

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ  
ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ  
«МИКРОПРОЦЕССОРНАЯ ТЕХНИКА»

Выпускная квалификационная работа  
по направлению подготовки 44.03.04 Профессиональное обучение  
(по отраслям)  
профилю подготовки «Энергетика»  
специализации «Компьютерные технологии автоматизации и управления»

Идентификационный код ВКР: 177

Екатеринбург 2017

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующий кафедрой ИС

\_\_\_\_\_ Н. С. Толстова

«\_\_\_\_\_» \_\_\_\_\_ 2017г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ЛАБОРАТОРНЫХ  
РАБОТ ПО ДИСЦИПЛИНЕ «МИКРОПРОЦЕССОРНАЯ ТЕХНИКА»**

Исполнитель:

обучающийся группы № КТэ-402

И.А. Набиев

Руководитель:

ассистент

М. Ю. Черноскутов

Нормоконтролер:

ст.преподаватель

Т. В. Рыжкова

## АННОТАЦИЯ

Выпускная квалификационная работа состоит из аппаратно-программного комплекса для лабораторных работ по дисциплине «Микропроцессорная техника» и пояснительной записки на 80 страницах, содержащей 65 рисунков, 13 таблиц, 22 источников литературы, а также 2 приложения на 7 страницах.

Ключевые слова: ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ, АППАРАТНЫЙ КОМПЛЕКС, ПРОГРАММНЫЙ КОМПЛЕКС, ЛАБОРАТОРНЫЕ РАБОТЫ.

**Набиев, И. А.** Аппаратно-программный комплекс для лабораторных работ по дисциплине «Микропроцессорная техника»: выпускная квалификационная работа / И. А. Набиева; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2017. — 80с.

Целью работы является разработка технического обеспечения для дисциплины «Микропроцессорная техника».

Для достижения указанной цели были поставлены следующие задачи:

- проанализировать литературу о микропроцессорной технике;
- разработать модули для решения задач лабораторных работ;
- разработать методические указания для реализации лабораторных работ на разработанных модулях.

Разработанные устройства позволяют ознакомить обучающихся с новыми протоколами обмена информации, управлять контактами ввода/вывода, программировать микроконтроллер на языке высокого уровня, а использование модульной структуры позволяет модернизировать и усовершенствовать данный комплекс.

# СОДЕРЖАНИЕ

|  |    |
|--|----|
| Введение.....  | 4  |
| 1 Микроконтролер как средство обучения микропроцессорной технике .....   | 6  |
| 1.1 Микропроцессор и его структура.....  | 6  |
| 1.1.1 Микроконтроллер как микропроцессор .....   | 6  |
| 1.2 Учебный микропроцессорный комплекс.....  | 7  |
| 1.3 Платформа Arduino UNO .....  | 8  |
| 1.3.1 Микроконтроллер ATmega328P .....   | 9  |
| 2 Разработка аппаратных средств для реализации лабораторных работ по дисциплине «микропроцессорная техника»..... | 11 |
| 2.1 Основной контроллер .....  | 11 |
| 2.1.1 Модули для контроллера.....  | 11 |
| 2.2 Разработка аппаратного обеспечения .....   | 12 |
| 2.2.1 Плата расширения.....  | 13 |
| 2.2.2 Модуль статических кнопок .....  | 16 |
| 2.2.3 Модуль матрицы из кнопок .....   | 17 |
| 2.2.4 Модуль матрицы из светодиодов .....  | 19 |
| 2.2.5 Модуль семисегментного индикатора .....  | 21 |
| 2.2.6 Модуль шкалы светодиодного индикатора.....   | 23 |
| 2.2.7 Модуль ШИМ.....  | 24 |
| 2.2.8 Модуль потенциометр .....  | 26 |
| 2.2.9 Модуль датчика температуры и давления.....   | 27 |
| 2.2.10 Модуль сдвигового регистра .....  | 28 |
| 2.2.11 Модуль I2C .....  | 31 |
| 2.3 Разработка методических указаний к лабораторным работам.....   | 33 |
| Заключение .....   | 70 |
| Список использованных источников .....   | 71 |
| Приложение А Задание на подготовку ВКР.....  | 74 |
| Приложение Б. Перечень элементов .....   | 76 |

## **ВВЕДЕНИЕ**

Задачи курса «Микропроцессорной техники» состоит в том, чтобы дать студенту базовые принципы функционирования микропроцессорных систем, предлагаются методы проектирования микропроцессорных систем на основе микроконтроллеров.

Микропроцессорная техника – область электроники которая быстро развивается, для её овладения необходимо усвоить организацию микропроцессорных систем.

Для качественного изучения материала по данной дисциплине была выбрана открытая платформа Arduino UNO на микро-ЭВМ фирмы Atmel «ATmega328P-PU».

Объектом данной выпускной квалификационной работы является техническое обеспечение дисциплины микропроцессорная техника.

Предметом работы является технические средства реализации лабораторных работ по дисциплине микропроцессорная техника.

Целью работы является разработать техническое обеспечение для дисциплины микропроцессорная техника.

Для достижения указанной цели были поставлены следующие задачи:

- проанализировать литературу о микропроцессорной технике;
- разработать модули для решения задач лабораторных работ;
- разработать методические указания для реализации лабораторных работ на разработанных модулях.

Разработанные модули для решения задач лабораторных работ на разработанных модулях.

Разработанные устройства позволяют ознакомить обучающихся с новыми протоколами обмена информации, управлять контактами ввода/вывода, программировать микроконтроллер на языке высокого уровня, а использование модульной структуры позволяет модернизировать и усовершенствовать данный комплекс.

Выпускная квалификационная работа состоит из двух частей.

В первой части проанализированы литературные источники, и предметная область.

Во второй части разработано техническое обеспечение аппаратной базы дисциплины микропроцессорная техника и методическое обеспечение к нему.

# 1 МИКРОКОНТРОЛЕР КАК СРЕДСТВО ОБУЧЕНИЯ МИКРОПРОЦЕССОРНОЙ ТЕХНИКЕ

## 1.1 Микропроцессор и его структура

Микропроцессор - устройство, отвечающее за выполнение арифметических, логических операций и операций управления, записанных в машинном коде [2]. Он способен выполнять множество операций. Для работы с ним нужно написать программу с системой команд, которые заложены в самом процессоре. Для выполнения команд, в структуру процессора входят внутренние регистры, арифметико-логическое устройство (АЛУ), мультиплексоры, буферы и другие узлы. Работа всех узлов синхронизируется общим внешним тактовым сигналом [1] (рисунок 1).

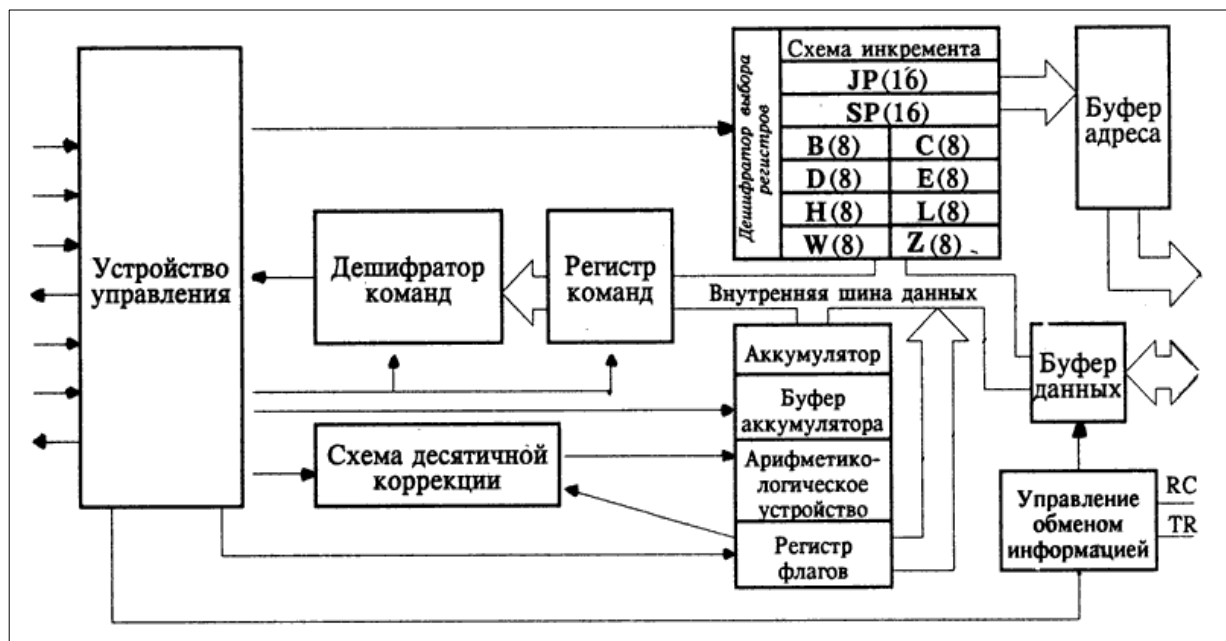


Рисунок 1 – Структура микропроцессора

### 1.1.1 Микроконтроллер как микропроцессор

Микропроцессор очень редко представляют, как отдельную микросхему. В дополнение на одном кристалле могут располагать вспомогательные

компоненты: периферию, оперативную и постоянную память и различные дополнительные устройства. Такую сборку называют – микроконтроллером.

Микроконтроллер может включать в себя различные периферийные устройства [1]:

- интерфейсы ввода\вывода, например: UART, I2C, USB и другие;
- преобразователи ЦАП и АЦП;
- таймер/счётчик;
- контроллеры дисплеев и клавиатур;
- другие периферийные устройства.

Таким образом микроконтроллер является однокристальным компьютером, который способен решать несложные задачи. Такие задачи как правило не ограничиваются исключительно вычислениями, а чаще направлены на управление оборудованием. Такое управление может производиться с помощью описанных выше интерфейсов (UART, I2C, и т.д.), либо простым использованием портов ввода-вывода.

## **1.2 Учебный микропроцессорный комплекс**

Так как целью данной работы является разработка технического обеспечения дисциплины микропроцессорная техника, то необходимо рассмотреть существующие решения, обратить внимание на их достоинства и недостатки.

Для выполнения лабораторных работ по данной дисциплине предусмотрено использование нескольких учебных комплексов: на базе микропроцессора Intel 8080 и Intel 8086. Здесь мной рассмотрена УМПК на базе микропроцессора K1810BM86, являющегося копией процессора фирмы Intel 8086. В данном комплексе присутствует периферия, такая как таймер и устройства ввода\вывода клавиатуры и дисплея. Микросхема «KP580BI53», является трёхканальным таймером счётчика интервалов, а ввод\вывод клави-



атуры и дисплея представлен микросхемой «КР580ВВ79». Дисплей представлен в виде семисегментных индикаторов.

Данный учебный микропроцессорный комплекс не имеет технической возможности подключения к персональному компьютеру. Из-за этого недостатка нет возможности программировать его напрямую из операционной системы. В начале нужно написать программный код на языке ассемблер в программе для перевода в систему команд для данного микропроцессора, а после переписать команды в сегмент кода с помощью клавиатуры УМПК. Такой подход в программировании микропроцессорного комплекса является проблематичным как с точки зрения непосредственно программирования, так и отладки из-за возможного внесения ошибочных данных и отсутствия возможности проверить их быстрым способом.

Ограниченный функционал микропроцессора и периферий не позволяет расширенно использовать различные современные интерфейсы. Отсутствие использования языка высокого уровня увеличивают порог вхождения и усложняют процесс программирования данного комплекса. Все компоненты расположены на одной печатной плате, что усложняет процесс ремонта данного комплекса в случае выхода из строя компонентов.

### **1.3 Платформа Arduino UNO**

Для решения проблем, вызванных использованием УМПК 8086 была выбрана платформа Arduino UNO [4], которая позволяет решить ряд проблем, такие как подключение к персональному компьютеру, программирование на языках как низкого, так и высокого уровня, использование различных интерфейсов (SPI, UART и других). Платформа Arduino UNO имеет широкий набор внутренних периферийных устройств, а также позволяет использовать различные внешние устройства. Для данной платформы имеется много различной литературы как на русском, так и на иностранных языках, что упрощает самообучение. Из-за того, что данная платформа имеет открытую архи-

тектуру, то данную систему можно свободно копировать и дополнять. Это дало сторонним производителям возможность создавать аналоги данной платформы, и продавать их по более низким ценам.

Для связи с компьютером использован микроконтроллер ATmega16U2 который выступает в качестве программатора и определяется в персональном компьютере как COM порт. Программирование происходит с помощью протокола STK500 [19].

Данная платформа также имеет регулятор напряжения до 5В выполненный на микросхеме AMS1117 [16], которой позволяет использовать на данной платформе отдельный разъём для подключения внешнего питания от 7 до 12В.

Язык программирования на Arduino представляет собой язык высокого уровня C++ [5] с некоторым набором сопроводительных библиотек [18]. Разработка ведётся в бесплатном программном обеспечении Arduino IDE. Данное программное обеспечение помогает новичкам освоиться в программном коде, и не предлагает настройку компилятора, что уменьшает риск возникновения связанных с этих проблем. Для компиляции используется открытый набор компиляторов GNU GCC [17].

### **1.3.1 Микроконтроллер ATmega328P**

Для решения технической базы для лабораторных работ была выбрана платформа Arduino UNO. Данный контроллер имеет следующие характеристики:

- цифровые порты ввода/вывода: 14 контактов (из них 6 подключены к широтно-импульсному модулятору);
- аналоговые порты ввода\вывода в количестве 6 штук;
- энергонезависимую флэш-память для программ – 32 Кб;
- оперативное запоминающее устройство - 2 Кб;
- постоянное запоминающее устройство – 1 Кб;

- тактовая частота на Arduino-совместимых платах – 16 МГц.

Также в данный контроллер встроены следующие периферийные устройства:

- два 8-битных таймера/счетчика;
- один 16-битный таймер/счетчик;
- счетчик реального времени с возможностью отдельного тактирования;
- 6-канальный 10-разрядный АЦП;
- программируемый последовательный интерфейс USART;
- последовательный интерфейс SPI;
- байтно-ориентированный последовательный интерфейс (совместим с I2C Philips);
- программируемый сторожевой таймер со встроенным генератором;
- прерывание и пробуждение по изменению на выводах.

Данный микроконтроллер имеет рабочее напряжение от 1.8 до 5.5 В, и температурный диапазон от -40 до 85 градусов по Цельсию.

## **2 РАЗРАБОТКА АППАРАТНЫХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «МИКРОПРОЦЕССОРНАЯ ТЕХНИКА»**

### **2.1 Основной контроллер**

Arduino UNO являясь открытой платформой позволяет собирать различные электронные устройства. Она состоит из аппаратной и программной частей. Для программирования и общения с компьютером требуется USB кабель.

Использование данной платформы позволяет использовать широкие возможности, так как контроллер может управлять цифровыми портами в количестве 13 штук, так и 6 аналоговыми портами ввода\вывода.

У данной платформы есть различные версии и модификации. Оригинальная Arduino UNO продаётся на официальном сайте за 1500 рублей не включая доставку, а также существуют различные аналоги, в ценовой категории от 400 рублей. Данная цена является бюджетной для контроллера данного уровня.

#### **2.1.1 Модули для контролера**

Для платформы Arduino UNO существуют различные электронные модули для осуществления индикации и измерения различных свойств. Данные модули являются законченными устройствами и располагаются на плате расширения, они подключаются к контролеру с помощью соединительных проводов, беспаячной макетной платы или платы расширения.

Для соединения нескольких модулей или отдельных электронных компонентов применяют беспаячную макетную плату. У неё есть проблемы с подключением различных компонентов, так-как контакты внутри макетной

платы легко разбалтываются и не позволяют качественно соединить электронные компоненты. В следствии чего появляется явление, называемое как дребезгом контактов. Сама беспаячная макетная плата имеет различные размеры, ее стоимость начинается от 200 рублей. Её можно использовать только при макетировании какого-либо устройства или при сборке небольших устройств, где наличие плохого контакта может быть диагностировано в короткие сроки.

Платы расширения так же позволяют подключать модули или компоненты для платформы, но к ним требуются соединительные провода, так как в основном они являются платой с разъёмами для упрощённого соединения шлейфов, в ценовой категории от 500 рублей.

Различные электронные компоненты, зачастую можно купить за небольшую цену, а в готовых устройствах они стоят в несколько раз больше обычной стоимости.

## **2.2 Разработка аппаратного обеспечения**

После изучения и сравнения различных готовых компонентов, представленных на рынке, было принято решение, что для реализации задач обучения лучше разработать свои модули и схему подключения. Собственная разработка должна исключать соединительные провода или беспаячные макетные платы, что, уменьшит возможность проблем с подключением модулей и недостатков, вызванных дребезгом контактов. Модульная структура предполагает сэндвичную систему подключения (рисунок 2), когда модули соединяются с платформой с помощью платы расширения, это в свою очередь упрощает разработку модулей и повышает работоспособность всей системы, так как при поломке одного из модулей нужно заменить только этот модуль.

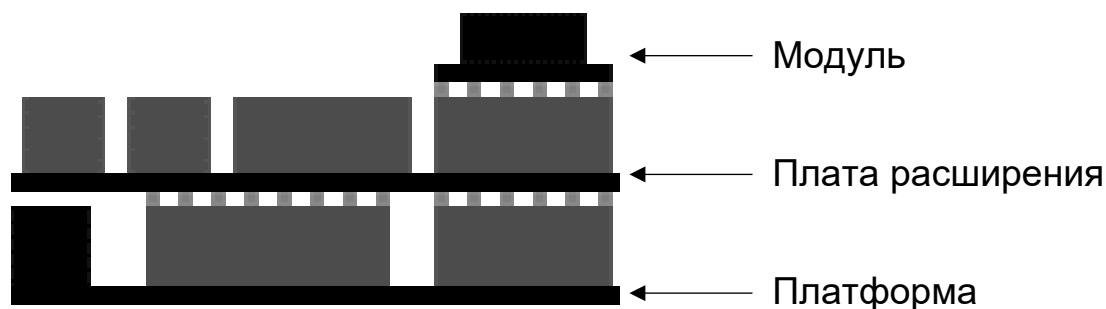


Рисунок 2 – Сэндвичная система подключения

Разработка велась в САПР (система автоматизированного проектирования) CircuitMaker [3], бесплатном аналоге САПР Altium Designer [15]. С его помощью можно создать весь пакет документов, необходимых в соответствии с Единой системой конструкторской документации: от принципиальной электрической схемы и чертежа печатной платы до спецификации и сборочного чертежа. Так же он позволяет сгенерировать файлы необходимые для производства. Плюсом данного продукта является бесплатность, возможность изготовления больших комплексных проектов, а также возможность совместной разработки с использованием облака. В то же время данный продукт требует постоянного подключения к сети Интернет. В ходе разработки были созданы стандартные библиотеки необходимых компонентов.

Перечень компонентов для каждой платы представлен в приложении Б.

### 2.2.1 Плата расширения

Основным назначением платы расширения является подключение модулей к платформе Arduino UNO без лишних соединительных точек. Для подключения модулей была изготовлена плата расширения, для которой разработана принципиальная схема подключения и макет печатной платы. Для соединения платы расширения с платформой Arduino UNO использованы коннекторы типа PLS, а для подключения модулей к плате расширения используются коннекторы типа PBS. Цена данной платы будет низкой, т.к. нет электронных компонентов, использован только односторонний металлизиро-

ванный текстолит и широко распространенные коннекторы. Каждый из трёх модулей, подключается к плате расширения универсальным портом (группой разъёмов), состоящим из разъёма питания, одного аналогового, и минимум двух (максимум шести) цифровых выводов. Для реализации протокола I2C на плате расширения так же выведены соответствующие контакты.

Плата расширения [17] содержит три слота под модули (рисунок 3) размером 17.7x43.1 мм. На месте для первого слота размещается универсальный контакт (0, +5В, аналоговый вывод), два цифровых и два контакта для i2c. Второй слот имеет универсальный контакт и шесть цифровых выводов. Третий слот имеет самый расширенный выбор: универсальный контакт, шесть цифровых выводов и два контакта для I2C.

Печатная плата расширения имеет размеры 54x54 мм.

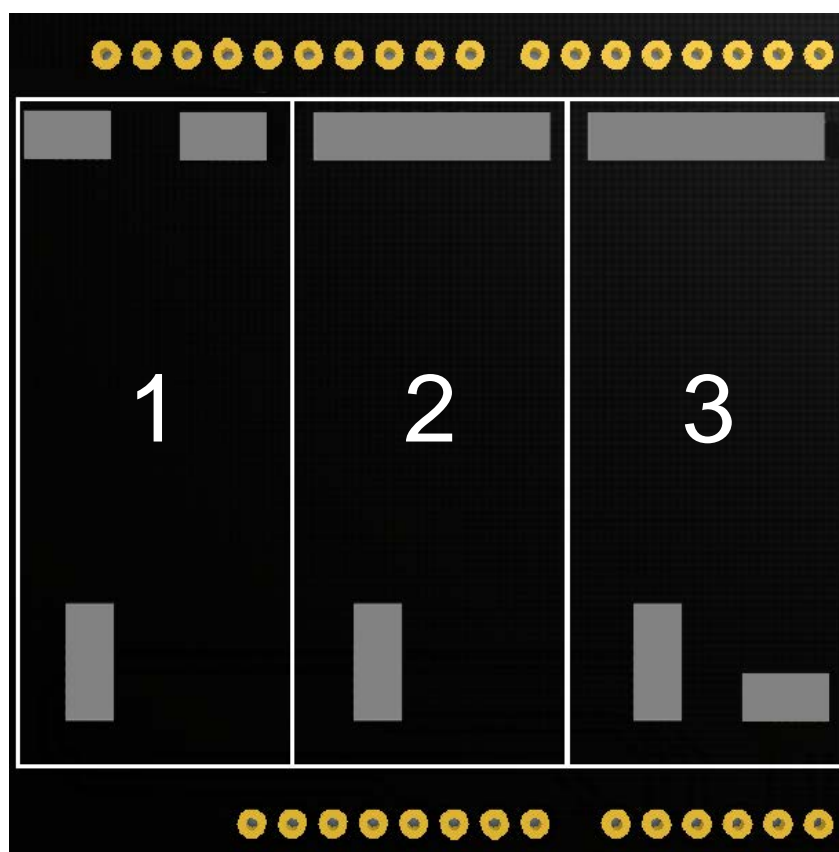


Рисунок 3 – Плата расширения с обозначением слотов

Принципиальная электрическая схема представлена на рисунке 4.

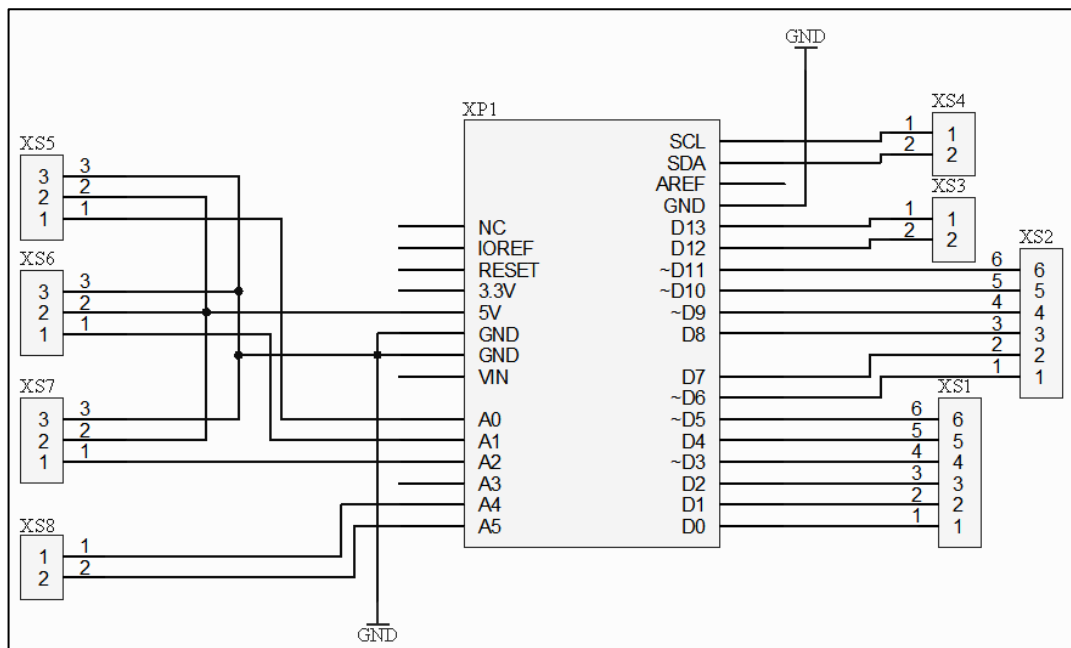


Рисунок 4 – Принципиальная электрическая схема платы расширения

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 5.

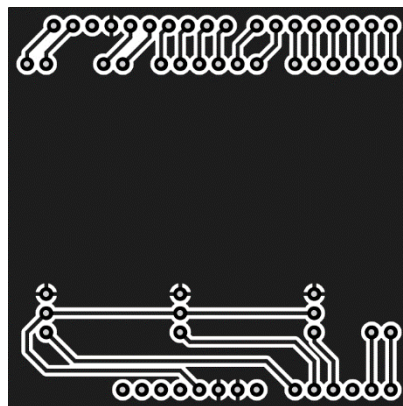


Рисунок 5 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 6.

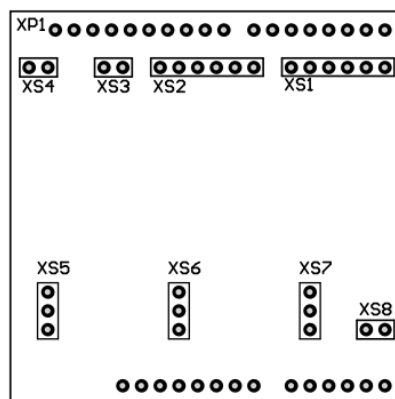


Рисунок 6 – Сборочный чертёж платы расширения



## 2.2.2 Модуль статических кнопок

В данном модуле используются 2 кнопки, нормально разомкнутые без фиксации. Они являются статическим вводом, так как один из контактов соединён с нулем вольт на плате, а другой из контакта кнопки подключён к цифровому выводу. Модуль можно подключать в первый и второй разъёмы, и нежелательно в третьей, так как его выводы будут приходить на 0 и 1 вывод Arduino, которые являются контактами для общения с персональным компьютером при загрузке программы.

Печатная плата модуля имеет размеры 17.7x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 7.

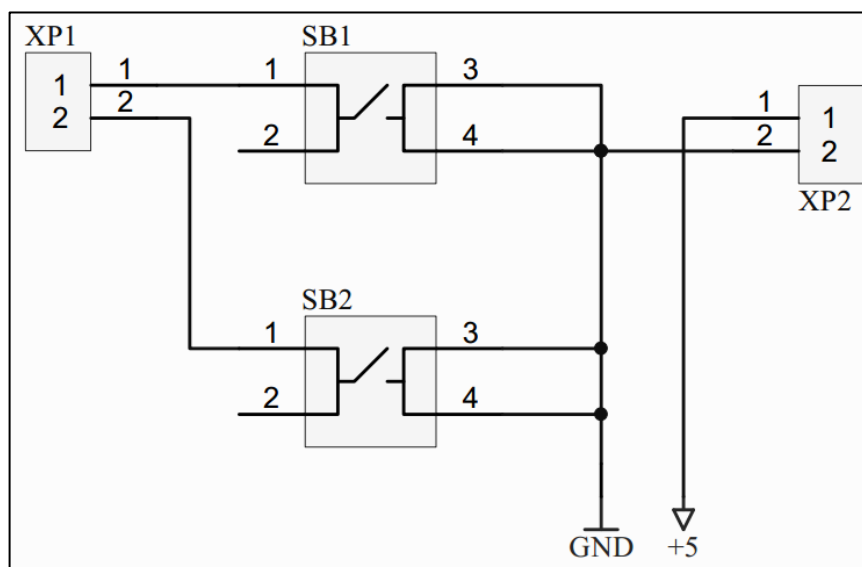


Рисунок 7 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 8.

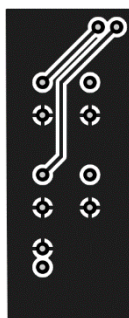


Рисунок 8 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 9.

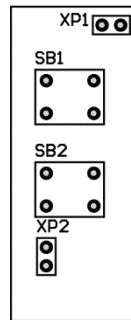


Рисунок 9 – Сборочный чертёж модуля

### 2.2.3 Модуль матрицы из кнопок

Для управления контроллера можно использовать кнопки, подключить их можно статически, но для девяти кнопок потребуется 9 контактов микроконтроллера. Для уменьшения контактов можно использовать матричный способ подключения (рисунок 10), который позволяет уменьшить количество контактов до шести за счёт того, что сканирование кнопок будет производиться рядами. Сам модуль соединяется с платой расширения в один из слотов, два или три. При использовании во втором слоте он своими размерами перекрывает доступ к использованию третьего, в связи с размерами платы.

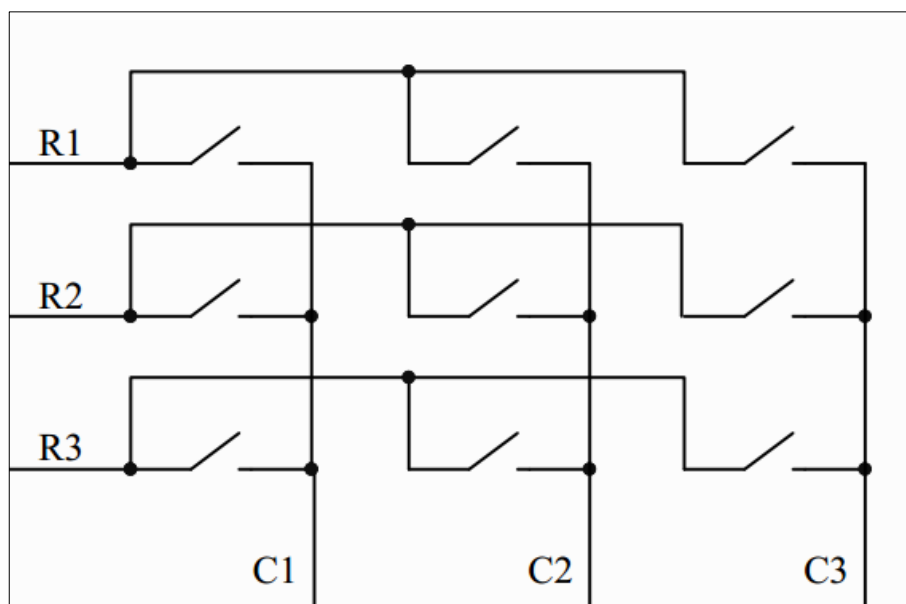


Рисунок 10 – Матричный способ соединения кнопок

В данном модуле используются 9 кнопок, расположенных квадратом 3x3. Один из контактов кнопки соединён горизонтально, а второй вертикально. Кнопки являются нормально разомкнутыми без фиксации, имеют 4 вывода, два из которых соединены внутри.

Печатная плата модуля имеет размеры 35.4x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 11.

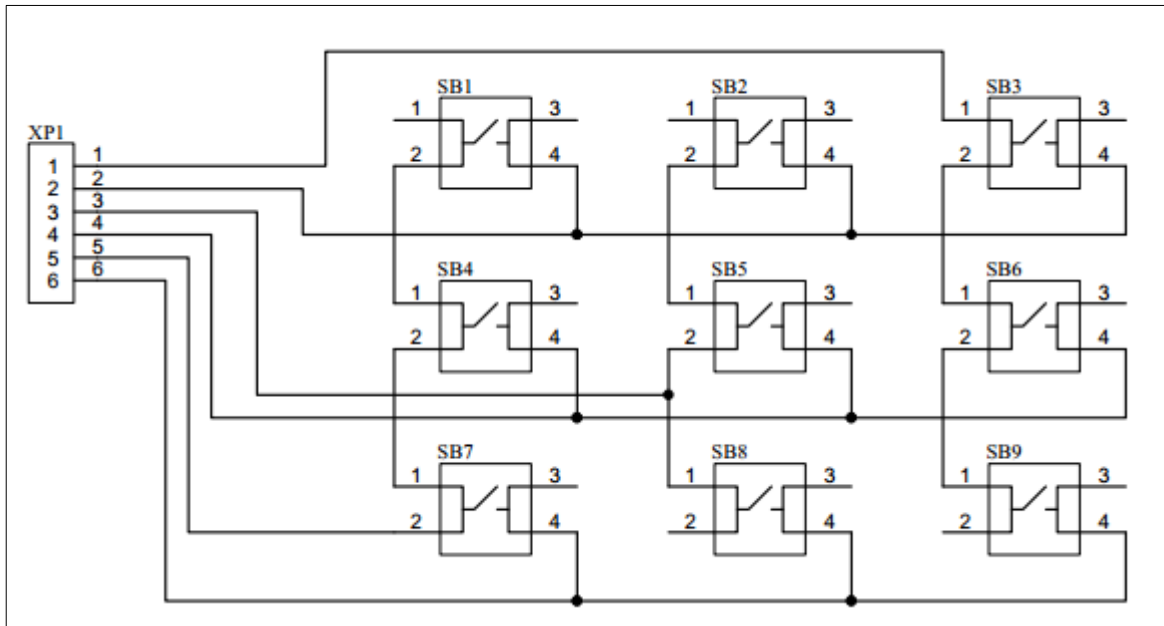


Рисунок 11 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 12.

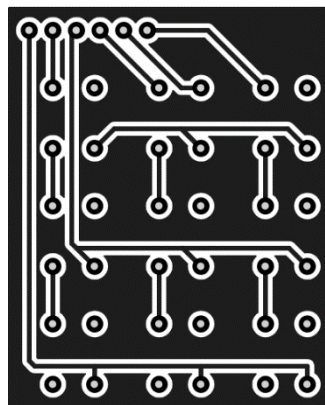


Рисунок 12 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 13.

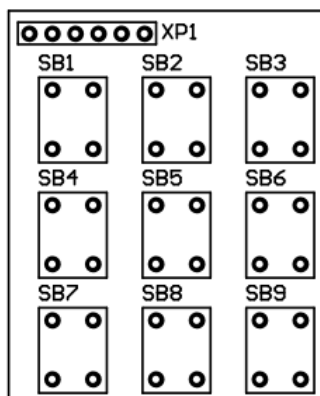


Рисунок 13 – Сборочный чертёж модуля

## 2.2.4 Модуль матрицы из светодиодов

Модуль соединяется с платой расширения аналогично модулю из матрицы кнопок, подключение компонентов так же аналогично. Модуль нужен для динамической индикации, так как она позволяет уменьшить нагрузку на электронные компоненты. Модуль использует 9 выводных светодиодов [9], расположенных квадратом 3x3. Анод контакта светодиода соединён в колонки через резистор 200 ом [7], а катод в ряды. Данный модуль имеет мультиплексированный вывод. Он позволяет, используя физическое свойство светодиода, зажигать только тот светодиод, которому правильно подается напряжение, если же на анод и катод подаётся +5В, то такой светодиод не загорается.

Печатная плата модуля имеет размеры 35.4x43.1 мм. Разводка данного модуля произведена на верхнем и нижнем слое: на верхнем слое припаивается резистор и аноды, а на нижнем слое соединяются катоды.

Принципиальная электрическая схема представлена на рисунке 14.

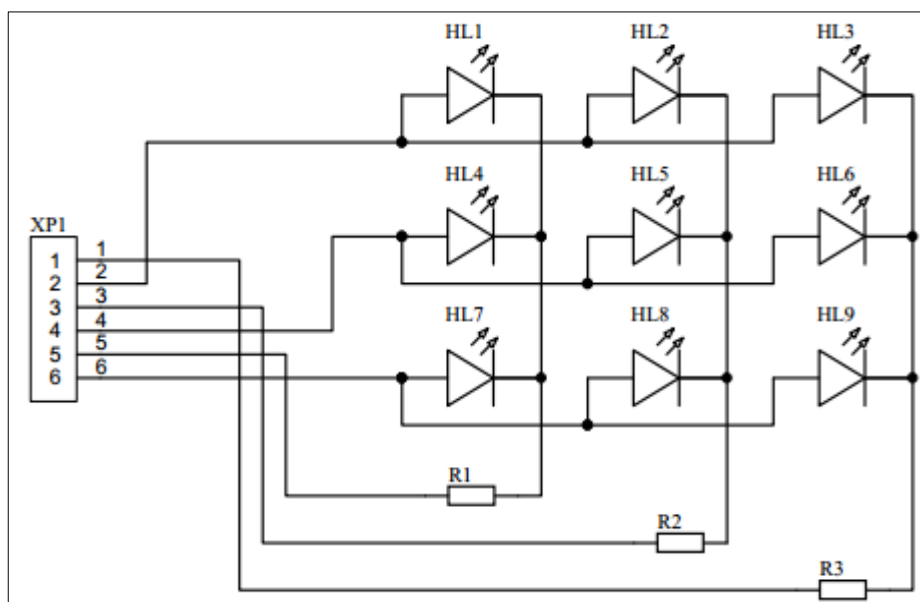


Рисунок 14 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунках 15 и 16.

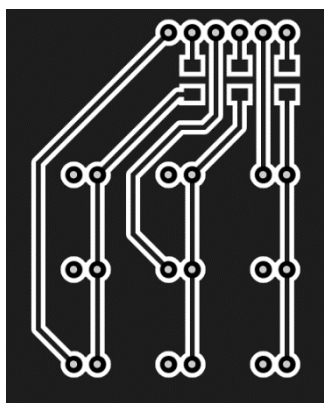


Рисунок 15 – Чертёж печатной платы. Верхний слой

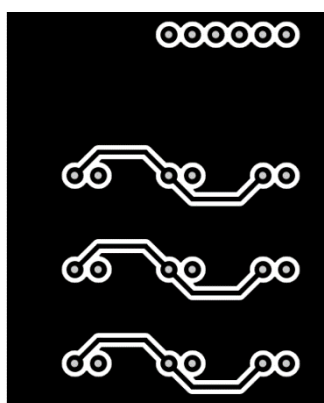


Рисунок 16 – Чертёж печатной платы. Нижний слой

Сборочный чертёж представлен на рисунке 17.

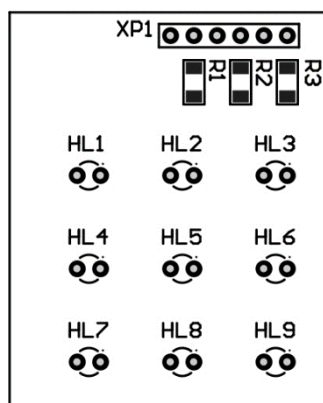


Рисунок 17 – Сборочный чертёж модуля

### 2.2.5 Модуль семисегментного индикатора

Семисегментный индикатор [9] представляет собой устройство для отображения арабских цифр с точкой. Индикатор состоит из восьми элементов светодиодной индикации – семи сегментов и точки. При включении комбинаций сегментов можно составить арабские цифры. Такие индикаторы часто встречаются в различной аппаратуре, и из-за их распространенности нужно знать, как на них выводится информация.

В модуле используется индикатор с общим анодом, а катоды подсоединены к цифровым контактам через резистор 200 ом [7]. Таким образом данный модуль занимает второй и третий слот, так как для его управления требуется 8 цифровых контактов. Каждый из сегментов зажигается отдельно друг от друга, поэтому можно создавать различные комбинации [13].

Печатная плата модуля имеет размеры 35.4x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 18.

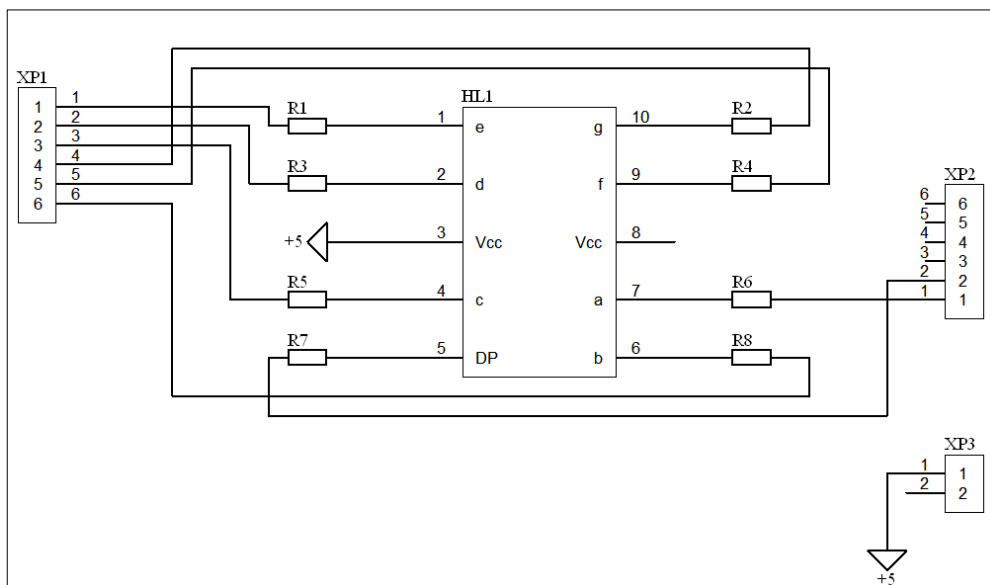


Рисунок 18 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 19.

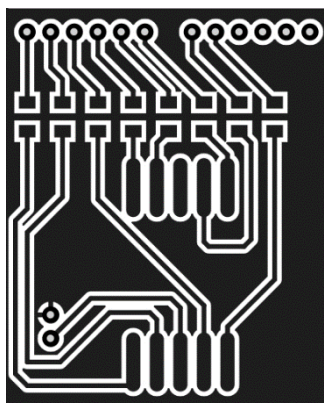


Рисунок 19 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 20.

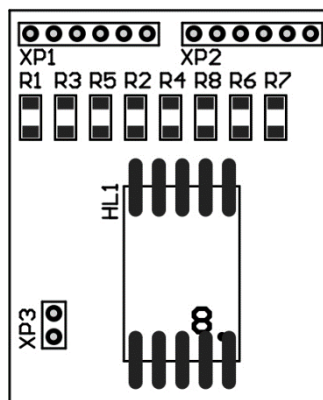


Рисунок 20 – Сборочный чертёж модуля

## 2.2.6 Модуль шкалы светодиодного индикатора

Шкальный индикатор [9] – это устройство, состоящее из нескольких независимых светодиодов, расположенных в форме шкалы и объединены в один корпус. Каждый сегмент имеет 2 вывода: анод и катод.

В данном модуле используется десятисегментный индикатор, его катоды соединены вместе с минусовым коннектором платы, а аноды соединены с цифровыми выводами через резистор 200ом [7]. Модуль занимает второй и третий слот, так как для его управления требуется 10 цифровых контактов. При таком соединении управление каждым сегментом происходит статично.

Печатная плата модуля имеет размеры 35.4x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 21.

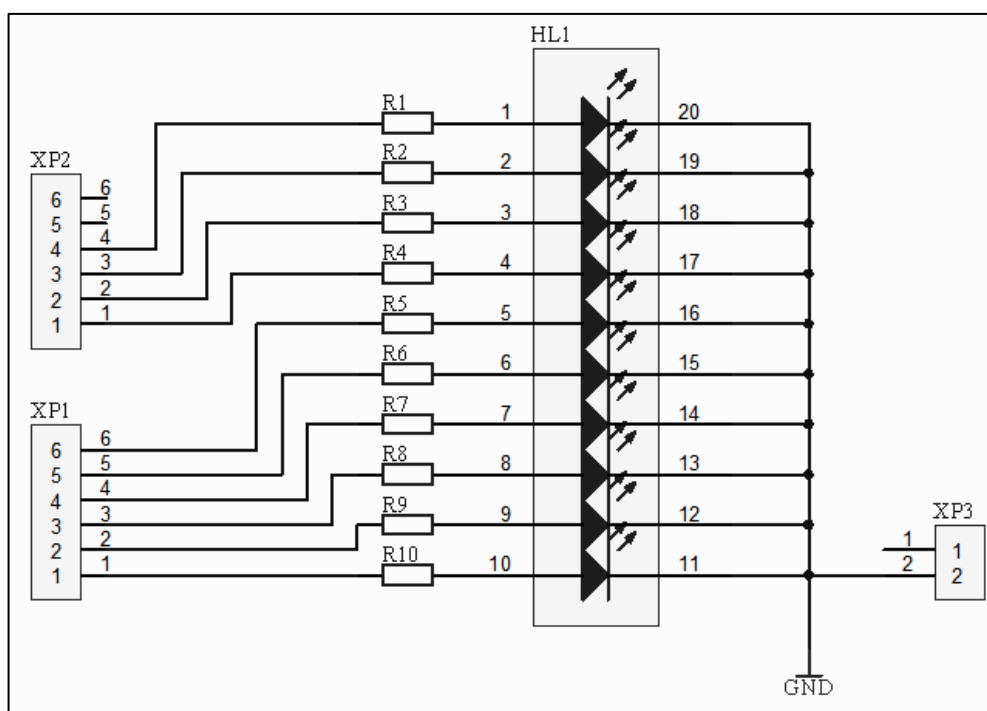


Рисунок 21 – Принципиальная электрическая схема модуля

Чертеж печатной платы на основе принципиальной электрической схемы представлен на рисунке 22.



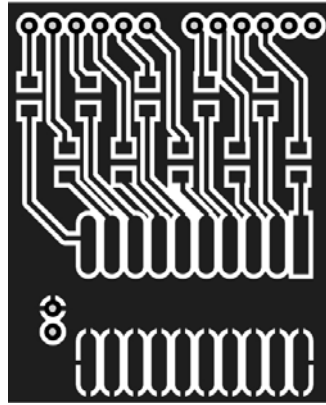


Рисунок 22 – Чертеж печатной платы. Верхний слой

Сборочный чертеж представлен на рисунке 23.

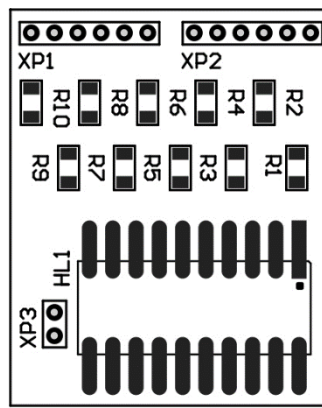


Рисунок 23 – Сборочный чертеж модуля

### 2.2.7 Модуль ШИМ

Широтно-Импульсная Модуляция – это управление мощностью путем изменения скважности импульсов. В Arduino есть контакты, которые могут выводить ШИМ сигналы, они представляются в быстром переключении между 0 и +5 вольт, это происходит настолько быстро, что человеческий глаз не успевает заметить. Если к такому контакту присоединить светодиод, то будет казаться, что он горит в пол силы, в зависимости от программы.

Модуль имеет 4 выводных светодиода [9]. Катоды светодиодов соединены вместе и подключены к нулю на платформе. Аноды подключены к цифровым контактам 6,9,10,11 через резистор 200ом [7]. Эти контакты имеют ШИМ подключения в Arduino. Сам модуль может быть установлен только во

второй слот. При необходимости данный модуль можно использовать для статической индикации.

Печатная плата модуля имеет размеры 17.7x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 24.

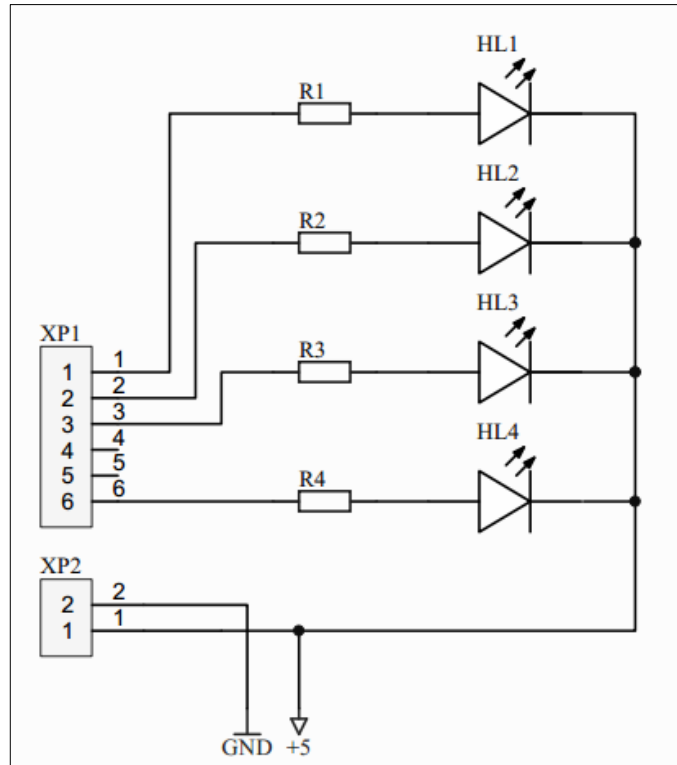


Рисунок 24 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 25.



Рисунок 25 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 26.

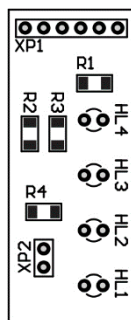


Рисунок 26 – Сборочный чертёж модуля

## 2.2.8 Модуль потенциометр

Потенциометр – регулируемый делитель электрического напряжения, или переменный резистор [8].

Модуль использует один слот и может быть использован в любом из трёх слотов, так как использует универсальный вход (+5, 0, аналоговый вывод). Установленный потенциометр имеет сопротивление 10Ком.

Печатная плата модуля имеет размеры 17.7x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 27.

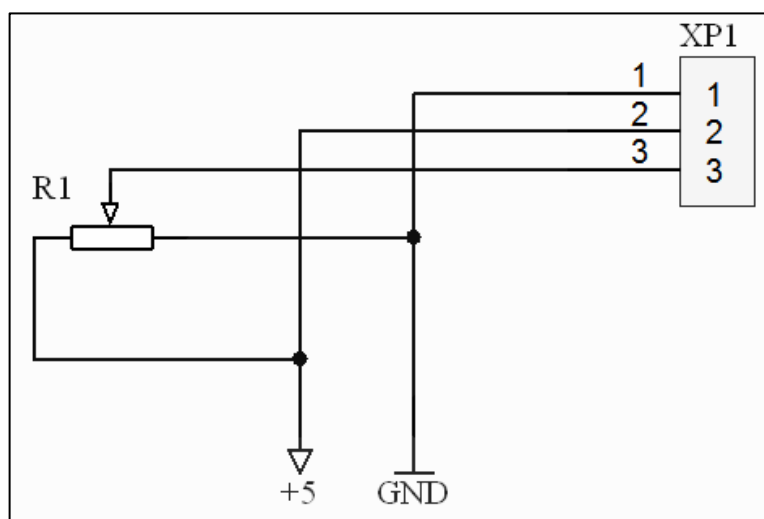


Рисунок 27 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 28.

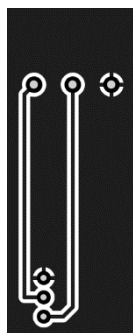


Рисунок 28 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 29.

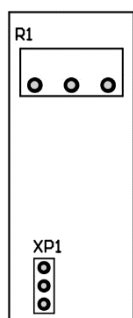


Рисунок 29 – Сборочный чертёж модуля

### 2.2.9 Модуль датчика температуры и давления

В качестве измерения температуры и давления используется датчик DHT22 [12], он имеет следующие характеристики:

- питание от 3 до +5 вольт;
- максимально потребляемый ток – 2.5мА;
- рассчитан на измерение уровня влажности в диапазоне от 0% до 100%, точность измерений находится в диапазоне 2%-5%;
- измеряет температуру в диапазоне от - 40 до 125 градусов с точностью плюс-минус 0.5 градусов по Цельсию;
- частота измерения датчика 1 раз в 2 секунды.

В модуле, кроме самого датчика, используется подтягивающий резистор 10кОм [7], для того что бы гарантировать на логическом входе высокий или низкий уровень. Подключается аналогично модулю статических кнопок.

Печатная плата модуля имеет размеры 17.7x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 29.

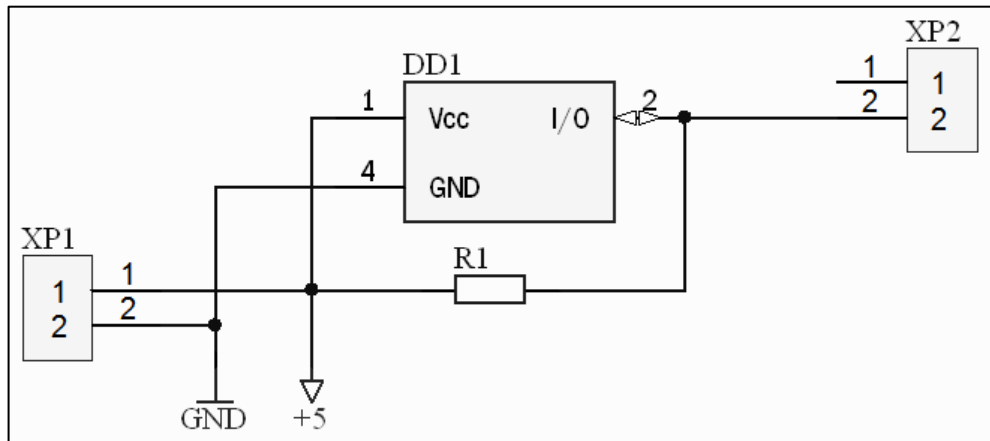


Рисунок 29 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 30.



Рисунок 30 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 31.

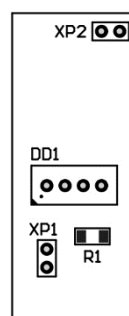


Рисунок 31 – Сборочный чертёж модуля

### 2.2.10 Модуль сдвигового регистра

Сдвиговый регистр выполнен на микросхеме 74НС595D [10], которая позволяет, используя три вывода микроконтроллера, управлять восемью кон-

тактами на одном сдвиговом регистре. Из данной микросхемы можно делать каскады, подключая их последовательно (через вывод Q7), и таким образом можно получить множество выводов. Микросхема имеет следующие характеристики:

- питание от 2 до +6 вольт;
- максимальный выходной ток 75 мА на весь регистр;
- рабочая частота 2-100 МГц.

Распиновка входов/выходов регистра на микросхеме 74hc595d представлена в таблице 1

Таблица 1 – Распиновка входов/выходов 74HC595D

| Номер контакта | Обозначение | Назначение  |
|----------------|-------------|---|
| Пины 1-7, 15   | Q0 " Q7     | Параллельные выходы   |
| Пин 8          | GND         | Земля   |
| Пин 9          | Q7"         | Выход для последовательного соединения регистров                  |
| Пин 10         | MR          | Сброс значений регистра. Сброс происходит при получении LOW       |
| Пин 11         | SH_CP       | Вход для тактовых импульсов                                       |
| Пин 12         | ST_CP       | Синхронизация ("защелкивание") выходов                            |
| Пин 13         | OE          | Вход для переключения состояния выходов из высокоомного в рабочее |
| Пин 14         | DS          | Вход для последовательных данных                                  |
| Пин 16         | Vcc         | Питание   |

Также в данном модуле используется четырёхразрядный семисегментный индикатор. Для отображения каждого из разрядов используется общий катод, а для того что бы разряд не загорался, используется физическое явление светодиода и подаётся на катод логическая единица. А аноды подключены к цифровым выводам сдвигового регистра.

Модуль использует только один слот: второй или третий, при использовании во втором слоте он перекрывает третий из-за габаритов платы.

Печатная плата модуля имеет размеры 68x58.4 мм. Большинство разводки выполнено на верхнем слое, а также несколько соединений выполнены на нижнем слое.

Принципиальная электрическая схема представлена на рисунке 32.

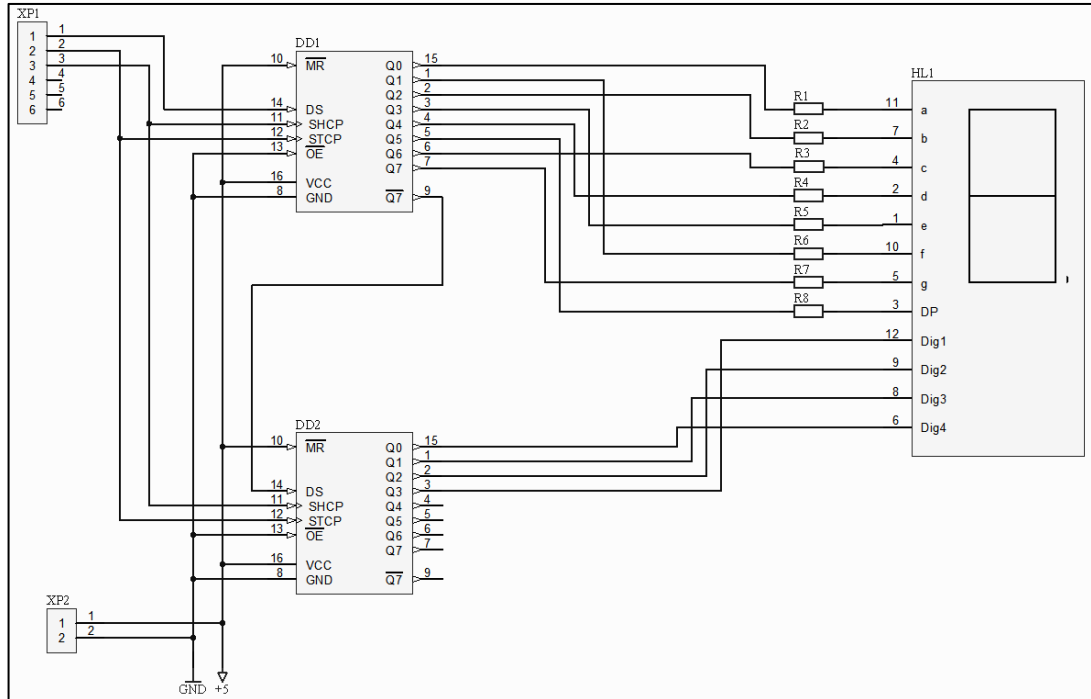


Рисунок 32 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунках 33 и 34.

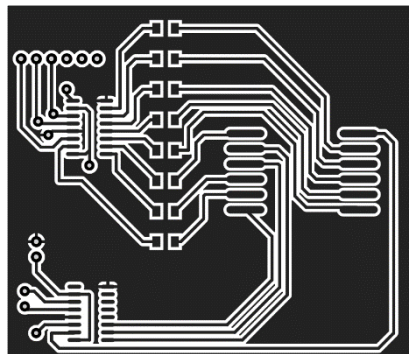


Рисунок 33 – Чертёж печатной платы. Верхний слой

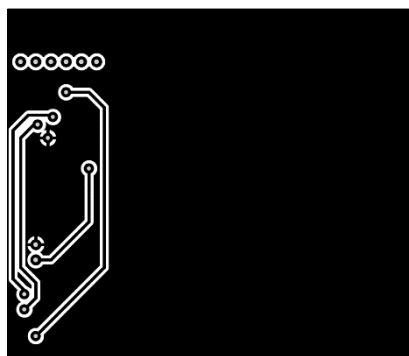


Рисунок 34 – Чертёж печатной платы. Нижний слой

Сборочный чертёж представлен на рисунке 35.

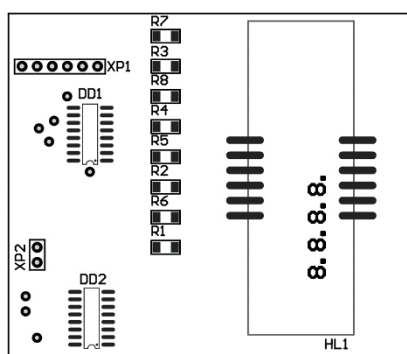


Рисунок 35 – Сборочный чертёж модуля

### 2.2.11 Модуль I2C

В качестве готового внешнего устройства был использован LCD экран 1602, подключенный через драйвер, который общается с микроконтроллером по протоколу i2c и выводит информацию на экран через его стандартный ввод. Данный экран позволяет вводить 16 символов в двух строках. Разрешение символов – 5x8 точек.

Модуль представляет из себя только переходник для данного готового модуля. Может подключаться только к первому слоту. Но при наличии соединительных проводов в данный модуль можно установить любое устройство, которое может работать с протоколом I2C.

Печатная плата модуля имеет размеры 17.7x43.1 мм.

Принципиальная электрическая схема представлена на рисунке 36.



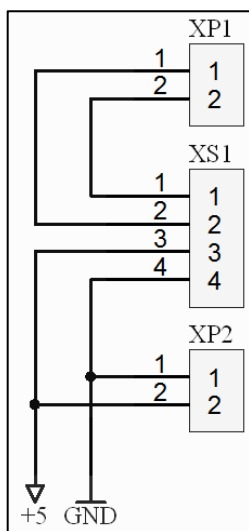


Рисунок 36 – Принципиальная электрическая схема модуля

Чертёж печатной платы на основе принципиальной электрической схемы представлен на рисунке 37.

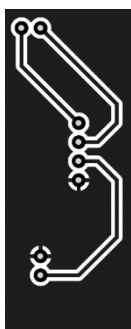


Рисунок 37 – Чертёж печатной платы. Верхний слой

Сборочный чертёж представлен на рисунке 38.

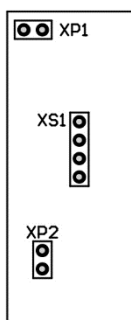


Рисунок 38 – Сборочный чертёж модуля

### 2.3 Разработка методических указаний к лабораторным работам

При работе с разработанными устройствами требуются определенные навыки и умения. Для их формирования было разработано методическое обеспечение. Лабораторные работы включают в себя: наглядные схемы, таблицы подключения, рисунки с использованием модулей, код программы.

Методическое обеспечение применяется при проведении лабораторных работ в рамках дисциплины «Микропроцессорная техника».

Данная дисциплина разделяется на 4 раздела:

1. Общие вопросы организации микропроцессорных систем.
2. Взаимодействие микропроцессорных компонентов.
3. Основные компоненты микропроцессорных систем.
4. Конструирование и эксплуатация микропроцессорных систем.

Лабораторные работы рассчитаны на 50 академических часов. В рамках лабораторной работы обучающиеся визуально оценивают целостность устройств, производят подготовку устройств к проведению лабораторных работ, настраивают необходимое программное обеспечение, выполняют индивидуальные и самостоятельные работы, оформляют отчет о проделанной работе, отвечают на контрольные вопросы. Темы лабораторных работ представлены в таблице 2.

Таблица 2 – Лабораторные работы

| №  | № раз-дела | Тема лабораторной работы                                    | Количество часов |
|----|------------|---|------------------|
| 1  | 1          | Микроконтроллер Arduino Uno                                 | 2                |
| 2  | 2          | Последовательный порт «Serial»                              | 5                |
| 3  | 3          | Семисегментный и шкальный статический индикатор             | 5                |
| 4  | 3          | Кнопки, статический ввод                                    | 4                |
| 5  | 3          | Матрица из светодиодов, динамическая индикация              | 6                |
| 6  | 3          | Семисегментный индикатор, подчинённый к сдвиговому регистру | 6                |
| 7  | 3          | Матрица из кнопок, динамический ввод                        | 6                |
| 8  | 3          | Светодиодная индикация, подключенная к ШИМ                  | 4                |
| 9  | 3          | Измерение температуры и влажности воздуха [14]              | 6                |
| 10 | 3          | Дисплей, подключенный с помощью протокола I2C               | 6                |

### **Лабораторная работа №1 – Микроконтроллер «Arduino UNO»**

Цель работы:

Ознакомиться с микроконтроллером «Arduino UNO» и работой индикаторного светодиода.

Теоретические сведения:

Arduino Uno – это открытая платформа, построенная на базе микроконтроллера «ATmega328» с частотой 16 МГц и памятью 32 кБ.

Подача напряжения для работы платы осуществляется через круглый разъём либо через порт USB. Напряжение питания платы Arduino, подаваемое

мое через круглый разъём может варьироваться от 5 – 12 Вольт. Плата Arduino имеет встроенный стабилизатор напряжения на 5 Вольт.

На данной платформе имеются 14 цифровых контактов(пин/вывод), все они могут выдавать напряжение от 0 (логический ноль) вольт до напряжения питания (логическая единица). Максимальная сила тока, выдаваемая каждым пином 40 мА. Некоторые из них имеют дополнительные функции:

1. Пины 0 и 1 могут использоваться для приёма передачи данных по USB.

2. Пины 2 и 3 могут использовать функцию внешних прерываний.

3. Пины 3,5,6,9,10,11 могут являться выходами с широтно-импульсной модуляцией.

4. Пин 13 – светодиодный индикатор, расположенный на плате (при подаче на данный контакт логической единицы индикатор загорается, при подаче логического нуля гаснет).

Некоторые выводы (обозначенные буквой А) подключены к мультиплексору АЦП, разрешение 10 бит (может принимать 1024 различных значений). По умолчанию у выводов диапазон измерения от 0 до 5 вольт, так же есть возможность изменить верхнюю границу путем подачи требуемого напряжения на контакт AREF.

Среда разработки Arduino (рисунок 39) представляет собой: текстовый редактор программного кода, область сообщений, консоли, панель инструментов и вкладки меню.

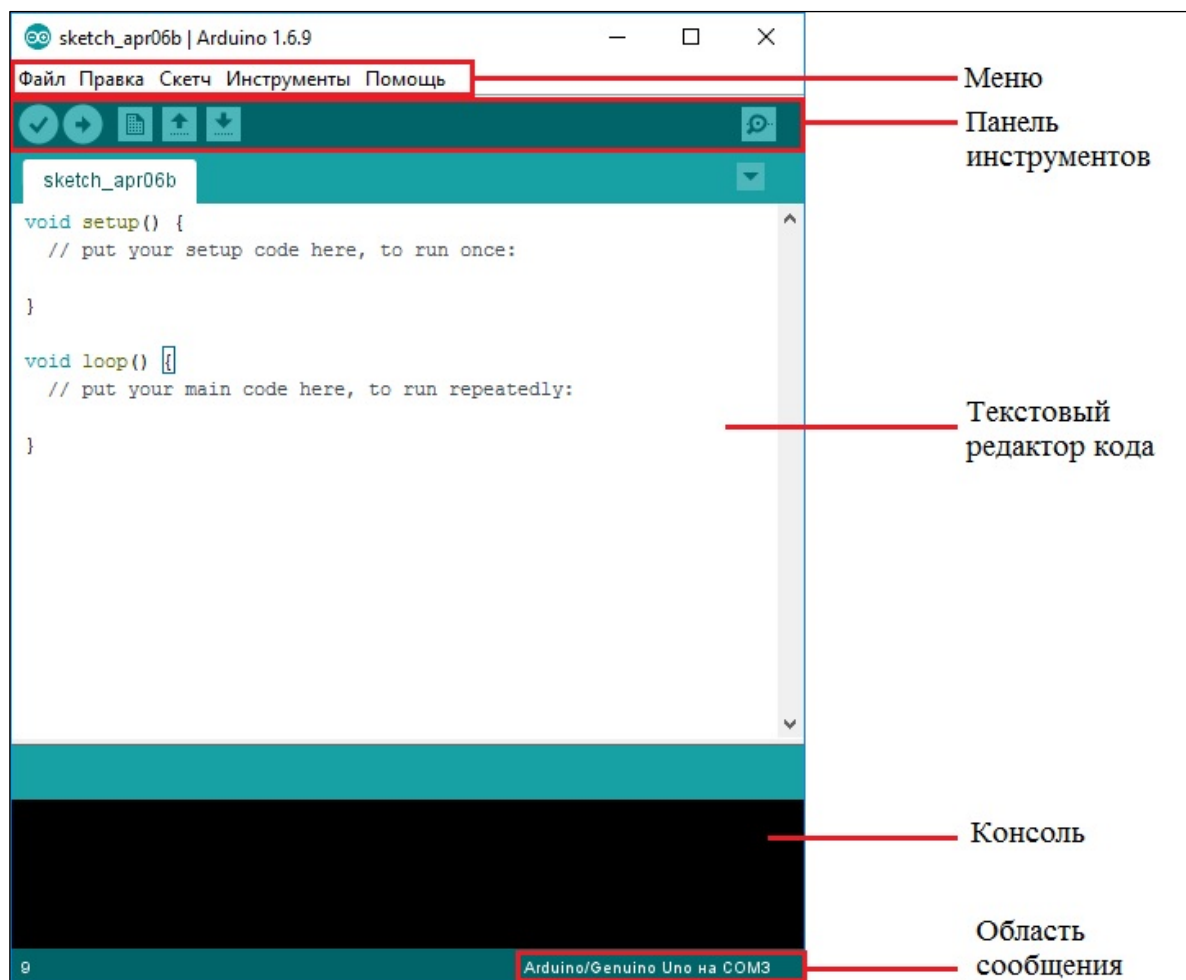


Рисунок 39 – Окно среды разработки

В данной лабораторной работе используются следующие функции:

1. Void setup() – используется для инициализации переменных, определений работы выводов и т.д. Данная функция используется только один раз после каждой подачи питания или сброса платы.

2. Внутри функции void setup используется функция pinMode(контакт, режим) которая служит для конфигурирования режима работы контакта. Режим работы может принимать значения OUTPUT(выход), INPUT(вход) или INPUT\_PULLUP (вход с программным подключением внутреннего подтягивающего резистора).

3. После выполнения функции setup начинается основная функция void loop(), которая выполняет основной код, выполняемый в цикле.

4. Функция `digitalWrite` (контакт, значение) отправляет на цифровой порт значение, которое было заранее сконфигурировано: `HIGH(+5V)` или `LOW(0V)`.

При быстром включении и выключении светодиода без задержки человеческий глаз не увидит изменения, для этого используем функцию задержки `delay`(время в мс), для того чтобы можно было заметить изменение светодиода.

#### Ход работы

Настроить среду разработки Arduino для загрузки программного кода (скетч).

Для этого выполняем следующий алгоритм работы:

1. Зайти в выпадающее меню инструменты, выбрать пункт плата, и выбрать «Arduino/Genuino Uno».
2. В выпадающее меню инструменты, выбрать пункт порт, и выбрать последовательный порт (COM), к которому подключена наша плата.
3. В окно кода ввести скетч (рисунок 40), и проверить его с помощью кнопки «проверить» на панели инструментов или с помощью горячей клавиши «Ctrl+R».

```
void setup()
{
  pinMode(13, OUTPUT); // настраиваем пин №13 в режим выхода
}

void loop() //цикл основной программы
{
  digitalWrite(13, HIGH); // подать на пин 13 «высокий сигнал»
  delay(500); //задержка на 500 миллисекунд
  digitalWrite(13, LOW); // подать на пин 13 «низкий сигнал»
  delay(500); //задержка на 500 миллисекунд
}
```

Рисунок 40 – Код программы для встроенного светодиода

Если после проверки не обнаружилось ошибок в окне консоли, то следует нажать кнопку «загрузка» на панели управления или с помощью горячей клавиши «Ctrl+U», загрузить программы на платформу. При загрузке на

плату Arduino мигают светодиоды «TX/RX». После окончания загрузки написанная программа автоматически начинает выполняться.

Вопросы для самоконтроля:

1. Что выполняет функция `void setup()`?
2. Что делает функция `digitalWrite(13, LOW)`?
3. Что делает функция `delay(500)`?
4. Что делает функция `pinMode(13, OUTPUT)`?
5. Что означает режим работы `INPUT_PULLUP`?
6. Что означает режим работы `OUTPUT`?

## **Лабораторная работа №2 – Последовательный порт «Serial»**

Цель работы:

Ознакомиться с набором функций «Serial» для подключения Arduino UNO к компьютеру.

Теоретические сведения:

Набор функций «Serial» служит для связи платформы с компьютером или другими устройствами, поддерживающими последовательный интерфейс обмена данными. Arduino UNO имеет один последовательный порт. Для обмена данными последовательного интерфейса используют цифровые порты 0 (RX) и 1 (TX), а также USB-порт. Важно учитывать, что если вы используете функции «Serial», то нельзя одновременно с этим использовать порты 0 и 1 для других целей.

Среда разработки имеет встроенный монитор последовательного интерфейса (Serial monitor). Для начала обмена данными необходимо запустить монитор нажатием кнопки «Serial monitor» и выставить ту же скорость связи, с которой вызвана инициализация последовательного интерфейса `begin ()`.

При использовании монитора последовательного интерфейса нужно учитывать, что при передаче информации с платформы на компьютер можно посылать любую информацию. А при передаче с компьютера на плату, передача символов происходит в соответствии с таблицей ASCII. Значит, что при отправке в монитор последовательного интерфейса символ «1», отправляет

данный символ в COM-порт в двоичном виде «00110001» (или «49» в десятичном)

В данной лабораторной работе используются следующие функции передачи данных:

1. `Serial.begin` (значение) – иницирует последовательное соединение и задает скорость передачи данных.

2. `Serial.available` () – получает количество байт(символов) доступных для чтения из последовательного интерфейса.

3. `Serial.read` () – считывает очередной доступный байт из буфера последовательного соединения.

4. `Serial.println` () – передает данные через последовательное соединение, как ASCII текст с следующим за ним символом переноса строки.

В данной лабораторной работе используется следующий оператор:

1. `If` (условие) – используется в сочетании с операторами сравнения, проверяет, достигнута ли истинность условия, например, превышает ли входное значение заданное число.

Ход работы

1. Написать скетч (рисунок 41) для отображения отправленной информации через монитор последовательного интерфейса.

2. Проверить правильность написанного программного кода и записать на платформу.

3. Проверить правильность выполнения программы в выводе сообщения, в мониторе последовательного интерфейса.



```

void setup() {
  Serial.begin(9600);
}

void loop() {
  int buffer[1000];
  int i = 0;
  int rez = 0;
  if (Serial.available()) {

    while (Serial.available() && i < 999)
    {
      buffer[i++] = Serial.read();
    }
    buffer[i++] = '\0';
    rez = Serial.read();
    Serial.println(rez);
  }
}

```

Рисунок 41 – Код программы для последовательного интерфейса

Индивидуальное задание:

Вывести в последовательный порт сообщение вида «Студент: [Фамилия Имя], группа: [Номер группы]», а при вводе сообщения в СОМ-порт приходило то же сообщение, что посылали с компьютера без изменения.

Самостоятельное изучение:

С помощью набора функций «Serial» реализовать задачу, задать время через окно последовательного интерфейса. После настройки времени, вывести его.

Вопросы для самоконтроля:

1. Что такое последовательный порт?
2. Как последовательный порт отправляет сообщения?
3. Что такое монитор последовательного интерфейса?
4. Что делает оператор if?
5. Что делает функция Serial.read?
6. Что делает функция Serial.available ()?

## Лабораторная работа №3 – Семисегментный и шкальный статический индикатор

Цель работы:

Ознакомиться с работой семисегментного и шкального индикатора, подключенного без мультиплексирования.

Теоретические сведения:

Семисегментный индикатор представляет собой устройство для отображения арабских цифр с точкой. Внешний вид индикатора состоит из восьми элементов светодиодной индикации – семь сегментов и точка (рисунок 42). При включении комбинаций сегментов можно составить арабские цифры.

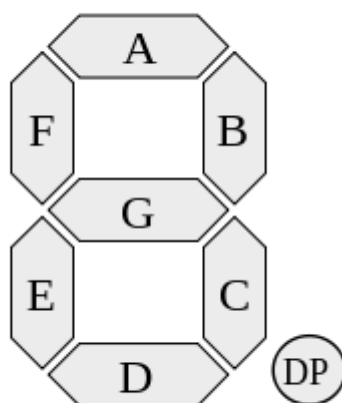


Рисунок 42 – Обозначение сегментов индикатора

Шкальный индикатор – это устройство, состоящее из нескольких независимых светодиодов, расположенных в форме шкалы и объединённых в один корпус. На каждый светодиод имеется 2 ножки (анод и катод). Основное назначение данного индикатора – это отображение уровня сигнала в различных устройствах.

В данной лабораторной работе используются следующие аппаратные модули:

1. Плата расширения (рисунок 43) представляет собой небольшую плату для подключения в платформу Arduino, и состоит из 3-х мест под модули. Для подключения позиционируем контакты в платформу таким образом, чтобы все контакты платы расширения подключались к контактам Arduino. Расположить платформу следует портом USB влево, а плату расширения горизонтальными контактами вверх, а вертикальными вниз.

(Платформу с USB-портом следует расположить слева, плату расширения с горизонтальными контактами сверху, а с вертикальными снизу).

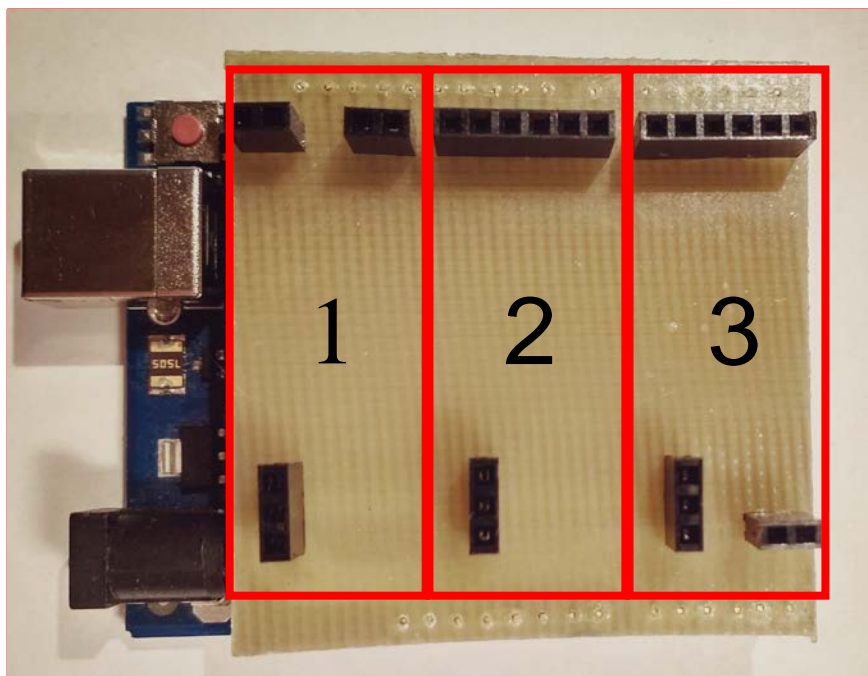


Рисунок 43 – Установка платы расширения

2. Модуль семисегментного индикатора подключают к плате расширения, в два правых горизонтальных модуля (рисунок 44). При таком расположении данный модуль занимает 2 слота. Точка (DP) у семисегментного индикатора находится снизу.

Схема соединения сегментов представлены в таблице 3.

Таблица 3 – Подключение сегментов

|         |   |   |   |    |    |   |   |    |
|---------|---|---|---|----|----|---|---|----|
| сегмент | A | B | C | D  | E  | F | G | DP |
| контакт | 5 | 6 | 9 | 10 | 11 | 7 | 8 | 4  |

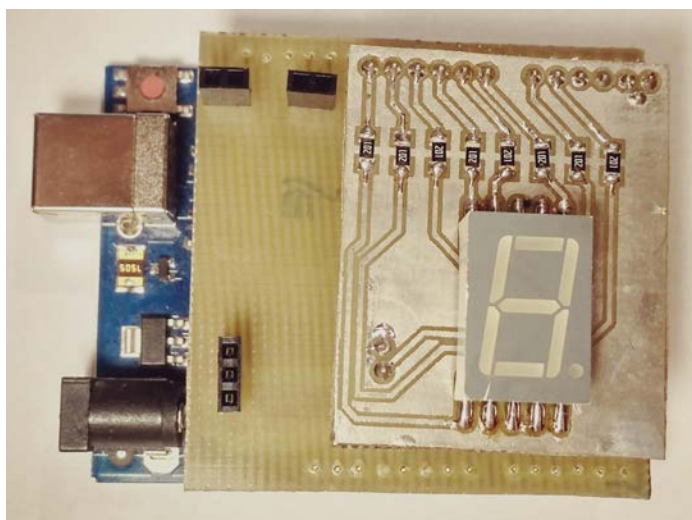


Рисунок 44 – Подключение модуля семисегментного индикатора

Модуль светодиодной шкалы (рисунок 45) подключается также, как и семисегментный индикатор и занимает 2 слота.

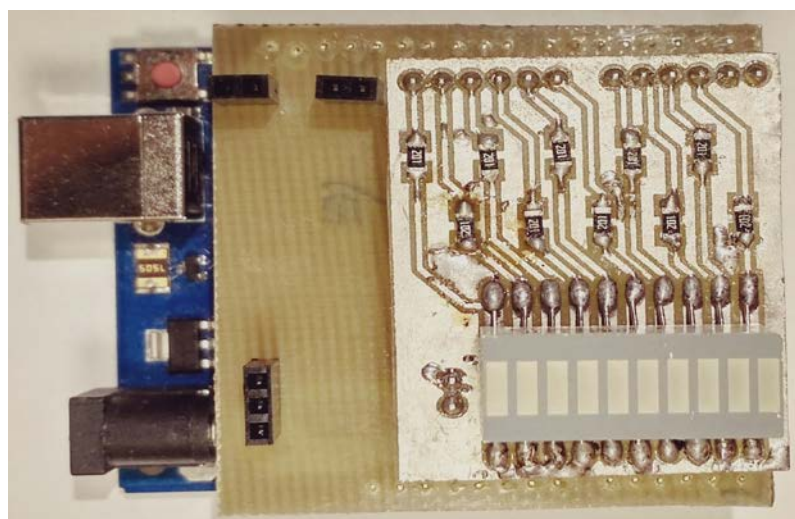


Рисунок 45 – Подключение модуля шкалы

Схема соединения сегментов шкалы слева на право, представлены в таблице 4.

Таблица 4 – Подключение сегментов шкалы

|         |    |    |   |   |   |   |   |   |   |   |
|---------|----|----|---|---|---|---|---|---|---|---|
| сегмент | A  | B  | C | D | E | F | G | H | I | J |
| контакт | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |

В данной лабораторной работе используются следующие операторы:

1. Switch-case – сравнивает значения переменной (var) со значением в case, а если не находит значение переменной, то использует код по умолчанию, если он присутствует. Case разделяются словом break (рисунок 46).

```
switch (var) {
  case 1:
    //выполняется, когда var равно 1
    break;
  case 2:
    //выполняется когда var равно 2
    break;
  default:
    // выполняется, если не выбрана ни одна альтернатива
    // default необязателен
}
```

Рисунок 46 – Синтаксис Switch

В данной лабораторной работе используется следующий цикл:

1. Цикл while (выражение) – это цикл который выполняется до тех пор, пока значение выражения не станет ложным. Проверять значение каждый раз в начале выполнения цикла. Если значение выражения никогда не будет ложным, то цикл будет бесконечно повторяться.

Ход работы

Работа с модулем семисегментного индикатора:

1. Написать программу счетчика (рисунок 47) вывода арабских цифр с использования оператора switch.

2. Проверить правильность написанного программного кода и записать на платформу.

3. Проверить правильность выполнения программы счетчика вывода арабских цифр.

|   |   |
|---|---|
| <pre> void setup() {   pinMode (4, OUTPUT);   digitalWrite (4, HIGH);   ...   pinMode (11, OUTPUT);   digitalWrite (11, HIGH); } int i; void loop() {   while (i&lt;=19){     switch (i) {       case 0:         digitalWrite (4, HIGH);         digitalWrite (5, LOW);         digitalWrite (6, LOW);         digitalWrite (7, LOW);         digitalWrite (8, HIGH);         digitalWrite (9, LOW);         digitalWrite (10, LOW);         digitalWrite (11, LOW);       </pre> | <pre>         break;         ...       case 9:         digitalWrite (4, HIGH);         digitalWrite (5, LOW);         digitalWrite (6, LOW);         digitalWrite (7, LOW);         digitalWrite (8, LOW);         digitalWrite (9, LOW);         digitalWrite (10, LOW);         digitalWrite (11, HIGH);         break;       }     }     delay (200);     i++;   }   i = 0; } } </pre> |
|---|---|

Рисунок 47 – Код программы для модуля светодиодной шкалы

Работа с модулем светодиодной шкалы:

1. Написать программу (рисунок 48) увеличения уровня и его повтора.

|   |  |
|---|--|
| <pre> void setup() {   pinMode (2, OUTPUT);   digitalWrite (2, LOW);   pinMode (3, OUTPUT);   digitalWrite (3, LOW);   pinMode (4, OUTPUT);   digitalWrite (4, LOW);   pinMode (5, OUTPUT);   digitalWrite (5, LOW);   pinMode (6, OUTPUT);   digitalWrite (6, LOW);   pinMode (7, OUTPUT);   digitalWrite (7, LOW);   pinMode (8, OUTPUT);   digitalWrite (8, LOW);   pinMode (9, OUTPUT);   digitalWrite (9, LOW);   pinMode (10, OUTPUT);   digitalWrite (10, LOW);   pinMode (11, OUTPUT);   digitalWrite (11, LOW); } </pre> | <pre> void loop() {   int val = 11;   int i = 0;   while (i &lt;= 20) {     if (i &lt; 10) {       digitalWrite(val, HIGH);       val--;     } else {       digitalWrite(val, LOW);       val++;     }     delay (100);     i++;   }   delay (500); } </pre> |
|---|--|

Рисунок 48 – Код программы для модуля светодиодной шкалы

2. Проверить правильность написанного программного кода и записать на платформу.

3. Проверить правильность выполнения программы, увеличение уровня.

Индивидуальное задание с модулем семисегментного индикатора:

Показать на модуле семисегментного индикатора последнюю цифру номера своего студенческого билета.

Индивидуальное задание с модулем шкального индикатора по вариантам (вариант спросить у преподавателя), вывода уровня наполнения:

1. Слева направо с фиксацией уровня в конце.
2. Справа налево с фиксацией уровня в конце.
3. Справа налево без фиксации уровня в конце.
4. Расхождения из середины в края без фиксации уровня в конце.
5. Расхождения из середины в края с фиксацией уровня в конце.
6. Схождения с края в середину с фиксацией уровня в конце.

Самостоятельное изучение для модуля семисегментного индикатора:

Выводить цифры от 0 до 9, которые отправляешь через монитор последовательного порта.

Самостоятельное изучение для модуля шкального индикатора:

Включать только те сегменты, которые отправляешь в монитор последовательного порта (A-J), при повторном вводе сегмента выключать.

Вопросы для самоконтроля:

1. Что такое семисегментный индикатор?
2. Что такое шкальный индикатор?
3. Как подключается модуль семисегментного индикатора в плату расширения?
4. Как подключается модуль шкального индикатора в плату расширения?
5. Как подключен сам семисегментный индикатор на плате?
6. Зачем нужен токоограничивающий резистор?

## Лабораторная работа №4 – Кнопки, статический ввод

Цель работы:

Ознакомиться с работой кнопок, подключенных статически

Теоретические сведения:

Тактильная кнопка – это механизм для коммуникации электрических цепей, построенных на том, что электрическая цепь будет замыкаться только тогда, когда есть давление на толкатель.

Кнопки подразделяются на несколько видов:

1. Нормально разомкнутые – в пассивном состоянии имеют разомкнутые контакты, в активном замкнутые.

2. Нормально замкнутые – в пассивном состоянии имеют замкнутые контакты, в активном разомкнутые.

Так же для всех видов есть тип переключения:

1. Без фиксации – при отпускании кнопки она переходит в пассивное состояние.

2. С фиксацией – при нажатии на кнопку она меняет свое положение в активное, при отпускании фиксирует состояние, а при повторном нажатии меняет свое положение на пассивное.

При использовании кнопок возникает явление, как дребезг контактов – это явление происходит в момент переключения кнопки, когда она замыкается или размыкается происходит множественное переключение за несколько миллисекунд.

А для подключения кнопок к Arduino нужно использовать подтягивающий резистор для уменьшения шумов на входном контакте.

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль статических кнопок (рисунок 49) подключается к плате расширения, во все модульные разъёмы, при использовании в третьем слоте не нужно использовать кнопки при загрузке программного кода в платформу Arduino.



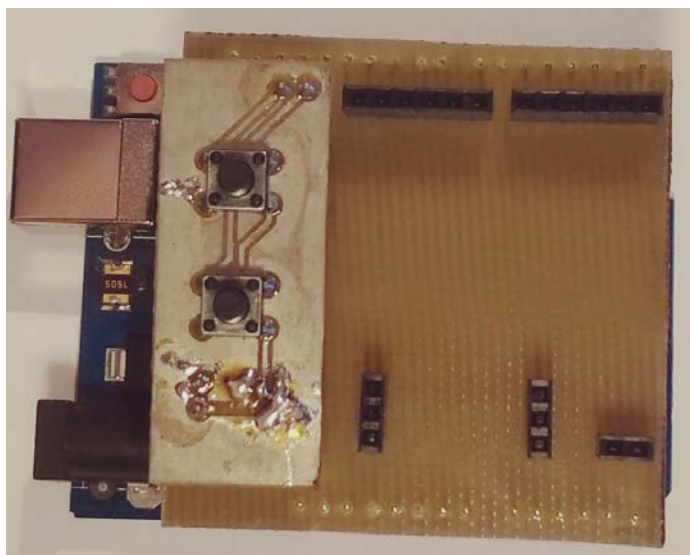


Рисунок 49 – Модуль статических кнопок в 1 слоте

Кнопки являются нормально разомкнутыми без фиксации. При подключении модуля в первый слот, верхняя кнопка подключается к цифровому контакту 13, а нижняя к 12. Второй контакт кнопок соединён с нулем на плате.

Ход работы

Работа с модулем статических кнопок:

1. Написать программу вывода в последовательный порт: название института - при нажатии на верхнюю кнопку и адреса - при нажатии на нижнюю (рисунок 50).
2. Проверить правильность написанного программного кода и записать на платформу.
3. Проверить правильность выполнения программы в мониторе последовательного интерфейса.

```

void setup() {
  pinMode (12, INPUT_PULLUP);
  pinMode (13, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop() {
  if (digitalRead (13) == 0) {
    Serial.println("RSVPU");
    delay (500);
  }
  if (digitalRead (12) == 0) {
    Serial.println("Mashinostroitelej 11");
    delay (500);
  }
}

```

Рисунок 50 – Код программы для кнопок

Индивидуальное задание:

Написать программу с выводом на верхнюю кнопку своей группы в формате «Группа: [номер группы]», при нажатии на нижнюю кнопку вывода имени и фамилии в формате «[Фамилия Имя]».

Самостоятельная работа:

Для самостоятельного изучения, используя модуль статических кнопок и семисегментного индикатора, реализовать задачу, в которой при нажатии на верхний контакт, значения счётчика увеличатся, а при нажатии на нижний уменьшатся.

Вопросы для самоконтроля:

1. Что такое дребезг контактов?
2. Зачем нужен подтягивающий резистор?
3. Что такое кнопка?
4. Что такое кнопка нормально разомкнутая?
5. Что такое фиксирующая кнопка?
6. В какой слот может подключаться модуль статических кнопок?

## Лабораторная работа №5 – Матрица из светодиодов, динамическая индикация

Цель работы:

Ознакомиться с работой матрицы светодиодов, с динамической индикацией.

Теоретические сведения:

Матрица из светодиодов представляет собой группу светодиодов, соединённых анодами в столбцах и общими катодами в рядах (рисунок 51). Этот способ позволяет уменьшить количество контактов для большего количества компонентов. Таким образом, используя матрицу из светодиодов 3x3, используется 6 контактов, а не 9.

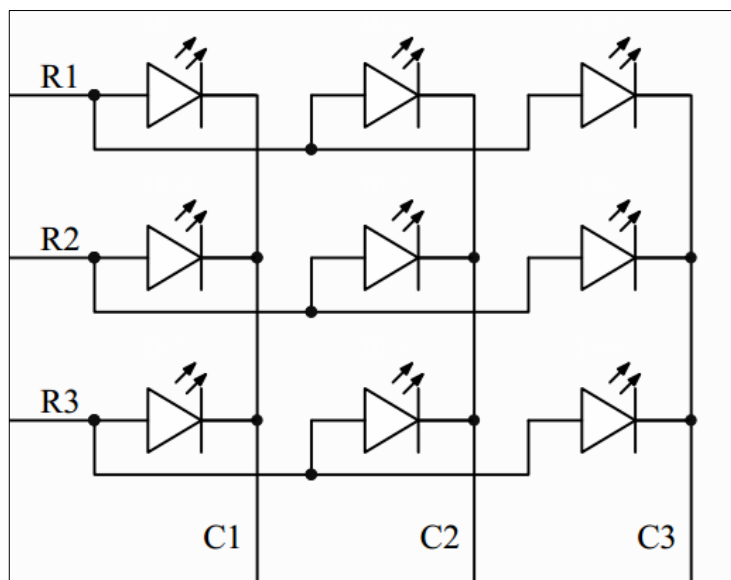


Рисунок 51 – Схема матрицы светодиодов

Чтобы получить изображение на таком устройстве нужно воспользоваться динамическим выводом: когда зажигается один ряд светодиодов, потом выключается и зажигается другой ряд. Но происходит это настолько быстро, что человеческий глаз не успевает заметить этого переключения, и кажется, что горит всё изображение.

При использовании матрицы используется физическое свойство светодиода, когда подаётся +5В на катод светодиода, он не загорается.

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль светодиодной матрицы (рисунок 52) подключается к плате расширения во 2 или 3 слот. Если использовать модуль во 2 слоте, то из-за размеров платы нет доступа к 3. А если использовать в 3 слоте, то надо будет каждый раз при записи кода в Arduino вынимать модуль, так как он может мешать записи кода.

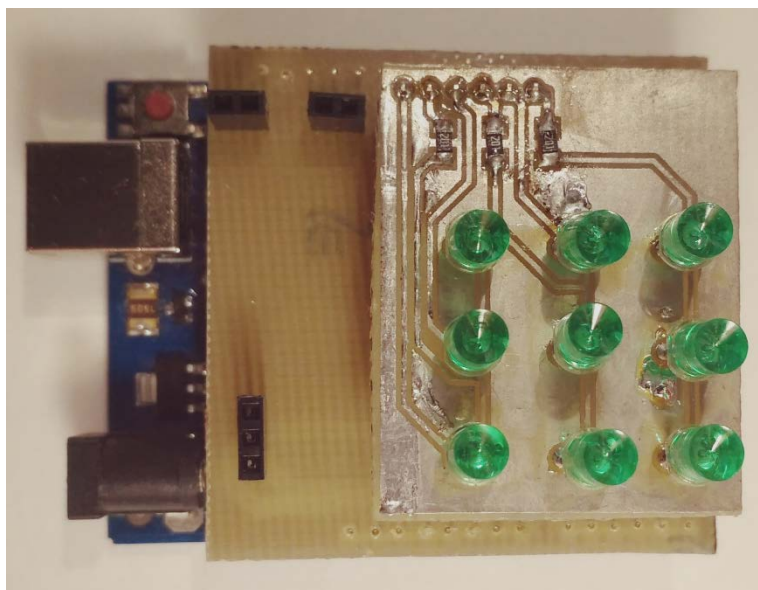


Рисунок 52 – Модуль светодиодной матрицы

При использовании во втором слоте контакты светодиодов подключены к платформе, как показано в таблице 5. При использовании в третьем слоте, таблица 6.

Таблица 5 – Схема подключение светодиодной матрицы в слоте 2

| Обозначение | R1 | R2 | R3 | C1 | C2 | C3 |
|-------------|----|----|----|----|----|----|
| Контакт     | 7  | 9  | 11 | 10 | 8  | 6  |

Таблица 6 – Схема подключение светодиодной матрицы в слоте 3

| Обозначение | R1 | R2 | R3 | C1 | C2 | C3 |
|-------------|----|----|----|----|----|----|
| Контакт     | 1  | 3  | 5  | 4  | 2  | 0  |

Колонки светодиодов подключены через токоограничивающий резистор 200 Ом, который защищает светодиод от перегорания.

В данной лабораторной работе используется следующая директива:

1. #define – эта директива позволяет дать имя константе, перед тем как программа будет скомпилирована. Определенные константы не занимают место в памяти Arduino, так как компилятор обращается к ним за значениями и служит для удобства программирования.

Можно сказать, что эта директива просто заменяет все имена на константу, обозначенную в ней.

Ход работы

Работа с модулем статических кнопок:

1. Написать программу вывода изображения снежинка (рисунок 53).

```
#define R1 7          void loop() {
#define R2 9          digitalWrite(R1, LOW);
#define R3 11         digitalWrite(R2, HIGH);
#define C1 10         digitalWrite(R3, HIGH);
#define C2 8          digitalWrite(C1, HIGH);
#define C3 6          digitalWrite(C2, LOW);
                     digitalWrite(C3, HIGH);
void setup() {       delay(1);
  pinMode(R1, OUTPUT); digitalWrite(R1, HIGH);
  pinMode(R2, OUTPUT); digitalWrite(C1, LOW);
  pinMode(R3, OUTPUT); digitalWrite(C2, HIGH);
  pinMode(C1, OUTPUT); digitalWrite(C3, LOW);
  pinMode(C2, OUTPUT); digitalWrite(R2, LOW);
  pinMode(C3, OUTPUT); delay(1);
}                   digitalWrite(R2, HIGH);
                     digitalWrite(C1, HIGH);
                     digitalWrite(C2, LOW);
                     digitalWrite(C3, HIGH);
                     digitalWrite(R3, LOW);
                     delay(1);
                     digitalWrite(R3, HIGH);
}                   }
```

Рисунок 53 – Код программы для матрицы из светодиодов

2. Проверить правильность написанного программного кода и записать на платформу.

3. Проверить правильность выполнения программы: на модуле должна гореть снежинка без искажений.

Индивидуальное задание по вариантам (вариант спросить у преподавателя). Выведите изображение:

1. Ромб.

2. Квадрат.
3. Стрелка вправо.
4. Стрелка влево.
5. Косая черта, которая смотрит вправо.
6. Косая черта, которая смотрит влево.

Самостоятельная работа:

Вывести на модуль светодиодной матрицы буквы «о, х, н, п, т, к, с и», при отправке таковой в монитор последовательного интерфейса.

Вопросы для самоконтроля:

1. Что значит матричный способ подключения?
2. Зачем нужен матричный способ подключения?
3. Для чего нужен токоограничивающий резистор?
4. Что такое физическое свойство светодиода?
5. Как выдается изображение в матричном способе?
6. Почему удобно использовать модуль во 2 слоте?

### **Лабораторная работа №6 – Семисегментный индикатор, подключенный к сдвиговому регистру**

Цель работы:

Ознакомиться с работой внешнего устройства сдвигового регистра на микросхеме 74НС595D.

Теоретические сведения:

Сдвиговый регистр – это устройство, в котором последовательно подключены в цепочку триггеры. Основная задача – это сдвиг разрядов кода, записанные в триггеры. По тактовому сигналу содержимое каждого предыдущего триггера перезаписывается в следующий, по цепочке.

Микросхема 74НС595D представляет собой 8-и битный сдвиговый регистр с выходным регистром-защелкой (таблица 7). Он позволяет контролировать 8 выходов, используя несколько выводов на микроконтроллере.

Таблица 7 – Распиновка входов/выходов регистра

|  |         |         |   |
|--|---------|---------|---|
|  | 1-7, 15 | Q0 " Q7 | Параллельные выходы   |
|  | 8       | GND     | Земля   |
|  | 9       | Q7"     | Выход для последовательного соединения регистров                  |
|  | 10      | MR      | Сброс значений регистра. Сброс происходит при получении LOW       |
|  | 11      | SH_CP   | Вход для тактовых импульсов                                       |
|  | 12      | ST_CP   | Синхронизация ("защелкивание") выходов                            |
|  | 13      | OE      | Вход для переключения состояния выходов из высокоомного в рабочее |
|  | 14      | DS      | Вход для последовательных данных                                  |
|  | 16      | Vcc     | Питание   |

Чип преобразует последовательный сигнал на контакте DS в параллельные выходы на контактах Q0-Q7. При использовании в больших количествах выводов, данные микросхемы можно подключить каскадом, и при выполнении последовательных данных на первом сдвиговом регистре, регистры передвигаются на второй, через контакт Q7". Так можно увеличить количество выводов, используя только 3 контакта Arduino. Последовательная передача синхронизирована тактами на контакте SH\_CP. Так же для одновременного вывода всех переданных данных используется защёлка ST\_CP.

Данная микросхема имеет следующие характеристики:

- напряжение питания от 2 до 6 вольт;
- рабочая частота от 2 до 100 МГц;
- максимальный выходной ток 75 мА на весь регистр.

Для демонстрации работы сдвиговых регистров к выходам подключен четырехразрядный семисегментный индикатор с общим катодом на каждый разряд. Индикация является динамической. Сначала загорается один сегмент, потом выключается, сегменты меняются и зажигается следующий. Происходит это настолько быстро, что человеческий глаз не успевает заметить.

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль сдвигового регистра (рисунок 54) может подключаться к плате расширения во 2 или 3 слот. Если использовать модуль во 2 слоте, то из-за размеров платы нет доступа к 3.

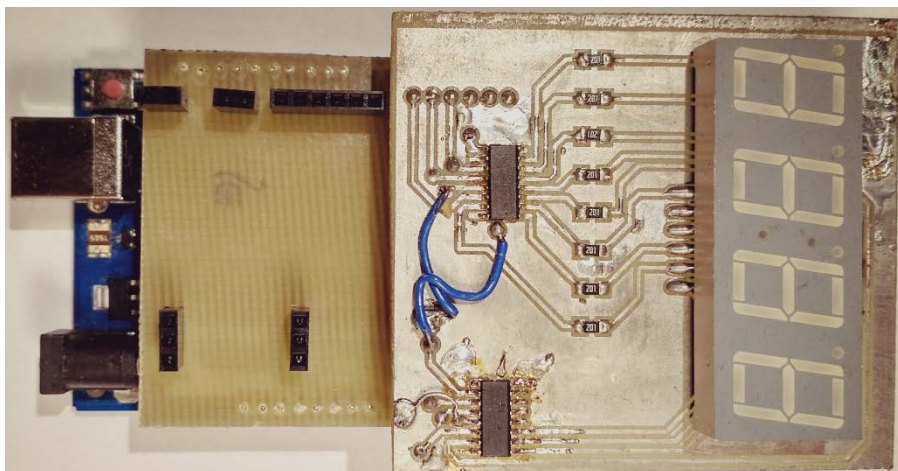


Рисунок 54 – Модуль сдвигового регистра

При использовании в третьем слоте контакты подключены к платформе, как показано в таблице 8, и во втором слоте в таблице 9.

Таблица 8 – Схема подключение модуля к третьему слоту

|            |       |       |    |
|------------|-------|-------|----|
| микросхема | SH_CP | ST_CP | DS |
| платформа  | 3     | 4     | 5  |

Таблица 9 – Схема подключение модуля к второму слоту

|            |       |       |    |
|------------|-------|-------|----|
| микросхема | SH_CP | ST_CP | DS |
| платформа  | 9     | 10    | 11 |

Анодные контакты сегментов индикатора подключены через токоограничивающий резистор.

В данной лабораторной работе используется следующая функция:

1. shiftOut (пин, порядок сдвига, данные) – эта функция, поочередно отправляет каждый бит на указанный вывод данных, после чего формирует импульс на тактовом выводе, сообщая внешнему устройству о поступлении нового бита.

Пин – вывод, куда отправляется бит.



Порядок сдвига – характеризует в каком порядке будут сдвигаться и выводиться биты. Принимает значения MSBFIRST (старший бит первым) или LSBFIRST (младший бит первым).

Данные – сдвигаемый байт данных.

Ход работы

Работа с модулем сдвигового регистра:

1. Написать программу вывода счетчика (рисунок 55).

```
int dataPin = 5;
int latchPin = 4;
int clockPin = 3;
char nomer;
void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

int i;
void loop() {
  while (i<9999)
  {
    showNumber(i);
    i++;
  }
  showNumber(i);
}

void showNumber(uint16_t n) {
  showSegment(0, n % 10);
  if (n > 9) {
    showSegment(1, (n / 10) % 10);
    if (n > 99) {
      showSegment(2, (n / 100) % 10);
      if (n > 999) {
        showSegment(3, (n / 1000) % 10);
      }
    }
  }
  showSegment(4, 999);
}

void showSegment(int segment, int num) {
  switch (num) {
    case 0:
      nomer = B11111010;
      break;
    case 1:
      nomer = B00100010;
      break;
    case 2:
      nomer = B10111001;
      break;
    case 3:
      nomer = B10101011;
      break;
    case 4:
      nomer = B01100011;
      break;
    case 5:
      nomer = B11001011;
      break;
    case 6:
      nomer = B11011011;
      break;
    case 7:
      nomer = B10100010;
      break;
    case 8:
      nomer = B11111011;
      break;
    case 9:
      nomer = B11101011;
      break;
    default:
      nomer = 0;
      break;
  }
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, MSBFIRST, ~(_BV(segment)));
  shiftOut(dataPin, clockPin, LSBFIRST, nomer);
  digitalWrite(latchPin, HIGH);
  delay(1);
}
```

Рисунок 55 – Код программы для сдвигового регистра

2. Проверить правильность написанного программного кода и записать на платформу.

3. Проверить правильность выполнения вывода счётчика на четырёх-разрядном семисегментном индикаторе до 9999.

Индивидуальное задание:

Для каждого написать программу с выводом последних четырех цифр своего студенческого билета на четырехразрядный семисегментный индикатор.

Самостоятельное изучение:

Используя модуль сдвигового регистра и модуль статических кнопок реализовать увеличение и уменьшение числа счётчика с выводом на четырехразрядный семисегментный индикатор.

Вопросы для самоконтроля:

1. Что такое сдвиговый регистр?
2. Что такое четырехразрядный семисегментный индикатор?
3. Для чего служит функция shiftOut?
4. Сколько выводов есть на микросхеме 74НС595D?
5. Какое максимальное число можно вывести на четырехразрядном семисегментном индикаторе?
6. Что за порядок сдвига MSBFIRST?

### **Лабораторная работа №7 – Матрица из кнопок, динамический ввод**

Цель работы:

Ознакомиться с работой матрицы из кнопок, динамическим вводом.

Теоретические сведения:

Матрица из кнопок представляет собой группу кнопок, соединённых одним контактом в столбцах, а другим в рядах (рисунок 56). Этот способ позволяет уменьшить количество контактов для большего количества компонентов. Таким образом, применяя матричный способ соединения 3х3, используется 6 контактов, а не 9.

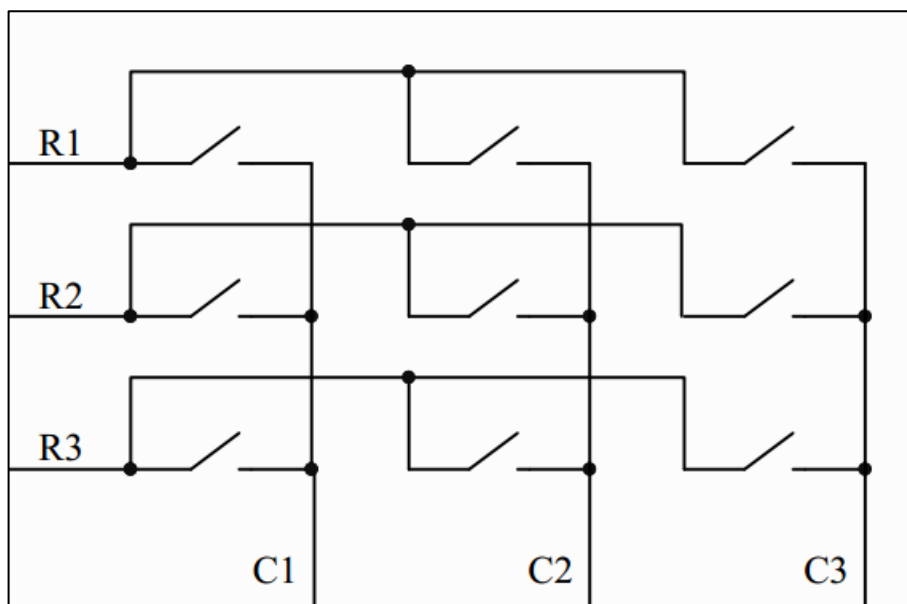


Рисунок 56 – Схема матрицы из кнопок

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль матрицы из кнопки (рисунок 57) подключается к плате расширения во 2 или 3 слот. Если использовать модуль во 2 слоте, то из-за размеров платы нет доступа к 3. А если использовать в 3 слоте, то нельзя использовать модуль, пока происходит запись кода в Arduino.

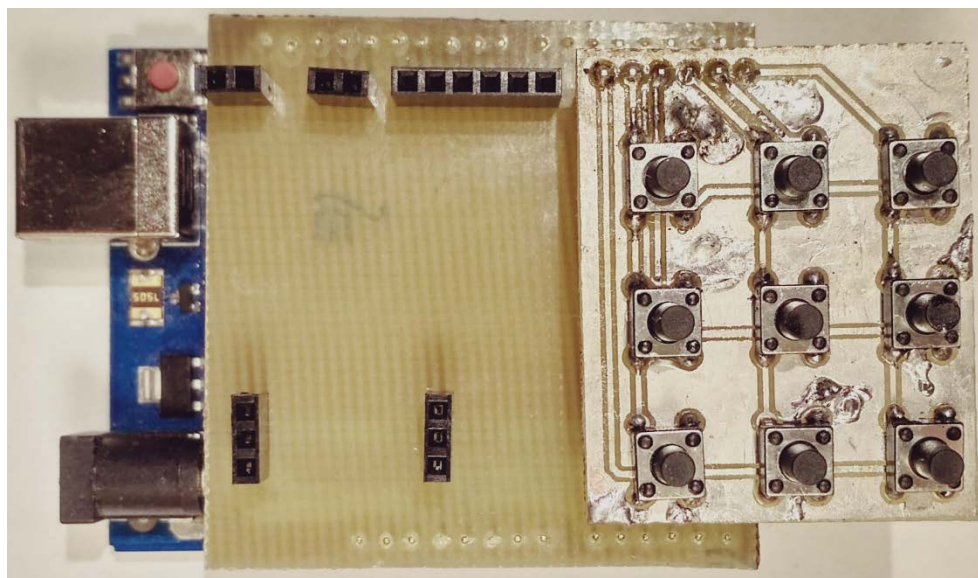


Рисунок 57 – Модуль статических кнопок в 1 слоте

Кнопки являются нормально разомкнутыми без фиксации. При использовании в третьем слоте контакты светодиодов подключены к платформе, как показано в таблице 10, а во втором слоте в таблице 11.

Таблица 10 – Схема подключение матрицы из кнопок, третий слот

| Обозначение | R1 | R2 | R3 | C1 | C2 | C3 |
|-------------|----|----|----|----|----|----|
| Контакт     | 1  | 3  | 5  | 4  | 2  | 0  |

Таблица 11 – Схема подключение матрицы из кнопок, второй слот

| Обозначение | R1 | R2 | R3 | C1 | C2 | C3 |
|-------------|----|----|----|----|----|----|
| Контакт     | 7  | 9  | 11 | 10 | 8  | 6  |

### Ход работы

Работа с модулем матрицы из кнопок:

1. Написать программу (рисунок 58) вывода в монитор последовательного интерфейса - название кнопки, которую нажимаешь.

```

void setup() {
  pinMode(C1, INPUT); //установка контакта C1 входом
  digitalWrite(C1, HIGH); //установка +5В, на контакте C1
  pinMode(C2, INPUT);
  digitalWrite(C2, HIGH);
  pinMode(C3, INPUT);
  digitalWrite(C3, HIGH);
  Serial.begin(9600);
}

void loop() {
  pinMode(R1, OUTPUT); //установка R1 выходом
  digitalWrite(R1, LOW); //установка на R1 низкого состояния
  int b1 = digitalRead(C1); //запись в переменную b1 состояние входа C1
  int b2 = digitalRead(C2); //запись в переменную b2 состояние входа C2
  int b3 = digitalRead(C3); //запись в переменную b3 состояние входа C3
  pinMode(R1, INPUT); //установка R1 входом*/

  pinMode(R2, OUTPUT);
  digitalWrite(R2, LOW);|
  int b1 = digitalRead(C1);
  int b2 = digitalRead(C2);
  int b3 = digitalRead(C3);
  pinMode(R3, OUTPUT);
  digitalWrite(R3, LOW);
  int b1 = digitalRead(C1);
  int b2 = digitalRead(C2);
  int b3 = digitalRead(C3);
  pinMode(R3, INPUT);

  if (b11 != b1) { //условие что b11 не равно b1
    b11 = b1; //если условие верно то в b11 записываем данные b
    if (b11 == 0) //условие проверки что в b11 записано 0 те кнопка замкнута
    {
      Serial.println("Кнопка 1"); //вывод монитор порта сообщения «Кнопка 1»
    }
  }
}
... //аналогично кнопке 1 записывается программа для следующих кнопок
} //конец программы

```

Рисунок 58 – Код программы для модуля матрицы из кнопок

2. Проверить правильность написанного программного кода и записать на платформу.

3. Проверить правильность выполнения программы в мониторе последовательного интерфейса.

Индивидуальное задание:

Для каждого студента написать программу с выводом в монитор последовательного интерфейса – сообщение формата [Фамилия Имя, Номер студенческого], после ввода кода на модуле матрицы кнопок.

Самостоятельное изучение:

Используя модуль матрицы кнопок и статических кнопок реализовать задачу, в которой можно увеличить количество считываний кнопок с 18 до 36.

Вопросы для самоконтроля:

1. Что такое кнопка?
2. Зачем нужна матрица?
3. Какие существуют проблемы при использовании кнопок?
4. Как подключается матрица из кнопок?
5. Что такое фиксирующая кнопка?
6. В какой слот можно подключать модуль матрицы из кнопок?

### **Лабораторная работа №8 - Светодиодная индикация, подключенная к ШИМ**

Цель работы:

Ознакомиться с работой светодиодов, подключенных к ШИМ.

Теоретические сведения:

Широтно-Импульсная модуляция (ШИМ) – это управление мощностью, путем изменения скважности импульсов. Получение прямоугольных импульсов сигнала, которые постоянно переключаются от +5В до 0В. Эти переключения происходят настолько быстро, что человеческий глаз не успевает этого заметить. Если подключить к такому контакту световой индикатор, то можно управлять его яркостью в зависимости от программы.

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль ШИМ (рисунок 59) подключается к плате расширения, только во второй слот.

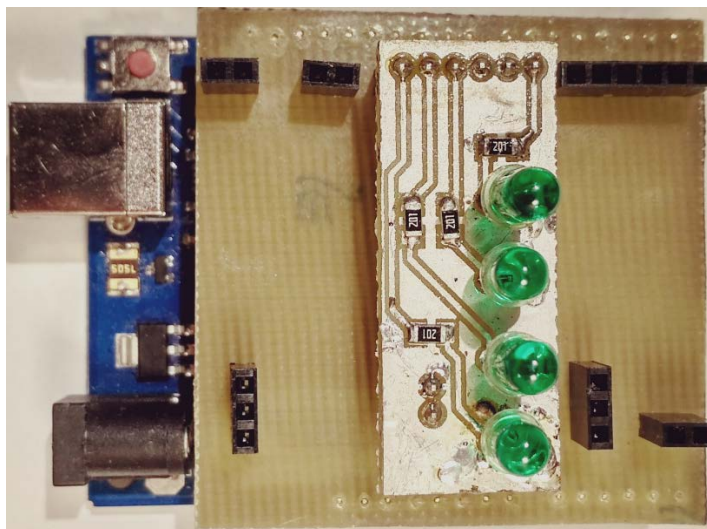


Рисунок 59 – Модуль ШИМ во втором слоте

Светодиоды подключены к ШИМ выводам платформы через токоограничивающий резистор. При подключении модуля во второй слот, светодиоды (обозначение сверху вниз) подключены к платформе как показано в таблице 12. Подключение выполнено только в таком варианте, так как именно эти контакты имеют подключение к ШИМ на платформе Arduino.

Таблица 12 – Схема подключения светодиодов

|           |   |   |    |    |
|-----------|---|---|----|----|
| Светодиод | 1 | 2 | 3  | 4  |
| Контакт   | 6 | 9 | 10 | 11 |

В данной лабораторной работе используется следующая функция:

1. `analogWrite (контакт, значение)` – данная функция формирует заданное аналоговое напряжения в виде ШИМ сигнала. Часто используется для управления яркостью светодиода, или скоростью вращения электродвигателя. Контакт – вход на который подаётся ШИМ сигнал. Значение – период рабочего цикла от 0 (0%) до 255 (100%).

Ход работы

Работа с модулем ШИМ:

1. Написать программу вывода изменения яркости первого светодиода (рисунок 60).
2. Проверить правильность написанного программного кода и записать на платформу.
3. Проверить правильность выполнения программы на модуле, изменения яркости первого светодиода.

```
int ledPin = 6;    // Светодиод подключен к выходу 9
int val;          // переменная для хранения значения

void setup()
{
  pinMode(ledPin, OUTPUT);    // установка порта на выход
}

void loop()
{
  val=0;
  while (val<255){
    analogWrite(ledPin, val); // задает значение ШИМ на контакте
    val++;
    delay(50);
  }
}
```

Рисунок 60 – Код программы для кнопок

Индивидуальное задание:

Для каждого написать программу с изменением яркости светодиодов 1-4 с задержкой для каждого светодиода по последним четырем цифрам номера зачетной книжки.

Самостоятельное изучение:

Используя модуль ШИМ и статические кнопки реализовать задачу, в которой при нажатии на верхний контакт яркость светодиода увеличивается, а на нижний – уменьшается.

Вопросы для самоконтроля:

1. Как расшифровывается ШИМ?
2. Зачем нужен ШИМ?
3. Зачем нужен токоограничивающий резистор?
4. В какой слот можно подключать данный модуль и почему?
5. К каким контактам подключен данный модуль?
6. Сколько контактов использует данный модуль?

### **Лабораторная работа №9 – Измерение температуры и влажности воздуха**

Цель работы:

Ознакомиться с работой измерения температуры и влажности воздуха, с использованием датчика DHT22.

Теоретические сведения:

Измерение температуры и влажности воздуха реализовано на датчике DHT22. Оно выполнено из двух частей – емкостного датчика влажности и термистора, а за счёт того, что внутри него есть аналого-цифровой преобразователь он выдаёт цифровой сигнал, который можно считать любым микроконтроллером. Данный датчик имеет следующие характеристики:

- определение влажности от 0 до 100%, с точностью до 5%;
- определение температуры от  $-40^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ , с точностью плюс-минус  $0.5^{\circ}\text{C}$ ;
- частота опроса 1 раз в 2 секунды;
- питание от 3.3 до 6 вольт.

В данной лабораторной работе используются библиотеки, они нужны для удобства программирования, чтобы не писать большой код программы для реализации считывания данных с датчика.



Для подключения библиотек используем следующий алгоритм:

1. Открыть выпадающее меню: скетч – библиотеки – управлять библиотеками.
2. В открывшемся окне в поле поиска ввести интересующую библиотеку и выбрать её.
3. Нажать на кнопку установить и дождаться окончания установки.

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль температуры и влажности воздуха (рисунок 61) подключается к плате расширения только в первый и второй слот.

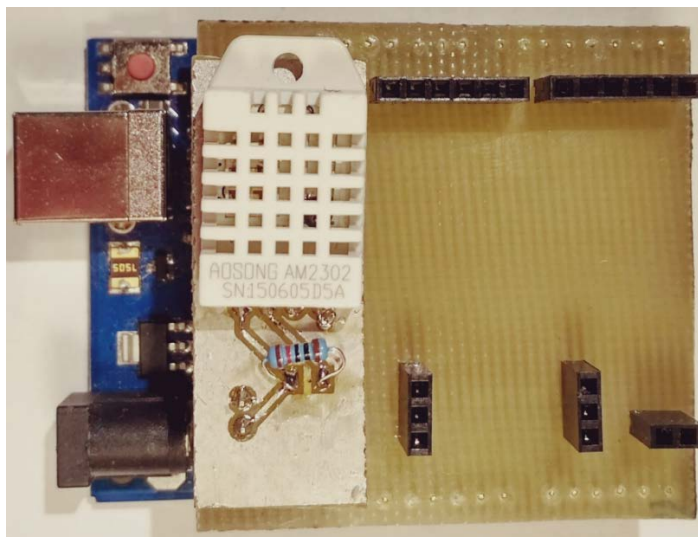


Рисунок 61 – Модуль температуры и влажности воздуха

Модуль выполнен на датчике DHT22, контакт данных подключен через подтягивающий резистор для уменьшения шумов и более точного считывания. При подключении модуля в первый слот, цифровой контакт для считывания платформы имеет номер 12, а во втором слоте 6.

В данной лабораторной работе используются следующие библиотеки:

1. Установить библиотеку – «DHT sensor library», она нужна для опроса датчика температуры и влажности воздуха. Синтаксис для подключения данной библиотеки имеет вид `[#include "DHT.h"]`.
2. Установить библиотеку – «Adafruit Unified sensor», она нужна для корректной работы библиотеки «DHT sensor library».

## Ход работы

Работа с модулем температуры и влажности:

1. Написать программу (рисунок 62) вывода в монитор последовательного интерфейса, температуру и влажность в формате [Температура: (значение), Влажность: (значение)].
2. Проверить правильность написанного программного кода и записать на платформу.
3. Проверить правильность выполнения программы в мониторе последовательного интерфейса.

```
#include "DHT.h" // подключение библиотеки
#define DHTPIN 12 // номер контакта, для считывания данных
DHT dht(DHTPIN, DHT22); //инициализация датчика

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000); // Задержка 2 секунды между измерениями
  float h = dht.readHumidity(); //Считываем влажность
  float t = dht.readTemperature(); // Считываем температуру
  if (isnan(h) || isnan(t)) { // Проверка удачно прошло ли считывание.
    Serial.println("Не удается считать показания");
    return;
  }
  Serial.print("Влажность: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Температура: ");
  Serial.print(t);
  Serial.println(" *C ");
}
```

Рисунок 62 – Код программы для модуля температуры и влажности

Индивидуальное задание:

Для каждого написать программу с выводом на верхнюю кнопку своей группы в формате «Группа: [номер группы]», при нажатии на нижнюю кнопку вывода имени и фамилии в формате «[Фамилия Имя]».

Самостоятельное изучение:

Используя модуль температуры, влажности и сдвигового регистра реализовать задачу, в которой будет выводиться температура в первых двух рядах семисегментного индикатора, а влажность в третьем и четвертом.

Вопросы для самоконтроля:

1. Зачем нужен данный модуль?
2. Для чего нужен подтягивающий резистор?
3. Какое напряжение питания на датчике?
4. Как часто можно опрашивать данный модуль?
5. Что находится внутри датчика DHT22?
6. В какой слот может подключаться данный модуль?

### **Лабораторная работа №10 – Дисплей, подключенный с помощью протокола I2C**

Цель работы:

Ознакомиться с работой подключения дисплея с помощью протокола I2C.

Теоретические сведения:

I2C (или ПС) – последовательная шина связи, позволяет увеличивать функционал, добавляя новые электронные устройства, не задействовав новые контакты микроконтроллера. Данный протокол был разработан компанией Philips в 80-х годах. Данный протокол был запатентован компанией Philips, и разработчики микроконтроллеров придумывали свои имена данному протоколу. Например, на микроконтроллерах ATmega этот протокол называется 2-wire Serial Interface.

Данный протокол используют для связи с внешними устройствами, такими как: дисплеи, компасы, камеры мобильных телефонов, часы реального времени, расширители выхода\вывода, и другие электронные устройства.

К микроконтроллеру Arduino по такому протоколу можно подключить максимум 127 устройств, со скоростью передачи 100 кбит/с.

Шина I2C представляет собой два проводника (не включая питания), и подтягивающие резисторы 10Ком к +5В (рисунок 63).

- SDA (Serial Data) – шина данных;
- SCS (Serial Clock) – тактирование.

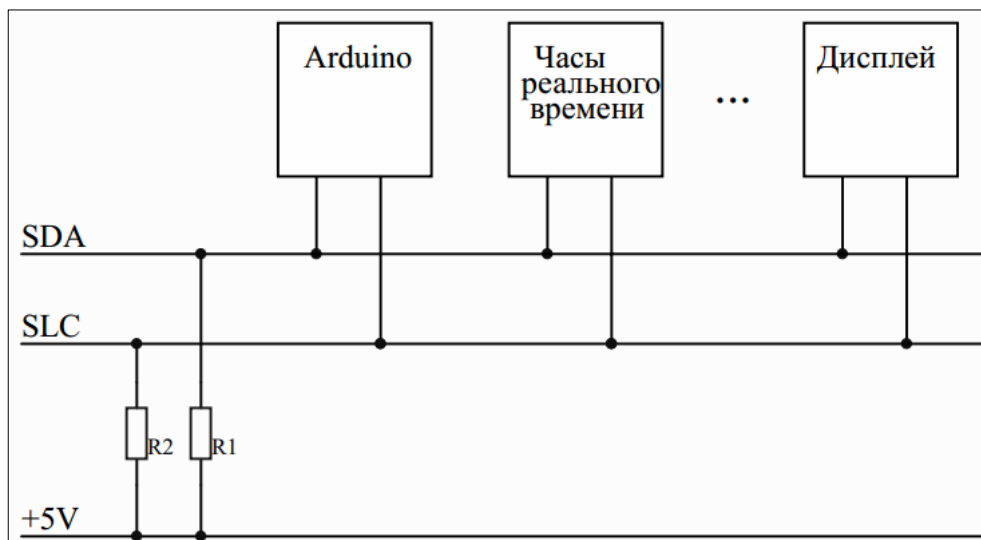


Рисунок 63 – Физическое представление шины I2C

В данном протоколе обычно есть один мастер (Arduino) и различное количество периферийных устройств. У каждого периферийного устройства есть свой собственный адрес (обычно уже прописан в самой периферии), и для того что бы мастеру передать или же считать данные с внешнего устройства нужно проверить не занята ли линия другим устройством, обратиться по адресу к данному устройству и в зависимости от регистра R/W записывать или ждать ответа. Пример запроса показан в таблице 13.

Таблица 13 – Пример запроса I2C

|       |       |   |   |   |   |   |   |   |     |     |        |   |   |   |   |   |   |   |     |      |
|-------|-------|---|---|---|---|---|---|---|-----|-----|--------|---|---|---|---|---|---|---|-----|------|
| Start | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | R/W | ACK | 0      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ACK | Stop |
|       | Адрес |   |   |   |   |   |   |   |     |     | Данные |   |   |   |   |   |   |   |     |      |

Внешним устройством является LCD дисплей 1602. Он позволяет выводить 16 символов в двух строках. Разрешение символов – 5x8 точек. Готовый модуль дисплея представляет из себя законченное устройство с припаянным драйвером для данного LCD дисплея. Общение с микроконтроллером осуществляется через протокол I2C.

В данной лабораторной работе используются следующие аппаратные модули:

1. Модуль переходник I2C (рисунок 64) подключается к плате расширения в первый слот, из-за своей конструкции.

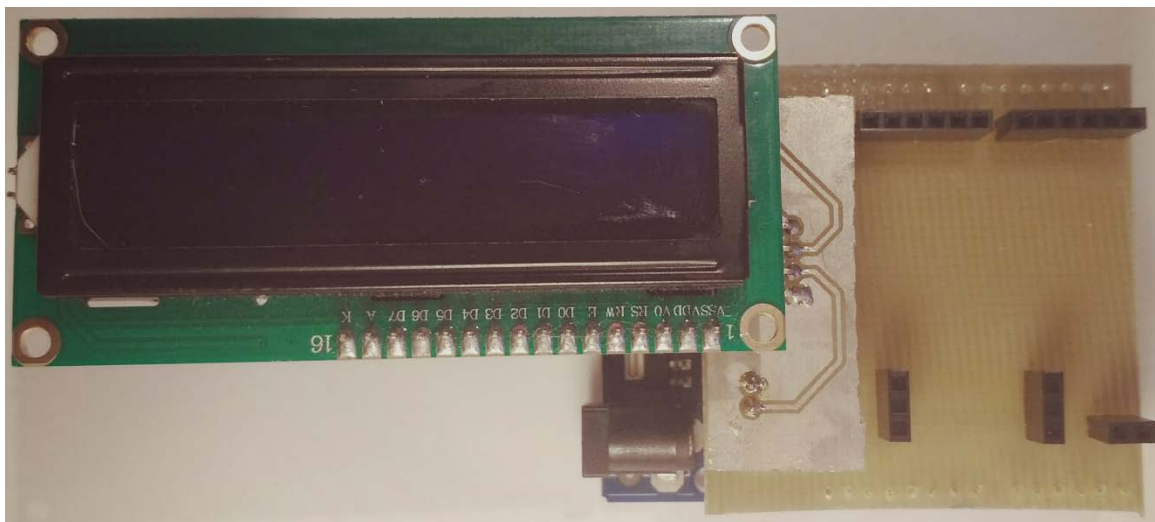


Рисунок 64 – Модуль переходника I2C, с подключенного в него дисплея

На платформе Arduino, контакт SDA подключен к аналоговому порту A4, а SCL к порту A5.

Ход работы

Работа с модулем переходника I2C с подключенного в него дисплея:

1. Написать программу вывода на дисплей: в первой строке – название, а во второй – адрес университета (рисунок 65).
2. Проверить правильность написанного программного кода и записать на платформу.
3. Проверить правильность выполнения на LCD дисплее.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // инициализация дисплея дисплей

void setup()
{
  lcd.init();
  lcd.backlight();// Включаем подсветку дисплея
}
void loop()
{
  lcd.setCursor(5, 0); // Устанавливаем курсор на первую строку и пятый символ.
  lcd.print("RSVPU"); // Выводим на экран сообщения RSVPU
  lcd.setCursor(1, 1); // Устанавливаем курсор на вторую строку и первый символ.
  lcd.print("Mash-stroi. 11"); // Выводим на экран сообщения RSVPU
}

```

Рисунок 65 – Код программы для I2C

Индивидуальное задание:

Для каждого написать программу с выводом на модуль переходника I2C с подключенного в него дисплея: в первой строчке – Фамилия и Имя, а во второй строчке – номер группы и номер зачетной книжки.

Самостоятельное изучение:

Используя модуль переходника I2C с подключенного в него дисплея и матрицы из кнопок реализовать задачу, в которой можно будет выводить часы на LCD дисплей, настройка времени осуществляется через модуль матрица из кнопок.

Вопросы для самоконтроля:

1. Что такое LCD?
2. Зачем нужен подтягивающий резистор?
3. Зачем нужен протокол I2C?
4. Как выглядит шина протокола I2C?
5. К каким контактам подключен протокол I2C в Arduino?
6. Сколько максимально можно подключить устройств к протоколу I2C в Arduino?

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута основная цель – разработано техническое обеспечение для дисциплины «Микропроцессорная техника».

Разработка устройств осуществлялась с использованием программного пакета CircuitMaker. В рамках разработки были созданы следующие документы:

- принципиальная электрическая схема;
- перечень элементов;
- чертёж печатной платы;
- сборочный чертёж.

Были разработаны методические указания для проведения лабораторных работ в рамках дисциплины «Микропроцессорная техника». Методическое обеспечение позволяет обучающимся широко рассмотреть область применения микроконтроллеров, узнать о различных способах подключения индикаторов и датчиков с использованием различных методов.

Разработанные устройства и методическое обеспечение в дальнейшем могут быть усовершенствованы на основании имеющейся документации.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Новиков Ю.В. Основы микропроцессорной техники. [Текст]: Учебное пособие / Ю.В. Новиков, П.К. Скоробогатов. – 4-е изд,испр. – М.: Бином, 2009. – 357 с.
2. Пухальский Г.И. Проектирование микропроцессорных устройств. [Текст]: Учебное пособие / Г.И. Пухальский. –М.: Политехника, 2001. – 544 с.
3. Сабунин А. Е Altium Designer. Новые решения в проектировании электронных устройств [Текст]. / А. Е. Сабунин – М : СОЛОН-Пресс, 2009. – 432 с.
4. Характеристики Arduino Uno [Электронный ресурс] / Режим доступа: – <http://wiki.amperka.ru/программирование:структура-программы-для-arduino> (дата обращения: 10.05.2017).
5. Программирование C++ для Arduino [Электронный ресурс] / Режим доступа: – <http://wiki.amperka.ru/продукты:arduino-uno> (дата обращения: 24.05.2017).
6. Работа со сдвиговым регистром [Электронный ресурс] / Режим доступа: [http://codius.ru/articles/Arduino\\_Сдвиговой\\_регистр\\_74НС595\\_или\\_размножаем\\_экономим\\_выходы\\_платы](http://codius.ru/articles/Arduino_Сдвиговой_регистр_74НС595_или_размножаем_экономим_выходы_платы) (дата обращения: 27.04.2017).
7. Резисторы [Электронный ресурс] / Режим доступа: <https://lib.chipdip.ru/244/DOC000244219.pdf> (дата обращения: 23.04.2017).
8. Резисторы, ток и напряжение [Электронный ресурс] / Режим доступа: <http://сhem.net/beginner/beginner87.php> (дата обращения: 22.04.2017).
9. Светодиодная индикация. [Электронный ресурс] / Режим доступа: <https://lib.chipdip.ru/151/DOC000151610.pdf> (дата обращения: 23.04.2017).



10. Сдвиговый регистр на базе микросхемы 74НС595D [Электронный ресурс] / Режим доступа: <https://lib.chipdip.ru/223/DOC000223773.pdf> (дата обращения: 29.04.2017).

11. Справочник языка программирования Arduino [Электронный ресурс] / Режим доступа: <http://arduino.ua/ru/prog/> (дата обращения: 23.05.2017).

12. Технические характеристики DHT11 и DHT22 [Электронный ресурс] / Режим доступа: <http://arduino-diy.com/arduino-datchiki-temperature-i-vlazhnosti-DHT11-i-DHT22> (дата обращения: 18.04.2017).

13. Управления семисегментными индикатором [Электронный ресурс] / Режим доступа: <http://сhem.net/beginner/beginner131.php> (дата обращения: 23.04.2017).

14. Adafruit DHT Humidity & Temperature Unified Sensor Library [Электронный ресурс] / Режим доступа: <https://github.com/adafruit/DHT-sensor-library> (дата обращения: 30.05.2017).

15. Altium Wiki [Электронный ресурс] / Режим доступа: <http://wiki.altium.com> (дата обращения: 16.04.2017).

16. AMS1117 [Электронный ресурс] / Режим доступа: <http://www.advanced-monolithic.com/pdf/ds1117.pdf> (дата обращения: 17.04.2017).

17. GCC, the GNU Compiler Collection [Электронный ресурс] / Режим доступа: <https://gcc.gnu.org> (дата обращения: 15.04.2017).

18. Libraries Arduino [Электронный ресурс] / Режим доступа: <https://github.com/arduino-libraries> (дата обращения: 29.04.2017).

19. РСВ [Электронный ресурс] / Режим доступа: [https://documentation.circuitmaker.com/display/СМАК/РСВ\\_Pnl-РСВ\(\(РСВ\)\)\\_СМ](https://documentation.circuitmaker.com/display/СМАК/РСВ_Pnl-РСВ((РСВ))_СМ) (дата обращения: 11.04.2017).

20. РСВ INSPECTOR [Электронный ресурс] / Режим доступа: [https://documentation.circuitmaker.com/display/СМАК/РСВ\\_Pnl-РСВInspector\(\(РСВ+Inspector\)\)\\_СМ](https://documentation.circuitmaker.com/display/СМАК/РСВ_Pnl-РСВInspector((РСВ+Inspector))_СМ) (дата обращения: 11.04.2017).

21. STK500 Communication Protocol [Электронный ресурс] / Режим доступа: <http://www.atmel.com/images/doc2591.pdf> (дата обращения: 18.04.2017).

22. Wire Library [Электронный ресурс] / Режим доступа: <https://www.arduino.cc/en/Reference/Wire> (дата обращения: 31.05.2017).

# ПРИЛОЖЕНИЕ А ЗАДАНИЕ НА ПОДГОТОВКУ ВКР

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное автономное образовательное учреждение**  
**высшего образования**

**«Российский государственный профессионально-педагогический университет»**

Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий  
направление 44.03.04 Профессиональное обучение (по отраслям)  
профиль «Энергетика»  
профилизация «Компьютерные технологии автоматизации и управления»

**УТВЕРЖДАЮ**

Заведующий кафедрой  
\_\_\_\_\_ Н. С. Толстова

« \_\_\_\_ » \_\_\_\_\_ 2017 г.

## **ЗАДАНИЕ**

### **на подготовку выпускной квалификационной работы**

Обучающегося (ейся) группы КТэ-402

Фамилия Набиева Имя Ильи Отчество Айдаровича

1. Тема Аппаратно-программный комплекс для лабораторных работ по дисциплине «Микропроцессорная техника»

утверждена распоряжением по институту от \_\_\_\_\_ г. №.

2. Руководитель М. Ю. Черноскутов

3. Место преддипломной практики ФГАОУ ВО РГППУ

4. Исходные данные к ВКР

Новиков Ю.В. Основы микропроцессорной техники

Пухальский Г.И. Проектирование микропроцессорных устройств

Сабунин А. Е Altium Designer. Новые решения в проектировании электронных устройств

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)

Изучение литературы по теме микропроцессорная техника

Разработка аппаратных средств обучения

Разработка методического указания к разрабатываемым устройствам

6. Перечень демонстрационных материалов

Презентация

Защитное слово

Пояснительная записка

Разрабатываемые устройства

#### 7. Календарный план выполнения выпускной квалификационной работы

| № п/п | Наименование этапа дипломной работы   | Срок выполнения этапа | Процент выполнения ВКР | Отметка руководителя о выполнении |
|-------|---|-----------------------|------------------------|-----------------------------------|
| 1     | Сбор информации по выпускной работе и сдача зачета по преддипломной практике  | 19.05.2017            | 15%                    |                                   |
| 2     | Выполнение работ по разрабатываемым вопросам их изложение в выпускной работе: |                       | 65%                    |                                   |
|       | Изучение литературы по теме микропроцессорная техника                         | 20.04.2017            | 15%                    |                                   |
|       | Разработка технических средств для дисциплины микропроцессорная техника       | 12.05.2017            | 30%                    |                                   |
|       | Разработка методического указания к разрабатываемым устройствам               | 30.05.2017            | 20%                    |                                   |
| 3     | Оформление текстовой части ВКР  | 10.06.2017            | 5%                     |                                   |
| 4     | Выполнение демонстрационных материалов к ВКР                                  | 12.06.2017            | 5%                     |                                   |
| 5     | Нормоконтроль   | 13.06.2017            | 5%                     |                                   |
| 6     | Подготовка доклада к защите в ГЭК   | 13.06.2017            | 5%                     |                                   |

#### 8. Консультанты по разделам выпускной квалификационной работы

| Наименование раздела | Консультант | Задание выдал |      | Задание принял |      |
|----------------------|-------------|---------------|------|----------------|------|
|                      |             | подпись       | дата | подпись        | дата |
|                      |             |               |      |                |      |
|                      |             |               |      |                |      |
|                      |             |               |      |                |      |

Руководитель \_\_\_\_\_  
подпись                      дата

Задание получил \_\_\_\_\_  
подпись студента                      дата

9. Выпускная квалификационная работа и все материалы проанализированы. Считаю возможным допустить Набиева Илью Айдаровича к защите выпускной квалификационной работы в государственной экзаменационной комиссии.

Руководитель \_\_\_\_\_  
подпись                      дата

10. Допустить Набиева Илью Айдаровича к защите выпускной квалификационной работы в государственной экзаменационной комиссии (протокол заседания кафедры от \_\_\_\_\_)

Заведующий кафедрой \_\_\_\_\_  
подпись                      дата

## ПРИЛОЖЕНИЕ Б. ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ

Таблица Б.1 – Перечень элементов платы расширения

| Поз. обозначение | Наименование | Кол | Примечание |
|------------------|--------------|-----|------------|
| 1                | 2            | 3   | 4          |
|                  | Разъемы      |     |            |
| XP1              | PLS-08       | 2   |            |
|                  | PLS-06       | 1   |            |
|                  | PLS-10       | 1   |            |
| XS1, XS2         | PBS-06       | 2   |            |
| XS3, XS4,<br>XS8 | PBS-02       | 3   |            |
| XS5, XS6,<br>XS7 | PBS-03       | 3   |            |

Таблица Б.2 – Перечень элементов модуля матрицы из кнопок

| Поз. обозначение  | Наименование        | Кол | Примечание |
|---|---------------------|-----|------------|
| 1   | 2                   | 3   | 4          |
|   | Разъемы             |     |            |
| XP1   | PLS-06              | 1   |            |
|   | Кнопки              |     |            |
| SB1, SB2,<br>SB3, SB4,<br>SB5, SB6,<br>SB7, SB8,<br>SB9 | KLS7-TS6601-9.5-180 | 9   |            |

Таблица Б.3 – Перечень элементов модуля матрицы из светодиодов

| Поз. обозначение  | Наименование            | Кол | Примечание |
|---|-------------------------|-----|------------|
| 1   | 2                       | 3   | 4          |
|   | Разъемы                 |     |            |
| XP1   | PLS-06                  | 1   |            |
|   | Светодиодные индикаторы |     |            |
| HL1, HL2,<br>HL3, HL4,<br>HL5, HL6,<br>HL7, HL8,<br>HL9 | L-7113SGD               | 9   |            |

|            |                                      |   |  |
|------------|--------------------------------------|---|--|
|            | Резисторы                            |   |  |
| R1, R2, R3 | RC1206FK-07200RL чип 1206 200 Ом, 1% | 3 |  |

Таблица Б.4 – Перечень элементов модуля семисегментного индикатора

| Поз. обозначение                     | Наименование                         | Кол | Примечание |
|--------------------------------------|--------------------------------------|-----|------------|
| 1                                    | 2                                    | 3   | 4          |
|                                      | Разъемы                              |     |            |
| XP1, XP2                             | PLS-06                               | 2   |            |
| XP3                                  | PLS-02                               | 1   |            |
|                                      | Семисегментный индикатор             |     |            |
| HL1                                  | SA56-11SRWA                          | 1   |            |
|                                      | Резисторы                            |     |            |
| R1, R2, R3,<br>R4, R5, R6,<br>R7, R8 | RC1206FK-07200RL чип 1206 200 Ом, 1% | 8   |            |

Таблица Б.5 – Перечень элементов модуля шкального индикатора

| Поз. обозначение                                 | Наименование                         | Кол | Примечание |
|--|--------------------------------------|-----|------------|
| 1  | 2                                    | 3   | 4          |
|  | Разъемы                              |     |            |
| XP1, XP2   | PLS-06                               | 2   |            |
| XP3  | PLS-02                               | 1   |            |
|  | Семисегментный индикатор             |     |            |
| HL1  | DC-10GWA                             | 1   |            |
|  | Резисторы                            |     |            |
| R1, R2, R3,<br>R4, R5, R6,<br>R7, R8, R9,<br>R10 | RC1206FK-07200RL чип 1206 200 Ом, 1% | 10  |            |

Таблица Б.6 – Перечень элементов модуля ШИМ

| Поз. обозначение       | Наименование                         | Кол | Примечание |
|------------------------|--------------------------------------|-----|------------|
| 1                      | 2                                    | 3   | 4          |
|                        | Разъемы                              |     |            |
| XP1                    | PLS-06                               | 1   |            |
| XP2                    | PLS-02                               | 1   |            |
|                        | Светодиодные индикаторы              |     |            |
| HL1, HL2,<br>HL3, HL4, | L-7113SGD                            | 4   |            |
|                        | Резисторы                            |     |            |
| R1, R2, R3,            | RC1206FK-07200RL чип 1206 200 Ом, 1% | 4   |            |

| Поз. обозначение | Наименование | Кол | Примечание |
|------------------|--------------|-----|------------|
| R4               |              |     |            |

Таблица Б.7 – Перечень элементов модуля ШИМ

| Поз. обозначение       | Наименование                         | Кол | Примечание |
|------------------------|--------------------------------------|-----|------------|
| 1                      | 2                                    | 3   | 4          |
|                        | Разъемы                              |     |            |
| XP1                    | PLS-06                               | 1   |            |
| XP2                    | PLS-02                               | 1   |            |
|                        | Светодиодные индикаторы              |     |            |
| HL1, HL2,<br>HL3, HL4, | L-7113SGD                            | 4   |            |
|                        | Резисторы                            |     |            |
| R1, R2, R3,<br>R4      | RC1206FK-07200RL чип 1206 200 Ом, 1% | 4   |            |

Таблица Б.8 – Перечень элементов модуля статических кнопок

| Поз. обозначение | Наименование        | Кол | Примечание |
|------------------|---------------------|-----|------------|
| 1                | 2                   | 3   | 4          |
|                  | Разъемы             |     |            |
| XP1, XP2         | PLS-02              | 2   |            |
|                  | Кнопки              |     |            |
| SB1, SB2         | KLS7-TS6601-9.5-180 | 2   |            |

Таблица Б.9 – Перечень элементов модуля потенциометра

| Поз. обозначение | Наименование | Кол | Примечание |
|------------------|--------------|-----|------------|
| 1                | 2            | 3   | 4          |
|                  | Разъемы      |     |            |
| XP1              | PLS-03       | 1   |            |
|                  | Потенциометр |     |            |
| R1               | 16K1-10C10K  | 1   |            |

Таблица Б.10 – Перечень элементов модуля датчика температуры и влажности воздуха

| Поз. обозначение | Наименование                           | Кол | Примечание |
|------------------|--|-----|------------|
| 1                | 2                                      | 3   | 4          |
|                  | Разъемы                                |     |            |
| XP1, XP2         | PLS-02                                 | 2   |            |
|                  | Датчик температуры и влажности воздуха |     |            |
| DD1              | DHT22                                  | 1   |            |

| Поз. обозначение | Наименование                        | Кол | Примечание |
|------------------|-------------------------------------|-----|------------|
|                  | Резистор                            |     |            |
| R1               | RC1206FK-0710KL чип 1206 10 кОм, 1% | 1   |            |

Таблица Б.11 – Перечень элементов модуля сдвигового регистра

| Поз. обозначение               | Наименование                              | Кол | Примечание |
|--------------------------------|---|-----|------------|
| 1                              | 2   | 3   | 4          |
|                                | Разъемы                                   |     |            |
| XP1                            | PLS-06                                    | 1   |            |
| XP2                            | PLS-02                                    | 1   |            |
|                                | Семисегментный четырёхразрядный индикатор |     |            |
| HL1                            | CA56-11EWA                                | 1   |            |
|                                | Микросхема                                |     |            |
| DD1, DD2                       | 74HC595D                                  | 2   |            |
|                                | Резисторы                                 |     |            |
| R1, R2, R3, R4, R5, R6, R7, R8 | RC1206FK-07200RL чип 1206 200 Ом, 1%      | 8   |            |

Таблица Б.12 – Перечень элементов модуля I2C

| Поз. обозначение | Наименование | Кол | Примечание |
|------------------|--------------|-----|------------|
| 1                | 2            | 3   | 4          |
|                  | Разъемы      |     |            |
| XP1, XP2         | PLS-02       | 2   |            |
| XS1              | PBS-04R      | 1   |            |