком логическом уровне на GPIO1 и при передаче данных по UART.
- 5 — GPIO0, порт ввода и вывода, также позволяет перевести устройство в режим программирования (при подключении порта к низкому логическому уровню и подачи напряжения).
- 3 — GPIO2, порт ввода и вывода.

Основные отличия Arduino от ESP8266

- ESP8266 имеет больший объем флеш-памяти, при этом у ESP8266 отсутствует энергонезависимая память;
- Процессор ESP8266 быстрее, чем у Arduino;



- Наличие Wi-Fi у ESP8266;
- ESP8266 потребляет больше тока, чем для Arduino;

Программирование ESP8266 в Arduino IDE

Программный комплект разработчика esp8266 включает в себя:

- Компилятор из пакета GNU Compiler Collection.
- Библиотеки, стеки протоколов Wi-Fi, TCP/IP.
- Средство загрузки информации в программу контроллера.
- Операционная IDE.

Изначально модули ESP8266 поставляются с прошивкой от фирмы-изготовителя. С ее помощью можно управлять модулем с внешнего микроконтроллера, реализовывать работу с Wi-Fi как с модемом. Также существует множество других готовых прошивок. Некоторые из них позволяют настраивать работу модуля при помощи WEB-интерфейса.

Можно программировать из среды Arduino IDE. При ее помощи можно легко писать скетчи и загружать их в ESP8266, прошивать ESP8266, при этом не требуется сама плата Arduino. Arduino IDE поддерживает все виды модулей ESP8266.

В настоящий момент для ESP8266 можно реализовать следующие функции:

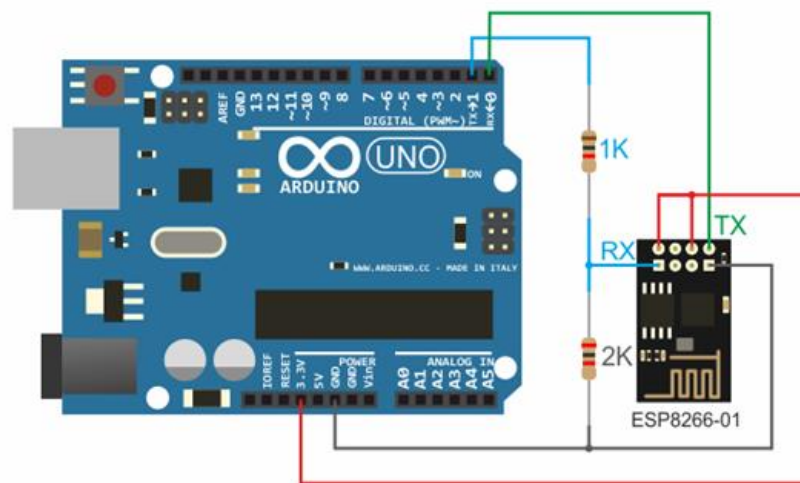
1. Основные функции языка Wiring. Управлять портами GPIO можно точно так же, как и пинами на плате Arduino: `pinMode`, `digitalRead`, `digitalWrite`, `analogWrite`. Команда `analogRead(A0)` позволяет считать значения АЦП. При помощи команды `analogWrite(pin, value)` можно подключить ШИМ на нужном выходе GPIO. При `value=0` ШИМ отключается, максимальное значение достигает константы, равной 1023. С помощью функций `attachInterrupt`, `detachInterrupt` можно выполнять прерывание на любом порте GPIO, кроме 16.
2. Тайминг и `delay`. Используя команды `millis` и `micros` можно вернуть мс и мкс, которые прошли с момента старта. `Delay` позволяет приостановить исполнение программы на нужное время. Также функция `delay(...)` позволяет поддерживать нормальную работу Wi-Fi, если в скетче присутствуют большие элементы, которые выполняются более 50 мс. `Yield()` – аналог функции `delay(0)`.
3. `Serial` и `Serial1` (UART0 и UART1). Работа `Serial` на ESP8266 аналогична работе на Arduino. Запись и чтение данных блокируют исполнение кода, если FIFO на 128 байт и программный буфер на 256 байт заполнены. Объект `Serial` пользуется аппаратным UART0, для него можно задать пины GPIO15 (TX) и GPIO13 (RX) вместо GPIO1(TX) и GPIO3(RX). Для этого после функции `Serial.begin()` нужно вызвать `Serial.swap()`. Аналогично `Serial1` использует UART1, который работает на передачу. Необходимый пин для этого GPIO2.
4. Макрос `PROGMEM`. Его работа аналогична работе в Arduino. Позволяет перемещать данные `read only` и строковые постоянные во flash-память. При этом в ESP8266 не сохраняются одинаковые константы, что приводит к дополнительной трате флеш-памяти.

5. I2C. Перед началом работы с шиной I2C выбираются шины с помощью функции `Wire.pins(int sda, int scl)`.

6. SPI, OneWire – поддерживаются полностью.

Перед подключением к Arduino важно помнить, что у ESP8266 напряжение питания не может быть выше 3,6, в то время как на плате Arduino напряжение равно 5 В. Соединить 2 микроконтроллера нужно с помощью резистивных делителей. Перед подключением модуля нужно ознакомиться с распиновкой выбранного ESP8266.

Схема подключения для ESP8266-01 представлена на рисунке.



3,3 В с Arduino – на Vcc&CH_PD на модуле ESP8266, земля с Arduino – к земле с ESP8266, 0 – TX, 1 – RX.

Для поддержки стабильной работы ESP8266 необходим источник постоянного напряжения на 3,3 В и максимальный ток 250 мА. Если питание происходит от конвертера USB-TTL, могут происходить неполадки и сбои в работе.

Работа с библиотекой Wi-Fi для ESP8266 схожа с библиотекой для обыкновенного шилда. Имеется несколько особенностей:

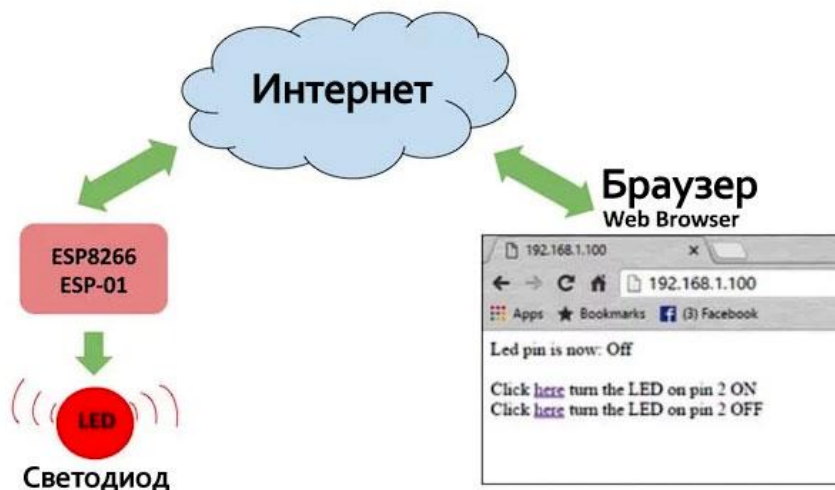
- `mode(m)` – для выбора одного из трех режимов: клиент, точка доступа или оба режима одновременно.
- `softAP(ssid)` – нужен для создания открытой точки доступа.
- `softAP(ssid, password)` – создает точку доступа с паролем, который должен состоять не менее чем из 8 знаков.
- `WiFi.macAddress(mac)` и `WiFi.softAPmacAddress(mac)` – определяет MAC адрес.
- `WiFi.localIP()` и `WiFi.softAPIP()` – определение IP адреса.
- `printDiag(Serial)`; — позволят узнать данные о диагностике.
- `WiFiUDP` – поддержка передачи и приема multicast пакета в режиме клиента.
- Работа выполняется по следующему алгоритму:



- Подключение USB-TTL к USB и к ESP.
- Запуск Arduino IDE.
- Выбрать в меню инструменты нужный порт, плату, частоту и размер flash-памяти.
- Файл — Примеры — ESP8266WiFi — WiFiWebServer.
- Записать в скетче SSID и пароль сети Wi-Fi.
- Начать компиляцию и загрузку кода.
- Дождаться окончания процесса прошивки, отсоединить GPIO0 от земли.
- Поставить скорость 115200.
- Произойдет подключение, будет записан адрес IP.
- Открыть браузер, ввести в адресной строке номер IP/gpio/1
- Посмотреть монитор порта, если к выходу GPIO2 подключен светодиод, он должен загореться.

Управление светодиодом через Веб интерфейс

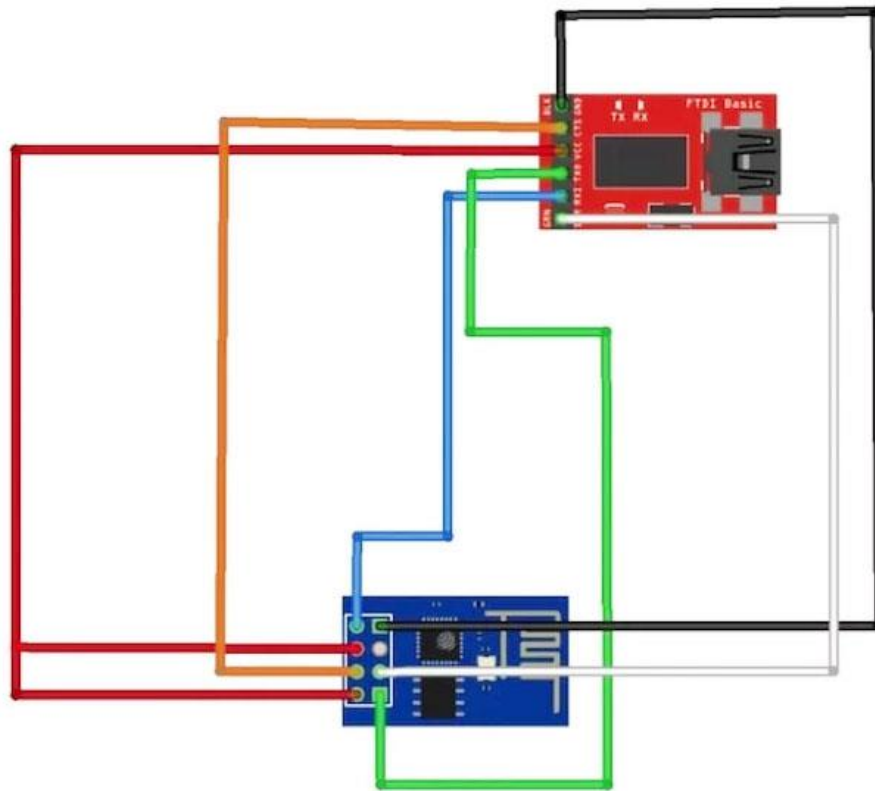
ESP8266 можно контролировать из локальной сети Wi-Fi или из Интернета (после переадресации портов). Модуль ESP-01 имеет контакты GPIO, которые могут быть запрограммированы для включения или выключения светодиода, или реле через Интернет. Модуль можно запрограммировать с помощью конвертера Arduino USB-to-TTL через последовательные контакты (RX, TX).



Мы можем использовать конвертер USB-to-TTL или использовать Arduino для программирования ESP8266. Вот три способа, которым вы можете следовать, чтобы загрузить код в ESP8266 - выберите тот, который вам подходит лучше всего. Обратитесь к диаграммам для каждого варианта и соответствующим образом настройте своё оборудование.

Подключение ESP8266 к компьютеру при помощи конвертера USB-to-TTL с использованием разъема DTR

Если вы используете конвертер USB-to-TTL с выводом DTR, загрузка будет идти гладко. Пожалуйста, имейте в виду, что серийный монитор не будет работать при этом.

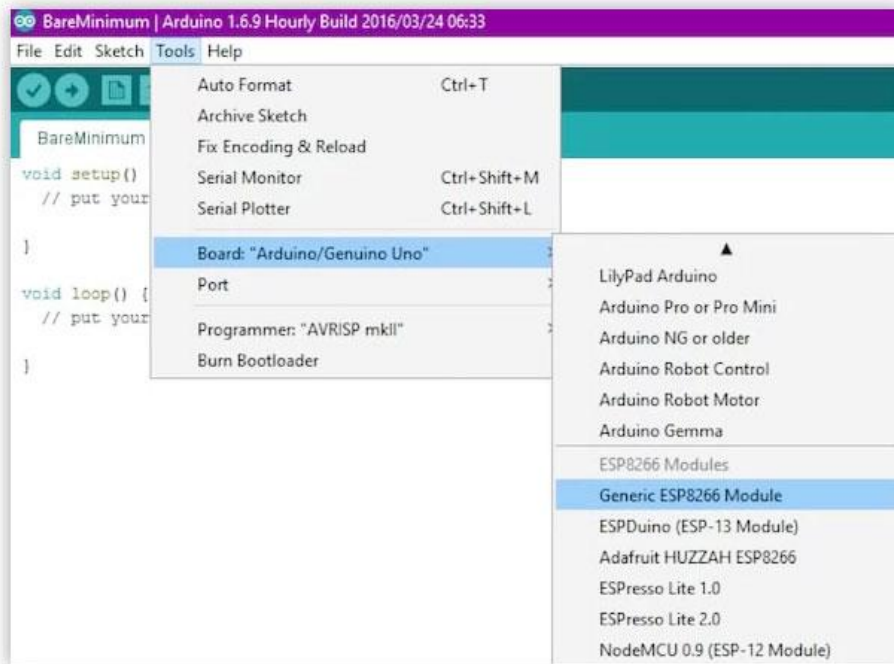


USB TTL → ESP8266 ESP-01
 GND → GND
 TX → RX
 RX → TX
 RTS → RST
 DTR → GPIO0

Загрузка кода ESP8266

Используйте любой из приведенных выше способов и откройте Arduino IDE, затем выберите плату ESP8266 в меню:

Tools → Board → Generic ESP8266 Module
(Инструменты → Плата → Модуль ESP8266)



Теперь скопируйте приведенный ниже код в Arduino IDE и нажмите кнопку загрузки.

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "YOUR_SSID"; //type your ssid
```

```
const char* password = "YOUR_PASSWORD"; //type your password
```

```
int ledPin = 2; // GPIO2 of ESP8266
```

```
WiFiServer server(80); //Service Port
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  delay(10);
```

```
  pinMode(ledPin, OUTPUT);
```

```
  digitalWrite(ledPin, LOW);
```

```
  // Connect to WiFi network
```

```
  Serial.println();
```

```
  Serial.println();
```

```
  Serial.print("Connecting to ");
```

```
  Serial.println(ssid);
```

```
  WiFi.begin(ssid, password);
```



```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

// Start the server
server.begin();
Serial.println("Server started");

// Print the IP address
Serial.print("Use this URL to connect: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // Match the request

  int value = LOW;
  if (request.indexOf("/LED=ON") != -1) {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
  }
}

```



```
if (request.indexOf("/LED=OFF") != -1){
  digitalWrite(ledPin, LOW);
  value = LOW;
}

//Set ledPin according to the request
//digitalWrite(ledPin, value);

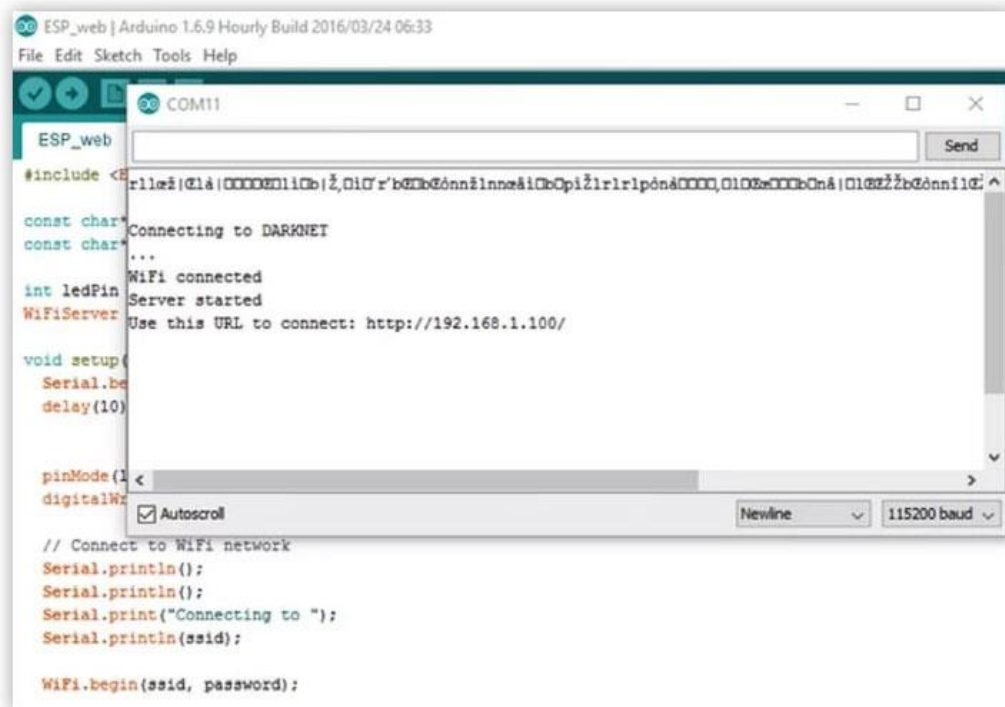
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");

client.print("Led pin is now: ");

if(value == HIGH) {
  client.print("On");
} else {
  client.print("Off");
}
client.println("<br><br>");
client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin
2 ON<br>");
client.println("Click <a href=\"/LED=OFF\">here turn the LED on pin 2
OFF<br>");
client.println("</html>");

delay(1);
Serial.println("Client disconnected");
Serial.println("");
}
```

Откройте последовательный монитор и откройте URL, показанный на вашем последовательном мониторе, через веб-браузер. Подключите GPIO 2 от ESP8266 к более длинному выводу светодиода.



```

ESP_web | Arduino 1.6.9 Hourly Build 2016/03/24 06:33
File Edit Sketch Tools Help
COM11
ESP_web
#include <
const char
const char
...
WiFi connected
Server started
Use this URL to connect: http://192.168.1.100/

void setup(
  Serial.be
  delay(10)

pinMode(1
digitalW

// Connect to WiFi network
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

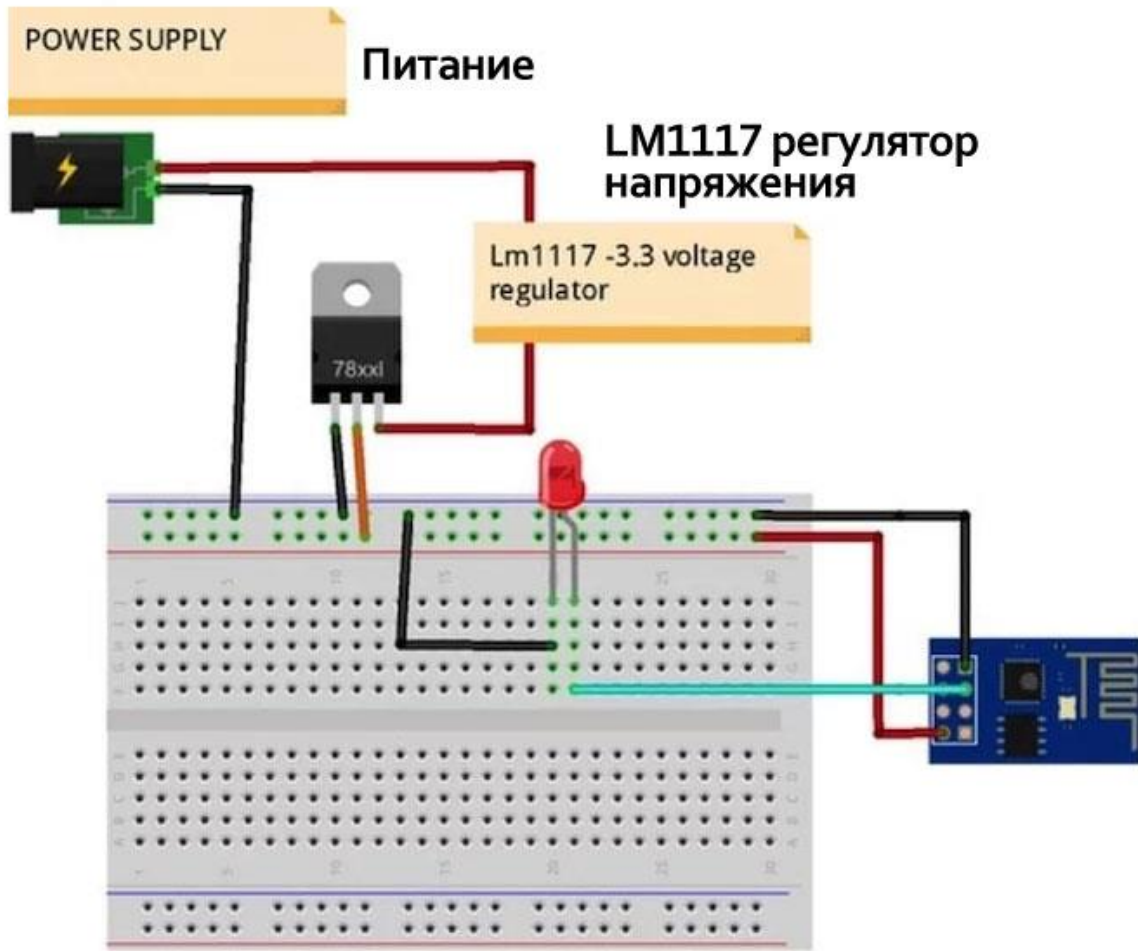
WiFi.begin(ssid, password);

```

Нажмите на соответствующие гиперссылки в браузере, чтобы включить или выключить светодиод.



Удалите все провода, которые были необходимы для загрузки кода. Модуль LM1117 используется для обеспечения регулируемого выхода 3,3 В. Это позволит вам сделать модуль ESP8266 или ESP-01 автономным.



СПИСОК ИСТОЧНИКОВ

1. <http://fritzing.org>
2. <http://inventor.autocad-lessons.ru/>
3. http://medinskiy.net/smart_home/
4. <http://telco.lk/?p=276>
5. <http://wiki.amperka.ru>
6. <https://arduino.cc>
7. <https://autodesk.com/>
8. <https://autodesk.ru/products/inventor/overview>
9. <https://raspberrypi.org>
10. <https://tinkercad.com/>



ПРИЛОЖЕНИЕ

Тест #1

1. Что такое сопротивление?
2. Что такое сила тока?
3. Нарисуйте классическую схему включения светодиода.
4. Нарисуйте схему делителя напряжения.
5. Что такое напряжение?
6. Основные части в программе Arduino.
7. Как называются выводы светодиода?
8. Какие существуют вывод у микроконтроллера Arduino?
9. Какой командой назначаются выводы ножек и где?
10. Что такое начальная установка Arduino?
11. Как работает основная часть программы?
12. Какие электрические компоненты вы знаете?
13. Как выполняются команды в Arduino?
 - a. Слева направо
 - b. Снизу вверх
 - c. Справа налево
 - d. По диагонали справа налево и вниз
 - e. Сверху вниз
 - f. По диагонали слева направо и вниз
 - g. По диагонали Слева направо и вверх
 - h. По диагонали Справа налево и вверх
 - i. По кругу
 - j. По квадрату
14. В какие скобки заключаются основные блоки программы?
 - a. Квадратные
 - b. Прямоугольные
 - c. Круглые
 - d. Зеленые
 - e. Синие
 - f. Фигурные
 - g. Треугольные
 - h. Овальные
 - i. шарообразные
15. Какой командой можно установить напряжение на ножке или его снять?
16. Какой командой прочитать состояние ножки?
17. Как обозначаются на плате Arduino разные ножки?
18. Что вы знаете из особенностей работы ножки номер 13?
19. Какие типы переменных вам известны?
20. Принцип работы фоторезистора?
21. Что такое потенциометр?