Эксперимент № 5. Определение цвета предмета. Фоторезистор и пьезодинамик.

Вам понадобится

1. Proteus 8.1 Professional
2. http://123D.circuits.io
3. Arduino IDE

Список деталей для эксперимента

- 1 плата Arduino Uno
- 1 беспаечная макетная плата
- 1 фоторезистор
- 4 резистора номиналом 100-220 Ом
- Пьезодинамик (Sounder(Proteus))
- RGB- светодиод
- провода «папа-папа»

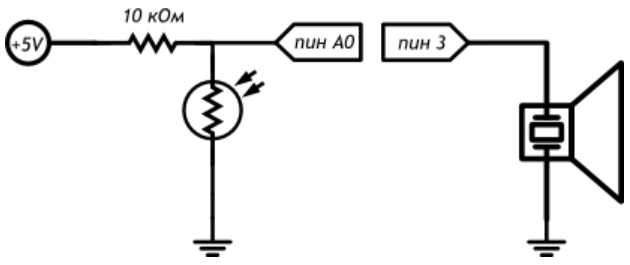
Повторение

1) Понятие фоторезистор, резистор
2) Пьезодинамик

В этом эксперименте мы имитируем действие музыкального инструмента терменвокс: изменяем высоту звучания бесконтактным путем, больше или меньше закрывая от света фоторезистор. Оригинальный инструмент был изобретён ещё в 1920 году, Львом Сергеевичем Терменом, человеком с непростой и насыщенной судьбой. А сейчас мы имеем возможность воспроизвести изобретение с помощью нехитрой электроники.

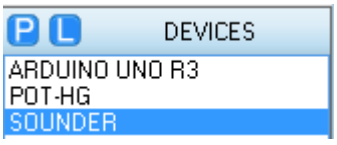
Пошаговое руководство для проведения эксперимента

Принципиальная схема

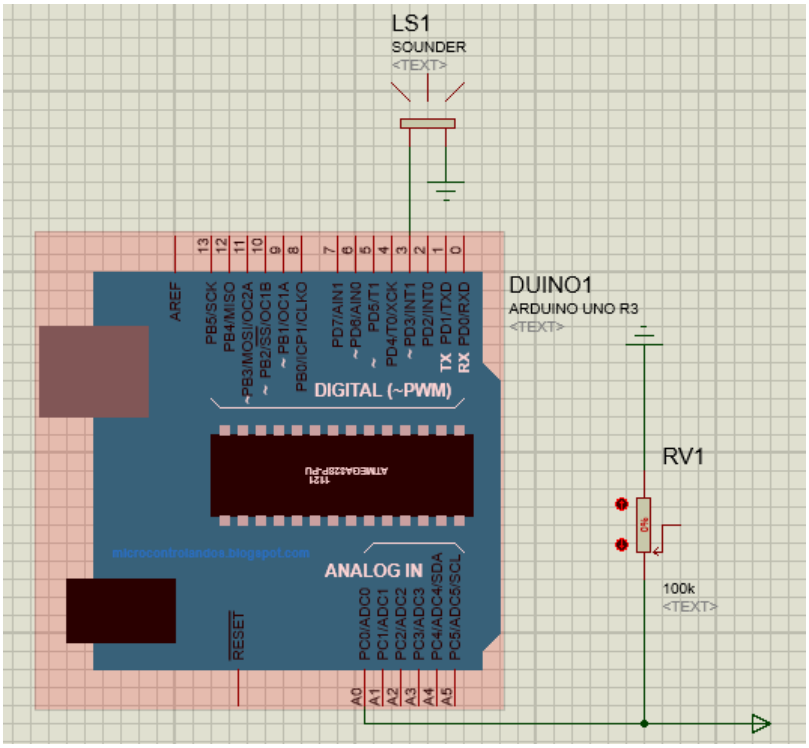


Разработка эксперимента в среде Proteus

Создайте новый эксперимент в среде Proteus. Подтвердите создание нового проекта нажатием кнопки **Finish.** Сохраните проект, дайте имя файлу – **PiezoColor.** В списке устройств добавьте по ключевому слову следующие устройства: **Arduino UNO R3**, **POT-HG** (потенциометр), **SOUNDER** (пьезодинамик) **.** В списке устройств Devices должны отобразиться все выбранные вами детали для эксперимента.



Соедените устройства как показано в примере ниже.



Обратите внимание. Значение сопротивления потенциометра можно менять щелкнув дважды левой кнопкой мыши на значении сопротивления возле графического отображения переменного резистора. 100k <TEXT>

Приложение. Код программы «Sounder» для эмуляция звука в Proteus.

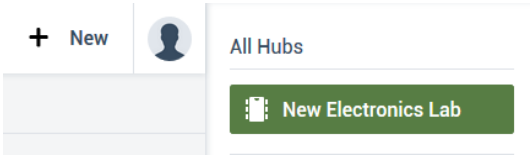
```
  // даём имена для пинов с пьезопищалкой (англ. buzzer) и фото-
  // резистором (англ. Light Dependent Resistor или просто LDR)
  #define BUZZER_PIN  3
  #define LDR_PIN     A0
  void setup()
  {
    // пин с пьезопищалкой — выход...
    pinMode(BUZZER_PIN, OUTPUT);
  }
  void loop()
  {
    int val;
    // считываем уровень освещённости так же, как для
    // потенциометра: в виде значения от 0 до 1023.
val = analogRead(LDR_PIN);
// "заставляем" динамик пищать со значением взятым с потенциометра + частота (от 1 Гц - 1023)
```

```
    tone(BUZZER_PIN, val+500);
    delay(100);
    tone(BUZZER_PIN, val+1000);
    delay(100);
  }
```

Для эмуляции с Arduino в Proteus используйте только **HEX-файлы**! Щелкните на рисунке платы дважды для открытия диалогового окна настройки микроконтроллера. В окне нажмите на кнопке со значком  и выберите **HEX** файл из папки с проектом. Нажмите кнопку **ОК**. Внизу экрана, в левой нижней части нажмите на кнопке **Run the simulation**  .

Эксперимент с **Proteus** завершен!

Приступим к эксперименту с использованием **HTPP://123D.CIRCUIT.IO**.Пройдите авторизацию используя учетную запись **Facebook.** После успешного прохождения авторизации, нажмите в верхней правой части сайта кнопку **NEW** и выберите **New Electronics Lab** для создания нового эксперимента.



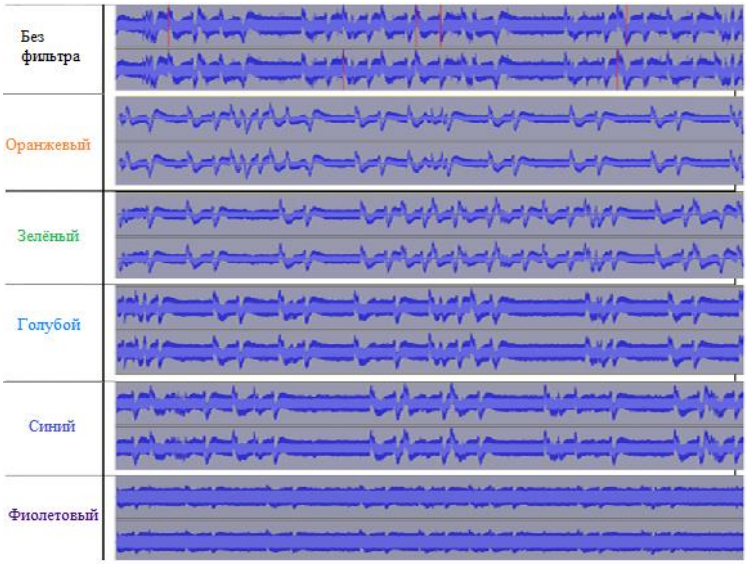
Проведем эксперимент по влиянию цвета на звук. Как услышать свет?

Соберем экспериментальную установку, состоящую из трёх частей:

1. Оптический терменвокс.
2. Штатив с лампой и набор световых фильтров разных цветов.
3. Микрофон и программа Audacity для записи звуковой волны.

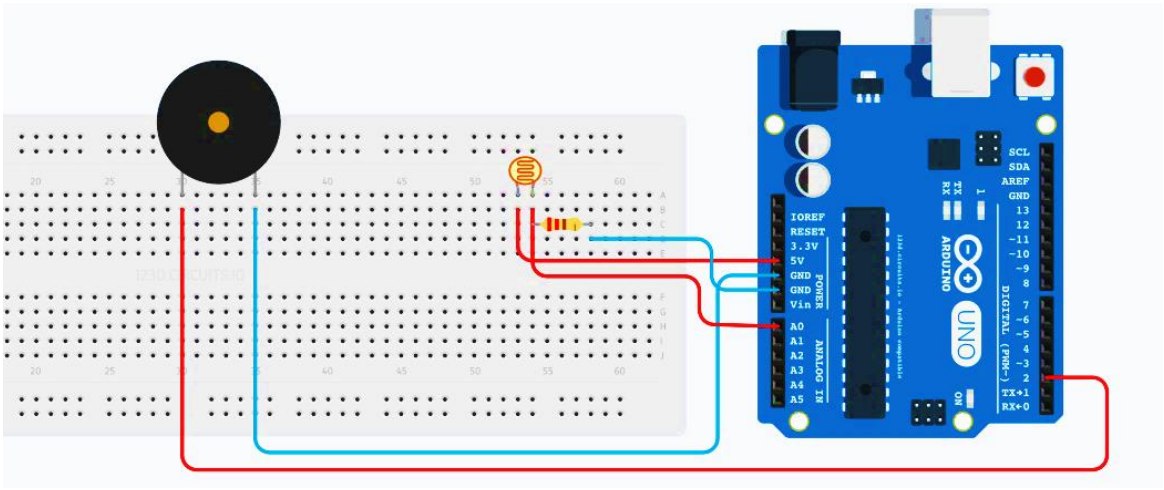
Мы разработали собственный метод исследования. Свет, пропущенный сквозь цветной фильтр, направляется на фоторезистор. Данные с фоторезистора отправляются на контроллер Arduino, где они обрабатываются и подаются на пьезопищалку. При смене излучаемого цвета (RGB-светодиод) происходит заметное на слух изменение тона звучания.

В ходе нескольких экспериментов было установлено, что спектр света при разложении излучаемого зеленого света светодиода имеет более высокое значение фотона. Мы не ограничимся собственными слуховыми ощущениями, и выразим их в физических единицах измерения, применив аудиоредактор Audacity (распространяется на условиях GNU General Public License). Частота колебаний волн измеряется в герцах. Показывает сколько раз в секунду происходит колебание. Длина волны — величина, обратная частоте, определяющая промежуток между колебаниями. Между частотой и длиной волны существует взаимосвязь: частота равна отношению скорости к длине волны. Каждый звук имеет свою частоту. Каждый цвет определяется своей длиной волны и имеет частоту, равную скорости света, делённую на длину волны. Проанализировали связь между численными характеристиками длин звуковых и цветовых волн и диапазонами их частот. Опытным путём установили связь между светом и звуком.



Обратите внимание. Полярность фоторезистора, как и обычного резистора, не играет роли. Его можно устанавливать любой стороной. В данном упражнении мы собираем простой вариант схемы включения пьезодинамика. Полярность пьезопищалки роли не играет: вы можете подключать любую из ее ножек к земле, любую к порту микроконтроллера. Вспомните как устроен делитель напряжения: фоторезистор помещается в позицию R2 — между аналоговым входом и землей. Так мы получаем резистивный фотосенсор.

Задание. Разместите на макетной плате компоненты как показано на рисунке ниже. Для соединения деталей между собой воспользуйтесь нажатиями левой кнопкой мыши. Нажмите на кнопке **Code Editor**. Скопируйте код проекта «Излучение» в Code Editor.



```
#define BUZZER_PIN  3
int photoPin = 7; // фоторезистор подключен 0-му аналоговому входу
int val = 0; // переменная для хранения значения входного напряжения

void setup()
{
pinMode(8, OUTPUT);
pinMode(12, OUTPUT);
pinMode(13, OUTPUT);
Serial.begin(9600);
  pinMode(BUZZER_PIN, OUTPUT);

  // ...а все остальные пины являются входами изначально,
  // всякий раз при подаче питания или сбросе микроконтроллера.
  // Поэтому, на самом деле, нам совершенно необязательно
  // настраивать LDR_PIN в режим входа: он и так им является
}

void loop()
{

int val, frequency;
val = analogRead(photoPin); // считываем значение с фоторезистора
Serial.println(val);
val = val/4; // конвертируем из 0-1023 к 0-255
   // рассчитываем частоту звучания пищалки в герцах (ноту),
  // используя функцию проекции (англ. map). Она отображает
  // значение из одного диапазона на другой, строя пропорцию.
  // В нашем случае [0; 1023] -> [3500; 4500]. Так мы получим
  // частоту от 3,5 до 4,5 кГц.

frequency = map(val, 0, 1023, 1500, 2500);
  // заставляем пин с пищалкой «вибрировать», т.е. звучать
  // (англ. tone) на заданной частоте 50 миллисекунд, при
  // условии, что цифровое значение при разложении зеленого света больше 50
  // зависит от количества света, попадающего на фоторезистор
// в интервале от 0-255
if (val>50) {
    tone(BUZZER_PIN, frequency, 50); }
  }
```

Нажмите на кнопке **Run Simulation**. Продолжите эксперименты с «Head» проверив себя и выполнив **задания для самостоятельного решения**.

Пояснения к коду

- Функция map(value, fromLow, fromHigh, toLow, toHigh) возвращает целочисленное значение из интервала [toLow, toHigh], которое является пропорциональным отображением содержимого value из интервала [fromLow, fromHigh]
- Верхние границы map не обязательно должны быть больше нижних и могут быть отрицательными. К примеру, значение из интервала [1, 10] можно отобразить в интервал [10,-5]
- Если при вычислении значения map образуется дробное значение, оно будет отброшено, а не округлено
- Функция marне будет отбрасывать значения за пределами указанных диапазонов, а также масштабирует их по заданному правилу.
- Если вам нужно ограничить множество допустимых значений, используйте функцию constrain(value, from, to), которая вернет:

value, если это значение попадает в диапазон [from, to]

from, если value меньше него

to, если value больше него

- Функция tone(pin, frequency, duration) заставляет пьезопищалку, подключенную к порту pin, издавать звук высотой frequency герц на протяжении duration миллисекунд
- Параметр duration не является обязательным. Если его не передать, звук включится навсегда. Чтобы его выключить, вам понадобится функция noTone(pin). Ей нужно передать номер порта с пищалкой, которую нужно выключить
- Одновременно можно управлять только одной пищалкой. Если во время звучания вызвать tone для другого порта, ничего не произойдет.
- Вызов tone для уже звучащего порта обновит частоту и длительность звучания

Вопросы для проверки себя

- Каким сопротивлением должен обладать фоторезистор, чтобы на аналоговый вход было подано напряжение 1 В?
- Можем ли мы регулировать яркость светодиода, подключенного к 11-му порту, во время звучания пьезопищалки?
- Каков будет результат вызова map(30,0,90,90,-90)?
- Как будет работать вызов tone без указания длительности звучания?
- Можно ли устроить полифоническое звучание с помощью функции tone?

Задания для самостоятельного решения

- Уберите из программы чтение датчика освещенности и пропищите азбукой Морзе позывной SOS: три точки, три тире, три точки
- Измените код программы так, чтобы с падением освещенности звук становился ниже (например, падал от 5 кГц до 2,5 кГц)
- Измените код программы так, чтобы звук терменвокса раздавался не непрерывно, а 10 раз в секунду с различимыми паузами