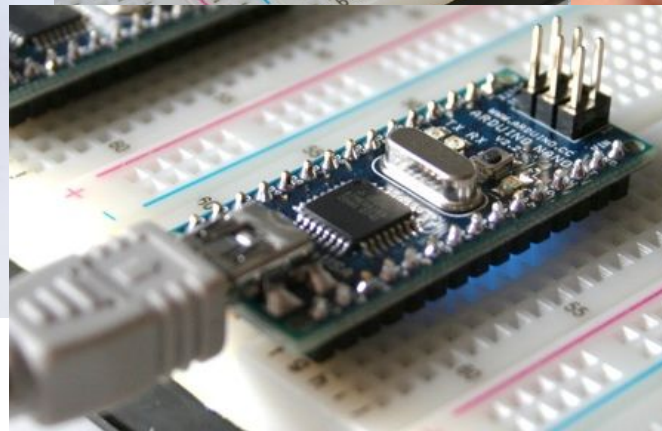
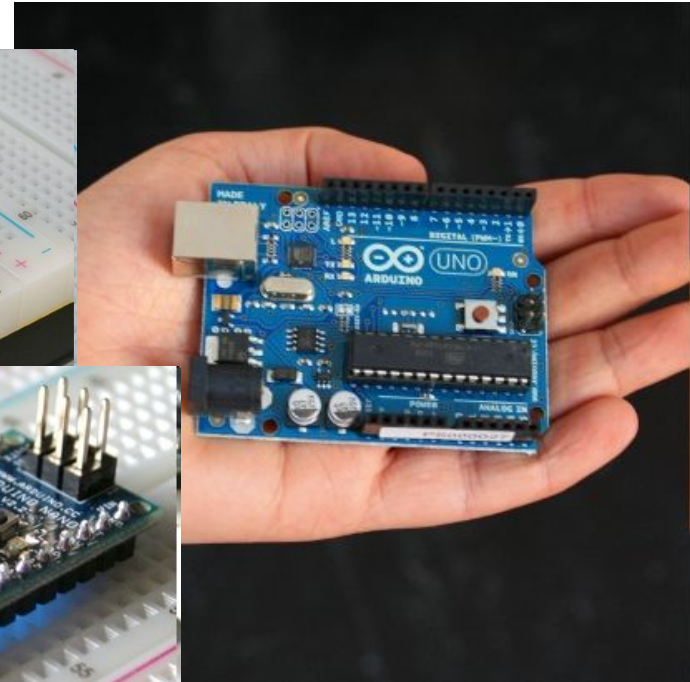
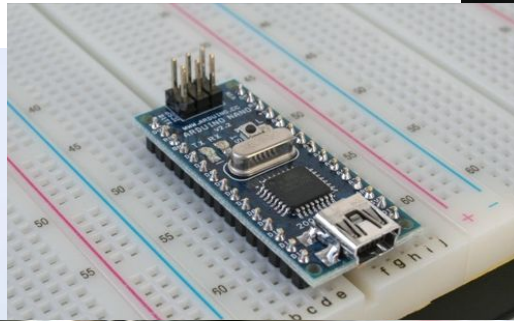


# Введение в Arduino



Докладчик: Яркоев Константин Евгеньевич



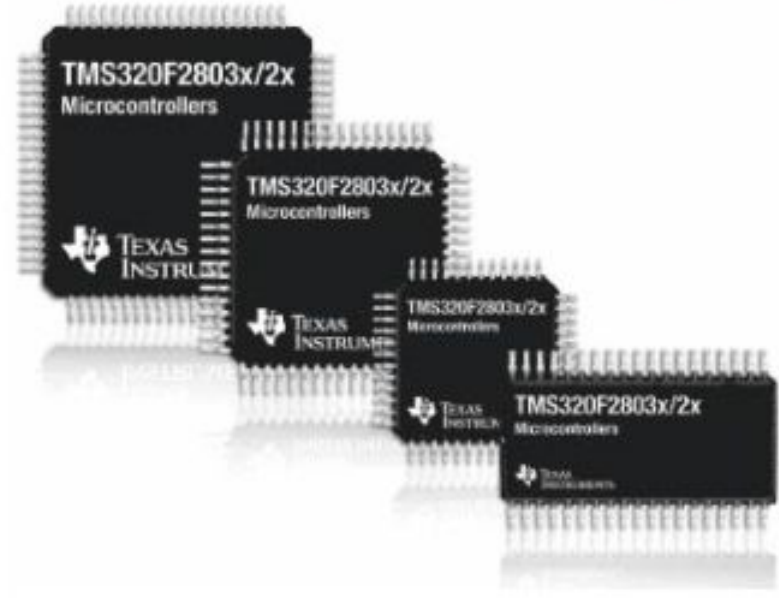
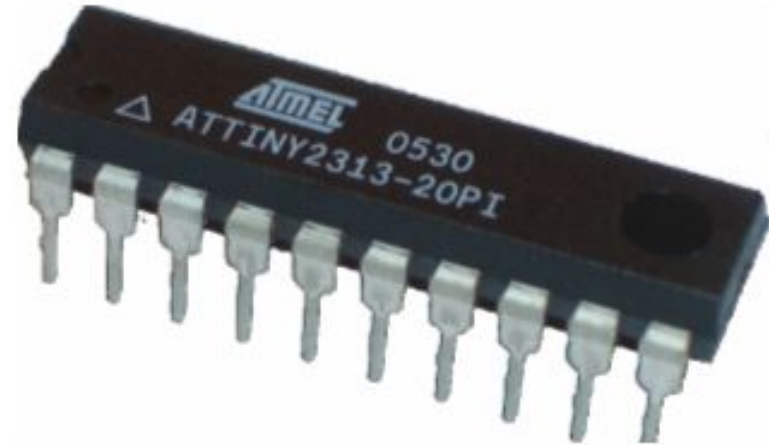
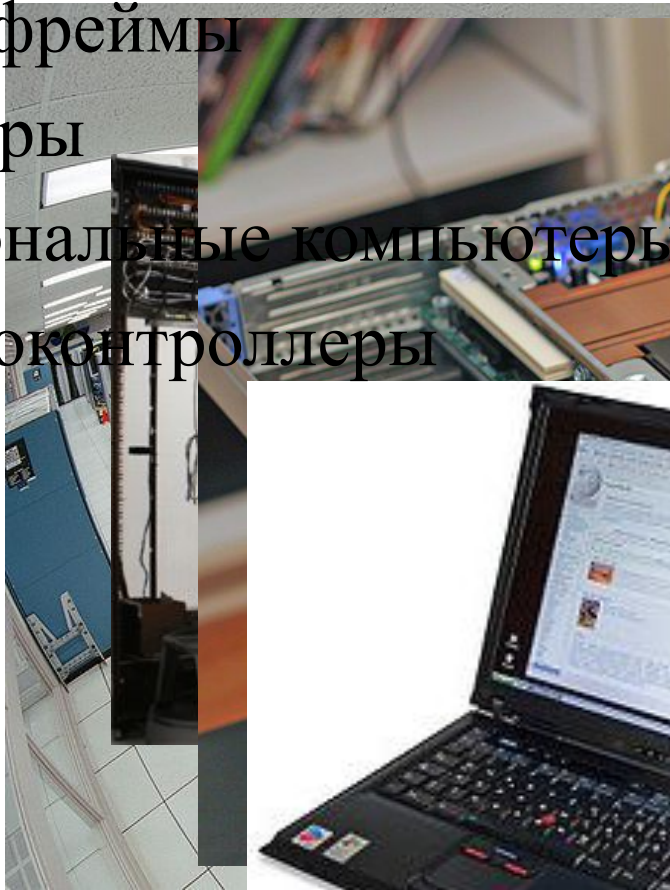
# Содержание

- Микроконтроллеры
- Введение в Arduino
- Основы на пальцах
- Hello, world!
- Проект велосипедной спецодежды



# Виды компьютеров

- Суперкомпьютеры
- Мейнфреймы
- Серверы
- Персональные компьютеры
- Микроконтроллеры



# Области использования МК

- Промышленность
- Медицина
- Транспорт
- Робототехника
- Бытовая техника, умный дом
- Игрушки



# Параметры МК

- Наличие/отсутствие
  - ОЗУ, ПЗУ
  - возможности перепрошивки
  - встроенного генератора тактовой частоты
  - сторожевого таймера
  - периферии
- Архитектура: 8, 16, 32 бит
- Различная частота процессора
- Специального назначения



# Содержание

- Микроконтроллеры
- Введение в Arduino
  - Платформа Arduino
  - Популярность платформы
  - Основные платы
  - Платы расширения
  - Плата Arduino Uno
  - Среда разработки
- Основы на пальцах
- Hello, world!
- Проект велосипедной спецодежды

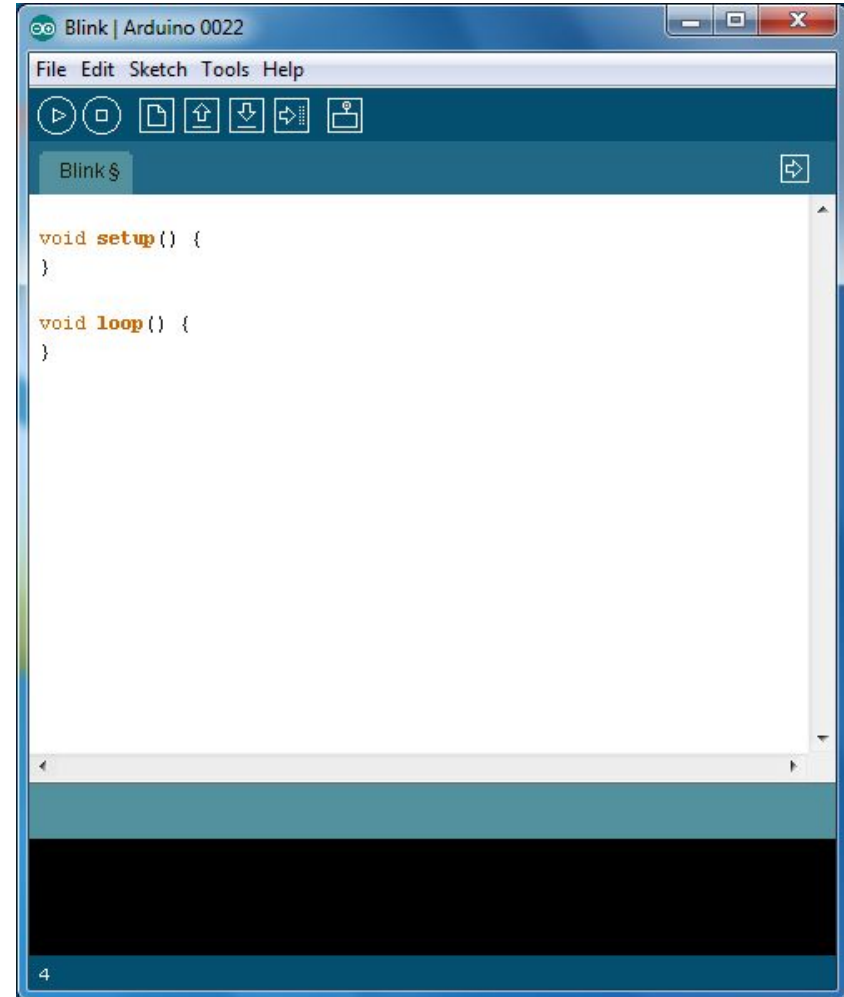
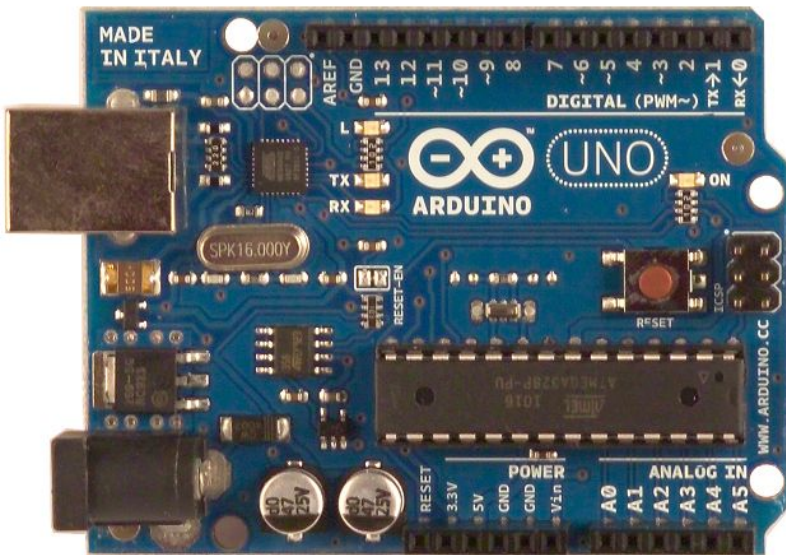




# Платформа Arduino

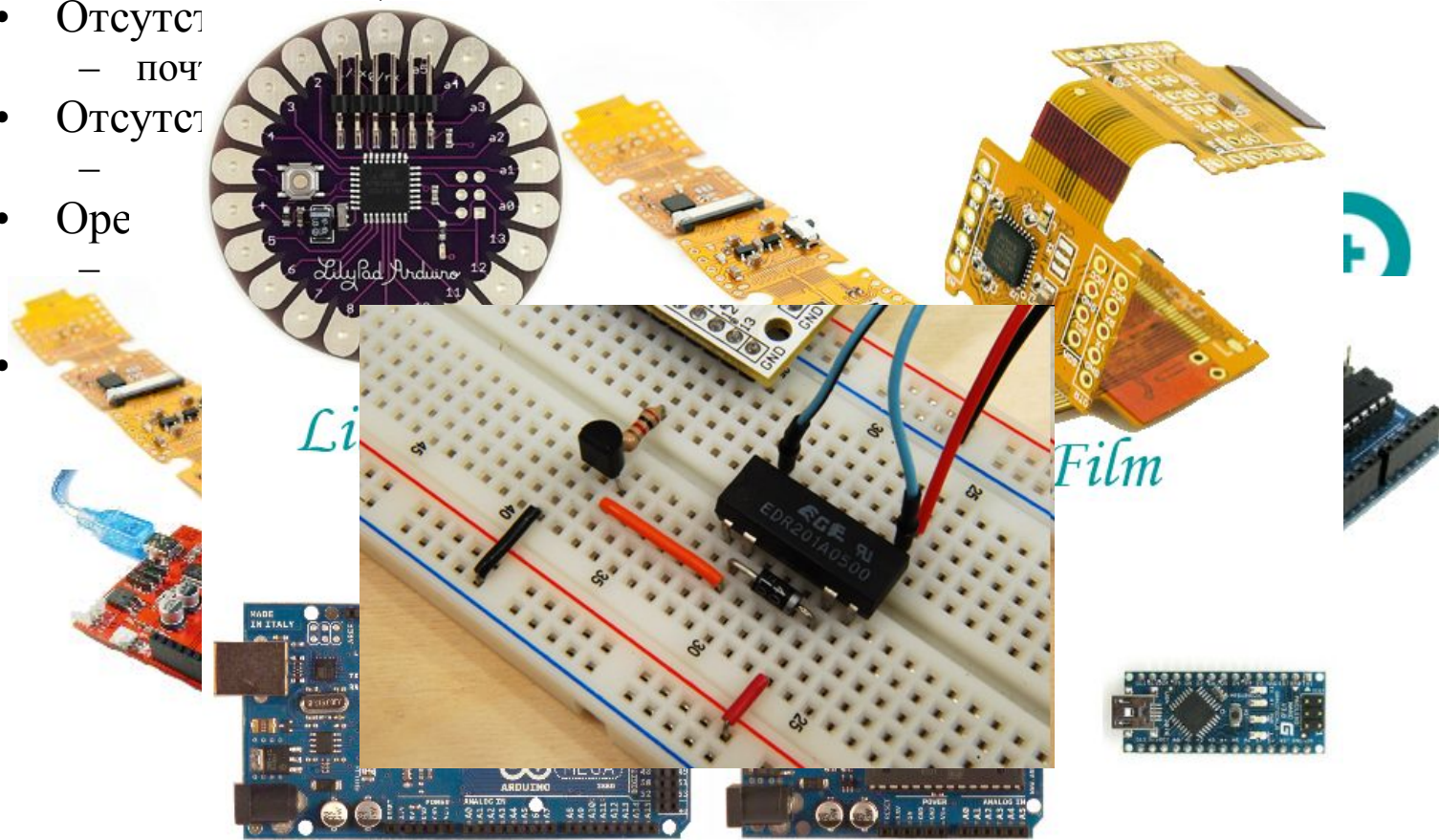
Электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов

- Среда разработки
- Платы



# Популярность платформы

- Низкий порог входа в мир МК
- Разнообразие плат. Две версии носимых плат: *LilyPad* и *Seeeduino Film*
- Кроссплатформенность среды разработки. Переносимость кода для разных плат Arduino.
- Отсутствие
- поч
- Отсутствие
- 
- Open
- 



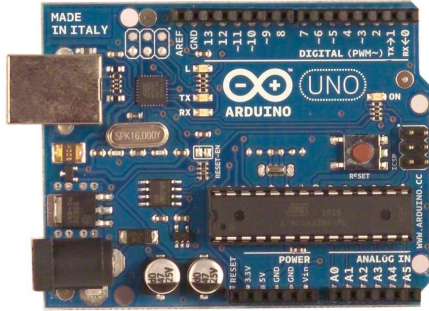
Mega

Uno

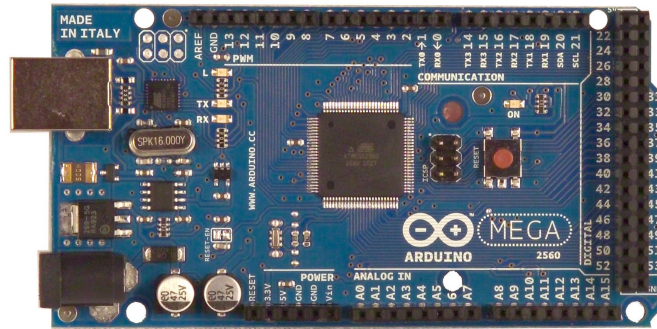
Nano



# Основные платы



- Uno
  - базовая платформа Arduino
  - 14 цифровых входов/выходов (из них 6 ШИМ)
  - 6 аналоговых входов
  - 1 последовательный порт UART
  - программируется через USB с токовой защитой
  - дополняется платами расширения



- Mega2560
  - 54 цифровых входа/выхода (из них 14 ШИМ)
  - 16 аналоговых входов
  - 4 последовательных порта UART
  - дополняется платами расширения
  - программируется через USB



- Nano
  - 14 цифровых входов/выходов (6 могут использоваться как выходы ШИМ)
  - 8 аналоговых входов
  - программируется через Mini-USB



# Платы расширения



GSM+GPS



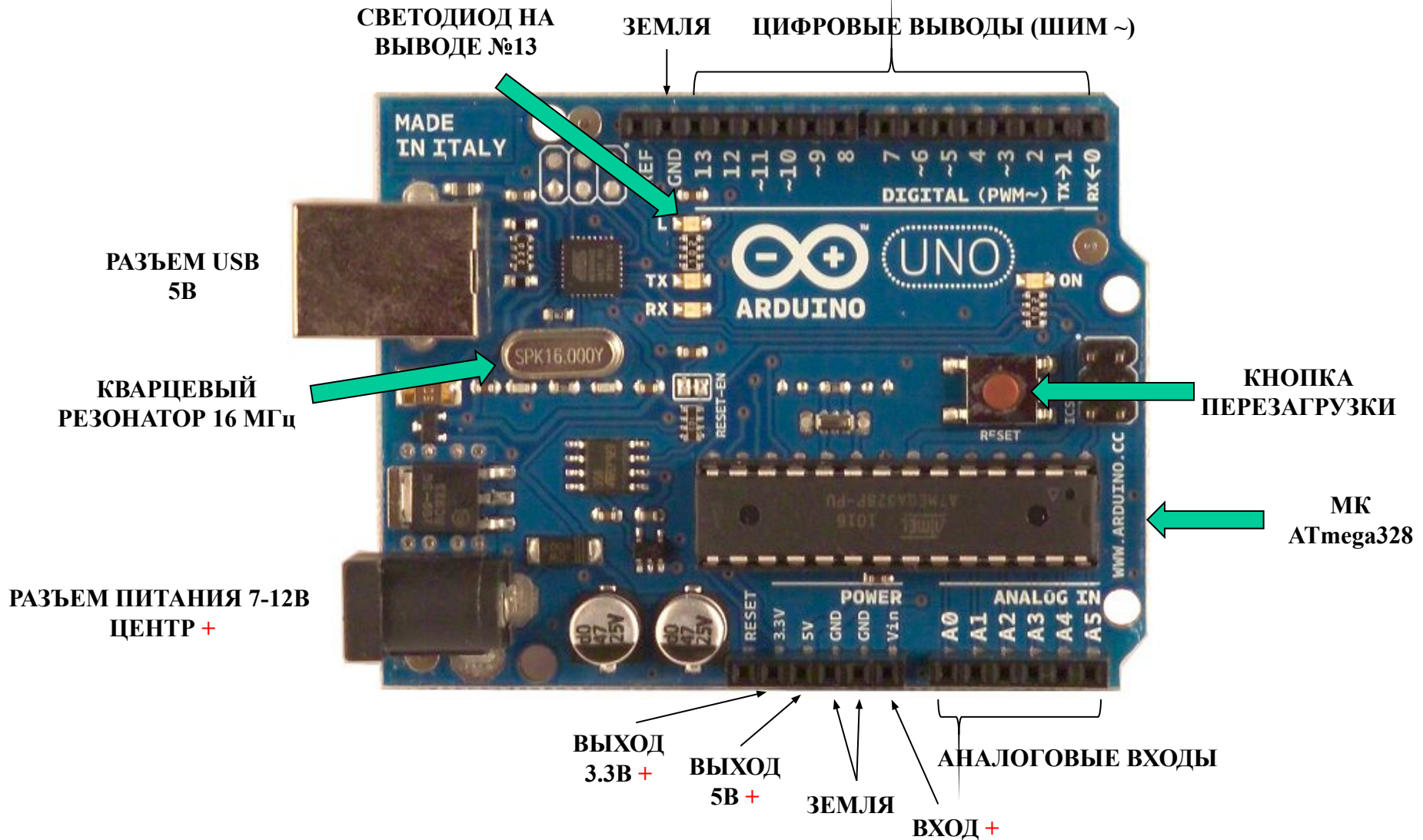
Ethernet+GPS



XBee+LCD

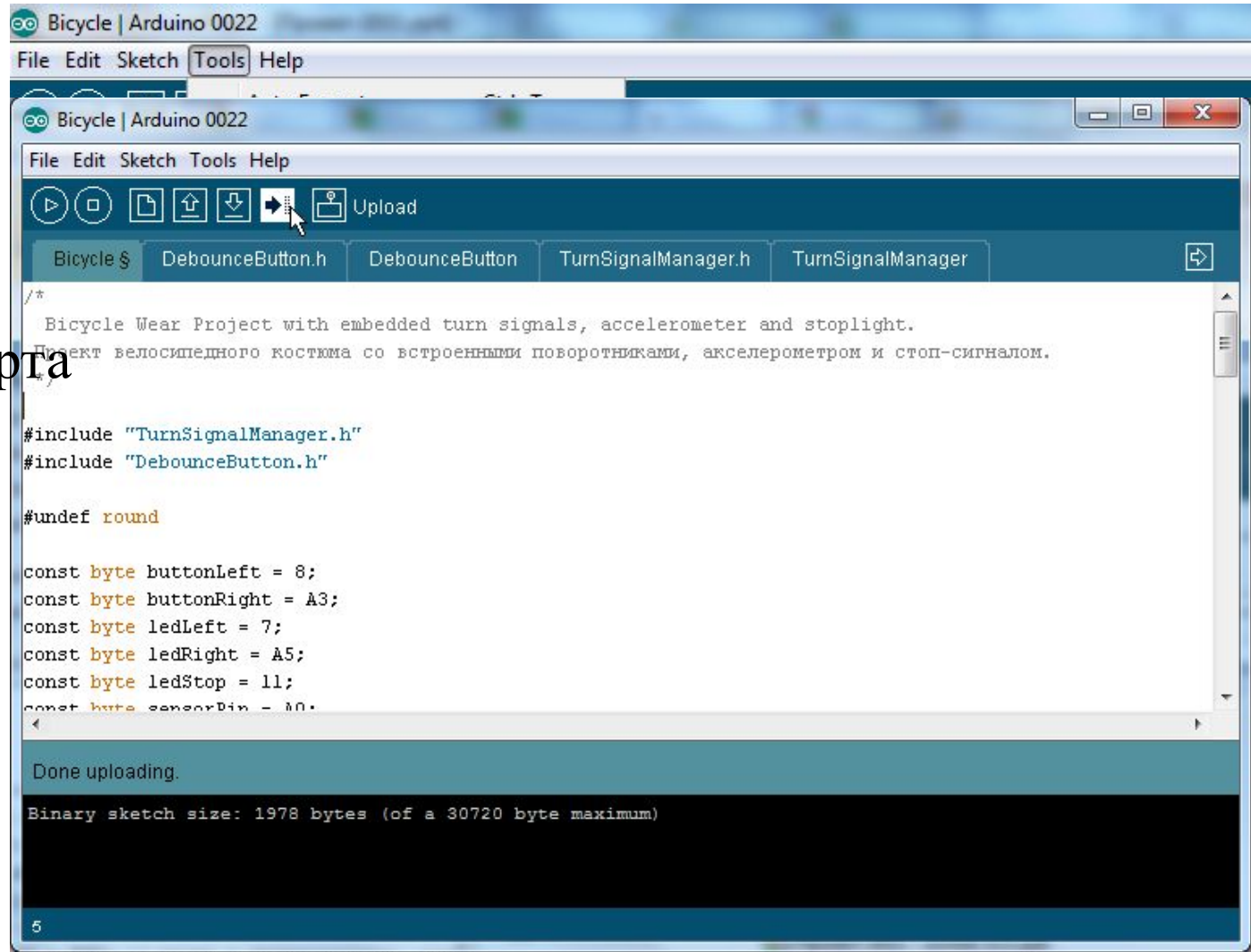


# Плата Arduino Uno



# Среда разработки

Выбор платы  
Выбор СОМ-порта  
Прошивка





# Содержание

- Микроконтроллеры
- Введение в Arduino
- Основы на пальцах
  - Ток, напряжение, сопротивление
  - Резистор
  - Конденсатор
  - Катушка индуктивности
  - Диод
  - Транзистор
  - Ноль и единица
- Hello, world!
- Проект велосипедной спецодежды





# Ток, напряжение, сопротивление

- Высота жидкости подобна напряжению
- Чем больше разность уровней, тем больше энергия
- Другое название напряжения – разность потенциалов
- Чем больше разность уровней тем быстрее и сильнее поток



- Трение жидкости о стенки трубы, а также выполнение какой-либо работы, обеспечивают сопротивление
- Чем сильнее сопротивление тем медленнее поток
- При большом сопротивлении можно увеличить поток, подняв давление – разность потенциалов



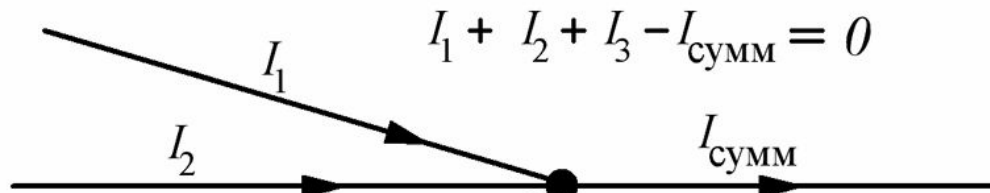
# Закон Ома

- Сила тока в цепи прямо пропорциональна напряжению и обратно пропорциональна полному сопротивлению цепи
- $I = U / R$
- $U$  – величина напряжения в вольтах
- $R$  – сумма всех сопротивлений в омах
- $I$  – протекающий по цепи ток в амперах



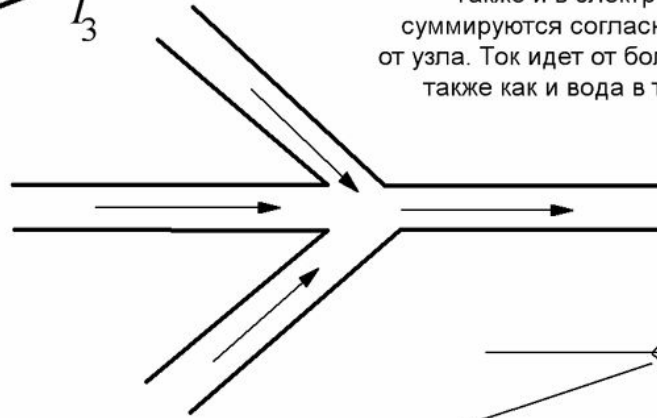
# Закон Кирхгофа

$$\sum_{j=1}^n I_j = 0$$

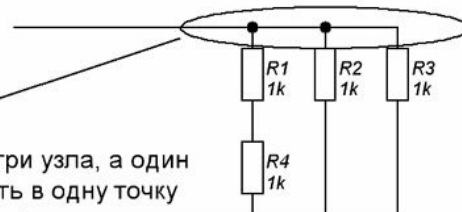


$$I_1 + I_2 + I_3 - I_{\text{сумм}} = 0$$

Вода в трубах не может взаться из ниоткуда, поэтому исходящий поток равен сумме входящих. Также и в электрической цепи. Токи в узле суммируются согласно их направлению: к узлу или от узла. Ток идет от большего потенциала к меньшему, также как и вода в трубе под давлением насоса.

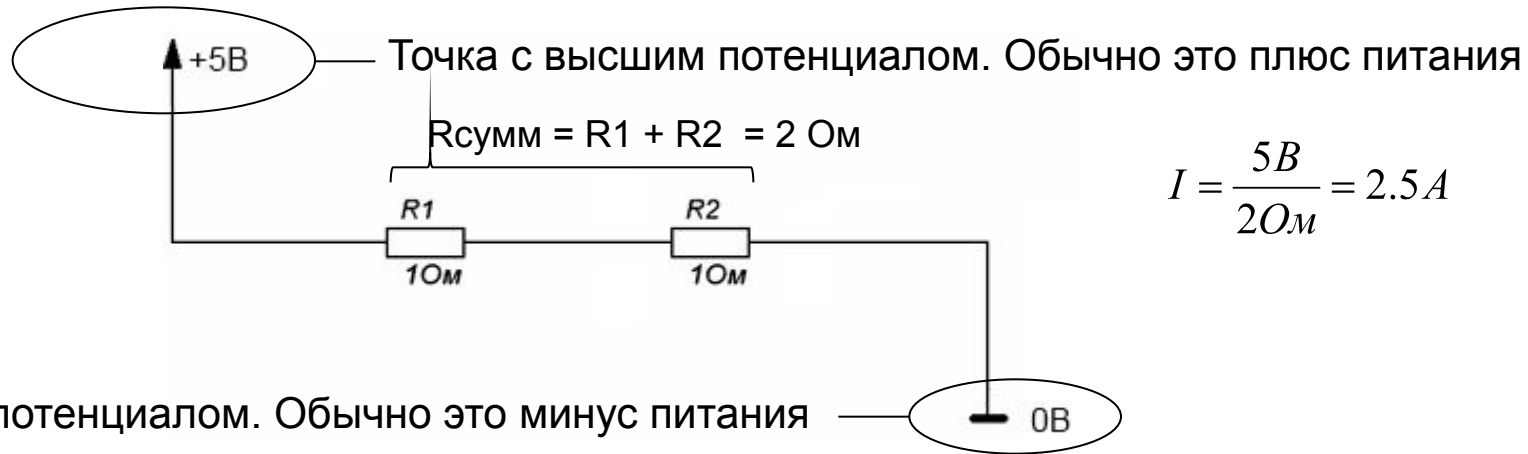


Понятие узла весьма условное. Например тут не три узла, а один так как их без проблем можно стянуть в одну точку

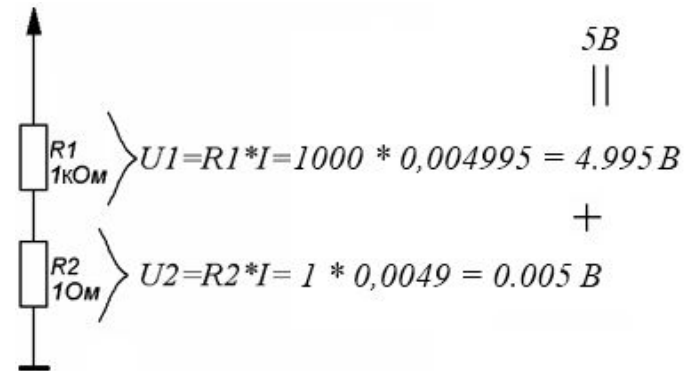
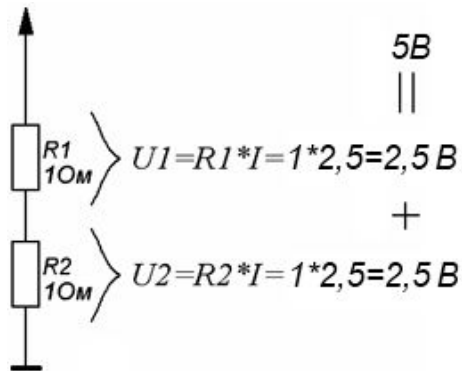


# Закон Ома на практике

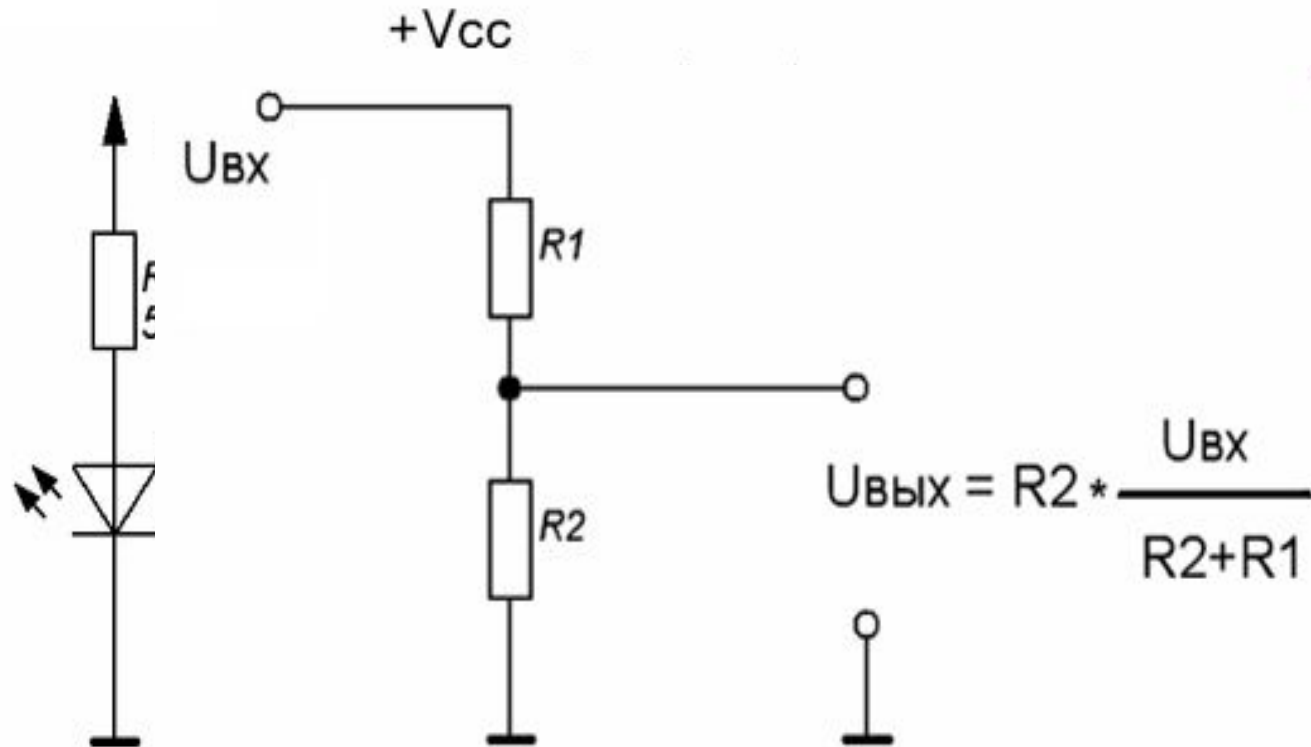
$$I = \frac{U}{R}$$



Распределение напряжения в зависимости от сопротивления:



# Резистор

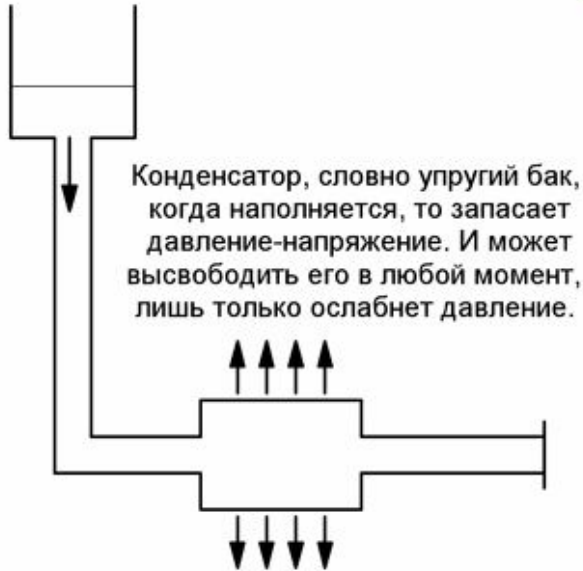


ОПростейший делитель напряжения

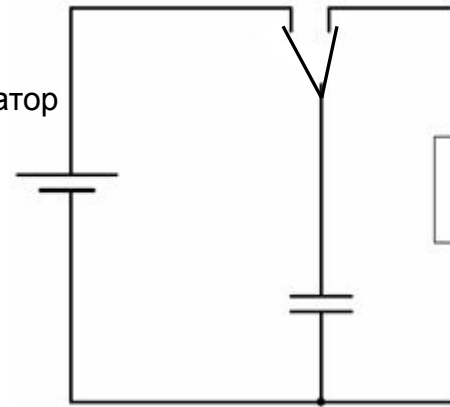




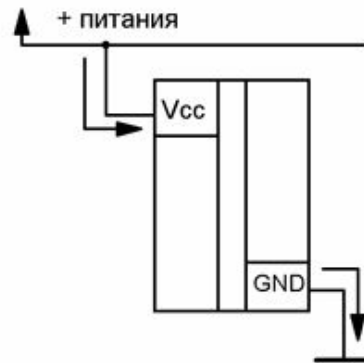
# Конденсатор



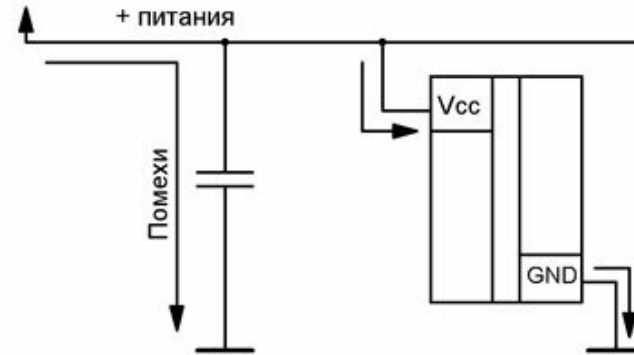
Сейчас конденсатор заряжается от источника



Но если переключить рубильник на другую цепь, то произойдет разряд конденсатора на резистор



Конденсатора нет, и все напряжение вместе с помехами идет через девайс. Что явно не идет ему на пользу

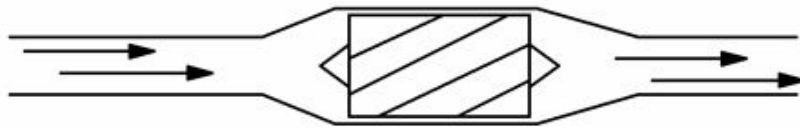


Когда кондер есть, то переменная составляющая в основном идет через него сразу на землю, т.к. сопротивление переменному току у него явно меньше, чем сопротивление девайса.

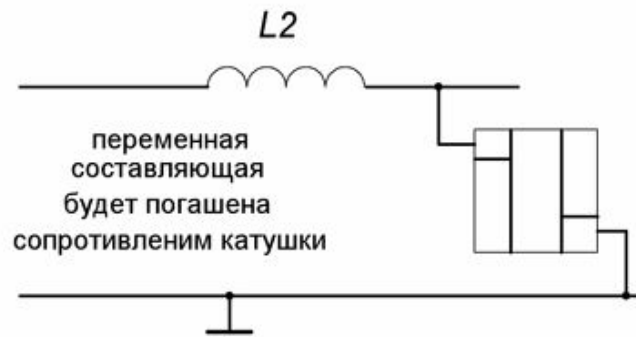
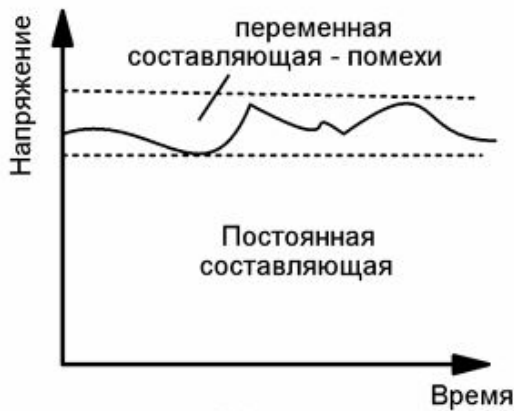
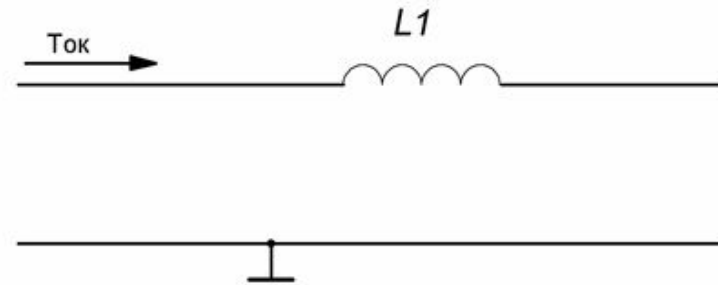
Ну, а постоянная составляющая уже идет через девайс. Так как конденсатор ее никогда не пропустит через себя



# Катушка индуктивности



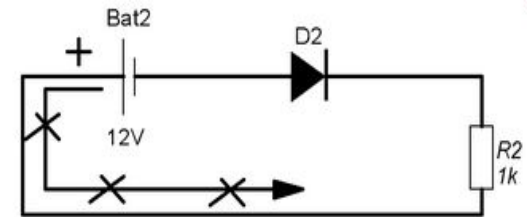
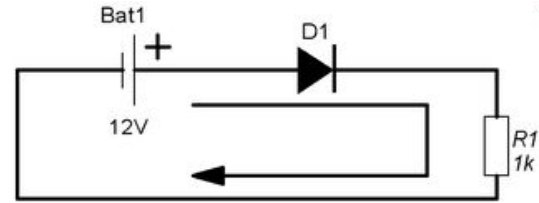
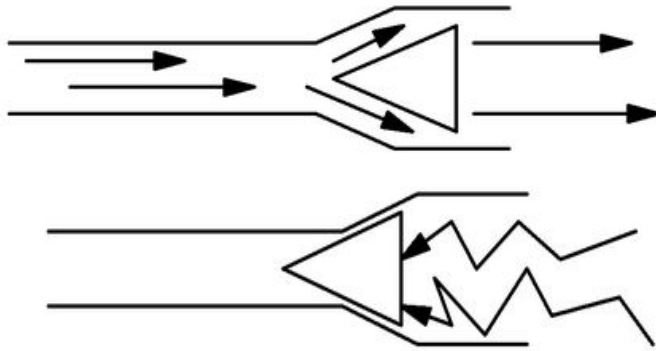
Катушка похожа на массивную турбину



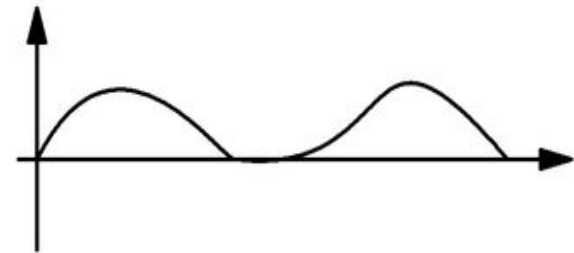
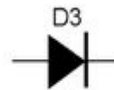
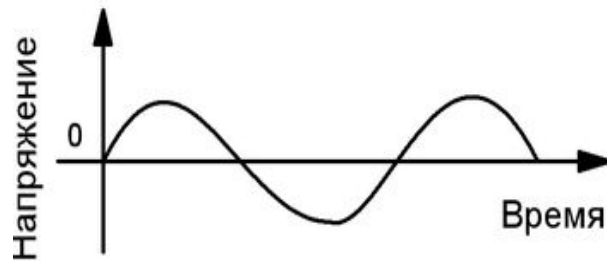
Переменная составляющая "завязнет" на индуктивном сопротивлении катушки и сильно ослабнет, а постоянная составляющая пройдет через катушку практически без потерь. В итоге, на девайсе будет почти "чистое" постоянное напряжение



# Диод



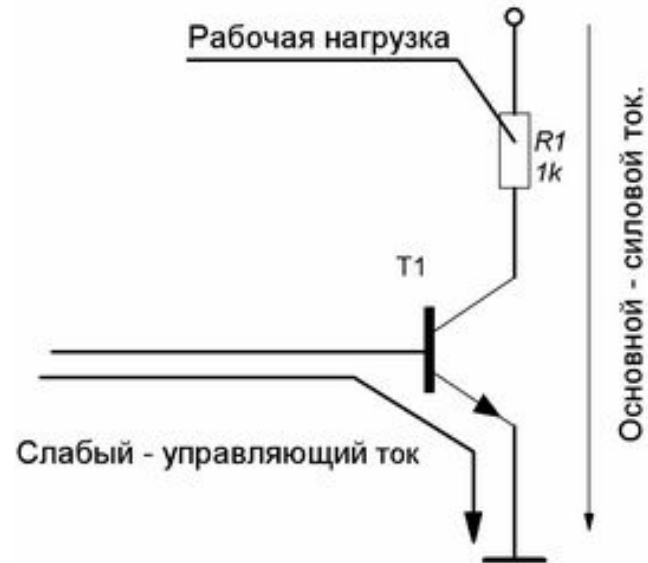
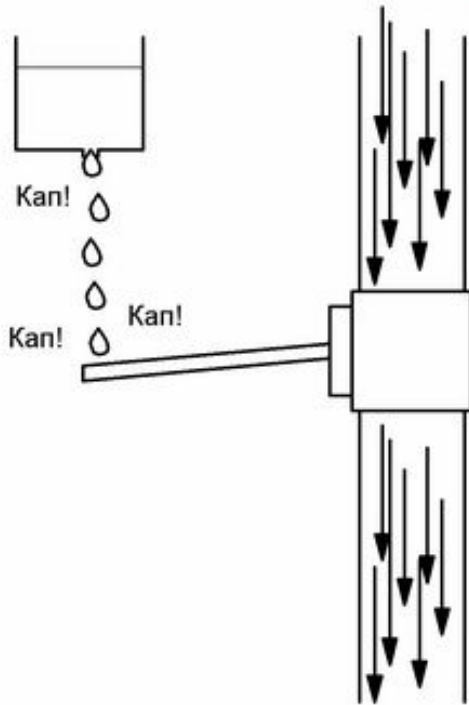
Диод - тот же ниппель, только электрический



Диод пропустил через себя только положительную полуволну переменного сигнала  
Все, что было ниже нуля (т.е. шло в другом направлении) "завязло" на диоде.



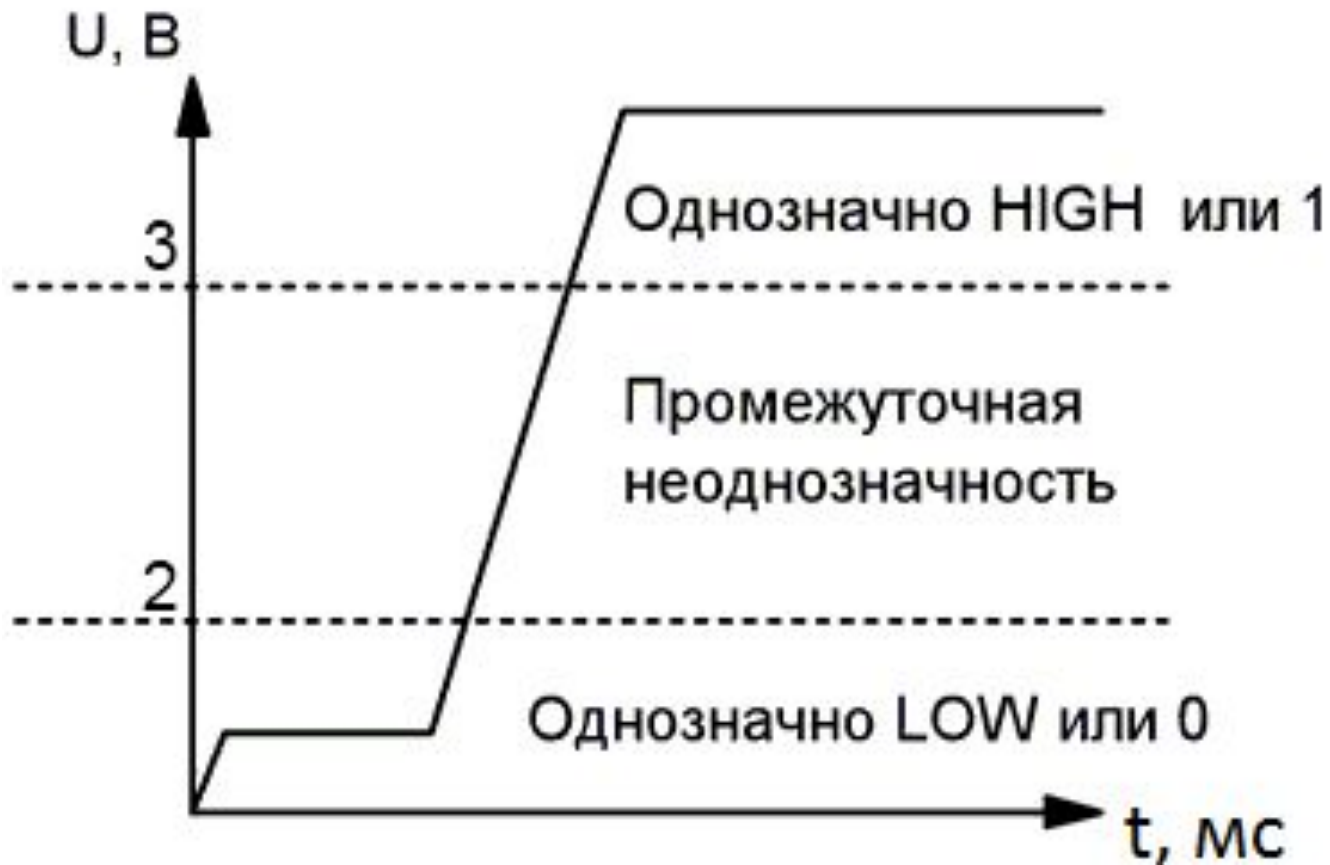
# Транзистор



- Транзистор подобен вентилю, где крошечная сила может управлять могучим потоком энергии, в сотни раз превышающим управляющий
- Транзистор позволяет слабым сигналом, например с ноги микроконтроллера, управлять мощной нагрузкой типа двигателя или лампочки.
- Если не хватит усиления одного транзистора, то их можно соединять каскадами

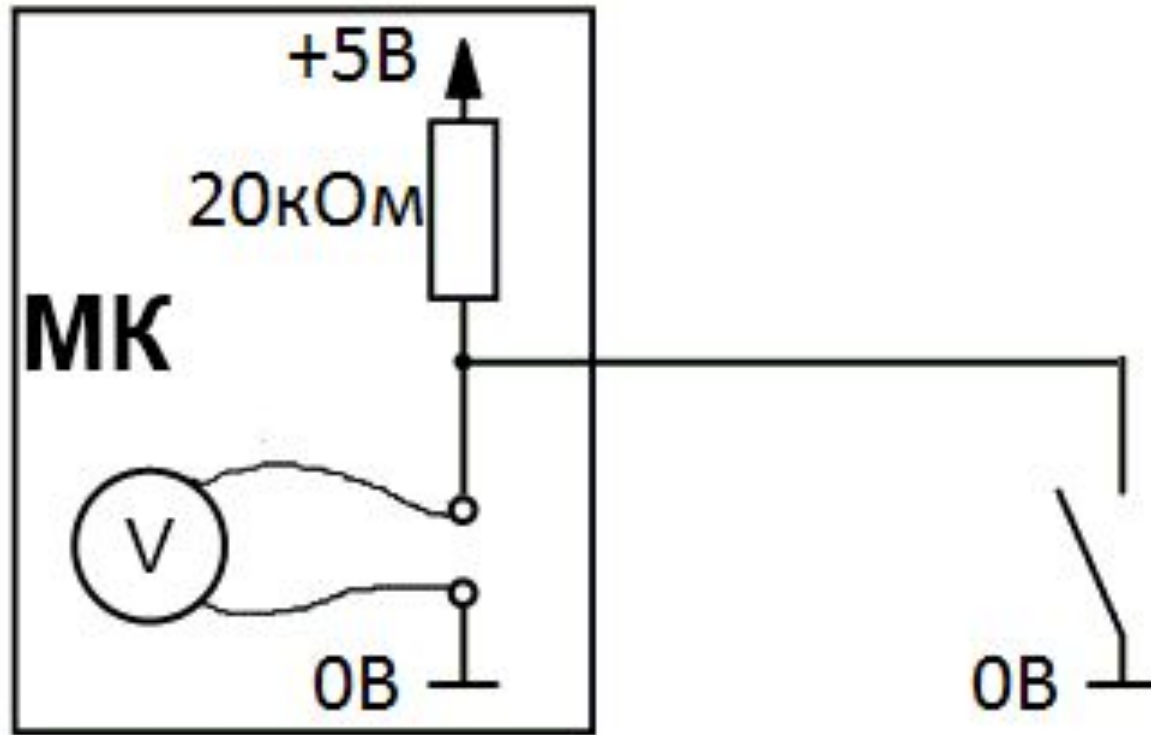


# Понятие нуля и единицы





# Подтяжка выводов до нужного напряжения



# Содержание

- Микроконтроллеры
- Введение в Arduino
- Основы на пальцах
- **Hello, world!**
  - План полета
  - Предполетная подготовка
  - Полет
  - Разбор полета
- Проект велосипедной спецодежды



# Hello, world!

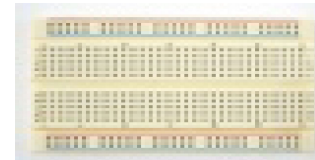
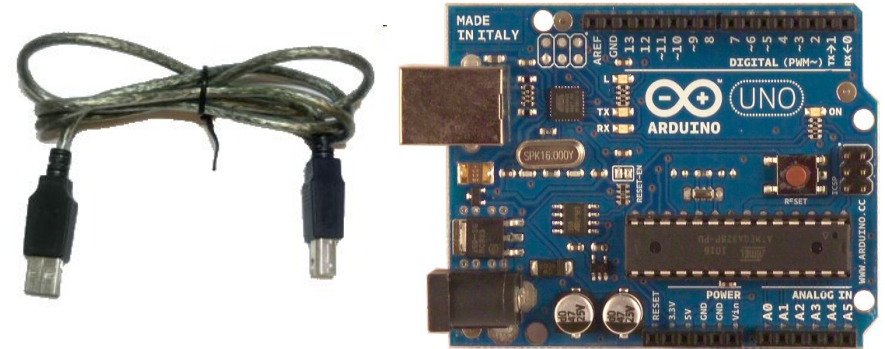
## План полета

- Предложить схему для мигания светодиодом
- Написать программу, которая мигает светодиодом
- Скомпилировать программу
- Загрузить программу в Arduino
- Программа выполняется = светодиод мигает



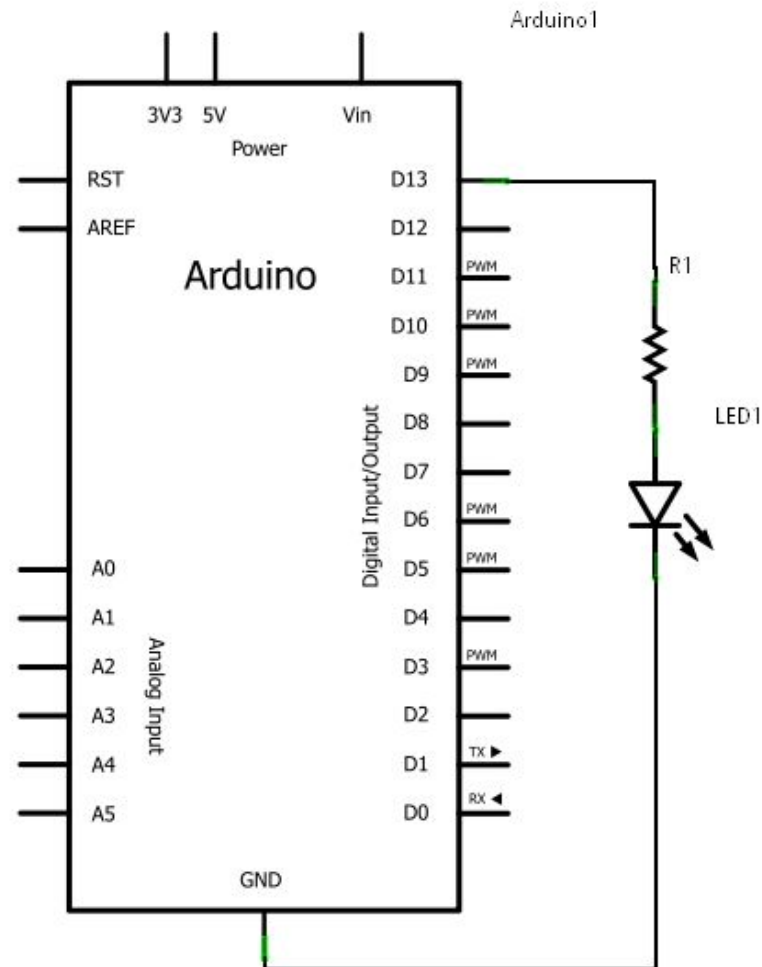
# Предполетная подготовка

- Проверить наличие Arduino-совместимой платы и USB-кабеля
- (Опционально) Проверить наличие макетной платы, соединительных проводов, диода и резистора на 150-500 Ом



# Предполетная подготовка

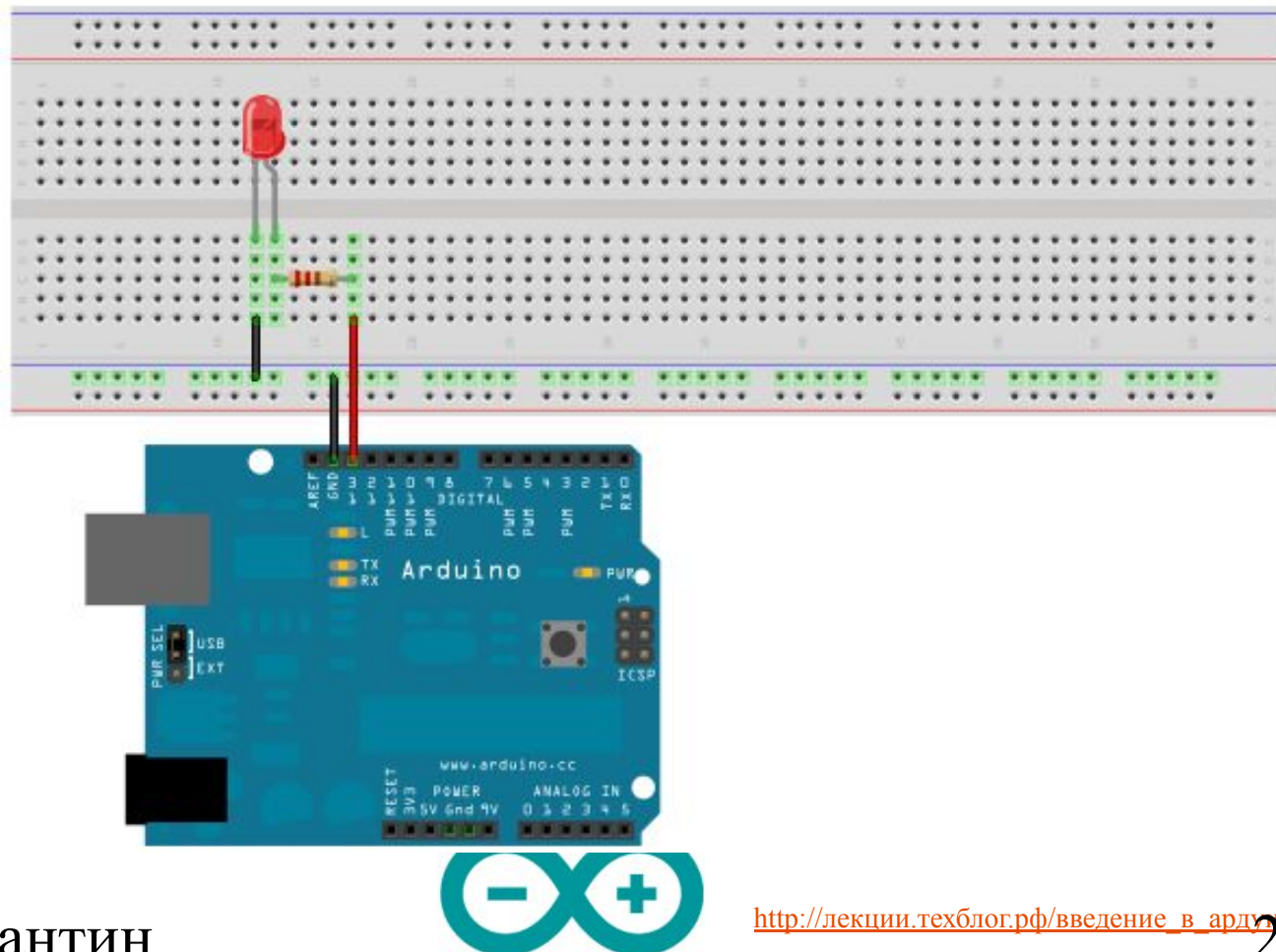
## Принципиальная схема





# Предполетная подготовка

## Макетная плата



# Полет

```
/*  
  Blink.  
  Включает светодиод на секунду, затем выключает на секунду в цикле.  
*/  
  
// Инициализация. Метод вызывается только 1 раз, когда стартует скетч, после подачи питания  
// или после сброса платы. Используется для инициализации переменных, определения режимов  
// работы выводов, запуска используемых библиотек  
void setup() {  
  pinMode(13, OUTPUT); // назначить 13-й вывод как выход  
}  
  
// Бесконечный цикл. После выполнения setup(), данный метод вызывается каждый раз после  
// завершения последнего оператора в цикле  
void loop() {  
  digitalWrite(13, HIGH); // включить светодиод на 13 выводе  
  delay(1000);           // подождать 1 секунду = 1000 миллисекунд  
  digitalWrite(13, LOW); // выключить светодиод на 13 выводе  
  delay(1000);           // подождать 1 секунду = 1000 миллисекунд  
}
```



# Разбор полета (1)

Тип возвращаемого значения,  
либо void, если ничего не  
возвращаем

Имя функции

Параметры функции и их тип

```
int multiply( int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```

Объявление переменной типа int

Инициализация переменной значением

Прекращение выполнения функции и  
возврат значения типа int

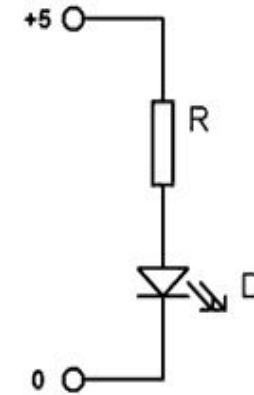
Оператор присваивания



# Разбор полета (2)

Характеристики диода:

- Тип корпуса
- Угол рассеивания, градусы
- Типовой (рабочий) ток, А
- Падение (рабочее) напряжения, В
- Цвет свечения (длина волны), нм



Пример:  $U_{\text{светодиода}} = 2\text{В}$ ,  $I_{\text{светодиода}} = 20\text{мА}$

$$U_{\text{резистора}} = U_{\text{питания}} - U_{\text{светодиода}} = 5\text{В} - 2\text{В} = 3\text{В}$$

$$R_{\text{резистора}} = \frac{U_{\text{резистора}}}{I_{\text{светодиода}}} = \frac{3\text{В}}{0.02\text{А}} = 150\text{ Ом}$$



# Разбор полета (3)

Недостаток программы: если мы захотим поменять вывод №13 на другой, мы должны внести исправления в нескольких местах.

Решение: введем глобальную переменную, хранящую номер вывода

```
/*  
  Blink2.  
  Включает светодиод на секунду, затем выключает на секунду в цикле.  
*/  
  
int ledPin = 13; // Глобальная переменная. Используется, чтобы хранить номер вывода  
  
// Инициализация.  
void setup() {  
  pinMode(ledPin, OUTPUT); // назначить ногу ledPin как выход  
}  
  
// Бесконечный цикл.  
void loop() {  
  digitalWrite(ledPin, HIGH); // включить светодиод на ноге ledPin  
  delay(1000);                // подождать 1 секунду = 1000 миллисекунд  
  digitalWrite(ledPin, LOW);  // выключить светодиод на ноге ledPin  
  delay(1000);                // подождать 1 секунду = 1000 миллисекунд  
}
```



# Разбор полета (4)

Недостаток программы: слишком много дублированного кода внутри цикла

Решение: введем глобальную переменную, хранящую текущее значение напряжения

```
/*
  Blink3.
  Включает светодиод на секунду, затем выключает на секунду в цикле.
*/

int ledPin = 13; // Используется, чтобы хранить номер вывода
boolean ledState = HIGH; // Используется, чтобы хранить текущее значение светодиода ВКЛ/ВЫКЛ

// Инициализация.
void setup() {
  pinMode(ledPin, OUTPUT); // назначить ногу ledPin как выход
}

// Бесконечный цикл.
void loop() {
  ledState = !ledState; // присвоить переменной ledState противоположное значение (воскл. знак означает отрицание)
  digitalWrite(ledPin, ledState); // задать на ноге ledPin значение ledState
  delay(1000); // подождать 1 секунду = 1000 миллисекунд
}
```





# Разбор полета (5)

Недостаток программы: `delay(1000)` означает, что процессор МК простаивает 1 секунду и мы не можем обрабатывать датчики/делать вычисления. Фактически, мы замедлили его работу до частоты 1 Гц вместо 16 МГц. Если сработает датчик, то мы сможем отследить его через 1 сек вместо 1/16000000 сек., либо вообще не сможем отследить событие

```
/*
  Blink4.
  Включает светодиод на секунду, затем выключает на секунду в цикле, не используя функцию Delay
*/

int ledPin = 13; // Используется, чтобы хранить номер вывода
boolean ledState = HIGH; // Используется, чтобы хранить текущее значение состояния светодиода ВКЛ/ВЫКЛ
long previousTimeStamp = 0; // Будет хранить время последнего изменения состояния светодиода

// Инициализация.
void setup() {
  pinMode(ledPin, OUTPUT); // назначить ногу ledPin как выход
}

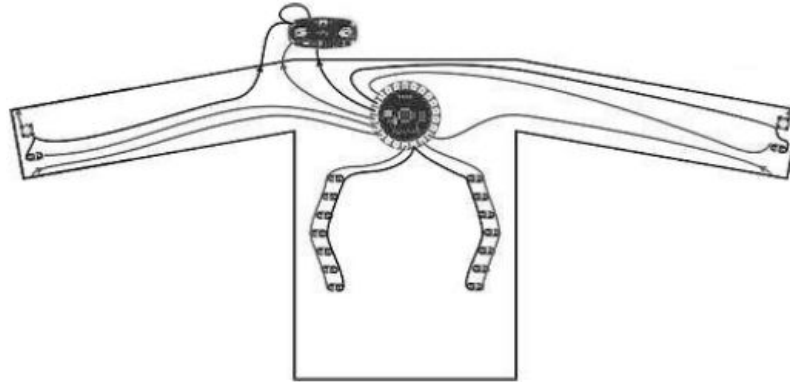
// Бесконечный цикл.
void loop() {
  // Объявляем локальную переменную currentTimeStamp, и инициализируем ее значением функции millis()
  // Функция millis() возвращает количество миллисекунд с момента начала выполнения текущей программы на плате Arduino
  long currentTimeStamp = millis();

  // проверяем, а не изменить ли нам напряжение на светодиоде? Меняем, только если текущее время в мс отличается
  // от времени последнего изменения больше чем на 1000 мс
  if( currentTimeStamp - previousTimeStamp > 1000 )
  {
    previousTimeStamp = currentTimeStamp; // запоминаем текущее время как время последнего изменения
    ledState = !ledState; // присвоить переменной ledState противоположное значение (воскл. знак означает отрицание)
    digitalWrite(ledPin, ledState); // подождать 1 секунду = 1000 миллисекунд
  }
}
```

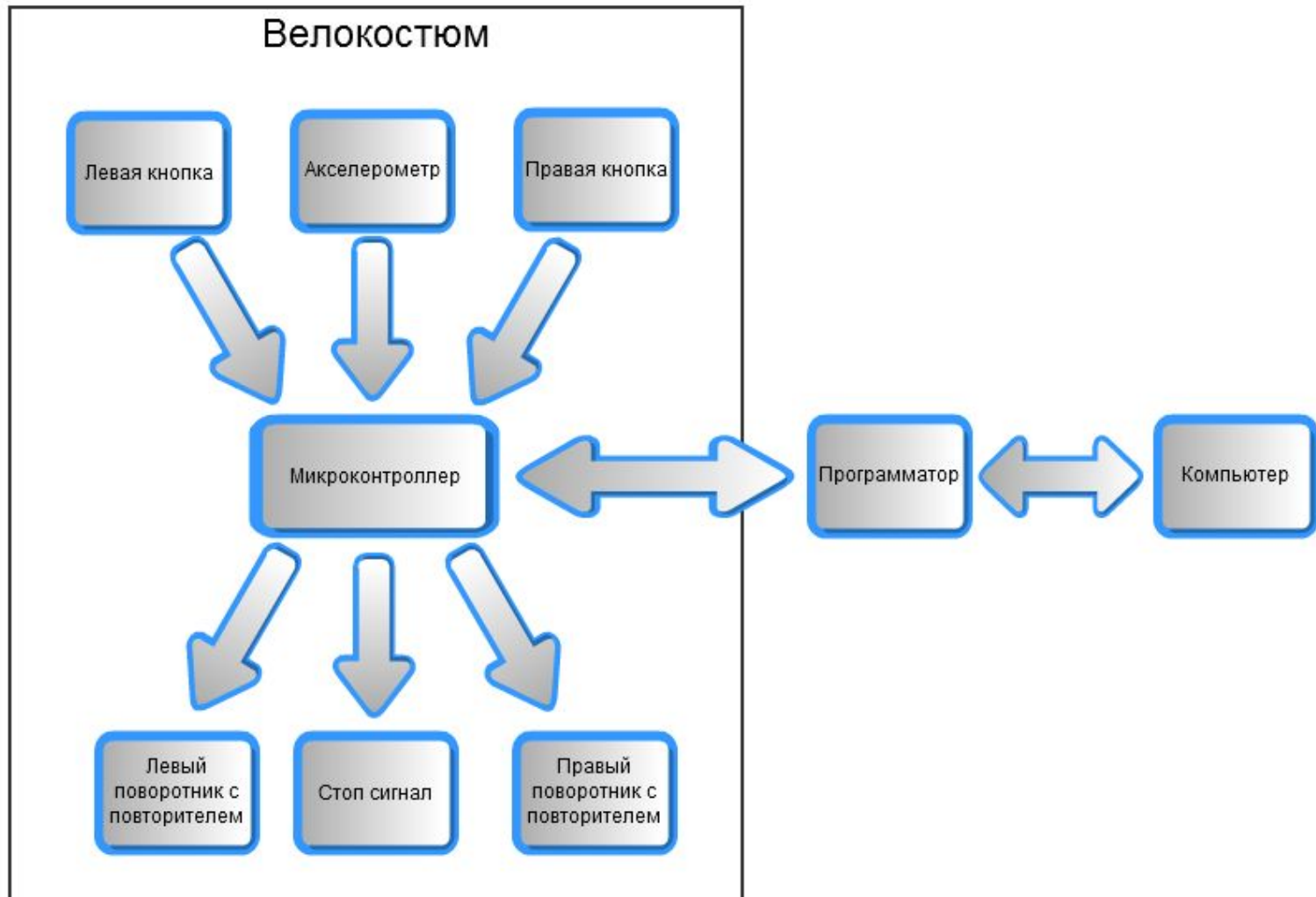


# Содержание

- Микроконтроллеры
- Введение в Arduino
- Основы на пальцах
- Hello, world!
- **Проект велосипедной спецодежды**



# Велосипедная спецодежда



# Материалы (1)

1. Токопроводящая нить
2. Arduino LilyPad
3. Пакет светодиодов для поворотников
4. Пакет светодиодов для стоп сигнала
5. Переходник USB-RS232
6. Блок питания
7. Акселерометр

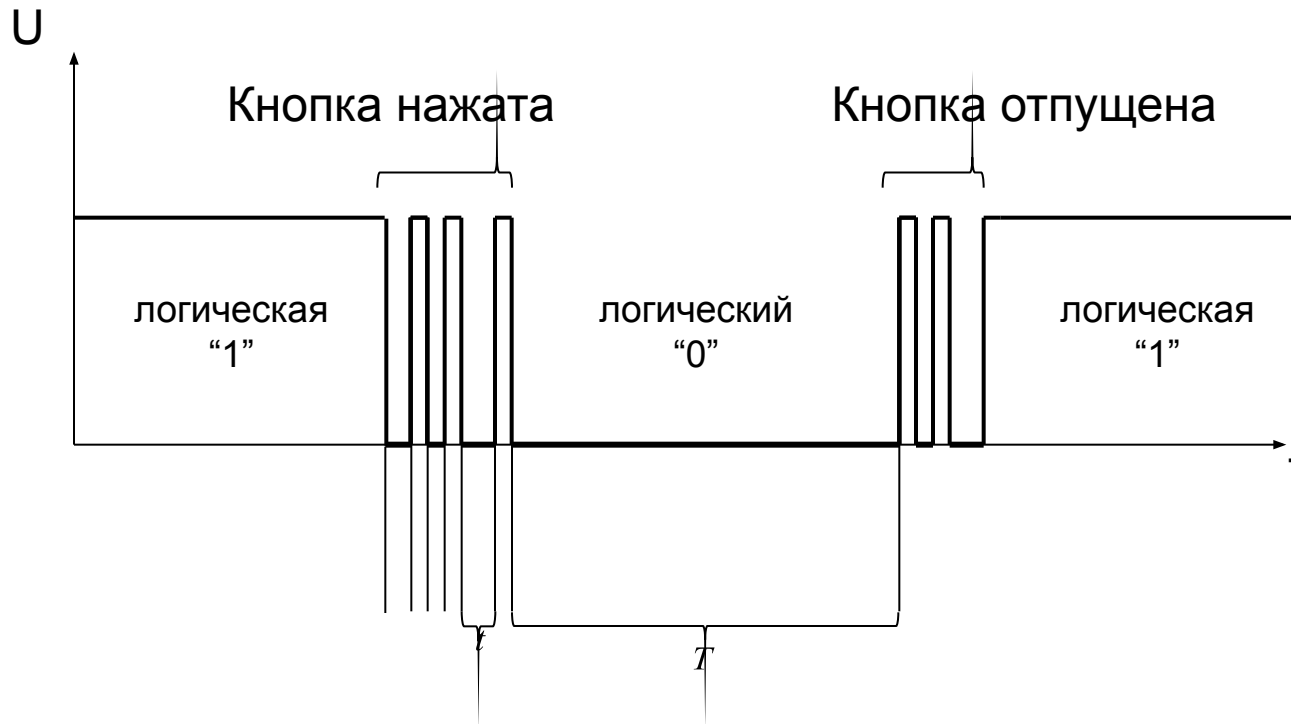


# Материалы (2)

8. Кнопки
9. Иголка
10. Ножницы
11. Мел или мыло для выкройки
12. Куртка



# Дребезг кнопки



# Самый простой способ решения проблемы

- 1. определить замыкание ключа*
- 2. подождать 10-30 миллисекунд*
- 3. если ключ до сих пор замкнут, то замыкание действительно имеет место; иначе – игнорировать его*





# Алгоритм работы программы

## Цикл:

- *Опрашиваем левую и правую кнопки с помощью адаптивного алгоритма определения дребезга кнопки*
- *Если нажата*
  - *левая кнопка и состояние ВЫКЛЮЧЕНЫ, то перейти в состояние ЛЕВЫЙ*
  - *левая кнопка и состояние ЛЕВЫЙ, то перейти в состояние ВЫКЛЮЧЕНЫ*
  - *левая кнопка и состояние ПРАВЫЙ, то перейти в состояние ЛЕВЫЙ*
  - *правая кнопка и состояние ВЫКЛЮЧЕНЫ, то перейти в состояние ПРАВЫЙ*
  - *правая кнопка и состояние ПРАВЫЙ, то перейти в состояние ВЫКЛЮЧЕНЫ*
  - *правая кнопка и состояние ЛЕВЫЙ, то перейти в состояние ПРАВЫЙ*
- *Помигать левым или правым поворотником (или не мигать) в зависимости от текущего состояния*
- *Опрашиваем датчик ускорения*
- *Если значение ускорение выше (ниже) порогового, то включить стоп-сигнал. Иначе выключить*



Программа, управляющая миганием поворотников:

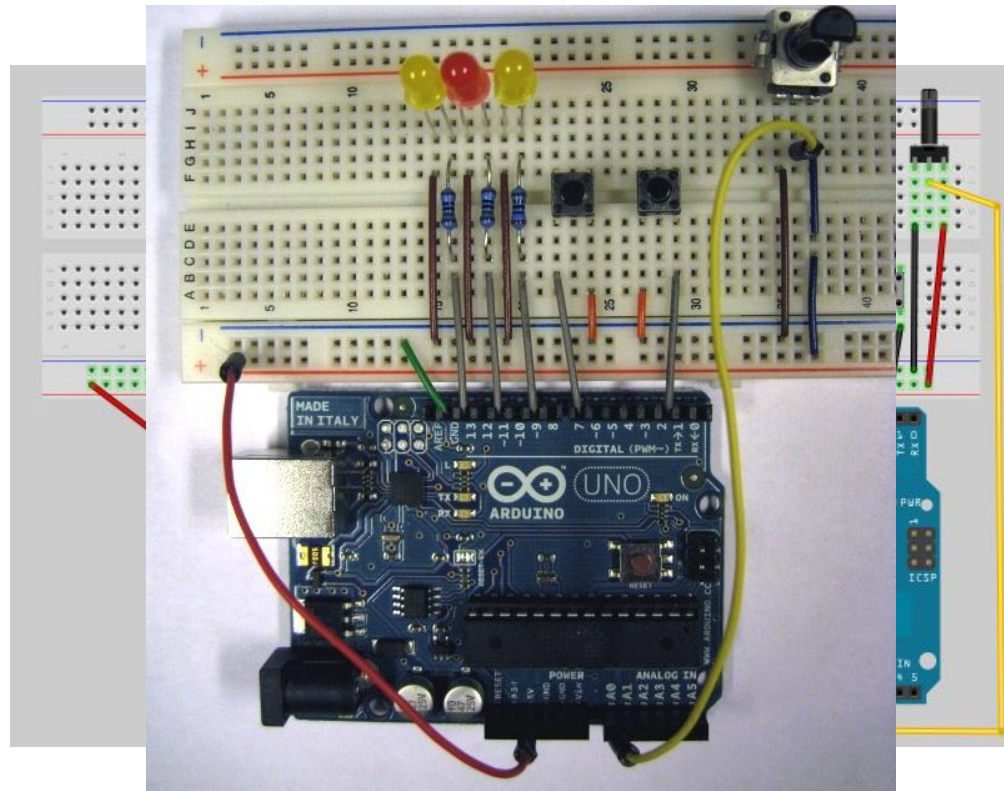
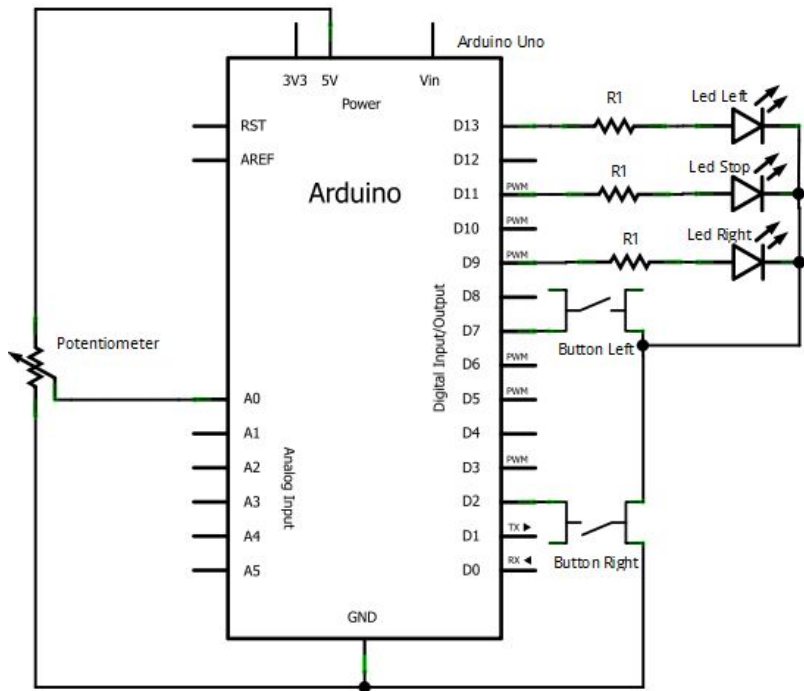
```
if ( leftSwitch.stateChanged ) {  
    if ( leftSwitch.lastState == LOW) { turnSignal.clickLeft(); }  
}  
if ( rightSwitch.stateChanged ) {  
    if ( rightSwitch.lastState == LOW) {turnSignal.clickRight(); }  
}
```

Программа, управляющая включением стоп-сигнала:

```
int sensorValue = analogRead(sensorPin);  
if( sensorValue < 300 )  
    { digitalWrite(ledStop, HIGH); }  
else  
    {digitalWrite(ledStop, LOW);}  
}
```



# Тестирование прототипа



Принципиальная схема с потенциометром вместо акселерометра

Прототипирование в Fritzing  
Тестирование на макетной плате в классе



# Пошив

- Нужно выбирать материал достаточной толщины, т.к. нитка достаточно толстая
- Следует шить так, чтобы стежки не были видны снаружи
- Необходимо минимизировать длину ниток
- После шитья нужно покрыть швы лаком для предотвращения окисления и короткого замыкания



# Статистика

- Потрачено
  - 10 метров нитки
  - 40 человеко-часов работы непрофессионала
  - 4400 рублей, из них

Товар (услуга)	Стоимость, руб.
Электроника	2000
Нитка	900
Доставка из США	600
Куртка	900



# Будущее проекта

В дальнейшем мы хотим развить наш проект, применив:

- гибкую плату Seeeduino Film для более комфортного ношения
- датчик света для автоматического включения светодиодов белого цвета в мерцающем режиме в темное время суток
- беспроводные технологии для разработки нового комплекса, состоящего из следующих устройств
  - модуля управляющей логики на руле
  - блока поворотников со стоп-сигналом.



# Заключение

- Платформа постоянно развивается
- С Ардуино многие вещи становятся простыми
- Ардуино можно и нужно изучать в школе
- “Не хочешь покупать? Сделай сам!”





# Список использованных источников информации

- Болл Стюарт Р. Аналоговые интерфейсы микроконтроллеров – М.: Издательский дом “Додэка-XXI”, 2007. – с. 119
- <http://Arduino.cc>
- <http://Arduino.ru>
- <http://easyelectronics.ru/osnovy-na-palcaх-chast-4.html>
- [http://web.media.mit.edu/~leah/LilyPad/build/turn\\_signal\\_jacket.html](http://web.media.mit.edu/~leah/LilyPad/build/turn_signal_jacket.html)
- <http://sparkfun.com>
- <http://www.seeedstudio.com/>



# Спасибо за внимание!

- Эту презентацию можно скачать здесь:
  - [http://лекции.техблог.рф/введение\\_в\\_ардуино.ppt](http://лекции.техблог.рф/введение_в_ардуино.ppt)
- Скоро: проект велосипедной одежды с инструкцией и исходным кодом
  - <http://техблог.рф/VELOODEZHA>

