Лабораторная работа № 1 Светодиод 2

Сборка элементов на плате 4

Лабораторная работа № 2 Управляемый «программно» светодиод 7

Лабораторная работа № 3 Управляемый вручную светодиод 10

Сборка элементов на плате 12

Лабораторная работа № 4.1 Пьезодинамик 13

Сборка элементов на плате 13

Лабораторная работа № 4.2 Управляемый пьезодинамик 15

Сборка элементов на плате 15

Лабораторная работа № 5 Фоторезистор 17

Сборка элементов на плате 19

Лабораторная работа № 6 Кнопка 20

Лабораторная работа № 7 Термистор 22

Лабораторная работа № 8 Синтезатор 24

Лабораторная работа № 9 Взаимодействие Arduino с семисегментным индикатором 25

Лабораторная работа № 10 Обмен данными Arduino с ПК 27

Лабораторная работа № 11 Дисплей LCD 12С интерфейс 28

Лабораторная работа № 12 Сервопривод 30

Лабораторная работа № 13 Шаговый двигатель 32

Лабораторная работа № 13 Двигатель постоянного тока 35

Лабораторная работа № 14 ИК-датчик и ИК пульт 37

Лабораторная работа № 15 Bluetooth модуль 40

Лабораторная работа № 16 Дальномер 42

Лабораторная работа № 17 Датчик скорости 44

Приложения 46

Работа с набором «Конструктор программируемых моделей инженерных систем». Первое подключение. 4

Звездные войны 46

Виртуальный тренажёр на сайте К. Полякова 49

Скетч «Светофор» на тренажёре К. Полякова 1

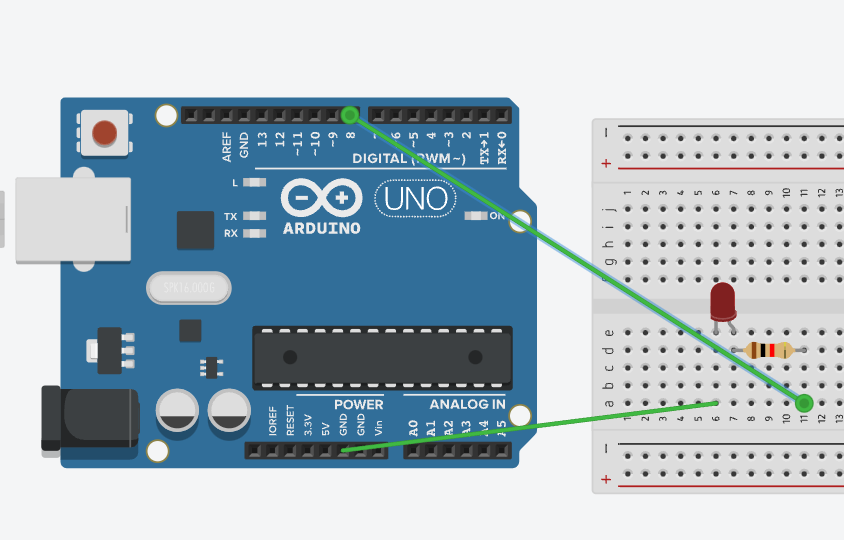
Скетч «Светофор» на Ардуино 7

Лабораторная работа № 1 Светодиод

**Цель:** Научиться работать с виртуальным тренажёром **Arduino**

1. Зайдите на сайт: https://www.tinkercad.com/joinclass/YXHUNU9GQUEQ1
2. Авторизуйтесь используя один их ниже представленных аккаунтов
3. Выберите **Цепи**– **Создать цепь**
4. Введите название проекта **Проект 1**
5. Перетащите на рабочее поле **Arduino Uno 3**и дайте ему имя **Ардуино**
6. Перетащите на рабочее поле **Малую макетную плату (Breadbord)**и дайте ему имя **Плата**
7. Перетащите на рабочее поле **Светодиод (Led)**и дайте ему имя**Светодиод**
8. Перетащите на рабочее поле **Резистор (Resistor)**и дайте ему имя**Резистор, 220 Ω**

|  |
| --- |
|  |
| **Поворот элемента соединитель** |

1.  Поверните **Резистор**, выберите соединитель **обычный**Соедините элементы так, как показано на рисунке**:**

1. Нажмите кнопку **Код**. Из выпадающего списка выберите **Текст**.
2. Введите код программы (скетча), можно скопировать и вставить:

int led = 8;

void setup()

{

pinMode(led, OUTPUT);

}

void loop()

{

digitalWrite(led, HIGH);

delay(1000);

digitalWrite(led, LOW);

delay(1000);

}

1. Нажмите кнопку **Начать моделирование** (светодиод должен моргать)
2. Нажмите кнопку **Остановить моделирование**(светодиод перестает моргать)

*Пояснение кода:*

**int led = 8;**  //объявление переменной целого типа, содержащей номер порта к которому мы подключили второй провод

**void setup()**  //обязательная процедура setup, запускаемая в начале программы; объявление процедур начинается словом void

**pinMode(led, OUTPUT);** //объявление используемого порта, **led** - номер порта, второй аргумент - тип использования порта - на вход (INPUT) или на выход (OUTPUT)

**void loop()** //обязательная процедура loop, запускаемая циклично после процедуры setup

**digitalWrite(led, HIGH);** //эта команда используется для включения или выключения напряжения на цифровом порте; **led**- номер порта, второй аргумент - включение **(HIGH)** или выключение **(LOW)**

**delay(1000);** //эта команда используется для ожидания между действиями, аргумент - время ожидания в **миллисекундах (1 с = 1000 мс)**

**Дополнительное задание.**Запустите модель с новыми параметрами:

Измените частоту мигания светодиода с периодом 2 с; 0, 5 с

Переключите светодиод на пин 7 и отредактируйте код на **int led = 7;**

Лабораторная работа № 1.1 Работа с набором «Конструктор программируемых моделей инженерных систем». Первое подключение.

1. Запустить на компьютере программу **Arduino IDE**
2. Подключите микроконтроллер через USB кабель к компьютеру
3. Выберите в программе нужную плату и порт. **Инструменты – плата –**  
   **Arduino Mega or Mega 2650**. **Инструменты** – **порт** – СОМ 4 (цифра может быть другой)

Наберите тестовую программу:  
  
int led = 13;

void setup()

{

pinMode(led, OUTPUT);

}

void loop()

{

digitalWrite(led, HIGH);

delay(1000);

digitalWrite(led, LOW);

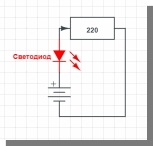
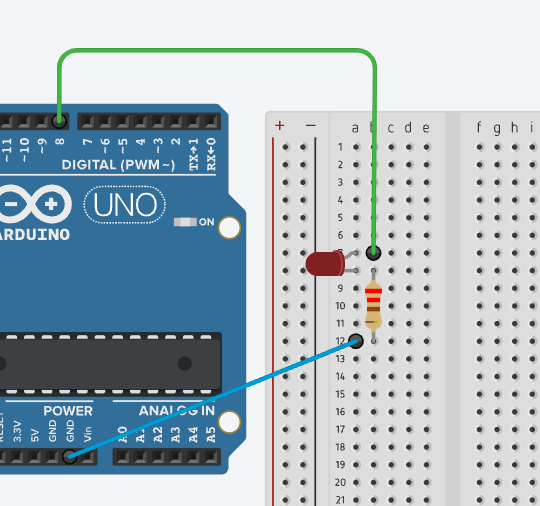
delay(1000);

}

1. Нажмите кнопку **Загрузка.**(программа попросит сохранить файл, сохраните, выбрав имя **test)  
   Компьютер должен быть подключен к Интернет! Разрешить брандмауэру выход программы в Интернет.**
2. После загрузки скетча на плате начнет мигать светодиод

Лабораторная работа № 1.2 **Сборка элементов на плате**

Оборудование: Макетная плата, 2 провода папа-папа, светодиод, резистор на 220 Ом.



Общие контакты

GND

Pin 8

**Сборка элементов на плате производится по схеме:**

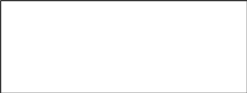
Верхний провод соединяет свободный конец светодиода и пин под номером **8** на плате

Нижний провод соединяет своболный конец резистора и контакт **GDN** (минус или земля)

Получается последовательная цепь.

Введите код в программу **Arduino IDE**  
int led = 8;

void setup()

1. 

Сохраните программу под именем Lab1.2

1. Нажмите кнопку **Загрузка.**
2. После загрузки скетча светодиод начнет мигать

{

pinMode(led, OUTPUT);

}

void loop()

{

digitalWrite(led, HIGH);

delay(1000);

digitalWrite(led, LOW);

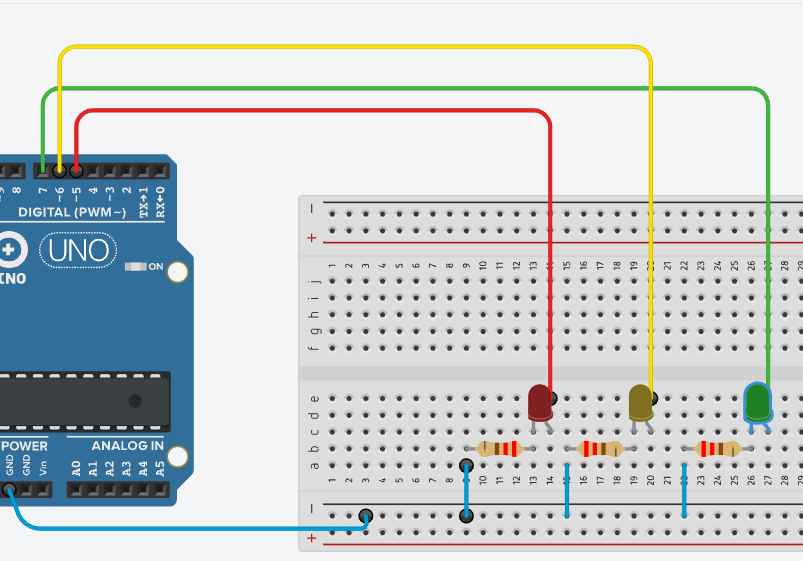
delay(1000);

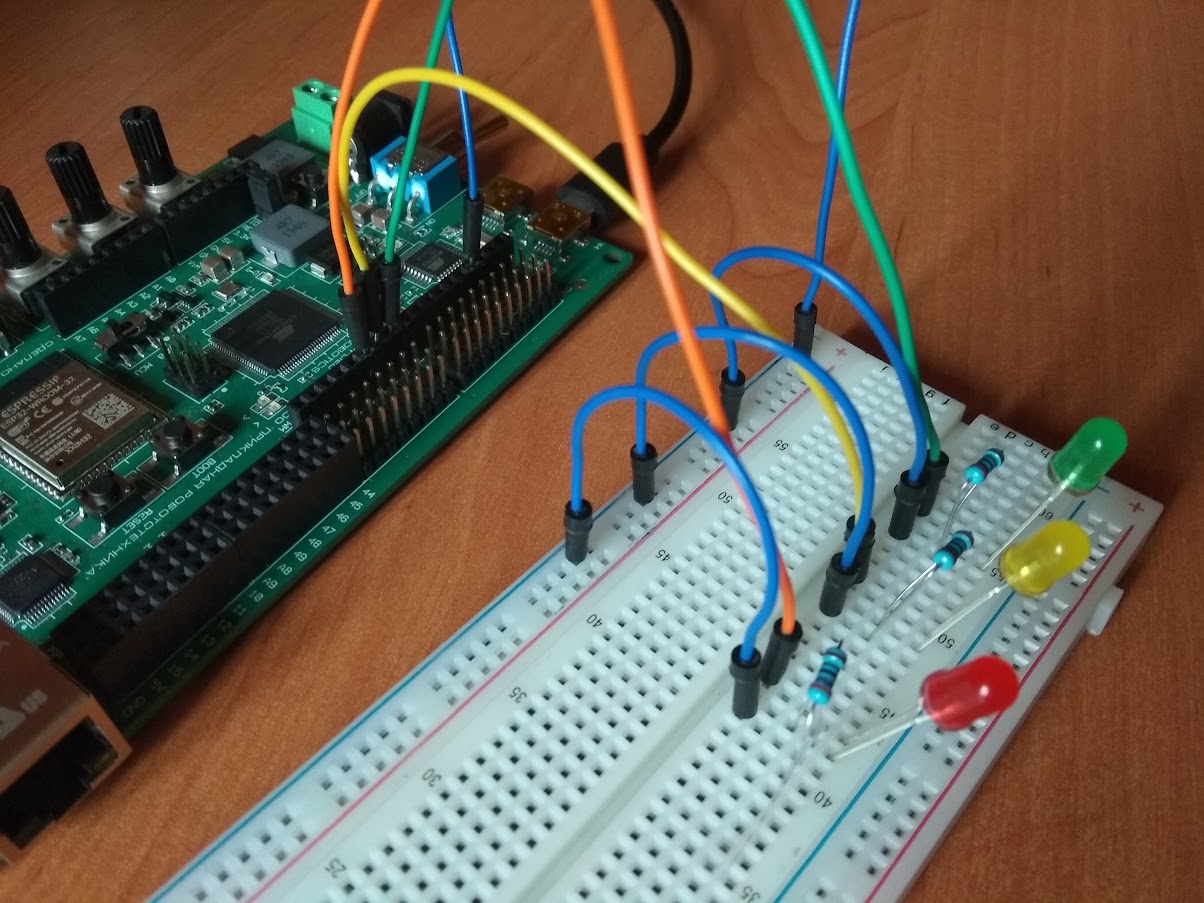
}

Приложение к ЛР № 1.2

|  |
| --- |
| Загрузка  Pin 8  GND |
| Если светодиод не горит, поменяйте контакты светодиода |

Лабораторная работа 1.3 Проект «Светофор» на Ардуино





**Код программы**

int led\_G = 7;

int led\_Y = 6;

int led\_R = 5;

void setup()

{

pinMode(led\_R, OUTPUT);

pinMode(led\_Y, OUTPUT);

pinMode(led\_G, OUTPUT);

}

void loop()

{

//red

digitalWrite(led\_R, HIGH);

delay(3000);

//===red-yellow

digitalWrite(led\_Y, HIGH);

delay(2000);

digitalWrite(led\_R, LOW);

digitalWrite(led\_Y, LOW);

//===green

digitalWrite(led\_G, HIGH);

delay(3000);

digitalWrite(led\_G, LOW);

//==yellow

digitalWrite(led\_Y, HIGH);

delay(2000);

digitalWrite(led\_Y, LOW);

delay(1000);

}

Лабораторная работа № 2 Управляемый «программно» светодиод

**Цель работы:** понять принцип работы широтно-импульсной модуляции (ШИМ). Научиться использовать ШИМ в проектах на базе Arduino.

Широтно-импульсная модуляция (ШИМ) - это метод получения изменяющегося аналогового значения посредством цифровых устройств.

Для ШИМ используется функция **analogWrite**(nun, значение); где пин - номер контакта Arduino, значение - число от О до 255. Например, значение 153 будет соответствовать величине тока в 3 вольта, а 76 - 1,5 вольт. 255 – 5 вольт. 0 – 0 вольт.

**Ход работы:**

Соберите схему (см. рис). Напишите код скетча.

|  |  |
| --- | --- |
|  | // ШИН  int led = 6;  void setup()  {  pinMode(led, OUTPUT);  }  void loop()  {  //3 Вольта  analogWrite(led, 153);  delay(1000);  //1,5 Вольта  analogWrite(led, 76);  delay(1000);  //5 Вольт  analogWrite(led, 255);  delay(1000);  } |

Светодиод должен мигать с переменной яркостью

**Доп. Задание. Измените код так, чтобы**

1. Изменилась скорость мигания светодиода
2. Изменилась яркость мигания светодиода

Источник: Лабораторная работа №4 «Широтно-импульсная модуляция» - Программирование микроконтроллера Arduino в информационно-управляющих системах (bstudy.net)

Лабораторная работа № 3 Управляемый вручную светодиод

**Цель:**Знакомство с устройством Потенциометра

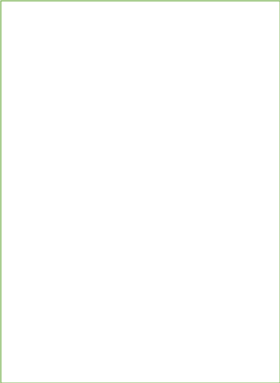
**Потенциометр** — это переменный резистор с регулируемым сопротивлением. Потенциометры используются в робототехнике как регуляторы различных параметров — громкости звука, мощности, напряжения и т.п. В нашей модели от поворота ручки потенциометра будет зависеть яркость светодиода

**Оборудование:** Малая макетная плата, резистор (220 В), светодиод, потенциометр,

Соединители: удлинитель -1, провода: папа-мама-3, папа-папа -3)

Соберите схему:  
Тип провода – Схема. После соединения измените цвет провода.

|  |
| --- |
| Земля |

// даём имена пинов со светодиодом

// и потенциометром

int led =9

int pot= A0

void setup()

{

// пин со светодиодом — выход

pinMode(led, OUTPUT);

// пин с потенциометром - вход

pinMode(pot, INPUT);

}

void loop()

{

// объявляем переменную x

int x;

// считываем напряжение с потенциометра:

// будет получено число от 0 до 1023

// делим его на 4, получится число в диапазоне

// 0-255 (дробная часть будет отброшена)

x = analogRead(pot) / 4;

// выдаём результат на светодиод

analogWrite(led, x);

}

**Код программы:**

int led = 9;

int pot = A0;

void setup()

{

pinMode(led, OUTPUT);

pinMode(pot, INPUT);

}

void loop()

{

int x;

x = analogRead(pot)/4;

analogWrite(led, x);

}

Загрузите программу и покрутите ручкой потенциометра – яркость светодиода должна изменятся

Источник:  
Arduino для начинающих. Урок 3. Подключение потенциометра | Занимательная робототехника (edurobots.ru)

**Сборка элементов на плате**

|  |
| --- |
|  |

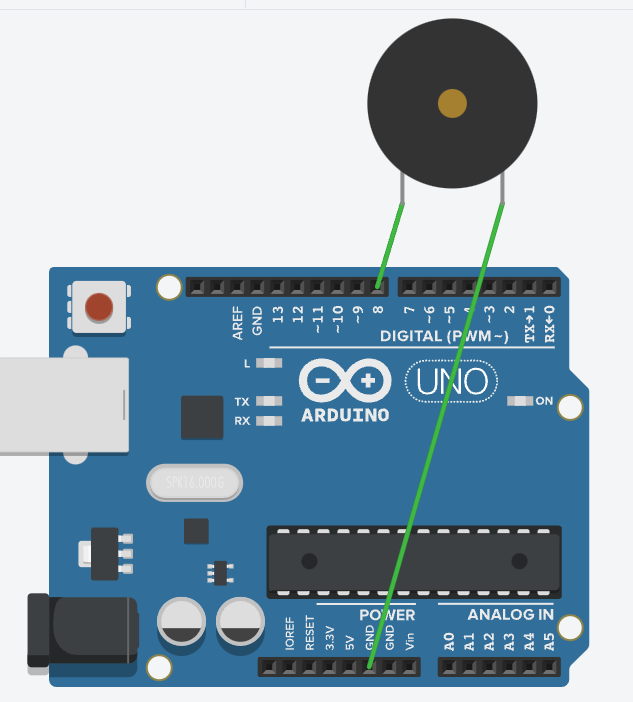
* Если светодиод не горит - поменяйте контакты светодиода
* Если от потенциометра идёт дым и запах отключите питание. Установите ручку потенциометра в среднее положение и повторите попытку
* У Потенциометра полярность крайних контактов (+ и -) произвольная

Лабораторная работа № 4.1 Пьезодинамик

**Цель:** Знакомство с работой пьезоэлемента

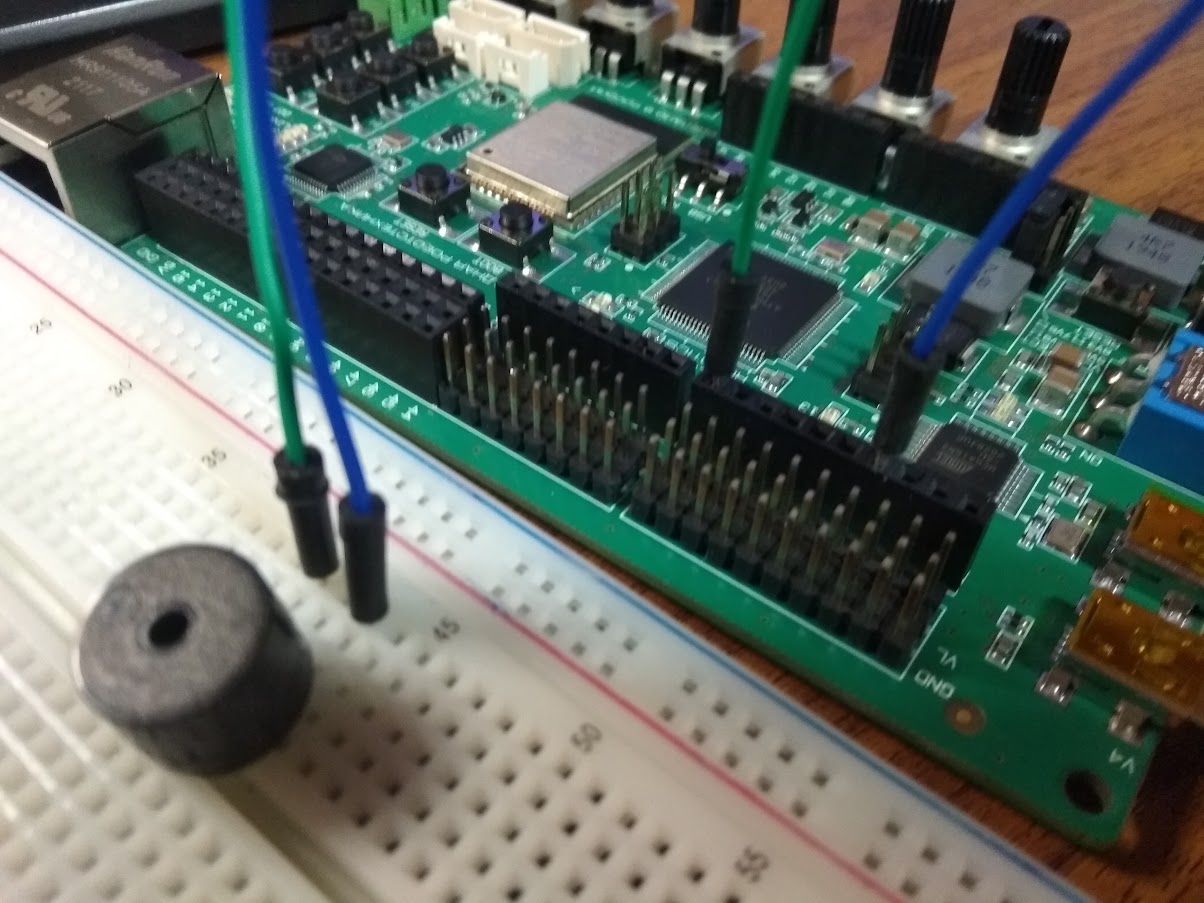
Источник:  
Arduino для начинающих. Урок 6. Подключение пьезоэлемента | Занимательная робототехника (edurobots.ru)

**Пьезоэлемент**— электромеханический преобразователь, одним из разновидностей которого является пьезоизлучатель звука, который также называют пьезодинамиком, просто звонком или английским **buzzer**. Пьезодинамик переводит электрическое напряжение в колебание мембраны. Эти колебания и создают звук



1. Доп. Информация:  
   Приложение: Звездные войны

**Сборка элементов на плате**



**Пояснение кода:**

int p = 3; //объявляем переменную с номером пина, на который мы

//подключили пьезоэлемент

void setup() //процедура setup

{

pinMode(p, OUTPUT); //объявляем пин как выход

}

void loop() //процедура loop

{

tone (p, 500); //включаем на 500 Гц

delay(100); //ждем 100 мс

tone(p, 1000); //включаем на 1000 Гц

delay(100); //ждем 100 мс

}

**Код программы:**

int p = 3;

void setup()

{

pinMode(p, OUTPUT);

}

void loop() //процедура loop

{

tone (p, 500);

delay(100);

tone(p, 1000);

delay(100);

}

Лабораторная работа № 4.2 Управляемый пьезодинамик

Цель: Получение звука переменной частоты с помощью потенциометра.

|  |
| --- |
|  |

int buzzer\_pin = 3;

int pot\_pin = A0;

void setup()

{

pinMode(buzzer\_pin, OUTPUT);

}

void loop() //процедура loop

{

int rotation, frequency;

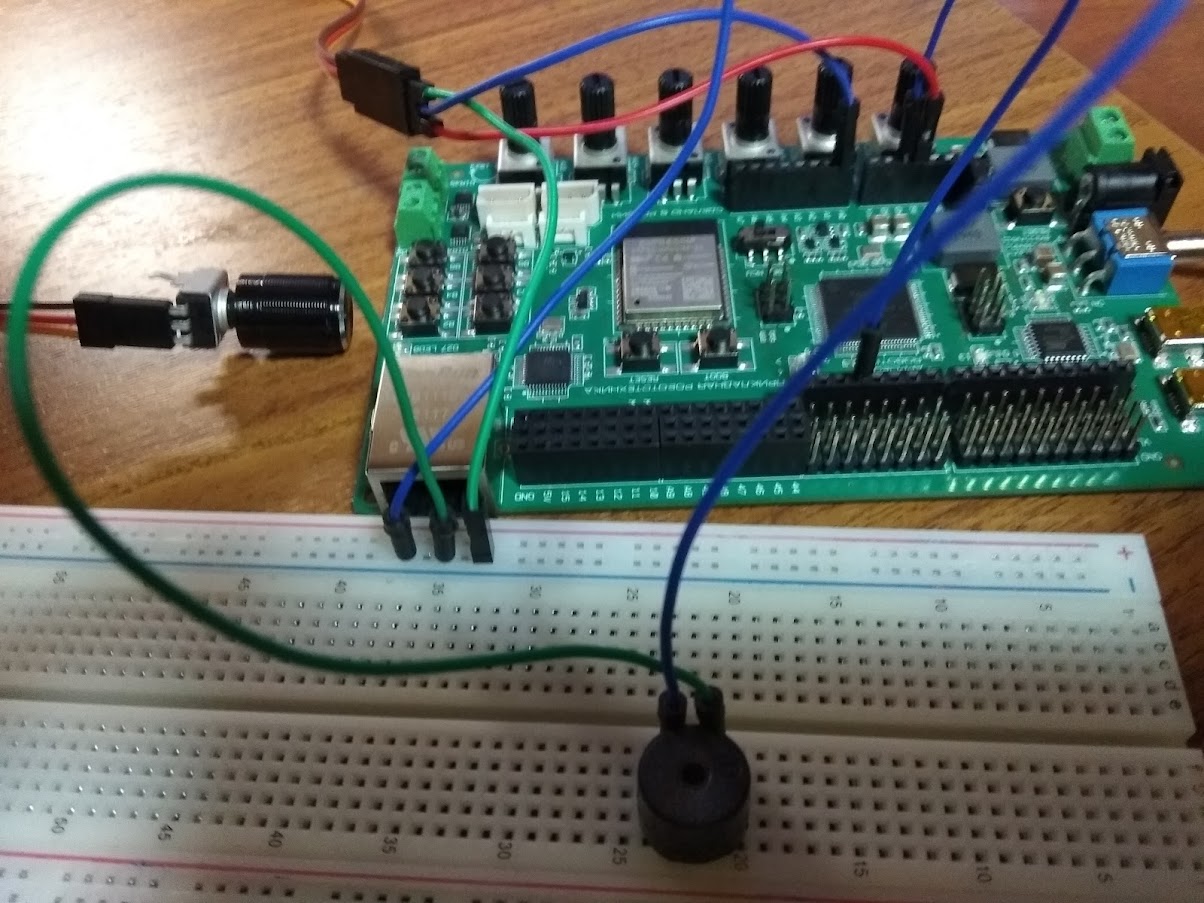
rotation = analogRead(pot\_pin);

frequency = map(rotation,0,1023,3500,4500);

tone (buzzer\_pin, frequency,20);

}

**Сборка элементов на плате**



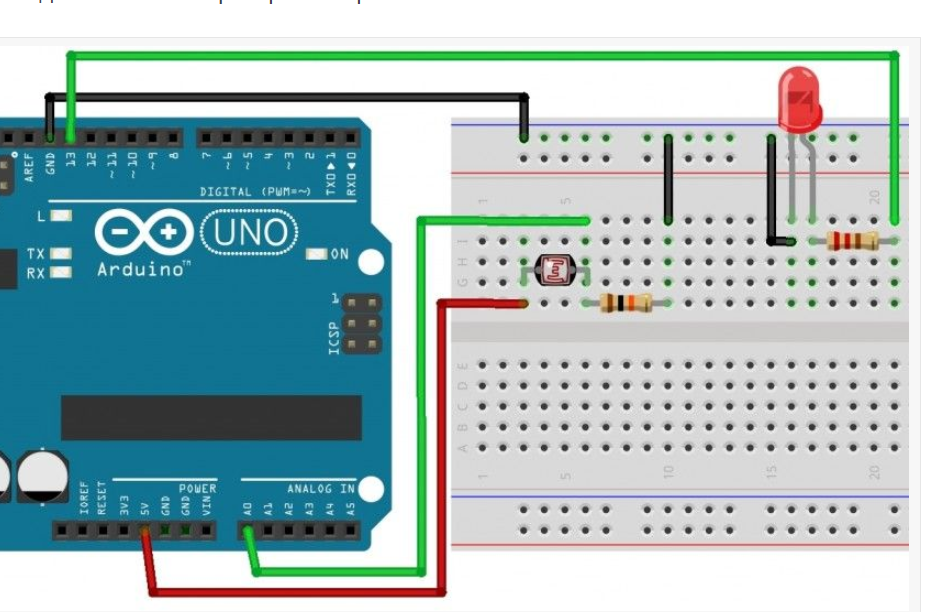
Лабораторная работа № 5 Фоторезистор

Цель: Знакомство работой фоторезистора

**Фоторезистор**— резистор, сопротивление которого зависит от яркости света, падающего на него. Фоторезисторы используются в робототехнике как датчики освещенности. Встроенный в робота фоторезистор позволяет определять степень освещенности, определять белые или черные участки на поверхности и в соответствие с этим двигаться по линии или совершать другие действия.

**Оборудование:** 6 проводов “папа-папа”, фоторезистор, светодиод, резистор на 220 Ом, резистор на 10 кОм

**Схема подключения:**



**Код программы:**

int led = 13; //переменная с номером пина светодиода

int ldr = 0; //и фоторезистора

void setup() //процедура setup

{

pinMode(led, OUTPUT); //указываем, что светодиод - выход

}

void loop() //процедура loop

{

if (analogRead(ldr)

//если показатель освещенности меньше 800, включаем светодиод

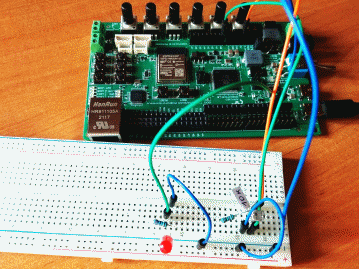
else digitalWrite(led, LOW); //иначе выключаем

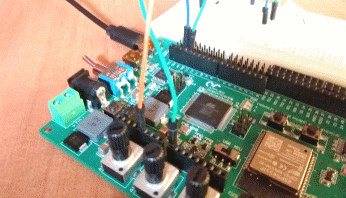
}

Источник:

Arduino для начинающих. Урок 7. Подключение фоторезистора | Занимательная робототехника (edurobots.ru)

**Сборка элементов на плате**

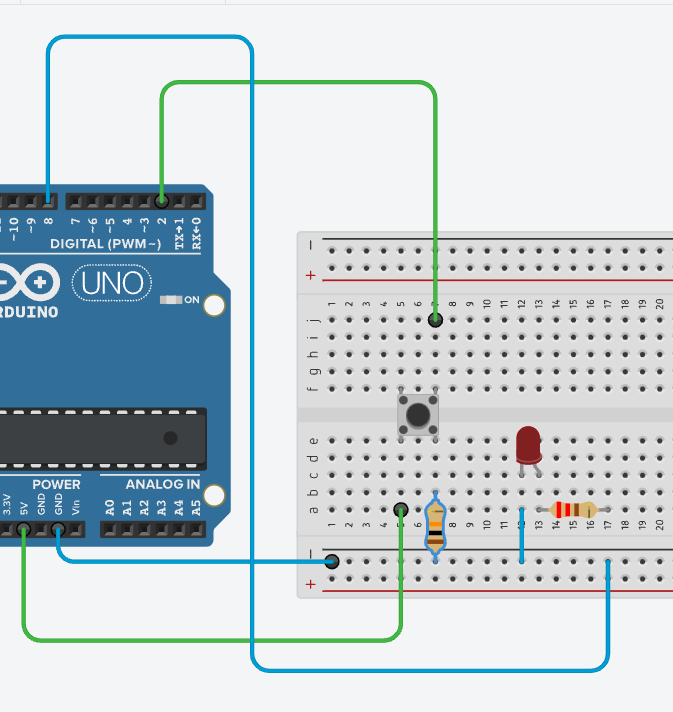




Лабораторная работа № 6 Кнопка

Цель: управление светодиодом с помощью кнопки

**Сегодня подключаем к ардуино кнопку и светодиод (при нажатой кнопке светодиод будет гореть, при отжатой — не гореть)**



**Код:**

int button = 2;

int led = 8;

void setup() {

pinMode(led, OUTPUT);

pinMode(button, INPUT);

}

void loop(){

if (digitalRead(button) == HIGH) {

digitalWrite(led, HIGH);

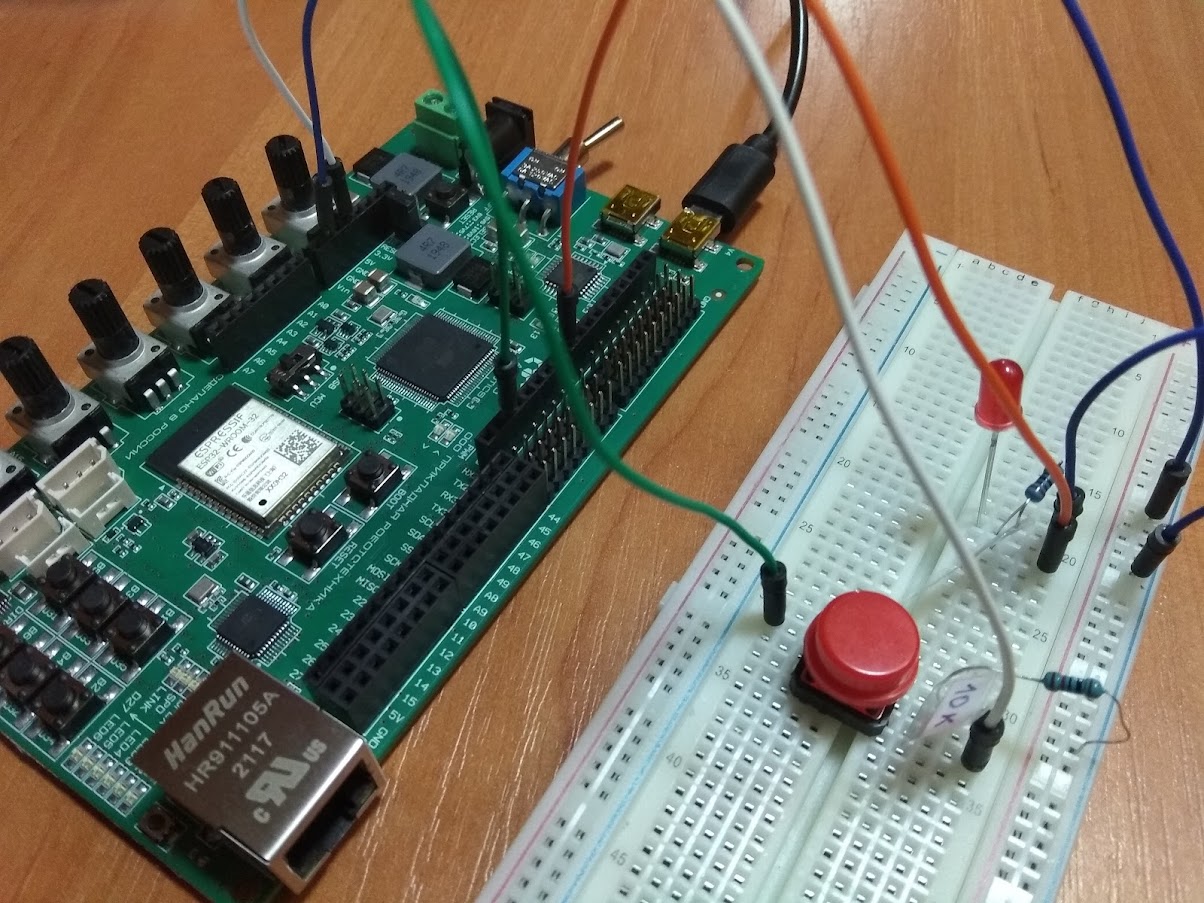
}

else {

digitalWrite(led, LOW);

}

}



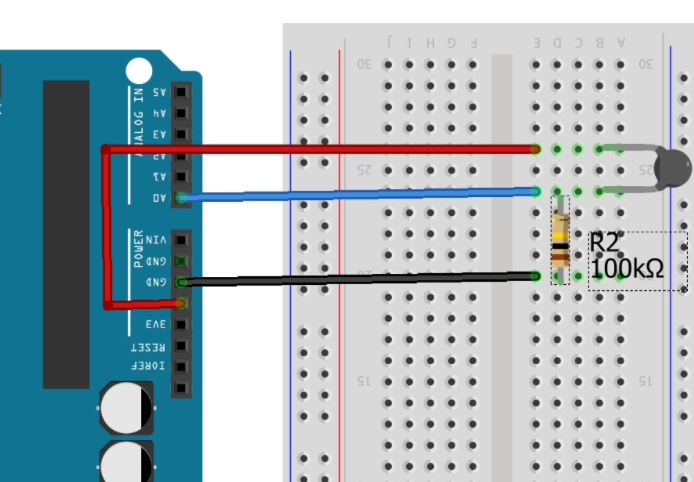
**Источник:**Arduino для начинающих. Урок 2. Подключение кнопки | Занимательная робототехника (edurobots.ru)

Лабораторная работа № 7 Термистор

Цель: Изучить работу термистора

**Терморезистор** (термистор) — полупроводниковый прибор, электрическое сопротивление которого изменяется в зависимости от его температуры.

10 КОм



**Код:**

#define B 3950 // B-коэффициент

#define SERIAL\_R 10000 // сопротивление последовательного резистора, 10 кОм

#define THERMISTOR\_R 10000 // номинальное сопротивления термистора, 100 кОм

#define NOMINAL\_T 25 // номинальная температура (при которой TR = 100 кОм)

const byte tempPin = A0;

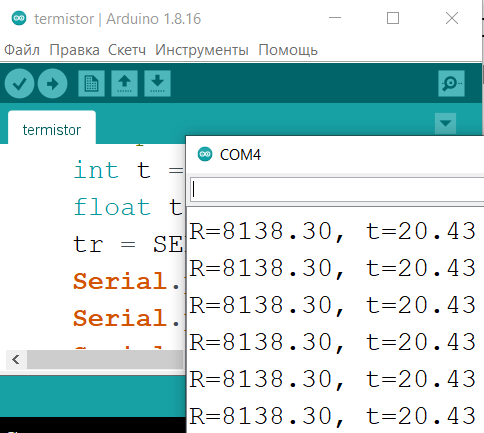
void setup() {

Serial.begin(9600);

pinMode( tempPin, INPUT );

}

Монитор порта

void loop() {

int t = analogRead( tempPin );

float tr = 1023.0 / t - 1;

tr = SERIAL\_R / tr;

Serial.print("R=");

Serial.print(tr);

Serial.print(", t=");

float steinhart;

steinhart = tr / THERMISTOR\_R;

steinhart = log(steinhart);

steinhart /= B;

steinhart -= 1.0 / (NOMINAL\_T + 273.15);

steinhart = 1.0 / steinhart;

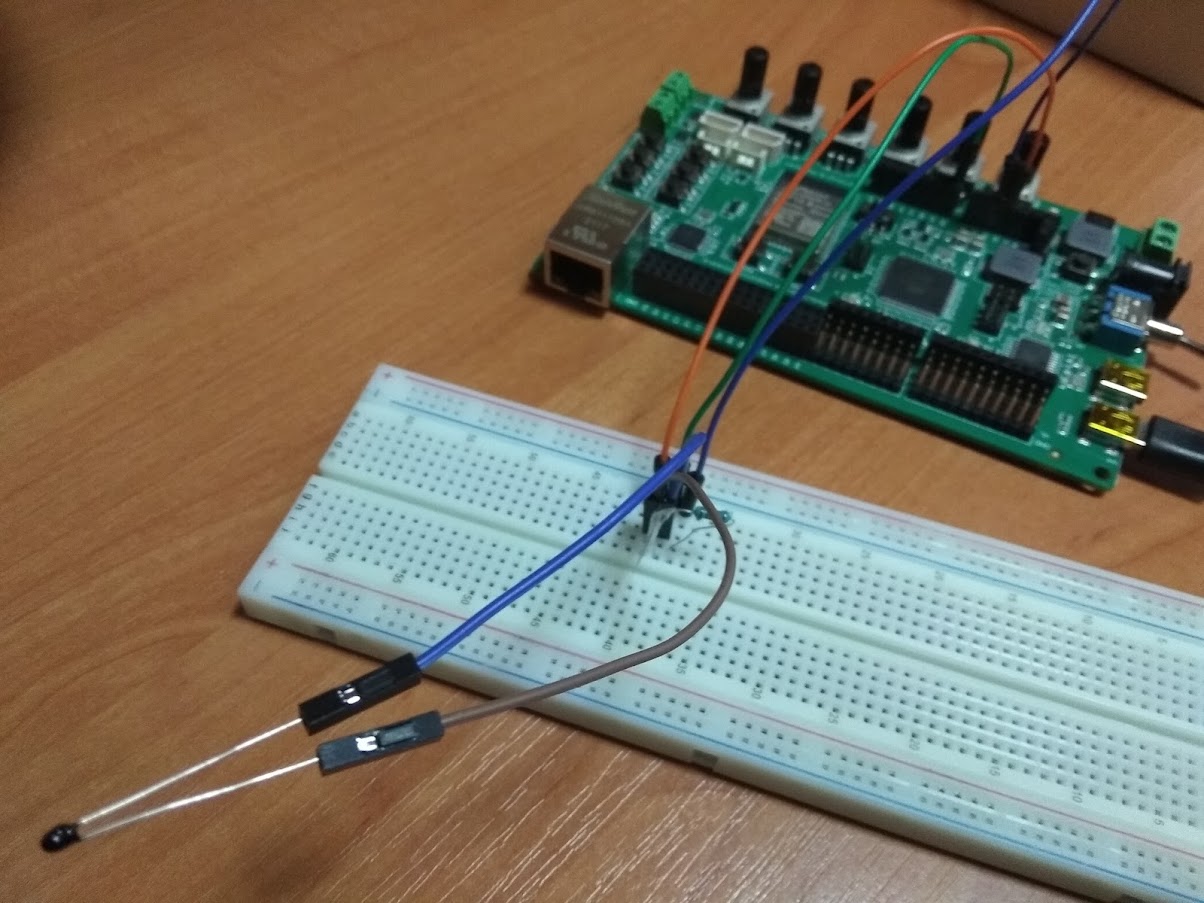
steinhart += 273.15;

Serial.println(abs(steinhart));

Подключение термистора к arduino. - У Павла! (psenyukov.ru)

delay(1000);

}

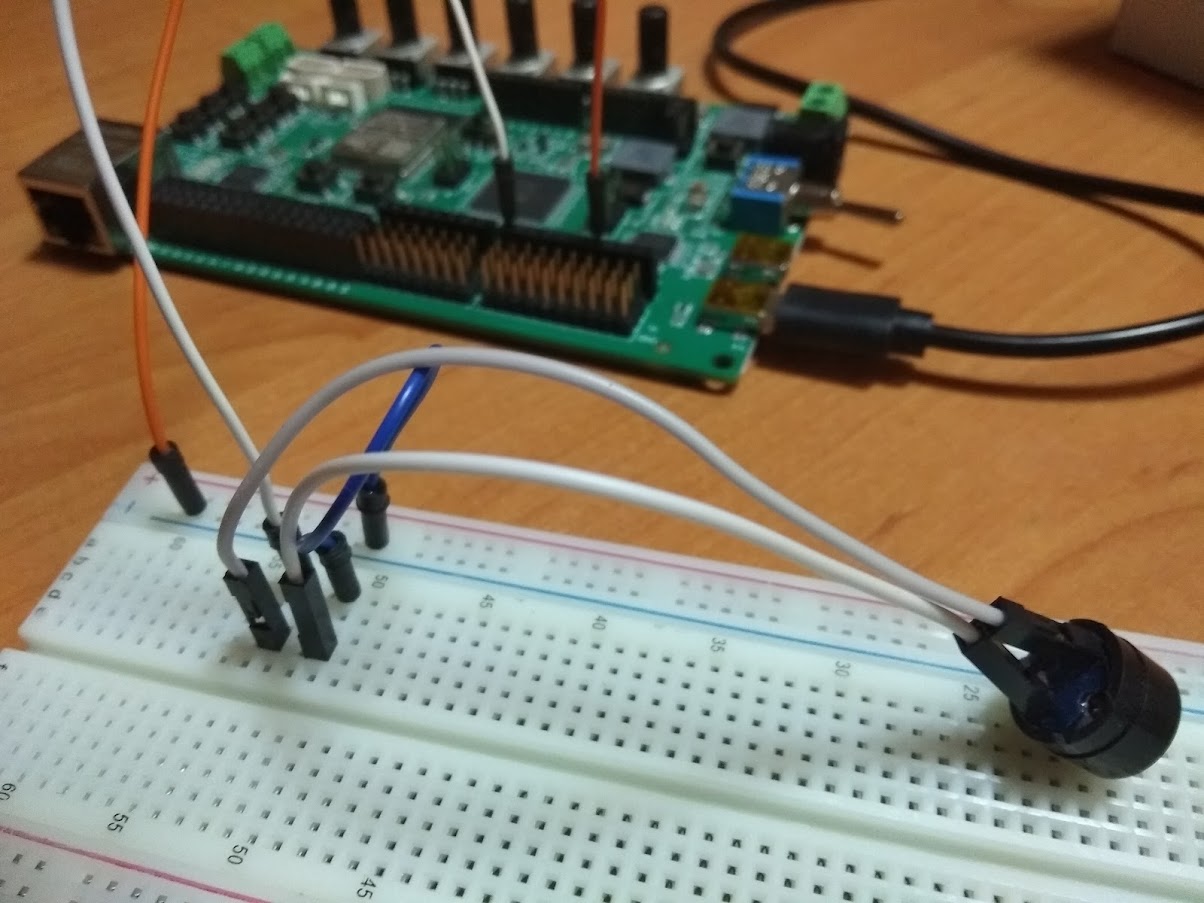


Лабораторная работа № 8 Синтезатор

Цель: получение звуков с помощью встроенных кнопок

Pin 8

GND



**Код:**

#define buz\_pin 8

#define first\_pin 30 // общее количество клавиш

#define button\_count 3

void setup()

{

pinMode(buz\_pin, OUTPUT);

}

void loop()

{

for (int i = 0; i

{

int buttonPin = i + first\_pin;

boolean buttonUp = digitalRead(buttonPin);

if (!buttonUp) {

int frequency = 400 + i \* 50;

tone(buz\_pin,frequency, 50);

delay(500);

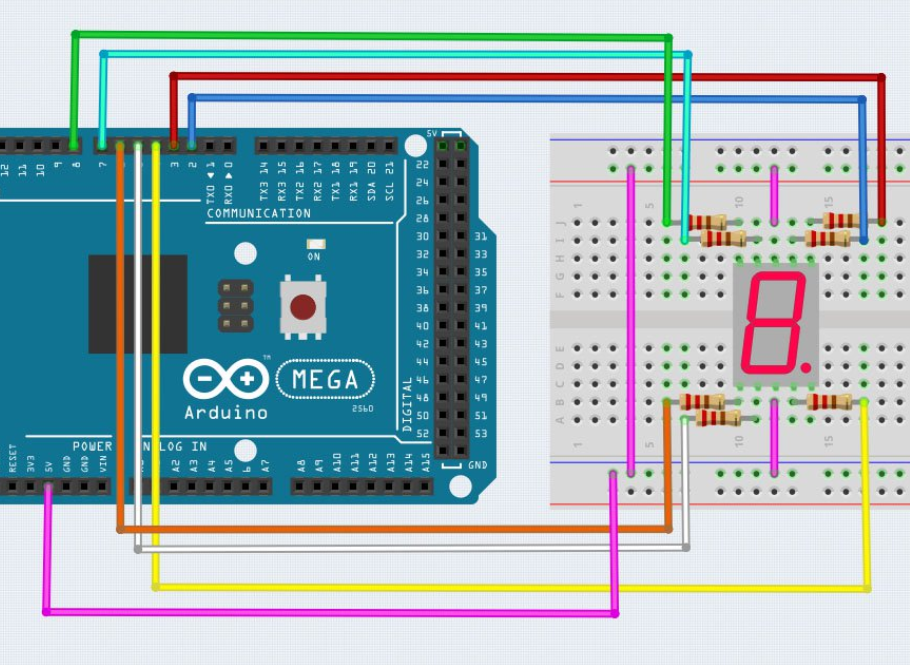
}

}

}

Лабораторная работа № 9 Взаимодействие Arduino с семисегментным индикатором

Цель: Знакомство с работой семисегментного индикатора



**Код: Обратный отсчёт**

/ создать массив для хранения конфигурации выводов индикатора для цифр

int num\_array[10][7] = { { 1,1,1,1,1,1,0 }, // 0

{ 0,1,1,0,0,0,0 }, // 1

{ 1,1,0,1,1,0,1 }, // 2

{ 1,1,1,1,0,0,1 }, // 3

{ 0,1,1,0,0,1,1 }, // 4

{ 1,0,1,1,0,1,1 }, // 5

{ 1,0,1,1,1,1,1 }, // 6

{ 1,1,1,0,0,0,0 }, // 7

{ 1,1,1,1,1,1,1 }, // 8

{ 1,1,1,1,0,1,1 }}; // 9

// объявление функции

void Num\_Write(int);

void setup()

{

// установить режимы работы выводов

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

}

void loop()

{

// цикл счетчика

for (int counter = 10; counter 0; --counter)

{

delay(1000);

Num\_Write(counter-1);

}

delay(3000);

}

// эта функция записывает значения в выводы, подключенные к индикатору

void Num\_Write(int number)

{

int pin= 2;

for (int j=0; j 7; j++)

{

digitalWrite(pin, num\_array[number][j]);

pin++;

}

}

Источник: Взаимодействие Arduino с семисегментным индикатором (radioprog.ru)

Лабораторная работа № 10 Обмен данными Arduino с ПК

Цель: Изучить механизм обмена данными между ПК и микроконтроллером

Микроконтроллер будет получать через последовательный порт некие команды и отправлять на ПК ответ, что та или иная команда принята.

**Код:**

void setup() {

  // Инициализация последовательного порта с указанием скорости обмена данными ( по умолчанию лучше использовать 9600 бод)

  Serial.begin(9600);

  // Устанавливаем таймаут (значение по умолчанию слишком велико)

  Serial.setTimeout(100);

}

void loop() {

  // Если поступили данные с ПК

  if (Serial.available() 0) {

    // Считываем полученные данные

    String command = Serial.readString();

    // Формируем ответ

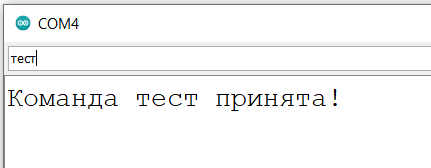
    String response = "Command " + command + " is accepted!";

    // Отправляем ответ ПК

    Serial.println(response);

}

}



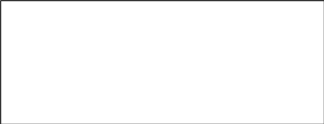
Источники:

Обмен данными между ПК и Arduino через последовательный порт (с примером на C#) | Стрелец Coder (streletzcoder.ru)

Arduino: Serial Monitor. Общаемся с компьютером (alexanderklimov.ru)

Лабораторная работа № 11 Дисплей LCD 12С интерфейс

Цель: Знакомство работой дисплея

Контакты - цвет проводов

GND – чёрный

5V - красный

SCL - оранжевый

SDA - жёлтый

Для работы понадобится библиотека LiquidCrystal\_I2C. Нужно скачать (https://iarduino.ru/file/134.html) и распаковать содержимое архива в папку Arduino\libraries\.

|  |
| --- |
|  |
|  |
|  |

**Код:**

#include // библиотека для управления устройствами по I2C

#include // подключаем библиотеку

LiquidCrystal\_I2C LCD(0x27,16,2); // присваиваем имя LCD для дисплея

void setup() {

LCD.init(); // инициализация LCD дисплея

LCD.backlight(); // включение подсветки дисплея

LCD.setCursor(3, 0); // ставим курсор на 1 символ первой строки

LCD.print("I LOVE"); // печатаем сообщение на первой строке

LCD.setCursor(2, 1); // ставим курсор на 1 символ второй строки

LCD.print("ARDUINO"); // печатаем сообщение на второй строке

Изменить работу программы так, чтобы она работала как

1)секундомер 2) калькулятор (считывает данные с клавиатуры и выводит на дисплей)

}

void loop() {

}  
Источник:

Arduino подключение LCD 1602 I2C (xn--18-6kcdusowgbt1a4b.xn--p1ai)

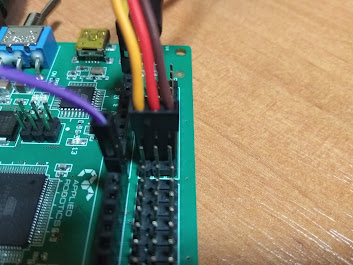
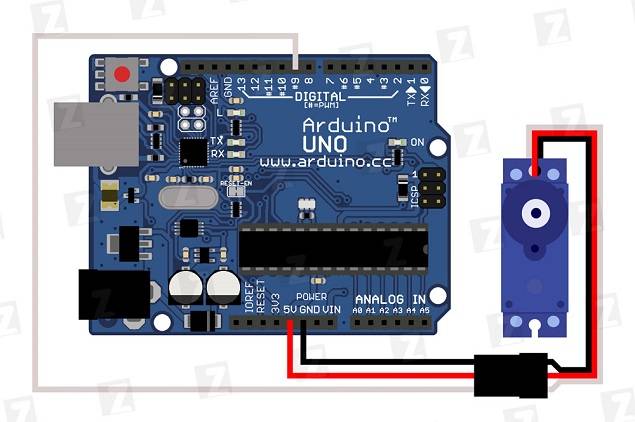
Лабораторная работа № 12 Сервопривод

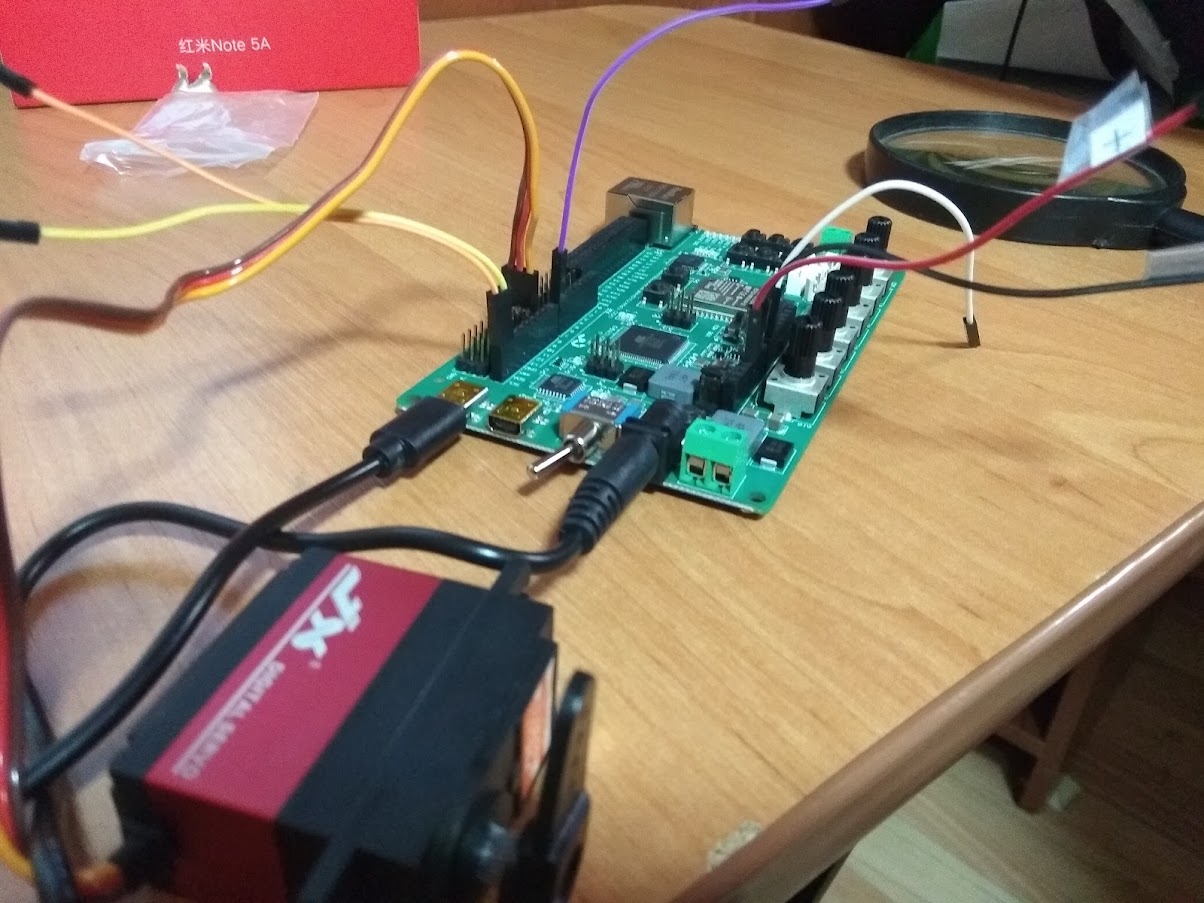
Цель: знакомство с работой сервопривода

**Сервопривод** — это мотор, положением вала которого мы можем управлять. От обычного мотора он отличается тем, что ему можно точно в градусах задать положение, в которое встанет вал. Сервоприводы используются для моделирования различных механических движений роботов.

Оборудование: сервопривод PDI-6221MG, блок питания

Подключить сервопривод к пину № 9 с помощью трехжильного шлейфа





Arduino Mega позволяет подключить до 48 сервоприводов.

**Код:**

#include //используем библиотеку для работы с сервоприводом

Servo servo; //объявляем переменную servo типа Servo

void setup() //процедура setup

{

servo.attach(9); //привязываем привод к порту 10

}

void loop() //процедура loop

{

servo.write(0); //ставим вал под 0

delay(2000); //ждем 2 секунды

servo.write(180); //ставим вал под 180

delay(2000); //ждем 2 секунды

}

Если возникает шум в работе, измените углы от 10 до 170 градусов

Источник:

1. Arduino для начинающих. Урок 4. Управление сервоприводом | Занимательная робототехника (edurobots.ru)
2. Что такое сервопривод (сервомотор) и как им управлять - Суперайс (supereyes.ru)

Лабораторная работа № 13 Шаговый двигатель

Цель: знакомство с работой шагового двигателя

**Шаговый двигатель** (stepper motor) предназначен для точного позиционирования или перемещения объекта на заданное количество шагов вала.

Оборудование: Шаговый двигатель, драйвер, провода

|  |
| --- |
|  |
|  |

**Код:**

// порты для подключения модуля ULN2003 к Arduino

#define in1 7

#define in2 8

#define in3 9

#define in4 10

int dl = 5; // время задержки между импульсами

void setup() {

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

}

void loop() {

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(dl);

digitalWrite(in1, HIGH);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

delay(dl);

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

delay(dl);

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, HIGH);

delay(dl);

}

Источник:

Arduino шаговый двигатель 28byj-48 (stepper motor) » Ардуино Уроки (xn--18-6kcdusowgbt1a4b.xn--p1ai)

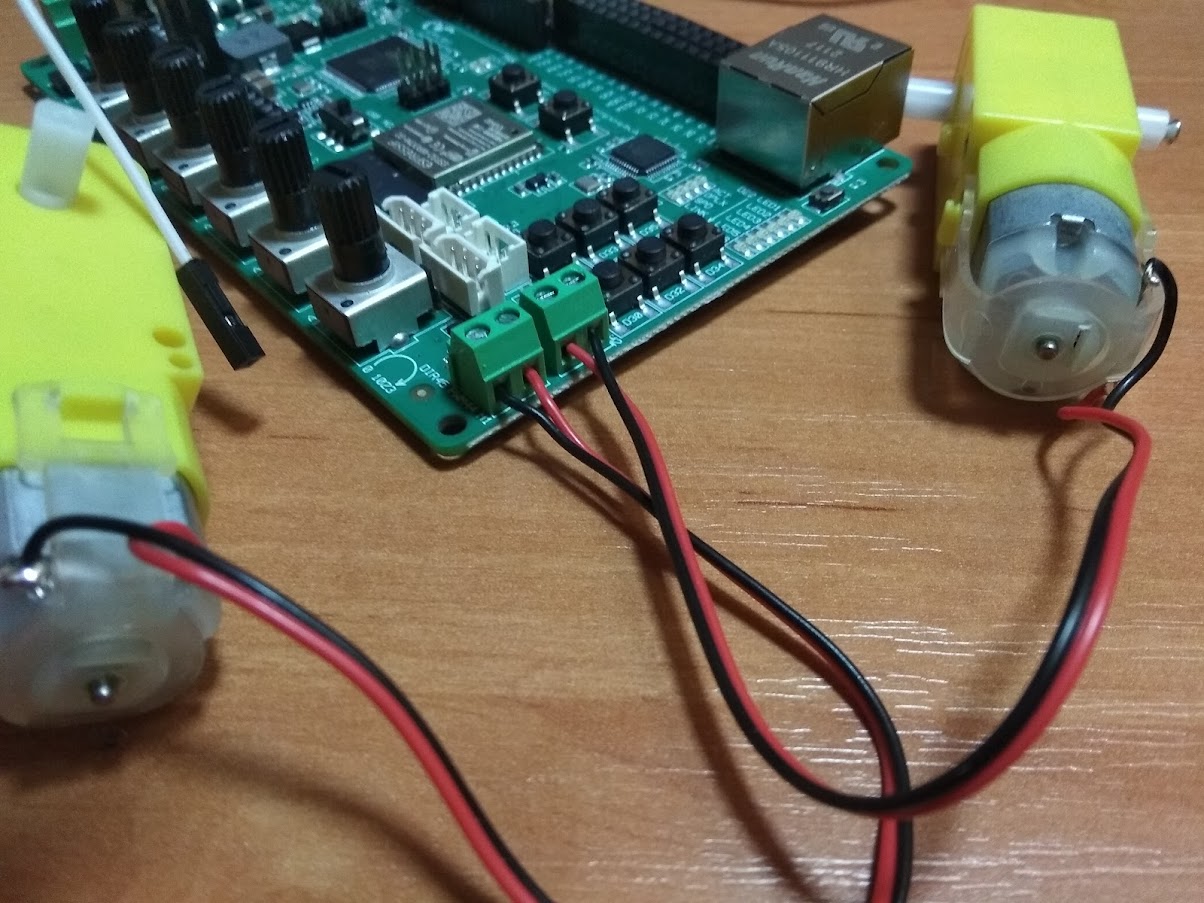
Лабораторная работа № 13 Двигатель постоянного тока

Цель: знакомство с двигателем постоянного тока

Двигатели постоянного работают при большой мощности тока, поэтому для них используется встроенный в микроконтроллер плата расширения – драйвер

Оборудование: DC-мотор (2 шт), источник питания

Прижать в гнёзда зеленого цвета с помощью маленькой отвертки провода двигателя



После запуска программы двигатели будут вращаться в одну потом в другую сторону

**Код:**

#define M1\_dir 45 //направление вращения 1 двигателя

#define M1\_Speed 44 //скорость вращения 1 двигателя

#define M2\_dir 47 //направление вращения 2 двигателя

#define M2\_Speed 46 //скорость вращения 2 двигателя

void setup() {

pinMode(M1\_dir, OUTPUT);

pinMode(M1\_Speed, OUTPUT);

pinMode(M2\_dir, OUTPUT);

pinMode(M2\_Speed, OUTPUT);

}

void loop() {

digitalWrite(M1\_dir, LOW);

analogWrite(M1\_Speed, 150);

digitalWrite(M2\_dir, LOW);

analogWrite(M2\_Speed, 150);

delay(2000);

digitalWrite(M1\_dir, HIGH);

analogWrite(M1\_Speed, 150);

digitalWrite(M2\_dir, HIGH);

analogWrite(M2\_Speed, 150);

delay(2000);

analogWrite(M1\_Speed, 0);

analogWrite(M2\_Speed, 0);

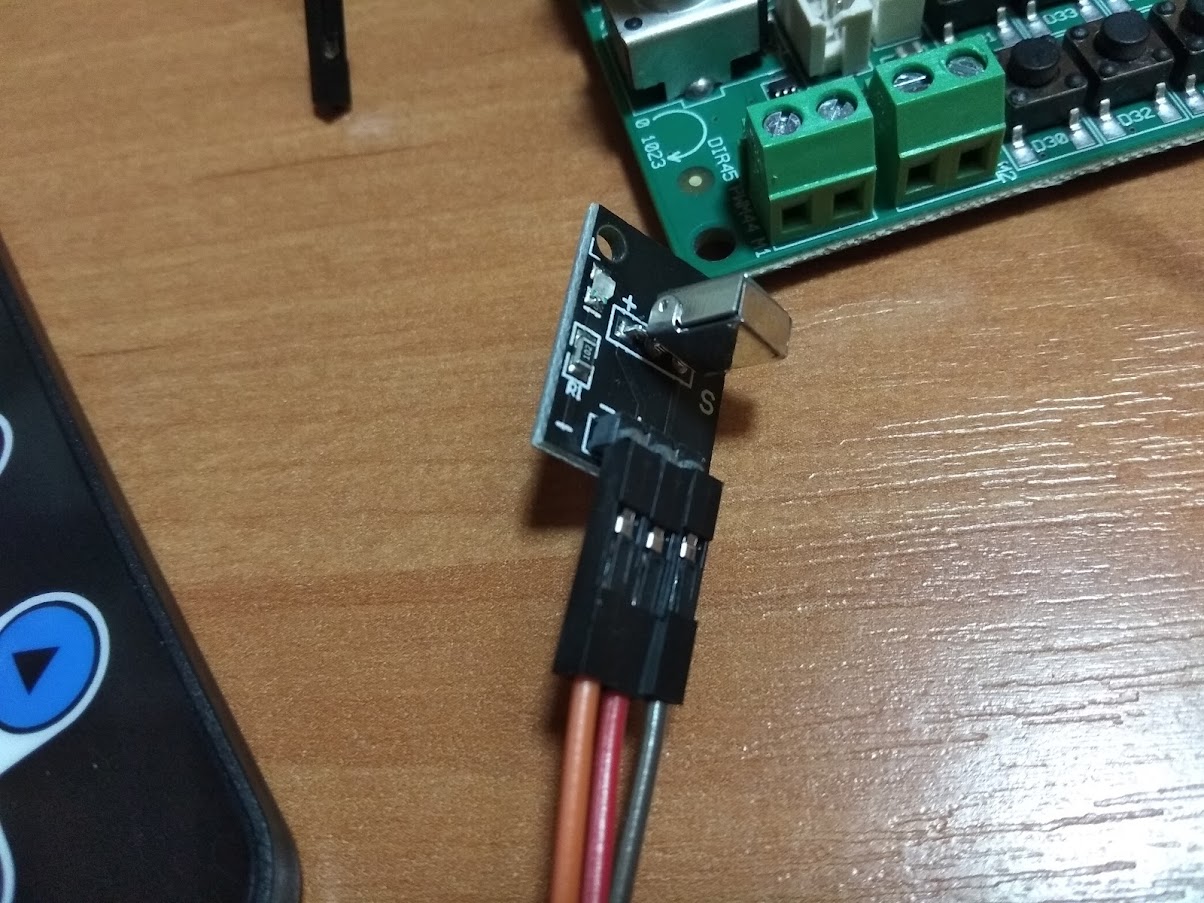
}

Лабораторная работа № 14 ИК-датчик и ИК пульт

Цель: знакомство с работой ИК-датчика и пульта

Инфракрасный пульт дистанционного управления — один из самых простых способов взаимодействия с электронными приборами. Так, практически в каждом доме есть несколько таких устройств: телевизор, музыкальный центр, видеоплеер, кондиционер. Но самое интересное применение инфракрасного пульта — дистанционное правление роботом.

Скачать и установить библиотеку **IRremote**



Красный провод - +5V

Оранжевый – GND (земля или минус)

Черный – пин 2

ИК датчик может принимать сигналы и из обычного пульта от телевизора

 С помощью программы будем принимать команды с пульта и выводить их в окно монитора.

**Код:**

#include "IRremote.h"

IRrecv irrecv(2); // указываем вывод, к которому подключен приемник

decode\_results results;

void setup() {

Serial.begin(9600); // выставляем скорость COM порта

irrecv.enableIRIn(); // запускаем прием

}

void loop() {

if ( irrecv.decode( &results )) { // если данные пришли

Serial.println( results.value, HEX ); // печатаем данные

irrecv.resume(); // принимаем следующую команду

}

}

**Управление яркостью светодиода (пин 13) с помощью ИК- пульта**

Из предыдущей программы определяем коды громкости пульта

20DF40BF и 20DFC03F

**#include "IRremote.h"**

IRrecv irrecv(2); // указываем вывод, к которому подключен приемник

decode\_results results;

void setup() {

irrecv.enableIRIn(); // запускаем прием

}

void loop() {

if ( irrecv.decode( &results )) { // если данные пришли

switch ( results.value ) {

case 0x20DF40BF:

digitalWrite( 13, HIGH );

break;

case 0x20DFC03F:

digitalWrite( 13, LOW );

break;

}

irrecv.resume(); // принимаем следующую команду

}

**Источник:**

Ардуино: инфракрасный пульт и приемник | Класс робототехники (robotclass.ru)

Лабораторная работа № 15 Bluetooth модуль

Цель: знакомство с работой Bluetooth модуля

Bluetooth модуль HC-05 используется для дистанционного управления ардуино или передачи данных с телефонов и гаджетов на микроконтроллер.

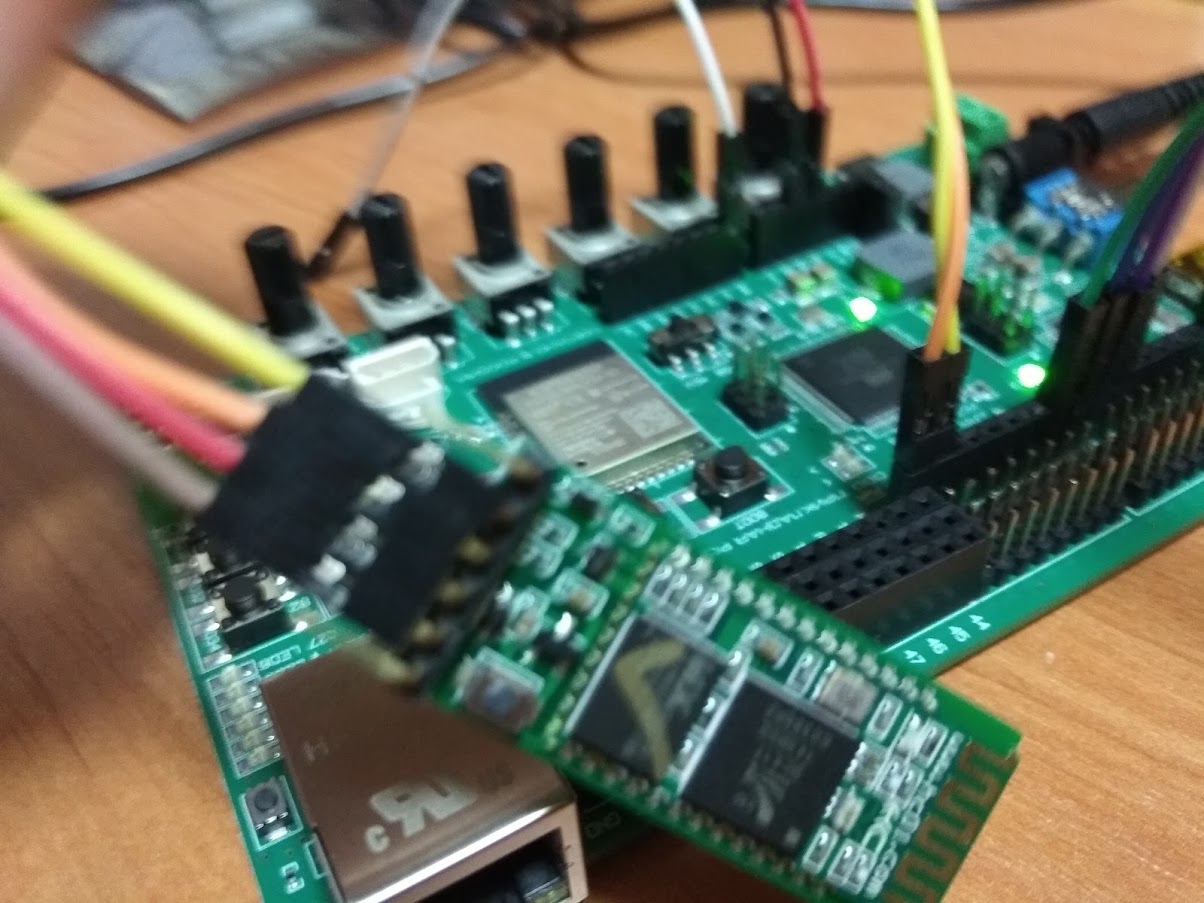
Нужно установить на телефон (Android) приложение Bluetooth терминал

Оборудование: Bluetooth модуль HC-05, источник питания, провода папа-мама 4 шт

При загрузке скэтча нужно отключить от питания Bluetooth модуль

Подключение контактов:

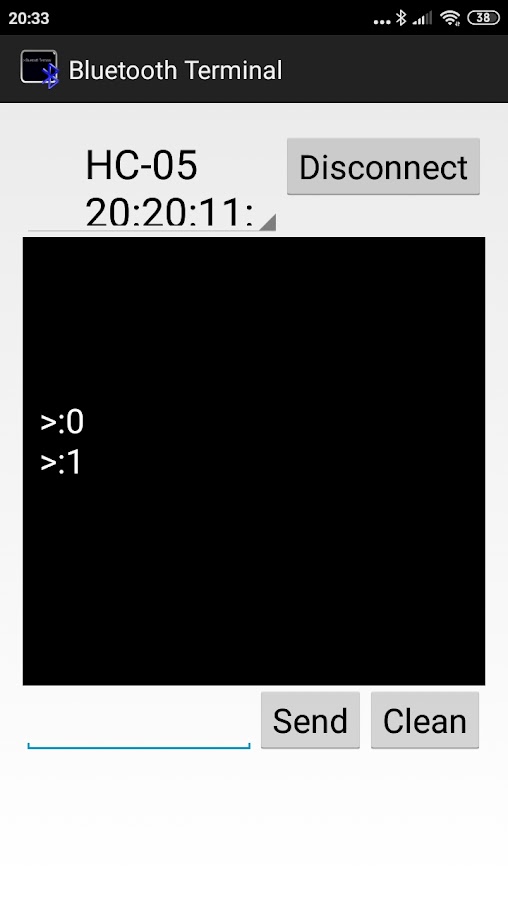
|  |  |
| --- | --- |
| **Arduino** | **Bluetooth** |
| Pin 1 (TX) | RXD |
| Pin 0 (RX) | TXD |
| GND | GND |
| 5V | VCC |



Напишем программу, которая будет с телефона управлять светодиодом (пин 13)

**Код:**

int val;

 int LED = 13;

void setup()

{

Serial.begin(9600);

pinMode(LED, OUTPUT);

digitalWrite(LED, HIGH);

}

void loop()

{

if (Serial.available())

{

val = Serial.read();

// При символе "1" включаем светодиод

if (val == '1')

{

digitalWrite(LED, HIGH);

}

// При символе "0" выключаем светодиод

if ( val == '0')

{

digitalWrite(LED, LOW);

}

}

}

**Подключение терминала:**

1. Включаем Bluetooth на телефоне и ищем новые устройства
2. Находим в списке расстройств "HC-05" и подключаемся к нему.
3. Телефон спросит пин-код. необходимо ввести "1234" или "0000"
4. Запустить терминал

Источник:

Урок 15. Bluetooth модуль HC-06 подключение к Arduino. Управление устройствами с телефона. - Описания, примеры, подключение к Arduino (iarduino.ru)

Лабораторная работа № 16 Дальномер

Цель: знакомство с работой дальномера (датчик расстояния)

Ультразвуковой дальномер рассчитан на определение расстояния до объектов в радиусе четырёх метров.

Работа модуля основана на принципе эхолокации. Модуль посылает ультразвуковой сигнал и принимает его отражение от объекта. Измерив время между отправкой и получением импульса, не сложно вычислить расстояние до препятствия.

Оборудование: Дальномер, провода папа-мама 4 шт

**Подключение контактов:**

|  |  |
| --- | --- |
| **Arduino** | **Дальномер** |
| Pin 11 | ECHO |
| Pin 10 | TRIG |
| GND | GND |
| 5V | VCC |

Нужно установить библиотеку NewPing.h

Программа выводит на монитор расстояние до объектов в см

Код:

#include

#define TRIGGER\_PIN 10

#define ECHO\_PIN 11

#define MAX\_DISTANCE 400

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE);

void setup() {

Serial.begin(9600);

}

void loop() {

delay(500);

Serial.print("Ping: ");

Serial.print(sonar.ping\_cm());

Serial.println("cm");

}



**Источник:**

Ультразвуковой дальномер HC-SR04: подключение, схема и примеры работы [Амперка / Вики] (amperka.ru)

Лабораторная работа № 17 Датчик скорости

Цель: знакомство с работой датчика скорости

Модуль датчика оборотов двигателя предназначен главным образом для определения скорости вращения вала электродвигателя.

Как правило, датчики измеряют величину благодаря регистрации определённых событий, затем количество событий соотносится с периодом времени, за которые они произошли.

Так в данном случае измеряется скорость – под событиями здесь понимаются импульсы, полученные в результате срабатывания оптического датчика во время вращения диска с прорезями. Датчик состоит из светодиода и фототранзистора, который воспринимает наличие или отсутствие излучения светодиода.

**Подключение контактов:**

|  |  |
| --- | --- |
| **Arduino** | **Датчик** |
| GND | GND |
| 2 | OUT |
| 5V | VCC |

**Код:**

int encoder\_pin = 2; // импульсные сигналы от модуля

unsigned int rpm; // количество оборотов в минуту

volatile byte pulses; // количество импульсов

unsigned long timeold;

// количество импульсов на оборот

unsigned int pulsesperturn = 12;

void counter()

{

//обновление счета импульсов

pulses++;

}

void setup()

{

Serial.begin(9600);

pinMode(encoder\_pin, INPUT);

//Прерывание 0 на цифровой линии 2

//Срабатывание триггера по спаду сигнала

attachInterrupt(0, counter, FALLING);

// Инициализация

pulses = 0;

rpm = 0;

timeold = 0;

}

void loop()

{

if (millis() - timeold = 1000) {

//Не обрабатывать прерывания во время счёта

detachInterrupt(0);

rpm = (60 \* 1000 / pulsesperturn )/ (millis() - timeold)\* pulses;

timeold = millis();

pulses = 0;

Serial.print("RPM = ");

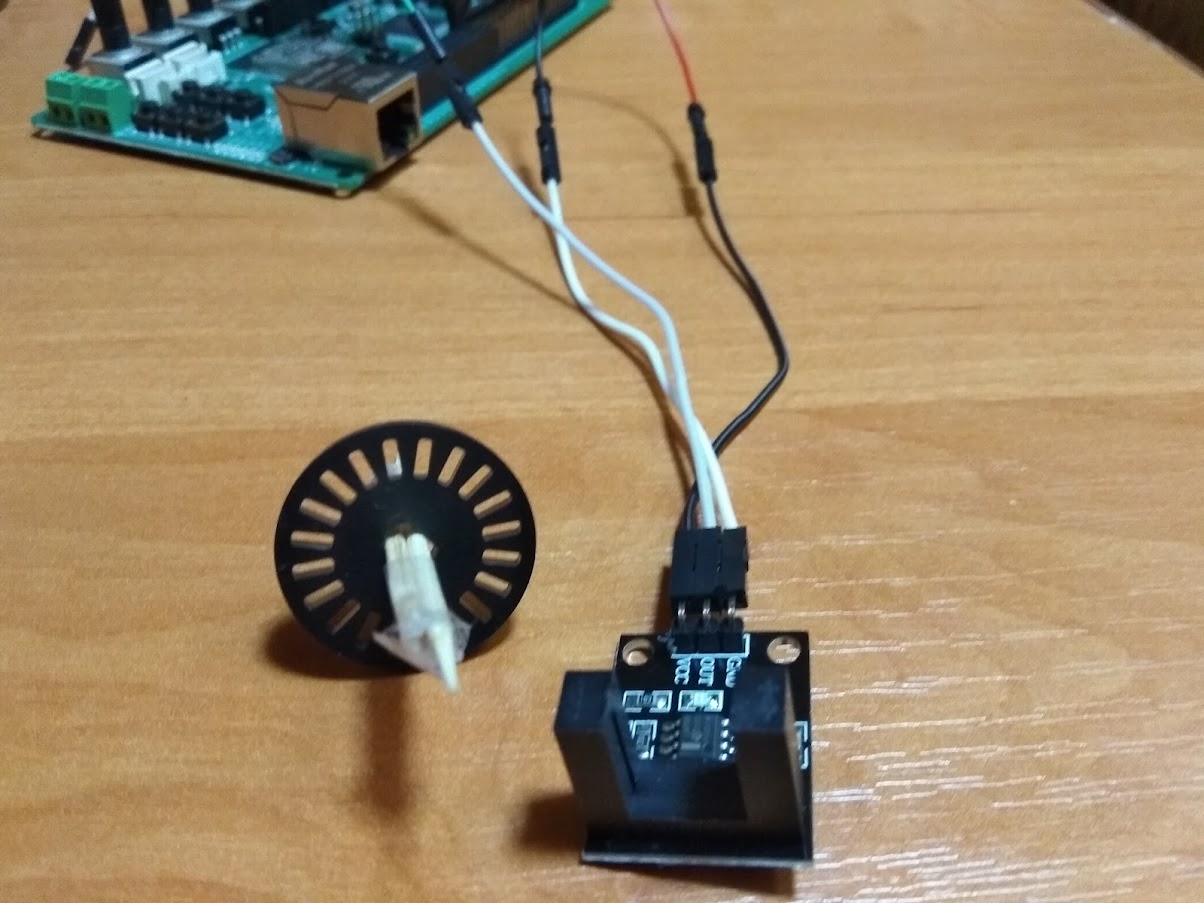
Serial.println(rpm,DEC);

//Перезагрузка процесса обработки прерываний

attachInterrupt(0, counter, FALLING);

}

}



**Источник:**

Arduino и модуль фотоимпульсного датчика скорости вращения двигателя » Digitrode.ru

Приложения

Звездные войны

|  |
| --- |
| Подключите к плате Пьезоэлемент (пьезопищалка)  Звук воспроизводится функцией **tone()**  **Синтаксис**  **tone(pin, frequency, duration)**  **Параметры**  **pin:** номер порта вход/выхода, на котором будет генерироваться сигнал  **frequency:** частота сигнала в Герцах  **duration:** длительность сигнала в миллисекундах  Чтобы остановить звук, используют функцию**noTone(pin)**  Напишем скетч, который сыграет мелодию из фильма «Звездные войны».  // Звездные войны  int led = 8;  void setup()  {  pinMode(led, OUTPUT);  }  void loop()  {  delay(2000);  tone(led,392,200);  delay(400);  tone(led,392,200);  delay(400);  tone(led,392,300);  delay(400);  tone(led,311,200);  delay(250);  tone(led,466,200);  delay(100);  tone(led,392,200);  delay(350);  tone(led,311,200);  delay(250);  tone(led,466,200);  delay(100);  tone(led,392,200);  delay(700);  tone(led,587,200);  delay(350);  tone(led,587,200);  delay(350);  tone(led,587,200);  delay(350);  tone(led,622,200);  delay(250);  tone(led,466,200);  delay(100);  tone(led,369,200);  delay(350);  tone(led,311,200);  delay(250);  tone(led,466,200);  delay(100);  tone(led,392,200);  delay(200);  noTone(led);  } |

**Применение циклов**  
byte led\_R; // случайный бит

const int Pin\_tone = 8; // номер порта зуммера

const byte COUNT\_NOTES = 39; // Количество нот

int frequences[COUNT\_NOTES] = {

392, 392, 392, 311, 466, 392, 311, 466, 392,

587, 587, 587, 622, 466, 369, 311, 466, 392,

784, 392, 392, 784, 739, 698, 659, 622, 659,

415, 554, 523, 493, 466, 440, 466,

311, 369, 311, 466, 392

};

int durations[COUNT\_NOTES] = {

350, 350, 350, 250, 100, 350, 250, 100, 700,

350, 350, 350, 250, 100, 350, 250, 100, 700,

350, 250, 100, 350, 250, 100, 100, 100, 450,

150, 350, 250, 100, 100, 100, 450,

150, 350, 250, 100, 750

};

void setup() {

pinMode(13, OUTPUT); // Настраиваем контакт на выход

pinMode(Pin\_tone, OUTPUT); // Настраиваем контакт на выход

}

void loop() {

for (int i = 0; i Цикл от 0 до количества нот

tone(Pin\_tone, frequences[i], durations[i] \* 2); // Включаем звук, определенной частоты

led\_R = random(0, 254); // Генерируем случайное число от 0 до 254

analogWrite(13, led\_R); // Зажигаем светодиод на случайно сгенерированную яркость

delay(durations[i] \* 2); // Пауза для заданной ноты

noTone(Pin\_tone); // Останавливаем звук

}

}

Источник:  
Урок 14. Музыка Star Wars на Arduino и RGB цветомузыка? Работа с tone(); - Описания, примеры, подключение к Arduino (iarduino.ru)

Виртуальный тренажёр на сайте К. Полякова

Откройте веб-страницу http://kpolyakov.spb.ru/school/robotics/arduino/arduino.htm

Удалите программу из окна слева и скопируйте в это окно новую программу:

//Звездные войны

пока 1

{

звук(15,392)

ждать(200)

звук(15,392)

ждать(200)

звук(15,392)

ждать(200)

звук(15,311)

ждать(250)

звук(15,466)

ждать(100)

звук(15,392)

ждать(350)

звук(15,311)

ждать(250)

звук(15,466)

ждать(100)

звук(15,392)

ждать(700)

звук(15,587)

ждать(350)

звук(15,587)

ждать(350)

звук(15,587)

ждать(350)

звук(15,622)

ждать(250)

звук(15,466)

ждать(100)

звук(15,369)

ждать(350)

звук(15,311)

ждать(250)

звук(15,466)

ждать(100)

звук(15,392)

ждать(200)

нетЗвука(15)

ждать(2000)

}

Щёлкнув по кнопке **запустить программу**, выполните её

Скетч «Светофор» на тренажёре К. Полякова

//1 секунду: горит только красный светодиод;

//0,5 секунды: горят красный и жёлтый светодиоды;

//1 секунду: горит только зелёный светодиод;

//0,5 секунды: горит только жёлтый светодиод.

пока 1

{

пин[0] = HIGH

ждать(3000)

пин[1] = HIGH

ждать(2000)

пин[0] = LOW

пин[1] = LOW

пин[2] = HIGH

ждать(3000)

пин[2] = LOW

пин[1] = HIGH

ждать(2000)

пин[1] = LOW

}