



**Южно-Уральский
государственный университет**
Кафедра автоматики и управления

Учебная практика

КАНАШЕВ

Евгений

Александрович

kanashevea@susu.ru



5 МИНУТ

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



10 минут

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



© Кана Е.А.,
ФГУ «ИИИ»
(НИУ)



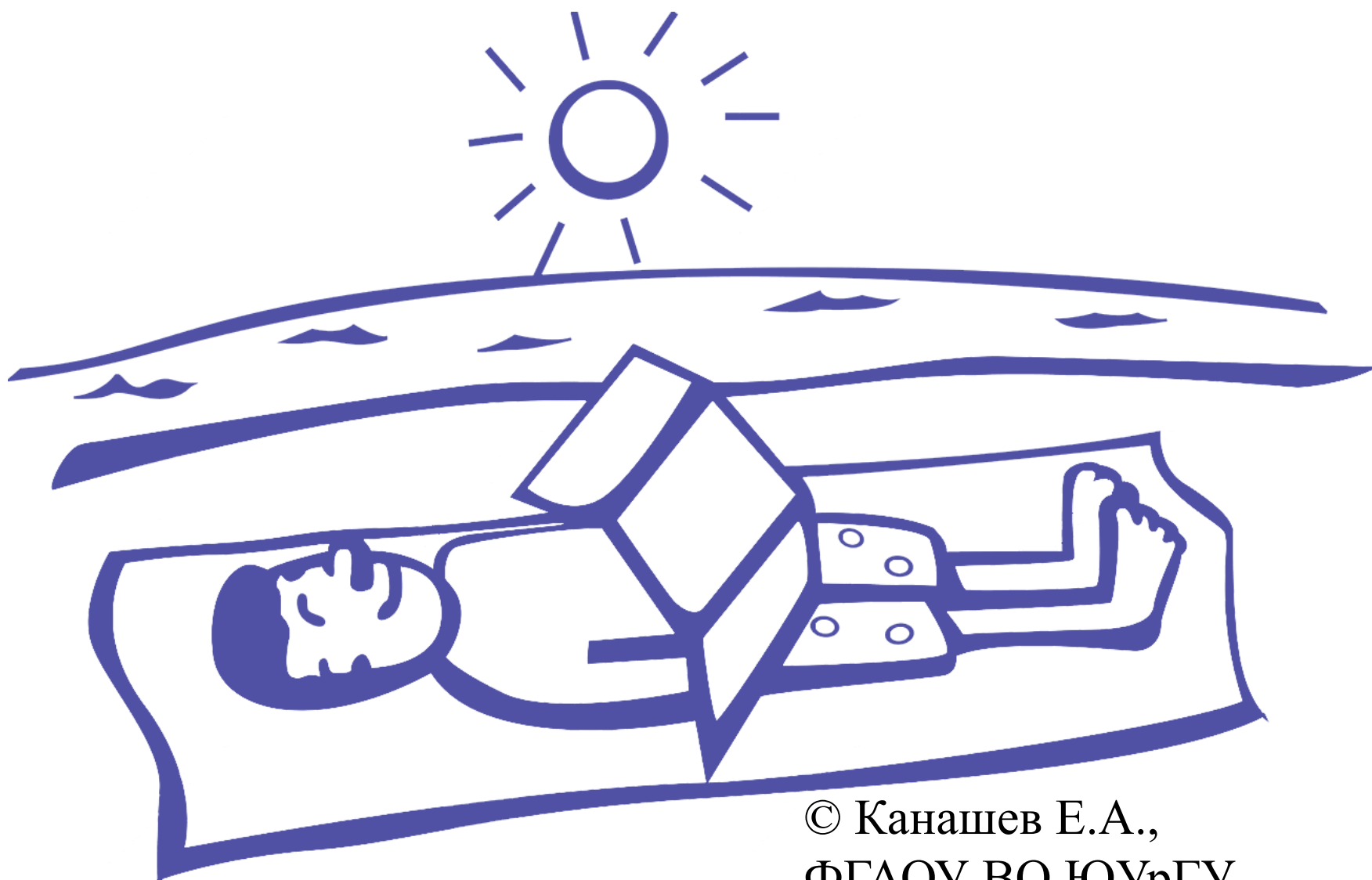
Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



© Кана Е.А.,
ФГУ
ФУрГУ
(НИУ)



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

СМК СТУ 2.0 - 2006

(с изменениями, приказ № 54 от 12.03.2008)

- 4.1.2 Практика является одним из видов учебного процесса, в ходе которого осуществляется непосредственная связь теоретического обучения с будущей практической деятельностью специалиста
- Целью практики является формирование профессиональной позиции будущего специалиста, владеющего стратегией планирования и организации своей деятельности, а также самостоятельно ставящего задачи профессионального и личностного самосовершенствования
- 4.1.3 Виды практик и их продолжительность определяются государственными образовательными стандартами высшего профессионального образования и утвержденными учебными планами по специальности

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Учебная практика

- представляет собой комплексные практические занятия, дополняемые другими видами учебного процесса, в ходе которых осуществляется формирование основных первичных профессиональных навыков, ознакомление со специальностью, подготовка к изучению общепрофессиональных и специальных дисциплин, приобретение навыков работы в коллективе
- проводится в учебных мастерских, в вычислительных центрах и лабораториях университета
- по итогам учебной практики предусматривается дифференцированный зачет, который проставляется руководителем практики от кафедры на основе отчетов, составляемых студентами в соответствии с программой практики

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(Национальный исследовательский университет)

Кафедра автоматике и управления

Отчет по учебной практике

Тема: Программирование микроконтроллеров

Студент _____
Группы ПЗ-198 курса I

Место прохождения учебной практики

ЮУрГУ. Кафедра автоматике и управления. ауд. 705. 712/36
(наименование учреждения, адрес)

Продолжительность - 4 недели

Сроки прохождения практики

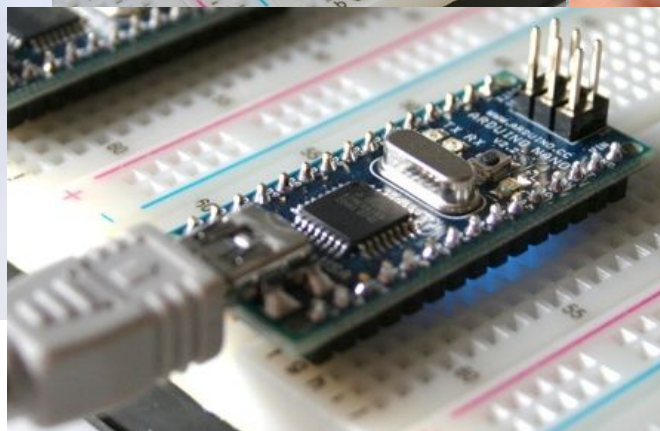
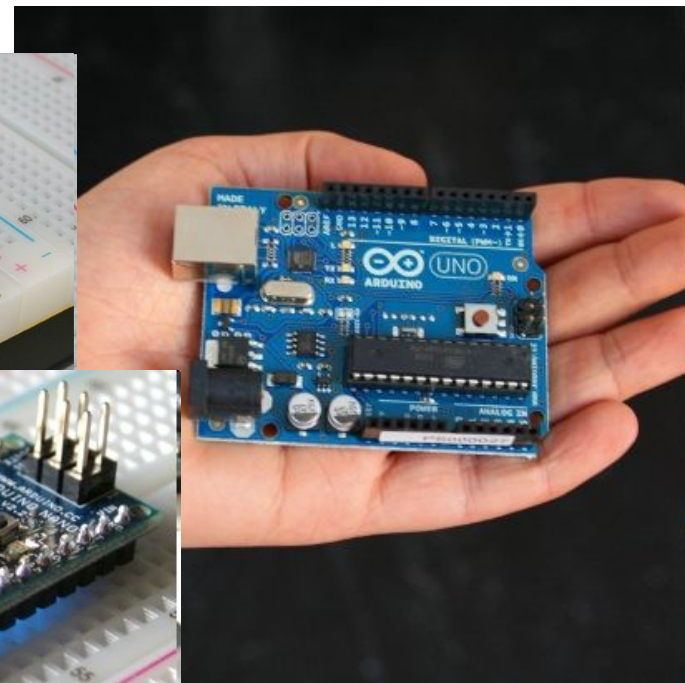
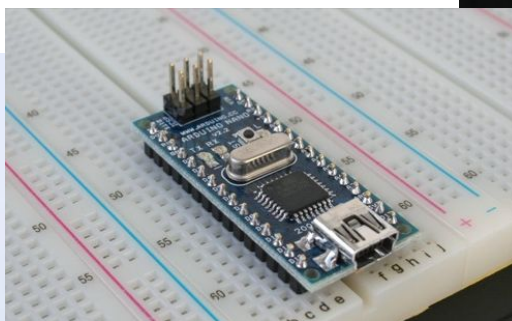
начало 26.06.2017 г. окончание 23.07.2017 г.
(дата начала и окончания практики)

Руководитель практики в группе Канашев Е.А.
Руководитель практики от кафедры Канашев Е.А.

Челябинск
2017

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Введение в Arduino



Содержание

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Виды компьютеров

- Суперкомпьютеры
- Мейнфреймы
- Серверы
- Персональные компьютеры
- Контроллеры
- Микроконтроллеры

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Виды компьютеров

- Суперкомпьютеры
- Мейнфреймы
- Серверы
- Персональные компьютеры
- Контроллеры
- Микроконтроллеры



Области использования МК

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Параметры МК

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Содержание

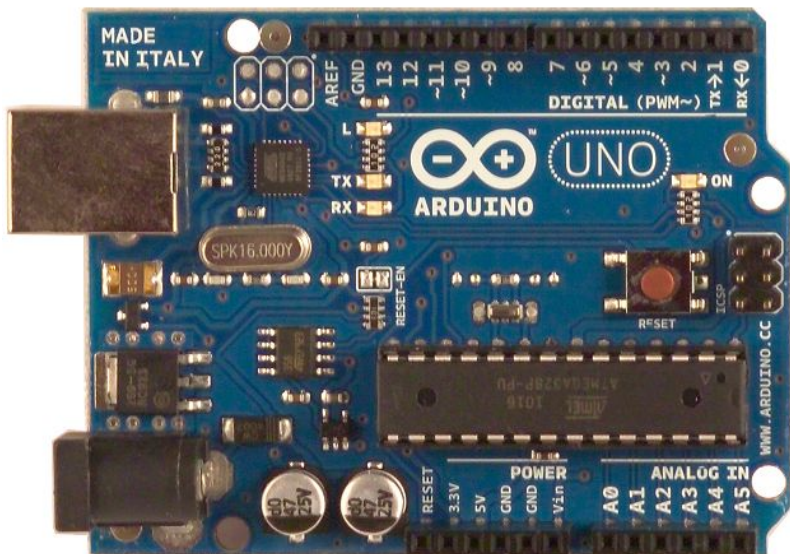
- Микроконтроллеры
- Введение в Arduino
 - Платформа Arduino
 - Популярность платформы
 - Основные платы
 - Платы расширения
 - Плата Arduino Uno
 - Среда разработки
 - Программирование
- Основы на пальцах
- Hello, world!

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Платформа Arduino

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

- Среда разработки
- Платы



(ИИУ)

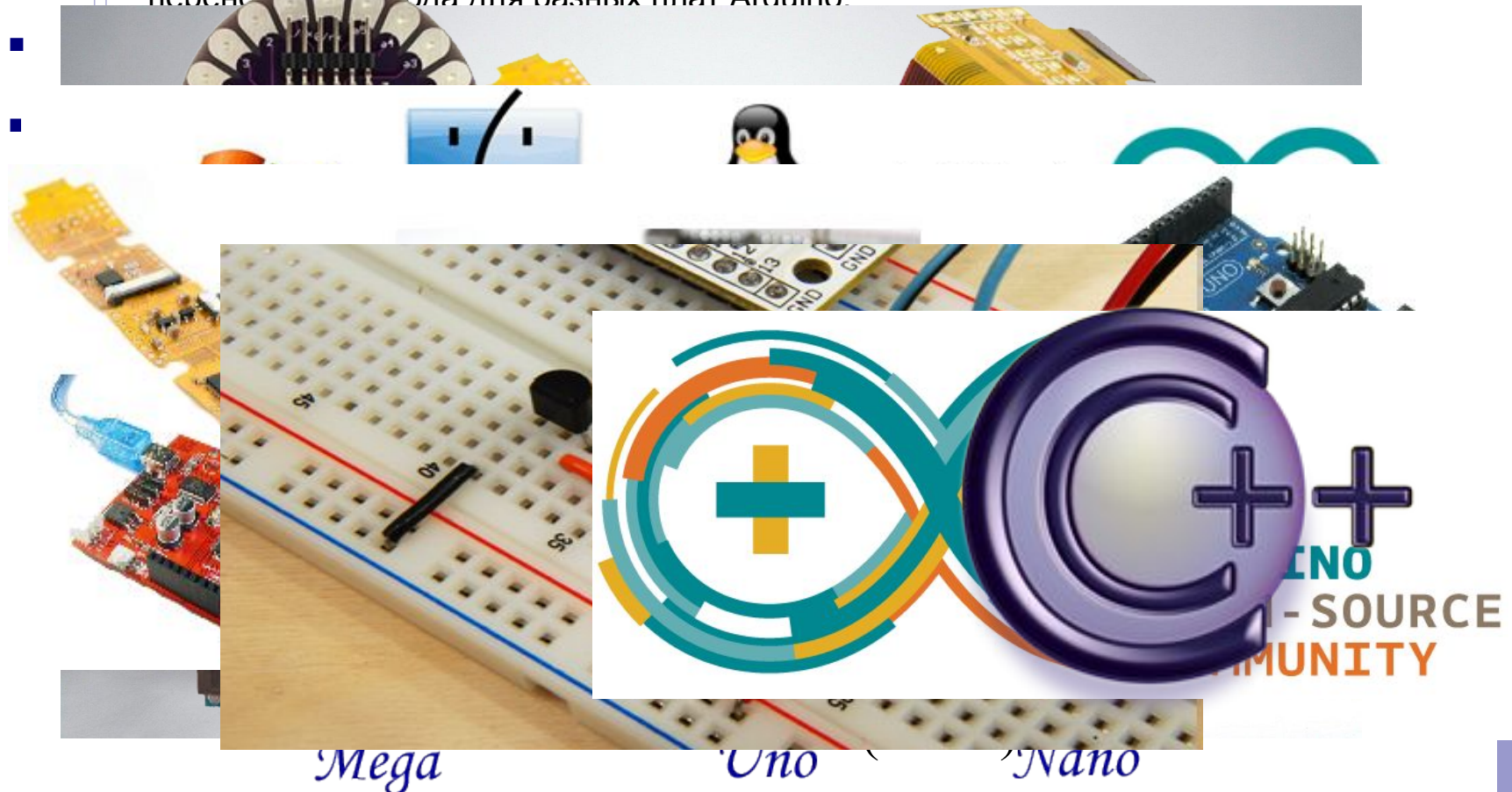
Популярность платформы

- Низкий порог входа в мир МК
- Разнообразии плат. Две версии носимых плат: *LilyPad* и *Seeeduino Film*
- Кроссплатформенность среды разработки
 - переносимость кода для разных плат Arduino.
- Отсутствие необходимости в программаторе
 - почти все платы имеют USB разъем
- Отсутствие необходимости в пайке
 - схемы собираются на беспаячной макетной плате.
- Open Source САПР системы для создания схем с платами Arduino
 - кроссплатформенные
 - бесплатные
- Язык программирования C/C++

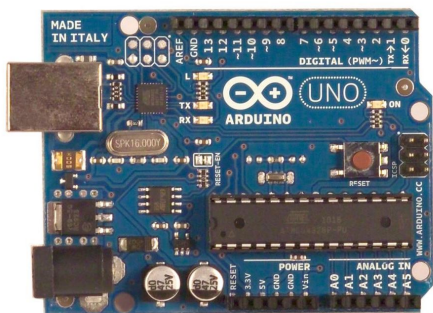
© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Популярность платформы

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



Основные платы

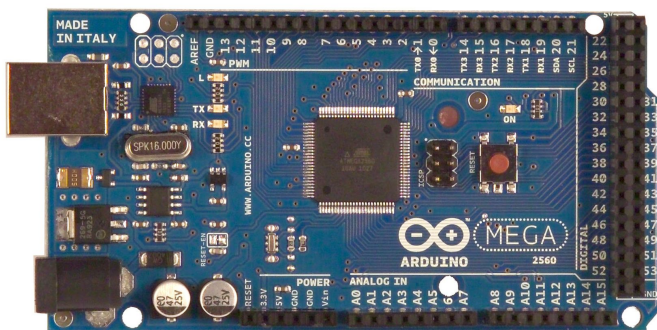


■ Uno

- 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



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Платы расширения



GSM+GPS



GSM+GPS



LED LCD



Bluetooth

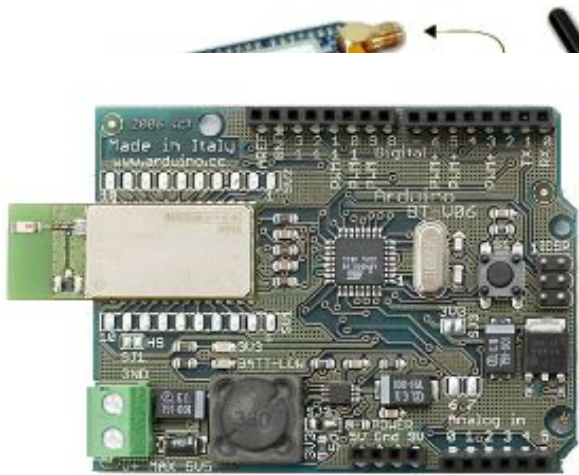


Ethernet



XBee (НИУ)

Платы расширения



GSM+GPS



Ethernet+GPS



LCD

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Плата Arduino Uno

СВЕТОДИОД НА
ВЫВОДЕ №13

ЗЕМЛЯ

ЦИФРОВЫЕ ВЫВОДЫ (ШИМ ~)

РАЗЪЕМ USB
5B

КВАРЦЕВЫЙ
РЕЗОНАТОР 16 МГц

КНОПКА
ПЕРЕЗАГРУЗКИ

РАЗЪЕМ ПИТАНИЯ 7-12В
ЦЕНТР +

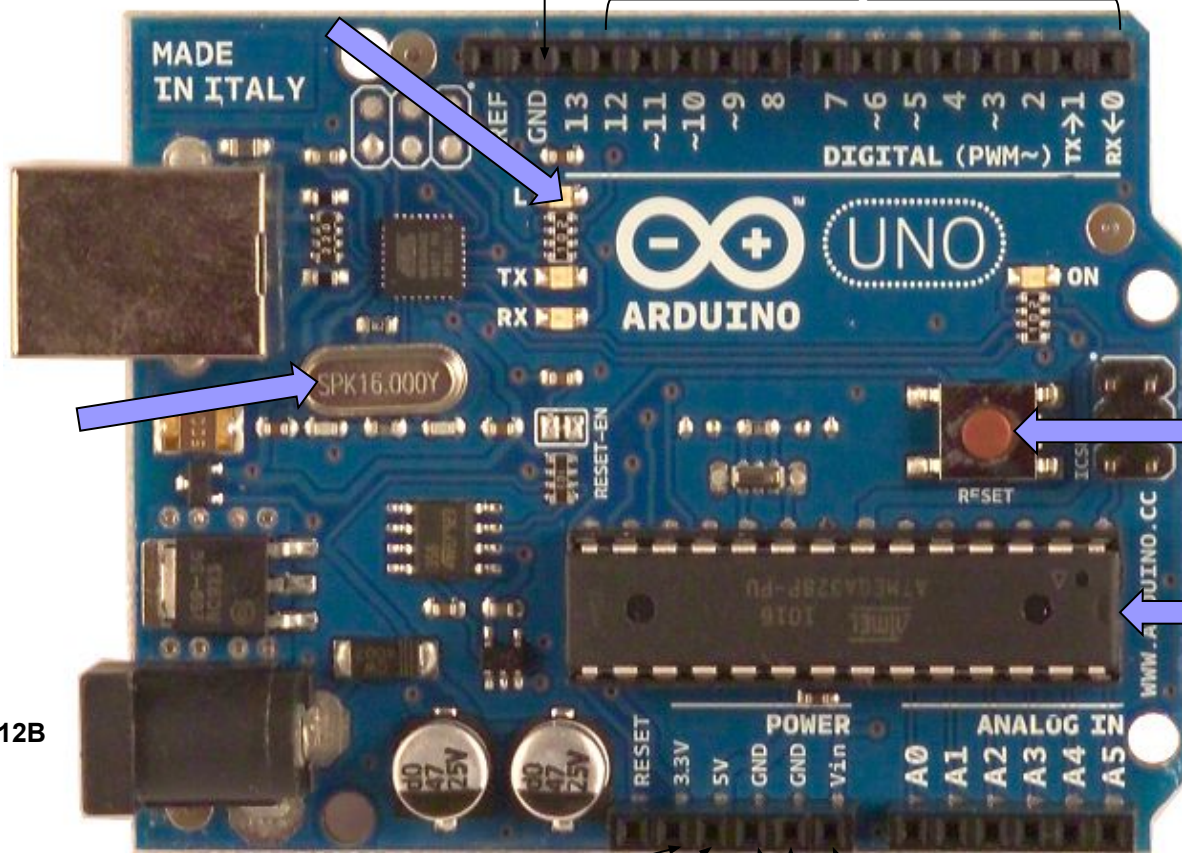
МК ATmega328

ВЫХОД
3.3В +

ВЫХОД
5В +

АНАЛОГОВЫЕ ВХОДЫ

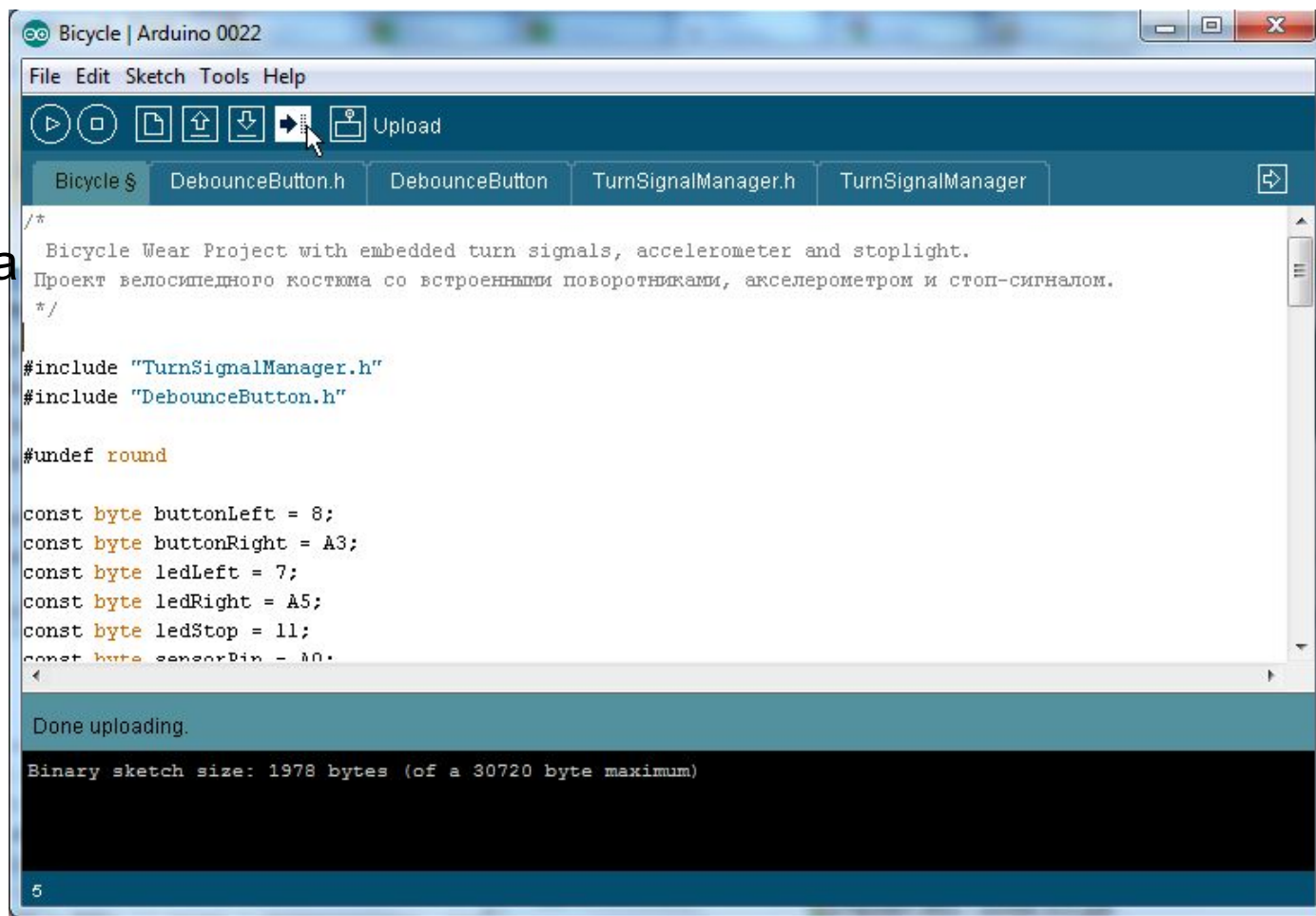
ЗЕМЛЯ
ВХОД +



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Среда разработки

Выбор платы
Выбор COM-порта
Прошивка



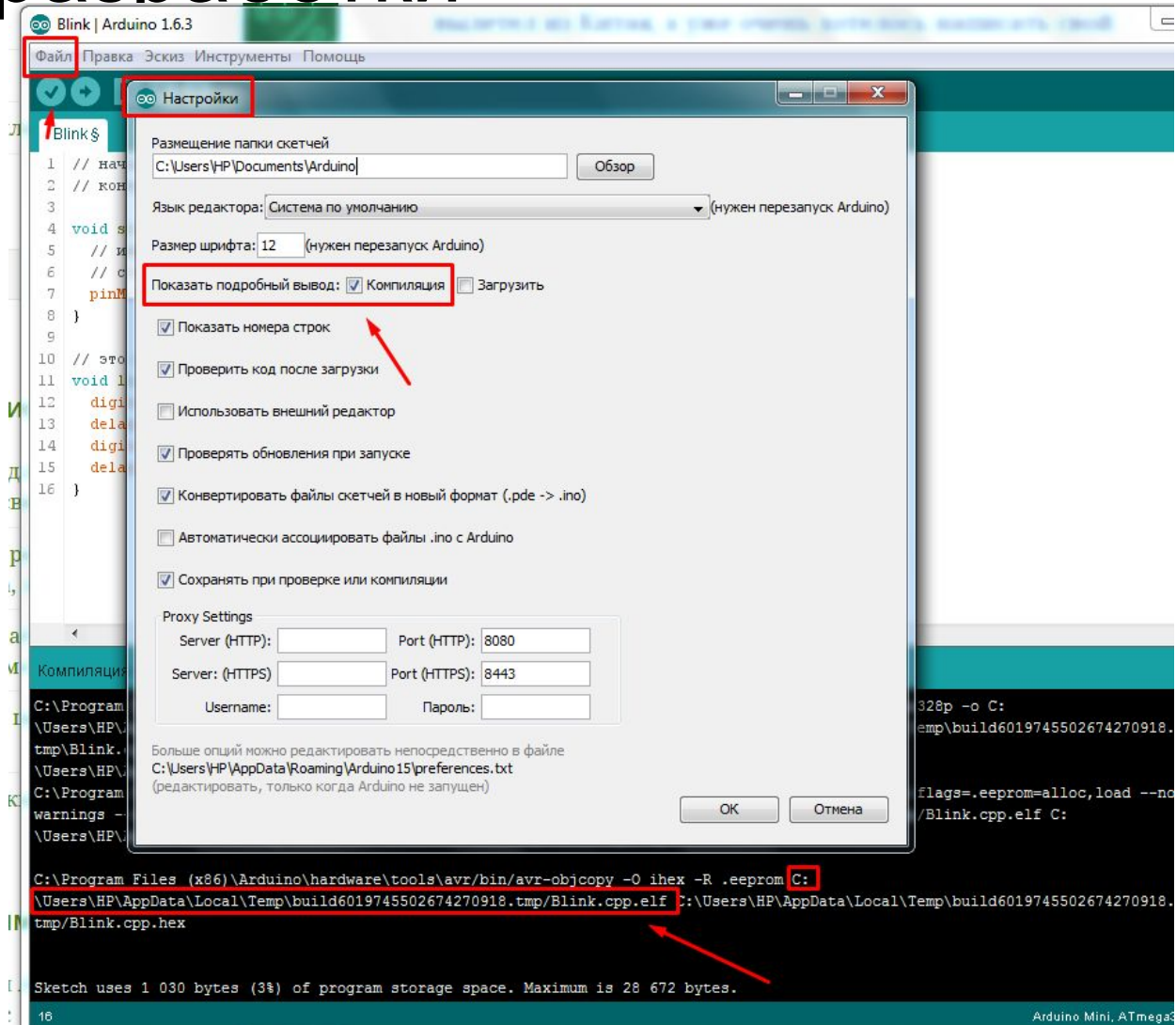
```
// подтягиваем ноги МК к 5В через внутренний подтягивающий резистор  
digitalWrite(buttonLeft, HIGH);  
digitalWrite(buttonRight, HIGH);
```

Arduino NG or older w/ ATmega168

Arduino NG or older w/ ATmega8

Среда разработки

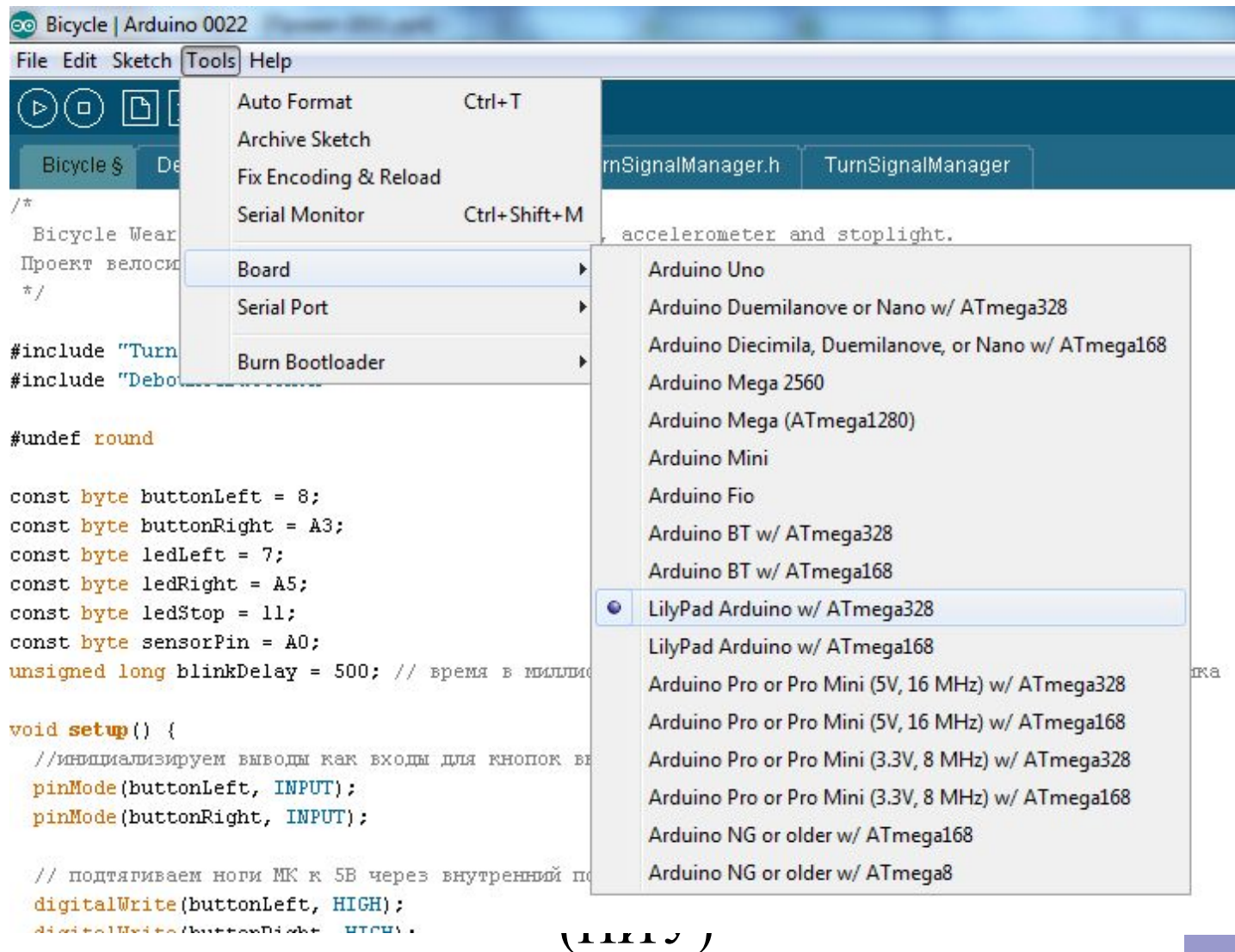
Настройка



(ПКУ)

Среда разработки

Выбор платы



The screenshot shows the Arduino IDE interface. The 'Tools' menu is open, and the 'Board' option is selected, displaying a list of available boards. The 'LilyPad Arduino w/ ATmega328' board is highlighted. The background shows a sketch with C++ code for a bicycle wear sensor.

```
File Edit Sketch Tools Help
Bicycle $ De
/*
Bicycle Wear
Проект велосипедного датчика износа покрышки
*/
#include "TurnSignalManager.h"
#include "Debounce.h"

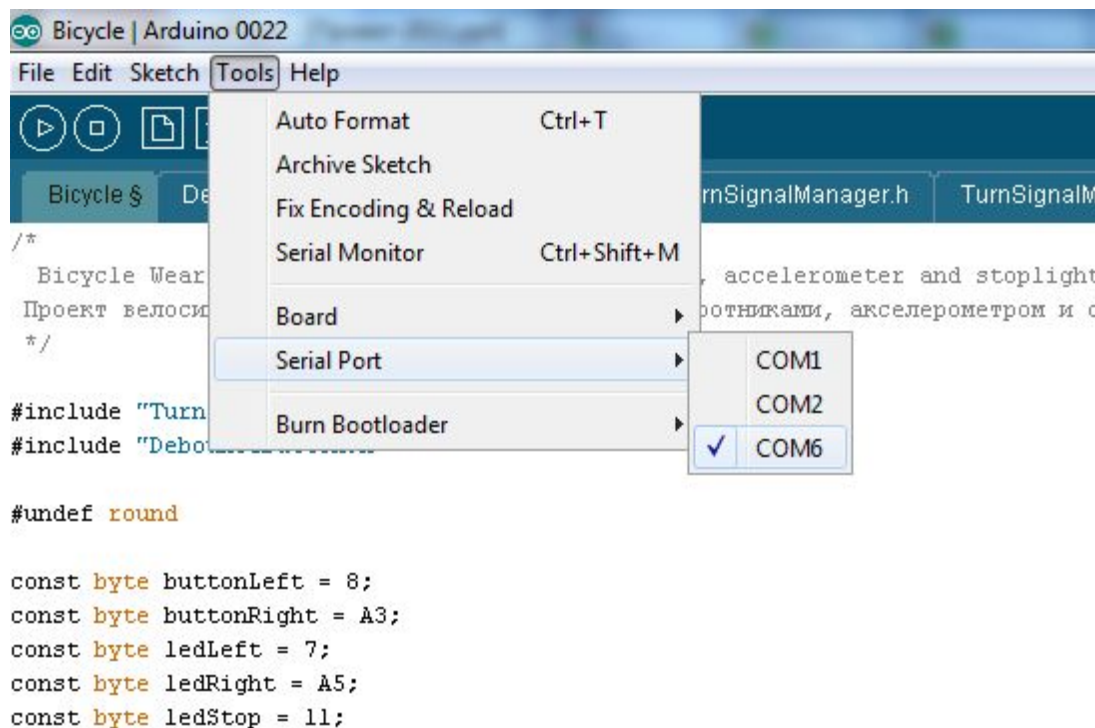
#define PIN_BUTTON_LEFT 8
#define PIN_BUTTON_RIGHT 9
#define PIN_LED_LEFT 7
#define PIN_LED_RIGHT 10
#define PIN_LED_STOP 11
#define PIN_SENSOR_PIN A0
#define BLINK_DELAY 500 // время в миллисекундах

void setup() {
  //инициализируем выводы как входы для кнопок
  pinMode(PIN_BUTTON_LEFT, INPUT);
  pinMode(PIN_BUTTON_RIGHT, INPUT);

  // подтягиваем ноги МК к 5В через внутренний подтягивающий резистор
  digitalWrite(PIN_BUTTON_LEFT, HIGH);
  digitalWrite(PIN_BUTTON_RIGHT, HIGH);
}
```


Среда разработки

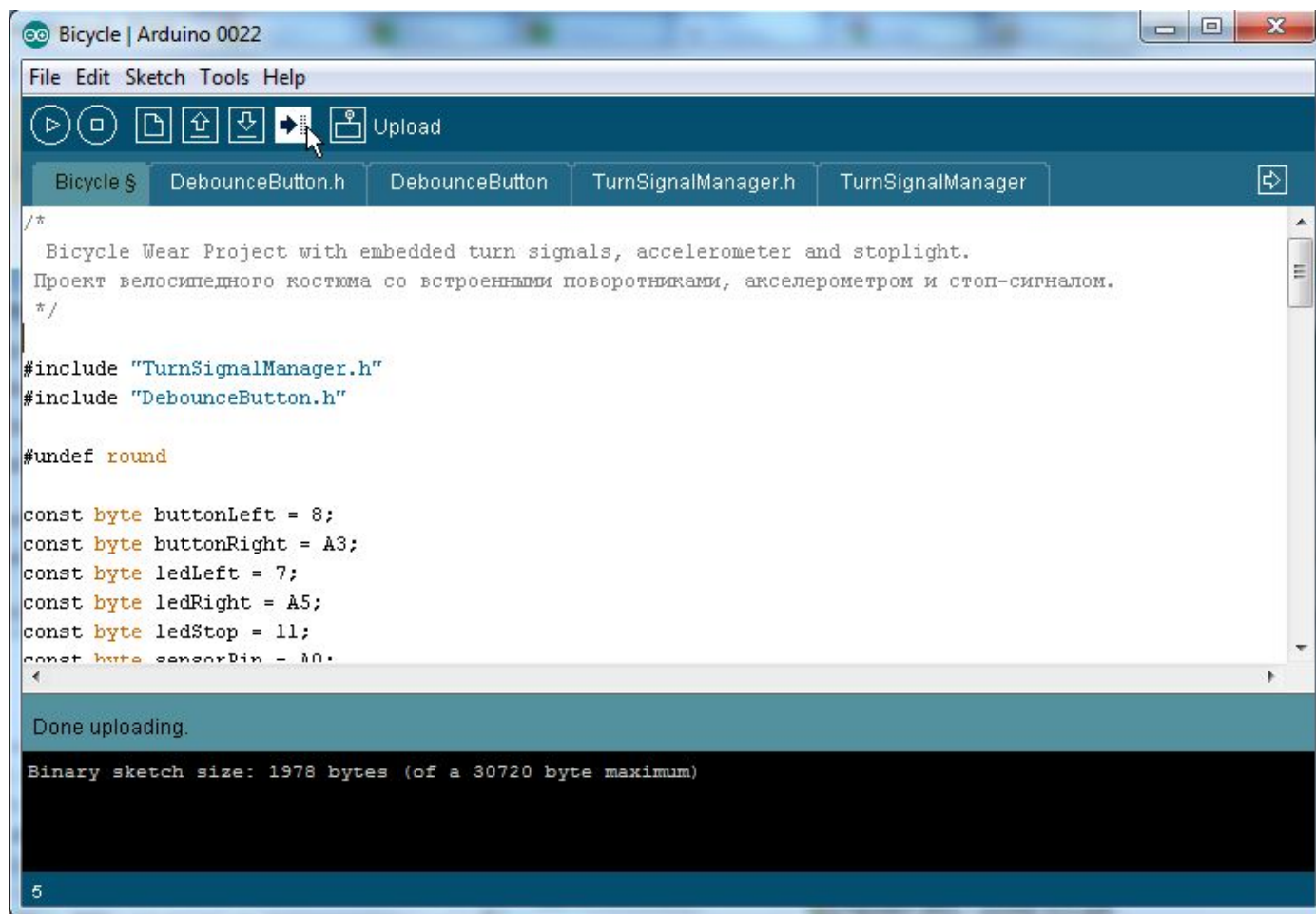
Выбор COM-порта



© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Среда разработки

Прошивка



(НИУ)

Программирование

- Язык C/C++
 - используется компилятор AVR-GCC
- Программы, называются *наброски* (или *скетчи* – варваризм от англ. sketch)
 - сохраняются в файлах с расширением `.ino`
 - обрабатываются препроцессором Arduino
 - **не содержат** функцию `main()`

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Программирование

```
int main(void)
{
    init();
    initVariant();
    #if defined(USBCON)
        USBDevice.attach();
    #endif
    setup();
    for (;;)
    {
        loop();
        if (serialEventRun) serialEventRun();
    }
    return 0;
}
```

setup() – функция вызывается однократно при старте микроконтроллера

loop() – функция вызывается после setup() в бесконечном цикле все время работы микроконтроллера

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

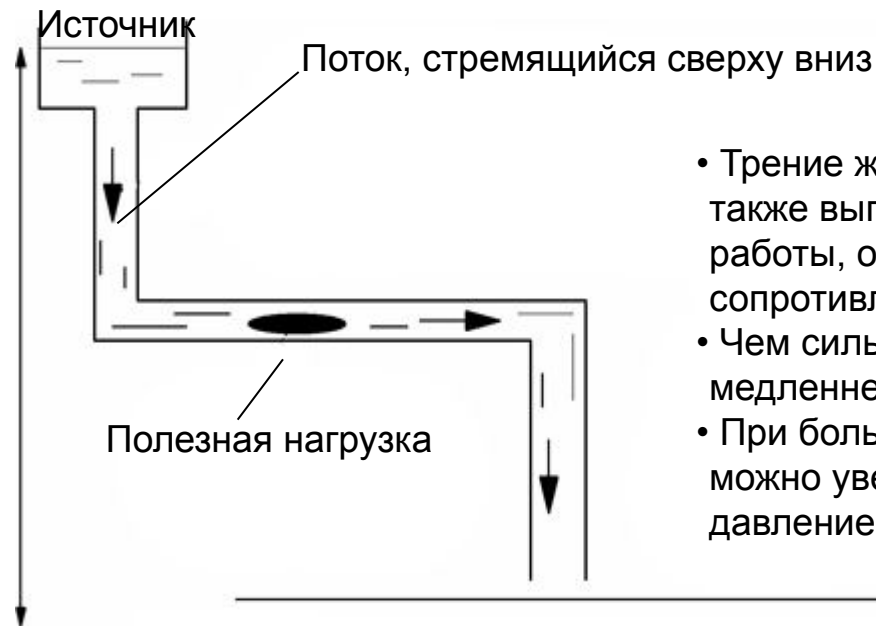
Содержание

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Закон Ома

- Сила тока в цепи прямо пропорциональна напряжению и обратно пропорциональна полному сопротивлению цепи

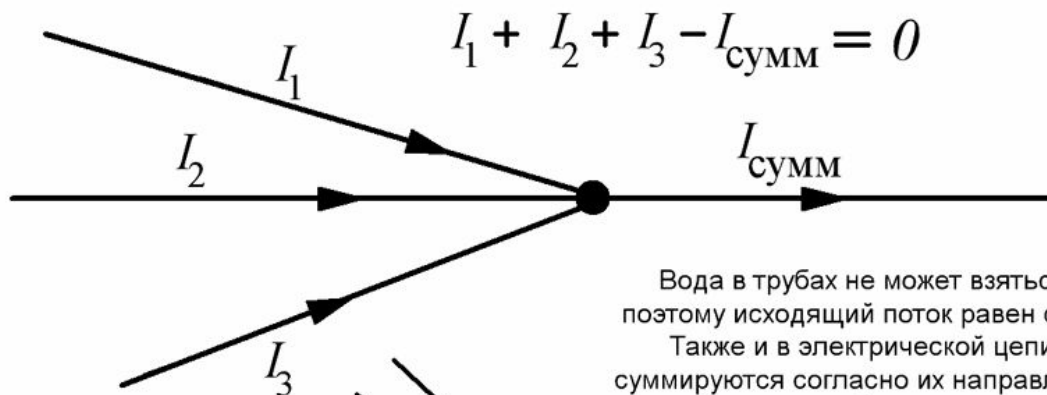
$$I = U / R$$

- U – величина напряжения, [В]
- R – сумма всех сопротивлений, [Ом]
- I – протекающий по цепи ток, [А]

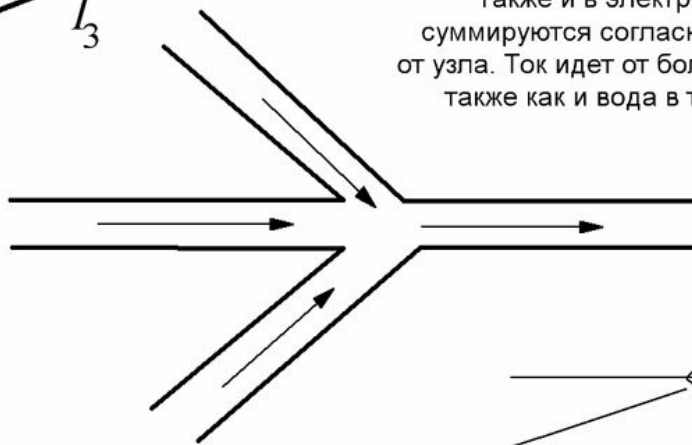
© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Закон Кирхгофа

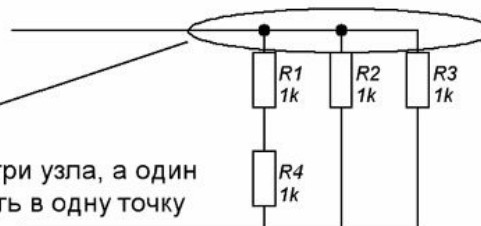
$$\sum_{j=1}^n I_j = 0$$



Вода в трубах не может взяться 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.

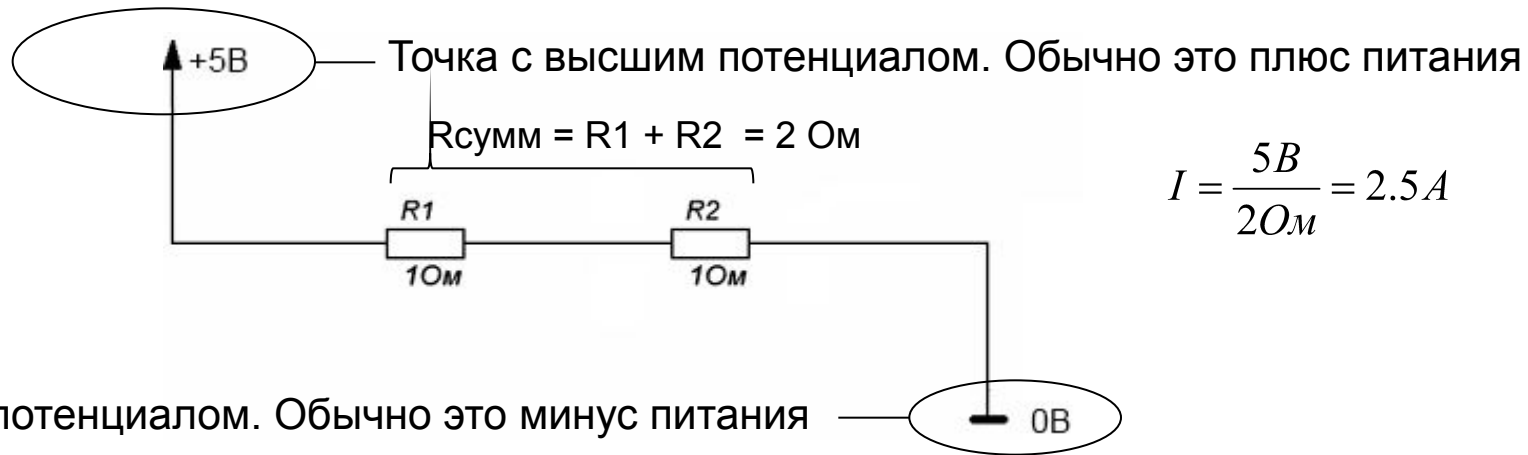


Понятие узла весьма условное. Например тут не три узла, а один так как их без проблем можно стянуть в одну точку

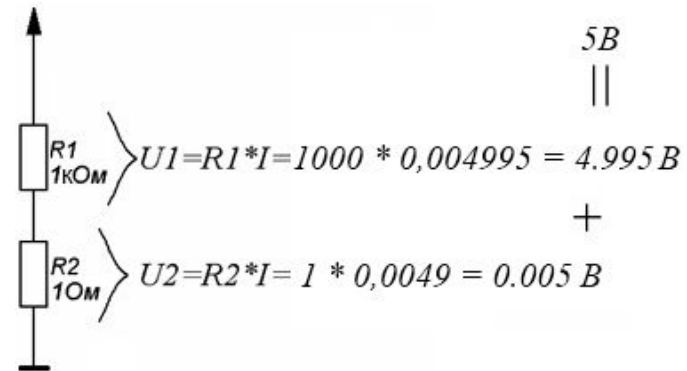
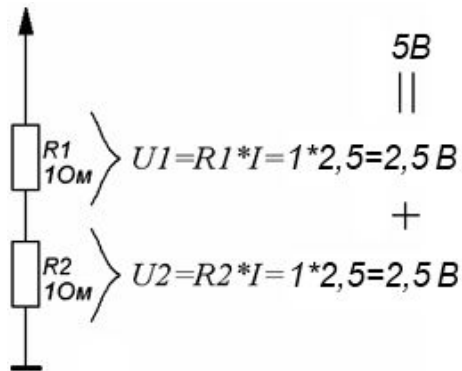


Закон Ома на практике

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

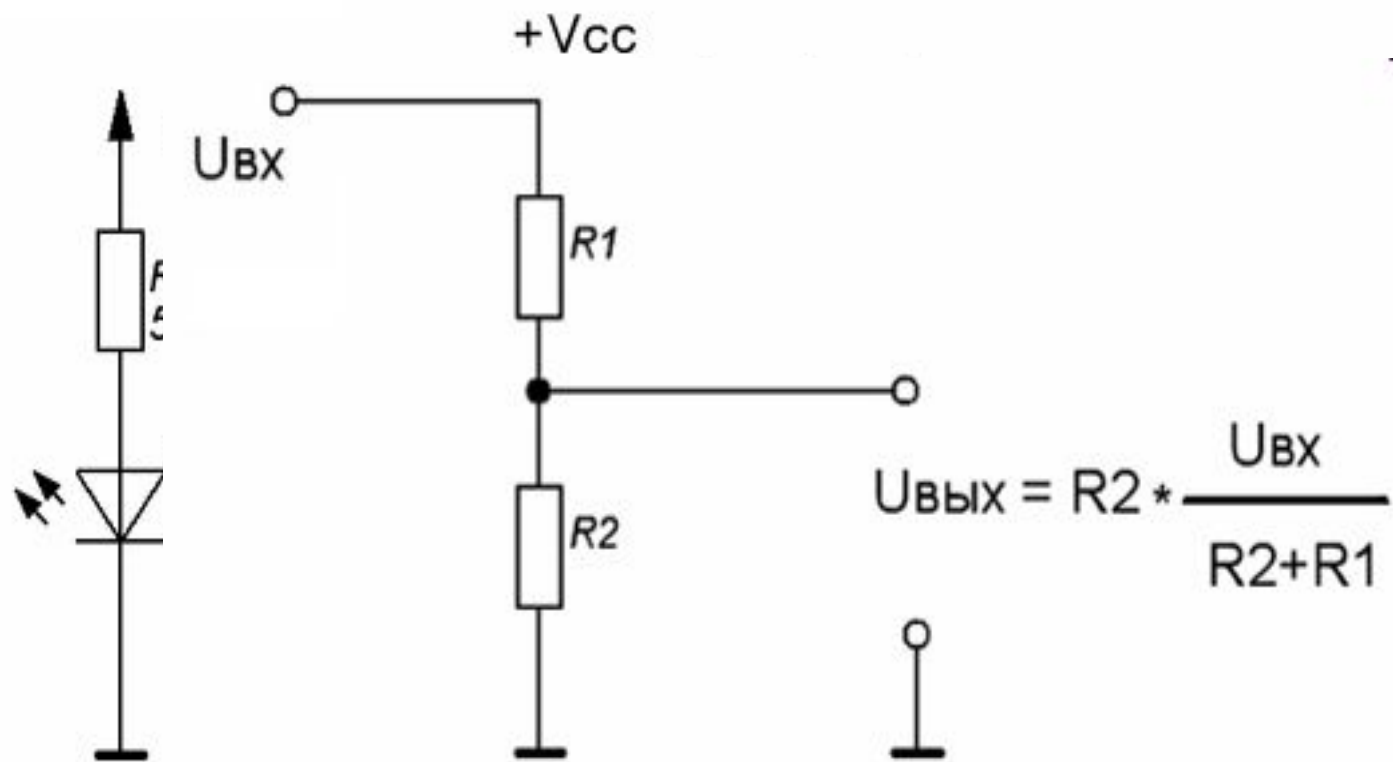


Распределение напряжения в зависимости от сопротивления:



Ф1А0У В0 Ю ур1 у
(НИУ)

Резистор



ОПростейший делитель напряжения

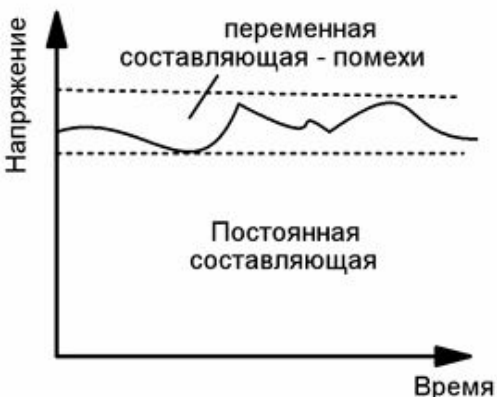
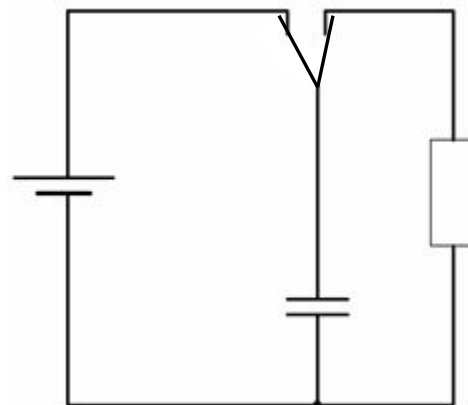
19

Конденсатор

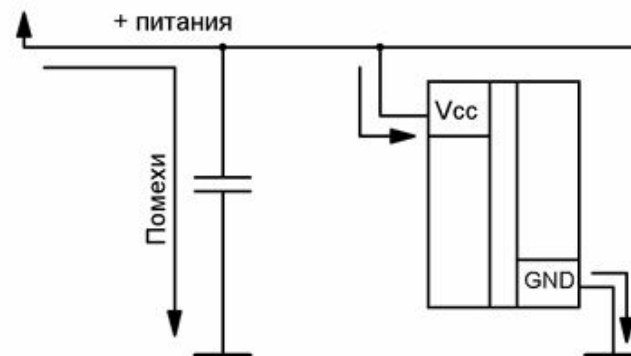
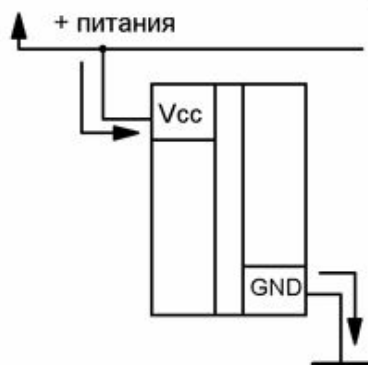
Конденсатор, словно упругий бак, когда наполняется, то запасает давление-напряжение. И может высвободить его в любой момент, лишь только ослабнет давление.

Сейчас конденсатор заряжается от источника

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

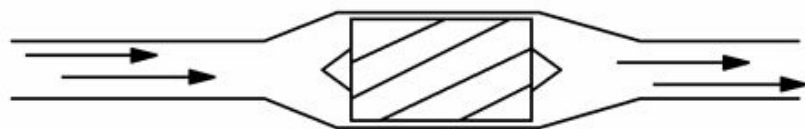


Конденсатора нет, и все напряжение вместе с помехами идет через девайс. Что явно не идет ему на пользу

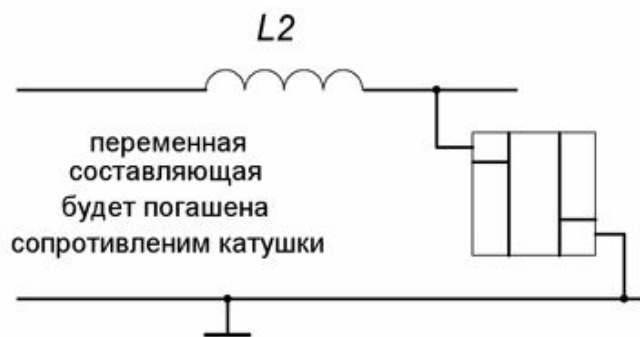
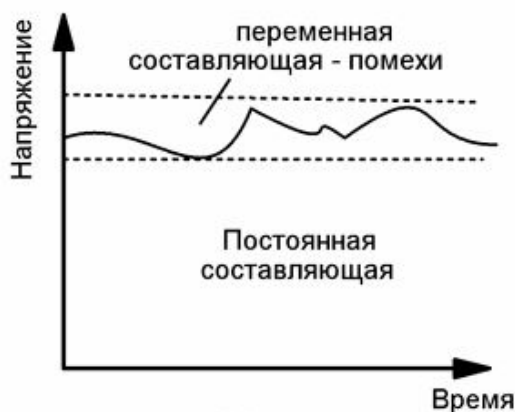
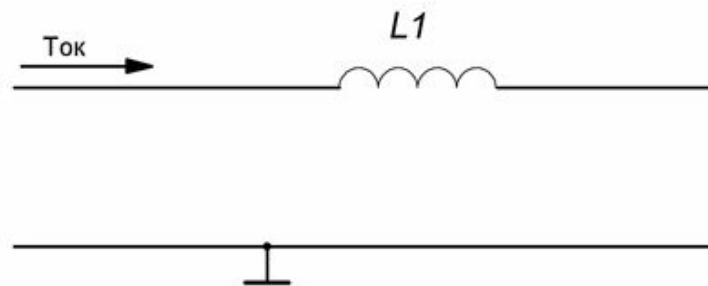


Когда кондер есть, то переменная составляющая в основном идет через него сразу на землю, т.к. сопротивление переменному току у него явно меньше, чем сопротивление девайса. Ну, а постоянная составляющая уже идет через девайс. Так как конденсатор ее никогда не пропустит через себя

Катушка индуктивности



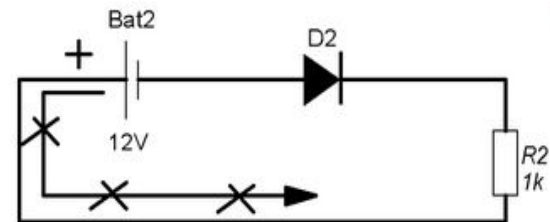
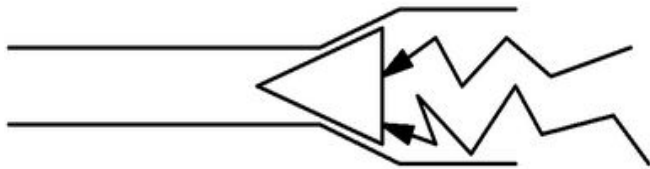
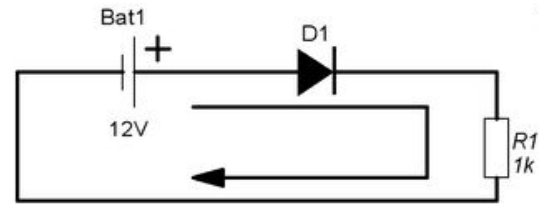
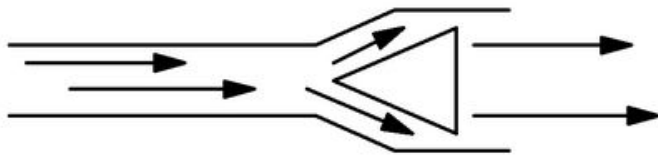
Катушка похожа на массивную турбину



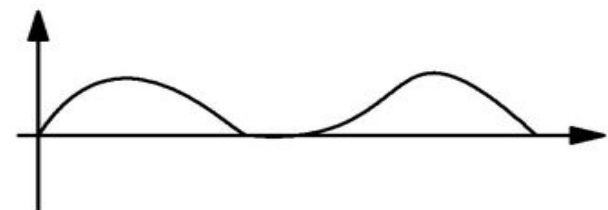
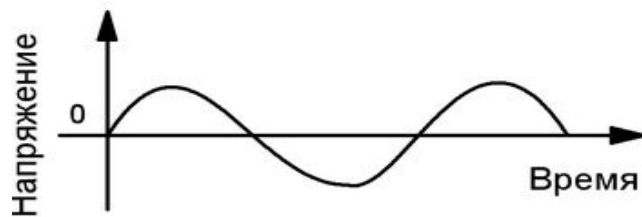
Переменная составляющая "завязнет" на индуктивном сопротивлении катушки и сильно ослабнет, а постоянная составляющая пройдет через катушку практически без потерь. В итоге, на девайсе будет почти "чистое" постоянное напряжение

ФЛАУУ ВУ Ю ур! у
(НИУ)

Диод



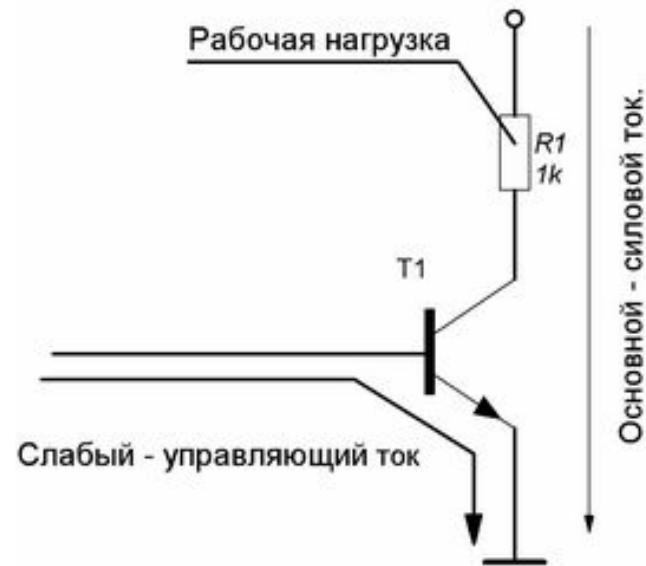
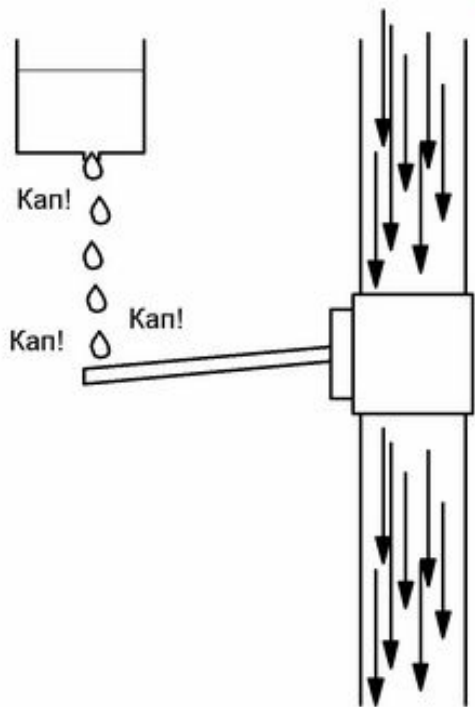
Диод - тот же ниппель, только электрический



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

(НИУ)

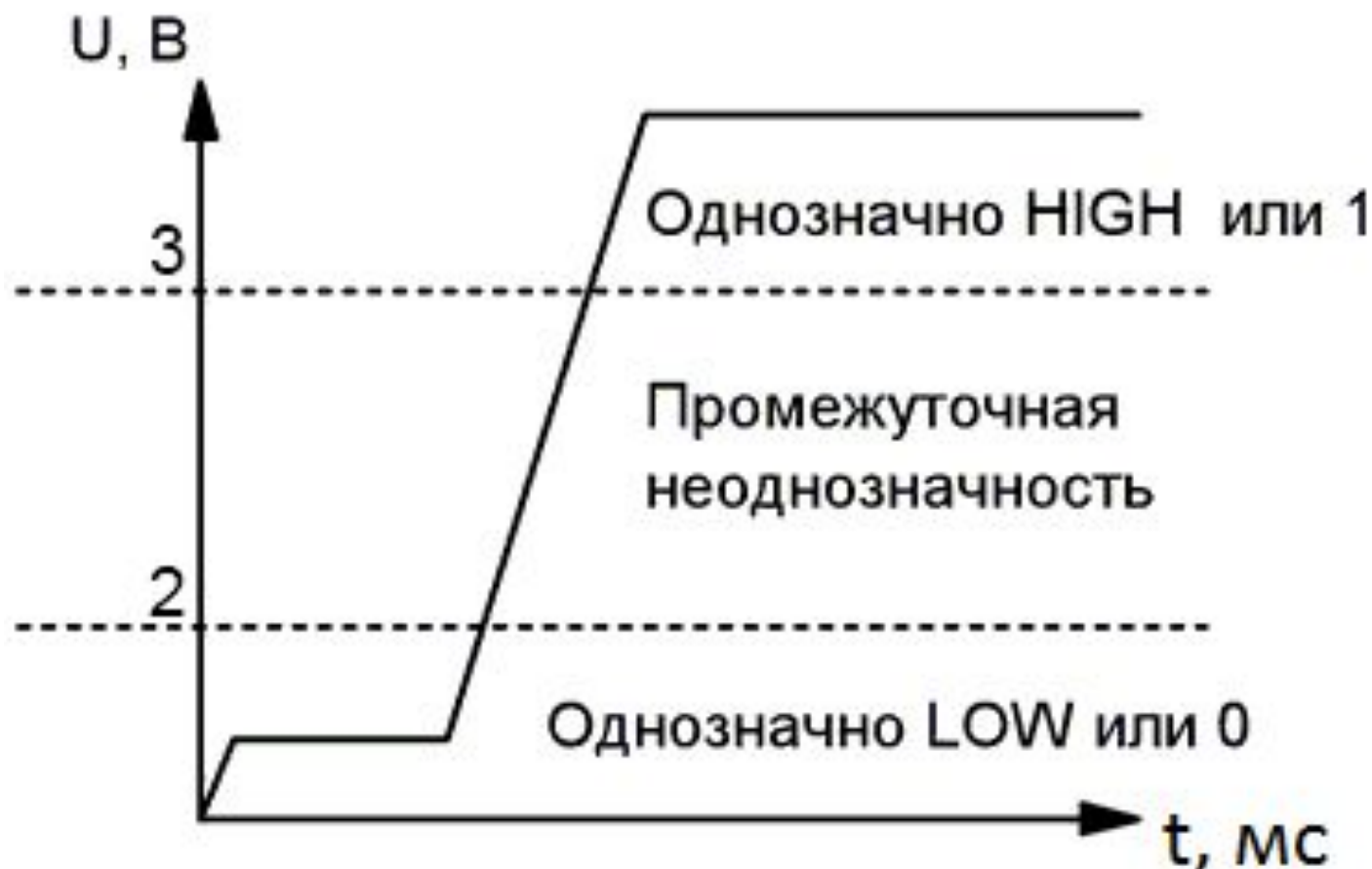
Транзистор



- 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

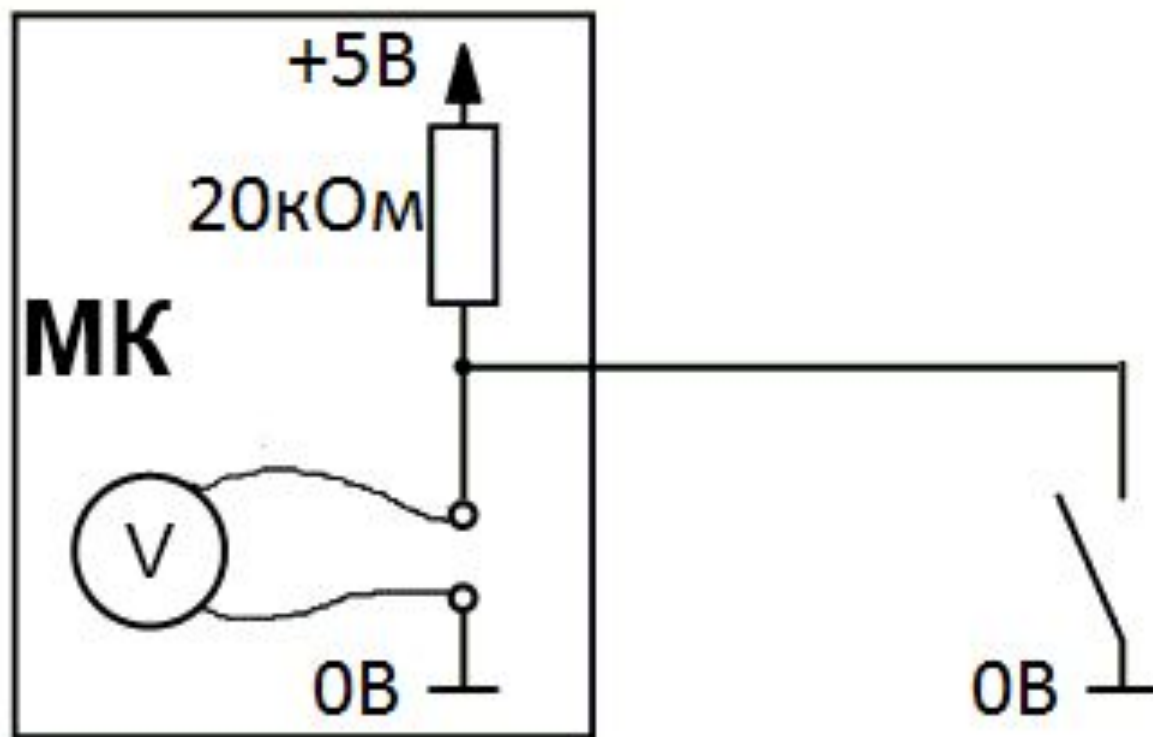
© Канашев Е.А.,
ФГАОУ ВО ЮрГУ
(НИУ)

Понятие нуля и единицы



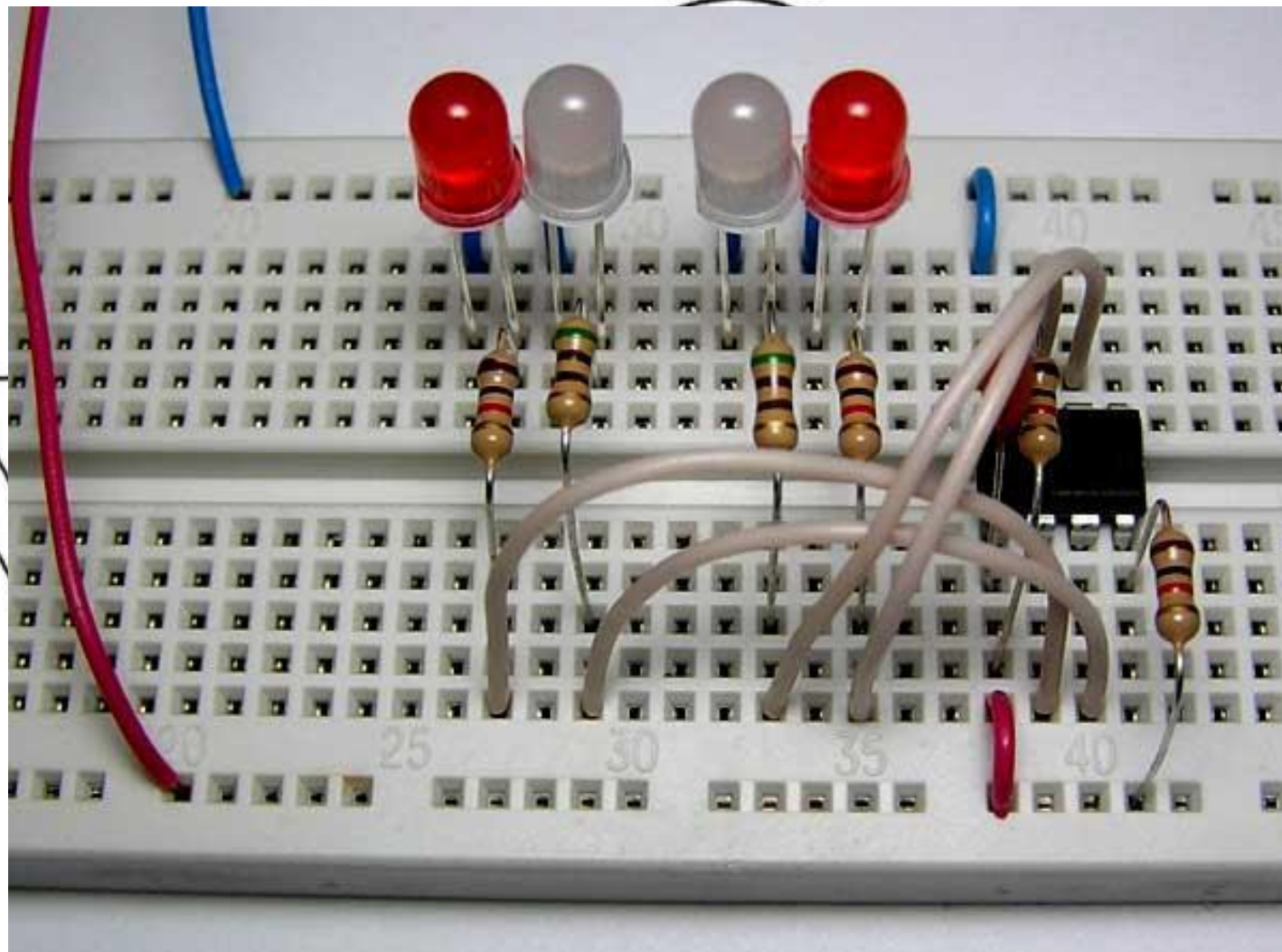
© Кашинцев Д.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Подтяжка выводов до нужного напряжения



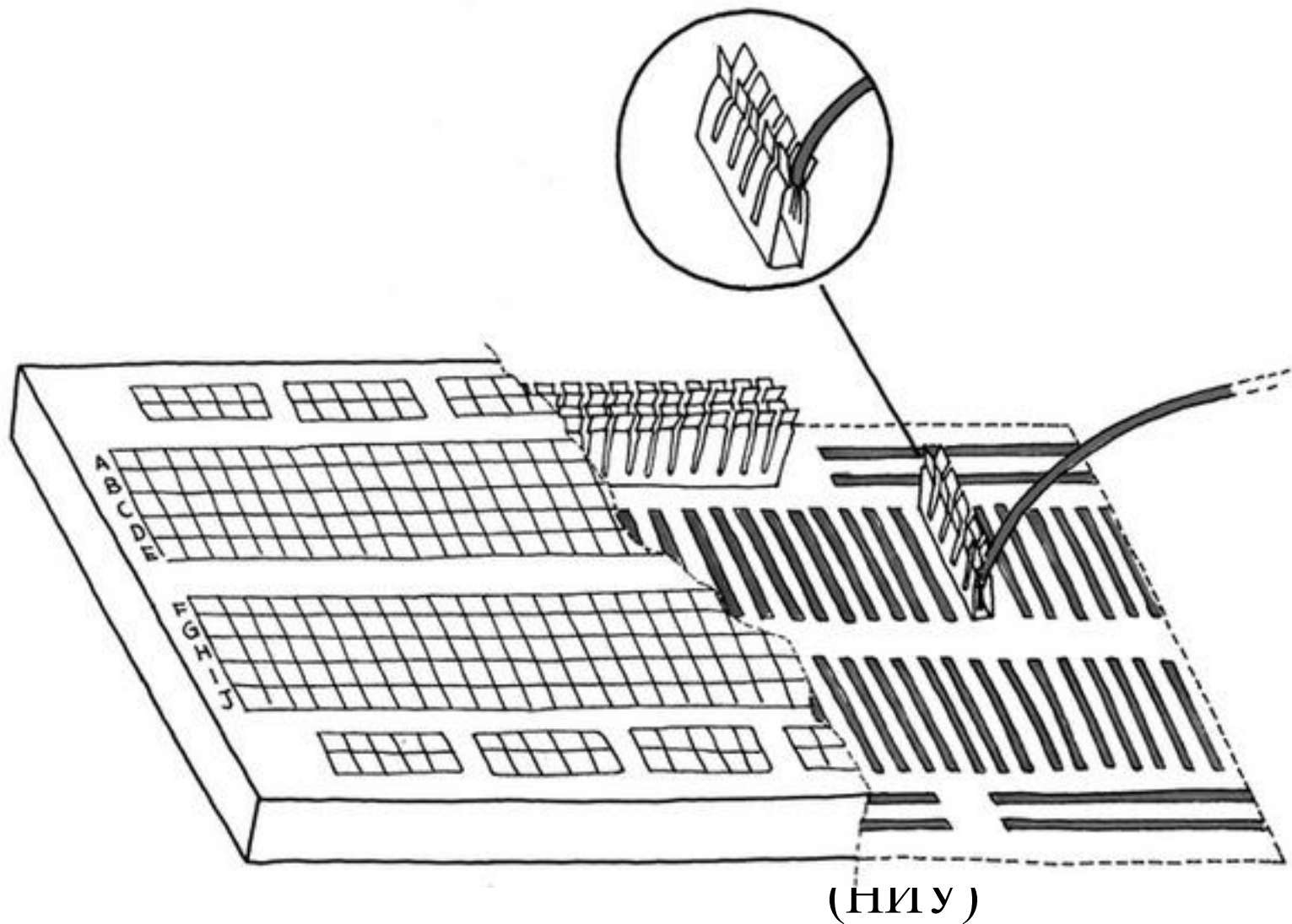
ФГАОУ ВО ЮУрГУ
(НИУ)

Обойдемся без пайки



(1111)

Обойдемся без пайки



Содержание

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Hello, world!

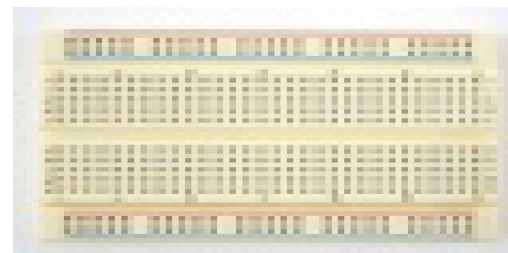
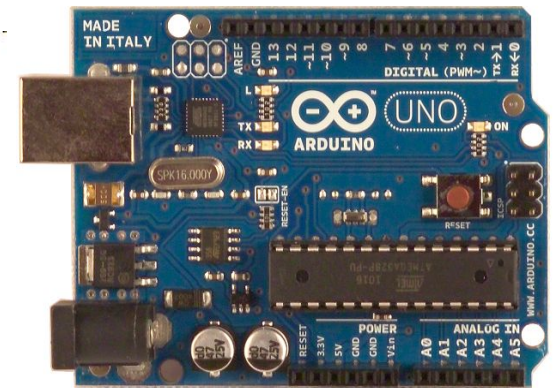
План полета

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

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

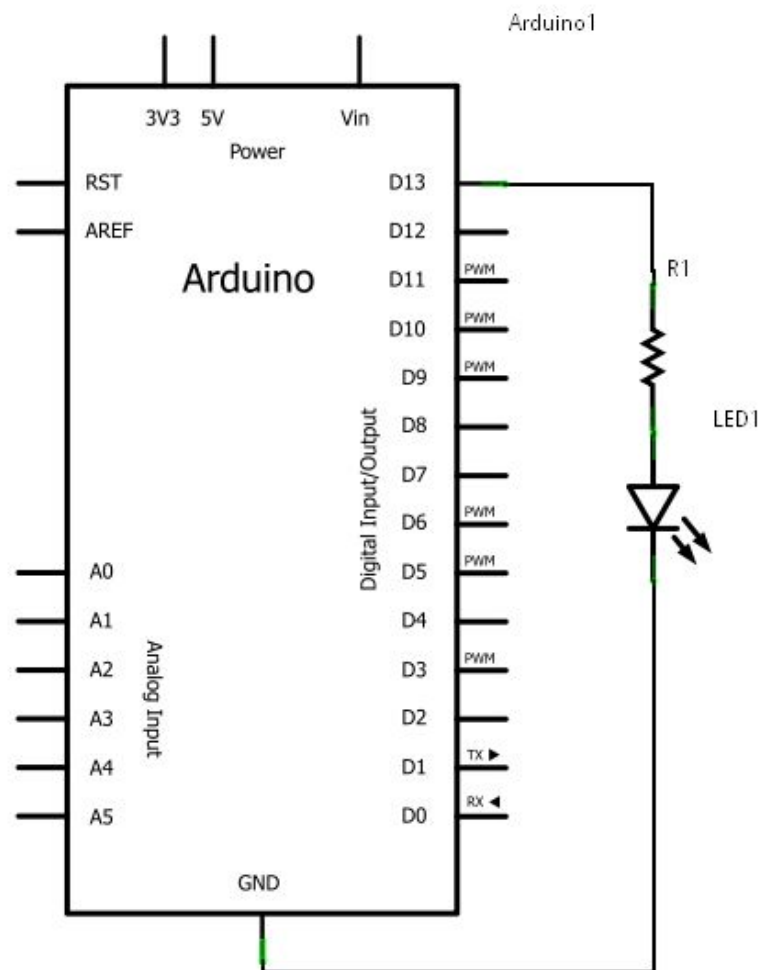
Предполетная подготовка

- 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



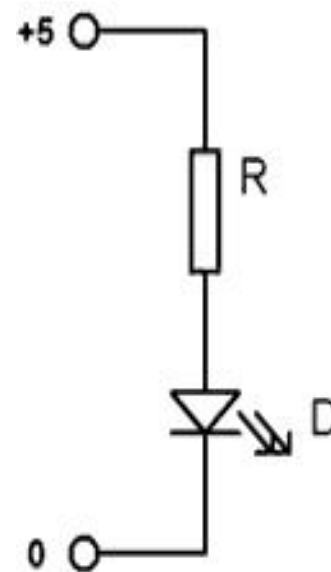
Е.А.,
ЮУрГУ

Предполетная подготовка

Расчет

Характеристики диода:

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



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

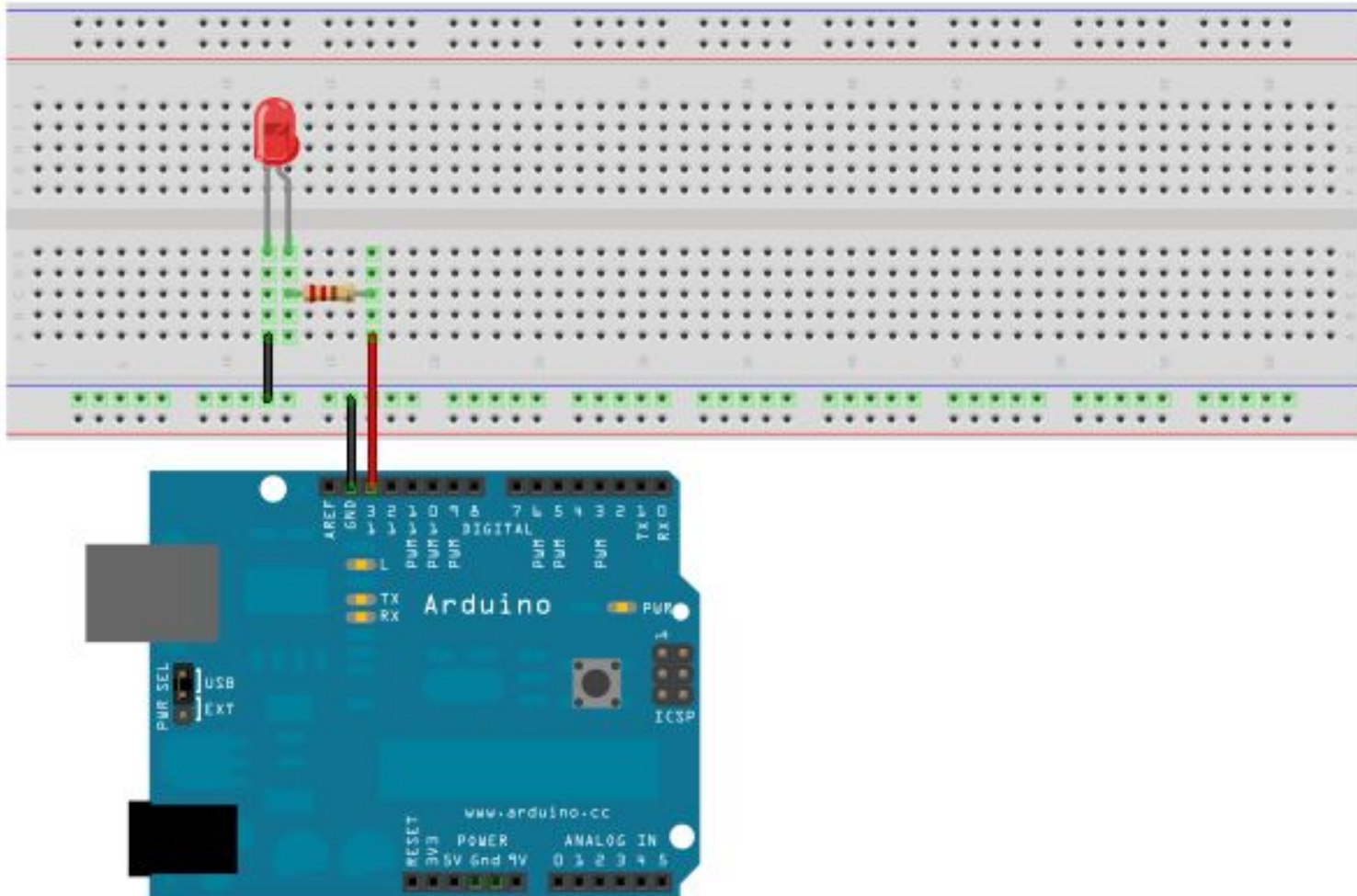
$$U_{\text{резистора}} = U_{\text{питания}} - U_{\text{светодиода}} = 5\text{В} - 2\text{В} = 3\text{В}$$

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

© Канащев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Предполетная подготовка

Макетная плата



(1 1 1 1 1 1)

Полет

```
/*  
  Зажигаем светодиод на одну секунду, затем выключаем его на  
  одну секунду в цикле.  
*/  
  
void setup() {  
  // Инициализируем цифровой вход/выход в режиме выхода.  
  // Выход 13 на большинстве плат Arduino подключен к светодиоду на плате.  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);    // зажигаем светодиод  
  delay(1000);              // ждем секунду  
  digitalWrite(13, LOW);    // выключаем светодиод  
  delay(1000);              // ждем секунду  
}
```

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Разбор полета

- Недостаток программы: если мы захотим поменять вывод №13 на другой, мы должны внести исправления в нескольких местах программы
- Решение: введем глобальную переменную, хранящую номер вывода

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Модификация 1

```
/*  
  Зажигаем светодиод на одну секунду, затем выключаем его на  
  одну секунду в цикле.  
*/  
  
int ledPin = 13; // Глобальная переменная для сохранения номер вывода  
  
void setup() {  
  pinMode(ledPin, OUTPUT); // назначить ножку ledPin, как выход  
}  
  
void loop() {  
  digitalWrite(ledPin, HIGH); // зажигаем светодиод  
  delay(1000);           // ждем секунду  
  digitalWrite(ledPin, LOW); // выключаем светодиод  
  delay(1000);           // ждем секунду  
}
```

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Разбор полета

- Недостаток программы: слишком много дублированного кода внутри цикла
- Решение: введем глобальную переменную, хранящую текущее состояние выхода

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Модификация 2

```
/*  
  Зажигаем светодиод на одну секунду, затем выключаем его на  
  одну секунду в цикле.  
*/  
  
int ledPin = 13; // Глобальная переменная для сохранения номер вывода  
boolean ledState = HIGH; // Переменная для сохранения текущего состояния  
  
void setup() {  
  pinMode(ledPin, OUTPUT); // назначить ножку ledPin, как выход  
}  
  
void loop() {  
  ledState =! ledState; // изменить состояние на противоположное  
  digitalWrite(ledPin, ledState); // задать ноге ledPin значение ledState  
  delay(1000); // ждем секунду  
}
```

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Разбор полета

- Недостаток программы: `delay(1000)` означает, что процессор 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
- Решение: исключить `delay()`

Модификация 3

```
int ledPin = 13; // Глобальная переменная для сохранения номер вывода
boolean ledState = HIGH; // Переменная для сохранения текущего состояния
long previousTimeStamp = 0; // Время последнего изменения состояния светодиода

void setup() {
  pinMode(ledPin, OUTPUT); // назначить ножку ledPin, как выход
}

void loop() {
  /* Объявляем локальную переменную currentTimeStamp, и инициализируем ее значением функции millis().
   * Функция millis() возвращает количество миллисекунд, прошедших с начала выполнения текущей
   * программы */
  long currentTimeStamp = millis();

  // Меняем состояние выхода только если текущее время отличается от времени последнего изменения
  // более чем на 1000 мс
  if (currentTimeStamp - previousTimeStamp > 1000)
  {
    previousTimeStamp = currentTimeStamp; // запомнить время изменения
    ledState = ! ledState; // изменить состояние на противоположное
    digitalWrite(ledPin, ledState); // задать ноге ledPin значение ledState
  }
}
```

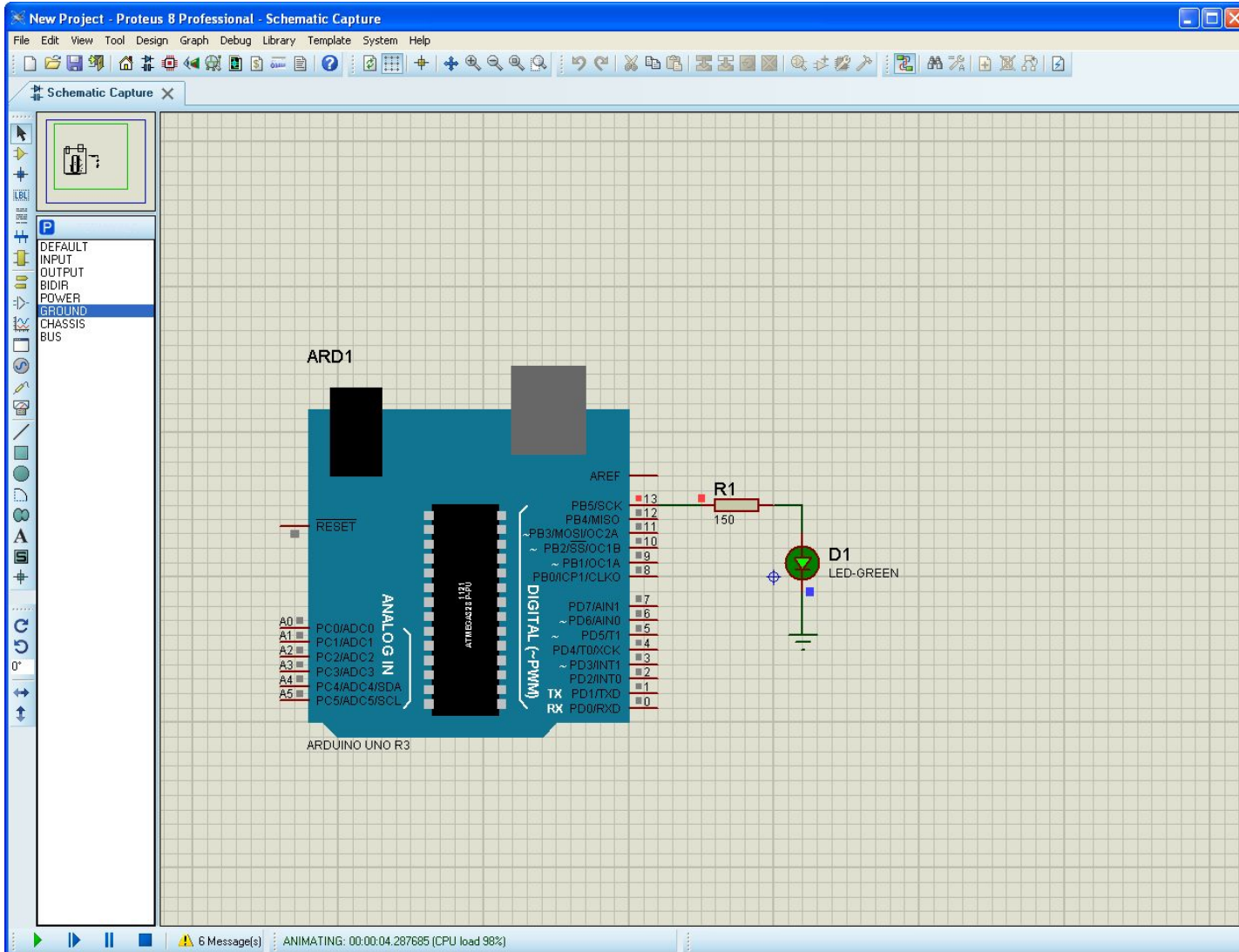
© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

ISIS Proteus

- пакет для разработки электронных устройств от схемы до печатной платы
- может симулировать различные электронные устройства
- содержит большое количество моделей компонентов
- может симулировать несколько МК в одном устройстве (AVR, PIC, Motorola, ARM)

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

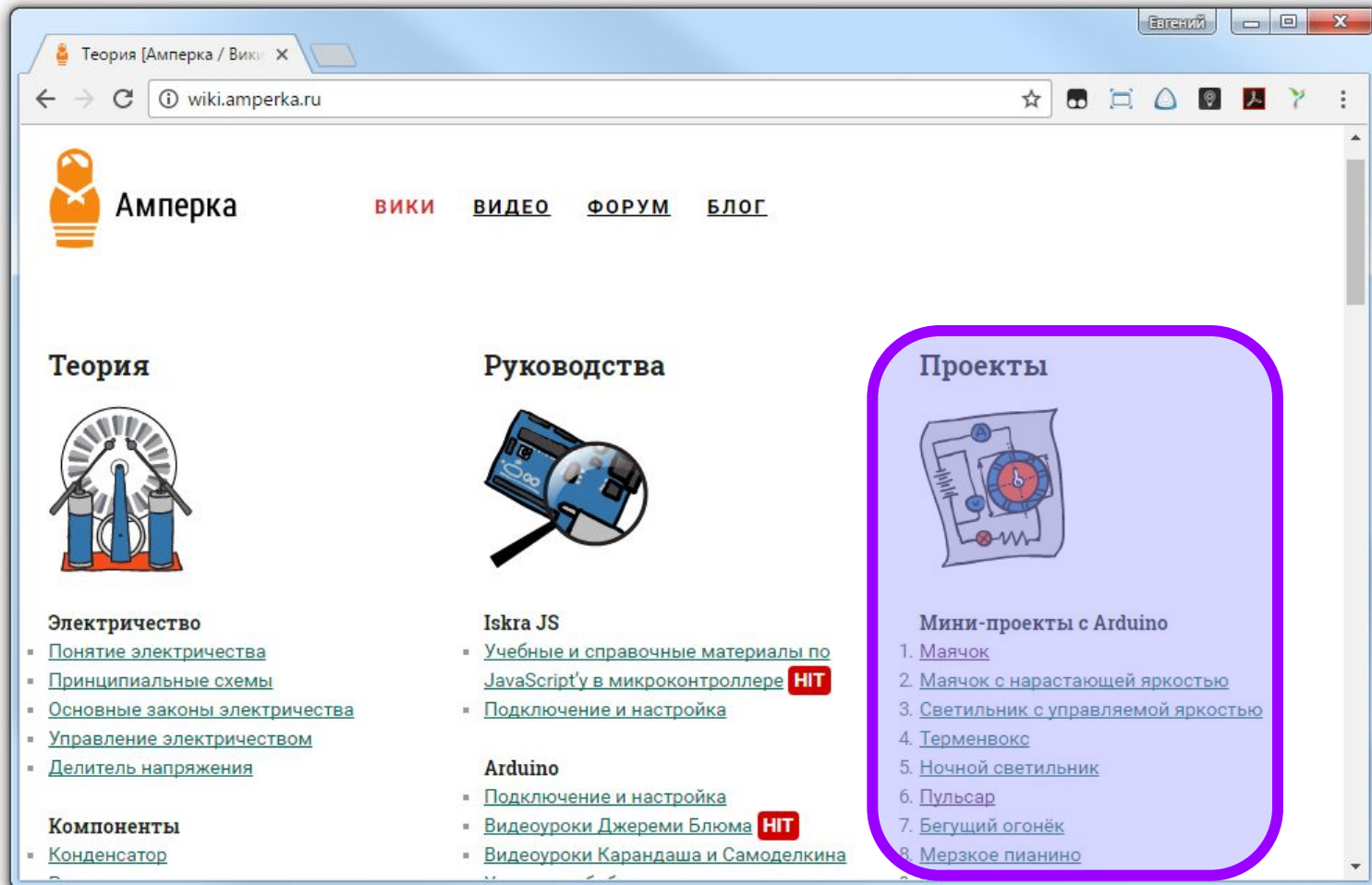
Proteus 8 Professional



(11119)

wiki.amperka.ru

■ Проекты



The screenshot shows the website wiki.amperka.ru in a browser window. The page has a navigation menu with 'ВИКИ', 'ВИДЕО', 'ФОРУМ', and 'БЛОГ'. The main content is divided into three columns: 'Теория', 'Руководства', and 'Проекты'. The 'Проекты' section is highlighted with a purple border and contains a list of mini-projects under the heading 'Мини-проекты с Arduino'.

Теория

- Электричество
 - [Понятие электричества](#)
 - [Принципиальные схемы](#)
 - [Основные законы электричества](#)
 - [Управление электричеством](#)
 - [Делитель напряжения](#)
- Компоненты
 - [Конденсатор](#)

Руководства

- Iskra JS
 - [Учебные и справочные материалы по JavaScript'у в микроконтроллере](#) **ННУ**
 - [Подключение и настройка](#)
- Arduino
 - [Подключение и настройка](#)
 - [Видеоуроки Джереми Блюма](#) **ННУ**
 - [Видеоуроки Карандаша и Самоделкина](#)

Проекты

Мини-проекты с Arduino

- [1. Маячок](#)
- [2. Маячок с нарастающей яркостью](#)
- [3. Светильник с управляемой яркостью](#)
- [4. Терменвокс](#)
- [5. Ночной светильник](#)
- [6. Пульсар](#)
- [7. Бегущий огонёк](#)
- [8. Мерзкое пианино](#)

(ННУ)

Мини-проекты с Arduino

1. Маячок
2. Маячок с нарастающей яркостью
3. Светильник с управляемой яркостью
4. Терменвокс
5. Ночной светильник
6. Пульсар
7. Бегущий огонёк
8. Мерзкое пианино
9. Миксер
10. Кнопочный переключатель
11. Светильник с кнопочным управлением
12. Кнопочные ковбои
13. Секундомер
14. Счётчик нажатий
15. Комнатный термометр
16. Метеостанция
17. Пантограф
18. Тестер батареек
19. Светильник, управляемый по USB
20. Перетягивание каната

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)

Задание на практику

- В течение семестра:
 - Выполнить все 20 проектов в Proteus
 - Оформить отчет
 - Титульный лист
 - Проект 1 – Маячок
 - Условие задачи
 - Схема электрическая принципиальная
 - Листинг программы (скетч)
 - Ответы на контрольные вопросы
 - ...
 - Проект 20 – Перетягивание каната
- На защите:
 - Иметь при себе
 - отчет
 - файлы Proteus
 - скетчи Arduino
 - Модифицировать программную и/или аппаратную часть проекта по указанию преподавателя
 - Продемонстрировать работоспособность в Proteus или Arduino



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-palcah-chast-4.html>
- http://web.media.mit.edu/~leah/LilyPad/build/turn_signal_jacket.html
- <http://sparkfun.com>
- <http://www.seeedstudio.com/>

© Канашев Е.А.,
ФГАОУ ВО ЮУрГУ
(НИУ)