Министерство образования Хабаровского края

Краевое государственное бюджетное образовательное учреждение

среднего профессионального образования

«Хабаровский колледж межотраслевых технологий и сферы обслуживания»

Учебно-методическое пособие для студентов

«Arduino: изучение контроллера в режимах дистанционного и аудиторного обучения»

по специальности 09.02.01 «Компьютерные системы и комплексы».

Разработано преподавателем

комиссии информационных дисциплин Мазур Т.В.

Хабаровск, 2020

Содержание

Пояснительная записка

Методическое пособие «Arduino: изучение контроллера в режимах дистанционного и аудиторного обучения»

предназначено для студентов специальности 09.02.01 «Компьютерные системы и комплексы» и может применяться при изучении МДК02.01 Микропроцессорные системы, раздел «Микроконтроллеры». Пособие включает в себя 10 практических работ, восемь из которых можно выполнить с использованием онлайн эмулятора Tinkercad. Методическое пособие применимо при изучении предмета в дистанционном режиме и в аудиторном.

Рассмотренный в пособии контроллер Arduino является популярным средством автоматизации, применяется при изготовлении устройств студентами в рамках дипломного проектирования, поэтому изучение его структуры, принципов присоединения устройств и программирования очень важно и актуально. Практические работы включают в себя название, цель, теоретические сведения, указания по сбору схемы и программированию контроллера, примеры программ, в которые нужно добавить недостающий код, требование к отчёту и др. Аудиторное выполнение задания предполагает сбор схемы, её программирование и тестирование. Дистанционное выполнение – это создание схемы в эмуляторе, программирование, эмуляция работы. Предполагается, что студент запишет видео экрана, оформит отчёт о проделанной работе и вышлет результат в таком виде преподавателю.

Лабораторная работа 1. Контроллер Arduino. Знакомство с эмулятором TinkercadArduino

Цель: познакомиться с назначением, порядком работы с контроллерами Arduino. Рассмотреть на примере ArduinoUno R3 структуру микроконтроллера, схему, описание, подключение. Зарегистрироваться и изучить принципы работы в эмуляторе TinkercadArduino.

Отчёт: не требуется.

Задание

Задание 1. Познакомьтесь с назначением и порядком работы с контроллерами Arduino.

https://arduinomaster.ru/#1508255432022-58d0f83f-1be

Задание 2. Рассмотрите структуру микроконтроллера на примере платы ArduinoUno R3: схему, описание, подключение устройств.

https://arduinomaster.ru/platy-arduino/plata-arduino-uno/

Задание 3. Так как в процессе дистанционного обучения у студентов не будет доступа к оборудованию, то все практические работы надо будет выполнять в эмуляторе. Один из лучших бесплатных эмуляторов TinkercadCircuitsArduino.

Для знакомства с TinkercadCircuitsArduino откройте ссылку на статью

https://arduinomaster.ru/program/simulyator-arduino-tinkercad-circuits/

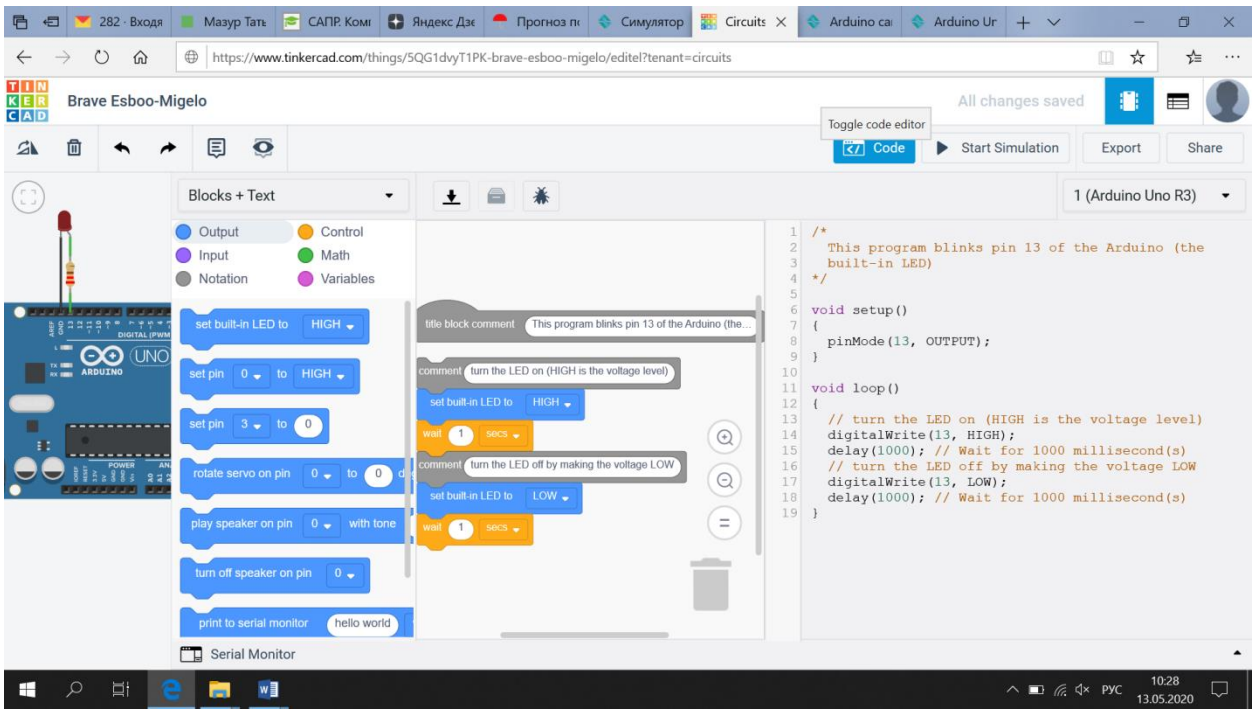
Получите аккаунт в системе. Регистрация в Tinkercad абсолютно бесплатная.

Создайте проект.

Познакомьтесь с интерфейсом программы, используя инструкции статьи. Вы должны научиться собирать схему: добавлять элемент, присоединять, редактировать, удалять, просматривать информацию.

Одной из интересных возможностей для новичков является третья закладка – Starters. Создатели сервиса подготовили несколько готовых схем, которые мы можем сразу же подгрузить в проект и редактировать на свое усмотрение.

Загрузите одну из схем, познакомьтесь со способами создания программы (скетча) в эмуляторе: блоками, текстом.



Запустите симулятор. Посмотрите, как будет работать схема.

Поэкспериментируйте с другими схемами.

Лабораторная работа 2. Программирование микроконтроллера АрдуиноUNO. Мигающий светодиод.

Цель: Получить навыки программирования микроконтроллера, организовывать мигание светодиода с заданными интервалами.

Методическое и информационное обеспечение

1. Партыка Т.Л., Попов И.И. Периферийные устройства вычислительной техники: учеб.пособие для сред. проф. образования М.: Форум, 2014 г.
2. Авдеев В.А. Периферийные устройства: интерфейсы, схемотехника, программирование М.: ДМК Пресс, 2015г.
3. Методические указания по выполнению лабораторной работы.

Оборудование: ПК, стартовый комплект устройств Ардуино, программный пакет ARDUINO IDE.

Формируемые компетенции:ПК1.1., ПК 2.2.

Задачи:

1. Собрать схему подключения светодиода к макетной плате по заданным условиям;
2. Подключить микроконтроллер к ПК;
3. Написать скетч мигания светодиода согласно варианту.

Краткие теоретические сведения

Arduino (ардуино) — популярная аппаратная вычислительная платформа, основными компонентами которой являются плата ввода-вывода и среда разработки. Arduino может использоваться как для создания автономных интерактивных объектов, роботов, так и подключаться к программному обеспечению, выполняемому на компьютере. Платы имеют аналоговые и цифровые порты, к которым можно подключить практически любое простое устройство: кнопка, датчик, мотор, экран.

Ардуино IDE. Аббревиатура IDE расшифровывается как IntegratedDevelopmentEnvironment, в переводе – интегрированная среда разработки. С помощью этой программы программисты пишут программы, причем делают это гораздо быстрее и удобнее, чем при использовании обычных текстовых редакторов.В рамках платформы Ардуино программа Arduino IDE делает то же – помогает программистам писать программы. С ее помощью скетч, написанный на языкеArduino, проверяется, превращается в C++, компилируется, загружается в Arduino. Как известно, первая программа, которую человек пишет при изучении программирования называется «HelloWorld!». Суть этой программы сводится к тому, чтобы после запуска на экране появилась указанная простая фраза.

В мире микроэлектроники, симметричной задачей смело можно назвать мигание светодиодом. Другими словами, нам необходимо управлять включением и выключением светодиода при помощи микроконтроллера. В нашем случае, таким контроллером станет Ардуино.

Устройство светодиода

Светодиод — это устройство, которое представляет собой полупроводниковый прибор, способный излучать свет при пропускании через него электрического тока в прямом направлении (от анода к катоду). Ниже приведена схема типичного светодиода с линзой (см. Рис.1). Существуют 2-хцветные и 3-х цветные (см. Рис.2) светодиоды. У 3-х цветных светодиодов сразу четыре ноги. Одна — катод, а три другие — аноды для трех разных цветов. Для того чтобы правильно включить светодиод в электрическую цепь, необходимо отличать катод от анода. Сделать это можно по двум признакам:

1) Анод светодиода имеет более длинный проводник.
2) Со стороны катода, корпус светодиода немного срезан.

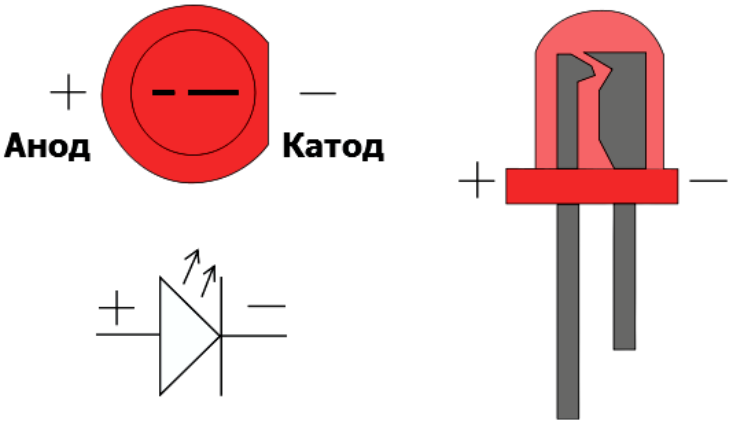


Рисунок 1.Схема светодиода

Рисунок 2.Трехцветный (RGB) светодиод

В современной микроэлектронике применяются миниатюрные светодиоды для поверхностного монтажа. Такие индикаторы, например, имеются на ArduinoUNO для информирования пользователя о состоянии системы. На рисунке 3 он светит зеленым цветом.

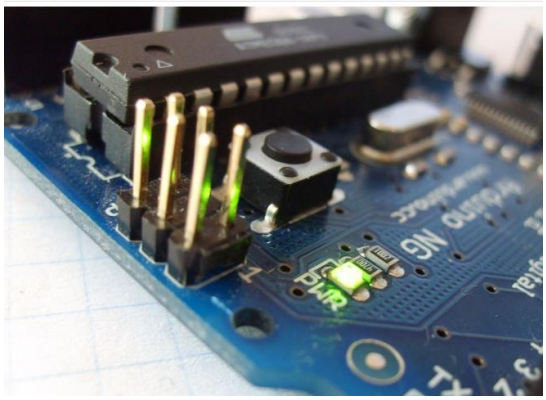


Рисунок 3.Светодиод для поверхностного монтажа

Работа выполняется на образовательном конструкторе «Мастер ARDUINO.

Ход работы:

Задание 1. Собрать схему подключения светодиода к макетной плате по заданным условиям;

Инструкция по выполнению:

1) Для подключения мигающего светодиода на Arduino (смотреть теоретический материал) найдитев образовательном конструкторе следующие компоненты:

1. Плата ArduinoUNO(Рис.4)

Рисунок 4. Плата ArduinoUNO

2. Макетную плату (breadboard)(Рис.5).

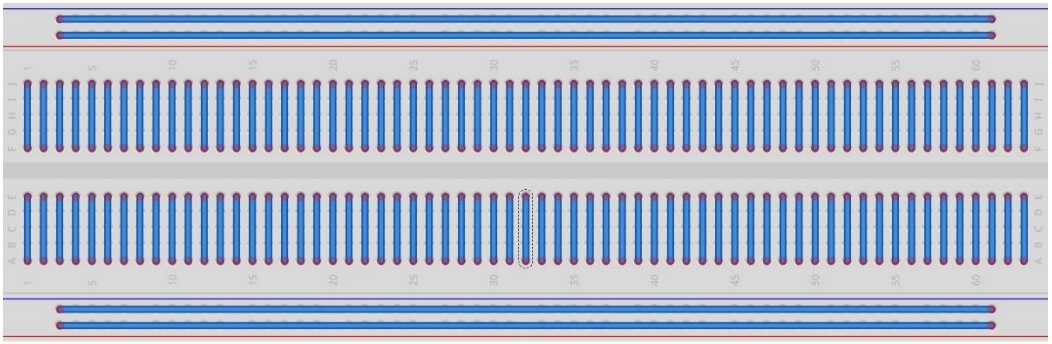


Рисунок 5. Макетная плата

3. 2 провода типа «папа-папа»(Рис.6)

Рисунок 6. Провода типа «Папа-папа»

4. Светодиод(Рис.7)

Рисунок 7. Светодиод

5. Резистор соответствующего номинала. (Правила расчёта номинала резистора смотреть в Примечании) Таблица расчёта приведена на Рис.8 Цветовая кодировка резисторов.

   Примечание: Резисторы, в особенности малой мощности — мелкие детали, резистор мощностью 0,125 Вт имеет длину несколько миллиметров и диаметр порядка миллиметра. Прочитать на такой детали номинал с десятичной запятой трудно, поэтому при указании номинала вместо десятичной точки пишут букву, соответствующую единицам измерения (К — для килоомов; М — для мегаомов; E, R или без указания единиц — для единиц Ом). Кроме того, любой номинал отображается максимум тремя символами. Например, 4K7 обозначает резистор сопротивлением 4,7 кОм, 1R0 — 1 Ом, M12 — 120 кОм (0,12 МОм) и т. д. Однако в таком виде наносить номиналы на маленькие резисторы сложно, и для них применяют маркировку цветными полосами.

   Для резисторов с точностью 20 % используют маркировку с тремя полосками, для резисторов с точностью 10 % и 5 % — маркировку с четырьмя полосками, для более точных резисторов — с пятью или шестью полосками. Первые две полоски всегда означают первые два знака номинала. Если полосок 3 или 4, третья полоска означает десятичный множитель, то есть степень десятки, которая умножается на число, состоящее из двух цифр, указанное первыми двумя полосками. Если полосок 4, последняя указывает точность резистора. Если полосок 5, третья означает третий знак сопротивления, четвёртая — десятичный множитель, пятая — точность. Шестая полоска, если она есть, указывает температурный коэффициент сопротивления (ТКС). Если эта полоска в 1,5 раза шире остальных, то она указывает надёжность резистора (% отказов на 1000 часов работы) (см. Рис.6).

   Следует отметить, что иногда встречаются резисторы с 5 полосами, но стандартной (5 или 10 %) точностью. В этом случае первые две полосы задают первые знаки номинала, третья — множитель, четвёртая — точность, а пятая — температурный коэффициент.

Цвет	как число	как десятичный множитель	как точность в %	как ТКС в ppm/°C	как % отказов
серебристый	—	$1 \cdot 10^{-2}$ = «0,01»	10	—	—
золотой	—	$1 \cdot 10^{-1}$ = «0,1»	5	—	—
чёрный	0	$1 \cdot 10^{0}$ = 1	—	—	—
коричневый	1	$1 \cdot 10^{1}$ = «10»	1	100	1 %
красный	2	$1 \cdot 10^{2}$ = «100»	2	50	0,1 %
оранжевый	3	$1 \cdot 10^{3}$ = «1000»	—	15	0,01 %
жёлтый	4	$1 \cdot 10^{4}$ = «10 000»	—	25	0,001 %
зелёный	5	$1 \cdot 10^{5}$ = «100 000»	0,5	—	—
синий	6	$1 \cdot 10^{6}$ = «1 000 000»	0,25	10	—
фиолетовый	7	$1 \cdot 10^{7}$ = «10 000 000»	0,1	5	—
серый	8	$1 \cdot 10^{8}$ = «100 000 000»	0,05	—	—
белый	9	$1 \cdot 10^{9}$ = «1 000 000 000»	—	1	—
отсутствует	—	—	20 %	—	—

Рисунок 8. Цветовая кодировка резисторов

Пример:

Допустим, на резисторе имеются четыре полосы: коричневая, чёрная, красная и золотая. Первые две полоски дают 1 0, третья 100, четвёртая даёт точность 5 %, итого — резистор сопротивлением 10·100 Ом = 1 кОм, с точностью ±5 %.

Запомнить цветную кодировку резисторов нетрудно: после чёрной 0 и коричневой 1 идёт последовательность цветов радуги. Так как маркировка была придумана в англоязычных странах, голубой и синий цвета не различаются.

Также для облегчения запоминания можно воспользоваться мнемоническим правилом: «Часто Каждый Красный Охотник Желает Знать, Сколько Фазанов Село в Болоте».

Для облегчения различные разработчики программного обеспечения создают программы, которые определяют сопротивление резистора.

Поскольку резистор — симметричная деталь, может возникнуть вопрос: «Начиная с какой стороны читать полоски?» Для четырёхполосной маркировки обычных резисторов с точностью 5 и 10 % вопрос решается просто: золотая или серебряная полоска всегда стоит в конце. Для трёхполосочного кода первая полоска стоит ближе к краю резистора, чем последняя.

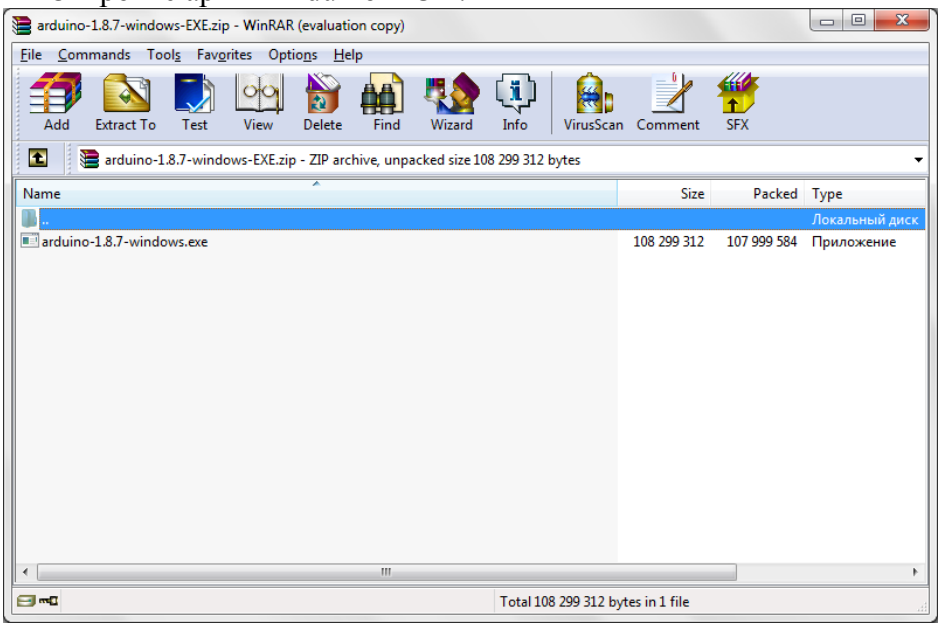
Задание 2. Подключить микроконтроллер к ПК;

Инструкция по выполнению:

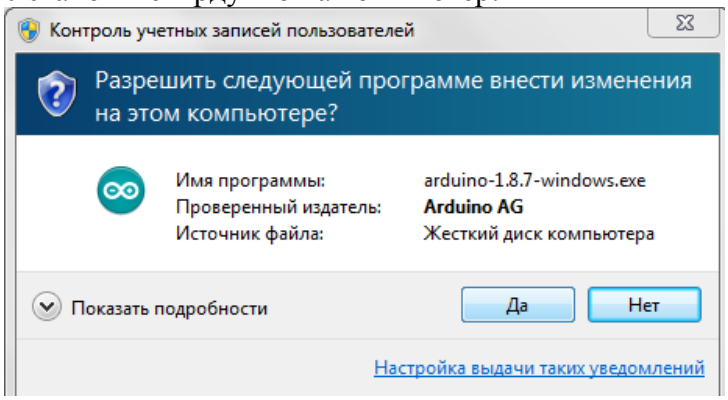
1) Установите среду разработкиArduino IDE. Установка среды производится обучающимися строго по приведённому алгоритму.

- Зайдите в папку Студент-Ардуино.

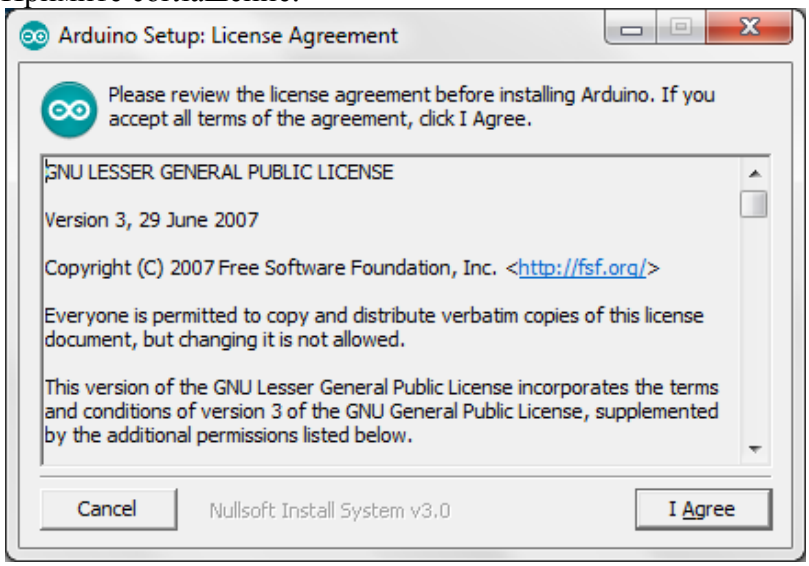
- Откройте архив Arduino-1-8-7:



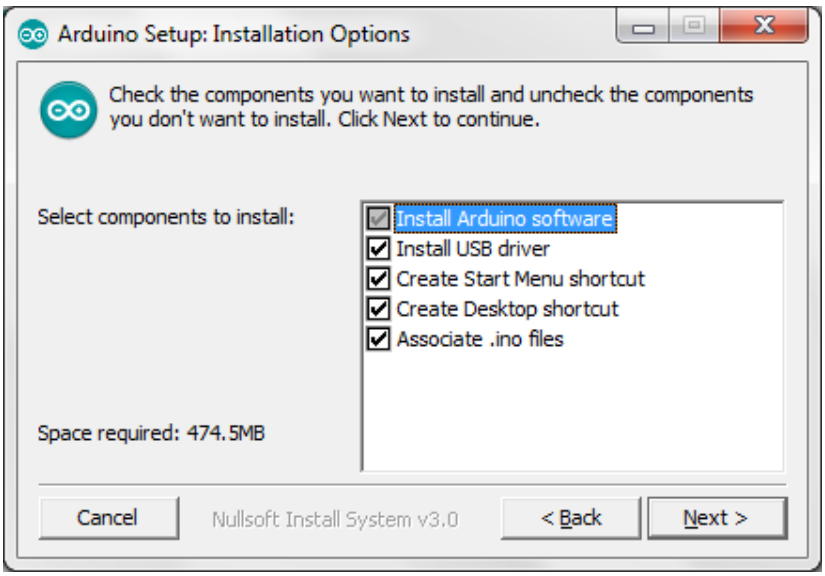
- Установите Ардуино на компьютер:



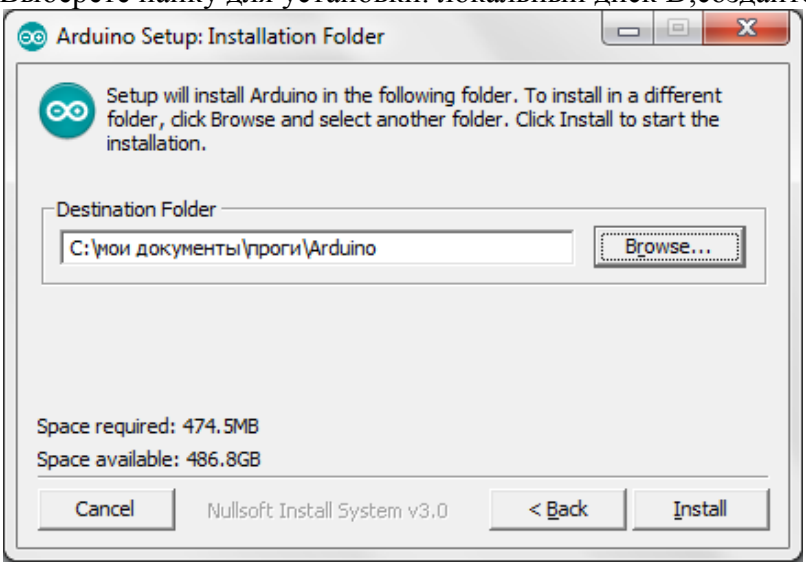
- Примите соглашение:



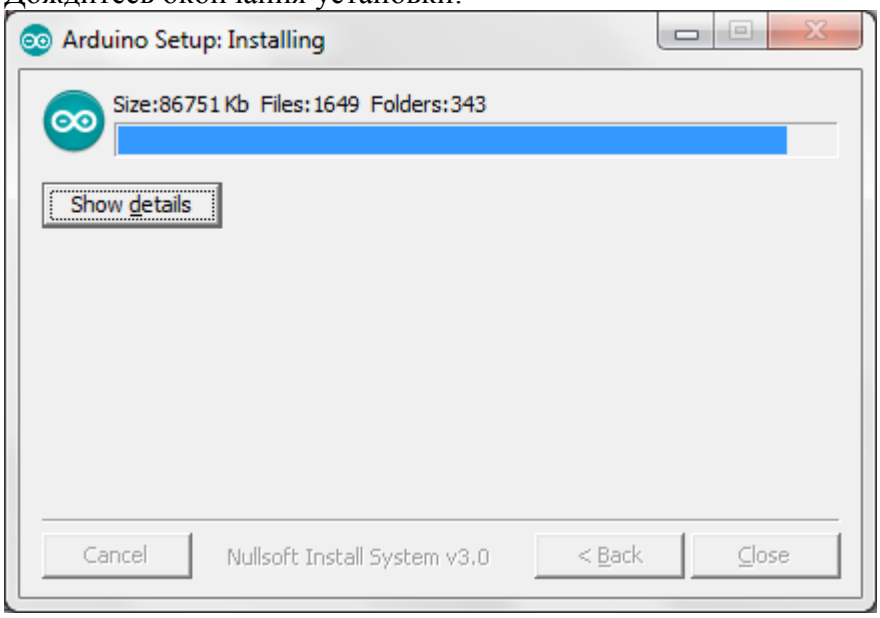
- Установите все компоненты ПО:



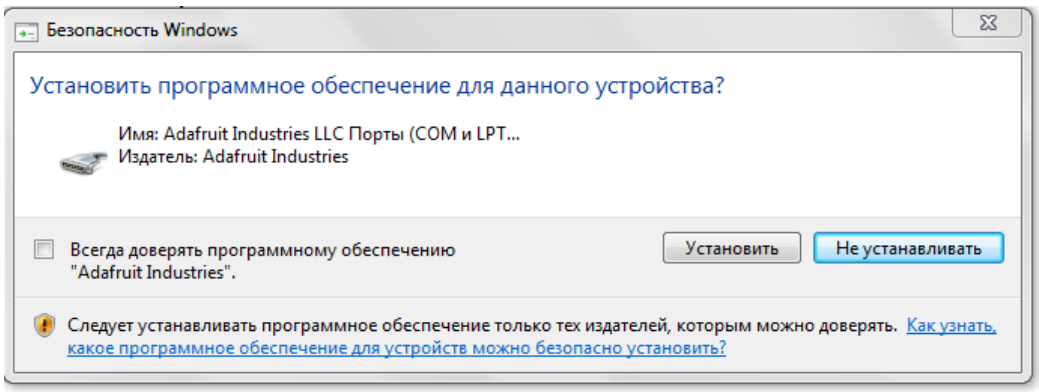
- Выберете папку для установки: локальный диск D,создайте папку Ардуино.



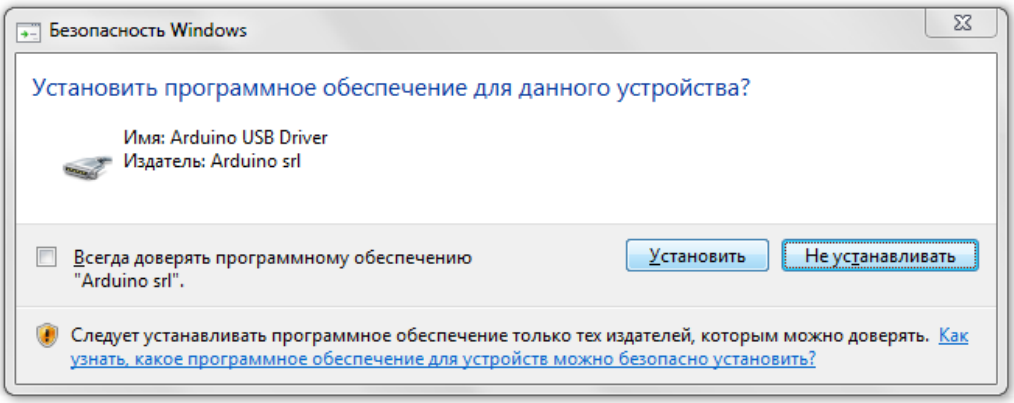
- Дождитесь окончания установки:



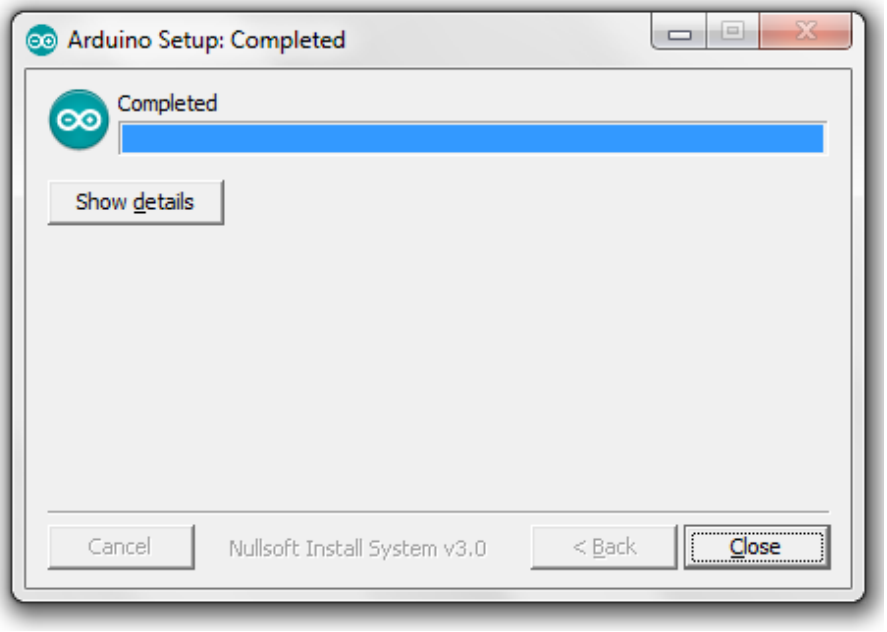
- Установите LLC порты:



- Установите USB драйвер

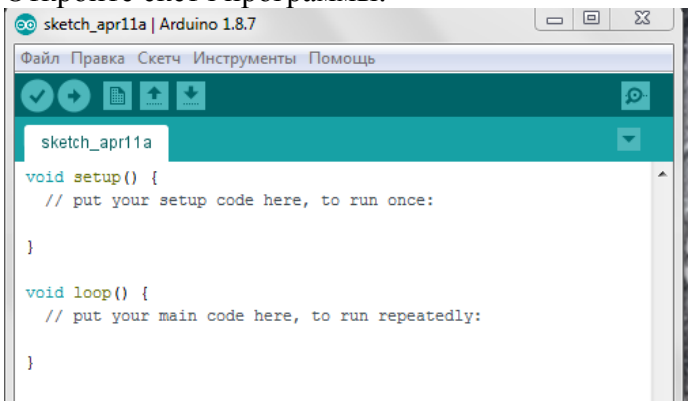


- Завершите установку ПО:



- Откройте программу Ардуино на рабочем столе:

- Откройте скетч программы:



1) Подключить схему к компьютеру. Подключение выполнять строго в соответствии с алгоритмом:

- Чтобы зажечь светодиод, нам потребуется подключить его к одному из цифровых выводов Ардуино.

- Катод светодиода подключим к минусу (земля), анод же соединим с выводом №8 через токозадающий резистор 150 Ом.

- Собрать схему. Пример подключения схемы приведен на Рис.9:

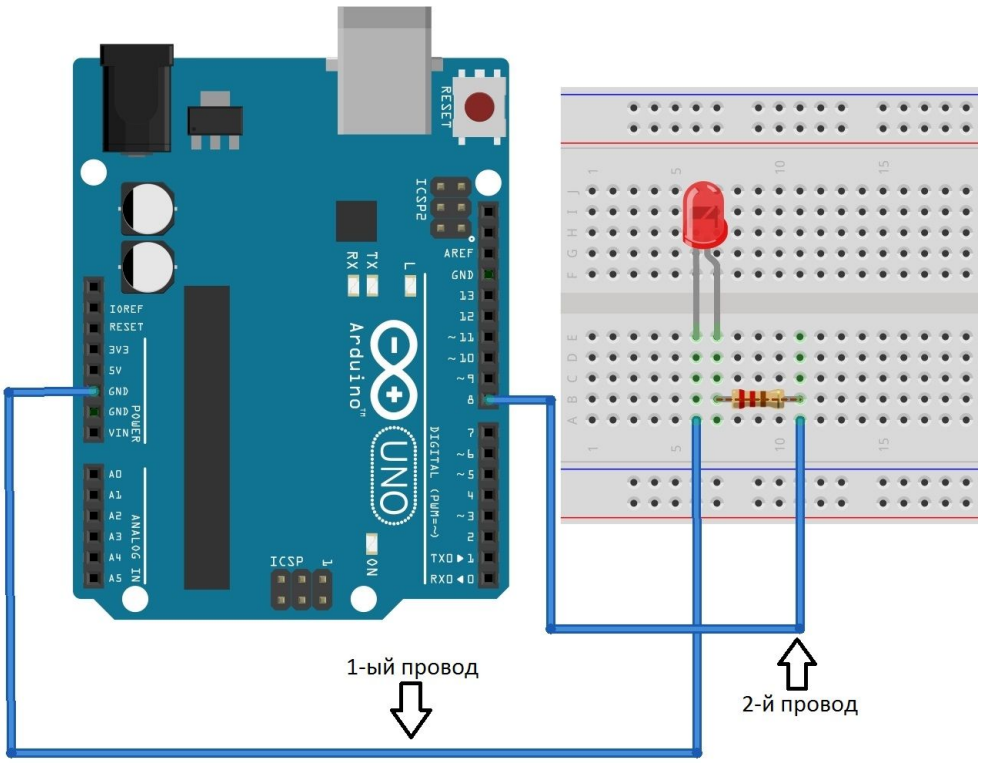


Рисунок 9. Схема подключения

Задание 3. Написать скетч мигания светодиода согласно варианту.

Инструкция по выполнению:

1) Загрузка программы:

- Чтобы подключить плату к компьютеру необходимо взять USBпровод. Соедините ArduinoUno с компьютером через USB-кабель. Вы увидите, как на плате загорится светодиод «ON», и начнёт мигать светодиод «L». (См.рис 10.) Это означает, что на плату подано питание, и микроконтроллер ArduinoUno начал выполнять прошитую на заводе программу «Blink» (мигание светодиодом).

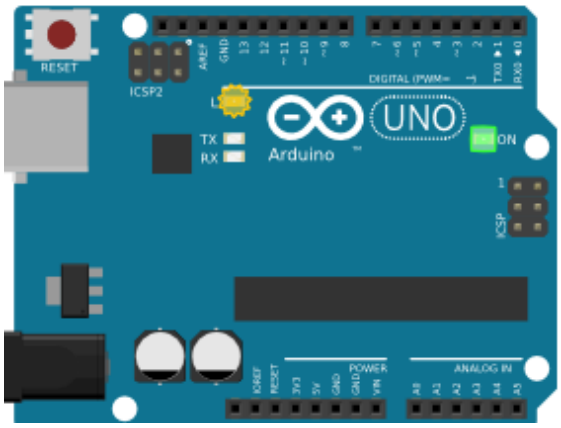


Рисунок 10.Питание подано

- Чтобы настроить Arduino IDE на работу с ArduinoUno, нам необходимо узнать, какой номер COM-порта присвоил компьютер ArduinoUno. Для этого нужно зайти в «Диспетчер устройств» Windows и раскрыть вкладку «Порты (COM и LPT)». На экране появится окно Диспетчера устройств (Рис.11):

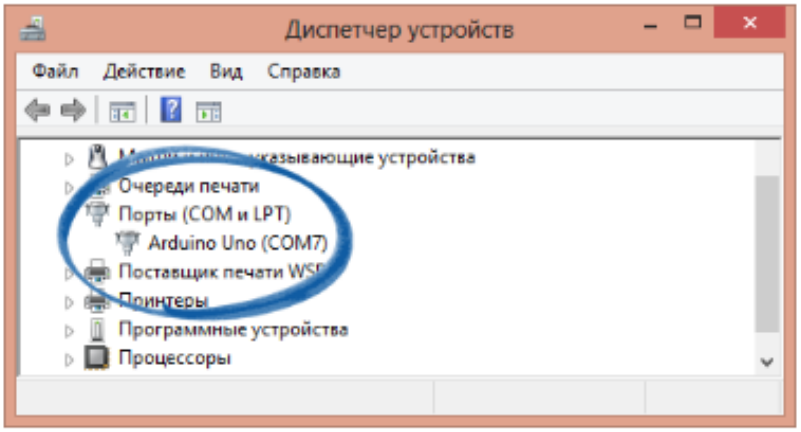


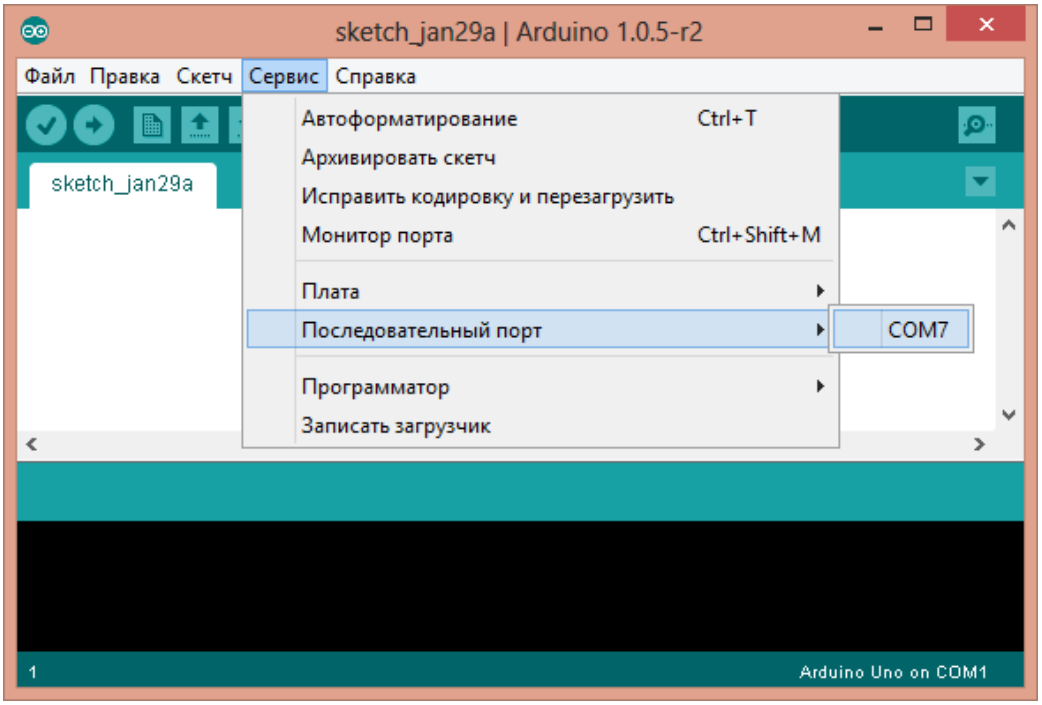
Рисунок 11.Диспетчер устройств

- Это означает, что операционная система распознала нашу плату ArduinoUno как COM-порт, подобрала для неё правильный драйвер и назначила этому COM-порту номер 7. Если мы подключим к компьютеру другую плату Arduino, то операционная система назначит ей другой номер. Поэтому, если у вас несколько плат Arduino, очень важно не запутаться в номерах COM-портов.

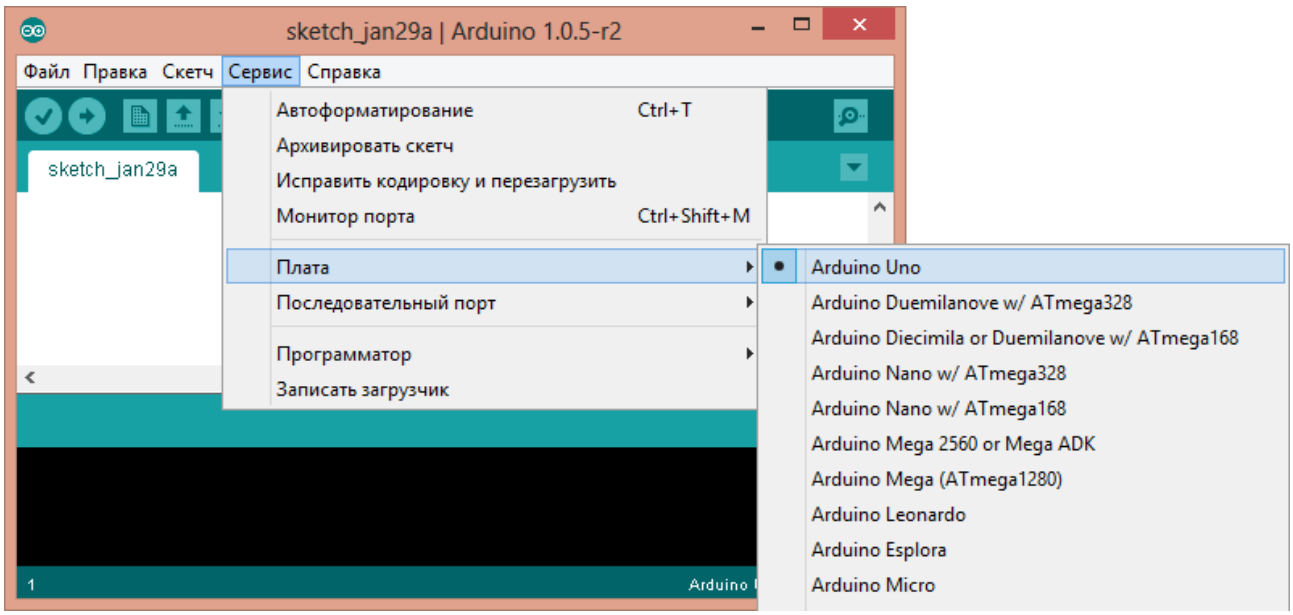
2) Настройка Arduino IDE на работу с ArduinoUno:

- Теперь нам необходимо сообщить Arduino IDE, что плата, с которой ей предстоит общаться, находится на COM-порту «COM7».

- Для этого переходим в меню «Сервис» → «Последовательный порт» и выбираем порт «COM7». Теперь Arduino IDE знает — что-то находится на порту «COM7». И с этим «чем-то» ей вскоре предстоит общаться.



Чтобы у Arduino IDE не осталось никаких сомнений, необходимо прямо указать: «Мы будем использовать ArduinoUno!». Для этого переходим в меню «Сервис» → «Плата» и выбираем нашу «ArduinoUno».



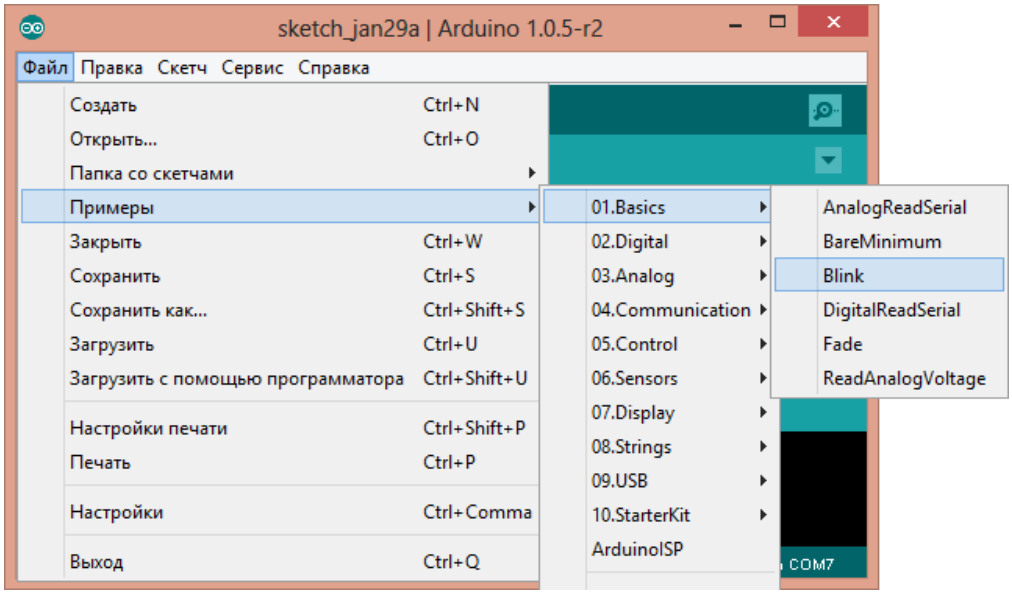
3) Анализ возникших проблем и выход из ситуации.

- Список последовательных портов пуст?
  - Значит ArduinoUno некорректно подключена. Вернитесь к пункту (3), чтобы отладить соединение.
- Arduino IDE невероятно тормозит при навигации по меню?
  - Отключите в диспетчере устройств все внешние устройства типа «BluetoothSerial». Например, виртуальное устройство для соединения с мобильным телефоном по Bluetooth может вызвать такое поведение.

4) Загрузка первого проверочного скетча.

Среда настроена, плата подключена. Теперь можно переходить к загрузке скетча.

Arduino IDE содержит очень много готовых примеров, в которых можно быстро подсмотреть решение какой-либо задачи.



Немного модифицируем код, чтобы увидеть разницу с заводским миганием светодиода.

Вместо строчки:

```
delay(1000);
```

напишем:

```
delay(100);
```

- теперь светодиод «L» должен загораться и гаснуть на десятую часть секунды. То есть в 10 раз быстрее, чем в заводской версии.

- загрузим наш скетч в ArduinoUno и проверим, так ли это? После загрузки светодиод начнёт мигать быстрее. Это значит, что всё получилось.

5).Напишите скетч8 секундного мигания светодиода, подключенного к 8 цифровому порту (все обучающиеся выполняют одинаковое подключение):

```
int led = 8;

Void setup() {

pinMode(led, OUTPUT); // установка 8-го контакта в режим вывода

}

voidloop() {

digitalWrite(led, HIGH); // вывод №8 в активное состояние

delay(1000); // пауза 1-секунда

digitalWrite(led, LOW); // вывод №8 в неактивное состояние

delay(1000); // пауза 1-секунда

}
```

- Нажмите кнопку загрузку программы
- Посмотрите, что получилось

6) Напишите скетч мигания светодиода

Задание 1. Светодиод подключен к 5 цифровому порту, контакт установлен на режим вывода, светодиод горит 5 секунд, пауза 3 секунды.

Задание 2. Светодиод подключен к 1 цифровому порту, контакт установлен на режим вывода, светодиод горит 10 секунд, пауза 4 секунды.

Сохраните результаты выполненной работы (написанные скетчи) в своей рабочей папке. Сформулируйте выводы, оформите отчет о проделанной работе.

Лабораторная работа 3. Программирование микроконтроллера АрдуиноUNO. Организация работы светофора.

Цель: получить навыки программирования микроконтроллера Ардуино, организовывать работу 3х светодиодов попеременно.

Методическое и информационное обеспечение.

1) Партыка Т.Л., Попов И.И. Периферийные устройства вычислительной техники: учеб. пособие для сред. проф. образования М.: Форум, 2014 г.
2) Авдеев В.А. Периферийные устройства: интерфейсы, схемотехника, программирование М.: ДМК Пресс, 2015г.
3) Методические указания по выполнению лабораторной работы.

Оборудование: ПК, стартовый комплект устройств Ардуино, программный пакет ARDUINO IDE.

Формируемые компетенции: ПК1.1., ПК 2.2.

Задачи:

1. Собрать схему подключения светодиода к макетной плате по заданным условиям;
2. Подключить микроконтроллер к ПК;
3. Написать скетч мигания 2х светодиодов согласно варианту;
4. Написать скетч мигания 3х светодиодов по типу работы светофора.

Краткие теоретические сведения

В процессор Arduino можно загрузить программу, которая будет управлять всеми этими устройствами по заданному алгоритму. Таким образом можно создать бесконечное количество уникальных гаджетов, сделанных своими руками и по собственной задумке. Программы для Arduino пишутся на обычном C++, дополненным простыми и понятными функциями для управления вводом/выводом на контактах. Если вы уже знаете C++ — Arduino станет дверью в новый мир, где программы не ограничены рамками компьютера, а взаимодействуют с окружающим миром и влияют на него. Arduino — это сердце конструктора, в котором нет конечного, определённого набора деталей, и нет ограничений в разнообразии того, что можно собрать. Всё ограничено лишь вашей фантазией.

Первую программу, которая управляла миганием светодиода, мы рассмотрели в предыдущей лабораторной работе. В этой части рассмотрим работу с группой светодиодов, а именно научимся управлять цветами светофора. Другими словами, нам необходимо управлять включением и выключением светодиодов при помощи микроконтроллера. В нашем случае, таким контроллером станет Ардуино.

Работа выполняется на образовательном конструкторе «Мастер ARDUINO»

Ход работы:

Задание 1. Собрать схему подключения светодиода к макетной плате по заданным условиям.

Инструкция по выполнению:

1) Для подключения мигающего светодиода на Arduino (смотреть теоретический материал) найдите в образовательном конструкторе следующие компоненты:
   - Плата ArduinoUNO;
   - breadboard;
   - 6 проводов «папа-папа»;
   - 3 светодиода: красный, зеленый, желтый;
   - 3 резистора;
   - Откройте среду разработки Arduino IDE.

Задание 2. Подключить микроконтроллер к ПК.

Инструкция по выполнению:

1) Перед вами схема включения 1 светодиода. Используя данный пример, организуйте включение 2х светодиодов.

- Чтобы зажечь 1 светодиод, нам потребуется подключить его к одному из цифровых выводов Ардуино.

- Катод светодиода подключим к минусу (земля), анод же соединим с выводом №8 через токозадающий резистор 150 Ом.

- На рисунке 9 показан пример подключения 1 светодиода, аналогично выполните схему подключения 2х светодиодов(второй порт для подключения выберите самостоятельно):

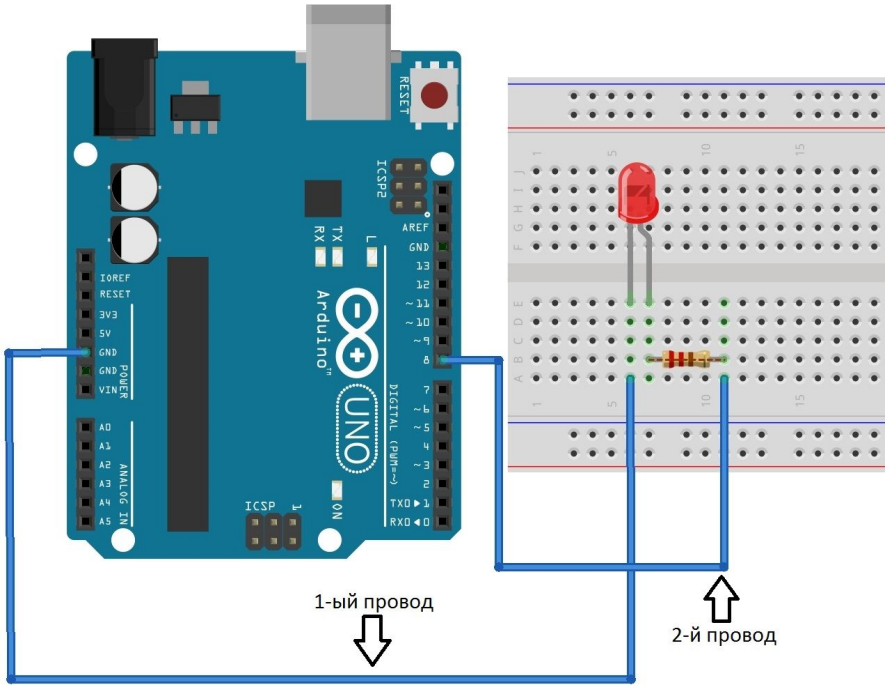


Рисунок 9. Схема подключения

2)Загрузка программы:

- Чтобы подключить плату к компьютеру необходимо взять USBпровод. Соедините ArduinoUno с компьютером через USB-кабель. Вы увидите, как на плате загорится светодиод «ON», и начнёт мигать светодиод «L». (См.рис 1.) Это означает, что на плату подано питание, и микроконтроллер ArduinoUno начал выполнять прошитую на заводе программу «Blink» (мигание светодиодом).

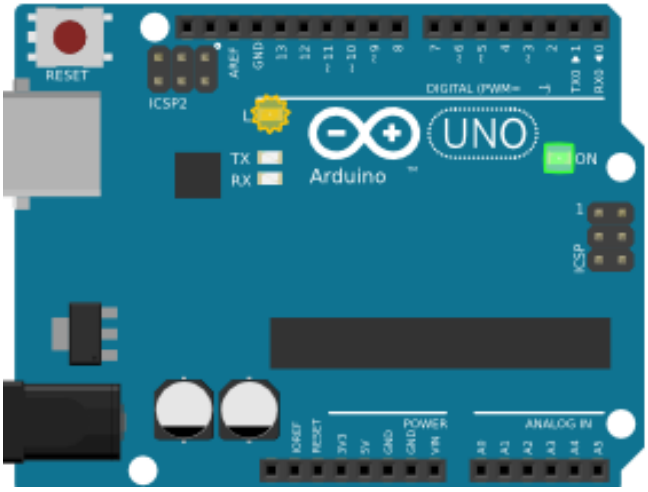


Рисунок 1.Питание подано

- Чтобы настроить Arduino IDE на работу с ArduinoUno, нам необходимо узнать, какой номер COM-порта присвоил компьютер ArduinoUno. Для этого нужно зайти в «Диспетчер устройств» Windows и раскрыть вкладку «Порты (COM и LPT)». Мы должны увидеть следующую картину см.Рис.2:

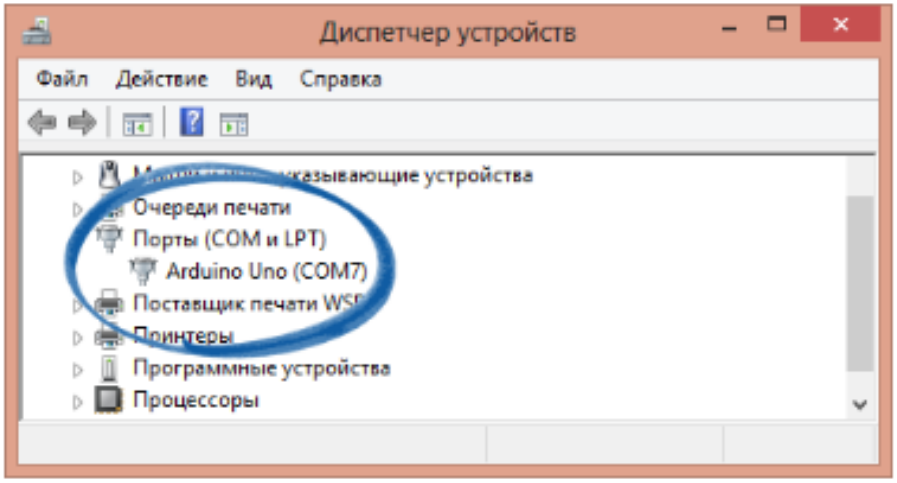


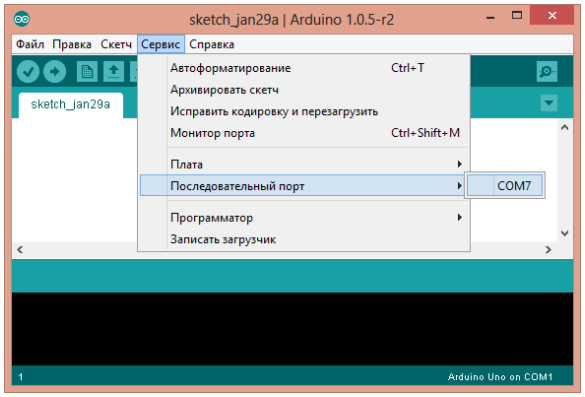
Рисунок 2.Диспетчер устройств

- Это означает, что операционная система распознала нашу плату ArduinoUno как COM-порт, подобрала для неё правильный драйвер и назначила этому COM-порту номер 7. Если мы подключим к компьютеру другую плату Arduino, то операционная система назначит ей другой номер. Поэтому, если у вас несколько плат Arduino, очень важно не запутаться в номерах COM-портов.

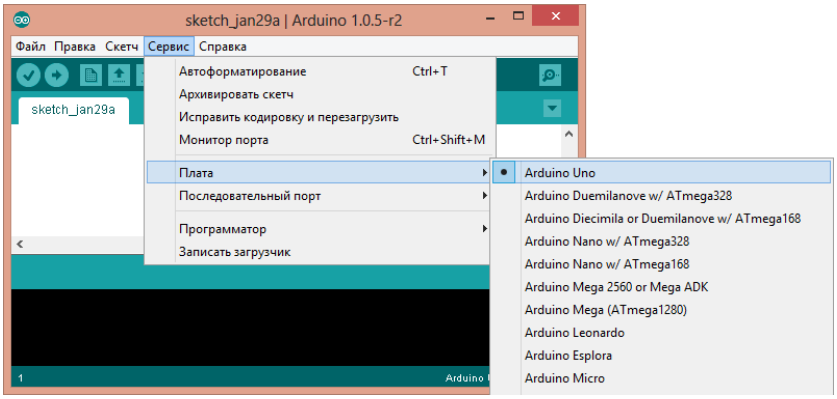
3) Настройка Arduino IDE на работу с ArduinoUno:

- Теперь нам необходимо сообщить Arduino IDE, что плата, с которой ей предстоит общаться, находится на COM-порту «COM7».

- Для этого переходим в меню «Сервис» → «Последовательный порт» и выбираем порт «COM7». Теперь Arduino IDE знает — что-то находится на порту «COM7». И с этим «чем-то» ей вскоре предстоит общаться.



Чтобы у Arduino IDE не осталось никаких сомнений, необходимо прямо указать: «Мы будем использовать ArduinoUno!». Для этого переходим в меню «Сервис» → «Плата» и выбираем нашу «ArduinoUno».



4) Анализ возникших проблем и выход из ситуации.

- Список последовательных портов пуст?

  - Значит ArduinoUno некорректно подключена. Вернитесь к пункту (3), чтобы отладить соединение.

- Arduino IDE невероятно тормозит при навигации по меню?

  - Отключите в диспетчере устройств все внешние устройства типа «BluetoothSerial». Например, виртуальное устройство для соединения с мобильным телефоном по Bluetooth может вызвать такое поведение.

Среда настроена, плата подключена. Теперь можно переходить к загрузке скетча

Задание 3. Написать скетч мигания 2х светодиодов согласно варианту

Инструкция по выполнению:

1)Перед вами скетч мигания 1 светодиода. Вам необходимо написать новый скетч, который будет управлять двумя светодиодами.

```
intled = 8;

voidsetup() {

pinMode(led, OUTPUT); // установка 2-го контакта в режим вывода

}

voidloop() {

digitalWrite(led, HIGH); // вывод №8 в активное состояние

delay(1000); // пауза 1-секунда

digitalWrite(led, LOW); // вывод №8 в неактивное состояние

delay(1000); // пауза 1-секунда

}
```

Задание для самостоятельной работы

Задание 3.1: Организуйте мигание двух светодиодов (желтый и красный) по правилу:

- Желтый светодиод мигает 4 секунды
- Задержка 1 секунда
- Красный светодиод мигает 6 секунд
- Задержка 1 секунда

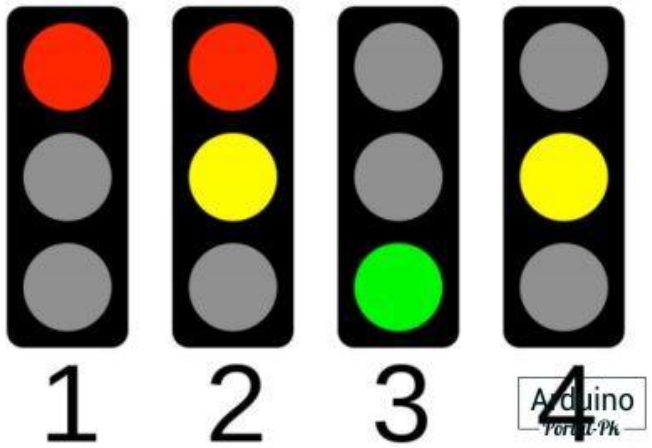
Задание 3.2: Организуйте мигание двух светодиодов (синий и красный) по правилу:

- Синий светодиод горит постоянно
- Красный светодиод мигает 10 секунд
- Задержка 4 секунды

Задание 4. Написать скетч мигания 3х светодиодов по типу работы светофора.

Инструкция по выполнению:

Организовать работу простого светофора из 3х светодиодов. Работать светофор должен согласно следующей схеме и правилу:



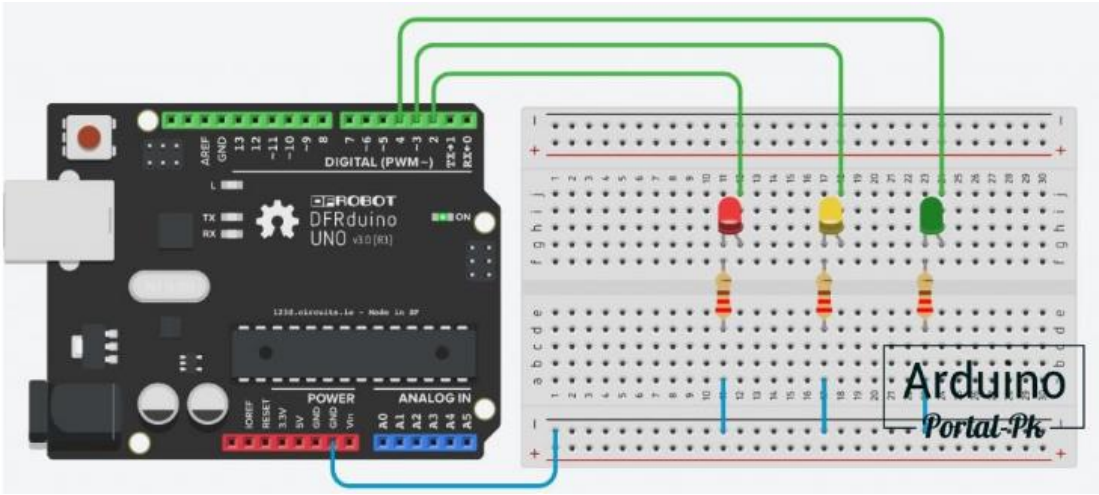
1. Светит только красный цвет нашего будущего светофора.

2. Не выключая красный сигнал светофора включаем желтый .

3. Выключаем красный и желтый включаем зеленый.

4. Выключаем зеленый сигнал светофора включаем желтый.

Примечание: Учитывая, что светодиодов у нас 3, а количество gnd портов на плате только 2, используйте следующую схему подключения, скетч напишите самостоятельно:



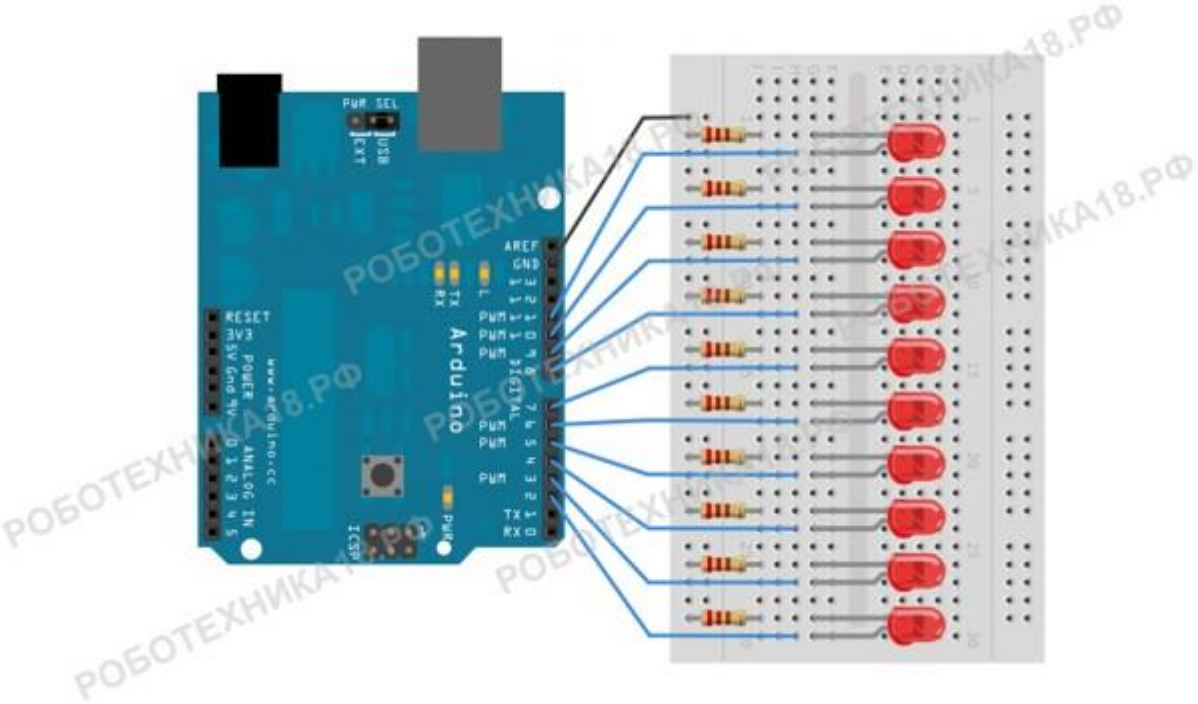
Сохраните результаты выполненной работы (написанные скетчи) в своей рабочей папке. Сформулируйте выводы, оформите отчет о проделанной работе.

Лабораторная работа 4. Последовательное включение светодиодов Ардуино. Использование циклов

Цель: научиться включать последовательно шкалу из десятка светодиодов, цикл **for**, который служит для повторения операторов в конструкции и служит своего рода циклом в цикле **loop**.

Ход работы:

Собрать схему



Для данной задачи можно использовать простейший скетч из второй лабораторной работы В скетче можно прописать последовательное включение всех светодиодов с заданной скоростью и дальнейшее их последовательное выключение. Но можно и воспользоваться циклом **for**, который значительно упростит нашу задачу. Во-первых, с помощью данного цикла мы можем задать количество светодиодов.

```
byte i;             // задаем переменную для цифровых выходов
byte LedMax = 13; // максимальное количество выходов (диодов)

void setup()
{
// задаем пины со 2 по 13, как выходы (OUTPUT)
for(i=2; i<LedMax; i++) { pinMode(i, OUTPUT); }
}
```

Вместо прописывания в скетче всех пинов с 2 по 13, как выходы, мы воспользовались одним циклом for. Сразу отметим, что использование всех выходов на Ардуино для последовательного включения светодиодов или подключения светодиодной шкалы не рационально. На практике для этого используют сдвиговый регистр 74hc165 или, еще более простой вариант драйвер светодиодов M5450B7 для микроконтроллера Ардуино.

Подключение светодиодной шкалы к Ардуино

Для этого занятия нам потребуется:

- плата Arduino Uno / Arduino Nano / Arduino Mega;
- макетная плата;
- 11 светодиодов или светодиодная шкала;
- 11 резисторов на 220 Ом;
- провода «папа-папа» и «папа-мама».

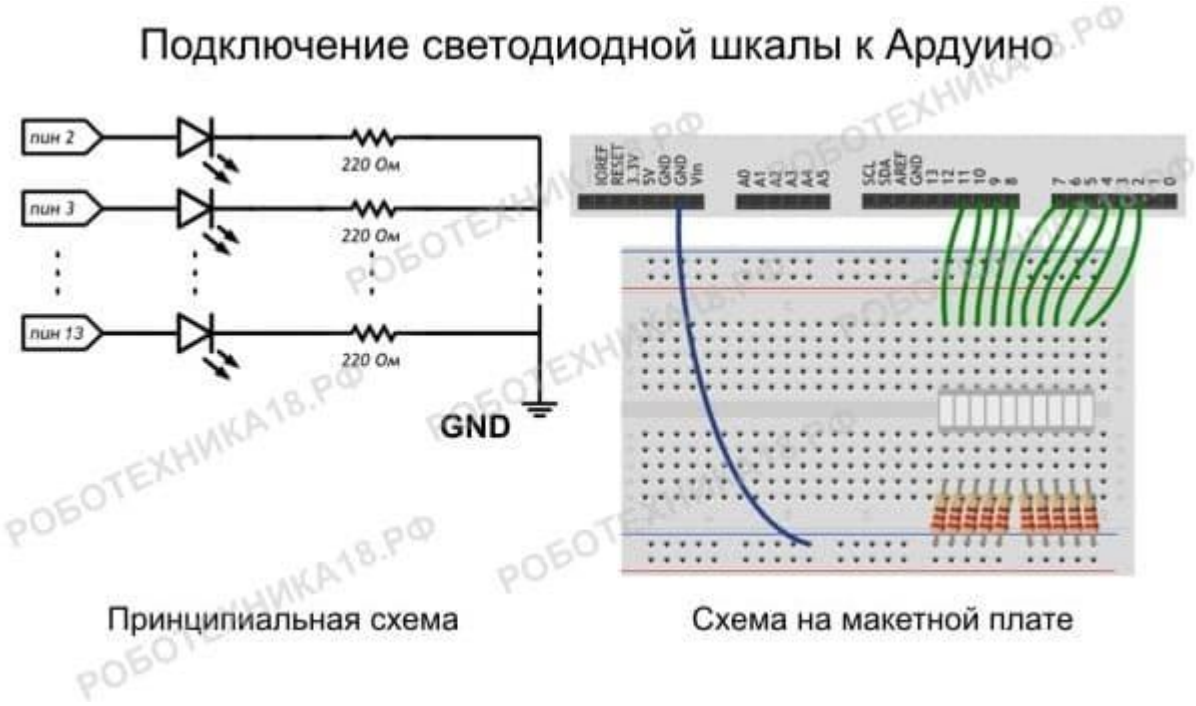


Схема подключения светодиодной шкалы к Ардуино

Соберите схему из светодиодов на макетной плате или подключите к Ардуино светодиодную шкалу, как на рисунке выше. Обратите внимание, что в светодиодной шкале нет встроенных резисторов, поэтому не забывайте их добавить в схему, чтобы диоды не сгорели. После сборки схемы скопируйте скетч последовательного включения светодиодной шкалы на Arduino UNO и загрузите в плату.

Скетч последовательного включения светодиодов

```
byte i;          // задаем переменную для цифровых выходов
byte LedMin = 2;  // первый пин со светодиодом
byte LedMax = 11; // последний пин со светодиодом

void setup() {
  // задаем пин со 2 по 11, как выход (OUTPUT)
  for(i=LedMin; i<=LedMax; i++) { pinMode(i, OUTPUT); }
}

void loop() {
  // последовательно зажигаем светодиоды со 2 по 11 пин
  for(i=LedMin; i<=LedMax; i++) {
    digitalWrite(i, HIGH);
    delay(50);
  }

  // последовательно гасим светодиоды с 11 по 2 пин
  for(i=LedMax; i>=LedMin; i--) {
    digitalWrite(i, LOW);
    delay(50);
  }
}
```

Пояснения к коду:

1. цикл loop в скетче содержит в себе два цикла for, работающих попеременно;
2. переменная типа byte может хранить 8-битное беззнаковое число от 0 до 255.

В следующем примере изменен порядок последовательного выключения светодиодов на Ардуино. Для этого во втором цикле for все условия повторяются, меняется лишь состояние пинов у светодиодов в цикле с HIGH на LOW. В коде используются глобальные переменные типа byte, поэтому их можно вызывать в любом месте программы, в отличии от локальных переменных в языке Ардуино.

Скетч последовательного выключения светодиодов

```
byte i;          // задаем переменную для цифровых выходов
byte LedMin = 2;  // первый пин со светодиодом
byte LedMax = 11; // последний пин со светодиодом

void setup() {
  // задаем пин со 2 по 11, как выход (OUTPUT)
  for(i=LedMin; i<=LedMax; i++) { pinMode(i, OUTPUT); }
}

void loop() {
  // последовательно зажигаем светодиоды со 2 по 11 пин
  for(i=LedMin; i<=LedMax; i++) {
    digitalWrite(i, HIGH);
    delay(50);
  }

  // последовательно гасим светодиоды со 2 по 11 пин
  for(i=LedMin; i<=LedMax; i++) {
    digitalWrite(i, LOW);
    delay(50);
  }
}
```

Пояснения к коду:

1. цикл for повторяется до тех пор, пока верно условие i>LedMin или i<LedMax.
2. если попробовать удалить из программы задержку delay(50);, то эффекта последовательного зажигания светодиодов не будет.

Задания для самостоятельного выполнения

Организовать включение светодиодов по порядку и выключение в обратном порядке.

Организовать включение светодиодов по порядку и выключение в этом же порядке.

Организовать включение светодиодов от края к центру с двух сторон и выключение в обратном порядке.

Включить светодиоды через один. Выключить в обратном порядке.

Включить светодиоды от края к центру и выключить в том же порядке.

Включить светодиоды от края к центру и выключить в обратном порядке.

Лабораторная работа 5. Программирование микроконтроллера АрдуиноUNO. Подключение кнопки.

Цель: Получить навыки программирования микроконтроллера Ардуино, организовывать работу кнопки.

Методическое и информационное обеспечение.

1) Партыка Т.Л., Попов И.И. Периферийные устройства вычислительной техники: учеб. пособие для сред. проф. образования М.: Форум, 2014 г.
2) Авдеев В.А. Периферийные устройства: интерфейсы, схемотехника, программирование М.: ДМК Пресс, 2015г.
3) Методические указания по выполнению лабораторной работы.

Формируемые компетенции: ПК1.1., ПК 2.2.

Задачи:

1. Собрать схему подключения светодиода к макетной плате по заданным условиям;
2. Подключить микроконтроллер к ПК;
3. Подключить кнопку к макетной плате;
4. Написать скетч мигания светодиода по нажатию кнопки согласно варианту.

Краткие теоретические сведения:

В процессор Arduino можно загрузить программу, которая будет управлять всеми этими устройствами по заданному алгоритму. Таким образом можно создать бесконечное количество уникальных гаджетов, сделанных своими руками и по собственной задумке. Программы для Arduino пишутся на обычном C++, дополненным простыми и понятными функциями для управления вводом/выводом на контактах. Если вы уже знаете C++ — Arduino станет дверью в новый мир, где программы не ограничены рамками компьютера, а взаимодействуют с окружающим миром и влияют на него. Arduino — это сердце конструктора, в котором нет конечного, определённого набора деталей, и нет ограничений в разнообразии того, что можно собрать. Всё ограничено лишь вашей фантазией.

Первую программу, которая управляла миганием светодиода, мы рассмотрели в предыдущих лабораторных работах. В этой лабораторной работе рассмотрим, как можно управлять с помощью кнопки светодиодами и другими устройствами.Это одна из базовых операций, которая неоднократно пригодиться вам в будущем и может использоваться для управления роботом на Arduino, т.к. по нажатию кнопки могут происходить самые разные действия. Подключение датчика кнопки к Ардуино требует определенных знаний и навыков. Поэтому внимательно ознакомьтесь с теоретическим материалом приведенным ниже! И только после этого приступайте к работе, строго следуя инструкции.

Кнопка

Кнопка (или кнопочный переключатель) – самый простой и доступный из всех видов датчиков. Нажав на нее, вы подаете контроллеру сигнал, который затем приводит к каким-то действиям: включаются светодиоды, издаются звуки, запускаются моторы. В своей жизни мы часто встречаемся с разными выключателями и хорошо знакомы с этим устройством.Что такое кнопка? По сути, это достаточно простое устройство, замыкающее и размыкающее электрическую сеть.

Выполнять это замыкание/размыкание можно в разных режимах, при этому фиксировать или не фиксировать свое положение. Соответственно, все кнопки можно поделить на две большие группы:

- Кнопки переключатели с фиксацией. Они возвращаются в исходное состояние после того, как их отпустили.В зависимости от начального состояния разделяют на нормально-замкнутые и нормально-разомкнутые кнопки.
- Кнопки без фиксации (тактовые кнопки). Они фиксируются и остаются в том положении, в котором их оставили.

Вариантов различных кнопок великое множество, это действительно один из самых распространенных видов электронных компонентов(см.Рис.1,2).

Рис.1.Разновидности кнопок

Рис.2.Разновидности кнопок

В наших проектах мы будем работать с очень простыми тактовыми кнопками с 4 ножками, которые идут практически в любом наборе ардуино. Кнопка представляет собой переключатель с двумя парами контактов. Контакты в одной паре соединены между собой, поэтому больше одного выключателя в схеме реализовать не удастся, но вы можете одновременно управлять двумя параллельными сегментами, это бывает полезно.

В зависимости от ситуации, вы можете создавать как схемы с нормально замкнутыми, так и с нормально разомкнутыми контактами – для этого нужно будет только соответствующим образом выполнить соединение в схеме.

Для удобства работы в комплекте с тактовой кнопкой обычно идет пластмассовый колпачок какого-то цвета, он достаточно очевидно надевается на кнопку и придает проекту приемлемый вид.

Дребезг кнопки Ардуино

В процессе работы с кнопками мы можем столкнуться с очень неприятным явлением, называемым дребезгом кнопки. Как следует из самого названия, явление это обуславливается дребезгом контактов внутри кнопочного переключателя. Металлические пластины соприкасаются друг с другом не мгновенно (хоть и очень быстро для наших глаз), поэтому на короткое время в зоне контакта возникают скачки и провалы напряжения. Если мы не предусмотрим появление таких “мусорных” сигналов, то будем реагировать на них каждый раз и можем привести наш проект к хаусу.

Для устранения дребезга используют программные и аппаратные решения. В двух словах лишь упомянем основные методы подавления дребезга:

- Добавляем в скетче паузу 10-50 миллисекунд между получением значений с пинаардуино.(Именно этот прием мы будем использовать в наших лабораторных работах)
- Если мы используем прерывания, то программный метд использоваться не может и мы формируем аппаратную защиту. Простейшая из них – RC фильтр с конденсатором и сопротивлением.
- Для более точного подавления дребезга используется аппаратный фильтр с использованием триггера шмидта. Этот вариант позволит получить на входе в Ардуино сигнал практически идеальной формы.

Переключение режимов с помощью кнопки

Чтобы написать скетч для работы кнопки уже установленной и подключенной(как именно это сделать подробно описывается в ходе работы) необходимо определить, была ли нажата кнопка! Поэтому надо просто зафиксировать факт ее нажатия и сохранить признак в специальной переменной.

Факт нажатия мы определяем с помощью функции digitalRead(). В результате мы получим HIGH (1, TRUE) или LOW(0, FALSE), в зависимости от того, как подключили кнопку. Если мы подключаем кнопку с помощью внутреннего подтягивающего резистора, то нажатие кнопки приведет к появлению на входе уровня 0 (FALSE).Для хранения информации о нажатии на кнопку можно использовать переменную типа boolean:booleankeyPressed = digitalRead(PIN_BUTTON)==LOW;

Почему мы используем такую конструкцию, а не сделали так:

booleankeyPressed = digitalRead(PIN_BUTTON);

Все дело в том, что digitalRead() может вернуть HIGH, но оно не будет означать нажатие кнопки. В случае использования схемы с подтягивающим резистором HIGH будет означать, что кнопка, наоборот, не нажата. В первом варианте (digitalRead(PIN_BUTTON)==LOW) мы сразу сравнили вход с нужным нам значением и определили, что кнопка нажата, хотя и на входе сейчас низкий уровень сигнала. И сохранили в переменную статус кнопки. Старайтесь явно указывать все выполняемые вами логические операции, чтобы делать свой код более прозрачным и избежать лишних глупых ошибок.

Как переключать режимы работы после нажатия кнопки?

Часто возникает ситуация, когда мы с помощью кнопок должны учитывать факт не только нажатия, но и отпускания кнопки. Например, нажав и отпустив кнопку, мы можем включить свет или переключить режим работы схемы. Другими словами, нам нужно как-то зафиксировать в коде факт нажатия на кнопку и использовать информацию в дальнейшем, даже если кнопка уже не нажата. Давайте посмотрим, как это можно сделать.

Логика работы программы очень проста:

- Запоминаем факт нажатия в служебной переменной.
- Ожидаем, пока не пройдут явления, связанные с дребезгом.
- Ожидаем факта отпускания кнопки.

Запоминаем факт отпускания и устанавливаем в отдельной переменной признак того, что кнопка была полноценно нажата.Очищаем служебную переменную.

Как определить нажатие нескольких кнопок?

Нужно просто запомнить состояние каждой из кнопок в соответствующей переменной или в массиве ардуино. Здесь главное понимать, что каждая новая кнопка – это занятыйпин. Поэтому если количество кнопок у вас будет большим, то возможно возникновение дефицита свободных контактов. Альтернативным вариантом является использование подключения кнопок на один аналоговый пин по схеме с резистивным делителем.

Работа выполняется на образовательном конструкторе «Мастер ARDUINO»

Ход работы:

Задание 1. Собрать схему подключения светодиода к макетной плате по заданным условиям.

Инструкция по выполнению:

1) Для подключения светодиода на Arduino (смотреть теоретический материал) найдите в образовательном конструкторе следующие компоненты:
   - 1 плата ArduinoUno
   - 1 беспаечная макетная плата
   - 1 тактовая кнопка
   - 1 резистор номиналом 150 Ом
   - 1 светодиод
   - 5 проводов «папа-папа»

Для дополнительного задания

- еще 1 кнопка
- еще 2 провода

Откройте среду разработки Arduino IDE.

Задание 2.Подключить микроконтроллер к ПК;

1) Перед вами схема включения кнопки и светодиода. Выполните подключение согласно схеме(см.рис.3):

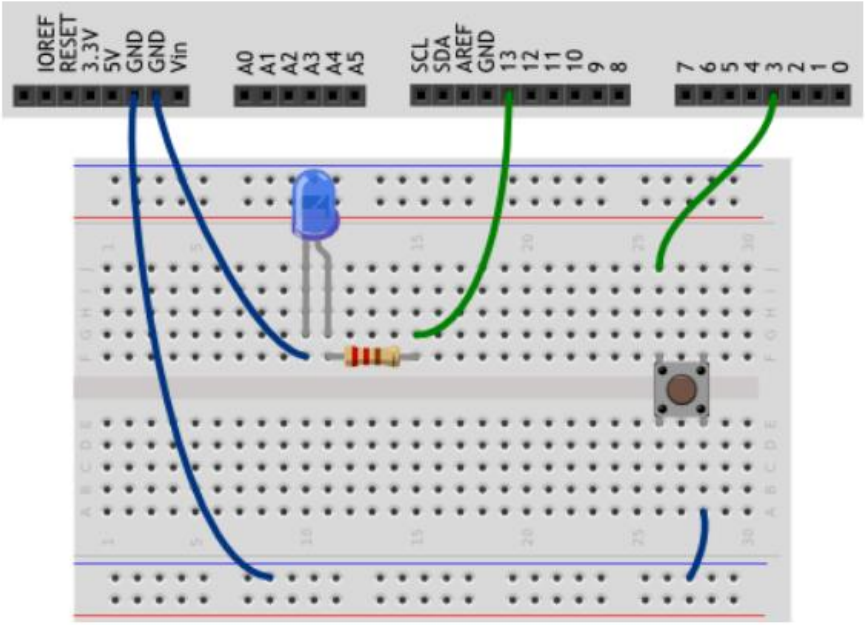


Рисунок 3. Схема подключения

Обратите внимание!

Мы могли бы один из контактов кнопки соединить проводом напрямую с одним из входов GND, но мы сначала «раздали» «землю» на длинную рельсу макетной платы. Если мы работаем с макетной платой, так поступать удобнее, т.к. в схеме могут появляться новые участки, которые тоже нужно будет соединить с «землей».

Также полезно руководствоваться соображениями аккуратности изделия, поэтому катод светодиода мы соединяем с другим входом GND отдельным проводом, который не мешает нам работать в середине макетной платы.

Задание 3. Подключить кнопку к макетной плате (см.теоретический материал).

Задание 4. Написать скетч мигания светодиода по нажатию кнопки.

Задание4.1. Светодиод горит только в том случае, когда зажаты и отпущены обе кнопки.

Вариант 4.2.Светодиод горит только в том случае, когда зажаты и не отпущены обе кнопки (необязательное, на дополнительную оценку).

Инструкция по выполнению:

1)Внимательно изучите предложенный ниже скетч. Напишите скетч, в котором светодиод включается по нажатию кнопки.

```
int BUTTON_PIN=  3;
int LED_PIN   =  13;

boolean buttonWasUp = true;  // была ли кнопка отпущена?
boolean ledEnabled = false;  // включен ли свет?

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}
voidloop()
{
  // определить момент «клика» несколько сложнее, чем факт того,
  // что кнопка сейчас просто нажата. Для определения клика мы
  // сначала понимаем, отпущена ли кнопка прямо сейчас...
boolean buttonIsUp = digitalRead(BUTTON_PIN);

// ...если «кнопка была отпущена и (&&) не отпущена сейчас»...
if (buttonWasUp&& !buttonIsUp) {
    // ...может это «клик», а может и ложный сигнал (дребезг),
// возникающий в момент замыкания/размыкания пластин кнопки,
    // поэтому даём кнопке полностью «успокоиться»...
delay(10);
    // ...и считываем сигнал снова
    buttonIsUp = digitalRead(BUTTON_PIN);
    if (!buttonIsUp) {  // если она всё ещё нажата...
// ...это клик! Переворачиваем сигнал светодиода
ledEnabled = !ledEnabled;
digitalWrite(LED_PIN, ledEnabled);
    }
} // запоминаем последнее состояние кнопки для новой итерации
buttonWasUp = buttonIsUp;
}
```

- Нажмите кнопку загрузку программы
- Посмотрите, что получилось.

Пояснения к коду

- Поскольку мы сконфигурировали вход кнопки как INPUT_PULLUP, при нажатии на кнопку на данном входе мы будем получать 0. Поэтому мы получим значение true («истина») в булевой переменной buttonIsUp («кнопка отпущена»), когда кнопка отпущена.

- Логический оператор && («и») возвращает значение «истина» только в случае истинности обоих его операндов. Взглянем на так называемую таблицу истинности для выражения buttonWasUp&& !buttonIsUp («кнопка была отпущена и кнопка не отпущена»):

buttonWasUp	buttonIsUp	!buttonIsUp	buttonWasUp&& !buttonIsUp

0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Здесь рассмотрены все возможные сочетания предыдущего и текущего состояний кнопки и мы видим, что наш условный оператор if сработает только в случае, когда кнопка нажата только что: предыдущее состояние 1 («была отпущена»), а текущее 0 («не отпущена»).

- Через 10 миллисекунд мы проверяем еще раз, нажата ли кнопка: этот интервал больше, чем длительность «дребезга», но меньше, чем время, за которое человек успел бы дважды нажать на кнопку. Если кнопка всё еще нажата, значит, это был не дребезг.

- Мы передаем в digitalWrite не конкретное значение HIGH или LOW, а просто булеву переменную ledEnabled. В зависимости от того, какое значение было для нее вычислено, светодиод будет зажигаться или гаситься.

- Последняя инструкция в buttonWasUp = buttonIsUp сохраняет текущее состояние кнопки в переменную предыдущего состояния, ведь на следующей итерации loop текущее состояние уже станет историей.

Сохраните результаты выполненной работы (написанные скетчи) в своей рабочей папке.
Сформулируйте выводы, оформите отчет о проделанной работе.

Лабораторная работа 6. Программирование микроконтроллера АрдуиноUNO. Подключение фоторезистора.

Цель: Получить навыки программирования микроконтроллера Ардуино, организовывать работу фоторезистора.

Методическое и информационное обеспечение.

1) Партыка Т.Л., Попов И.И. Периферийные устройства вычислительной техники: учеб.пособие для сред. проф. образования М.: Форум, 2014 г.
2) Авдеев В.А. Периферийные устройства: интерфейсы, схемотехника, программирование М.: ДМК Пресс, 2015г.

Формируемые компетенции: ПК1.1., ПК 2.2.

Задачи:

1. Собрать схему подключения светодиода к макетной плате по заданным условиям;
2. Подключить микроконтроллер к ПК;
3. Подключить фоторезистор к макетной плате;
4. Написать скетч мигания светодиода в зависимости данных подаваемых на фоторезистор и устанавливаемого порогового значения.

Краткие теоретические сведения:

Датчики освещенности (освещения), построенные на базе фоторезисторов, довольно часто используются в реальных ардуино проектах. Они относительно просты, не дороги, их легко найти и купить в любом интернет-магазине. Фоторезистор ардуино позволяет контролировать уровень освещенности и реагировать на его изменение. В этой лабораторной работе мы рассмотрим, что такое фоторезистор, как работает датчик освещенности на его основе, как правильно подключить датчик в платамArduino.Фоторезисторы достаточно активно применяются в самых разнообразных системах. Самый распространенный вариант применения — фонари уличного освещения. Если на город опускается ночь или стало пасмурно, то огни включаются автоматически. Можно сделать из фоторезистора экономную лампочку для дома, включающуюся не по расписанию, а в зависимости от освещения. На базе датчика освещенности можно сделать даже охранную систему, которая будет срабатывать сразу после того, как закрытый шкаф или сейф открыли и осветили. Как всегда, сфера применения любых датчиков ардуино ограничена лишь нашей фантазией. Поэтому внимательно ознакомьтесь с теоретическим материалом приведенным ниже! И только после этого приступайте к работе, строго следуя инструкции.

Фоторезистор Ардуино

Фоторезистор, как следует из названия, имеет прямое отношение к резисторам, которые часто встречаются практически в любых электронных схемах. Основной характеристикой обычного резистора является величина его сопротивления. От него зависят напряжение и ток, с помощью резистора мы выставляем нужные режимы работы других компонентов. Как правило, значение сопротивления у резистора в одних и тех же условиях эксплуатации практически не меняется.

В отличие от обычного резистора, фоторезистор может менять свое сопротивление в зависимости от уровня окружающего освещения. Это означает, что в электронной схеме будут постоянно меняться параметры, в первую очередь нас интересует напряжение, падающее на

фоторезисторе. Фиксируя эти изменения напряжения на аналоговых пинахардуино, мы можем менять логику работы схемы, создавая тем самым адаптирующиеся под вешние условия устройства.

Самый популярный и доступный вариант датчика на рынке – это модели массового выпуска китайских компаний, клоны изделий производителя VT. Там не всегда можно разораться, кто и что именно производит тот или иной поставщик, но для начала работы с фоторезисторами вполне подойдет самый простой вариант(см.Рис.1).

Рисунок 1.Фоторезистор

Начинающему ардуинщику можно посоветовать купить готовый фотомодуль, который выглядит вот так (рис.2.):

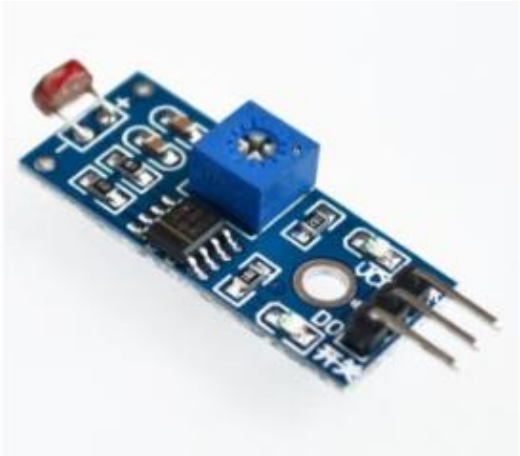


Рисунок 2. Фотомодуль

На этом модуле уже есть все необходимые элементы для простого подключения фоторезистора к плате ардуино. В некоторых модулях реализована схема с компаратором и доступен цифровой выход и подстроечный резистор для управления.

Российскому радиолюбителю можно посоветовать обратить на российский датчик ФР. Встречающиеся в продаже ФР1-3, ФР1-4 и т.п. — выпускались ещё в союзовские времена. Но, несмотря на это, ФР1-3 – более точная деталь. Из этого следует и разница в цене За ФР просят не более 400 рублей. ФР1-3 будет стоить больше тысячи рублей за штуку.

Маркировка фоторезистора

Современная маркировка моделей, выпускаемых в России, довольно простая. Первые две буквы — ФотоРезистор, цифры после чёрточки обозначают номер разработки. ФР -765 — фоторезистор, разработка 765. Обычно маркируется прямо на корпусе детали

У датчика VT в схеме маркировке указаны диапазон сопротивлений. Например:

VT83N1 — 12-100кОм (12K – освещенный, 100K – в темноте)

VT93N2 — 48-500кОм (48K –освещенный, 100K – в темноте).

Иногда для уточнения информации о моделях продавец предоставляет специальный документ от производителя. Кроме параметров работы там же указывается точность детали. У всех моделей диапазон чувствительности расположен в видимой части спектра. Собирая датчик света нужно понимать, что точность срабатывания — понятие условное. Даже у моделей одного производителя, одной партии, одной закупки отличаться она может на 50% и более.

На заводе детали настраиваются на длину волны от красного до зелёного света. Большинство при этом «видит» и инфракрасное излучение. Особо точные детали могут улавливать даже ультрафиолет.

Достоинства и недостатки датчика

Основным недостатком фоторезисторов является чувствительность к спектру. В зависимости от типа падающего света сопротивление может меняется на несколько порядков. К минусам также относится низкая скорость реакции на изменение освещённости. Если свет мигает — датчик не успевает отреагировать. Если же частота изменения довольно велика — резистор вообще перестанет «видеть», что освещённость меняется.

К плюсам можно отнести простоту и доступность. Прямое изменение сопротивления в зависимости от попадающего на неё света позволяет упростить электрическую схему подключения. Сам фоторезистор очень дешев, входит в состав многочисленных наборов и конструкторов ардуино, поэтому доступен практически любому начинающему ардуинщику.

Подключение фоторезистора к Ардуино

В проектах arduino фоторезистор используется как датчик освещения. Получая от него информацию, плата может включать или выключать реле, запускать двигатели, отсылать сообщения. Естественно, при этом мы должны правильно подключить датчик.

Схема подключения датчика освещенности к ардуино довольна проста. Если мы используем фоторезистор, то в схеме подключения датчик реализован как делитель напряжения. Одно плечо меняется от уровня освещённости, второе – подаёт напряжение на аналоговый вход. В микросхеме контроллера это напряжение преобразуется в цифровые данные через АЦП. Т.к. сопротивление датчика при попадании на него света уменьшается, то и значение падающего на нем напряжения будет уменьшаться.

В зависимости от того, в каком плече делителя мы поставили фоторезистор, на аналоговый вход будет подаваться или повышенное или уменьшенное напряжение. В том случае, если одна нога фоторезистора подключена к земле, то максимальное значение напряжения будет соответствовать темноте (сопротивление фоторезистора максимальное, почти все напряжение падает на нем), а минимальное – хорошему освещению (сопротивление близко к нулю, напряжение минимальное). Если мы подключим плечо фоторезистора к питанию, то поведение будет противоположным.

Сам монтаж платы не должен вызывать трудностей. Так как фоторезистор не имеет полярности, подключить можно любой стороной, к плате его можно припаять, подсоединить проводами с помощью монтажной платы или использовать обычные клипсы (крокодилы) для соединения. Источником питания в схеме является сам ардуино. Фоторезистор подсоединяется одной ногой к земле, другая подключается к АЦП платы (в нашем примере – АО). К этой же ноге подключаем резистор 10 кОм. Естественно, подключать фоторезистор можно не только на аналоговый пин А0, но и на любой другой.

Несколько слов относительно дополнительного резистора на 10К. У него в нашей схеме две функции: ограничивать ток в цепи и формировать нужное напряжение в схеме с делителем. Ограничение тока нужно в ситуации, когда полностью освещенный фоторезистор резко уменьшает свое сопротивление. А формирование напряжения – для предсказуемых значений на аналоговом порту. На самом деле для нормальной работы с нашими фоторезисторами хватит и сопротивления 1К.

Меняя значение резистора мы можем “сдвигать” уровень чувствительности в “темную” и “светлую” сторону. Так, 10 К даст быстрое переключение наступления света. В случае 1К датчик света будет более точно определять высокий уровень освещенности.

Если на плате представлен цифровой выход, то отправляем его на цифровыепины. Если аналоговый – то на аналоговые. В первом случае мы получим сигнал срабатывания – превышения уровня освещенности (порог срабатывания может быть настроен с помощью резистора подстройки). С аналоговых же пинов мы сможем получать величину напряжения, пропорциональную реальному уровню освещенности.

Работа выполняется на образовательном конструкторе «Мастер ARDUINO»

Ход работы:

Задание 1.Собрать схему подключения светодиода к макетной плате по заданным условиям.

Инструкция по выполнению:

1)Найдите в образовательном конструкторе следующие компоненты:

- 1 плата ArduinoUno
- 1 беспаечная макетная плата
- 1 резистор номиналом 150 Ом
- 1 резистор номиналом 1КОм
- 1 светодиод
- 5 проводов «папа-папа»
- 1 фоторезистор

Откройте среду разработки Arduino IDE.

3) Перед вами схема подключения фоторезистора и светодиода. Выполните подключение согласно схеме(см.рис.3):

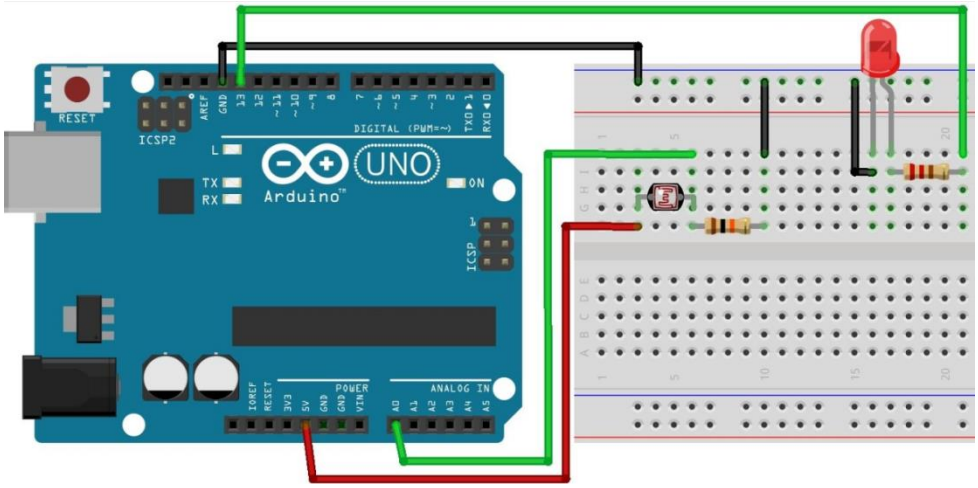


Рисунок 3. Схема подключения

Мы подключили схему с фоторезистором к ардуино, убедились, что все сделали правильно. Теперь осталось запрограммировать контроллер.

Написать скетч для датчика освещенности довольно просто. Нам нужно только снять текущее значение напряжения с того аналогового пина, к которому подключен датчик. Делается это с помощью известной нам всем функции analogRead(). Затем мы можем выполнять какие-то действия, в зависимости от уровня освещенности. Давайте напишем скетч для датчика освещенности, включающего или выключающего светодиод, подключенный по схеме, укзанной выше;

Задание 2. Подключить фоторезистор к макетной плате(см.теоретический материал);

Задание 3. Написать скетч мигания светодиода в зависимости данных подаваемых на фоторезистор и устанавливаемого порогового значения.

Инструкция по выполнению:

1)Внимательно изучите предложенный ниже скетч. Напишите скетч, в котором светодиод включается, если в помещении темно согласно варианту. Прикрывая фоторезистор (руками или светонепроницаемым предметом), можем наблюдать включение и выключение светодиода. Изменяя в коде пороговый параметр, можем заставлять включать/выключать лампочку при разном уровне освещения.

Алгоритм работы таков:

- Определяем уровень сигнала с аналогового пина.
- Сравниваем уровень с пороговым значением. Максимально значение будет соответствовать темноте, минимальное – максимальной освещенности. Пороговое значение выберем равное 300.
- Если уровень меньше порогового – темно, нужно включать светодиод.
- Иначе – выключаем светодиод.

```
Int led = ;
int PHOTO_SENSOR= ;

void setup() {
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH);
}

void loop() {
```

```
  int val = analogRead(PHOTO_SENSOR);
  if (val <) {
    // Светло, выключаем реле
    digitalWrite(led, HIGH);
  } else {
    // Темновато, включаем лампочку
    digitalWrite(led,  LOW);
}
}
```

Вариант 1.Фоторезистор реагирует на совещение, начиная с порогового занчения 300
Вариант 2.Фоторезистор реагирует на совещение, начиная с порогового занчения 1100

2)Изменяя пороговый параметр фоторезистора, включайте/выключайте лампочку при разном уровне освещения.

Сохраните результаты выполненной работы (написанные скетчи) в своей рабочей папке. Сформулируйте выводы, оформите отчет о проделанной работе.

Лабораторная работа 7. Подключение клавиатуры к микроконтроллеру Arduino.

Цель: научиться подключать клавиатуру к микроконтроллеру Arduino, научиться программировать выполнение заданного действия при нажатии определенной клавиши.

Теоретические сведения

В мире электроники клавиатурой называют устройство, состоящее из отдельных кнопок, которое позволяет вводить информацию в компьютер или в иной электронный прибор.

Популярная в Ардуино проектах матричная мембранная клавиатура имеет 12 или 16 кнопок. С нижней части клавиатуры выходит шлейф, заканчивающийся dupont розеткой.

Схема подключения матричной клавиатуры к Arduino

Подключать ее к плате следует 8 выводами, каждый из них считывает значения с определенных строк и столбцов. Подключать их следует к выводам на панели Digital. Контакты на клавиатуре отсчитываются слева на право.

Клавиатура 4*4	Arduino (uno)
Контакт 1	pin 11
Контакт 2	pin 10
Контакт 3	pin 9
Контакт 4	pin 8
Контакт 5	pin 7
Контакт 6	pin 6
Контакт 7	pin 5
Контакт 8	pin 4

Контакты к которым подключаем клавиатуру, могут быть перенастроены на любые другие цифровые контакты. Настраиваются здесь:

```
byterowPins[ROWS] = {7, 6, 5, 4}; //к каким выводам подключаем управление строками

bytecolPins[COLS] = {11, 10, 9, 8}; //к каким выводам подключаем управление столбцами
```

Чтобы работать с клавиатурой можно воспользоваться библиотекой Keypad.

Если вы работаете в среде ArduidoIDE, библиотеку надо скачать и поместить в папку libraries. В эмуляторе эта библиотека уже присутствует.

Скетч

```
#include<Keypad.h> // подключаем нашу библиотеку
constbyte ROWS = 4; //число строк у нашей клавиатуры
constbyte COLS = 4; //число столбцов у нашей клавиатуры
charhexaKeys[ROWS][COLS] = {
{'1','4','7','*'}, // здесь мы располагаем названия наших клавиш, как на клавиатуре,для //удобства
пользования
{'2','5','8','0'},
{'3','6','9','#'},
{'A','B','C','D'}
};
byterowPins[ROWS] = {7, 6, 5, 4}; //к каким выводам подключаем управление строками
bytecolPins[COLS] = {11, 10, 9, 8}; //к каким выводам подключаем управление столбцами
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
void setup(){
Serial.begin(9600);
}
void loop(){
char customKey = customKeypad.getKey();
if (customKey){
Serial.println(customKey);
}
}
```

Скетч очень простой. Стоит отметить первые строчки кода. Сначала подключаем библиотеку, затем указываем сколько строк и столбцов у клавиатуры, а потом нужно правильно расположить названия клавиш, чтобы было удобнее работать.

Если это сделать неправильно, то, например, при нажатии цифры 4, в порт выйдет цифра 6 или любой другой символ. Это можно определить опытным путем и расположить символы, как они расположены на клавиатуре.

Далее нужно указать к каким выводам на плате подключаем управление строками и столбцами.

В функции *voidsetup* указываем скорость последовательного соединения с монитором порта *9600 бод*. Функция нужна только для подачи питания на модули. В функции *VoidLoop* прописываем условие. Переменная *Char* используется для хранения только одного символа, например, 1, А или 5, что подходит к ситуации. Если нажатие зафиксировано, то происходит вывод символа в монитор порта с помощью функции *SerialPrint.* В скобках нужно указывать, какую переменную выводим в порт.

Если все сделано верно, в мониторе порта получим символ, на который нажимали. Не забудьте в мониторе порта внизу справа указать скорость передачи данных такую же, как в скетче.

Задания

Задание 1. Провтестируйте скетч, убедитесь в правильности работы.

Задание 2. Подключите светодиод (2, 3 практические работы), доработайте скетч, чтобы при нажатии клавиши с номером 8 светодиод загорался.

Задание 3. Доработайте скетч, чтобы светодиод выключался по нажатию клавиши с номером 9.

Задание 4. Отключите от микросхемы выводы, отвечающие в клавиатуре за знаки и буквы. Подключите максимальное возможное количество светодиодов. Осуществите включение каждого из них «своей» клавишей на клавиатуре.

Указания:

Проверка нажатия клавиши с номером 8.

```
if (customKey=='8') {digitalWrite(3, HIGH); … }
```

Вариант с использованием case:

```
switch (customKey)
{

case '1':
digitalWrite(3, HIGH);
break;
case '2':
digitalWrite(3, LOW);
break;}
```

Оператор *break* служит командой выхода из оператора *Case*.

Использованные источники

1. **Артур Сальбах. Arduino для начинающих. Урок 10. Подключение матричной клавиатуры и интересные схемы. 04.03.2017 http://edurobots.ru/2017/03/arduino-keypad/**

2. Урок 13. Подключение клавиатуры 4*4 и пьезоизлучателя к Arduino https://lesson.iarduino.ru/page/urok-13-podklyuchenie-klaviatury-4-4-i-pezoizluchatelya-k-arduino

3. **NikitaKitov.Ардуино: подключение матричной клавиатуры**

   https://robotclass.ru/tutorials/matrix_keyboard/

Лабораторная работа 8. Программирование контроллера Arduino. Подключение дисплея

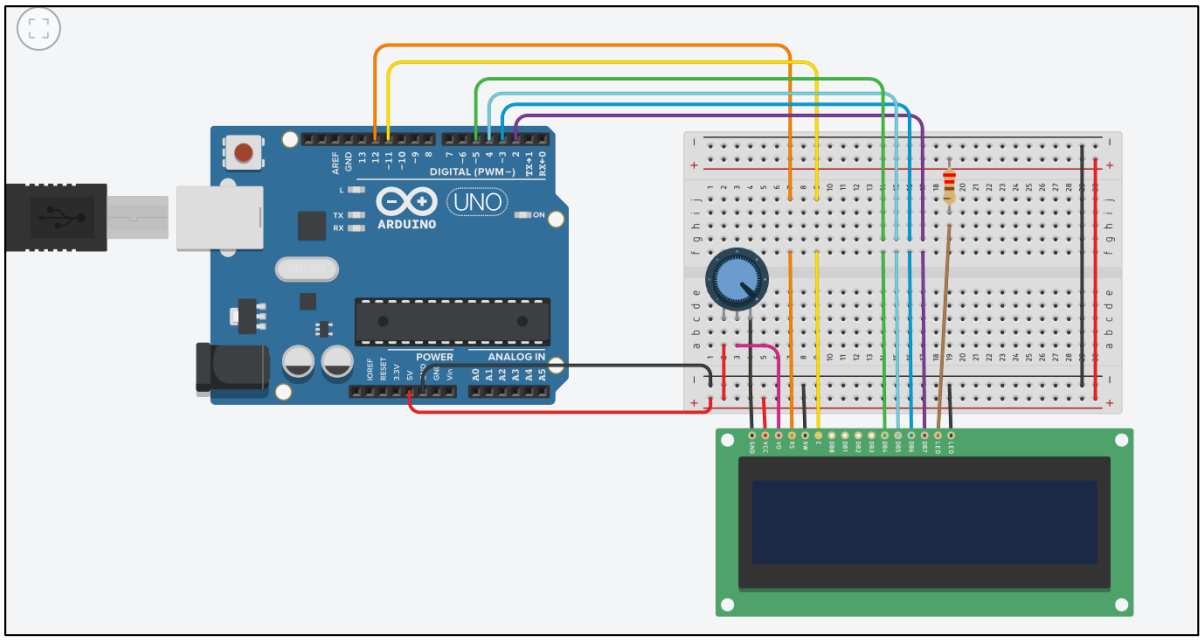
Цель: научиться подключать дисплей, использовать его совместно с клавиатурой.

Теоретические сведения

Подключение символьного ЖК дисплея 1602

У дисплея 1602 есть 16 выводов. Обычно они нумеруются слева-направо, если смотреть на него так как на картинке. Иногда выводы подписываются, типа: DB0, DB1, EN и т.п. А иногда просто указывают номер вывода. В любом случае, список выводов всегда одинаковый:

1 — «GND», земля (минус питания);
2 — «Vcc», питание +5В;
3 — «VEE», контраст;
4 — «RS», выбор регистра;
5 — «R/W», направление передачи данных (запись/чтение);
6 — «EN», синхронизация;
7-14 — «DB0», «DB1», .., «DB7″- шина данных;
15 — анод подсветки (+5В);
16 — катод подсветки (земля).



Для работы с ЖК дисплеями различных размеров и типов, в редакторе Arduino IDE имеется специальная библиотека **LiquidCrystal**. Чтобы подключить библиотеку, запишем первой строчкой нашей программы следующее выражение:

```
#include<LiquidCrystal.h>
```

Далее, нам потребуется указать какие выводы Ардуино мы использовали для подключения дисплея. Этуинформациюмыукажемпринициализациимодуля:

```
// initialize the library with the numbers of the interface pins

LiquidCrystallcd(12, 11, 5, 4, 3, 2);
```

Здесь первые два аргумента — это выводы RS и EN, а оставшиеся четыре – линии шины данных DB4-DB7.

Далее, укажем размер дисплея с помощью команды «begin». Напоминаю, в нашем дисплее имеется две строки, по 16 символов в каждой.

Наконец, для вывода текста нам понадобится простая функция «print». Вывод с помощью этой функции всем известной фразы будет выглядеть следующим образом:

```
void setup() {

  // set up the LCD's number of columns and rows:

lcd.begin(16, 2);

  // Print a message to the LCD.

lcd.print("hello, world!");

}
```

Управлениеяркостью

```
void loop() {

  // set the cursor to column 0, line 1

  // (note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(0, 1);

  // print the number of seconds since reset:

lcd.print(millis() / 1000);

}
```

Задание для самостоятельной работы

Задание 1. Составьте таблицу подключения клавиатуры к контроллеру (какие выводы к каким контактам подключаются).

Задание 2. Протестируйте схему подключения дисплея.

Задание 3. Подключите клавиатуру к системе. Так как свободных пинов мало, то нужно подключить часть клавиатуру 3x3 (числа от 1 до 9), оставив 2 контакта свободными. Используя материалы предыдущей работы отредактируйте скетч так, чтобы цифра с нажатой клавиши на клавиатуре отображалась на мониторе.

Задание 4. Определите, является ли введённое число чётным или нечётным. Результат отобразите на клавиатуре.

Полезные ссылки:

1. Олег Евсегнеев «Ардуино: вывод текста на ЖК дисплей 1602» https://robotclass.ru/tutorials/arduino-lcd1602/
2. Артур СальбахArduino для начинающих. Урок 10. Подключение матричной клавиатуры и интересные схемы http://edurobots.ru/2017/03/arduino-keypad/

Практическая работа 8. Измерение температуры через TMP36 Ардуино в Тинкеркаде

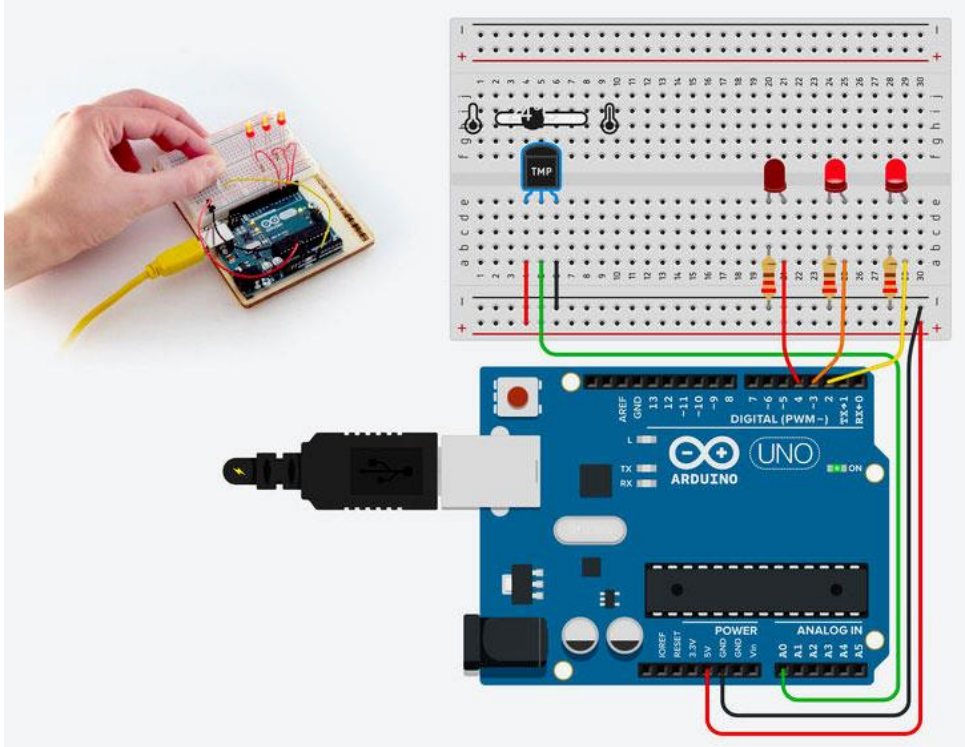
Цель: познакомиться с порядком подключение и приёмами работы температурного датчика.

Задание

Задание 1. По материалам статьи

https://arduinoplus.ru/izmerjaem-temperaturu-kozhi-cherez-tmp36-s-arduino-v-tinkerkade/

смоделируйте схему по примеру и организуйте работу температурного датчика.



Задание 2 (самостоятельное). В полученную схему добавьте дисплей и добейтесь показа температуры на экране.

Оформите отчёт по сделанной работе.

Лабораторная работа 9. Сервопривод. Терменвокс.

Задача 1. Сервопривод.

1.1. Из набора имеющихся компонентов соберите схему, которая реализует плавный поворот сервопривода с 0 до 180 градусов и обратно (Проект 1.1).

1.2. Подключите ультразвуковой дальномер и реализуйте связь между значениями с дальномера и положением сервопривода (Проект 1.2).

1.3. Реализуйте отправку информации о ходе выполнения программы на монитор порта (Проект 1.2).

1.4. Создайте простейшую сигнализацию с использованием динамика при приближении объекта.

1.5. Подключите к схеме экран. Выведите информацию на него о ходе выполнения программы. (Проект 1.2).

Задача 2. Терменвокс.

Спроектируйте музыкальный инструмент терменвокс, с помощью которого можно менять высоту звучания бесконтактным путем, больше или меньше закрывая от света фоторезистор.

Оригинальный инструмент был изобретён ещё в 1920 году, Львом Сергеевичем Терменом.

2.1. Реализуйте устройство в соответствии с принципиальной схемой (Проект 2.1).

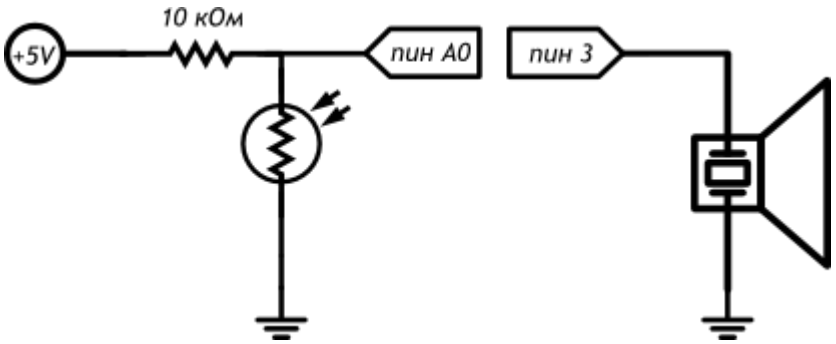


Рисунок 1 – Принципиальная схема **Терменвокса**

Напишите скетч, который считываетуровень освещённости в виде значения от 0 до 1023.

Рассчитайте частоту звучания пищалки в герцах,используя функцию проекции (англ. map). Она отображаетзначение из одного диапазона на другой, строя пропорцию.Частоту от 3,5 до 4,5 кГц можно получить, отображая [0; 1023] на [3500; 4500].

Сгенерируйте звук на заданной высоте в течение 20 миллисекунд.

2.2. Уберите из программы чтение датчика освещенности и пропищите азбукой Морзе позывной SOS: три точки, три тире, три точки (Проект 2.2).

2.3. 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 (Проект 2.3).

Ларабота 11.

Создание мини-проекта с микроконтроллером Arduino

Цель: познакомиться с процессом разработки проекта, реализовать его в эмуляторе и проверить действие, выполнить описание проекта по плану.

Задание

Для разработки проекта выберите любую тему из предложенного списка. Можно предложить свою. При выборе темы убедитесь, что возможности эмулятора допускают реализацию проектируемого устройства, то есть проверьте наличие компонентов и библиотек. После выбора темы зафиксируйте её на общем форуме в журнале «Микропроцессорные системы» для того, чтобы темы не повторялись

Темы проектов

1. Кодовый замок
2. Использование датчика движения
3. Мини-пианино
4. Часы реального времени
5. RGB-светодиод. Широтно-импульсная модуляция. Переливаемся цветами радуги
6. Счётчик нажатий
7. Метеостанция
8. Светильник, управляемый по USB
9. Игра. Перетягивание каната (кто из игроков быстрее 20 раз нажмёт на кнопку)
10. Использование двигателя
11. Простая система безопасности на Arduino своими руками
12. Управление шлагбаумом
13. Световой будильник

Найдите информацию о проекте, реализуйте его в эмуляторе, проверьте действие.

Опишите проект по следующему плану.

1. Назначение
2. Актуальность
3. Необходимые детали
4. Схема подключение деталей (описание, комментарии)
5. Скрин с эмулятора с подключенными деталями
6. Используемые библиотеки
7. Скетч с комментариями
8. Описание работы (функционирования)
9. Тестирование
10. Видео экрана с реализацией
11. Вывод