

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Северо-Кавказский федеральный университет»**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по дисциплине**

«Промышленный интернет»

Направление подготовки 09.04.02 «Информационные системы и технологии»

Направленность (профиль) «Управление данными»

Ставрополь
2026

ВВЕДЕНИЕ

Цель преподавания дисциплины «Промышленный интернет» состоит в получении теоретических знаний, практических навыков и компетенций, необходимых для проведения исследований в проектировании сетей Промышленного интернет (ПИ) – основы построения современных систем домашней автоматизации, автоматизации зданий, промышленной автоматизации, охраны, персональной медицины, энергетики.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-1: Способен оценивать современные достижения, усваивать и перерабатывать научные методы и способы деятельности для получения новых знаний и решения задач прикладного, междисциплинарного и межатраслевого характера	ИД-1 _{ОПК-1} Приобретает и адаптирует новые знания междисциплинарного и межатраслевого характера. ИД-2 _{ОПК-1} Оценивает современные достижения в области новых знаний при решении задач прикладного, междисциплинарного и межатраслевого характера. ИД-3 _{ОПК-1} Владеет способностью усваивать и перерабатывать научные методы и способы деятельности для решения задач прикладного, междисциплинарного и межатраслевого характера.	Сформированы умения и навыки определения полноты информации, степень ее адекватности для решения проблемной ситуации, проведения анализа проблемной ситуации как систему, выявляя ее составляющие и связи между ними, оценки надежности источников информации, выработки стратегии действий для решения проблемной ситуации. Имеются знания о методах работы с противоречивой информацией из разных источников. Выполняет обобщение, структурирование и критический анализ профессиональной информации, анализирует профессиональную информацию, делает выводы и рекомендации на основе анализа профессиональной информации

ЗАНЯТИЕ 1. ОСНОВЫ РАЗРАБОТКИ ПРОЕКТОВ С ИСПОЛЬЗОВАНИЕМ ARDUINO

Цель и содержание

Цель работы: изучение основных механизмов, оборудования, технических средств, основ микроконтроллерных устройств и получение практических навыков работы над проектами IoT.

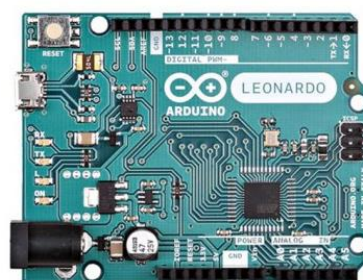
В рамках лабораторной работы необходимо решить следующие задачи:

- изучить принципы работы со средой Arduino IDE;
- научиться подключать плату Arduino;
- научиться загружать, обновлять скетчи Arduino;
- изучить основные возможности плат Arduino.

Теоретическая часть

На современном этапе развития аппаратной част IoT существует большое количество различных микроконтроллеров, на базе которых возможно построение проектов IoT и IIoT. В данном курсе предполагается использование контроллера Arduino, но даже ограничив круг микроконтроллеров до одного (Arduino), у разработчика остается широкий спектр аппаратных платформ для реализации проектов промышленного интернета и интернета вещей.

На рисунке 1.1 представлены различные платы, которые объединены общим названием Arduino.



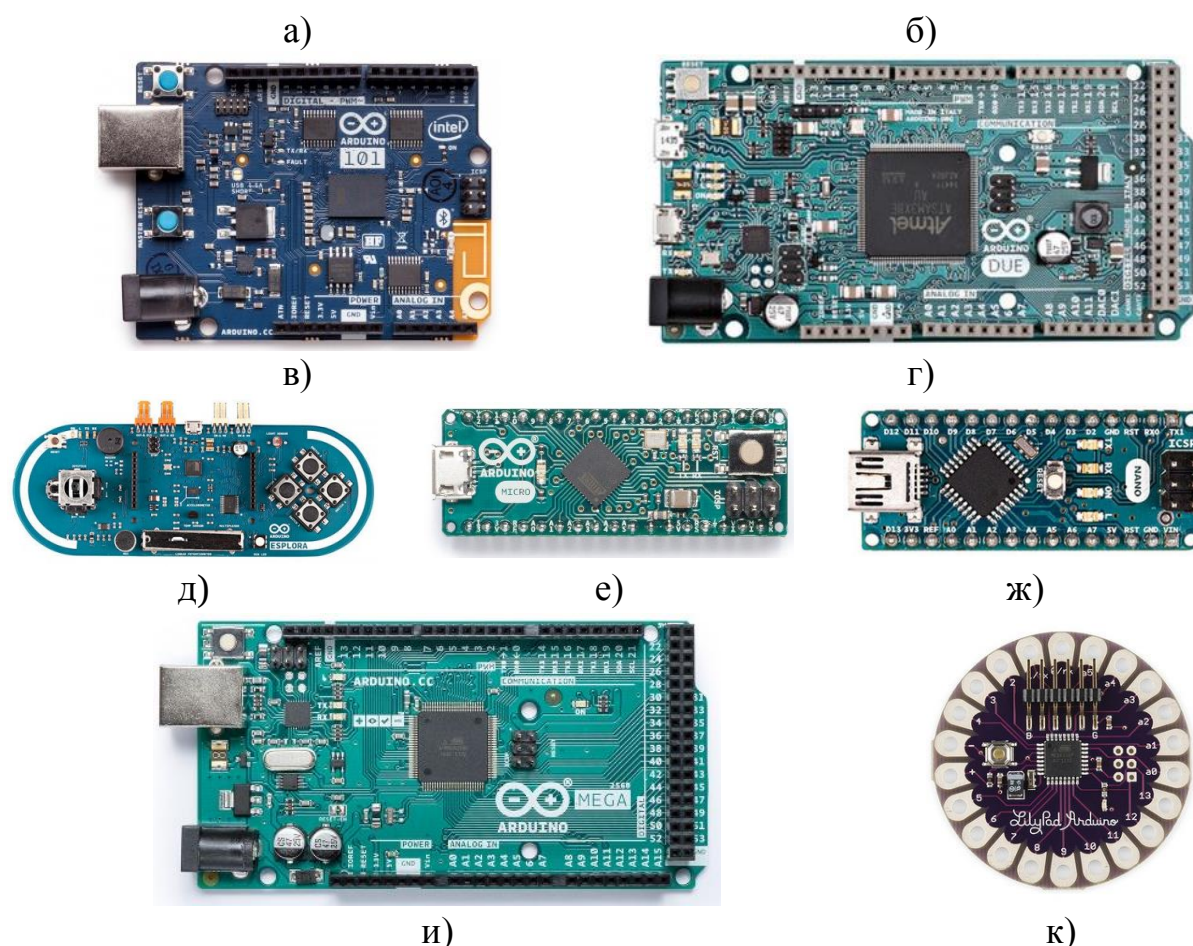


Рисунок 1.1 – Контроллеры Arduino: а) Arduino UNO; б) Arduino Leonardo; в) Arduino 101; г) Arduino Due; д) Arduino Esplora; е) Arduino Micro; ж) Arduino Nano; и) Arduino Mega 2560 Rev3; к) Arduino LilyPad

На плате Leonardo установлен контроллер ATmega32u4 со встроенным интерфейсом USB. Это уменьшает стоимость изделия и дает возможность использовать плату в качестве USB-устройства, например как эмулятор джойстика или клавиатуры. На плате Arduino Mega2560 установлен контроллер ATmega2560, имеющий 54 цифровых входа-выхода, что позволяет подключать еще больше устройств. У Arduino Mega 2560 увеличено число аналоговых входов и последовательных портов (четыре против одного у Arduino Uno). В отличие от остальных плат Arduino, использующих 8-разрядные контроллеры AVR, плата Due (рис. 1.6) создана на базе 32-разрядного процессора Atmel SAM3X8E ARM Cortex-M3 с тактовой частотой 84 МГц. Отличительные особенности платы: повышенная точность аналого-цифрового преобразователя, настраиваемая частота сигнала ШИМ, отдельные выводы циф-

роаналогового преобразователя, наличие встроенного последовательного порта.

Конструкция миниатюрной платы Arduino Nano такова, что ее можно установить в панельку для микросхем. Плата Mega ADK (рис. 1.8) очень похожа на Arduino Mega 2560, но у Mega ADK есть дополнительная функциональность интерфейса USB, позволяющая ему соединяться с телефоном на базе Android.

Уникальность платы Arduino LilyPad в том, что она разработана как часть одежды. Ее можно вшить в ткань вместе с датчиками, светодиодами. Для программирования платы необходим кабель FTDI.

Для определенности будем ориентироваться на наиболее распространённый контроллер Arduino UNO, который характеризуется широким распространением (особенно при прототипировании устройств), развитым сообществом разработчиков.

Arduino Uno (рис. 1.2) – основная плата линейки Arduino, она будет использоваться в большинстве проектов данного курса. Плата укомплектована микроконтроллером ATmega328 и микросхемой 16U2 преобразователя USB. Микроконтроллер ATmega328 может быть выполнен в исполнении DIP или SMD.

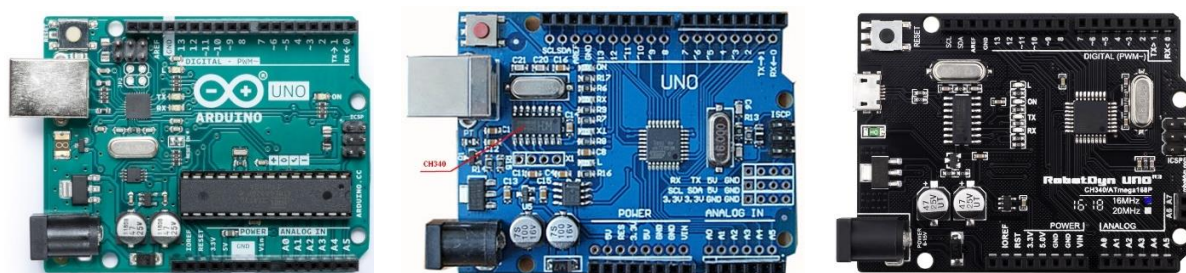


Рисунок 1.2 – Arduino UNO и так называемые клоны Arduino

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Постановка задачи: запрограммировать учебный скетч мигания встроенным светодиодом с использованием Arduino UNO; загрузить скетч на плату; научиться обновлять скетч.

1. Зайдите на официальный сайт Arduino <http://www.arduino.cc> и загрузите последнюю версию Arduino IDE со страницы Download.
2. Запустите среду программирования Arduino IDE, подключите плату Arduino UNO, выберите требуемый COM-порт в среде разработки (рисунок 1.3а).

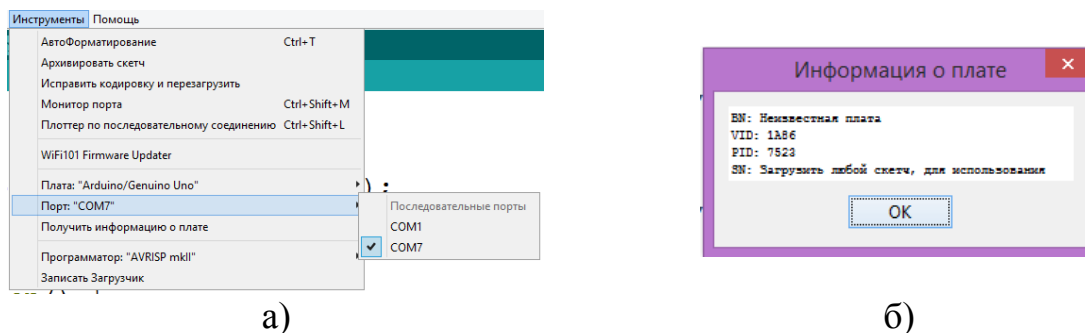


Рисунок 1.3 – Выбор порта в среде программирования

3. Выберите команду Инструменты → Информация о плате. Если подключение выполнено правильно, то отобразится информация о плате (рисунок 1.3б).

4. Загрузите готовый скетч мигания встроенным светодиодом выбрав пункт меню: Файл → Примеры → 01. Basics → Blink. После загрузки скетча в среду программирования (рисунок 1.4) проверьте скетч и загрузите его на плату.

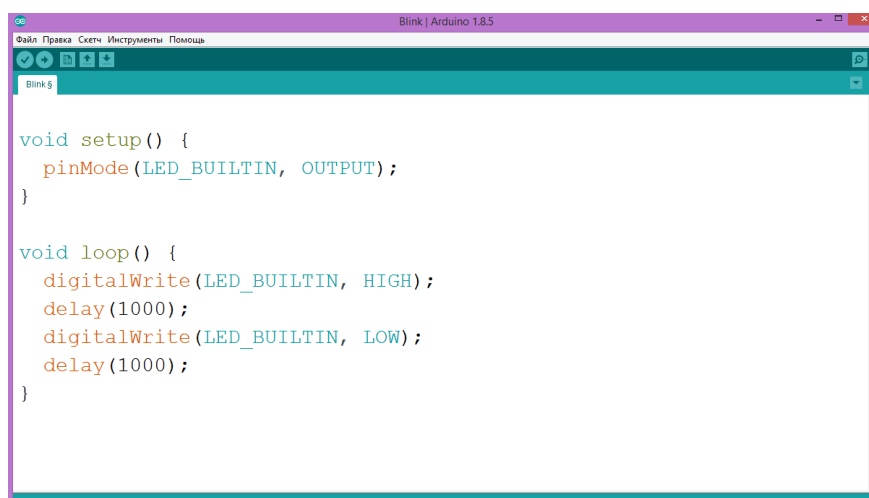


Рисунок 1.4 – Внешний вид среды программирования

После загрузки скетча встроенный светодиод должен начать мигать.

Выполнение индивидуального задания

Для выполнения индивидуального задания необходимо модернизировать скетч Blink в соответствии с вариантом индивидуального задания, загрузить его на плату Arduino, продемонстрировать работающую плату с выпол-

ненным заданием. Для заданий, содержащих использование азбуки Морзе, на рисунке 1.5 представлена кодировка символов.

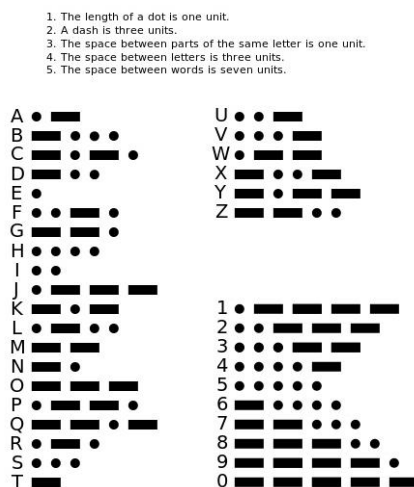


Рисунок 1.5 – International Morse Code

Таблица 1.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Светодиод мигает с равномерно нарастающей частотой: от 1 вспышки в 5 секунд до 2 вспышек в секунду.
2	Светодиод циклически подает сигнал SOS азбукой Морзе (рисунок 1.5).
3	Встроенный светодиод мигает с равномерно убывающей частотой: от 2 вспышек в секунду до 1 вспышки в 5 секунд.
4	Встроенный светодиод циклически мигает по следующему закону $(i(n), t(n))$, где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (мс): $i(n) = \{\text{HIGH, LOW, HIGH, LOW, HIGH, LOW}\}$, $t(n) = \{500, 2000, 500, 2000, 1000, 3000\}$.
5	Встроенный светодиод циклически мигает по следующему закону $(i(n), t(n))$, где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH, LOW, HIGH, LOW, HIGH, LOW}\}$, $t(n) = \{1, 2, 2, 2, 3, 2\}$.
6	Встроенный светодиод циклически мигает по следующему закону $(i(n), t(n))$, где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH, LOW, HIGH, LOW, HIGH, LOW}\}$, $t(n) = \{1, 0.5, 2, 0.5, 3, 0.5\}$.

7	Встроенный светодиод циклически мигает по следующему закону ($i(n)$, $t(n)$), где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH}, \text{LOW}\}$, $t(n) = \{2, 0.5\}$.
8	Встроенный светодиод циклически мигает по следующему закону ($i(n)$, $t(n)$), где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}\}$, $t(n) = \{0.5, 0.5, 1, 0.5, 2, 0.5, 3, 0.5\}$.
9	Встроенный светодиод циклически мигает по следующему закону ($i(n)$, $t(n)$), где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}\}$, $t(n) = \{6, 1, 5, 1, 4, 1, 3, 1\}$.
10	Встроенный светодиод циклически мигает по следующему закону ($i(n)$, $t(n)$), где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}\}$, $t(n) = \{0.5, 0.5, 0.5, 0.5, 0.5, 3\}$.
11	Встроенный светодиод циклически мигает по следующему закону ($i(n)$, $t(n)$), где n – номер периода; i – величина сигнала (LOW/HIGH); t – длина периода (с): $i(n) = \{\text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}, \text{HIGH}, \text{LOW}\}$, $t(n) = \{0.5, 1, 0.5, 1, 0.5, 3\}$.
12	Реализовать циклическое мигание светодиода для вывода текущей даты азбукой Морзе.
13	Реализовать циклическое мигание светодиода для вывода своего имени азбукой Морзе.
14	Реализовать циклическое мигание светодиода для вывода названия текущего месяца азбукой Морзе.
15	Реализовать циклическое мигание светодиода для вывода текущей даты азбукой Морзе.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Какие типы контроллеров Arduino существуют на рынке? Что такое клоны Arduino?

2. Опишите основные шаги необходимые для организации рабочего места программиста, занимающегося разработкой проектов IoT на базе Arduino.

3. Какие среды разработки можно использовать для программирования скетчей Arduino? Какая IDE наиболее удобна, поясните свое мнение.

4. Назовите цены на официальные платы Arduino и их клоны. Рассчитайте приблизительную стоимость проекта с использованием допустим 5 микроконтроллеров Arduino Nano, 10 микроконтроллеров LiLyPad, 3 контроллеров Arduino Uno.

5. Поясните смысл терминов IoT, IIoT.

6. Назовите наиболее производительную плату семейства Arduino.

7. Какие аппаратные семейства микроконтроллеров, кроме Arduino, существуют на рынке?

8. Для каких проектов можно использовать контроллеры Arduino? Приведите примеры, оцените нехватка каких знаний, умений и навыков не позволит реализовать эти проекты на текущем этапе изучения дисциплины.

9. Поясните различия в функциональном назначении плат Arduino UNO, Arduino Pro Mini и Arduino Mega.

10. Опишите назначение и синтаксис следующих функций, используемых при программировании микроконтроллеров Arduino: `delay()`, `digitalWrite()`, `pinMode()`, `setup()`, `loop()`. Опишите параметры этих функций, их назначение и допустимые значения.

Список литературы

Для выполнения лабораторной работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7]

ЗАНЯТИЕ 2. ИСПОЛЬЗОВАНИЕ ЦИФРОВЫХ ВХОДОВ

Цель и содержание

Цель работы: научиться разрабатывать скетчи для Arduino на основе чтения и записи цифровых входов платы.

В рамках лабораторной работы необходимо решить следующие задачи:

- изучить назначение и состав цифровых входов (пинов);
- научиться использовать функции `digitalWrite( )` и `digitalRead( )`;
- научиться создавать простейшие устройства с применением цифровых входов.

Теоретическая часть

Выводы платформы Arduino могут работать как входы или как выходы. Также аналоговые входы Arduino (АТmega) могут конфигурироваться и работать так же, как и цифровые порты ввода/вывода.

Выводы Arduino настроены как порты ввода, поэтому не требуется декларации в функции `pinMode( )`. Сконфигурированные порты ввода находятся в высокоимпедансном состоянии. Это означает, что порт ввода дает слишком малую нагрузку на схему, в которую он включен. Для перевода порта ввода из одного состояния в другое требуется маленькое значение тока. Если к выводу ничего не подключено, то значения на нем будут принимать случайные величины, наводимые электрическими помехами.

Если на порт ввода не поступает сигнал, то рекомендуется задать порту известное состояние. Это делается добавлением подтягивающих резисторов 10 кОм, подключающих вход либо к питанию +5 В, либо к земле. Микроконтроллер АТmega имеет программируемые встроенные подтягивающие резисторы 20 кОм. Программирование данных резисторов осуществляется так:

```
pinMode(pin, INPUT); // назначить выводу порт ввода  
digitalWrite(pin, HIGH); // включить подтягивающий резистор
```

Выводы, сконфигурированные как порты вывода находятся в низкоимпедансном состоянии. Данные выводы могут пропускать через себя достаточно большой ток. Выводы микросхемы ATmega могут быть источником тока до 40 мА. Такого значения тока недостаточно для большинства реле, соленоидов и двигателей. Короткие замыкания выводов Arduino или попытки подключить энергоемкие устройства могут повредить выходные транзисторы вывода или весь микроконтроллер ATmega.

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk Tinkercad (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Постановка задачи: спроектировать схему, написать скетч и продемонстрировать работу устройства на основе Arduino, которое меняет режим мигания светодиодом в ответ на нажатие тактовой кнопки.

1. Схема подключения компонентов, требуемых для решения задачи, показана на рис. 2.1.

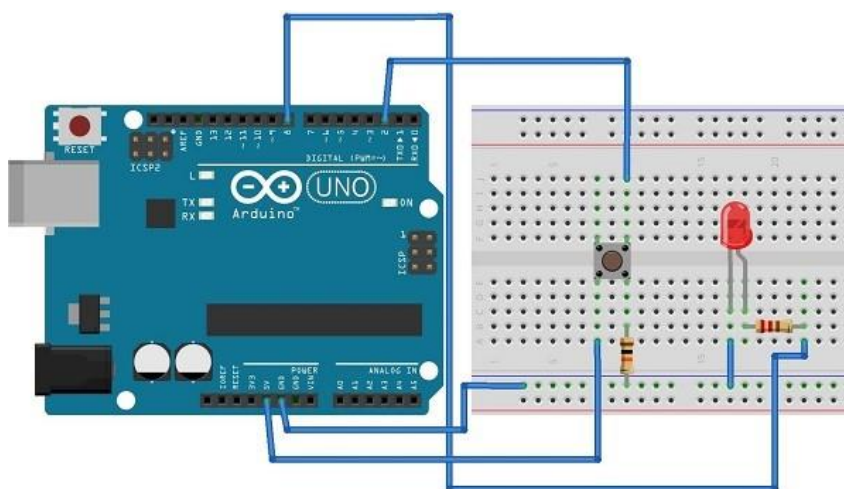


Рисунок 2.1 – Схема подключения электронных компонентов

2. Скетч для загрузки:

```
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);  
    }  
}
```

```
}
```

3. Загрузите скетч в виртуальную среду или на физическое устройство, проверьте работоспособность.

4. Измените исходный код функции loop() следующим образом:

```
void loop(){  
  if (digitalRead(button) == HIGH) {  
    if (current_mode == 0){  
      current_mode = 1;  
      digitalWrite(led, HIGH);  
      delay(2000);  
    }  
    else{  
      current_mode = 0;  
      digitalWrite(led, LOW);  
      delay(2000);  
    }  
  }  
}
```

5. Сравните режим работы двух реализованных скетчей.

Выполнение индивидуального задания

Для выполнения индивидуального задания необходимо модернизировать скетчи, представленные в данной лабораторной работе в соответствии с вариантом индивидуального задания, загрузить его на плату Arduino, продемонстрировать работающую плату с выполненным заданием. Для заданий, содержащих использование азбуки Морзе, на рисунке 2.2 представлена кодировка символов.

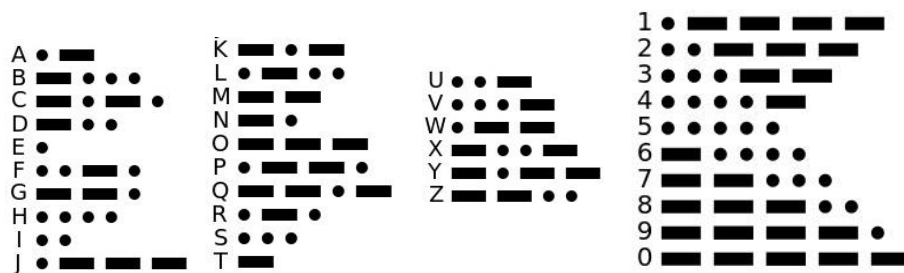


Рисунок 2.2 – International Morse Code

Следует обратить внимание на необходимость решения задачи разделения кодовых комбинаций отдельных символов.

Таблица 2.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит букву «К», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
2	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «FOG», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
3	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «САТ», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
4	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «E1000», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.

5	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «KIT», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
6	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «EASY», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
7	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «WOO», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
8	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «DOG», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
9	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «BALL», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.

10	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «SNOW», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
11	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «CAN», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
12	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «DO», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
13	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «SEE», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
14	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «ERA», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.

15	Пользователь должен ввести пароль с использованием тактовой кнопки (пароль вводится азбукой Морзе). Если пользователь вводит слово «ВЕЕ», то загорается светодиод. Светодиод горит 5 секунд, потом гаснет, то есть устройство вновь готово к вводу пароля. Необходимо самостоятельно продумать стартовую последовательность для сброса пароля. Например, если пользователь вводит «Е», то светодиод мигает 3 раза, то есть оповещает пользователя о том, что устройство готово принимать новый пароль.
----	--

Содержание отчета и его форма

Отчет по работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Что такое цифровой вход микроконтроллера? Сколько цифровых входов имеет Arduino UNO?
2. Поясните назначение функции `pinMode( )`. Опишите входные параметры данной функции.
3. Поясните назначение функции `digitalWrite( )`. Опишите входные параметры данной функции.

4. Поясните назначение функции `digitalRead()`. Опишите входные параметры данной функции.
5. Для какой цели при работе с электронным и компонентами применяются подтаивающие резисторы?
6. Поясните, каким образом можно узнать номинал резистора по его внешнему виду?
7. Назовите электронные компоненты, которые могут осуществлять работу с Arduino через цифровые входы.
8. Что такое тактовая кнопка? Поясните отличие в функциональном поведении тактовой кнопки от кнопки с фиксацией (типа ON/OFF).
9. Чем отличаются (и отличается ли вообще) устройства на этапе прототипирования и на этапе реального развертывания? Какие аппаратные решения применяются при каждом из режимов работы?
10. Оцените стоимость решения, реализованного в данной работе (светодиод, подтягивающие резисторы, плата Arduino, соединители, тактовая кнопка).

Список литературы

Для выполнения лабораторной работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 3. АНАЛОГОВЫЕ ВХОДЫ И ВЗАИМОДЕЙСТВИЕ С КОМПЬЮТЕРОМ

Цель и содержание

Цель работы: изучить теоретические основы взаимодействия устройств на основе микроконтроллеров с компьютером.

В рамках лабораторной работы необходимо решить следующие задачи:

- научиться использовать аналоговые входы;
- научиться использовать библиотеку Serial;
- научиться осуществлять мониторинг выполнения программы, реализовывать отладочные процедуры;
- научиться использовать последовательный порт для решения практических задач.

Теоретическая часть

Микроконтроллеры ATmega, используемые в Arduino, содержат шестиканальный аналого-цифровой преобразователь (АЦП). Разрешение преобразователя составляет 10 битов, что позволяет на выходе получать значения от 0 до 1023. Аналоговые входы могут использоваться как цифровые выводы портов ввода/вывода, при этом они имеют номера от 14 до 19:

```
pinMode(14,OUTPUT);
```

```
digitalWrite(14, HIGH);
```

Для вывода, работавшего ранее как цифровой порт вывода, команда `analogRead( )` будет работать некорректно. В этом случае рекомендуется сконфигурировать его как аналоговый вход.

Широтно-импульсная модуляция (ШИМ) – это операция получения изменяющегося аналогового значения посредством цифровых устройств. Подавая на выход сигнал, состоящий из высоких и низких уровней, мы модели-

руем напряжение между максимальным значением (5 В) и минимальным (0 В). Длительность включения максимального значения называется шириной импульса. Для получения различных аналоговых величин изменяется ширина импульса. В результате на выходе будет величина напряжения, равная площади под импульсами (рис. 3.1).

Вызов функции `analogWrite( )` с масштабом 0 – 255 означает, что значение `analogWrite(255)` будет соответствовать 5 В (100% рабочий цикл – постоянное включение 5 В), а значение `analogWrite(127)` – 2,5 В (50% рабочий цикл).

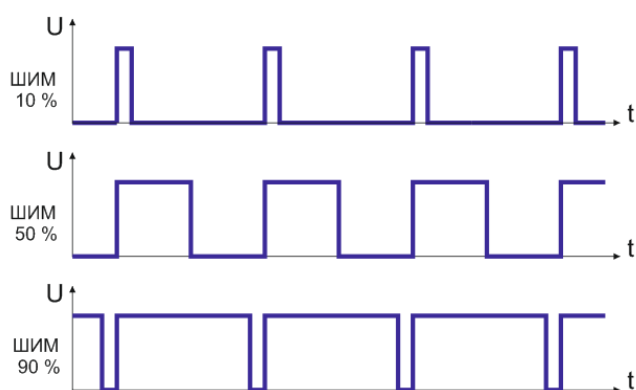


Рисунок 3.1 – Процентное измерение коэффициента заполнения ШИМ

На этапе прототипирования может оказаться важным визуализировать значения, получаемые на входах Arduino. Для этого широко применяется библиотека `Serial`. Набор функций `Serial` служит для связи устройства Arduino с компьютером или другими устройствами, поддерживающими последовательный интерфейс обмена данными. Все платы Arduino имеют хотя бы один последовательный порт UART.

Для обмена данными `Serial` служат цифровые порты ввода/вывода 0 (RX) и 1 (TX), а также порт USB. Важно учитывать, что если вы используете функции `Serial`, то нельзя одновременно с этим задействовать порты 0 и 1 для других целей.

Среда разработки Arduino имеет встроенный монитор последовательного интерфейса (`Serial monitor`). Для начала обмена данными необходимо

запустить монитор нажатием кнопки Serial monitor и выставить ту же скорость связи (baud rate), с которой вызвана функция begin().

Функция Serial.begin() иницирует последовательное соединение и задает скорость передачи данных в бит/с (бод). Для обмена данными с компьютером используйте следующие значения: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 или 115200. При соединении через порты входа/выхода 0 и 1 могут быть использованы другие значения скорости, требуемые устройством, с которым будет осуществляться обмен данными.

Синтаксис функции Serial.begin():

Serial.begin(speed)

Параметр: speed – скорость в бит/с (бод) (long).

Функция Serial.end() закрывает последовательное соединение, порты RX и TX освобождаются и могут быть использованы для ввода/вывода. Синтаксис функции Serial.end():

Serial.end()

Функция Serial.available() получает количество байтов (символов), доступных для чтения из последовательного интерфейса связи. Это те байты, которые уже поступили и записаны в буфер последовательного порта. Буфер может хранить до 128 байтов. Синтаксис функции Serial.available():

Serial.available()

Функция Serial.read() считывает очередной доступный байт из буфера последовательного соединения. Синтаксис функции Serial.read():

Serial.read()

Функция Serial.flush() ожидает окончания передачи исходящих данных (до версии Arduino 1.0 функция очищала буфер последовательного соединения). Синтаксис функции Serial.flush():

Serial.flush()

Функция Serial.print() передает данные через последовательный порт как текст ASCII. Эта функция может принимать различные типы данных. С помощью второго опционального параметра можно задать базис (систему счисления) для чисел. Синтаксис функции Serial.print():

Serial.print(val)

Serial.print(val, format)

Параметры функции: `val` – данные для передачи через последовательное соединение; `format` – базис для целых чисел или количество знаков после запятой для вещественных: `BYTE`, `BIN` (двоичный), `OCT` (восьмеричный), `DEC` (десятеричный), `HEX` (шестнадцатеричный); для вещественных (дробных) чисел второй параметр задает количество знаков после запятой.

Функция `Serial.println()` передает данные через последовательный порт как текст ASCII вместе со следующим за ним символом переноса строки (символ ASCII 13 или `'\r'`) и символом новой строки (ASCII 10 или `'\n'`). Параметры и типы данных для этой функции такие же, как и для функции `Serial.print()`.

Функция `Serial.write()` передает данные как бинарный код через последовательное соединение. Данные посылаются единичным байтом или серией байтов. Для того, чтобы передать данные как символы, следует использовать другую функцию – `print()`.

Синтаксис функции `Serial.write()`:

Serial.write(val)

Serial.write(str)

Serial.print(buf, len)

Параметры: `val` – один байт; `str` – строка как серия байтов; `buf` – массив байтов; `len` – длина массива.

Функция `Serial.peek()` возвращает следующий доступный байт (символ) из буфера входящего последовательно соединения, не удаляя его из этого буфера. То есть успешный вызов этой функции вернет то же значение, что и следующий за ним вызов функции `read()`. Синтаксис функции `Serial.peek()`:

Serial.peek()

Пример использования рассмотренных функций:

```
int in=0; // переменная для хранения полученного байта
void setup() {
    // устанавливаем последовательное соединение
```

```

    Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0) { // если есть доступные данные
        in = Serial.read(); // считываем байт
        Serial.println(in, DEC); // отсылаем то, что получили
    }
}

```

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Постановка задачи: соберите схему с использованием физических компонентов и платы Arduino или в виртуальной среде и реализуйте устройство на основе потенциометра: при изменении положения потенциометра должна изменяться яркость свечения светодиода.

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. Это одна из базовых схем.

1. Схема для решения задачи представлена на рисунке 3.1.

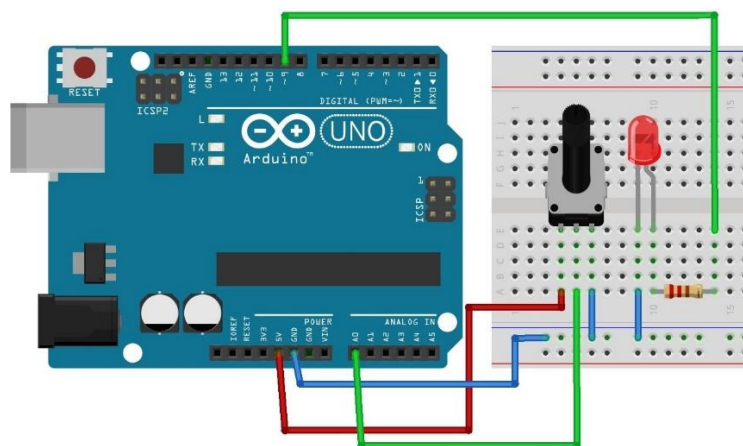


Рисунок 3.1 – Схема подключения электронных компонентов

2. Скетч для работы с представленной схемой:

```
#define led 9
#define pot A0
void setup() {
    pinMode(led, OUTPUT);
    pinMode(pot, INPUT);
}
void loop() {
    int x;
    // 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);
}

```

3. Загрузите скетч, изучите работу полученного устройства.

Выполнение индивидуального задания

Для выполнения индивидуального задания необходимо модернизировать скетч, представленный в данной работе, в соответствии с индивидуальным вариантом (таблица 3.1).

Таблица 3.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет яркостью светодиода, но в дискретном режиме, то есть при вращении ручки потенциометра яркость меняется с одинаковым шагом, скачком.
2	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет тремя светодиодами – при вращении ручки потенциометра происходит последовательное зажигание светодиодов (в каждый момент времени горит только один светодиод).
3	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет тремя светодиодами – при вращении ручки потенциометра происходит последовательное зажигание светодиодов (в каждый момент времени горят сразу несколько светодиодов, то есть чем выше напряжение на шине потенциометра, тем большее количество светодиодов зажигается).
4	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет пятью светодиодами – при вращении ручки потенциометра происходит случайное зажигание одного из светодиодов.

5	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет пятью светодиодами – при вращении ручки потенциометра происходит последовательное зажигание одного из светодиодов. Каждый светодиод зажигается на минимальной яркости, при дальнейшем вращении ручки потенциометра, данный светодиод достигает максимальной яркости и тухнет. Затем загорается следующий светодиод, функционирующий по тому же принципу.
6	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет тремя светодиодами: один светодиод меняет свою яркость плавно, два других часто мигают при достижении потенциометром минимального или максимального напряжения.
7	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет тремя светодиодами: весь возможный диапазон напряжений потенциометра делится на 3 равные части; при попадании напряжения потенциометра в один из диапазонов начинает часто мигать соответствующий светодиод.
8	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет яркостью светодиода, при этом напряжение потенциометра выводится на монитор последовательного порта.
9	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет пятью светодиодами: весь возможный диапазон напряжений потенциометра делится на 5 равных частей; при попадании напряжения потенциометра в один из диапазонов начинает часто мигать соответствующий светодиод. Необходимо осуществлять вывод номера мигающего светодиода в монитор последовательного порта.
10	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет двумя светодиодами: если напряжение на входе потенциометра ниже 3В, то мигает один светодиод, в противном случае – другой. Необходимо осуществлять вывод номера мигающего светодиода в монитор последовательного порта.
11	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет пятью светодиодами: для каждого светодиода необходимо задать пороговое напряжение (0 – 5 В). Если это пороговое напряжение достигнуто, то загорается соответствующий светодиод. Необходимо осуществлять вывод номеров горящих светодиодов в монитор последовательного порта.

12	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет тремя светодиодами: один светодиод меняет свою яркость плавно, два других часто мигают при достижении потенциометром минимального или максимального напряжения.
13	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет яркостью светодиода, но в дискретном режиме, то есть при вращении ручки потенциометра яркость меняется с одинаковым шагом, скачком. При достижении крайних положений потенциометром, светодиод начинает часто мигать.
14	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет тремя светодиодами – при вращении ручки потенциометра происходит последовательное зажигание светодиодов (в каждый момент времени горит только один светодиод).
15	Разработать электронную схему и написать скетч для решения следующей задачи: пользователь с использованием потенциометра управляет пятью светодиодами: с помощью пятью светодиодами отобразите направление вращения ручки потенциометра.

Содержание отчета и его форма

Отчет по работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Что такое потенциометр? Каков внутренний механизм работы данного электронного компонента?
2. Что такое аналоговый вход? Каковы отличия при программировании аналоговых и цифровых входов? Сколько аналоговых входов имеет Arduino UNO?
3. Поясните содержание терминов: «встроенные карты», «параключ/значение».
4. Что такое ШИМ? Как ШИМ используется при проектировании реальных устройств IoT?
5. Для чего предназначен набор функций Serial?
6. Опишите назначение функций и их параметров: `Serial.begin( )`, `Serial.end( )`, `Serial.available( )`.
7. Опишите назначение функций и их параметров: `Serial.read( )`, `Serial.flush( )`, `Serial.print( )`, `Serial.println( )`, `Serial.write( )`, `Serial.peek( )`.
8. Как при проектировании устройств на базе Arduino используется монитор последовательного порта?
9. Какие датчики по своей природе являются аналоговыми? Перечислите аналоговые датчики и опишите их назначение.
10. Какие датчики по своей природе являются цифровыми? Может ли датчик быть одновременно и аналоговым и цифровым (можно ли один и тот же датчик подключать и к аналоговым входам и к цифровым)?

Список литературы

Для выполнения лабораторной работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 4. ПРОГРАММИРОВАНИЕ СИСТЕМ НА ОСНОВЕ СЕНСОРОВ

Цель и содержание

Цель работы: изучить принципы работы с датчиками Arduino.

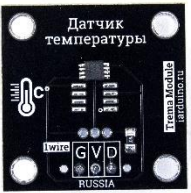
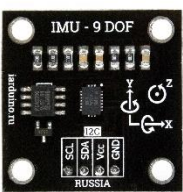
В рамках лабораторной работы необходимо решить следующие задачи:

- научиться выбирать датчик для решения практических задач;
- научиться разрабатывать схемы устройств на базе Arduino для решения задач с применением распространенных простейших датчиков;
- научиться обрабатывать сигналы с датчиков различного типа, а также обеспечивать реакцию системы на подобные события.

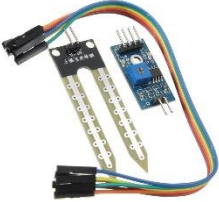
Теоретическая часть

При работе с устройствами на базе Arduino разработчику доступно большое количество различных датчиков (таблица 4.1).

Таблица 4.1 – Распространенные датчики, совместимые с Arduino

Датчик	Описание
	Трета-модуль Цифровой термометр – измеряет температуру окружающей среды.
	Трета-модуль IMU-сенсор 9 DOF (9 степеней свободы) – это модуль, позволяющий определять своё положение в пространстве. Помимо положения, с модуля можно получать следующие данные: угловое ускорение (проекция кажущегося ускорения), истинное ускорение, угловую скорость, индукцию магнитного поля и температуру. Все параметры, кроме температуры, выводятся для трёх осей X, Y и Z. Положение и направление осей указано на плате модуля.

 Barometer module (Barometr) with a black PCB and white components. It features a circular pressure sensor and various electronic components. The text 'Barometr' is printed at the top, and 'RUSSIA' is at the bottom.	Тема-модуль Барометр BMP280 – это цифровой модуль, подключаемый по шине I2C (адрес 0x77) и позволяющий получить текущие значения атмосферного давления и температуры окружающей среды. Еще одним применением данного модуля является определение высоты, которая зависит от давления и рассчитывается по международной барометрической формуле.
 Sound sensor module (Датчик звука) with a black PCB. It features a circular microphone and various electronic components. The text 'Датчик звука' is at the top, and 'AGVS RUSSIA' is at the bottom.	Тема-модуль Датчик звука – позволяет получить аналоговое значение соответствующее уровню громкости звука. Модуль обладает высокой чувствительностью. Сигнал на выходе модуля не повторяет форму звукового сигнала, а соответствует уровню его громкости в любой промежуток времени.
 Line sensor module (Датчик линии) with a black PCB. It features a small sensor and various electronic components. The text 'Датчик линии' is visible on the module.	Тема-модуль Датчик линии – позволяет различать светлые и темные поверхности, при неизменном расстоянии до них. А если неизменным остаётся цвет отражающей поверхности, то датчик позволяет определять расстояние до неё.
 GY-521 module with a blue PCB. It features a small sensor and various electronic components. The text 'GY-521' is visible on the module.	Модуль GY-521 на микросхеме MPU6050 – это 3-осевой гироскоп и акселерометр на три координаты. Его можно использовать для определения положения в пространстве.
 Infrared sensor module (инфракрасный датчик препятствий) with a yellow and black cable. It features a small sensor and various electronic components.	инфракрасный датчик препятствий E18-D80NK – состоит из инфракрасного излучателя и приемника, работает на расстоянии от 3 до 80 см и подойдёт для обнаружения приближающихся предметов или препятствий.
 DHT11 module with a blue PCB. It features a small sensor and various electronic components.	Датчик температуры и влажности DHT11 – позволяет получать показания температуры и влажности по одной линии данных. Модуль будет полезен в проектах климат-контроля для теплиц или помещений.
 APDS9930 module with a purple PCB. It features a small sensor and various electronic components.	Датчик освещённости и приближения APDS9930 – способен определять близость объектов (препятствий) и уровень освещённости в Lux. У старшей модели APDS9960 более расширенный функционал, помимо определения близости и освещённости, они способны определять цвет и жесты.

	Простой в использовании датчик движения, который работает в инфракрасном диапазоне. После включения выполняется инициализация – датчик делает ИК снимок пустой комнаты. Через 1-2 секунды после инициализации датчик готов к работе. Он будет сигнализировать о любом движении в охраняемой комнате.
	RC522 – RFID-модуль 13.56 МГц. Главная задача RFID-модуля RC522 – опознавание «свой – чужой». Чаще всего модуль используется как считыватель RFID-карт в системах контроля доступа. Данные модули стоят и в метрополитене, для считывания оставшихся поездок. В Arduino RFID-модулям тоже можно придумать много полезных применений.
	Дальномер HC-SR04 позволяет измерять расстояния от 2 до 400 см с точностью до 0.3 см при помощи ультразвука.
	ESP8266 с реле – Wi-Fi реле на базе ESP-01. Модуль с ESP8266 и реле предназначен для дистанционного и автоматического управления электроприборами. Реле может коммутировать как постоянный, так и переменный ток.
	KY-040 – энкодер (модуль датчика вращения). Электронный потенциометр или энкодер часто применяют для подстройки параметров устройства – он надёжнее, не ограничен в количестве оборотов и прост в интерпретации сигнала.
	Датчик влажности почвы.
	Трехпроводной цифровой мини-вольтметр 0.28" (DC 0 ~ 100V). Встраиваемый трехпроводной цифровой вольтметр способен измерять постоянный ток в диапазоне напряжений: 0 ~ 100V, при подключении внешнего питания.

	KY-008 – лазерный модуль. Модуль лазерного светодиода KY-008 может быть полезен при создании различных защитных устройств, как то сигнализаций и барьеров.
	SW-18010P – датчик вибрации. Датчик вибрации чётко детектирует наличие колебания, встряски, ударов и вибрации. Датчик вибрации воспринимает механические воздействия на прибор, в котором он установлен. Может использоваться в охранных системах. Возможно применять в движущейся робототехнике для регистрации столкновения с препятствием.
	MAX9814 – модуль электретного микрофона с усилителем. Модуль маломощного усилителя электретного микрофона на базе MAX9814 с регулируемым коэффициентом усиления. Усилитель характеризуется отличным подавлением шумов по питанию, а также возможностью ручной установки коэффициента усиления.
	Датчик дождя, капель, пара. Датчик дождя имеет чувствительный сенсор, с площади которого анализируется попавшая влага. Устройство также оснащено компаратором и имеет два выхода: аналоговый и цифровой, для максимального удобства подключения.
	KY-038 – датчик звука (шума) с микрофоном. Микрофонный датчик звука Keyes KY-038 умеет определять уровень громкости вокруг него. Он может применяться, например, для включения света по хлопку.
	Датчик пульса. Датчик пульса создан для определения человеческого пульса. Для этого достаточно приложить датчик к мочке уха или подушечке пальца. Главная особенность датчика в том, что его не нужно подстраивать под каждого человека – он подойдет всем, производя измерения быстро и точно.
	NRF24L01 SMD + PA (SMD) – модуль беспроводной связи 2.4 ГГц. nRF24l01 – модуль, уже давно ставший стандартом де-факто беспроводных коммуникаций устройств на Arduino.

	Датчик уровня воды. Датчик уровня воды поможет вам быстро и эффективно создать на Arduino устройство по контролю за аквариумом, чайником и т.д. Имеет погружаемый тип установки и снятие показателей уровня воды при помощи аналогового анализа сопротивления.
	ESP-12F – Wi-Fi модуль на базе ESP8266. Недорогой и мощный модуль на основе ESP8266 идеально подходит для IoT-проектов, где требуется компактность и беспроводная связь с внешним миром по сети Интернет.

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk Tinkercad (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Постановка задачи: реализовать схему, написать скетч для управления светодиодом посредством аналогового датчика звука KY-038. Осуществить вывод уровня звука в монитор последовательного порта.

1. На рис. 4.1 представлена схема реализованного устройства.

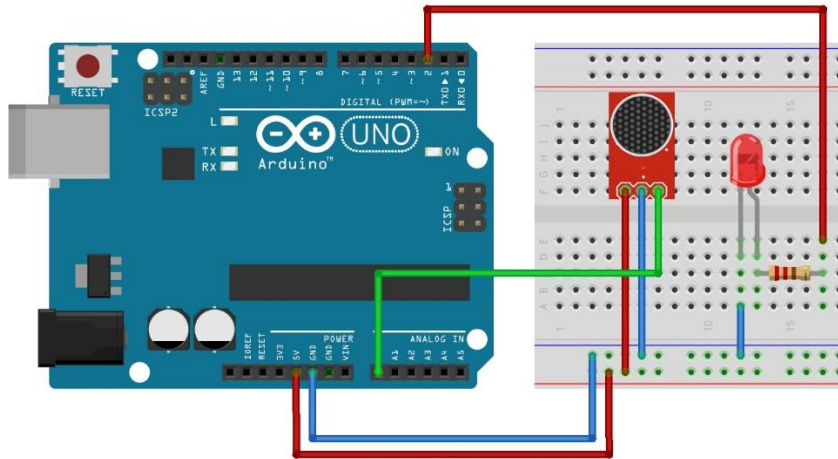


Рисунок 4.1 – Схема подключения электронных компонентов

2. Создайте программу в среде Arduino IDE и загрузите полученный скетч на плату:

```
#define led 2
#define noise A0
#define threshold 512
int x = 0;
void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  pinMode(noise, INPUT);
}
void loop() {
  x = analogRead(noise);
  if (x < threshold){
    digitalWrite(led, HIGH);
    delay(1000);
  }
}
```

```

    digitalWrite(led, LOW);
  }
  Serial.println(x);
}

```

3. Запустите устройство и проверьте работоспособность системы.

Выполнение индивидуального задания

После выполнения учебного задания необходимо выполнить задание в соответствии с индивидуальным вариантом (таблица 4.1).

Таблица 4.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Реализовать схему, написать скетч для управления светодиодом посредством датчика температуры. Осуществить вывод уровня звука в монитор последовательного порта.
2	Реализовать схему, написать скетч для управления светодиодом посредством датчика давления. Осуществить вывод уровня звука в монитор последовательного порта.
3	Реализовать схему, написать скетч для управления светодиодом посредством сенсорной кнопки. Осуществить вывод уровня звука в монитор последовательного порта.
4	Реализовать схему, написать скетч для управления светодиодом посредством цифрового датчика наклона. Осуществить вывод уровня звука в монитор последовательного порта.
5	Реализовать схему, написать скетч для управления светодиодом посредством датчика Холла. Осуществить вывод уровня звука в монитор последовательного порта.
6	Реализовать схему, написать скетч для управления светодиодом посредством датчика влажности. Осуществить вывод уровня звука в монитор последовательного порта.
7	Реализовать схему, написать скетч для управления светодиодом посредством датчика уровня воды. Осуществить вывод уровня звука в монитор последовательного порта.
8	Реализовать схему, написать скетч для управления светодиодом посредством датчика вибрации. Осуществить вывод уровня звука в монитор последовательного порта.

9	Реализовать схему, написать скетч для управления светодиодом посредством датчика CO ₂ . Осуществить вывод уровня звука в монитор последовательного порта.
10	Реализовать схему, написать скетч для управления светодиодом посредством датчика движения. Осуществить вывод уровня звука в монитор последовательного порта.
11	Реализовать схему, написать скетч для управления светодиодом посредством датчика освещенности. Осуществить вывод уровня звука в монитор последовательного порта.
12	Реализовать схему, написать скетч для управления светодиодом посредством датчика линии. Осуществить вывод уровня звука в монитор последовательного порта.
13	Реализовать схему, написать скетч для управления светодиодом посредством ультразвукового дальномера. Осуществить вывод уровня звука в монитор последовательного порта.
14	Реализовать схему, написать скетч для управления светодиодом посредством датчика температуры и кнопки. Осуществить вывод уровня звука в монитор последовательного порта.
15	Реализовать схему, написать скетч для управления светодиодом посредством датчика движения и кнопки. Осуществить вывод уровня звука в монитор последовательного порта.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Опишите основные датчики Arduino, их назначение, виды.
2. Чем отличается датчик, сенсор, плата расширения?
3. Чем отличается цифровой датчик от аналогового (например температурный)?
4. Поясните принцип работы аналогового датчика звука (например KY-038).
5. Что такое датчик Холла? Поясните принцип работы данного датчика.
6. Поясните принцип работы датчика наклона. Какие типы датчиков наклона бывают?
7. Что такое геркон? На какой датчик, по принципу работы, похож геркон?
8. По какому принципу работает инфракрасный датчик движения?
9. Какие платы расширения для Arduino направлены для организации беспроводной связи?
10. Что такое технология RFID? Какое практическое применение RFID можно предложить для применения в реальных проектах?

Список литературы

Для выполнения работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 5. ЭНЕРГОНЕЗАВИСИМАЯ ПАМЯТЬ В ПРОЕКТАХ IOT

Цель и содержание

Цель работы: научиться использовать энергонезависимую память в проектах IoT.

В рамках лабораторной работы необходимо решить следующие задачи:

- научиться подключать модули чтения карт памяти;
- научиться считывать и записывать информацию на карты памяти;
- научиться использовать данные с карты памяти для реализации вычислений.

Теоретическая часть

Есть множество плат расширения (переходников) для подключения SD-карт к Arduino. Все рассмотреть в книге невозможно, опишем некоторые из популярных, указав их достоинства и недостатки.

Общие черты всех переходников SD-карт:

- подключаются к плате Arduino по интерфейсу SPI, через 6 контактов ICSP платы Arduino или к контактам цифровых выводов (11, 12 и 13 на Uno или 50, 51 и 52 на Mega);
- имеют вход выбора (CS), который может быть контактом по умолчанию (53 на Mega, 10 на других платах Arduino);
- подают питание 3,3 В на SD-карту и согласуют логические уровни.

В рамках данной лабораторной работы будет использоваться плата расширения MicroSD Card (рисунок 5.1).

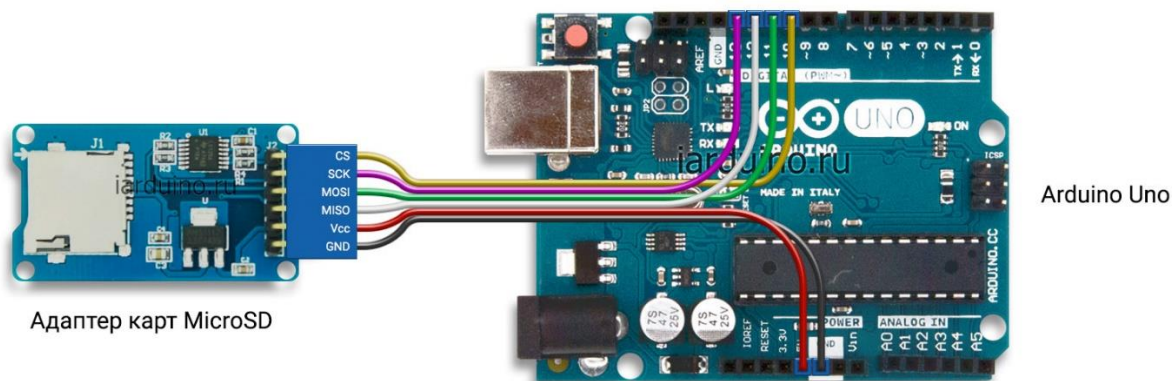


Рисунок 5.1 – Плата расширения MicroSD Card, подключенная к Arduino

Связь платы Arduino с SD-картой осуществляется через SPI-интерфейс: контакты MOSI, MISO, SCLK (тактовые импульсы) и CS (выбор чипа). Для выполнения примеров этой главы мы будем использовать библиотеку SD. Это предполагает, что задействованы аппаратные SPI-контакты на плате Arduino, а в качестве CS назначен или контакт по умолчанию или определенный пользователем. Библиотека SD для правильной работы требует установки контакта для CS по умолчанию в качестве выхода, даже если для CS задан другой вывод. В случае Uno это вывод 10, для платы Mega это вывод 53.

Библиотека SD позволит создать несложный пример записи данных на SD-карту. Будем получать данные с Serial-порта и записывать их на карту в формате csv. Данные хранятся в файле под названием data.csv, в дальнейшем его можно открыть на компьютере. Важно отметить, что при форматировании карты в FAT16 имена файлов должны быть записаны в формате "8.3", т. е. расширение должно состоять из трех символов, а имя файла – не более чем из восьми символов.

Базовые синтаксические конструкции при работе с SD-картой:

Создание переменной myFile класса File библиотеки SD, которая будет служить «указателем» на файл (переменная потребуется для открытия и закрытия файла, а так же для работы с открытым файлом):

File myFile;

Открытие файла "iarduino.txt" для чтения и записи, начиная с конца файла (если файла с именем "iarduino.txt" – нет, то он будет создан):

```
myFile = SD.open("iarduino.txt", FILE_WRITE);
```

Запись строки в файл "iarduino.txt" (запись строки в открытый файл "iarduino.txt", без добавления символов перехода на новую строку):

```
myFile.print("начало строки, ");
```

Запись строки в открытый файл "iarduino.txt", с добавлением символов перехода на новую строку:

```
myFile.println("продолжение строки.");
```

Закрытие файла "iarduino.txt":

```
myFile.close();
```

Открытие файла "iarduino.txt" для чтения, начиная с начала файла (если файла с именем "iarduino.txt" – нет, то он не будет открыт):

```
myFile = SD.open("iarduino.txt");
```

Чтение строк из файла "iarduino.txt":

```
while(myFile.available()){ // если есть данные для чтения  
    // читаем очередной байт из файла  
    // и выводим его в последовательный порт  
    Serial.write(myFile.read());  
}
```

Удаление файла "iarduino.txt" (функция возвращает результат удаления true/false):

```
SD.remove("iarduino.txt");
```

Проверка существования файла "iarduino.txt":

```
if (SD.exists("example.txt")){ Serial.write("OK"); }
```

Проверка открытия файла "iarduino.txt" (условие будет выполнено, если файл был успешно открыт):

```
if (myFile) { Serial.write("OK"); }
```

Создание каталога "my_Dir" (функция возвращает результат создания true/false):

```
SD.mkdir("my_Dir");
```

Удаление каталога "my_Dir" (функция возвращает результат удаления true/false. Удаляемый каталог должен быть пустым):

SD.rmdir("my_Dir");

Таким образом, функции работы с картой памяти достаточно простые, но посредством данной технологии, проекты Arduino обеспечиваются возможностью работать с энергонезависимой памятью.

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

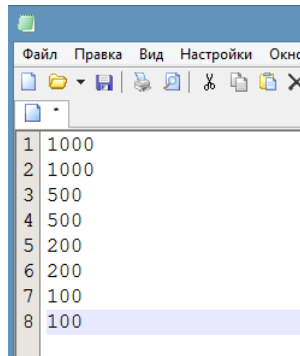
Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Постановка задачи: реализовать чтение данных из файла data.csv. После Полученные данные интерпретируются как длительность импульса мигания светодиодом.

1. Создадим файл data.csv и запишем в него следующие данные:



1	1000
2	1000
3	500
4	500
5	200
6	200
7	100
8	100

Рисунок 5.2 – Данные на MicroSD Card

2. Убедитесь, что переходник SD правильно подключен к плате Arduino и в него установлена SD-карта. Для MicroSD Shield это будет выглядеть так, как на рис. 5.3.

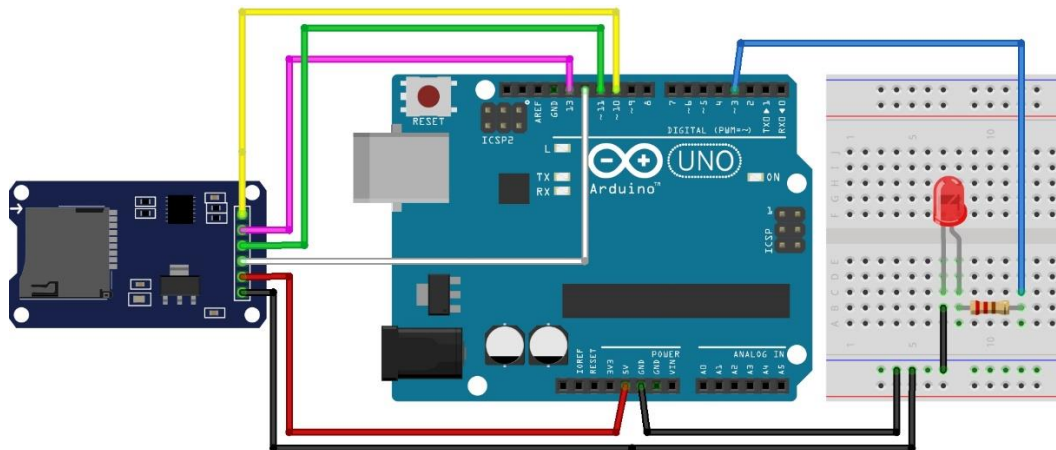


Рисунок 5.3 – Схема подключения платы расширения MicroSD Card

3. Напишите и загрузите скетч микроконтроллера:

// подключение библиотек SPI и SD:

#include <SPI.h>

#include <SD.h>

```

// указываем номер вывода arduino
// подключенного к выводу CS адаптера
const uint8_t PIN_CS = 10;
const int PIN_LED = 3;
// создаём переменную myFile класса File,
// которая будет служить "указателем" на открытый файл
File myFile;

bool file_exists(String fname){
    if(SD.exists(fname)){
        return true;
    } else{
        return false;
    }
}

void setup(){
    // открываем последовательный порт на скорости 9600 бод
    Serial.begin(9600);
    pinMode(PIN_LED, OUTPUT);
    // ожидаем соединение последовательного порта
    while(!Serial){;}
    // используем ответ инициализации,
    // для определения работоспособности карты и адаптера
    // инициализация SD карты с указанием номера вывода CS
    if(!SD.begin(PIN_CS)){
        // ошибка инициализации. карта не обнаружена
        // или не подключён (неправильно подключён) адаптер карт MicroSD
        Serial.println("SD-card not found"); return;
    } else {
        Serial.println("SD-card found. OK");
    }
}

String file_name = "data.csv";

if(file_exists(file_name)){
    Serial.println("File found");
}

```

```

    } else{
        Serial.println("File not found");
    }

    // Открываем файл
    myFile = SD.open(file_name);
    if(myFile){
        Serial.println("OK. File is opened");
    }
}

void loop(){
    int N = 2; int current_state = LOW;
    while(myFile.available()){
        // если есть данные для чтения
        // читаем очередной байт из файла
        // и выводим его в последовательный порт
        int n = myFile.parseInt();
        Serial.println(n);
        if (current_state == LOW){
            digitalWrite(PIN_LED, HIGH);
            current_state = HIGH;
        } else {
            digitalWrite(PIN_LED, LOW);
            current_state = LOW;
        }
        delay(n);
    }

    myFile.close();
    return;
}

```

4. Проверьте корректность работы собранного устройства и выполните задание в соответствии с индивидуальным вариантом.

Выполнение индивидуального задания

В каждом варианте индивидуального задания необходимо выполнить чтение и/или запись данных на sd карту.

Таблица 5.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Контроллер считывает данные с SD-накопителя и выводит их на монитор последовательного порта. В текстовом файле могут содержаться строки, числа.
2	Контроллер считывает данные с SD-накопителя и на основе считанных чисел осуществляет мигание светодиодом.
3	Контроллер считывает 5 чисел с SD-накопителя и рассчитывает сумму чисел, которую выводит на монитор последовательного порта.
4	Контроллер считывает 5 чисел с SD-накопителя и рассчитывает произведение чисел, которую выводит на монитор последовательного порта.
5	Контроллер считывает 5 чисел с SD-накопителя и рассчитывает среднее арифметическое чисел, которую выводит на монитор последовательного порта.
6	Контроллер считывает 53 числа с SD-накопителя и выводит на монитор последовательного порта максимальное число.
7	Контроллер считывает 100 чисел с SD-накопителя и выводит на монитор последовательного порта минимальное число.
8	Контроллер записывает на SD-карту числа, которые пользователь вводит в монитор порта.
9	Контроллер записывает на SD-карту числа, которые поступают с датчика температуры.
10	Контроллер записывает на SD-карту числа, которые поступают с датчика давления.
11	Контроллер записывает на SD-карту числа, которые поступают с датчика наклона.
12	Контроллер записывает на SD-карту числа, которые поступают с датчика вибрации.
13	Контроллер записывает на SD-карту числа, которые поступают с потенциометра.
14	Контроллер записывает на SD-карту числа, которые поступают с датчика звука.
15	Контроллер записывает на SD-карту числа, которые поступают с датчика влажности.

Содержание отчета и его форма

Отчет по работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Опишите особенности плат расширения для SD-карт.
2. Какая функция используется для записи данных на SD-карту?
3. Опишите функции для работы с каталогами на SD-накопителе.
4. Опишите процедуру создания нового файла на SD-накопителе.
5. Опишите процедуру чтения данных из файла на SD-накопителе.
6. С помощью какой синтаксической конструкции можно определить работоспособность SD-карты?
7. Как определить наличие файла на SD-диске?
8. Подумайте, каким образом реализовать просмотр каталогов SD-карты на Arduino (файловый браузер)?
9. Опишите процесс подключения платы расширения MicroSD Card к плате Arduino.

10. Какие модификации плат расширения для работы с SD-накопителями существуют на рынке?
11. Опишите стратегию использования СУБД в проектах Arduino.

Список литературы

Для выполнения работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 6. СДВИГОВЫЙ РЕГИСТР И СВЕТОДИОДНАЯ МАТРИЦА

Цель и содержание

Цель работы: научиться работать со светодиодной матрицей, освоить работу со сдвиговым регистром.

В рамках лабораторной работы необходимо решить следующие задачи:

- получить навыки программирования устройств с использованием светодиодных матриц;
- получить навыки работы со сдвиговым регистром;
- изучить теоретические основы функционирования сдвигового регистра.

Теоретическая часть

Сдвиговый регистр – это устройство, которое принимает поток последовательных битов и одновременно выводит их значения на параллельных контактах ввода-вывода. Регистры сдвига часто применяются для управления большим количеством светодиодов, например, семисегментными индикаторами или светодиодными матрицами. Прежде чем обсуждать взаимодействие сдвиговых регистров с Arduino, посмотрим на рис. 6.1, где изображены входы и выходы регистра сдвига. Далее будет показано, как состояние входов влияет на выходы.

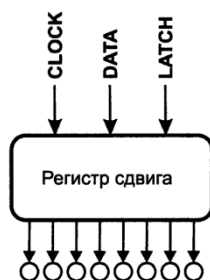


Рисунок 6.1 – Схема сдвигового регистра

Восемь кругов схематично изображают светодиоды, соединенные с восемью выходами сдвигового регистра. Три входа – это линии для подключения сдвигового регистра к плате Arduino. То есть вместо задействования 8 цифровых выходов платы Arduino будут задействованы только 3 контакта для управления 8 светодиодами.

В качестве сдвигового регистра мы будем применять микросхему 74HC595, цоколевка которой изображена на рис. 6.2.

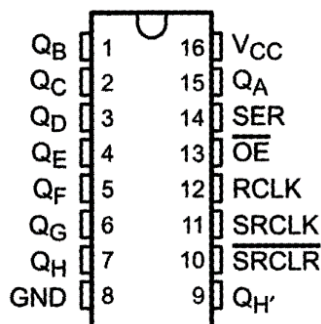


Рисунок 6.2 – Цоколевка микросхемы 74HC595

Назначение контактов сдвигового регистра:

- QA-QH – восемь параллельных выходов сдвигового регистра;
- GND – соединяется с земляной шиной платы Arduino;
- SER – это вход данных (DATA на рис. 6.1). По этому входу передаются 8 последовательных битов данных для установки значений на параллельных выходах;
- SRCLK – это тактовый вход (CLOCK на рис. 6.1). При подаче импульса высокого напряжения high на этот вход происходит считывание одного бита данных с входа DATA в сдвиговый регистр. Для получения всех 8 битов данных необходимо подать 8 импульсов на этот контакт;
- RCLK – это вход (LATCH на рис. 6.1) называется также защелкой и служит для одновременного вывода последовательных данных на параллельные выходы.

Регистр сдвига является синхронным устройством, он принимает данные по нарастающему фронту тактового сигнала. Каждый раз, когда сигнал на входе CLOCK меняется с низкого на высокий, все значения, хранящиеся в

восьми выходных ячейках, смещаются на одну позицию. Данные из последней ячейки либо сбрасываются, либо передаются на выход QH* (при каскадном подключении микросхемы 74HC595). Одновременно последовательные данные на входе DATA сдвигаются на одну позицию. За восемь тактов предыдущие значения в ячейках регистра уходят, а новые загружаются. Подача высокого уровня на вход LATCH выводит значения, хранящиеся в ячейках, на выходы регистра. Этот процесс проиллюстрирован на рис. 6.3.

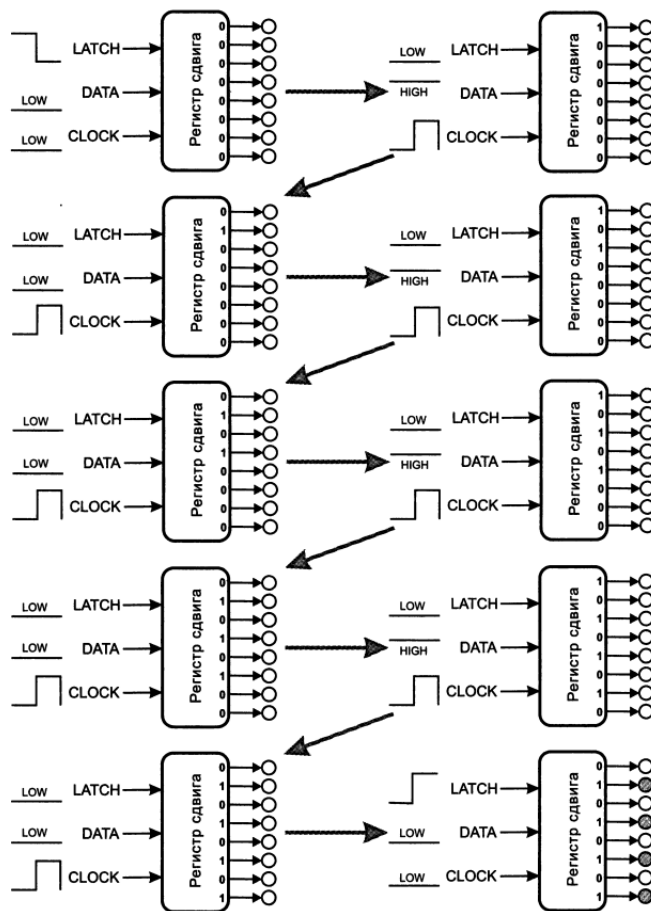


Рисунок 6.3 – Перемещение данных по сдвиговому регистру

Оборудование и материалы

Для выполнения лабораторной работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

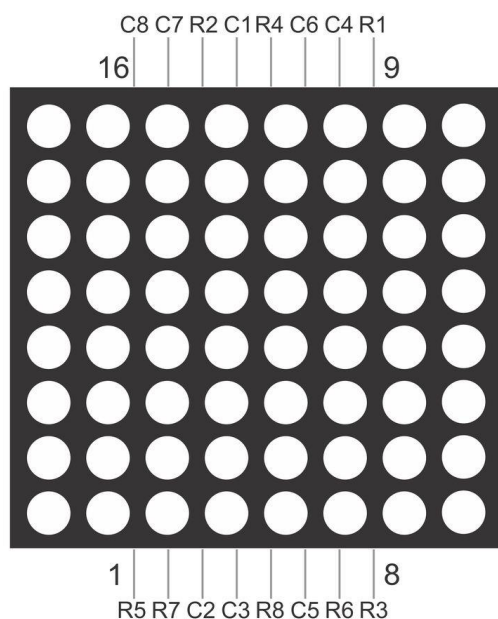
Методика и порядок выполнения работы

Решение учебной задачи

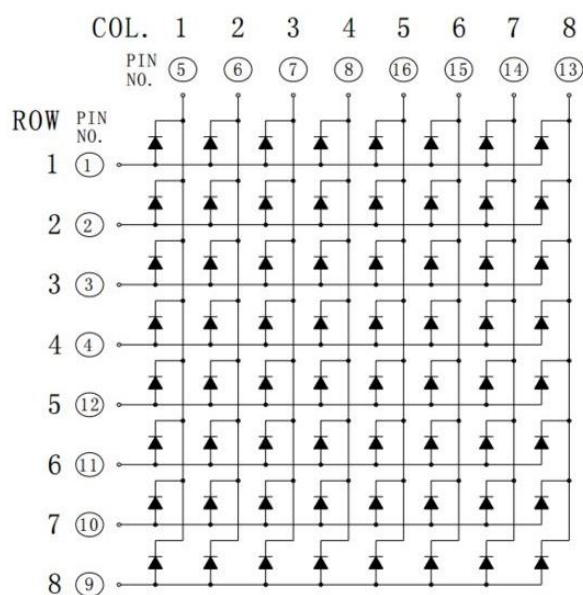
Постановка задачи: реализовать электронную схему управления устройством (датчиком, актюатором), требующим задействования большого количества выходов микроконтроллера. Целесообразно реализовать рассматриваемую схему без использования сдвигового регистра, затем реализовать требуемую схему с использованием сдвигового регистра. На основе реализации лабораторных опытов необходимо оценить эффективность применения сдвигового регистра. В данном учебном примере рассмотрим, когда на индикаторной матрице последовательно зажигаются светодиоды, расположенные по периметру светодиодной матрицы.

1. Рассмотрим светодиодную матрицу или матричный индикатор (рисунок 6.4а), а также принципиальную схему данного элемента (рисунок 6.4б).

Следует обратить внимание на нумерацию контактов матричного индикатора.



a)



б)

Рисунок 6.4 – Простейшая светодиодная матрица 8x8: а) внешний вид элемента; б) принципиальная схема

2. Схема устройства для управления светодиодной матрицей на основе алгоритма с использованием платы Arduino UNO (рисунок 6.5) выглядит следующим образом:

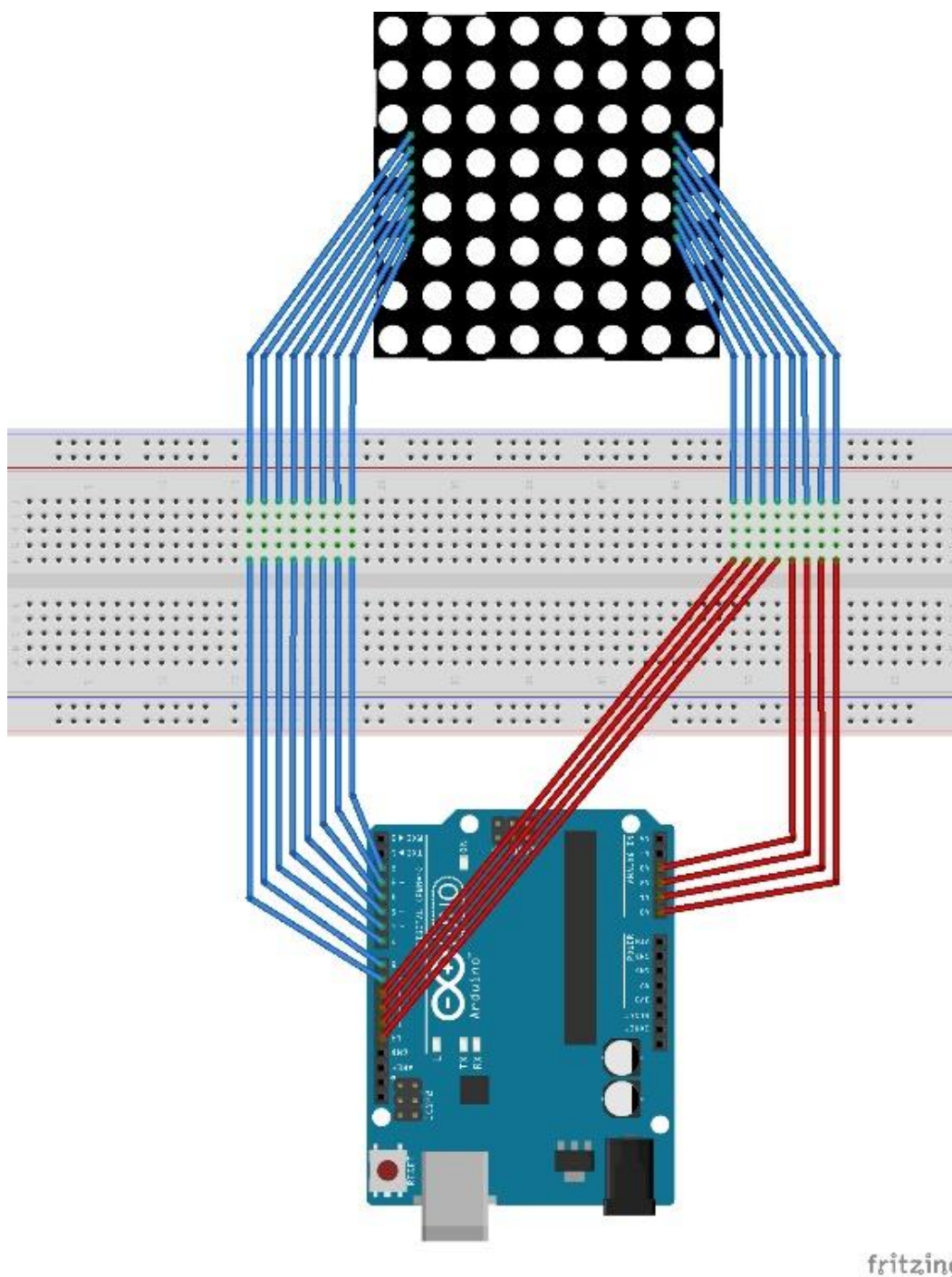


Рисунок 6.5 –Принципиальная схема управления светодиодной матрицей без использования сдвигового регистра

3. Для управления светодиодной матрицей используется следующий скетч:

```
int r1 = 2, r2 = 3,
    r3 = 4, r4 = 5,
    r5 = 6, r6 = 7,
    r7 = 8, r8 = 9;
int c1 = 10, c2 = 11,
```

```

    c3 = 12, c4 = 13,
    c5 = A0, c6 = A1,
    c7 = A2, c8 = A3;

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);
    ////////////
    pinMode(c1, OUTPUT); pinMode(c2, OUTPUT);
    pinMode(c3, OUTPUT); pinMode(c4, OUTPUT);
    pinMode(c5, OUTPUT); pinMode(c6, OUTPUT);
    pinMode(c7, OUTPUT); pinMode(c8, OUTPUT);

    pinMode(r1, OUTPUT); pinMode(r2, OUTPUT);
    pinMode(r3, OUTPUT); pinMode(r4, OUTPUT);
    pinMode(r5, OUTPUT); pinMode(r6, OUTPUT);
    pinMode(r7, OUTPUT); pinMode(r8, OUTPUT);
    ////////////
    digitalWrite(c1, LOW); digitalWrite(c2, HIGH);
    digitalWrite(c3, HIGH); digitalWrite(c4, HIGH);
    digitalWrite(c5, HIGH); digitalWrite(c6, HIGH);
    digitalWrite(c7, HIGH); digitalWrite(c8, HIGH);

    digitalWrite(r1, HIGH); digitalWrite(r2, LOW);
    digitalWrite(r3, LOW); digitalWrite(r4, LOW);
    digitalWrite(r5, LOW); digitalWrite(r6, LOW);
    digitalWrite(r7, LOW); digitalWrite(r8, LOW);
}

void loop() {

}

```

Данный скетч зажигает один светодиод, расположенный в первой колонке и в первой строке матрицы. Для динамического управления светодио-

дами такая структура программы не оптимальна. Можно модифицировать скетч следующим образом:

```
int r1 = 2, r2 = 3, r3 = 4, r4 = 5,
    r5 = 6, r6 = 7, r7 = 8, r8 = 9;
int c1 = 10, c2 = 11, c3 = 12, c4 = 13,
    c5 = A0, c6 = A1, c7 = A2, c8 = A3;

int c[8] = {c1, c2, c3, c4, c5, c6, c7, c8};
int r[8] = {r1, r2, r3, r4, r5, r6, r7, r8};

int cv[8] = {LOW,HIGH,HIGH,HIGH,HIGH,HIGH,HIGH,HIGH};
int rv[8] = {HIGH,LOW,LOW,LOW,LOW,LOW,LOW,LOW};
void updateCols(){
    for(int i = 0; i<8; i++)
        digitalWrite(c[i], cv[i]); }
void updateRows(){
    for(int i = 0; i<8; i++)
        digitalWrite(r[i], rv[i]); }
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);
    for(int i = 0; i<8; i++) {
        pinMode(c[i], OUTPUT);
        pinMode(r[i], OUTPUT); }
    updateRows();
    updateCols();
}

void loop() { }
```

4. Для динамического изменения положения светящегося светодиода необходимо изменить код функции loop() следующим образом:

```
short curCol = 0;
short curRow = 0;
short dir = 0;
void loop() {
```

```

delay(500);
if((curRow < 7) && (dir == 0)){
    digitalWrite(r[curRow], LOW);
    curRow++;
    digitalWrite(r[curRow], HIGH);
    if(curRow == 7) dir = 1;
}
else if((curCol < 7) && (dir == 1)){
    digitalWrite(c[curCol], HIGH);
    curCol++;
    digitalWrite(c[curCol], LOW);
    if(curCol == 7) dir = 2;
}
else if((curRow > 0) && (dir == 2)){
    digitalWrite(r[curRow], LOW);
    curRow--;
    digitalWrite(r[curRow], HIGH);
    if(curRow == 0) dir = 3;
}
else if((curCol > 0) && (dir == 3)){
    digitalWrite(c[curCol], HIGH);
    curCol--;
    digitalWrite(c[curCol], LOW);
    if(curCol == 0) dir = 0;
}
}
}

```

5. Проверьте корректность работы собранного устройства и выполните задание в соответствии с индивидуальным вариантом.

6. Реализуем аналогичную схему с использованием сдвигового регистра 74НС595. Данный элемент позволит снизить количество выходов контроллера, задействованных в управлении (рисунок 6.6).

Выполнение индивидуального задания

1. Для схемы, представленной на рисунке 6.6 напишите скетч, который реализует поведение системы, логичное поведению системы, представленной на рисунке 6.5.

2. Выполните задание в соответствии с индивидуальным вариантом.

Таблица 6.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Реализуйте схему с использованием датчика влажности, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит).
2	Контроллер считывает данные с SD-накопителя и на основе считанных данных производится управление светодиодной матрицей посредством одного сдвигового регистра.
3	Контроллер считывает числа с SD-накопителя и отображает соответствующие числа с использованием сегментного индикатора.
4	Реализуйте схему управления светодиодной матрицей с использованием 2-х сдвиговых регистров, то есть реализуйте последовательное соединение сдвиговых регистров.
5	Контроллер считывает данные с SD-накопителя и на основе считанных данных производится управление группой светодиодов посредством одного сдвигового регистра (чем больше число, тем больше светодиодов горит).
6	Реализуйте схему с использованием потенциометра, группы светодиодов и сдвигового регистра для отображения значения потенциометра посредством группы светодиодов (чем выше уровень, тем больше светодиодов горит).
7	Реализуйте схему с использованием датчика температуры, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит).
8	Реализуйте схему с использованием датчика магнитного поля, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит)..
9	Реализуйте схему с использованием аналогового датчика шума, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит).

10	Реализуйте схему с использованием манипулятора (джойстика), группы светодиодов и сдвигового регистра для отображения значения с манипулятора посредством группы светодиодов (светодиоды загораются в зависимости от направления манипулятора)..
11	Реализуйте схему с использованием датчика CO ₂ , группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит).
12	Реализуйте схему с использованием датчика влажности, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит).
13	Реализуйте схему с использованием датчика наклона, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит)..
14	Реализуйте схему с использованием датчика освещенности, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит).
15	Реализуйте схему с использованием датчика давления, группы светодиодов и сдвигового регистра для отображения значения с датчика посредством группы светодиодов (чем выше уровень сигнала с датчика, тем больше светодиодов горит)..

Содержание отчета и его форма

Отчет по работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Опишите назначение контактов сдвигового регистра 74HC595.
2. Поясните цикл работы со сдвиговым регистром. Назовите основные контакты, используемые в процессе передачи данных от микроконтроллера к сдвиговому регистру.
3. Назовите основные проекты, которые могут быть оптимизированы при использовании сдвиговых регистров.
4. Назовите модели сдвиговых регистров. Поясните отличия различных моделей.
5. Перечислите функции языка Си, которые используются для работы со сдвиговым регистром.
6. Поясните принцип работы матричных индикаторов. Поясните основные отличия различных моделей индикаторов. Поясните назначение контактов матричного индикатора.
7. Поясните алгоритм решения следующей задачи: необходимо зажечь 2 светодиода индикаторной матрицы, расположенных в разных строках и столбцах.
8. Поясните алгоритм реализации светодиодной индикации при использовании схем с использованием сдвиговых регистров.
9. Поясните принцип проектирования схем с каскадами сдвиговых регистров. Поясните назначение контактов, задействованных при последовательном соединении сдвиговых регистров.
10. В каких датчиках, актюаторах, устройствах в проектах Arduino осуществляется последовательная передача данных, а в каких параллельная?

Список литературы

Для выполнения работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 7. МЕХАНИЗМ ПРЕРЫВАНИЙ

Цель и содержание

Цель работы: освоение технологии работы с аппаратными прерываниями Arduino.

В рамках работы необходимо решить следующие задачи:

- изучение теоретических основ прерываний на аппаратной платформе Arduino;
- получение практических навыков разработки систем на базе прерываний Arduino;
- получение навыков по выявлению оптимальной стратегии проектирования и разработки устройств: на базе прерываний или с использованием цикла опроса;
- развитие навыков асинхронного программирования.

Теоретическая часть

Аппаратные прерывания происходят при наступлении (или изменении) заданного состояния на входах-выходах. Аппаратное прерывание полезно, например, когда нужно изменить значение переменной, не проверяя непрерывно состояние кнопки. То есть прерывание представляет собой процесс приостановки выполнения основной программы и переключение на выполнение процедуры, обрабатывающей некоторое событие (рисунок 7.1).

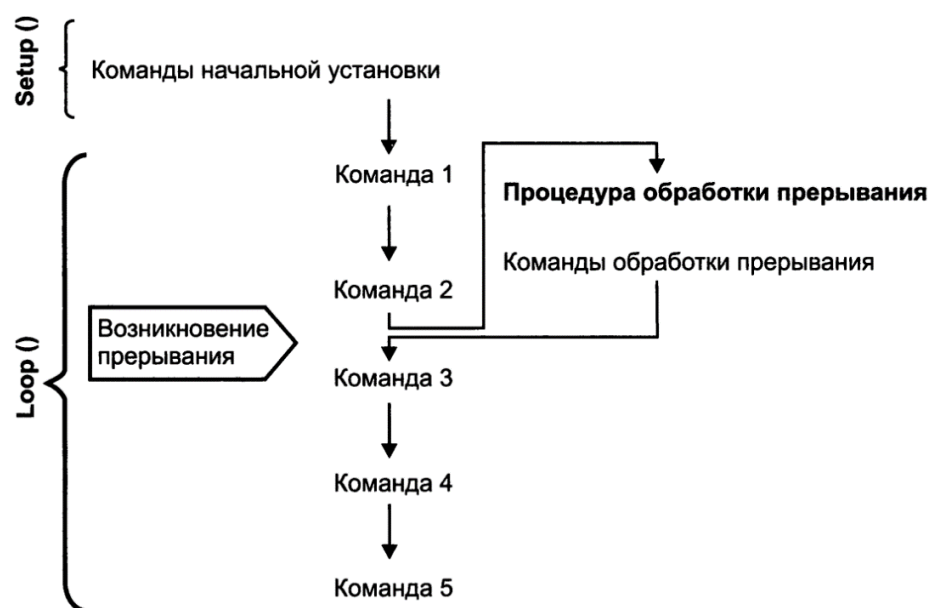


Рисунок 7.1 – Схема прерывания

Следует понимать, что проекты Arduino – это простейшие проекты на основе микроконтроллеров. Механизм прерываний усложняет программы, но его использования в отдельных случаях является целесообразным. Прерывания используются для обеспечения псевдомногозадачности.

В платах Arduino для аппаратных прерываний выделены специальные контакты. Прерывания обозначаются с помощью идентификационного номера, которому сопоставлен определенный контакт (табл. 7.1).

Таблица 7.1 – Контакты прерываний для плат Arduino

Плата	INT1	INT2	INT3	INT4	INT5	INT6
UNO	Pin2	Pin3	–	–	–	–
Mega	Pin2	Pin3	Pin21	Pin20	Pin19	Pin18
Leonardo	Pin3	Pin2	Pin0	Pin1	–	–

Для вызова прерывания существует встроенная функция `attachInterrupt ( )`. Ее первый аргумент – это идентификатор прерывания (для плат из таблицы 7.1) или номер контакта Arduino (для платы Due). Например, вызов функции:

`attachInterrupt(0, func, mode)`

для платы Arduino UNO назначает прерывание физическому контакту 2.

Второй аргумент функции `attachInterrupt ( )` – это имя той функции, которая будет вызвана при обработке прерывания. Третий параметр `mode` – это режим запуска прерывания. Существуют следующие режимы: `LOW`, `CHANGE`, `RISING` и `FALLING` (для платы Due также есть режим `HIGH`). Режимы `CHANGE`, `RISING` и `FALLING` выполняется только один раз при изменении состояния входа. Режимы `HIGH` и `LOW` вызывают процедуру обработки прерывания непрерывно, блокируя остальную часть программы.

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Постановка задачи: требуется разработать механизм на основе механизмов прерываний, позволяющий мгновенно изменять цвет RGB-светодиода (желтый, красный, зеленый). При этом яркость светодиода изменяется непрерывно от максимального значения до минимального.

Решение:

1. Схема рассматриваемого устройства без использования механизма прерываний представлена на рисунке 7.2.

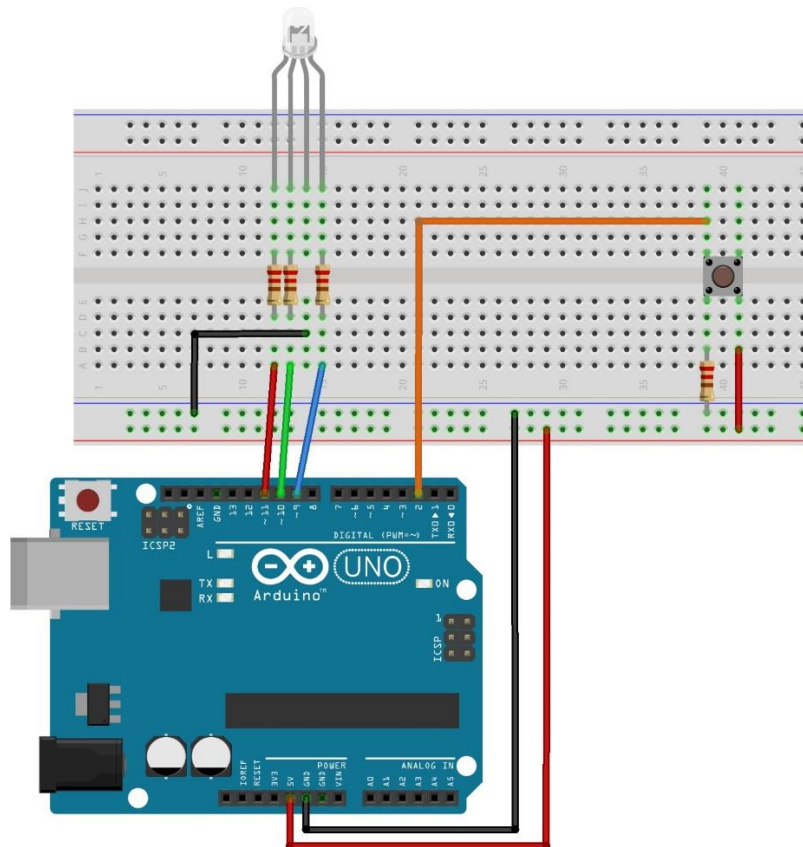


Рисунок 7.2 – Схема для изменения цвета RGB-светодиода без использования прерываний

2. Скетч, решающий учебную задачу выглядит следующим образом:

```
const int redPin = 11, greenPin = 10, bluePin = 9;  
// Interrupt 0 on pin 2
```

```

const int button_int = 2;

int selectedPin = redPin;

void swap(){
  Serial.print("swap - ");
  analogWrite(selectedPin, 0);
  if(selectedPin == redPin) {
    selectedPin = greenPin;
    Serial.println("GREEN");
  }
  else if (selectedPin == greenPin) {
    selectedPin = bluePin;
    Serial.println("BLUE");
  }
  else if (selectedPin == bluePin) {
    selectedPin = redPin;
    Serial.println("RED");
  }
}

void setup() {
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  // Rising
  pinMode(button_int, INPUT);
}

int state = 0;
void loop() {
  for(int i = 0; i < 256; i++)
  {
    analogWrite(selectedPin, i);
    delay(10);
    state = digitalRead(button_int);
  }
}

```

```

    if(state == HIGH) swap();
  }
  for(int i = 255; i >= 0; i--)
  {
    analogWrite(selectedPin, i);
    delay(10);
    state = digitalRead(button_int);
    if(state == HIGH) swap();
  }
}

```

3. При использовании механизма прерываний принципиальная схема устройства не меняется (рисунок 7.2), но скетч принимает вид:

```

const int redPin = 11, greenPin = 10, bluePin = 9;
// Interrupt 0 on pin 2
const int button_int = 0;

int selectedPin = redPin;

void swap(){
  Serial.print("swap - ");
  analogWrite(selectedPin, 0);
  if(selectedPin == redPin) {
    selectedPin = greenPin;
    Serial.println("GREEN");
  }
  else if (selectedPin == greenPin) {
    selectedPin = bluePin;
    Serial.println("BLUE");
  }
  else if (selectedPin == bluePin) {
    selectedPin = redPin;
    Serial.println("RED");
  }
}

```

```

void setup() {
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  // Rising
  attachInterrupt(button_int, swap, RISING);
}

```

```

int state = 0;
void loop() {
  for(int i = 0; i < 256; i++)
  {
    analogWrite(selectedPin, i);
    delay(10);
  }
  for(int i = 255; i >= 0; i--)
  {
    analogWrite(selectedPin, i);
    delay(10);
  }
}

```

4. Необходимо проанализировать приведенные скетчи, установить различия в коде, а также проанализировать поведение устройства при разлной реализации (с использованием прерываний и без).

5. Выполните индивидуальное задание.

Выполнение индивидуального задания

В каждом варианте индивидуального задания необходимо реализовать схему, написать скетч.

Таблица 7.2 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Реализуйте схему с использованием датчика давления, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
2	Реализуйте схему с использованием датчика влажности, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
3	Реализуйте схему с использованием манипулятора (джойстика), синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния актюатора.
4	Реализуйте схему с использованием датчика наклона, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
5	Реализуйте схему с использованием датчика освещенности, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
6	Реализуйте схему с использованием потенциометра, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния актюатора.
7	Реализуйте схему с использованием датчика температуры, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
8	Реализуйте схему с использованием датчика магнитного поля, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
9	Реализуйте схему с использованием датчика шума, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
10	Реализуйте схему с использованием манипулятора (джойстика), синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния актюатора.
11	Реализуйте схему с использованием датчика CO ₂ , синего, красного и зеленого светодиодов, позволяющую циклически переключать

	чать цвет горящего светодиода при изменении состояния датчика.
12	Реализуйте схему с использованием датчика влажности, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
13	Реализуйте схему с использованием датчика наклона, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
14	Реализуйте схему с использованием ультразвукового датчика, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.
15	Реализуйте схему с использованием датчика давления, синего, красного и зеленого светодиодов, позволяющую циклически переключать цвет горящего светодиода при изменении состояния датчика.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Поясните принцип выполнения программы Arduino в синхронном режиме.
2. Поясните принцип выполнения программы Arduino в асинхронном режиме.
3. Установите соответствия между номерами прерываний и выходами микроконтроллера. Поясните отличия механизма прерываний для разных плат.
4. Поясните особенности хода выполнения программы при задействовании механизма прерываний.
5. Опишите недостатки цифровых схем, реализуемых с механизмом прерываний. Поясните случаи, в которых целесообразно применение механизма опроса входов.
6. Поясните понятие «многозадачность». Как многозадачность реализуется в микроконтроллерах Arduino?
7. Что такое «дребезг»? Поясните принцип устранения данного явления.
8. Каките типы прерываний существуют в Arduino? Опишите их принцип действия и варианты применения.
9. Возможно ли применение в одной программе аппаратных прерываний, прерываний от таймера и непосредственный опрос входов?
10. Для чего применяется RC-цепочка и регистр Шмидта? Приведите пример схемы.

Список литературы

Для выполнения лабораторной работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 8. ДВИГАТЕЛИ

Цель и содержание

Цель работы: научиться проектировать устройства с использованием простейших двигателей.

В рамках лабораторной работы необходимо решить следующие задачи:

- получение практических навыков работы с сервоприводами;
- изучение теоретических основ функционирования электрических моторов;
- получение навыков программирования устройств с использованием электрических двигателей;
- получение навыков управления сервоприводом с использованием различных датчиков и актюаторов.

Теоретическая часть

Основные сведения о сервоприводах

В данной работе рассматриваются простейшие моторы – сервоприводы. Под сервоприводом чаще всего понимают механизм с электромотором, который можно попросить повернуться в заданный угол и удерживать это положение. Точнее, сервопривод – это привод с управлением через отрицательную обратную связь, позволяющую точно управлять параметрами движения. Сервоприводом является любой тип механического привода, имеющий в составе датчик (положения, скорости, усилия и т.п.) и блок управления приводом, автоматически поддерживающий необходимые параметры на датчике и устройстве согласно заданному внешнему значению.

Таким образом, следует запомнить:

1. Сервопривод получает на вход значение управляющего параметра. Например, угол поворота.

2. Блок управления сравнивает это значение со значением на своём датчике.

3. На основе результата сравнения привод производит некоторое действие: например, поворот, ускорение или замедление так, чтобы значение с внутреннего датчика стало как можно ближе к значению внешнего управляющего параметра.

Наиболее распространены сервоприводы, которые удерживают заданный угол, и сервоприводы, поддерживающие заданную скорость вращения. Простой сервопривод представлен на рисунке 8.1.



Рисунок 8.1 – Образец сервопривода

На рисунке 8.2 показана схема устройства сервопривода.

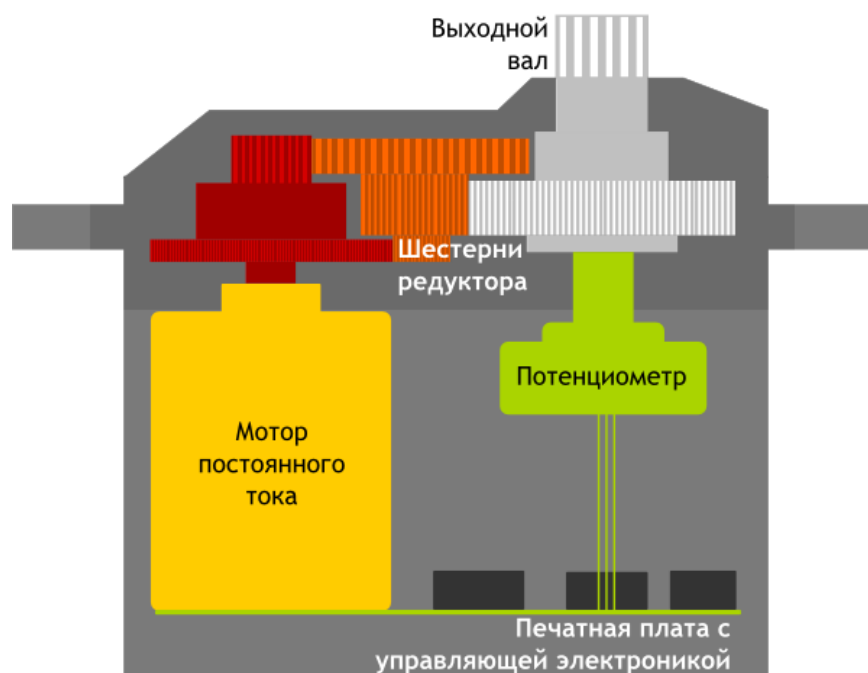


Рисунок 8.2 – Схема сервопривода

Привод – электромотор с редуктором. Чтобы преобразовать электричество в механический поворот, необходим электромотор. Однако зачастую скорость вращения мотора бывает слишком большой для практического использования. Для понижения скорости используется редуктор: механизм из шестерней, передающий и преобразующий крутящий момент.

Включая и выключая электромотор, можно вращать выходной вал – конечную шестерню сервопривода, к которой можно прикрепить нечто, чем мы хотим управлять. Однако, для того чтобы положение контролировалось устройством, необходим датчик обратной связи — энкодер, который будет преобразовывать угол поворота обратно в электрический сигнал. Для этого часто используется потенциометр. При повороте бегунка потенциометра происходит изменение его сопротивления, пропорциональное углу поворота. Таким образом, с его помощью можно установить текущее положение механизма.

Кроме электромотора, редуктора и потенциометра в сервоприводе имеется электронная начинка, которая отвечает за приём внешнего параметра, считывание значений с потенциометра, их сравнение и включение.

ние/выключение мотора. Она-то и отвечает за поддержание отрицательной обратной связи.

К сервоприводу тянется три провода. Два из них отвечают за питание мотора, третий доставляет управляющий сигнал, который используется для выставления положения устройства.

Характеристики сервоприводов

Крутящий момент и скорость поворота. Момент силы, или крутящий момент – векторная физическая величина, равная произведению радиус-вектора, проведенного от оси вращения к точке приложения силы, на вектор этой силы. Характеризует вращательное действие силы на твёрдое тело. Эта характеристика показывает, насколько тяжёлый груз сервопривод способен удержать в покое на рычаге заданной длины. Если крутящий момент сервопривода равен $5 \text{ кг} \times \text{см}$, то это значит, что сервопривод удержит на весу в горизонтальном положении рычаг длины 1 см, на свободный конец которого подвесили 5 кг. Или, что эквивалентно, рычаг длины 5 см, к которому подвесили 1 кг.

Скорость сервопривода измеряется интервалом времени, который требуется рычагу сервопривода, чтобы повернуться на 60° . Характеристика $0,1 \text{ с}/60^\circ$ означает, что сервопривод поворачивается на 60° за 0,1 с. Из неё не сложно вычислить скорость в более привычной величине, оборотах в минуту, но так сложилось, что при описании сервоприводов чаще всего используют такую единицу.

Стоит отметить, что иногда приходится искать компромисс между этими двумя характеристиками, так как если мы хотим надёжный, выдерживающий большой вес сервопривод, то мы должны быть готовы, что эта могучая установка будет медленно поворачиваться. А если мы хотим очень быстрый привод, то его будет относительно легко вывести из положения равновесия. При использовании одного и того же мотора баланс определяет конфигурация шестерней в редукторе.

Конечно, мы всегда можем взять установку, потребляющую большую мощность, главное, чтобы её характеристики удовлетворяли нашим потребностям.

Форм-фактор. Сервоприводы различаются по размерам. И хотя официальной классификации не существует, производители давно придерживаются нескольких размеров с общепринятым расположением крепёжных элементов. Их можно разделить на: маленькие, стандартные, большие.

Внутренний интерфейс. Сервоприводы бывают аналоговые и цифровые. Внешне они ничем не отличаются: электромоторы, редукторы, потенциометры у них одинаковые, различаются они лишь внутренней управляющей электроникой. Вместо специальной микросхемы аналогового сервопривода у цифрового собрата можно заметить на плате микропроцессор, который принимает импульсы, анализирует их и управляет мотором. Таким образом, в физическом исполнении отличие лишь в способе обработки импульсов и управлении мотором.

Оба типа сервопривода принимают одинаковые управляющие импульсы. После этого аналоговый сервопривод принимает решение, надо ли изменить положение, и в случае необходимости посылает сигнал на мотор. Происходит это обычно с частотой 50 Гц. Таким образом получаем 20 мс — минимальное время реакции. В это время любое внешнее воздействие способно изменить положение сервопривода. Но это не единственная проблема. В состоянии покоя на электромотор не подаётся напряжение, в случае небольшого отклонения от равновесия на электромотор подаётся короткий сигнал малой мощности. Чем больше отклонение, тем мощнее сигнал. Таким образом, при малых отклонениях сервопривод не сможет быстро вращать мотор или развивать большой момент. Образуются «мёртвые зоны» по времени и расстоянию.

Эти проблемы можно решать за счёт увеличения частоты приёма, обработки сигнала и управления электромотором. Цифровые сервоприводы используют специальный процессор, который получает управляющие

импульсы, обрабатывает их и посылает сигналы на мотор с частотой 200 Гц и более. Получается, что цифровой сервопривод способен быстрее реагировать на внешние воздействия, быстрее развивать необходимую скорость и крутящий момент, а значит, лучше удерживать заданную позицию, что хорошо. Конечно, при этом он потребляет больше электроэнергии. Также цифровые сервоприводы сложнее в производстве, а потому стоят заметно дороже. Собственно, эти два недостатка – все минусы, которые есть у цифровых сервоприводов. В техническом плане они безоговорочно побеждают аналоговые сервоприводы.

Подключение сервопривода к Arduino и программирование

Многие сервоприводы могут быть подключены к Arduino непосредственно. Для этого от них идёт шлейф из трёх проводов:

- 1) красный — питание; подключается к контакту 5 В или напрямую к источнику питания;
- 2) коричневый или чёрный – земля;
- 3) жёлтый или белый – сигнал; подключается к цифровому выходу Arduino.

Можно генерировать управляющие импульсы самостоятельно, но это настолько распространённая задача, что для её упрощения существует стандартная библиотека Servo. Библиотека Servo позволяет осуществлять программное управление сервоприводами. Для этого заводится переменная типа Servo. Управление осуществляется следующими функциями:

- 1) `attach( )` – присоединяет переменную к конкретному пину. Возможны два варианта синтаксиса для этой функции: `servo.attach(pin)` и `servo.attach(pin, min, max)`. При этом `pin` – номер пина, к которому присоединяют сервопривод, `min` и `max` – длины импульсов в микросекундах, отвечающих за углы поворота 0° и 180°. По умолчанию выставляются равными 544 мкс и 2400 мкс соответственно.

2) `write( )` – отдаёт команду сервоприводу принять некоторое значение параметра. Синтаксис следующий: `servo.write(angle)`, где `angle` — угол, на который должен повернуться сервопривод.

3) `writeMicroseconds( )` – отдаёт команду послать на сервопривод импульс определённой длины, является низкоуровневым аналогом предыдущей команды. Синтаксис следующий: `servo.writeMicroseconds(uS)`, где `uS` – длина импульса в микросекундах.

4) `read( )` – читает текущее значение угла, в котором находится сервопривод. Синтаксис следующий: `servo.read()`, возвращается целое значение от 0 до 180.

5) `attached( )` – проверка, была ли присоединена переменная к конкретному пину. Синтаксис следующий: `servo.attached( )`, возвращается логическая истина, если переменная была присоединена к какому-либо пину, или ложь в обратном случае.

6) `detach( )` – производит действие, обратное действию `attach( )`, то есть отсоединяет переменную от пина, к которому она была приписана. Синтаксис следующий: `servo.detach( )`.

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.

2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).

3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Решение учебной задачи

Спроектировать цифровое устройство с использованием сервомотора, которое по нажатию кнопки осуществляет непрерывное вращение сервопривода; при повторном нажатии осуществляется остановка вращения. При вращении сервопривода должен гореть светодиод.

Решение:

1. Спроектируем схему данного устройства (рисунок 8.1):

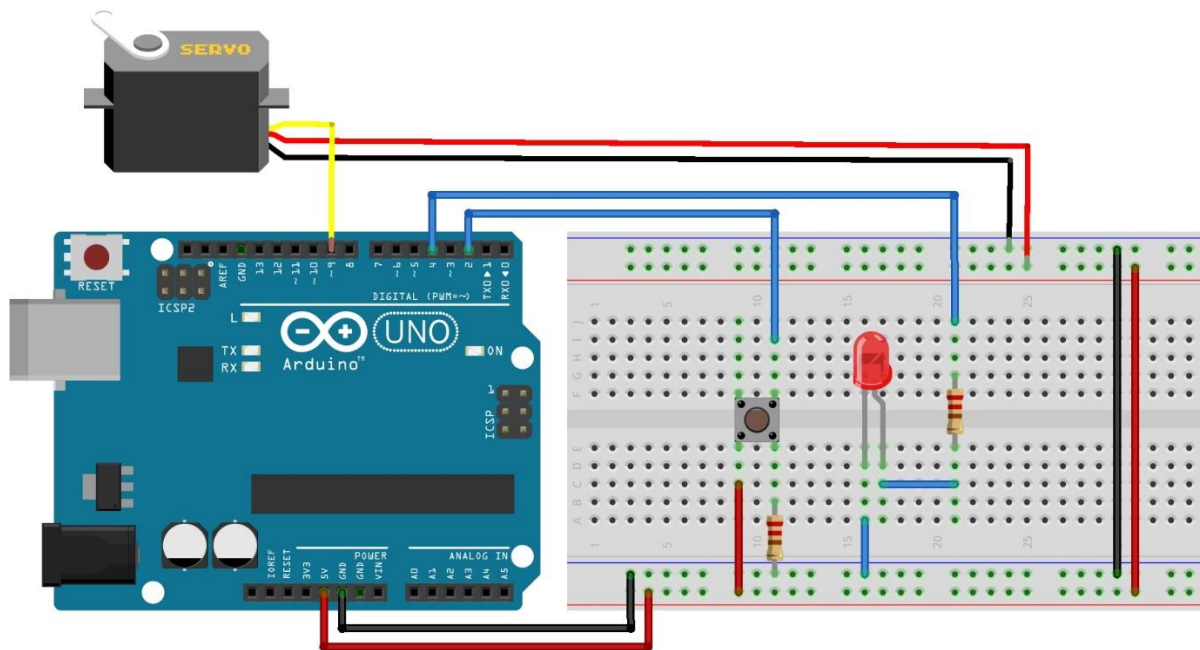


Рисунок 8.1 – Схема управления сервоприводом с использованием кнопки

2. Скетч для данной схемы, который выполняет поставленную задачу, выглядит следующим образом:

```
#include <Servo.h>

Servo srv;
const int lightPin = 4,
        servoPin = 9,
        btn_interrupt = 0;

bool isRotate = false;
short btnValue = LOW;
short srvValue = 0;
short srvStep = 90;

void swap(){
    isRotate = !isRotate;
    if(isRotate) Serial.println("start rotate");
    else Serial.println("stop rotate");
}

void setup() {
    Serial.begin(9600);
    pinMode(lightPin, OUTPUT);
    pinMode(servoPin, OUTPUT);
    attachInterrupt(btn_interrupt, swap, RISING);
    srv.attach(servoPin);
    digitalWrite(lightPin, LOW);
}

void rotateStep(){
    if(isRotate){
        srvValue = srvValue + srvStep;
        if((srvValue >= 170) || (srvValue <= 0)) srvStep = -srvStep;
        srv.write(srvValue);
        delay(300);
    }
}
```

```

}

void loop() {
  if(isRotate){
    digitalWrite(lightPin, HIGH);
  } else{
    digitalWrite(lightPin, LOW);
  }
  rotateStep();
}

```

3. Проанализировав схему и скетч можно сделать вывод, что устройство на основе тактовой кнопки будет подвержено эффектудребезга. Подумайте, каким образом можно модифицировать схему или скетч для минимизациидребезга.

4. Выполните индивидуальное задание.

Выполнение индивидуального задания

В каждом варианте индивидуального задания необходимо реализовать схему, написать скетч.

Таблица 8.1 – Варианты индивидуальных заданий

№ варианта	Постановка задачи
1	Реализуйте схему с использованием серводвигателя, потенциометра, светодиода. При вращении ручки потенциометра происходит вращение сервопривода на пропорциональный угол. Во время вращения горит светодиод.
2	Реализуйте схему с использованием серводвигателя, датчика температуры, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод.
3	Реализуйте схему с использованием серводвигателя, датчика влажности, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время

	вращения сервомотора горит светодиод.
4	Реализуйте схему с использованием серводвигателя, датчика давления, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод.
5	Реализуйте схему с использованием серводвигателя, датчика шума, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод.
6	Реализуйте схему с использованием серводвигателя, датчика освещенности, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод.
7	Реализуйте схему с использованием серводвигателя, джойстика, светодиода. При изменении положения джойстика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод.
8	Реализуйте схему с использованием серводвигателя, датчика CO ₂ , светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод.
9	Реализуйте схему с использованием серводвигателя, модуля SD-карты, светодиода. Вращение сервопривода происходит на углы, значения которых считываются с SD-карты.
10	Реализуйте схему с использованием серводвигателя, нескольких тактовых кнопок, светодиодов. При нажатии на первую кнопку происходит вращение сервопривода на угол 1, при этом горит диод 1; при нажатии второй кнопки – вращение на угол 2, горит диод 2; и т.д.
11	Реализуйте схему с использованием серводвигателя, датчика шума, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод, происходит запись значений температуры на SD-карту.
12	Реализуйте схему с использованием серводвигателя, датчика влажности, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод, происходит запись значений температуры на SD-карту.
13	Реализуйте схему с использованием серводвигателя, датчика температуры, светодиода. При изменении выхода датчика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод, происходит вывод значений датчика в монитор порта.

14	Реализуйте схему с использованием серводвигателя, потенциометра, светодиода. При вращении ручки потенциометра происходит вращение сервопривода на пропорциональный угол. Во время вращения горит светодиод, а также осуществляется вывод значений с потенциометра в монитор порта.
15	Реализуйте схему с использованием серводвигателя, джойстика, светодиода. При изменении положения джойстика происходит вращение сервопривода на пропорциональный угол. Во время вращения сервомотора горит светодиод, а также в монитор порта выводится информация о манипуляциях джойстиком.

Содержание отчета и его форма

Отчет по работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Опишите принцип действия серводвигателя.
2. Назовите основные характеристики серводвигателей.
3. Как подключить серводвигатель к Arduino?
4. Какие типы двигателей, отличных от серводвигателей, существуют в проектах Arduino?

5. Что такое шаговый двигатель? Чем данный тип двигателя отличается от сервопривода?
6. Как учитывается крутящий момент при проектировании устройств на основе Arduino?
7. Какая библиотека используется в проектах Arduino при работе с серводвигателями?
8. Перечислите основные функции библиотеки Servo.
9. Какие серводвигатели использовать предпочтительнее: цифровые или аналоговые?
10. Опишите технологию подключения мощного двигателя к Arduino.

Список литературы

Для выполнения лабораторной работы необходимо изучить материал, представленный в [1, 2, 5, 6, 7].

ЗАНЯТИЕ 9. КОМПЛЕКСНЫЙ ПРОЕКТ С ИСПОЛЬЗОВАНИЕМ МИКРОКОНТРОЛЛЕРА

Цель и содержание

Цель работы: разработка комплексного проекта, обобщающего знания и умения, приобретенные в рамках лабораторного практикума.

В рамках работы необходимо решить следующие задачи:

- разработка цифровой схемы, решающей практическую задачу;
- программирование скетча сложной многокомпонентной цифровой схемы;
- анализ и выявление слабых мест спроектированного устройства, описание путей модернизации спроектированного устройства.

Теоретическая часть

В процессе работы необходимо использовать теоретические знания полученные в ходе выполнения предшествующих лабораторных работ.

Оборудование и материалы

Для выполнения работы необходимо использовать следующее программное обеспечение (возможно развертывание программного обеспечения на операционных системах Linux или Windows):

1. Программа для проектирования электронной схемы и визуализации принципиальной схемы подключения устройств Fritzing.
2. Виртуальная среда прототипирования и эмуляции проектов в области электроники (онлайн версия) Autodesk TinkerCAD (<https://www.tinkercad.com>).
3. Среда разработки Arduino IDE (desktop- или онлайн-версия).

Указания по технике безопасности

Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Необходимо разработать схему на основе Arduino для решения практической задачи (научной, производственной или бытовой). Примерный перечень проектов представлен в таблице. По согласованию с преподавателем учащийся может предложить собственную тему для разработки.

Таблица 9.1 – Тематика индивидуальных проектов

№ варианта	Постановка задачи
1	Разработка системы автоматического полива.
2	Разработка сигнализации на основе датчика света.
3	Разработка сигнализации на основе датчика шума.
4	Разработка электронной игры.
5	Разработка декоративного светильника, реагирующего на окружающую среду.
6	Разработка ультразвукового радара
7	Разработка сигнализации на основе датчика наклона
8	Разработка модуля управления лампой на основе датчика движения
9	Разработка метеостанции
10	Разработка системы доступа на основе метки RFID
11	Разработка системы управления микроклиматом с логированием данных на SD-карту.
12	Разработка оптической линии связи
13	Разработка LED-светильника с использованием сдвиговых регистров

14	Робот, выполняющий команды пользователя от джойстика
15	Робот, управляемый звуком

Содержание отчета и его форма

Отчет по работе должен содержать:

1. Номер и название работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Какие типы контроллеров Arduino существуют на рынке? Что такое клоны Arduino?
2. Опишите основные шаги необходимые для организации рабочего места программиста, занимающегося разработкой проектов IoT на базе Arduino.
3. Что такое цифровой вход микроконтроллера? Сколько цифровых входов имеет Arduino UNO?
4. Поясните назначение функции `pinMode()`. Опишите входные параметры данной функции.
5. Перечислите датчики, которые являются совместимыми с платами Arduino. Что означает эта совместимость?
6. Как реализовать взаимодействие нескольких плат Arduino?

7. Опишите возможности микроконтроллерной техники в области реализации коммуникаций между различными микроконтроллерами, датчиками, актюаторами. Какие протоколы для этого существуют?
8. Опишите решения, которые доступны на рынке для реализации взаимодействия плат Arduino посредством сети WiFi.
9. Опишите различия и сходства проектов IoT и IIoT.
10. Опишите перспективные технологии IIoT, которые претендуют на использование в качестве основы для цифрового мира будущего.

Список литературы

Для выполнения лабораторной работы необходимо изучить материал, представленный в [1-3].

Литература

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации: учеб. пособие для вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; Под ред. А. П. Пятибратова. – М.: КноРус, 2017. – 372 с.
2. Росляков А.В.. Интернет вещей / Росляков А.В., Ваняшин С.В., Гребешков А.Ю., Самсонов М.Ю. Москва: ПГУТИ, ООО «Издательство Ас Гард», 2014. – 340 с.
3. Тихвинский В.О. Сети IoT/M2M: технологии, архитектура и приложения / Тихвинский В.О., Коваль В.А., Бочечка Г.С., Бабин А.И. М.: Издательский дом Медиа Пабlishер, 2017 – 320 с.

Перечень основной и дополнительной литературы

Перечень основной литературы

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации: учеб. пособие для вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; Под ред. А. П. Пятибратова. – М.: КноРус, 2017. – 372 с.

2. Росляков А.В.. Интернет вещей / Росляков А.В., Ваяшин С.В., Гребешков А.Ю., Самсонов М.Ю. Москва: ПГУТИ, ООО «Издательство Ас Гард», 2014. – 340 с.

3. Тихвинский В.О. Сети IoT/M2M: технологии, архитектура и приложения / Тихвинский В.О., Коваль В.А., Бочечка Г.С., Бабин А.И. М.: Издательский дом Медиа Паблишер, 2017 – 320 с.

Перечень дополнительной литературы:

1. Industry 4.0: The Industrial Internet of Things 1st ed. Edition by Alasdair Gilchrist, 2016

2. А. Садков. Беспроводные сенсорные сети. Учебный курс. Лаборатория Физич-ских Основ и Технологий Беспроводной Связи.

3. Principles of Cyber-Physical Systems - Rajeev Alur, MIT Press, 2015

4. Internet of Things, 1st Edition - Rajkumar Buyya, Amir Vahid Dastjerdi, Morgan Kaufmann, 2016

5. Building the Web of Things With examples in Node.js and Raspberry Pi - Dominique D. Guinard and Vlad M. Trifa, June 2016 ISBN 9781617292682, 344 pages

6. Internet of Things: A Hands-On Approach - Arshdeep Bahga, Vijay Madiseti, 2014

7. Getting Started with Intel Edison: Sensors, Actuators, Bluetooth, and Wi-Fi on the Ti-ny Atom-Powered Linux Module by Stephanie Moyerman, 2015

8. Building the Web of Things With examples in Node.js and Raspberry Pi - Dominique D. Guinard and Vlad M. Trifa, June 2016 ISBN 9781617292682, 344 pages

9. Building Internet of Things with the Arduino by Charalampos Doukas, 2012

10. Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us by Emily Gertz, Patrick Di Justo, 2012

11. Wiring the IoT, Connecting Hardware with Raspberry Pi, Node-Red, and MQTT by Dr. Lucy Rogers, Dr. Andy Stanford-Clark, 2017
12. Learning IBM Bluemix by Sreelatha Sankaranarayanan, 2016
13. Embedded Systems: Real-Time Interfacing to Arm Cortex-M Microcontrollers 2nd ed. Edition by Jonathan W. Valvano, 2016