

Arduino и бионика

Введение в микроконтроллеры с Arduino

Занятие 2



13 ноября 2007 - machineproject — Тод Е. Курт

Перевод на русский язык

16 сентября 2011 — robofreak.ru — Татьяна Волкова

Программа на сегодня

- Случайное поведение
- RGB-светодиоды
- Смешивание цветов
- Аналоговый вход с переменными резисторами
- Потенциометры и фоторезисторы
- Основы последовательного ввода-вывода
- Проигрывание звука пьезоизлучателем

Очень много всего. Посмотрим, как далеко мы зайдём.

Повторение: мигающий светодиод

Удостоверьтесь, что всё
по-прежнему работает

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

```
void setup() {
  pinMode(ledPin, OUTPUT); // sets t
}
void loop() {
  digitalWrite(ledPin, HIGH); // sets t
  delay(1000);                // waits
  digitalWrite(ledPin, LOW);  // sets t
  delay(1000);                // waits
}
```



КОМПИЛЯЦИЯ

Done compiling.



загрузка



TX/RX мигают



скетч
стартует

Загрузите
«File/Sketchbook/Examples/Digital/Blink»

Измените значение в «delay()», чтобы изменить частоту мигания

Известная хорошая конфигурация

Правило экспериментирования #1:

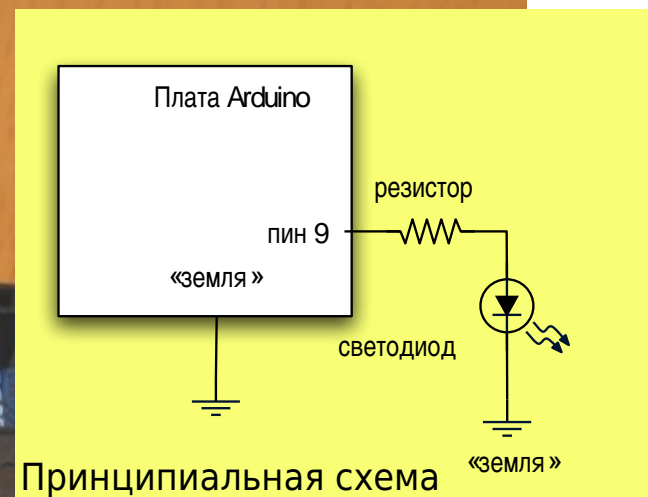
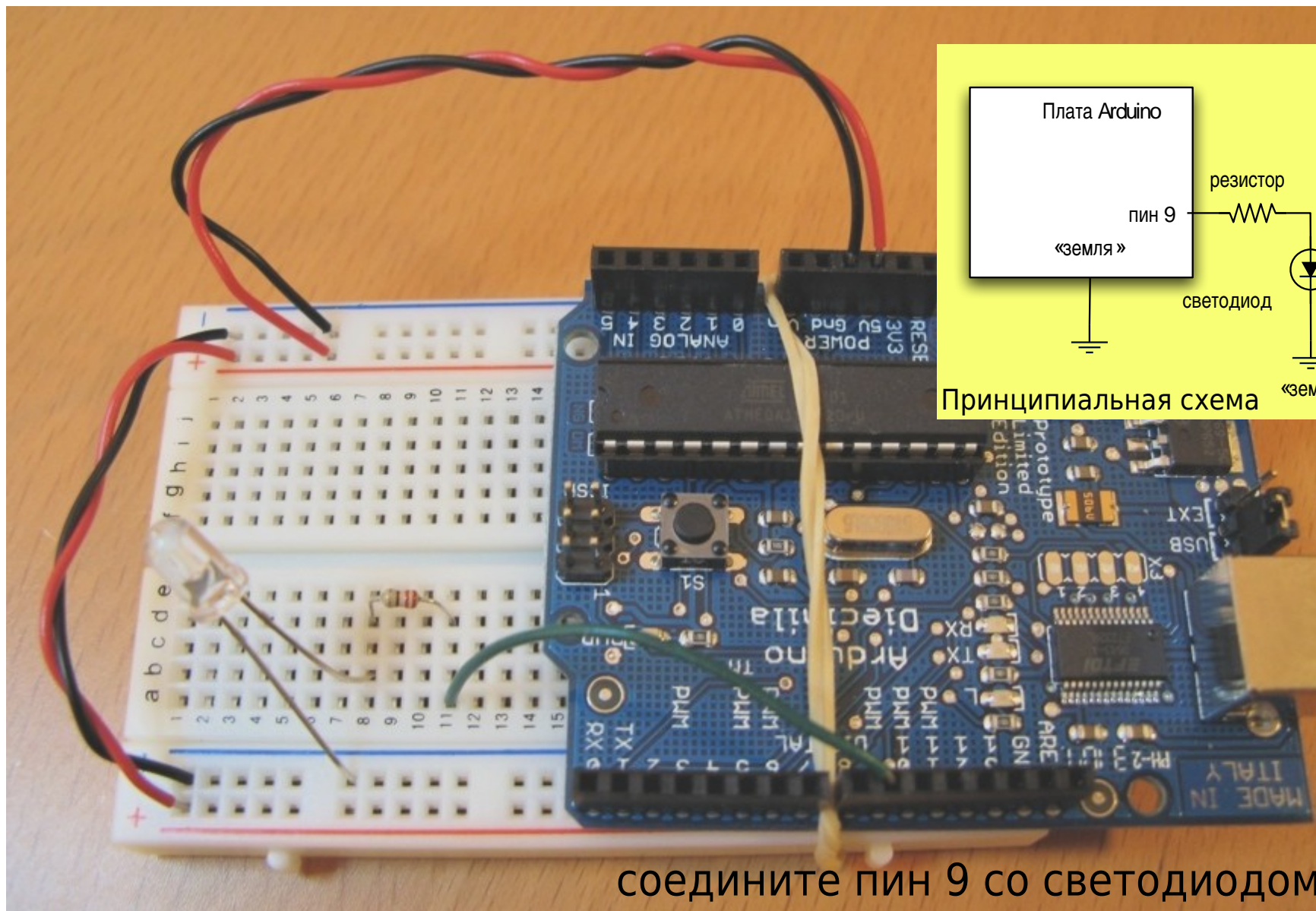
Прежде чем пробовать что-то новое

вернитесь к известному Вам
рабочему состоянию

*Так что потратьте несколько минут и
заставьте «Мигалку» снова работать*

Заставьте работать весь Ваш цикл редактирование->компиляция->загрузка->запуск
Даже если для Вас это стало уже второй натурой и Вам кажется, что это уже не требуется,
всё равно сделайте это.
Особенно когда возникают «мистические» проблемы, возвращайтесь к известному состоянию

Собираем установку



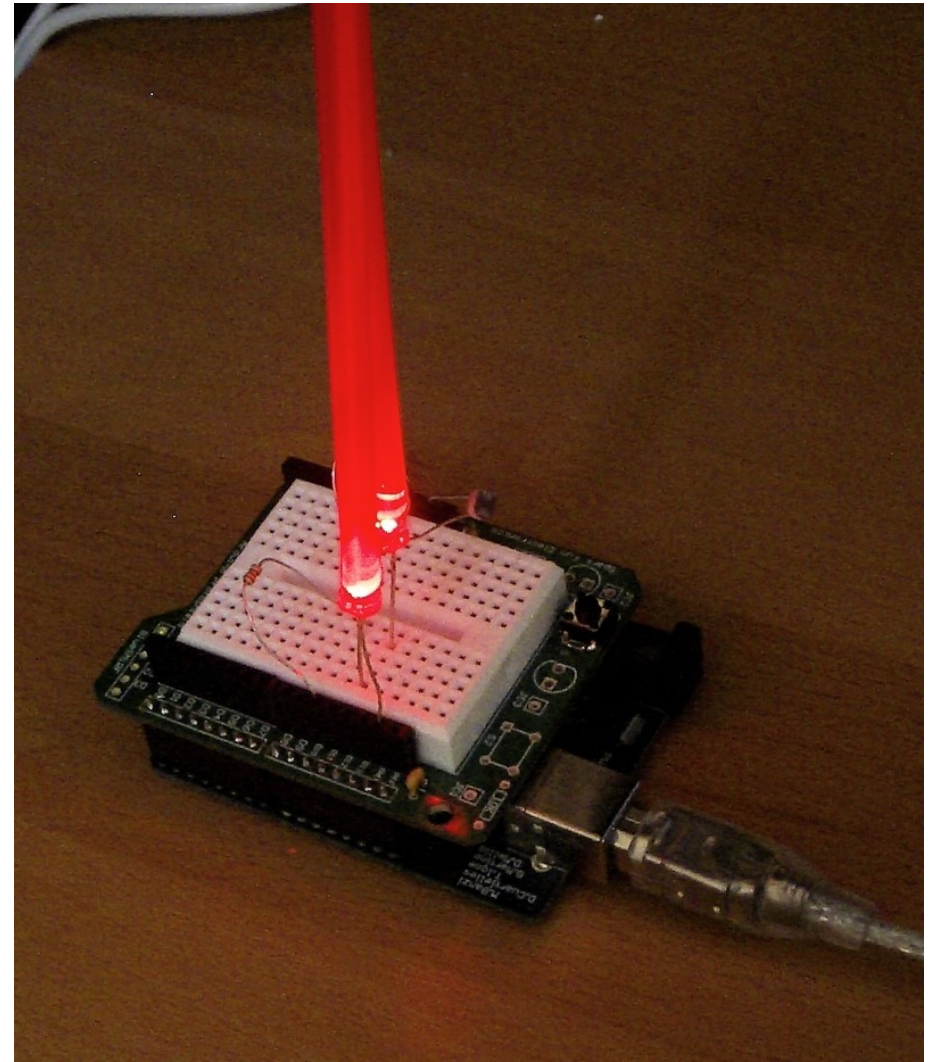
соедините пин 9 со светодиодом

Вопросы / Повторение

Есть ли вопросы, комментарии или проблемы?

Шаг в сторону: трубки со светодиодами

Наденьте соломинки на Ваши светодиоды, чтобы их свечение было более заметно



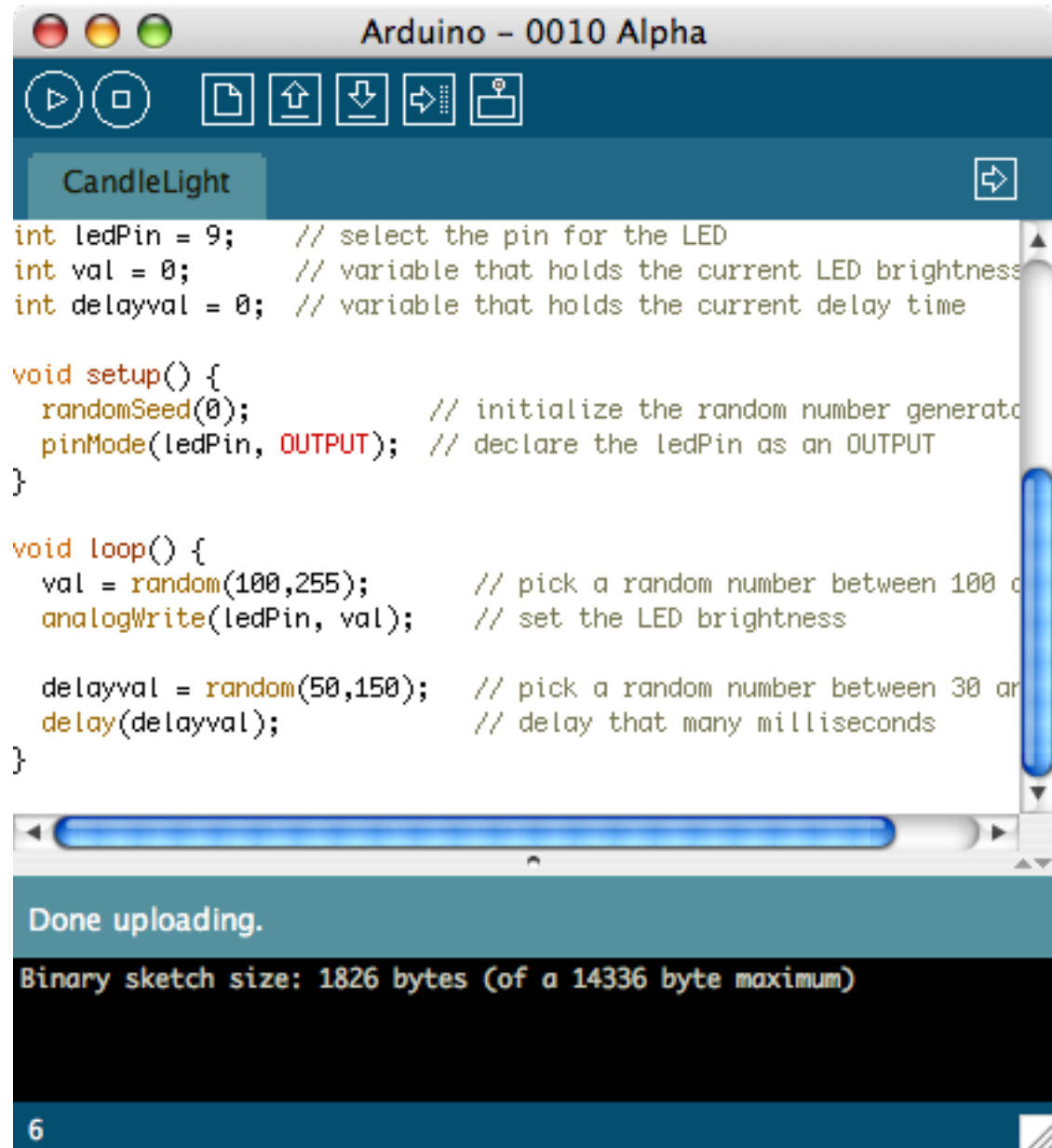
У меня есть целая коробка разноцветных соломинок, для светодиодов любого цвета, какой захотите.

Случайное поведение

«CandleLight»

использует простой генератор псевдо-случайных чисел, чтобы изображать пламя

Используйте `random(min, max)` чтобы выбрать число в промежутке от `min` до `max`



```
Arduino - 0010 Alpha

CandleLight

int ledPin = 9; // select the pin for the LED
int val = 0; // variable that holds the current LED brightness
int delayval = 0; // variable that holds the current delay time

void setup() {
  randomSeed(0); // initialize the random number generator
  pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}

void loop() {
  val = random(100,255); // pick a random number between 100 and 255
  analogWrite(ledPin, val); // set the LED brightness

  delayval = random(50,150); // pick a random number between 30 and 150
  delay(delayval); // delay that many milliseconds
}

Done uploading.
Binary sketch size: 1826 bytes (of a 14336 byte maximum)

6
```

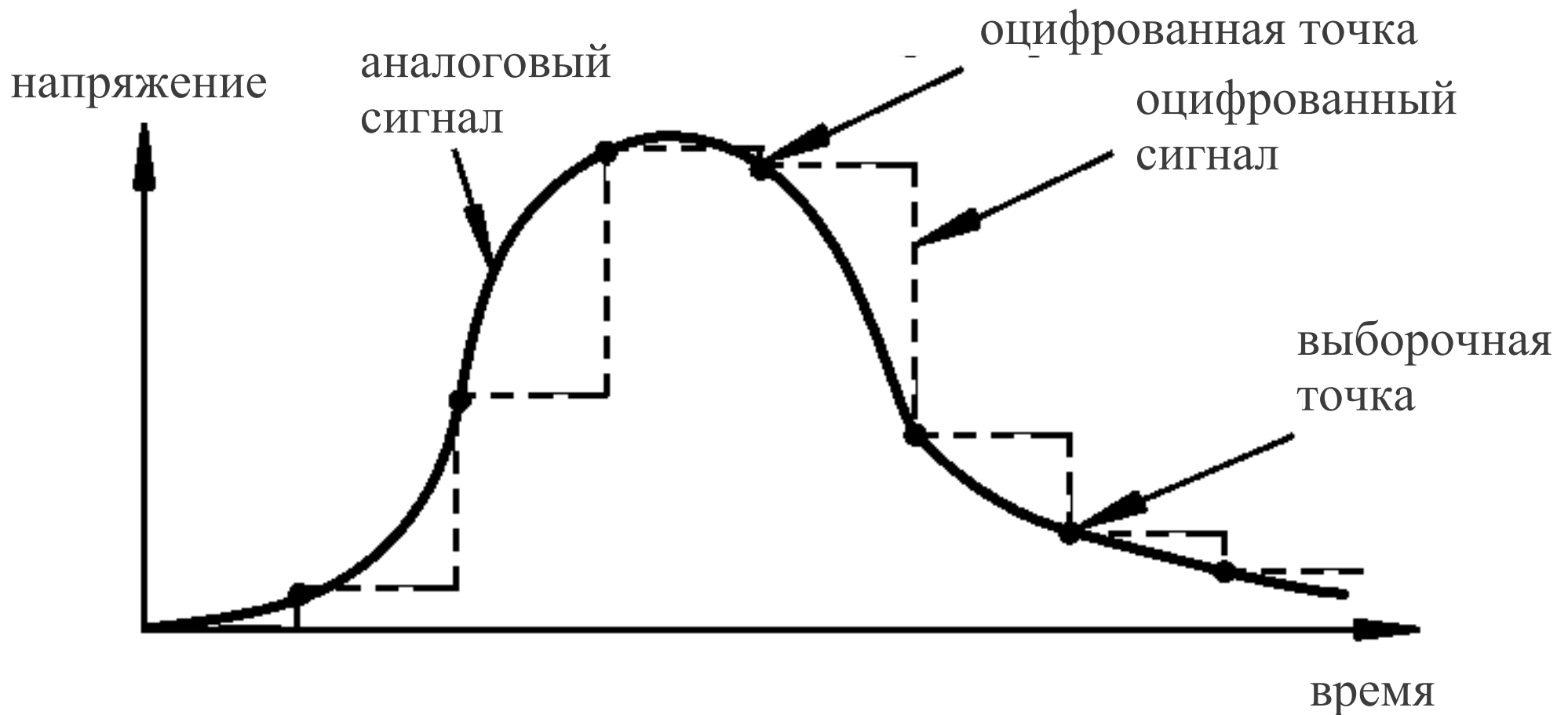
Этот скетч есть в раздаточном материале.

Ещё можно использовать случайные числа, чтобы делать случайный выбор

Имейте в виду: они не вполне «случайные», но для большинства задач ничего другого и не нужно

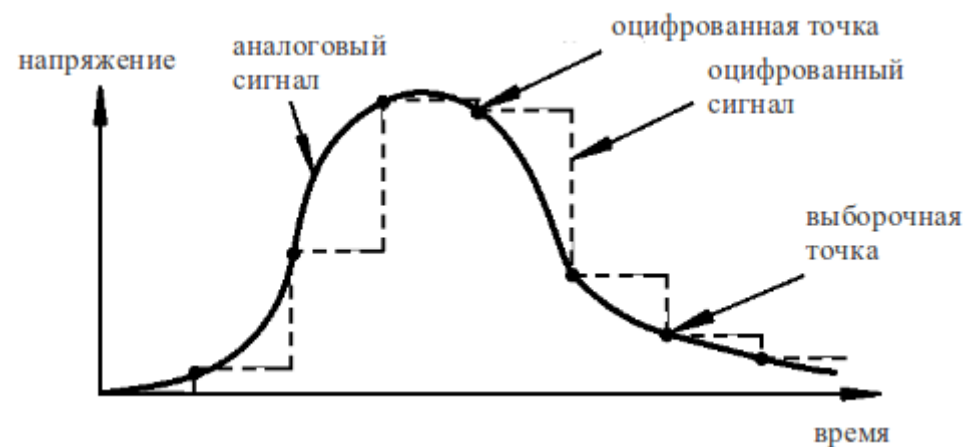
Аналоговый вход

Для компьютеров аналоговый сигнал воспринимается по кусочкам



Аналоговый вход

- Много состояний, не только два (HIGH/LOW)
- Число состояний (или значений) - *разрешение*
- Типичные для компьютеров разрешения:
 - 8 бит = 256 значений
 - 16 бит = 65,536 значений
 - 32 бита = 4,294,967,296 значений



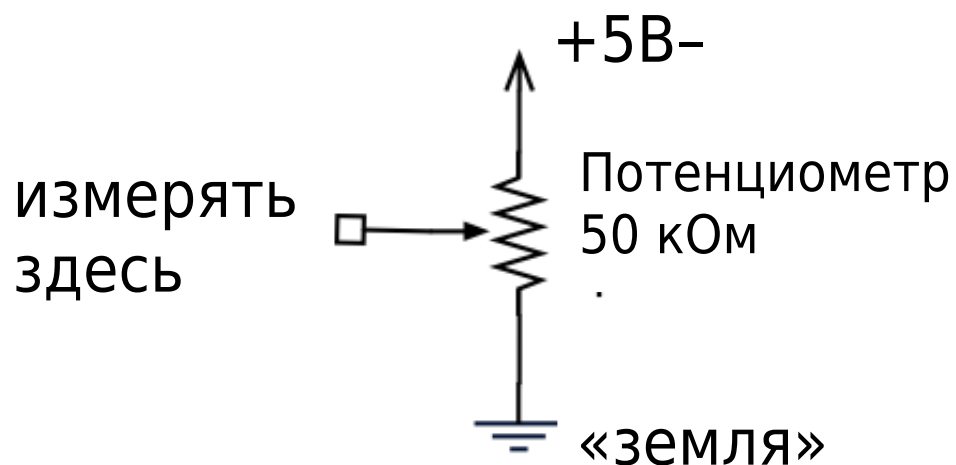
Аналоговый вход

- Arduino (ATmega168) имеет 6 входов АЦП
- АЦП - Аналого-цифровой преобразователь
- Считывает напряжение между 0 и 5 вольт
- Разрешение 10 бит (1024 значений)
- Другими словами, $5/1024 = 4.8$ мВ - наименьшая разница в напряжении, которую Вы можете измерить

Аналоговый вход

Да-да, но как сделать меняющееся напряжение?

С *потенциометром* или, по-другому, *реостатом*



+5V-
измерять
«земля»-



Потенциометр,
который у Вас

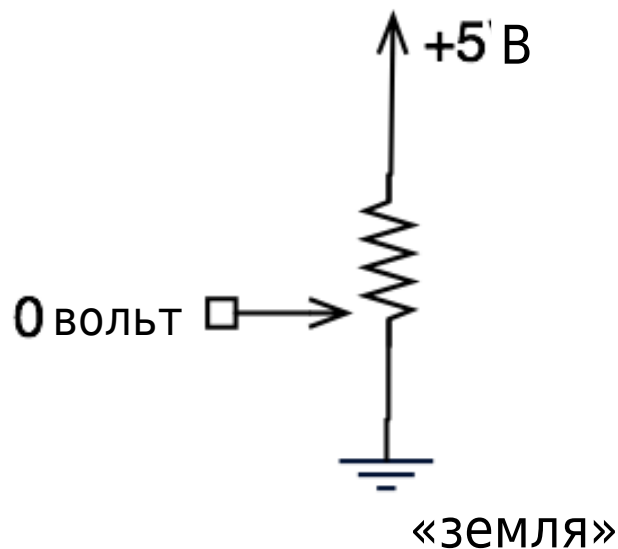


потенциометры ещё выглядят так

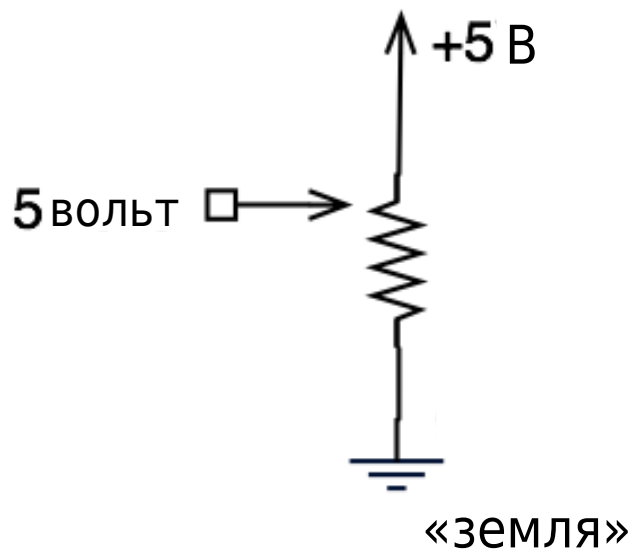
Потенциометры

Вращать ручку — всё равно, что перемещать место соприкосновения стрелки и напряжения на резисторе

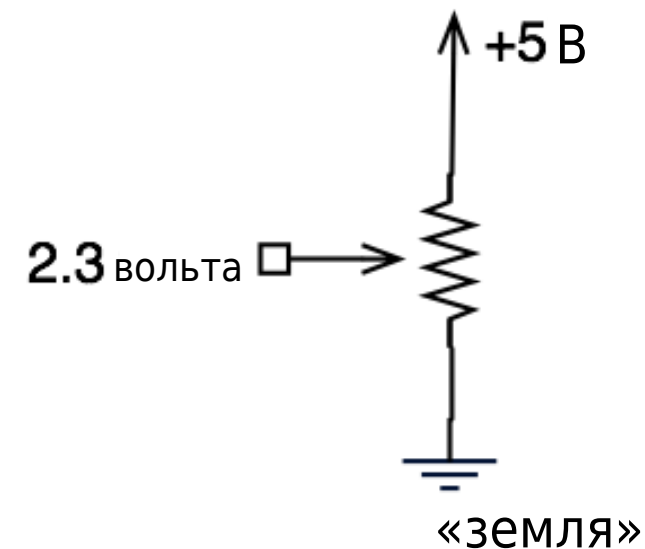
Поворот против часовой стрелки



Поворот по часовой стрелке



Где-то посередине



Если резистор подключён к разнице в напряжении, например от +5В до «земли», то напряжение, измеренное в любой точке по всей длине резистора, пропорционально расстоянию до одной из сторон.

Если разобрать потенциометр, внутри маленькая контактная щётка, прямо как на схемотехническом символе. Но, возможно, направления будут другие (по часовой и против часовой).

Для чего хороши потенциометры?

- Когда нужен вход в заданном диапазоне
 - (для нас привычны ручки настройки)
- Измерять вращательное положение
 - рулевое колесо, шарнир робота, и пр.
- Но для нас важнее, что потенциометры - хороший пример *резистивного датчика*

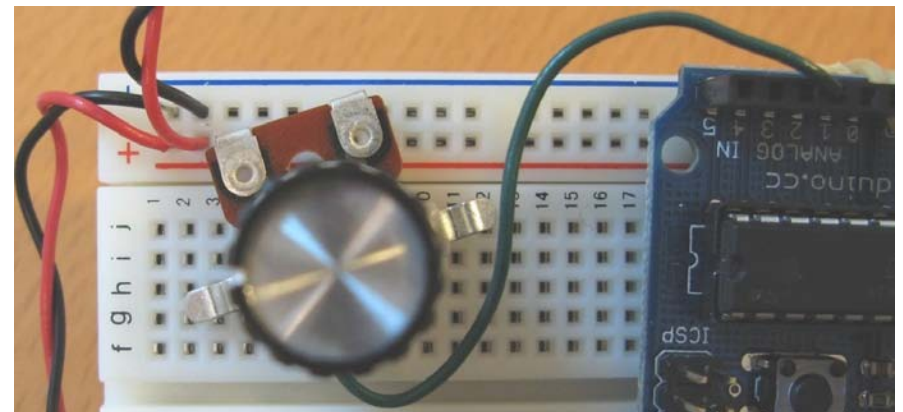
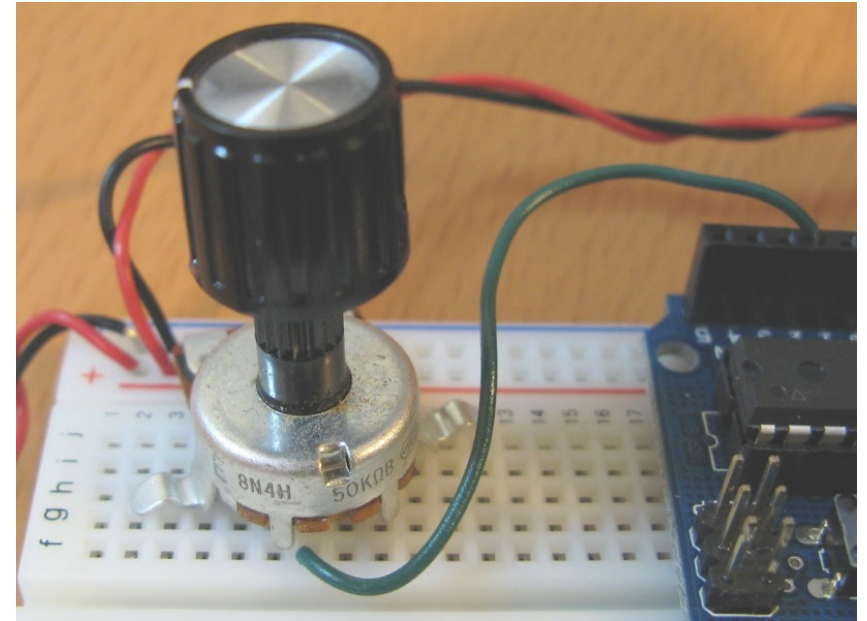
Аналоговый вход на Arduino

Вставьте потенциометр прямо в макетную плату

Две «ножки» подключите к шинам +5В и «земле»
(красный + и синий -)

Средний вывод подключается к ряду гнёзд
(здесь — 7-й ряд)

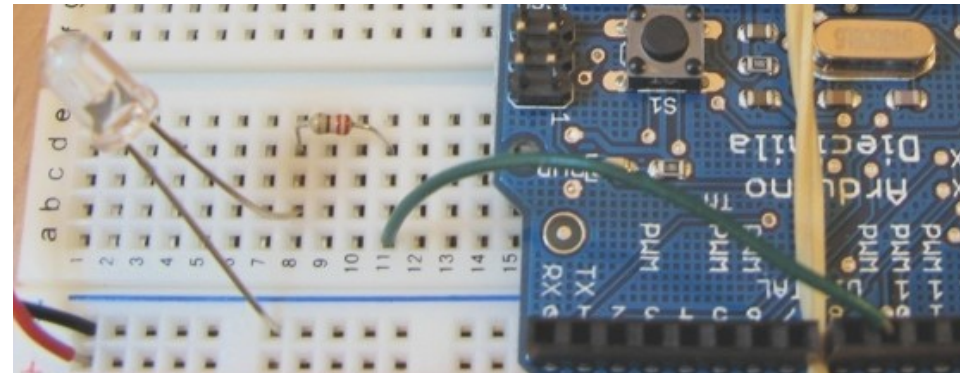
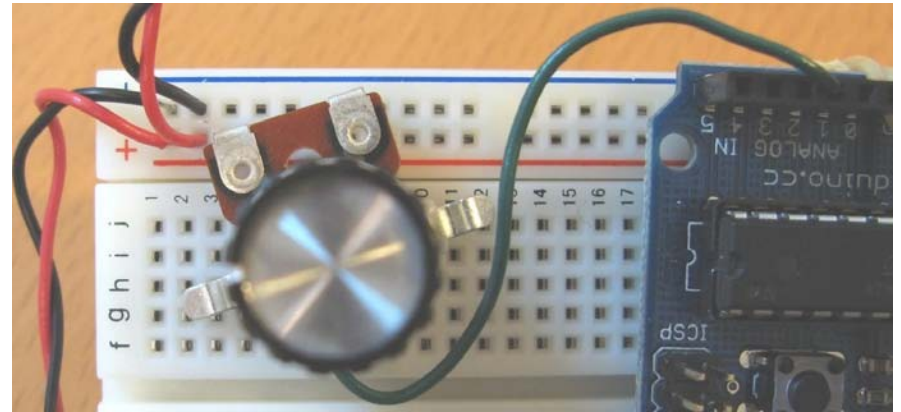
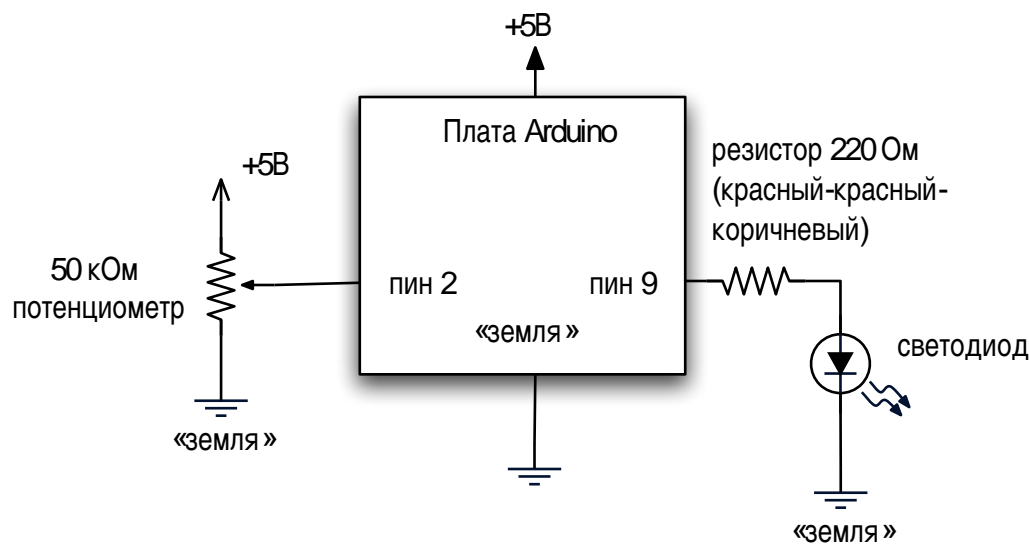
Пропустите провод от этого ряда на аналоговый вход 2



Почему мы используем аналоговый вход 2? Потому что он посередине.
Нет особой причины, все 6 аналоговых входов работают одинаково.

Схема со светодиодом и потенциометром

Вот что теперь должно быть на вашей плате



В схемах входы обычно слева, выходы справа
Ещё положительные полюсы напряжения обычно наверху,
отрицательные обычно снизу.

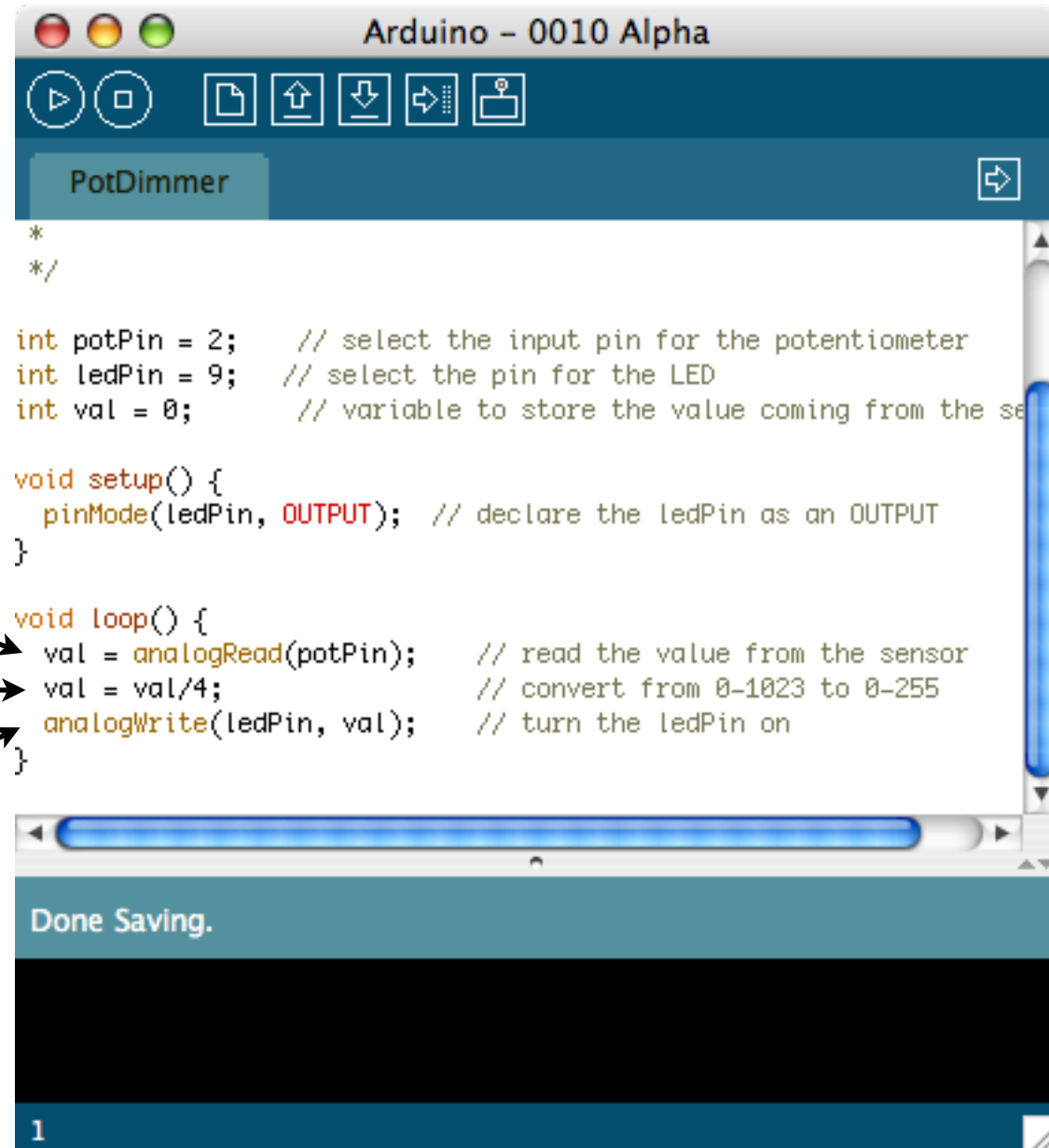
Меняем яркость вручную

«PotDimmer»

Поверните ручку,
чтобы изменить
яркость светодиода

ВХОД
→
обработка
→
ВХОДНЫХ ДАННЫХ
→
ВЫХОД

Практически все
встраиваемые системы
имеют цикл
ВХОД → обработка → ВЫХОД



```
Arduino - 0010 Alpha

PotDimmer

*
*/

int potPin = 2; // select the input pin for the potentiometer
int ledPin = 9; // select the pin for the LED
int val = 0; // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}

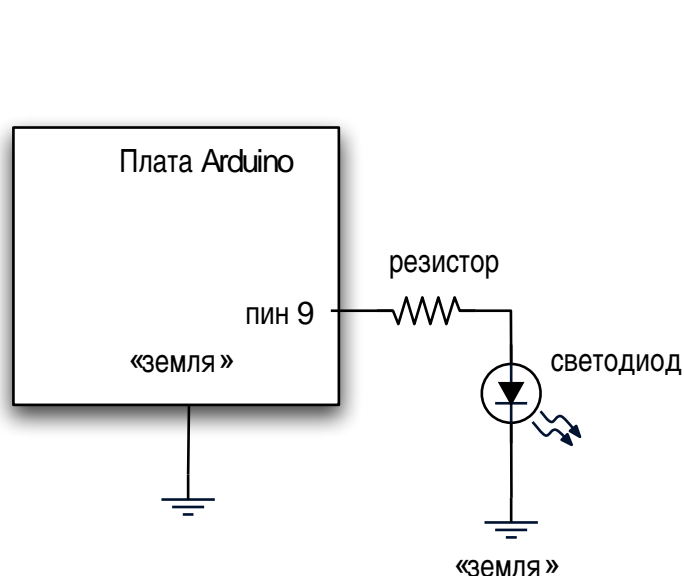
void loop() {
  val = analogRead(potPin); // read the value from the sensor
  val = val/4; // convert from 0-1023 to 0-255
  analogWrite(ledPin, val); // turn the ledPin on
}
```

Done Saving.

1

Этот скетч есть в раздаточном материале

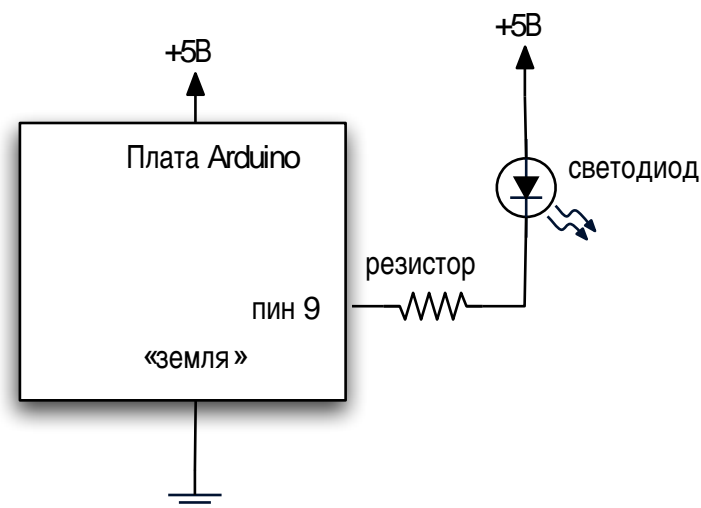
Два способа ПОДКЛЮЧАТЬ СВЕТОДИОДЫ



Чтобы ВКЛЮЧИТЬ: `digitalWrite(9,HIGH)`

Чтобы ВЫКЛЮЧИТЬ: `digitalWrite(9,LOW)`

Задать яркость: `analogWrite(9,val)`



Чтобы ВКЛЮЧИТЬ: `digitalWrite(9,LOW)`

Чтобы ВЫКЛЮЧИТЬ: `digitalWrite(9,HIGH)`

Задать яркость: `analogWrite(9,255-val)`

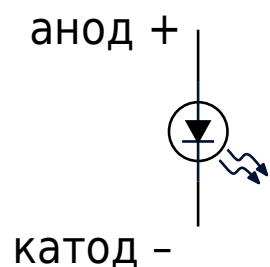
Мы использовали ту, что слева, потому что она более понятна.

Но Вы увидите и такие подключения, как на схеме справа.

Причина в том, что некоторые схемы лучше переключаются на «землю», чем на +5В

RGB-светодиоды

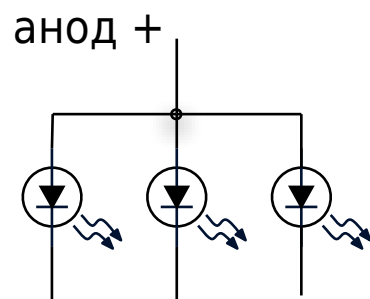
Обычный светодиод



анод +
катод -

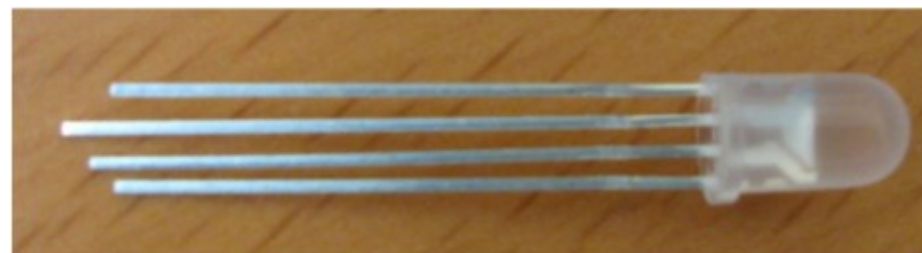


RGB-светодиод



красный синий зелёный

красный катод -
анод +
синий катод -
зелёный катод -



в действительности - 3 светодиода в одном корпусе

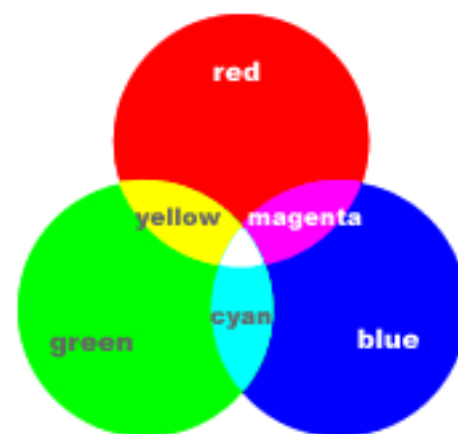
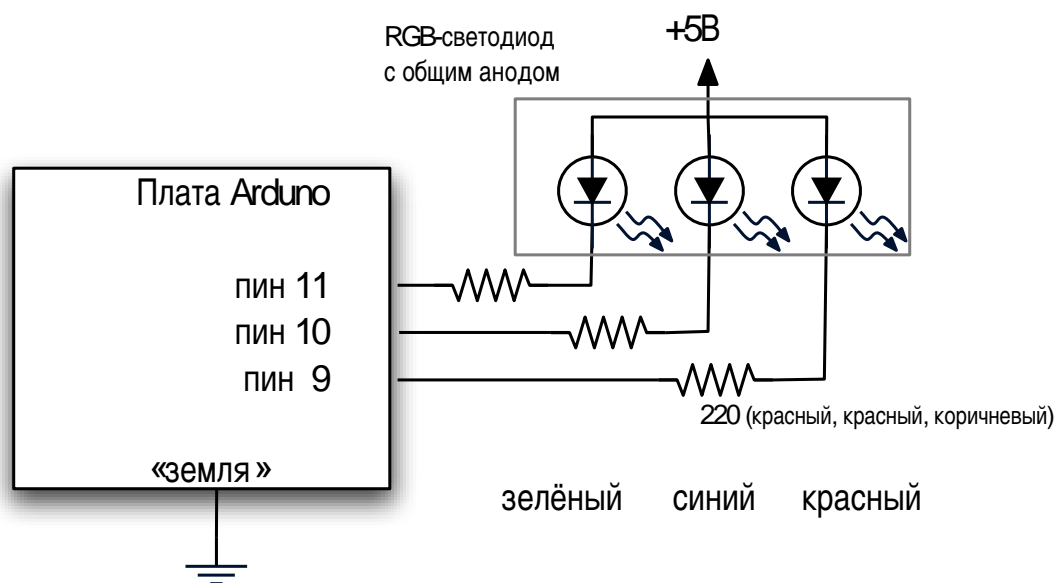
RGB-светодиод, он же «трёхцветный светодиод»

RGB-светодиоды с общим анодом гораздо больше распространены, чем с общим катодом.

Вот почему мы немного меняем управляющую логику.

Смешивание цветов

Всего 3 светодиодами можно сделать любой* цвет

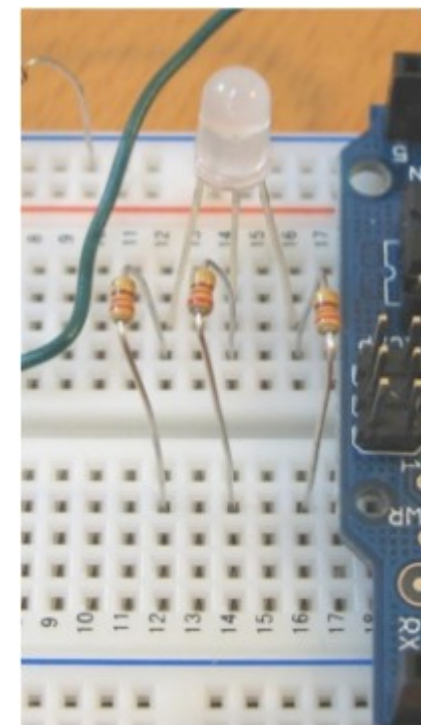
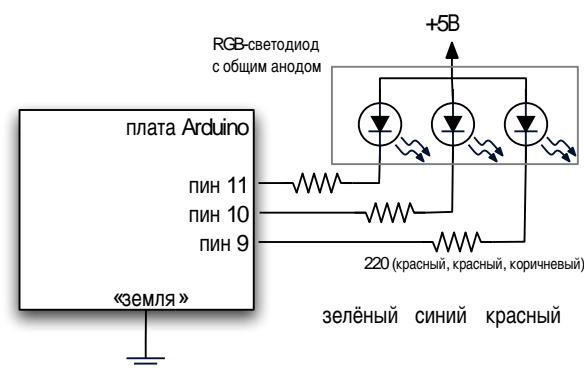
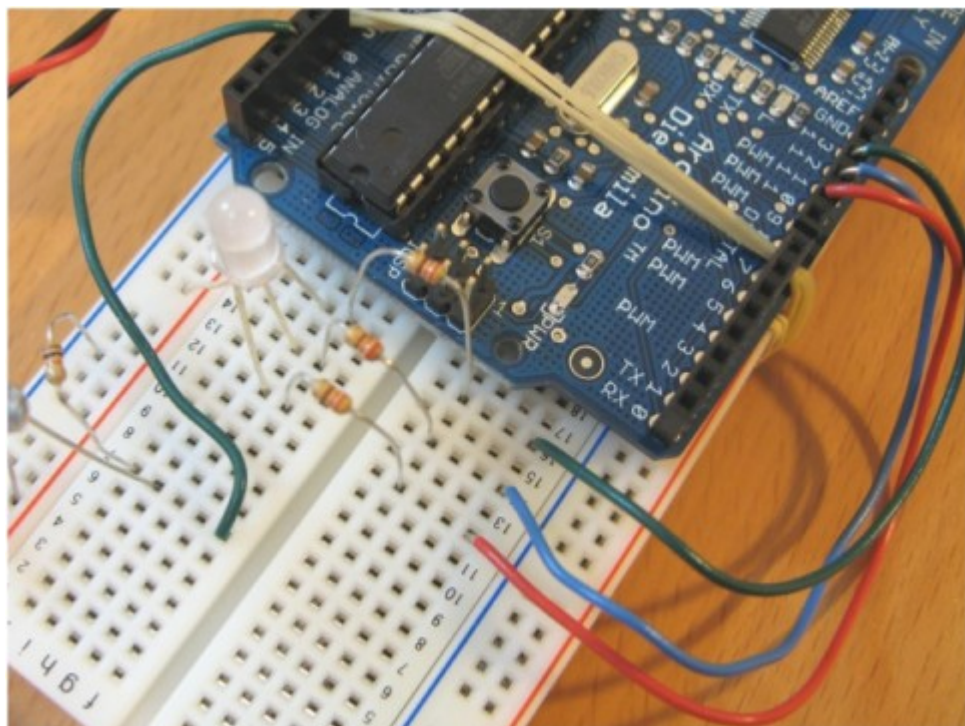


С RGB можно
сделать любой цвет
(кроме чёрного)

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

- *если оставить в стороне разницу между аддитивной/субтрактивной моделью, существует много неприятных неустранимых причин, по которым сложно получить нужное смешение:
- физика светодиодов такова, что светодиоды разных цветов дают разное количество света
 - наши глаза выдают нелинейно зависимый результат относительно спектра. Например, мы более чувствительны к зелёному, нежели к красному
 - в большинстве RGB-светодиодов линза не фокусирует все цветовые пятна в одну и ту же точку

Сборка схемы с RGB-светодиодом



- немного изогните самый длинный вывод и вставьте его в +5В шину (красную)
- вставьте остальные выводы в ряды (здесь 12,14,и 16)
- подключите резисторы 220 ом (красный-красный-коричневый) через середину к соответствующим рядам
- пропустите провода от резисторов к пинам 9,10,11 на плате Arduino, можно взять их по цветам светодиода

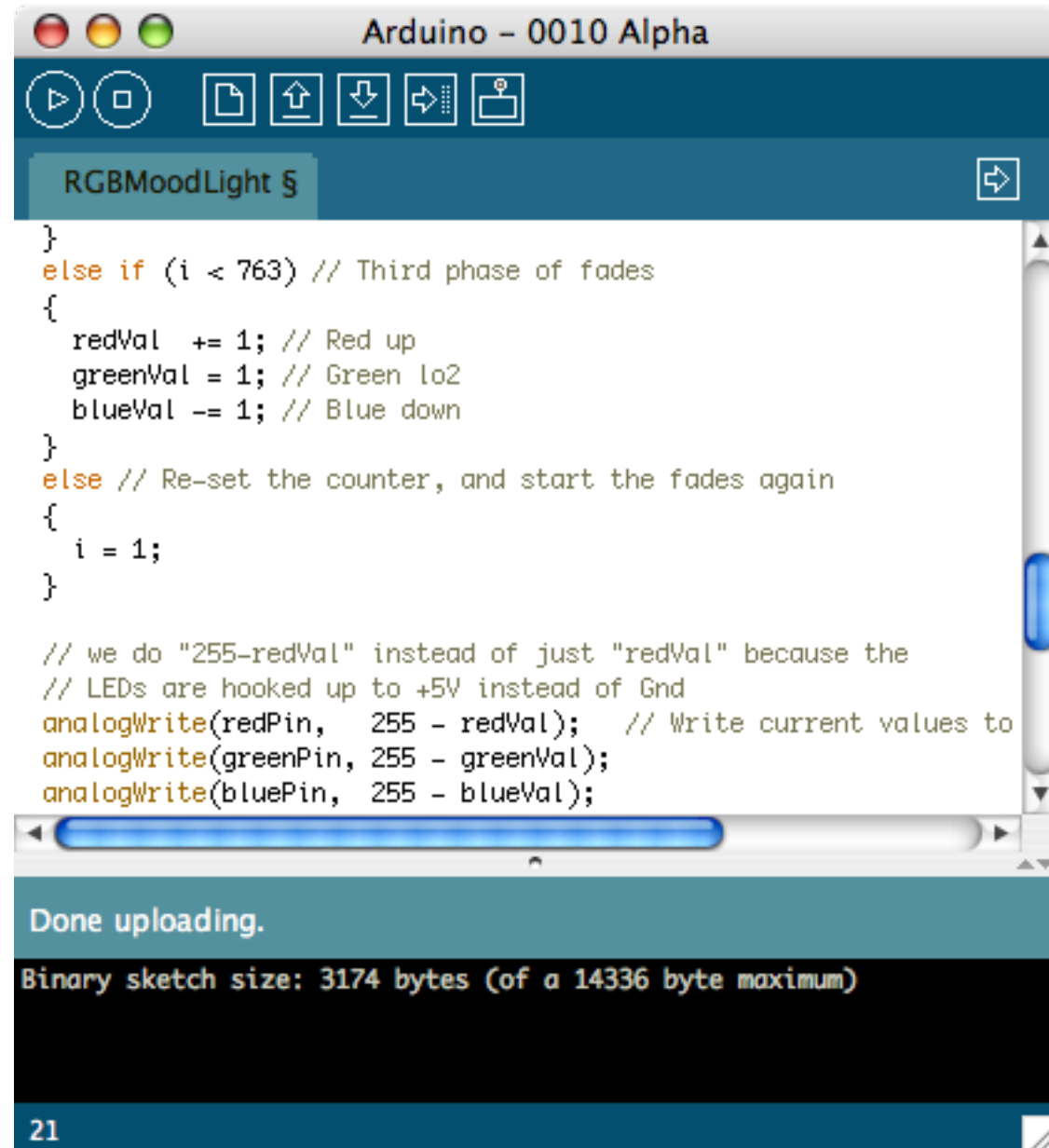
Не обращайте внимания на зелёный провод на картинках, это другая схема. Можно оставить потенциометр из прошлой схемы, если сможете.

Гашение RGB-цветов

“RGBMoodLight”

Медленное гашение
и смешивание цветов

Также печатает текущие значения
цветов в последовательный порт



```
Arduino - 0010 Alpha
RGBMoodLight 5
}
else if (i < 763) // Third phase of fades
{
  redVal += 1; // Red up
  greenVal = 1; // Green lo2
  blueVal -= 1; // Blue down
}
else // Re-set the counter, and start the fades again
{
  i = 1;
}

// we do "255-redVal" instead of just "redVal" because the
// LEDs are hooked up to +5V instead of Gnd
analogWrite(redPin, 255 - redVal); // Write current values to
analogWrite(greenPin, 255 - greenVal);
analogWrite(bluePin, 255 - blueVal);

Done uploading.
Binary sketch size: 3174 bytes (of a 14336 byte maximum)
21
```

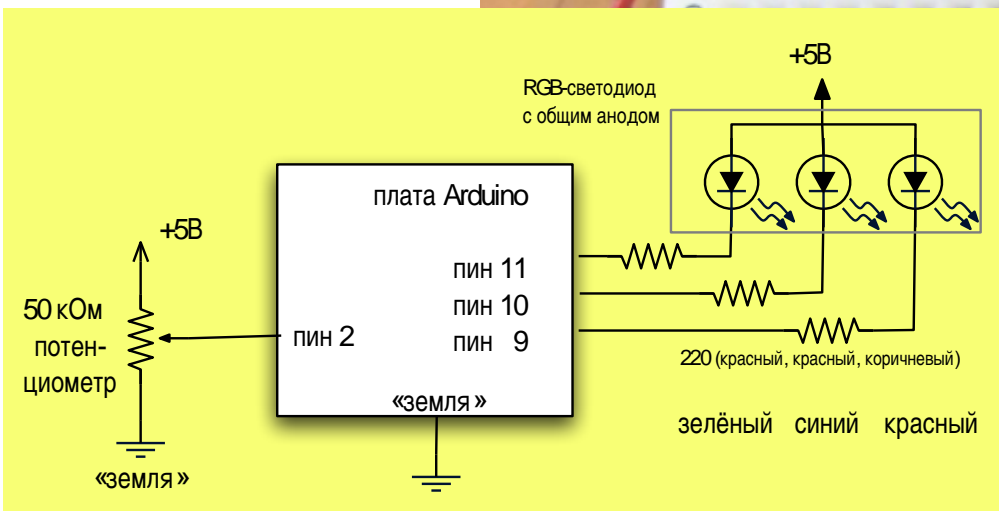
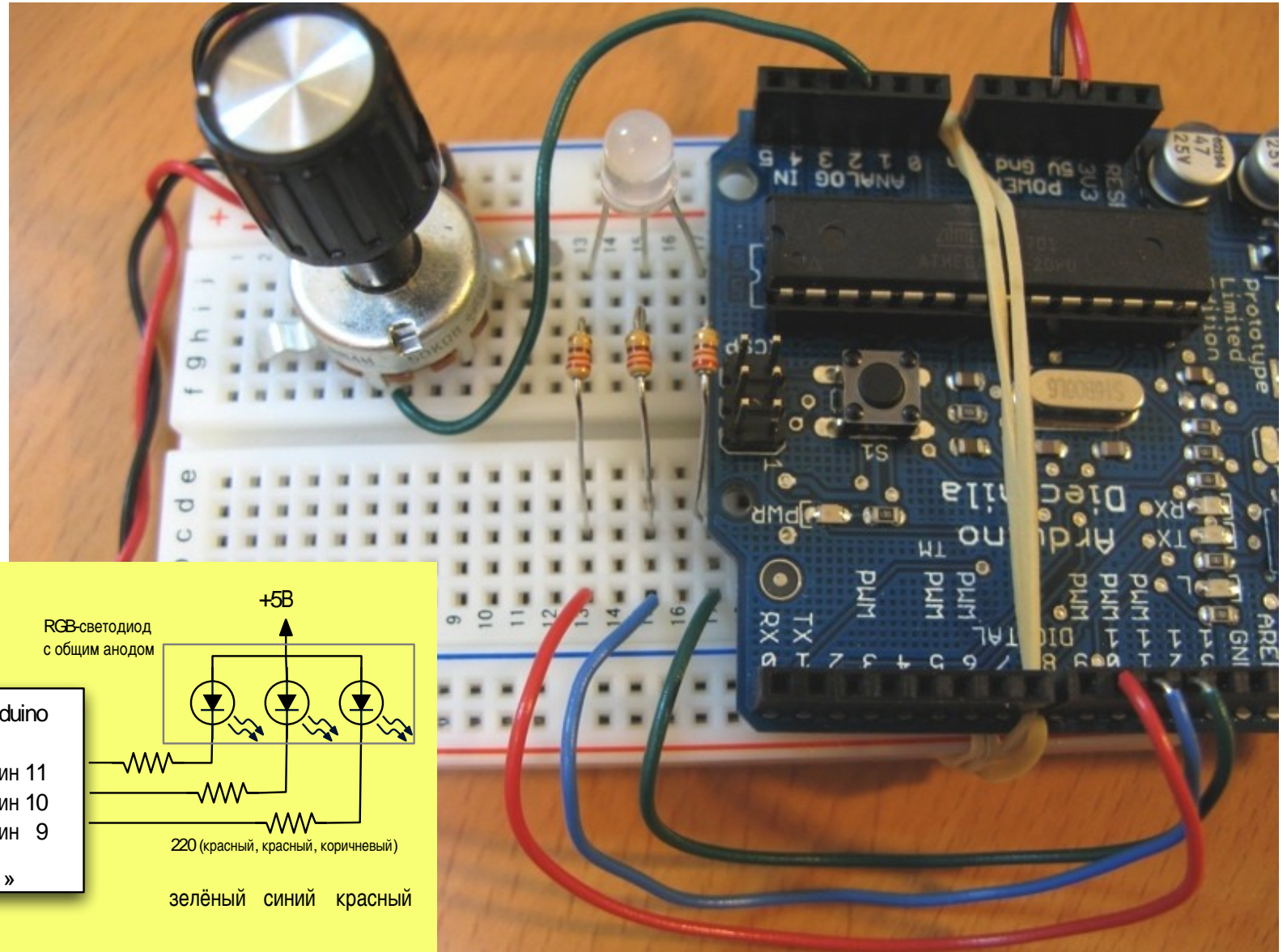
Этот скетч есть в раздаточном материале.

Через минуту мы доберёмся и до работы с последовательным портом.

В скетче просто то наращиваются, то уменьшаются значения красного, зелёного и синего цветов, и передаются в analogWrite()

ИСТОЧНИК: <http://www.arduino.cc/en/Tutorial/DimmingLEDs>

RGB-светодиод, управляемый потенциометром

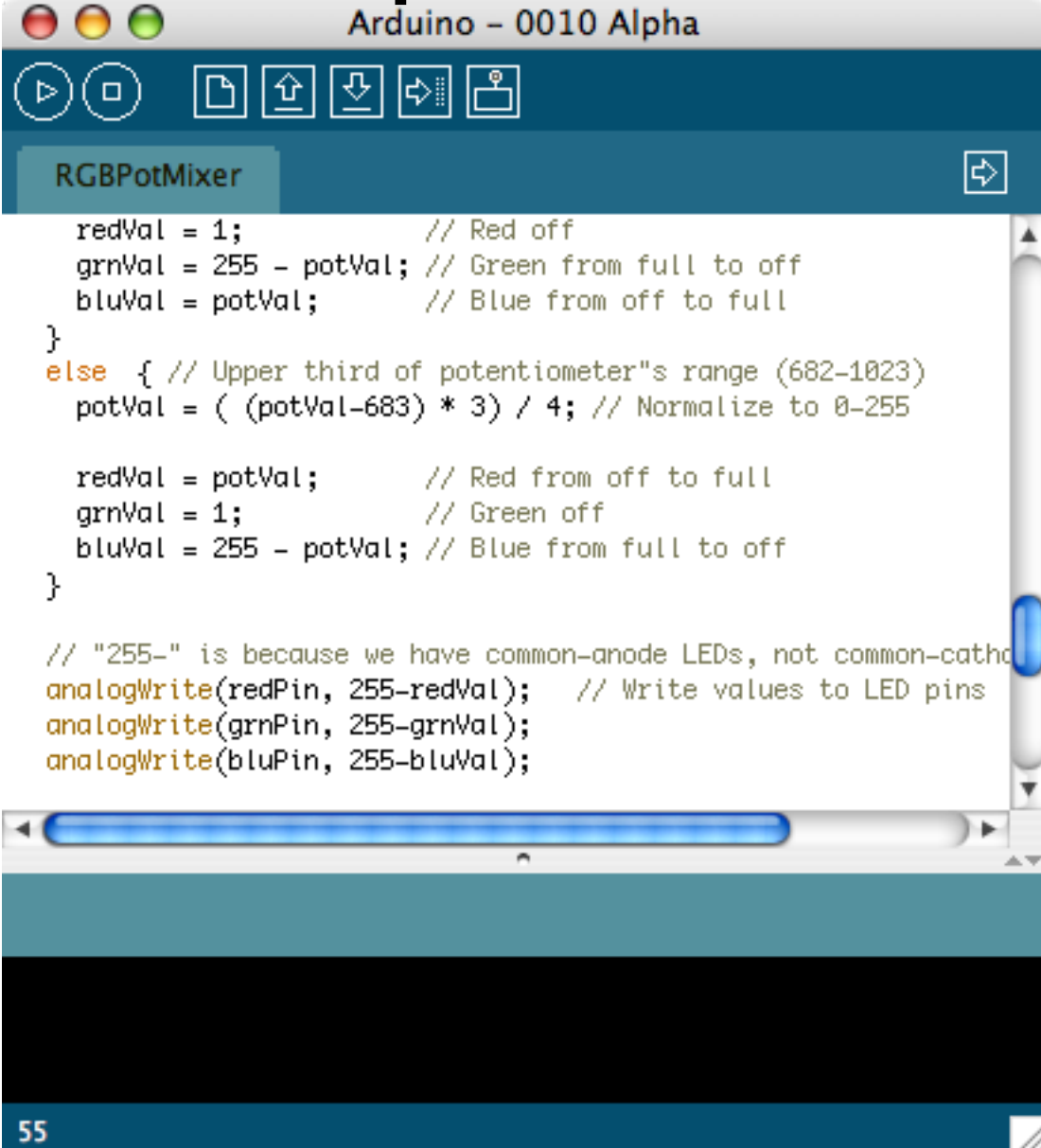


RGB-светодиод, управляемый потенциометром

“RGBPotMixer”

Используйте
потенциометр
из прошлых задач,
чтобы управлять
смешиванием цветов

Код превращает одномерную
входную величину в “секторы”, где
каждый сектор — отдельный цвет



```
Arduino - 0010 Alpha
RGBPotMixer
redVal = 1; // Red off
grnVal = 255 - potVal; // Green from full to off
bluVal = potVal; // Blue from off to full
}
else { // Upper third of potentiometer's range (682-1023)
  potVal = ( (potVal-683) * 3) / 4; // Normalize to 0-255

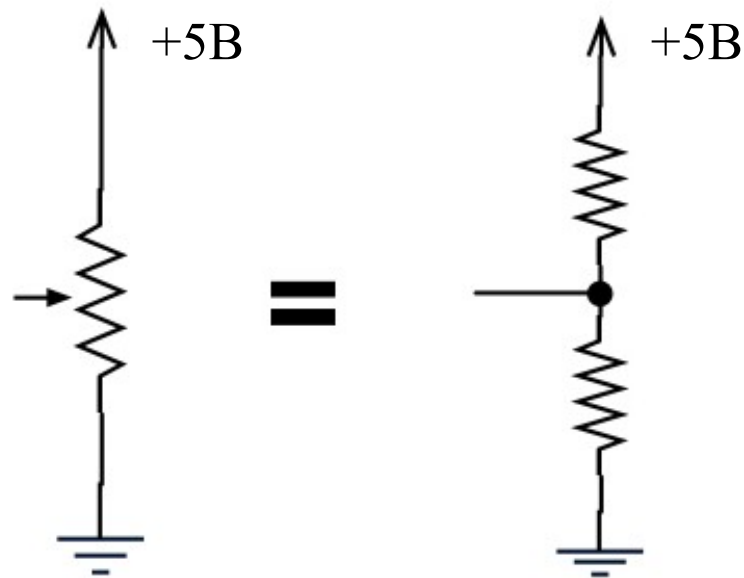
  redVal = potVal; // Red from off to full
  grnVal = 1; // Green off
  bluVal = 255 - potVal; // Blue from full to off
}

// "255-" is because we have common-anode LEDs, not common-cathode
analogWrite(redPin, 255-redVal); // Write values to LED pins
analogWrite(grnPin, 255-grnVal);
analogWrite(bluPin, 255-bluVal);
55
```

Ещё посмотрите “RGBPotMixer2” с другим вариантом.
Как изменить, чтобы настроить яркость?

Определяем темноту

- Потенциометр — пример *делителя напряжения*
- Делитель расщепляет напряжение на части
- Как два резистора, но с изменяемым сопротивлением

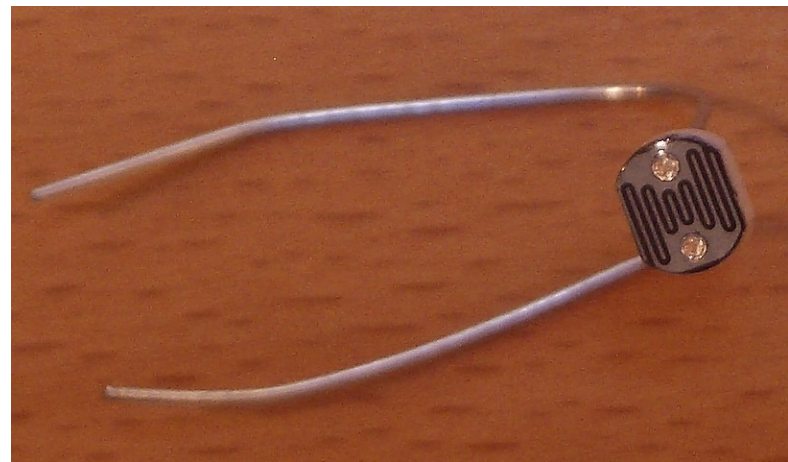


Определяем темноту: фотоэлементы

- Он же фоторезистор, светочувствительный резистор
- *Переменный* резистор
- Свет ярче == сопротивление ниже
- В Вашем диапазоне 0-10кОм - 1МОм

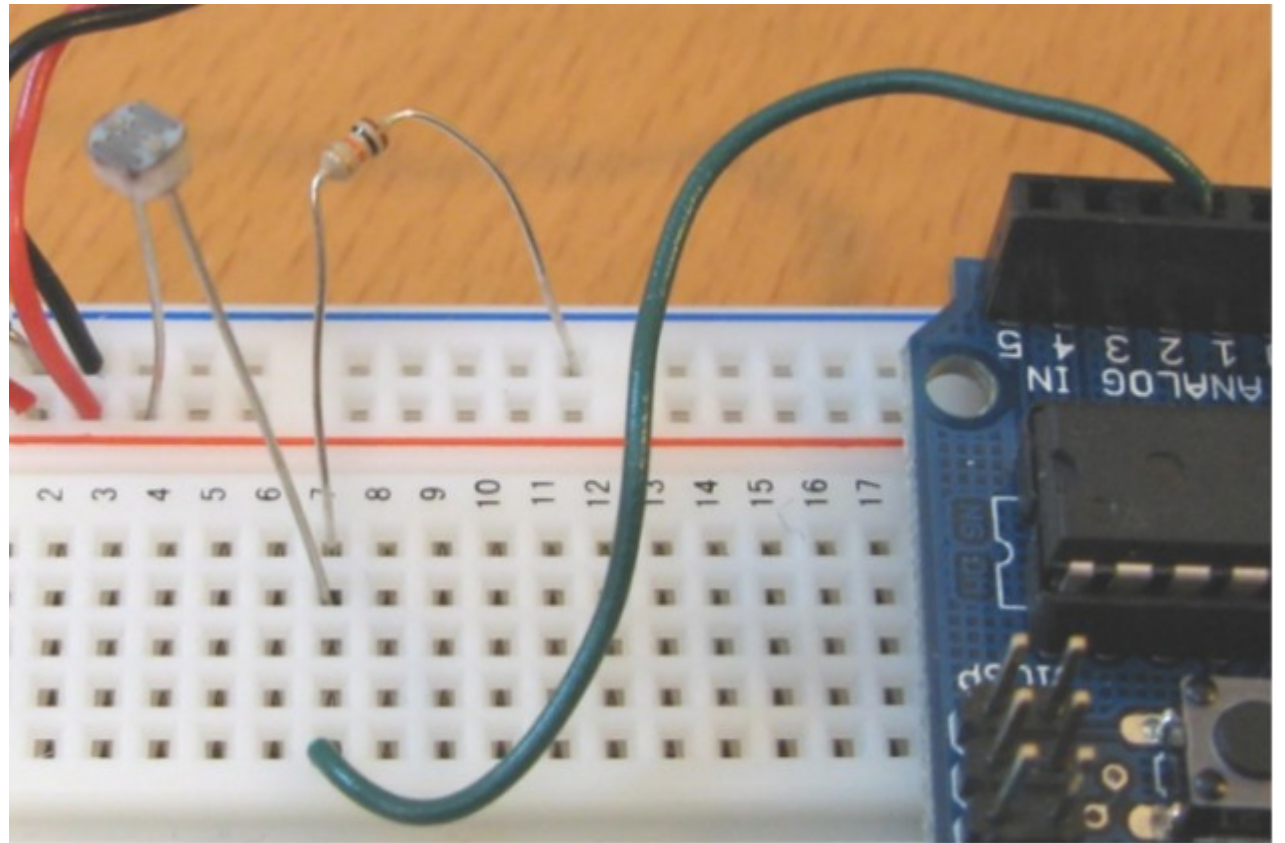
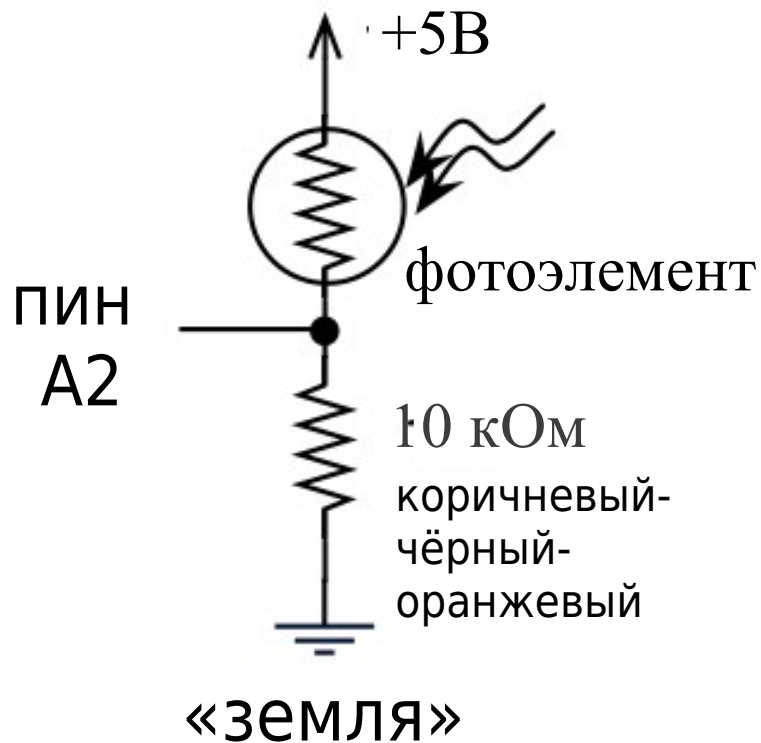


СИМВОЛ НА СХЕМЕ



Довольно дешёвые. Можно купить мешок из 100 штук в Jameco за \$20

Схема с фотоэлементом

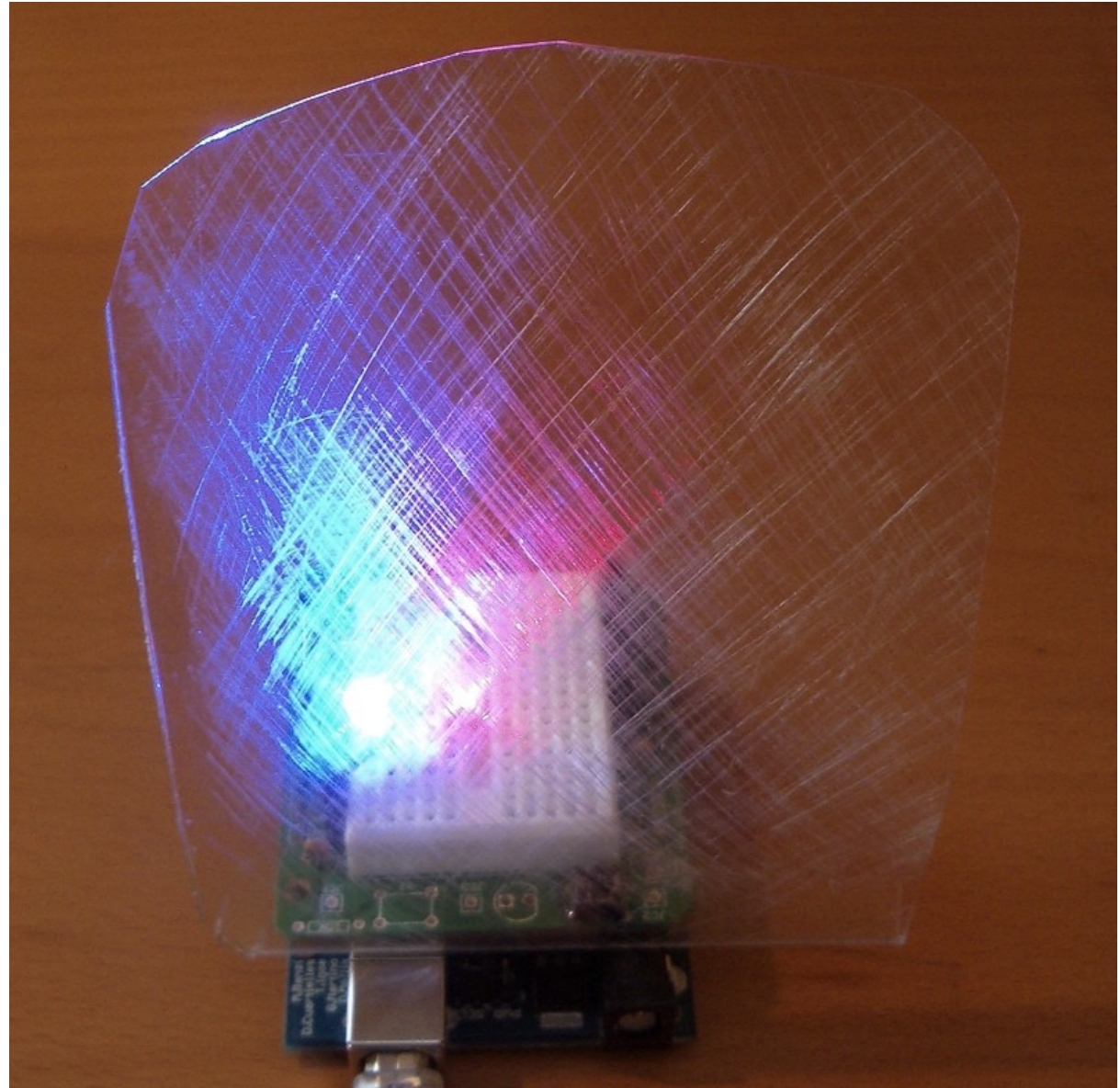


Попробуйте с одной из старых программ RGBPotMixer

Выглядит очень похоже на схему с потенциометром, не правда ли?

Лампа настроения

Рассеиватель
сделан из куска
пластика,
поцарапанного
наждачной
бумагой



Ещё можно использовать хрустящую скомканную обёртку от цветов - получится интересный рассеиватель.

Резистивные датчики



термистор
(температура)

4 см

13 см

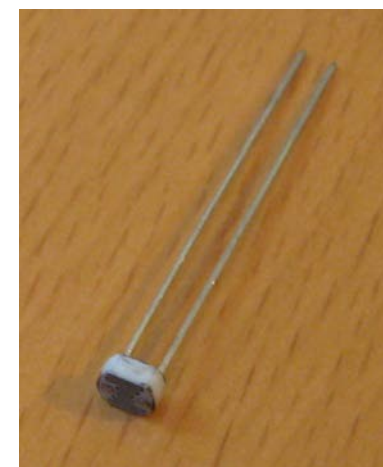
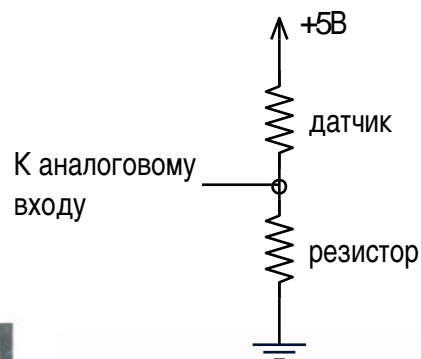
5 см



(давление)



Для всех случаев
схема — одна и та же



фотоэлемент
(свет)



датчик изгиба
(наклон, деформация)

датчики давления
воздуха, и многие
другие

Общение с другими

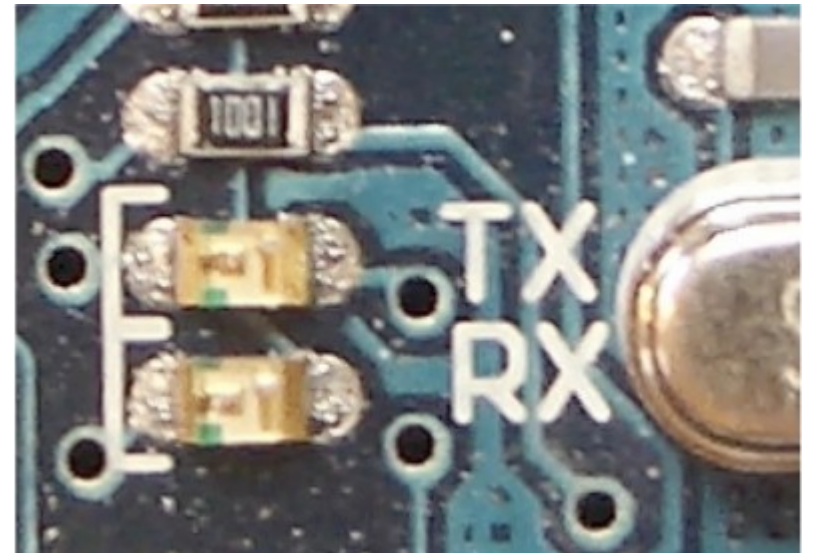
- Arduino может использовать один и тот же USB-кабель для программирования и общения с компьютерами
- Для общения с другими устройствами используются команды `Serial`
 - `Serial.begin()` – подготовка к соединению через последовательный порт
 - `Serial.print()` – послать данные компьютеру
 - `Serial.read()` – получить данные с компьютера

Можно общаться не только с компьютерами

Большинство других, более сложных, чем простые сенсоры/эффекторы, устройств, тоже имеют последовательный интерфейс подключения..

Наблюдайте за светодиодами TX/RX

- TX - передача компьютеру
- RX - приём от компьютера
- Мигают во время загрузки программы или коммуникации



Arduino говорит «Привет»

“SerialHelloWorld”

Посылает “Hello world!”
Вашему компьютеру

Нажмите на кнопку
“Serial Monitor”, чтобы
увидеть результат

Посмотрите на светодиод
TX и сравните со светодиодом
на 13 пине

```
pinMode(ledPin, OUTPUT); // declare the LED's pin as output
Serial.begin(19200); // connect to the serial port

void loop () {
  // Serial.available() is a way to see if there's serial data
  // without pausing your code
  if( Serial.available() ) {
    val = Serial.read(); // read the serial port
    if( val == 'H' ) { // if it's an 'H', blink the light
      digitalWrite(ledPin, HIGH);
      delay(1000);
      digitalWrite(ledPin, LOW);
    }
  }
}
```

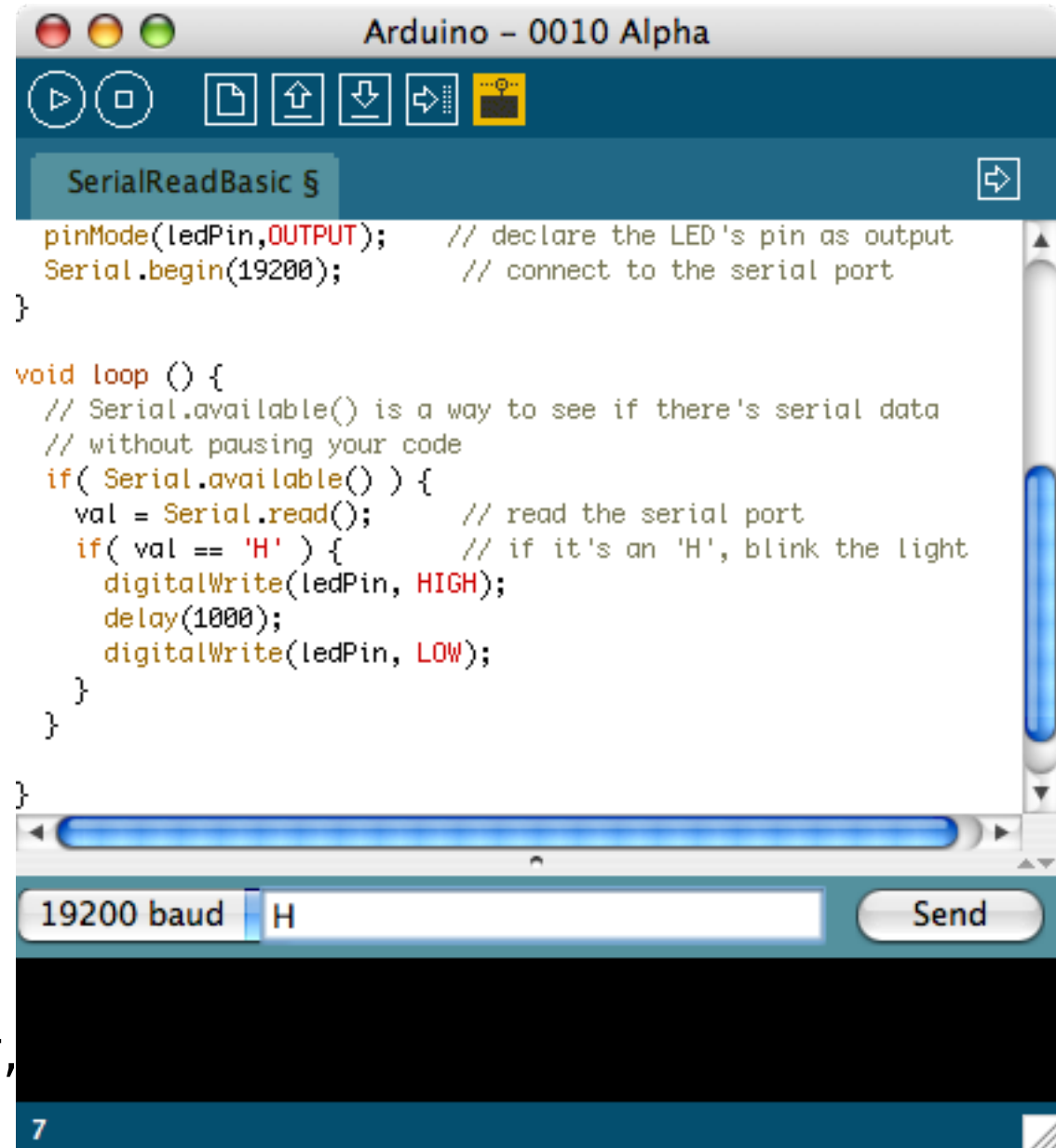
Этот скетч есть в раздаточном материале; впрочем, он очень короткий.
Используется встроенный в плату светодиод на 13 пине, не нужно собирать никакой схемы.

Говорим Arduino, что делать

“SerialReadBasic”

Вы посылаете “H”,
светодиод мигает

В “Serial Monitor”,
напечатайте “H”,
нажмите Send



The screenshot shows the Arduino IDE interface. The main window displays the sketch "SerialReadBasic" with the following code:

```
pinMode(ledPin, OUTPUT); // declare the LED's pin as output
Serial.begin(19200);      // connect to the serial port
}

void loop () {
  // Serial.available() is a way to see if there's serial data
  // without pausing your code
  if( Serial.available() ) {
    val = Serial.read(); // read the serial port
    if( val == 'H' ) {   // if it's an 'H', blink the light
      digitalWrite(ledPin, HIGH);
      delay(1000);
      digitalWrite(ledPin, LOW);
    }
  }
}
```

Below the code editor is the Serial Monitor window. It shows a baud rate of 19200, a text input field containing the character 'H', and a 'Send' button. The output area is currently empty.

`Serial.available()` сообщает,
есть ли данные для чтения

Этот скетч есть в раздаточном материале

Всегда проверяйте `Serial.available()` или `Serial.read() != -1` чтобы проверить, есть ли актуальные данные для считывания.

Можно поменять, чтобы печатать “hello world” после того, как что-то приходит, но до проверки ‘H’. Таким образом можно проверить, что получение успешно.

Коммуникация с Arduino

Это просто коммуникация по последовательному интерфейсу


- По секрету, Arduino не умеет работать с USB
- А только через “последовательный порт”, как старый протокол RS-232
- Все микроконтроллеры могут общаться через последовательный порт
- Немногие - через USB
- Последовательный порт - просто, USB - сложно



Последовательный терминал из стародавних времён

Коммуникация через последовательный порт

- “Последовательный”, потому что данные разбиваются на отдельные биты, которые посылаются один за другим по одному проводу.
- Один ASCII-символ ‘B’ посылается так:

‘B’ = 0 1 0 0 0 0 1 0
= L H L L L L H L
= 

- Пин меняет состояние, как если бы мигал светодиод
- Можно запрограммировать это при помощи функций `digitalWrite()` и `delay()`
- Один провод - для посылки, другой для получения.

Имейте в виду: один сигнальный провод. Вам по-прежнему нужен ещё провод для заземления.

Arduino и USB-to-serial

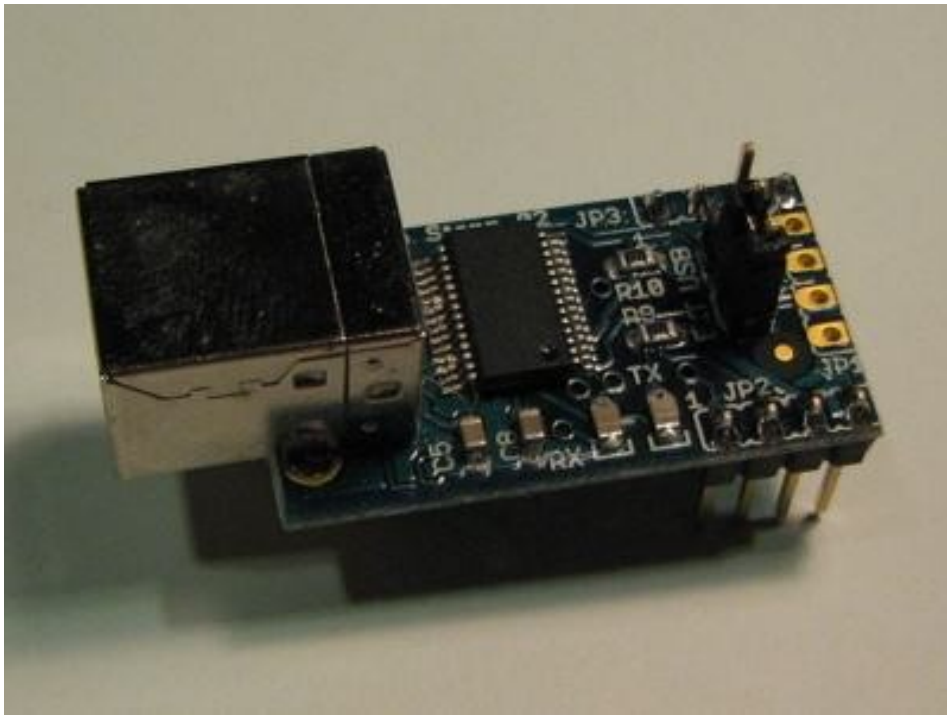
На самом деле, на плате Arduino две схемы



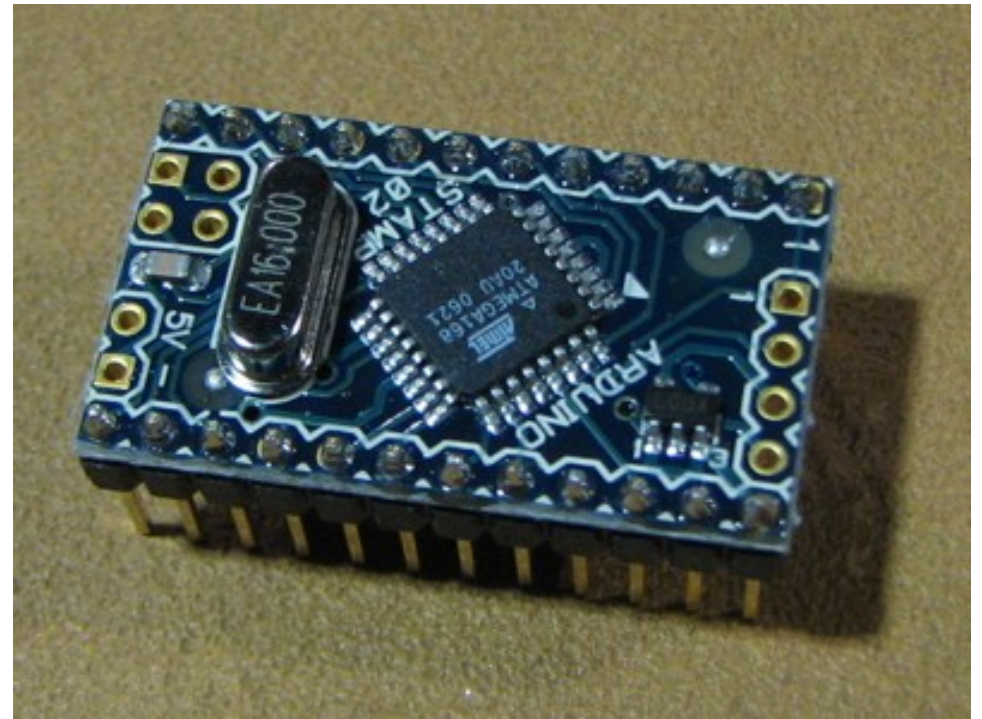
Изначально платы Arduino были с последовательным интерфейсом RS-232, а не USB.

Arduino Mini

Arduino Mini разделяет две схемы



Arduino Mini USB-адаптер

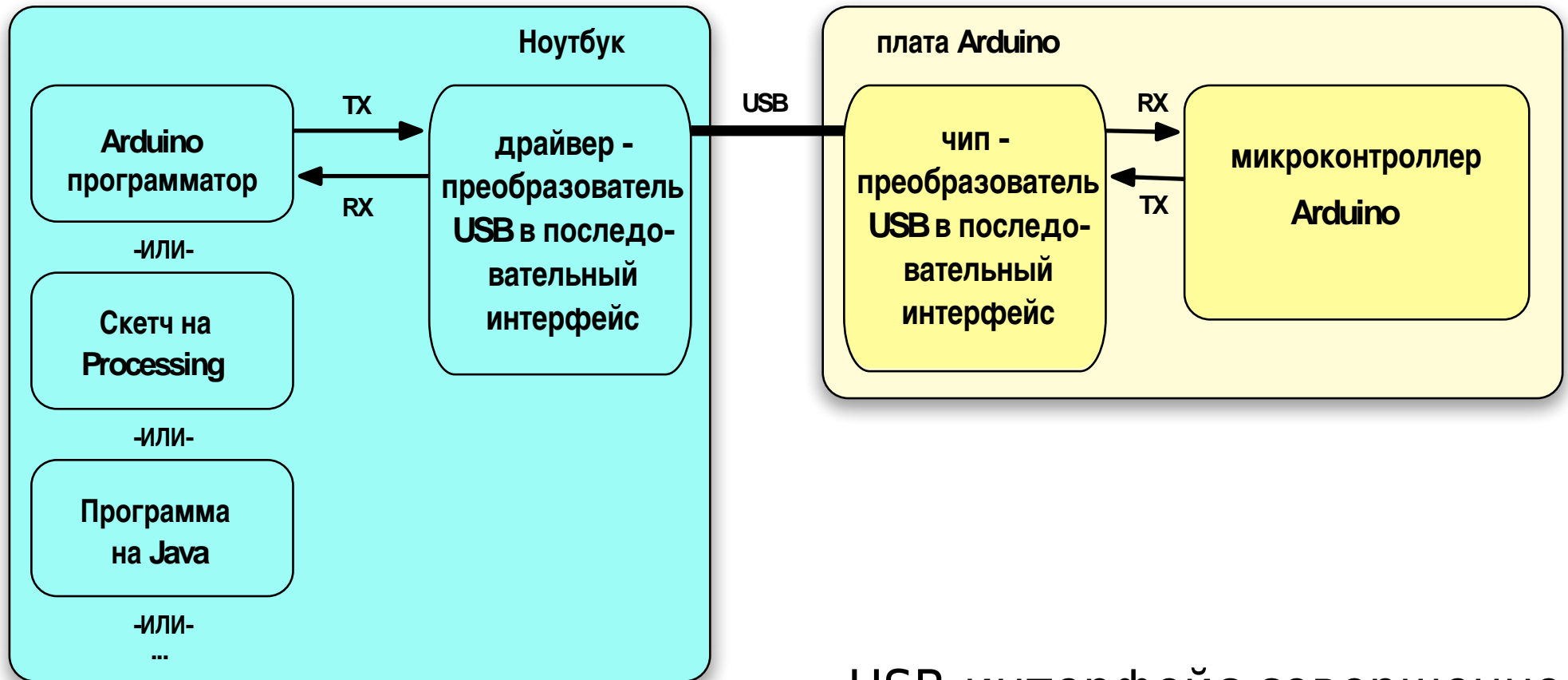


Arduino Mini

Также известны как "Arduino Stamp"

Если обмен данными с компьютером не требуется, то функционал USB-to-serial избыточен.

От Arduino к компьютеру



USB-интерфейс совершенно необязателен в Arduino.
Но он значительно всё упрощает

Изначально платы Arduino были с последовательным интерфейсом RS-232, а не USB.
Все программы, которые общаются с Arduino (даже Arduino IDE) думают, что общаются через последовательный порт.

Arduino и USB

- Так как Arduino работает с последовательным портом,
- А не с USB,
- Подключение к таким устройствам, как USB-флэшки, USB-жёсткие диски, USB-вебкамеры и пр. *не возможно*

Также, в протоколе USB есть разделение на «хост» и «периферию». Чтобы устройство было USB «хостом», необходимо много работы по управлению питанием и логикой, и это трудная задача для крошечного 8кВ микроконтроллера. Он может быть периферийным устройством. Фактически, есть открытый проект “AVR-USB”, который позволяет чипам AVR, наподобие тех, что используются в Arduino, быть полноценными USB-периферийными устройствами. См: <http://www.obdev.at/products/avrusb/>

Управление компьютером

- Можно посылать данные от Arduino к компьютеру командой `Serial.print()`
- Есть много разных вариантов для любых Ваших нужд:

```
int val = 123;
Serial.print(val);      // sends 3 ASCII chars "123"
Serial.print(val,DEC); // same as above
Serial.print(val,HEX); // sends 2 ASCII chars "7B"
Serial.print(val,BIN); // sends 8 ASCII chars "01111011"
Serial.print(val,BYTE); // sends 1 byte, the verbatim value
```


Управление компьютером

Вы пишете одну программу на Arduino, одну на компьютере

В Arduino: считать показания датчика, послать данные как байт

```
void loop() {  
  val = analogRead(analogInput); // read the value on analog input  
  Serial.print(val/4,BYTE);      // print a byte value out  
  delay(50);                    // wait a bit to not overload the port  
}
```

В Processing: прочитав байт, сделать с ним что-нибудь

```
import processing.serial.*;  
  
Serial myPort; // The serial port  
  
void setup() {  
  String portname = "/dev/tty.usbserial-A3000Xv0";  
  myPort = new Serial(this, myPort, 9600);  
}  
  
void draw() {  
  while (myPort.available() > 0) {  
    int inByte = myPort.read();  
    println(inByte);  
  }  
}
```

Но написание программ в Processing будет позже.

Управление компьютером

- Программа-адресат на компьютере может быть на любом языке, в котором возможна работа с последовательным портом
- C/C++, Perl, PHP, Java, Max/MSP, Python, Visual Basic, и т.д.
- Выберите Ваш любимый язык, напишите небольшой код для управления Arduino

Если интересно, я могу объяснить, как это делается в подробностях, практически в каждом из вышеуказанных языков.

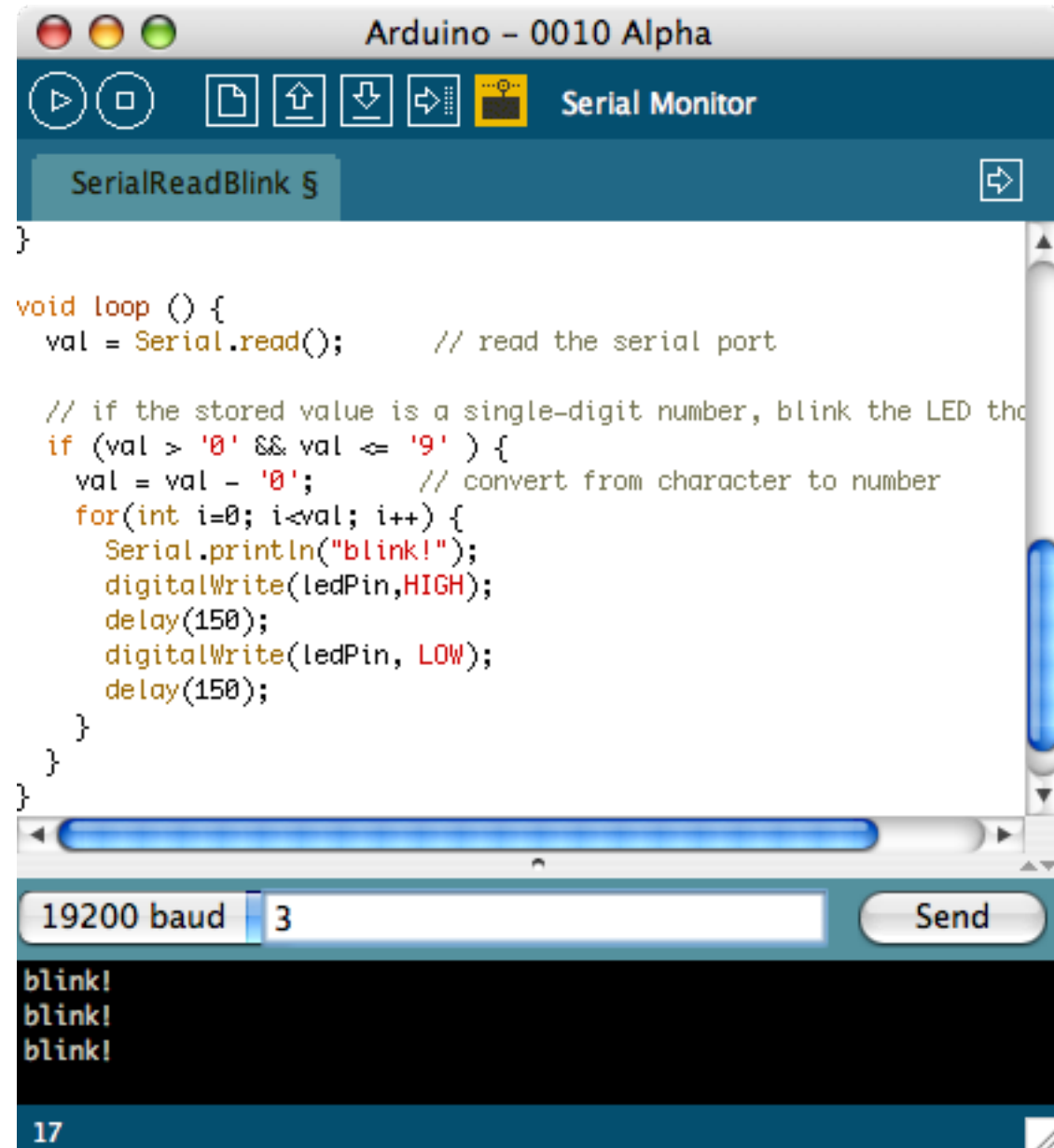
Управление Arduino, снова

"SerialReadBlink"

Введите число от 1 до 9, и светодиод мигнёт столько раз

Преобразует введённые ASCII значения в числа

Многие задачи управления сводятся к задачам преобразования данных



The screenshot shows the Arduino IDE Serial Monitor window titled "Arduino - 0010 Alpha". The window contains the following code:

```
SerialReadBlink 5  
}  
  
void loop () {  
  val = Serial.read();    // read the serial port  
  
  // if the stored value is a single-digit number, blink the LED the  
  if (val > '0' && val <= '9') {  
    val = val - '0';      // convert from character to number  
    for(int i=0; i<val; i++) {  
      Serial.println("blink!");  
      digitalWrite(ledPin,HIGH);  
      delay(150);  
      digitalWrite(ledPin, LOW);  
      delay(150);  
    }  
  }  
}
```

At the bottom of the window, the serial port is set to 19200 baud and the baud rate divider is set to 3. The "Send" button is visible. The output area shows three "blink!" messages.

Этот скетч тоже есть в раздаточном материале


RGB-светодиод, управляемый через последовательный порт

“SerialRGBLED”

Посылает в Arduino
цветовые команды

напр. “r200”, “g50”, “b0”

Скетч обрабатывает то,
что Вы вводите, и
изменяет цвет свечения



```
Arduino - 0010 Alpha
Serial Monitor
SerialRGBLED
void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();
    serInString[0] = 0; // indicates we've used th
    if(colorCode == 'r')
      analogWrite(redPin, 255-colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, 255-colorVal);
  }
}
```

19200 baud | b255| Send

enter color command (e.g. 'r43') :
setting color r to 255

32

Этот скетч есть в раздаточном материале.

Цветовая команда состоит из двух частей: colorCode и colorValue

colorCode — это символ, 'r', 'g', или 'b'.

colorValue — это число в диапазоне 0-255.

В скетче показана устаревшая обработка символьных строк в Arduino.

К сожалению, это всё ещё пока сложно.

Считываем строки из последовательного порта

- Функция `Serial.available()` упрощает считывание строк
- Её можно использовать для считывания всех данных, посылаемых с компьютера по последовательному протоколу
- Функция `readSerialString()` справа берет строку символов и приклеивает к ней данные, полученные из последовательного порта

```
//read a string from the serial and store it in an array
//you must supply the array variable
void readSerialString (char *strArray) {
    int i = 0;
    if(!Serial.available()) {
        return;
    }
    while (Serial.available()) {
        strArray[i] = Serial.read();
        i++;
    }
    strArray[i] = 0; // indicate end of read string
}
```

Не обращайте внимания на символ указателя ("*")

Будьте осторожны и не вызывайте `readSerialString()` слишком часто, иначе вы будете считывать части строк

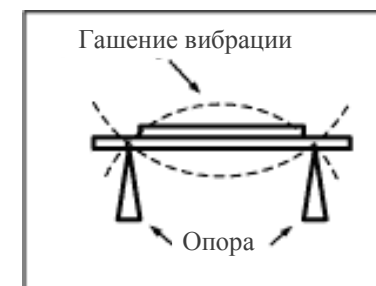
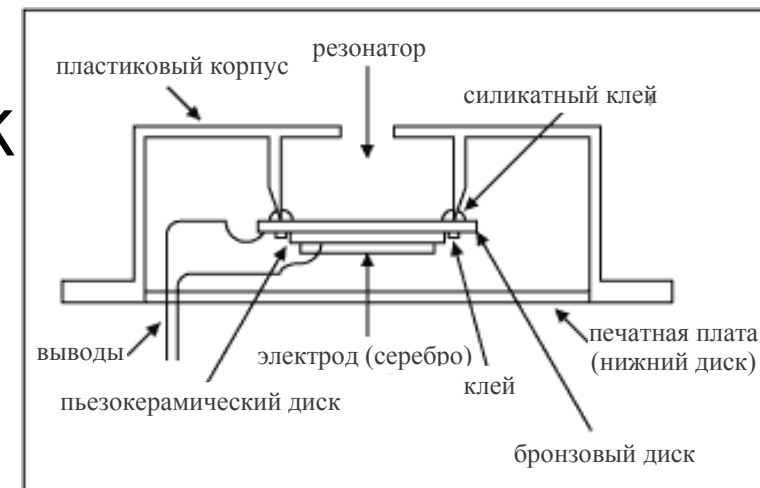
Пьезоэлектричество

- Сложное слово – *пъезо* по-гречески означает “давлю”
- Некоторые кристаллы при сжатии вырабатывают электрическую искру
- Оказывается, этот процесс действует и в обратную сторону тоже
- На кварцевый кристалл подаётся электричество, и он сжимается
- Пьезо-пищалки таким образом звучат (если нечто перемещать туда-обратно, оно будет колебать воздух)

У пьезо-пищалок внутри нет кристаллов кварца, а есть специальная керамика, которая тоже обладает пьезоэлектрическим свойством.

Пьезо-пищалки

- Два провода, красный и чёрный. Полярность важна: чёрный=земля
- Примените изменяющееся напряжение, получите звук
- Корпус пищалки содержит пьезоэлемент и имеет резонансную полость для звука



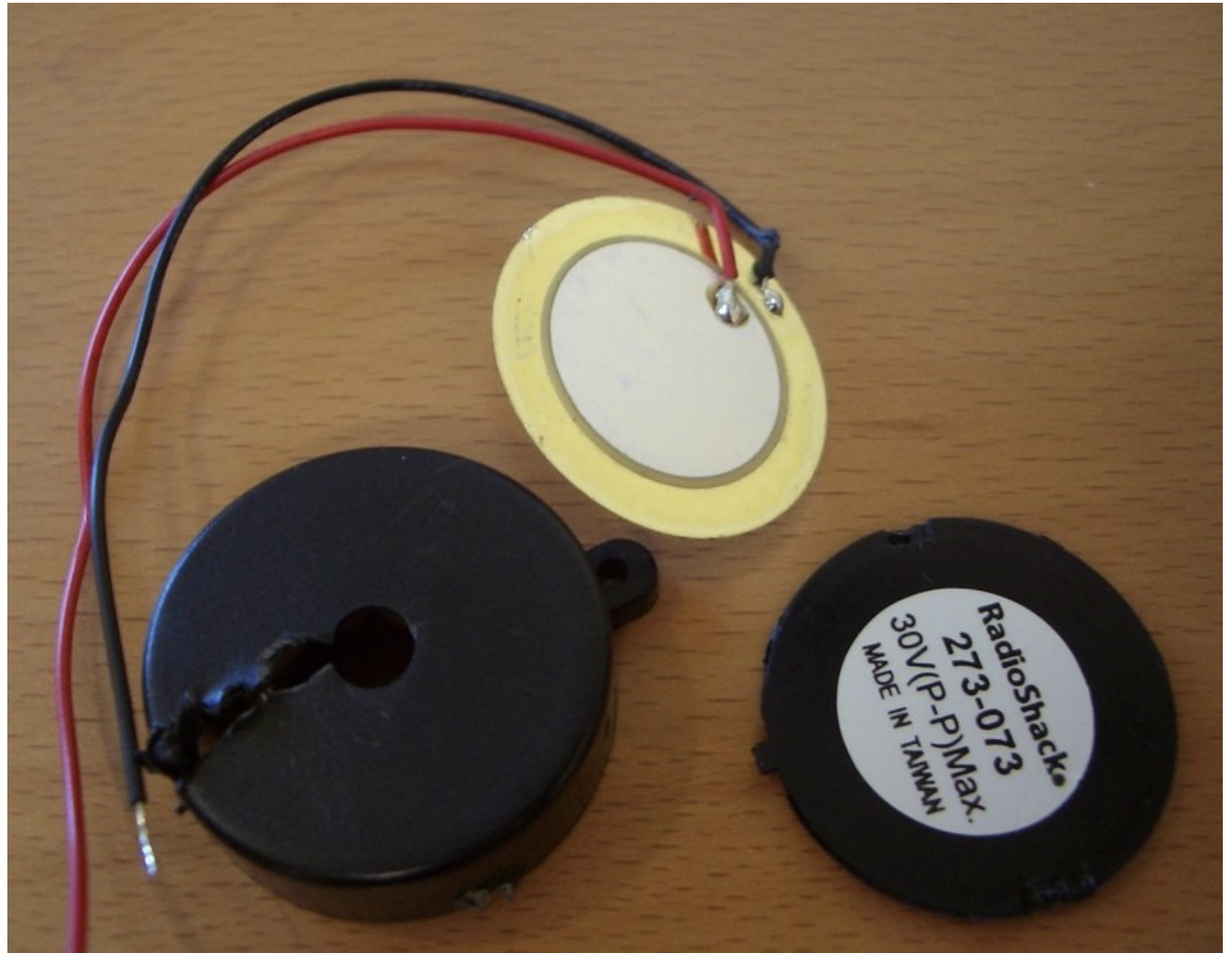
Колебание напряжения последовательно сжимает и отпускает пьезоэлемент. Должно быть изменяющееся напряжение, ровный логический уровень HIGH или LOW не подойдёт

Что внутри у пьезо-пищалки?

Добраться до
пьезоэлемента
довольно легко

Будьте осторожны -
белый диск и есть тот
самый пьезоэлемент

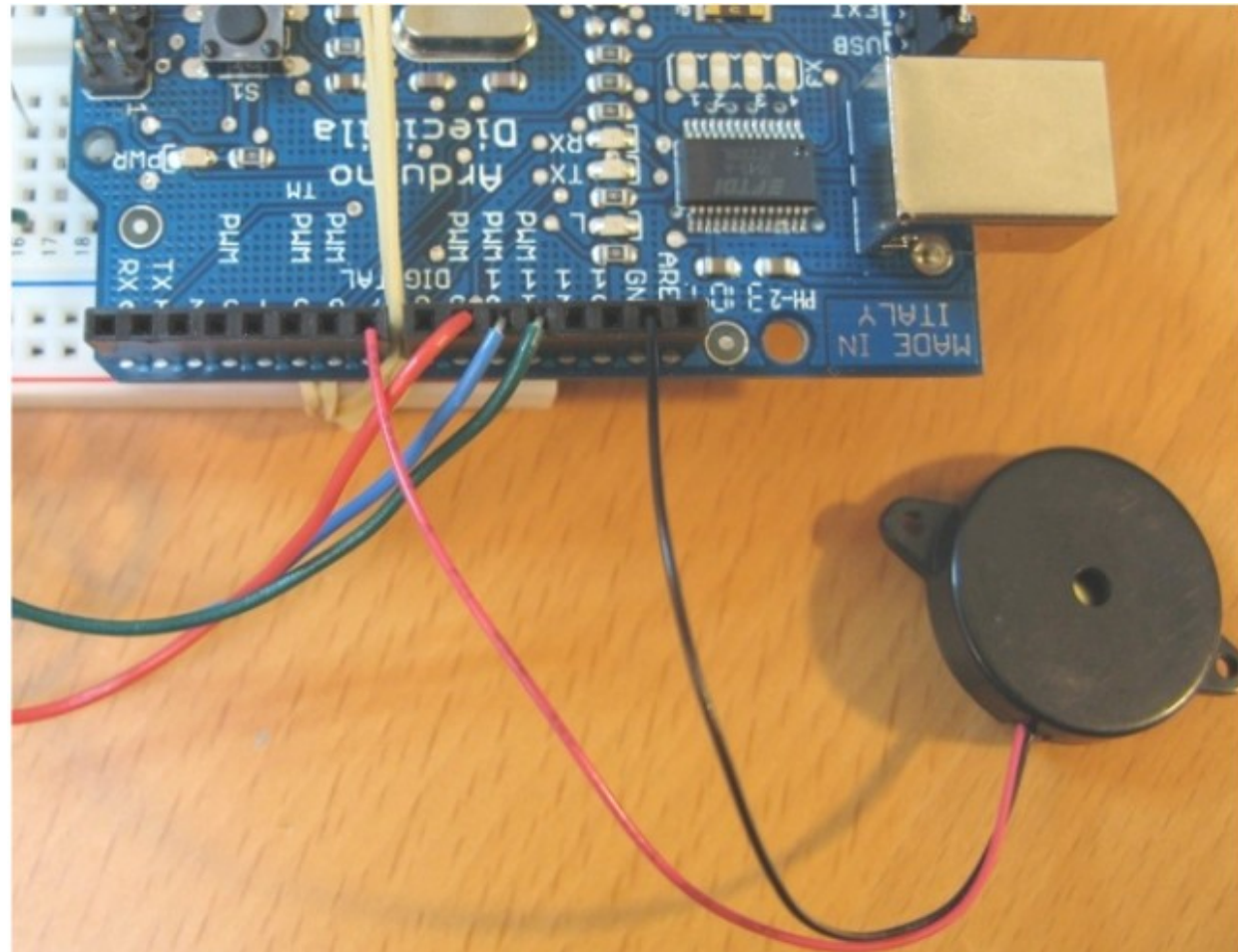
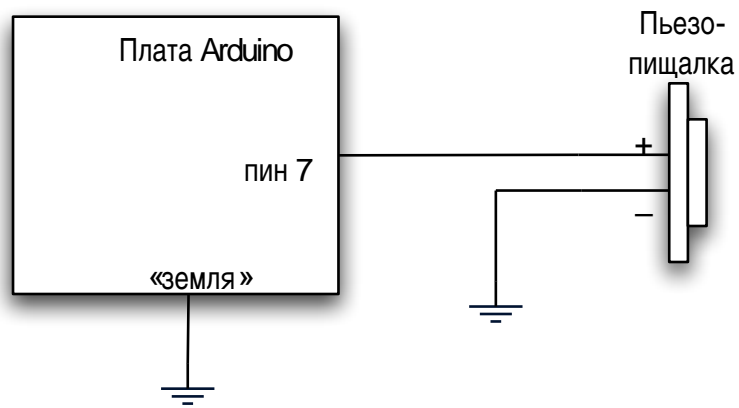
Просто выньте его
из корпуса, чтобы
использовать как
датчик



Мои \$1.99 мне уже не вернут назад из Radio Shack

Разумеется, Вы скорее всего разрушите оболочку, чтобы достать элемент.
Именно оболочка имеет подходящую структуру и резонансную полость для создания громкого звука

Пьезо-пищалка



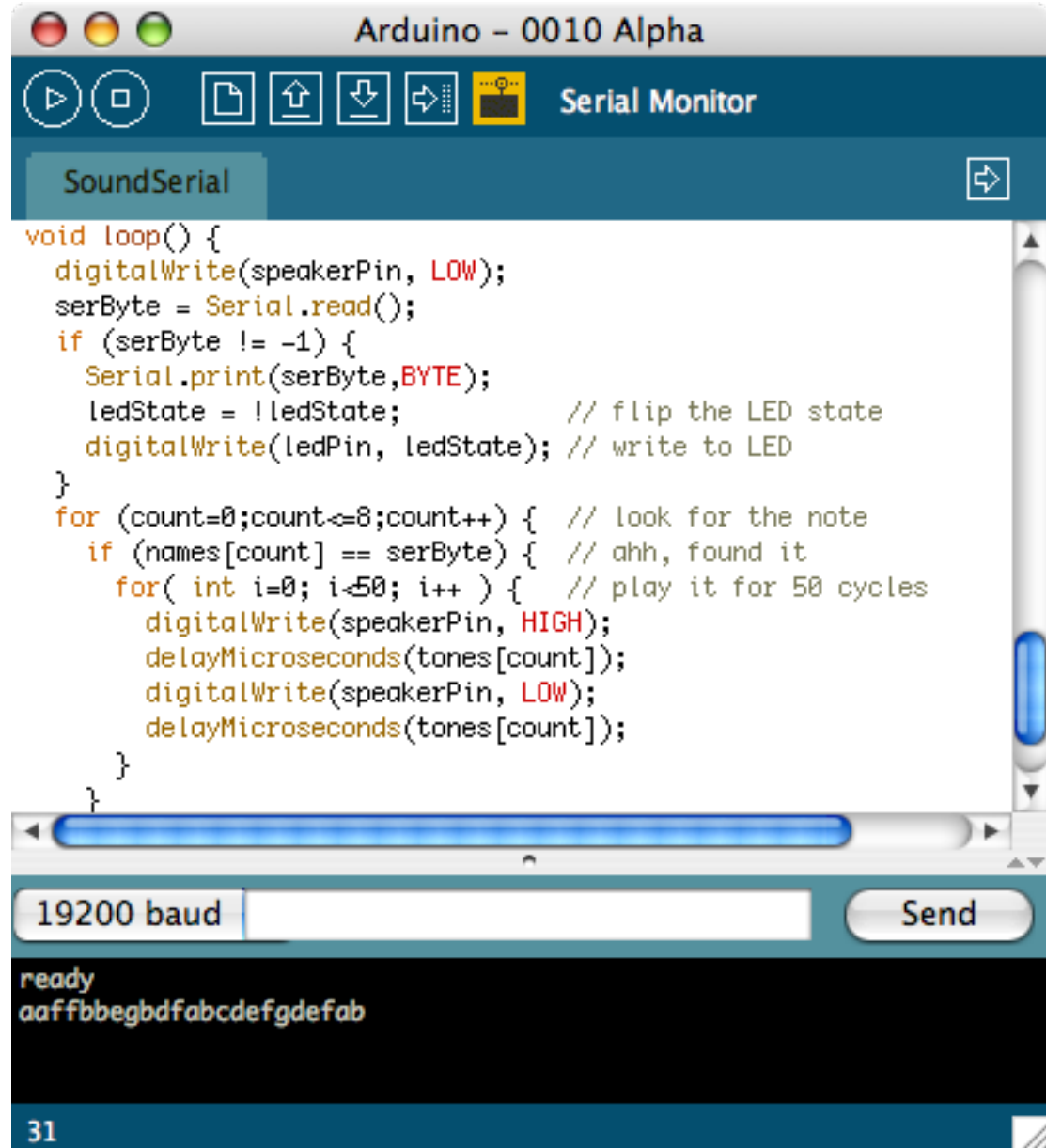
Выводы пьезо-пищалки очень тонкие. Отверстия в макетной плате держат их лучше, чем гнёзда, так что можно использовать дополнительные перемычки из проводов. Или можно вставить перемычки в отверстия вместе с выводами пьезо-пищалки, чтобы лучше держалось.

Сыграем мелодию

“SoundSerial”

Сыграем на
пьезо-пищалке
через Serial Monitor

Печатайте буквы
из “cdefgabC”, чтобы
получить мелодию



```
Arduino - 0010 Alpha
Serial Monitor
SoundSerial
void loop() {
  digitalWrite(speakerPin, LOW);
  serByte = Serial.read();
  if (serByte != -1) {
    Serial.print(serByte, BYTE);
    ledState = !ledState; // flip the LED state
    digitalWrite(ledPin, ledState); // write to LED
  }
  for (count=0;count<=8;count++) { // look for the note
    if (names[count] == serByte) { // ahh, found it
      for( int i=0; i<50; i++ ) { // play it for 50 cycles
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tones[count]);
        digitalWrite(speakerPin, LOW);
        delayMicroseconds(tones[count]);
      }
    }
  }
}
```

19200 baud Send

ready
aaffbbegbdfabcdefgdefab

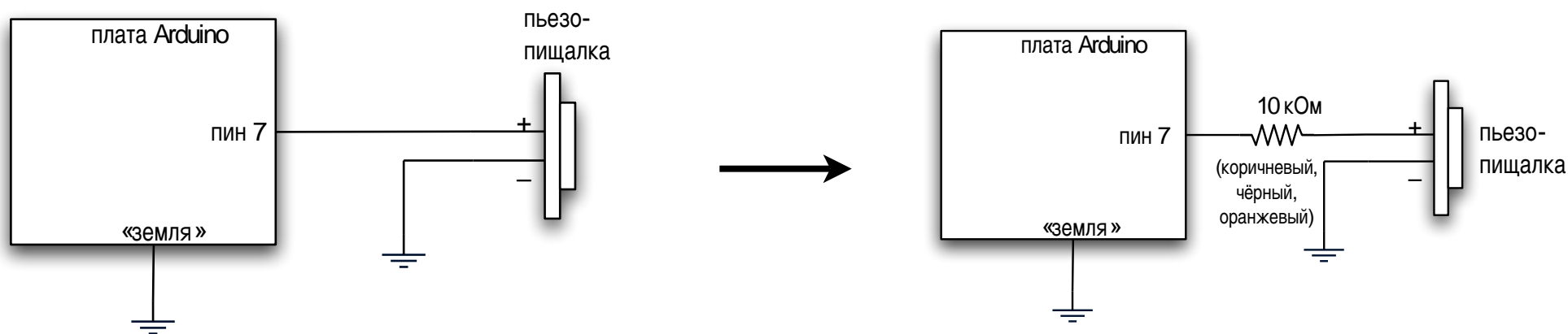
31

Этот скетч есть в раздаточном материале
Заметили, какая проблема с этим скетчем?

Разные ноты играют разное количество времени. 50 циклов ноты С низкой октавы занимают вовсе не столько же времени, сколько 50 циклов ноты В высокой октавы

Сделаем потише

Самый простой способ:
добавить резистор



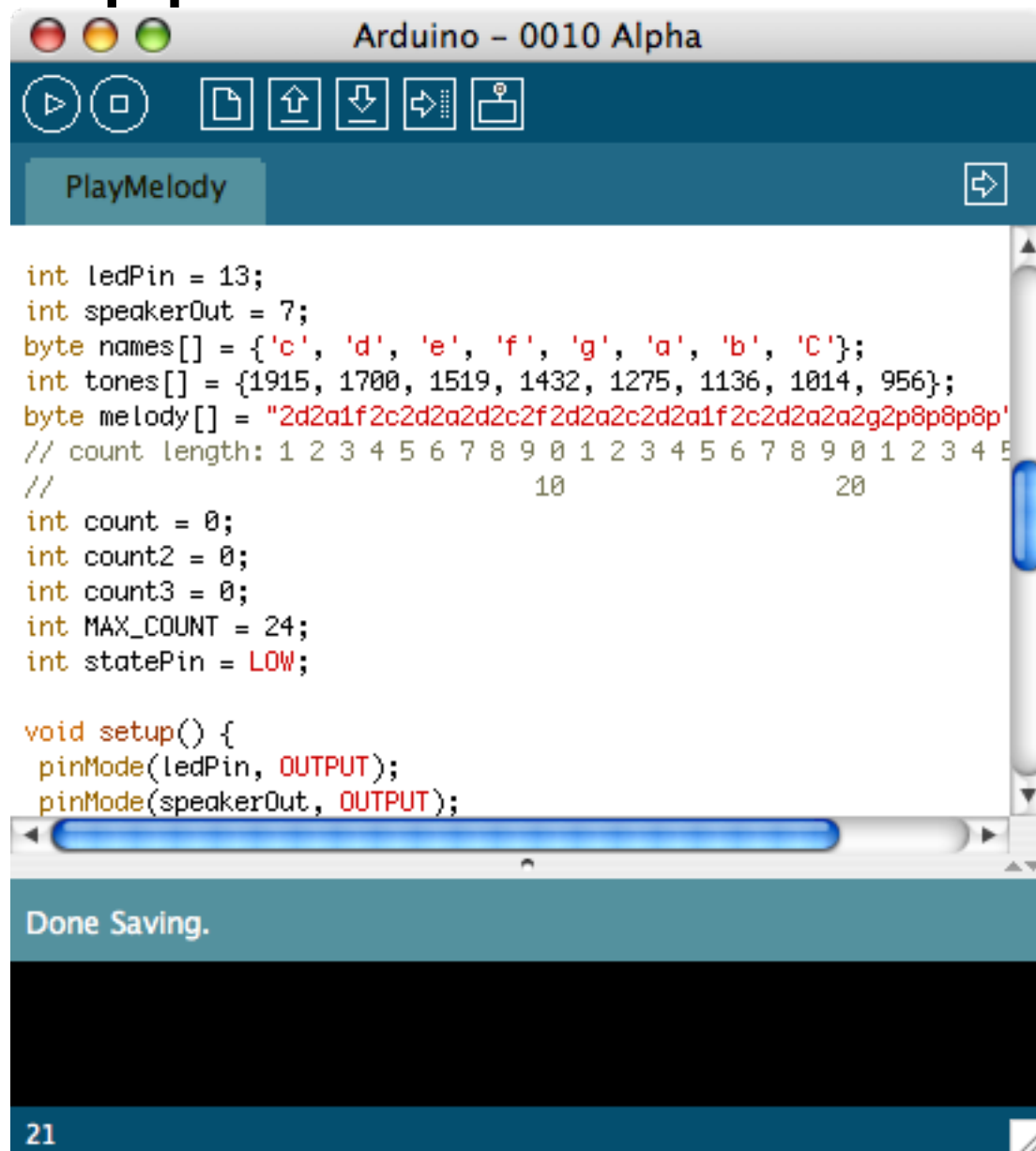
Как часто бывает в электронике, если вы хотите что-то уменьшить, добавьте резистор. Лучше подойдет номиналом 1 кОм, но у нас таких нет в наборах. Это может показаться неважным сейчас, но потерпите до следующего проекта.

Воспроизведём записанную мелодию

“PlayMelody”

Играет мелодию,
которая хранится
в Arduino

Можно питать от батарейки,
играть мелодию по нажатию
кнопки, менять скорость
воспроизведения
фоторезистором, и т.д.



```
int ledPin = 13;
int speakerOut = 7;
byte names[] = {'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C'};
int tones[] = {1915, 1700, 1519, 1432, 1275, 1136, 1014, 956};
byte melody[] = "2d2a1f2c2d2a2d2c2f2d2a2c2d2a1f2c2d2a2a2g2p8p8p8p"
// count length: 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
//                                     10                20
int count = 0;
int count2 = 0;
int count3 = 0;
int MAX_COUNT = 24;
int statePin = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(speakerOut, OUTPUT);
}
```

Done Saving.

21

Мелодии закодированы примерно так, как раньше рингтоны на мобильных телефонах
Логика исполнения мелодии довольно сложно понять, так как здесь очень важно просчитать время.

Сделаем терменвокс

“ooo-уиии-ooooo”

Та самая машина,
создающая жутковатые
звуки.

Измеряет электрическое
поле Вашего тела.

Не требуется
прикосновений!

Мы будем использовать
свет вместо
электромагнитного поля



Лев Термен

Можно услышать в фильме «Звездный путь», музыке Beach Boys, фильмах ужасов, «Марс атакует!», и плохих песнях в стиле New Age.

Работает как сенсорные кнопки, но здесь не требуется никаких касаний.

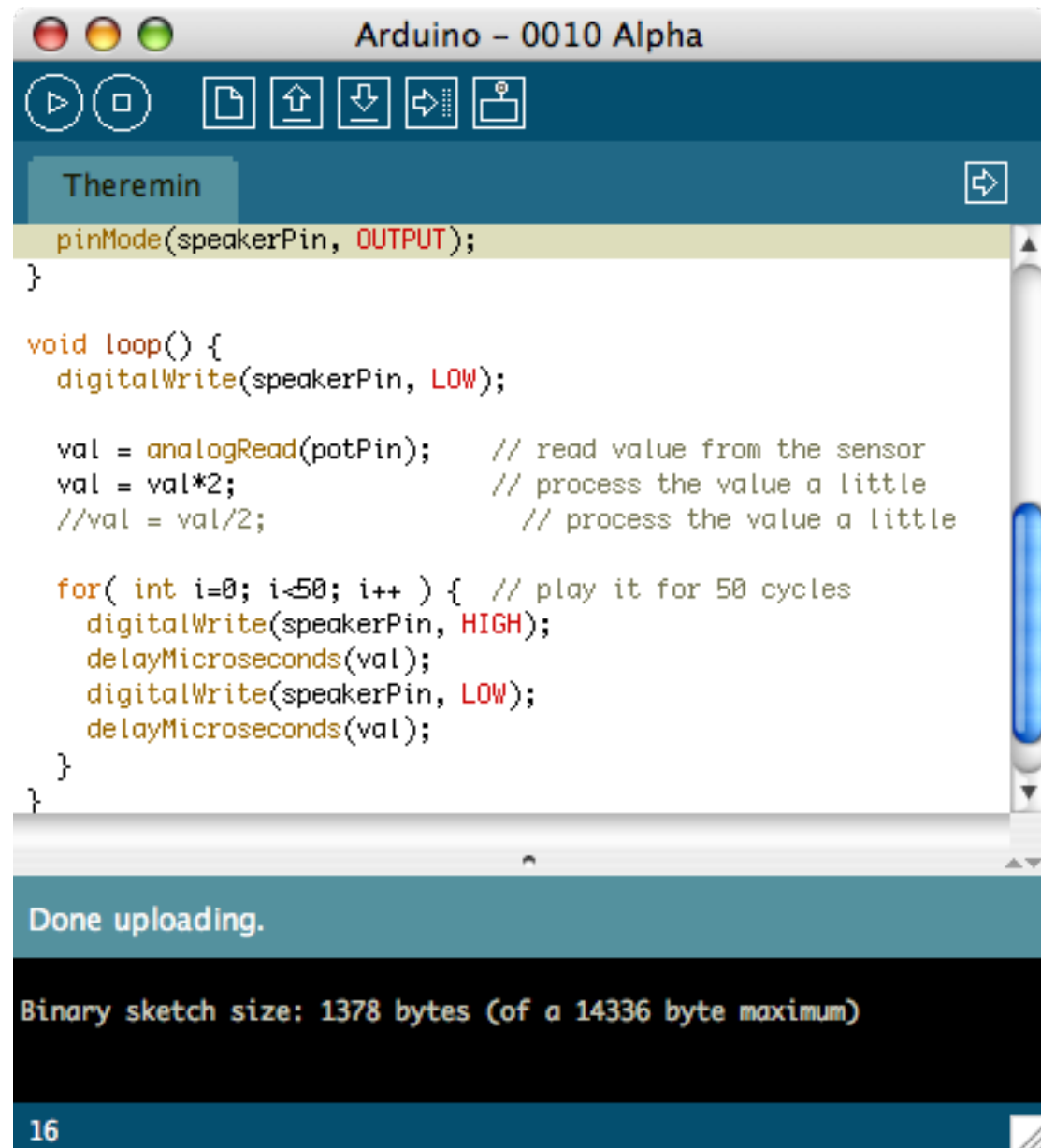
По следующему принципу: Ваше тело становится конденсатором с переменной ёмкостью.

Световой терменвокс

“Theremin”

Проведите рукой над фоторезистором, чтобы изменить высоту звука

Поиграйте с обработкой величины и счётчиком циклов, чтобы изменить чувствительность, высоту и тембр



```
Arduino - 0010 Alpha

Theremin

pinMode(speakerPin, OUTPUT);
}

void loop() {
  digitalWrite(speakerPin, LOW);

  val = analogRead(potPin); // read value from the sensor
  val = val*2; // process the value a little
  //val = val/2; // process the value a little

  for( int i=0; i<50; i++ ) { // play it for 50 cycles
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(val);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(val);
  }
}

Done uploading.

Binary sketch size: 1378 bytes (of a 14336 byte maximum)

16
```

Да, пока звучит скорее как плохая видеоигра, чем как страшный фильм
Звук прерывается, потому что необходимо время на считывание данных сенсором
Это можно исправить более сложным программированием с использованием таймеров и прерываний.
Этот звук очень скоро надоедает

Другие устройства с последовательным подключением



Wi-Fi

Ethernet



графический ЖК-экран

8-канальный сервоконтроллер

Примеры устройств с последовательным подключением



Робот-пылесос Roomba

Идём дальше

- Пьезо-пищалки
 - Можно соединить несколько пищалок для полифонического звука
 - Могут играть звуковые волны, отличные от простых квадратных, при помощи ШИМ-технологии
 - Могут также служить устройствами ввода (это будет объяснено позднее)

Идём дальше

- Коммуникация через СОМ-порт
 - Не только для соединения Arduino с компьютером
 - Многие другие устройства общаются через последовательный порт
 - Например, старые клавиатуры и мыши (годятся в качестве сенсоров!)
 - Интерфейсные платы (графические ЖК-экраны, сервоконтроллеры, RFID-считыватели, Ethernet, Wi-Fi)

Идём дальше

- RGB-светодиоды
 - Вы можете с лёгкостью повторить функционал лампы настроения (\$150)
 - Сделайте индикатор статуса Вашего компьютера
 - Контролируемое компьютером направленное освещение (цветовая волна вдоль стен)



Впрочем, лампа настроения не подключается к компьютеру, а использует пейджинговую сеть Ambient Devices: <http://www.ambientdevices.com/>

КОНЕЦ занятия 2

<http://todbot.com/blog/bionocarduino/>

Тод Е. Курт

tod@todbot.com

Не стесняйтесь писать мне на почту, если есть вопросы.