

Список литературы

- 1 **Иванов, В. Н.** Электроника и микропроцессорная техника : учебник / В. Н. Иванов, И. О. Мартынова. – Москва : Академия, 2016. – 288 с.
- 2 **Титов, В. С.** Проектирование аналоговых и цифровых устройств : учебное пособие / В. С. Титов, В. И. Иванов, М. В. Бобырь. – Москва : ИНФРА-М, 2016. – 143 с.
- 3 **Миленина, С. А.** Электротехника, электроника и схемотехника : учебник и практикум для академ. бакалавриата / С. А. Миленина ; под ред. Н. К. Миленина. – Москва : Юрайт, 2015. – 399 с.
- 4 **Бладыко, Ю. В.** Электроника. Практикум: учебное пособие / Ю. В. Бладыко. – Минск: ИВЦ Минфина, 2016. – 190 с.: ил.
- 5 **Horowitz, P.** The art of electronics. Third Edition / P. Horowitz, W. Hill. – Cambridge: University Press, 2015. – 1192 с.

ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Физические методы контроля»

СХЕМОТЕХНИКА АНАЛОГОВЫХ И ЦИФРОВЫХ УСТРОЙСТВ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
12.03.04 «Биотехнические системы и технологии»
дневной формы обучения*



Могилев 2018



УДК 621.3.049.77
ББК 32.844+32.85
С 92

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Физические методы контроля» «12» января 2018 г.,
протокол № 6

Составители: канд. техн. наук, доц. А. А. Афанасьев;
ст. преподаватель Е. Н. Прокопенко

Рецензент канд. техн. наук, доц. С. В. Болотов

В методических рекомендациях кратко изложены изучаемые теоретические сведения, приведен порядок выполнения экспериментальных исследований, указана структура отчета о выполненной работе и дан список контрольных вопросов для самопроверки по каждой теме. Составлены в соответствии с рабочей программой по дисциплине «Схемотехника аналоговых и цифровых устройств» для студентов специальности 12.03.04 «Биотехнические системы и технологии» дневной формы обучения.

Учебно-методическое издание

СХЕМОТЕХНИКА АНАЛОГОВЫХ И ЦИФРОВЫХ УСТРОЙСТВ

Ответственный за выпуск	С. С. Сергеев
Технический редактор	А. А. Подошевка
Компьютерная верстка	М. М. Дударева

Подписано в печать Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. Уч.-изд. л. Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018

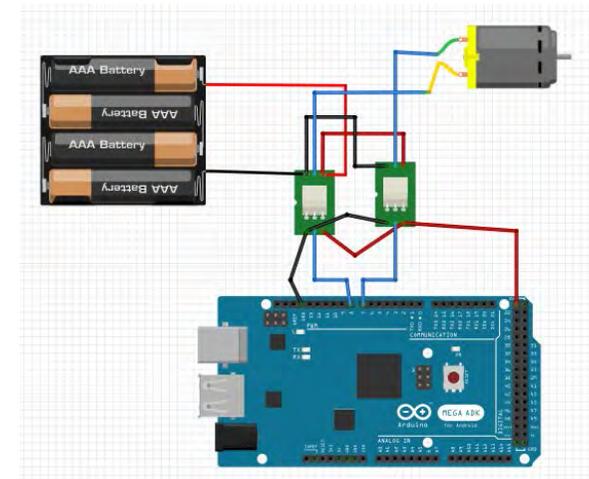


Рисунок 8.6 – Схема подключения электродвигателя к контроллеру Arduino, обеспечивающая реверс вращения

8.2 Порядок выполнения работы

8.2.1 Подключите реле SRD-05VDC к контроллеру Arduino, контроллер Arduino подключите к ПК.

8.2.2 Загрузите в контроллер Arduino программу Rele1, исследуйте её выполнение.

8.2.3 Составьте и загрузите в контроллер Arduino программу Motor1, обеспечивающую работу электродвигателя в обычном режиме, и исследуйте её выполнение.

8.2.4 Загрузите в контроллер Arduino программу Motor2, исследуйте её выполнение.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схемы подключения к Arduino реле, электродвигателя в обычном и реверсивном режимах, программы и результаты экспериментальных исследований, выводы.

Контрольные вопросы

- 1 Опишите устройство и принцип работы реле.
- 2 Опишите принцип управления работой электродвигателя в обычном режиме.
- 3 Опишите принцип управления работой электродвигателя в реверсивном режиме.



Код программы, обеспечивающий реверсивный режим работы электродвигателя, представлен далее.

Код программы Motor2

```
int Relay = 8;
int Relay1 = 9;
void setup()
{
  pinMode(Relay, OUTPUT);
  pinMode(Relay1, OUTPUT);
}
void loop()
{
  digitalWrite(Relay, LOW); // реле включено
  delay(1000);
  digitalWrite(Relay, HIGH); // реле выключено
  delay(1000);
  digitalWrite(Relay1, LOW); // реле включено
  delay(1000);
  digitalWrite(Relay1, HIGH); // реле выключено
  delay(1000);
}
```

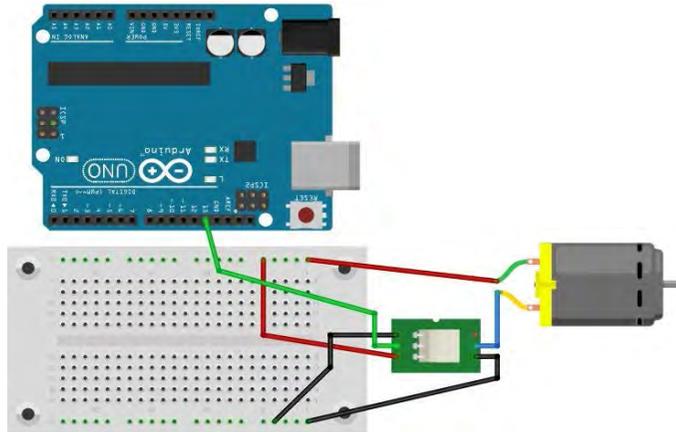


Рисунок 8.5 – Схема подключения электродвигателя к контроллеру Arduino через реле

Содержание

1 Лабораторная работа № 1. Моделирование и исследование работы цифрового прибора на микроконтроллере с аналоговым датчиком.....	4
2 Лабораторная работа № 2. Моделирование и исследование работы цифрового прибора на микроконтроллере с дискретным датчиком	7
3 Лабораторная работа № 3. Моделирование и исследование работы порогового устройства на микроконтроллере.....	10
4 Лабораторная работа № 4. Моделирование и исследование работы генератора прямоугольных импульсов на микроконтроллере	13
5 Лабораторная работа № 5. Моделирование и исследование работы генератора временных интервалов на микроконтроллере	16
6 Лабораторная работа № 6. Моделирование и исследование работы индикаторных устройств цифрового прибора на микроконтроллере	20
7 Лабораторная работа № 7. Моделирование и исследование работы средств отображения информации цифрового прибора на микроконтроллере.....	23
8 Лабораторная работа № 8. Моделирование и исследование работы исполнительного устройства цифрового прибора на микроконтроллере.....	27
Список литературы	32

1 Лабораторная работа № 1. Моделирование и исследование работы цифрового прибора на микроконтроллере с аналоговым датчиком

Цель работы: исследовать работу цифрового прибора на микроконтроллере с аналоговым датчиком.

1.1 Основные теоретические положения

Макет цифрового прибора собирается на макетной плате на основе контроллера Arduino Leonardo, созданного на базе микросхемы Atmel ATmega32U4. В контроллере предусмотрен обмен данными с ЭВМ через USB-порт. Внешний вид контроллера Arduino Leonardo приведен на рисунке 1.1.



Рисунок 1.1 – Внешний вид контроллера Arduino Leonardo

Контроллер Arduino Leonardo имеет 20 цифровых входов-выходов (семь из которых могут использоваться как выходы ШИМ и 12 – как аналоговые входы), кварцевый генератор на 16 МГц, разъем микро USB, силовой разъем, разъем ICSP, кнопку перезагрузки.

Для работы необходимо подключить контроллер к компьютеру посредством кабеля USB либо подать питание при помощи адаптера AC/DC или батареи.

В качестве аналогового датчика используется потенциометр.

Потенциометр – механическое устройство, у которого изменяется сопротивление при повороте его вала. Если потенциометр подключить к источнику питания (ИП), то он будет выполнять роль делителя напряжения и с его помощью можно плавно изменять напряжение $U_{\text{вых}}$ от 0 до + 5 В, которое затем подается на аналоговый вход Arduino (рисунок 1.2).

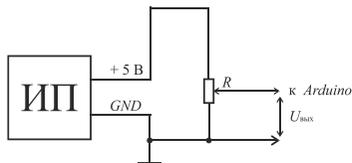


Рисунок 1.2 – Схема подключения потенциометра к источнику питания

Пример программного кода *Rele1*

```
// Реле-модуль подключен к цифровому выводу 4
int Relay = 4;
void setup()
{
  pinMode(Relay, OUTPUT);
}
void loop()
{
  digitalWrite(Relay, LOW); // реле включено
  delay(2000);
  digitalWrite(Relay, HIGH); // реле выключено
  delay(2000);
}
```

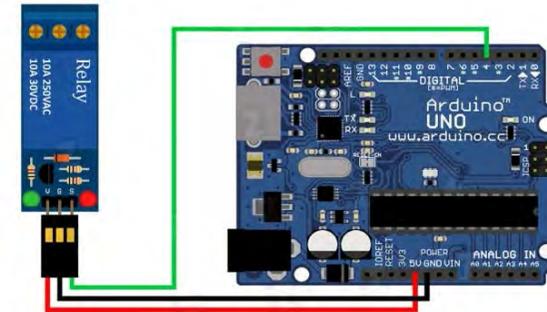


Рисунок 8.4 – Схема подключения реле к контроллеру Arduino

8.1.2 Управление электродвигателем. Напрямую к цифровым контактам Arduino можно подключать только устройства, потребляющие небольшую мощность, например, светодиод или спикер. Для доказательства следует рассмотреть возможность подключения электродвигателя с сопротивлением 10 Ом. При напряжении +5 В (выдаваемым пином Arduino) в соответствии с законом Ома $U = IR$ через мотор потечет ток $I = U/R = 5/10 = 0,5 \text{ A} = 500 \text{ mA}$, что намного больше, чем допустимый ток через цифровой контакт Arduino, равный 40 мА. Поэтому при подключении электродвигателя непосредственно к контакту Arduino контроллер может выйти из строя.

Простым вариантом управления электродвигателем является использование реле. Схема подключения электродвигателя к контроллеру Arduino через реле приведена на рисунке 8.5.

Работу электродвигателя также можно реализовать в реверсивном режиме (обеспечить вращение в обе стороны). Для этого нужно использовать два реле (рисунок 8.6).

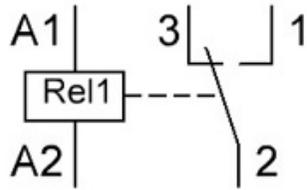


Рисунок 8.2 – Коммутационная схема реле SRD-05VDC

Подключение модуля к плате Arduino:

- VCC на + 5 В на Arduino;
- GND на любой из GND-пинов Arduino;
- IN на любой из цифровых входов/выходов Arduino.

Электрическая функциональная схема реле SRD-05VDC приведена на рисунке 8.3. В управлении реле используются следующие детали: резистор R1, *p-n-p*-транзистор VT1, диод VD1 и реле Rel1. Два светодиода установлены для индикации: LED1 (красный) предназначен для индикации подачи питания на модуль реле, LED2 (зеленый) свидетельствует о замыкании контактов реле.

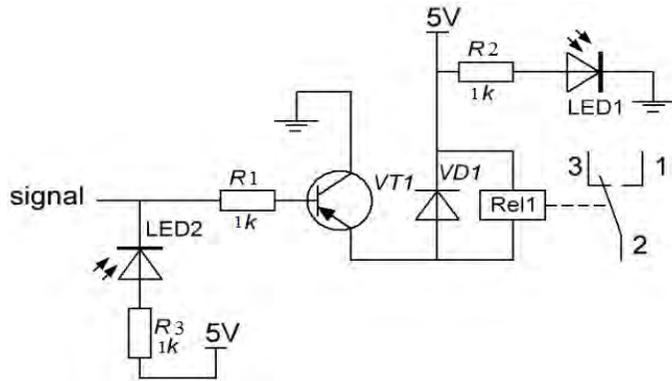


Рисунок 8.3 – Электрическая схема реле SRD-05VDC

Рассмотрим, как работает схема. При включении контроллера Arduino его цифровые выводы находятся в высокоомном состоянии, транзистор VT1 закрыт. Так как транзистор *p-n-p*-типа, то для его открытия нужно подать на базу 0 В. Для этого используется функция **digitalWrite(pin, LOW)**. Таким образом VT1 открыт, через управляющую цепь течет ток и реле срабатывает. Для отключения реле следует закрыть транзистор VT1, подав на базу +5 В, вызвав функцию **digitalWrite(pin, HIGH)**.

Схема подключения реле к контроллеру Arduino представлена на рисунке 8.4. Далее приведен код программы включения и выключения реле с интервалом в 2 с.

1.2 Порядок выполнения работы

1.2.1 Присоедините три провода, припаянные к потенциометру, к плате Arduino, как показано на рисунке 1.3 в соответствии со схемой, приведенной на рисунке 1.4. Один из крайних выводов потенциометра соедините с землей (GND) Arduino, второй – с выводом 5V Arduino, третий – с аналоговым входом A0 Arduino.

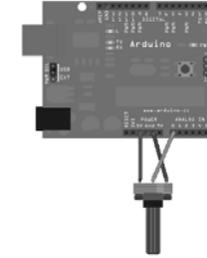


Рисунок 1.3 – Внешний вид подключения потенциометра к плате Arduino

При вращении вала потенциометра можно изменять сопротивление по обе стороны от центрального вывода. Это приводит к изменению напряжения на центральном выводе. Когда напряжение между центральным выводом и выводом, подключенным к 5 В, близится к нулю, напряжение на центральном выводе приближается к 5 В. Если повернуть ручку в другую сторону, то сопротивления поменяются местами и напряжение на центральном контакте приблизится к 0 В. Это напряжение является аналоговым сигналом, который следует подавать на аналоговый вход Arduino.

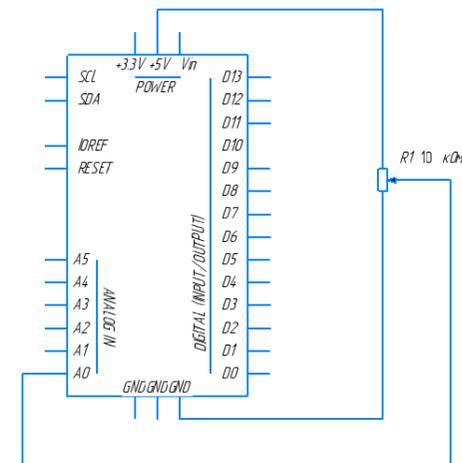


Рисунок 1.4 – Схема подключения потенциометра к контроллеру Arduino

Arduino имеет встроенный 10-разрядный аналого-цифровой преобразователь (АЦП), который считывает значения напряжения и преобразовывает их в числа от 0 до 1023.

Если ручка потенциометра повернута до упора в одну сторону, то на вход АЦП подается 0 В и результат преобразования равен 0. Если ручка потенциометра повернута до упора в другую сторону, то на вход АЦП подается 5 В и результат преобразования равен 1023. В промежуточных положениях ручки потенциометра АЦП возвращает число между 0 и 1023, которое пропорционально напряжению на его среднем выводе.

1.2.2 Подключите плату Arduino к ПК через USB-порт.

1.2.3 Запишите программу ReadAnalogVoltage в память контроллера Arduino, переслав её из ПК. В данной программе в функции установки необходимо начать последовательную передачу между Arduino и ПК со скоростью 9600 бит в секунду командой

```
Serial.begin(9600);
```

Далее в основном цикле программы следует создать переменную для хранения значения напряжения (которая может изменяться от 0 до 1023, лучше всего подойдет тип int), которое приходит с потенциометра.

```
int sensorValue = analogRead(A0);
```

Эта информация отразится на мониторе в виде десятичных (DEC) значений. Можно сделать это, используя команду Serial.println(), с помощью следующей строки кода:

```
Serial.println(sensorValue, DEC)
```

Открыв Serial Monitor в среде ArduinoM, можно увидеть поток цифр от нуля до 1023, соответствующих положению ручки потенциометра. Если повернуть ручку, эти показания изменятся почти мгновенно.

Полный текст программы

```
/*
AnalogReadSerial
Считывает аналоговые значения с вывода 0, выводит значения на
монитор.
*/
// установки:
void setup()
{
// инициализируем последовательную передачу данных со скоростью
9600 бит в секунду:
Serial.begin(9600);
}
// основной цикл:
void loop()
{
```

2 Как обеспечивается динамический режим работы индикатора TOT-3361AH-IN?

3 Как осуществляется управление работой индикаторов в микроконтроллерных системах?

4 Опишите устройство и принцип работы 4-разрядного LED-дисплея с I2C драйвером TM1637.

5 Опишите устройство и принцип работы LCD-дисплея 1602.

8 Лабораторная работа № 8. Моделирование и исследование работы исполнительного устройства цифрового прибора на микроконтроллере

Цель работы: исследовать работу подключаемых к цифровому прибору на микроконтроллере исполнительных устройств.

8.1 Основные теоретические положения

8.1.1 Программное управление мощными исполнительными устройствами цифрового прибора может быть реализовано с помощью электромагнитного реле. На рисунке 8.1 показан внешний вид реле SRD-05VDC.



Рисунок 8.1 – Внешний вид реле SRD-05VDC

Данное реле управляется напряжением 5 В и способно коммутировать до 10 А 30 В DC и 10 А 250 В AC.

Реле имеет две отдельные цепи: цепь управления, представленную контактами A1, A2, и управляемую цепь, представленную контактами 1, 2, 3. Цепи никак не связаны между собой (рисунок 8.2).

Релейный модуль имеет три вывода:

- 1) VCC – «+» питания;
- 2) GND – «-» питания;
- 3) IN – вывод входного сигнала.

Для работы с данным модулем необходимо установить библиотеку LiquidCrystal_I2C1602V1.

На рисунке 7.5 показана микросхема жидкокристаллического индикатора WM-C1602N совместно с клавиатурой. В состав платы входит LCD-дисплей и шесть кнопок. Пины 4...10 задействованы для управления LCD. На аналоговый пин 0 считываются сигналы с кнопок. Дисплей в данной сборке работает в четырехбитном режиме.

Подключение производится путем стыковки дисплея с платой из семейства Arduino (Diecimila, Duemilanove, UNO, Mega, Mega2560, Mega ADK).

7.2 Порядок выполнения работы

7.2.1 Подключите к плате Arduino LED-дисплей с I2C драйвером TM1637. Подключите плату Arduino к ПК через USB-порт.

7.2.2 Запишите программу D-led-display в память контроллера Arduino, переслав её из ПК, и исследуйте её выполнение.



Рисунок 7.5 – Жидкокристаллический индикатор WM-C1602N с клавиатурой

7.2.3 Подключите к плате Arduino дисплей WM-C1602N с клавиатурой. Подключите плату Arduino к ПК через USB-порт.

7.2.4 Запишите программу D-LCD-display в память контроллера Arduino, переслав её из ПК, и исследуйте её выполнение.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему подключения к контроллеру Arduino Nano 4-разрядного LED-дисплея с I2C драйвером TM1637, схему подключения к контроллеру Arduino LCD-дисплея 1602, результаты экспериментальных исследований, выводы.

Контрольные вопросы

1 Опишите устройство и принцип работы цифро-буквенного светодиодного индикатора TOT-3361AH-IN.

```
// читаем значение на аналоговом входе 0:
int sensorValue = analogRead(A0);
// выводим на монитор считанное значение:
Serial.println(sensorValue);
delay(1); // задержка в промежутке между считываниями для
стабильности:
}
```

1.2.4 Вращая ручку потенциометра, наблюдайте на экране дисплея ПК поток цифр от 0 до 1023, соответствующих положению ручки потенциометра.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему подключения потенциометра к Arduino, выводы по результатам экспериментальных исследований.

Контрольные вопросы

- 1 Какова разрядность встроенного в контроллер Arduino АЦП?
- 2 В каком диапазоне изменяется напряжение на выходе потенциометра?
- 3 В каком диапазоне формируется результат преобразования на выходе АЦП?
- 4 На какую величину должно измениться напряжение на входе АЦП, чтобы результат преобразования на выходе АЦП изменился на единицу младшего разряда?

2 Лабораторная работа № 2. Моделирование и исследование работы цифрового прибора на микроконтроллере с дискретным датчиком

Цель работы: исследовать работу цифрового прибора на микроконтроллере с дискретным датчиком.

2.1 Основные теоретические положения

В качестве дискретного датчика используется кнопка, схема подключения которой к Arduino приведена на рисунке 2.1.

Внешний вид макета цифрового прибора на микроконтроллере с дискретным датчиком показан на рисунке 2.2.

При нажатии кнопки соединяются две точки цепи. Когда кнопка не нажата, нет никакой связи между двумя ее выводами, так что нижний её вывод подключен к земле через резистор R1 и на вывод D4 подается сигнал LOW, или логический нуль. При нажатии на кнопку устанавливается соединение между

двумя выводами кнопки и на вывод *D4* подается 5 В, он переходит в состояние HIGH, или логическую единицу.

Если не использовать резистор *R1* и оставить нижний вывод кнопки неподключенным к общей шине, то напряжение на нем будет изменяться хаотично. Это приведет к случайному формированию на *D4* напряжения высокого или низкого уровня.

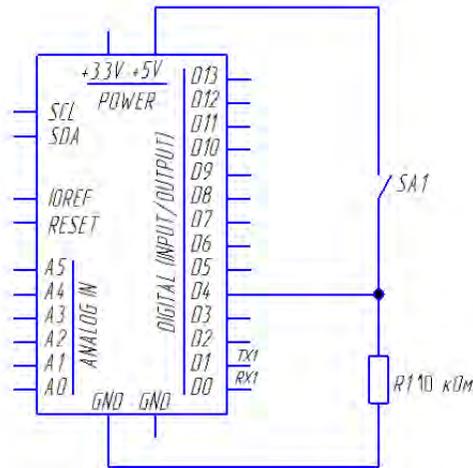


Рисунок 2.1 – Схема подключения кнопки к контроллеру Arduino

2.2 Порядок выполнения работы

2.2.1 Соедините проводниками кнопку, Arduino и резистор в соответствии с рисунком 2.1. Внешний вид устройства для исследования совместно с макетной платой приведен на рисунке 2.2.

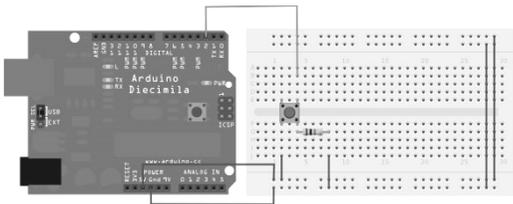


Рисунок 2.2 – Внешний вид подключения кнопки к плате Arduino

2.2.2 Подключите плату Arduino к ПК через USB-порт.

2.2.3 Запишите программу DigitalReadSerial в память контроллера Arduino, переслав её из ПК. В этой программе прежде всего нужно в функции настройки

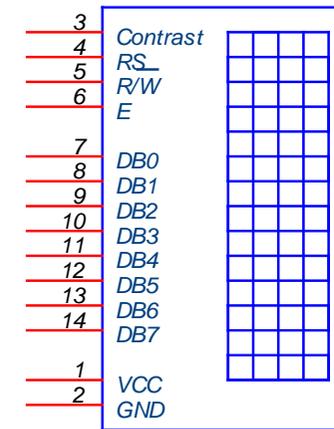


Рисунок 7.3 – Условное обозначение на схемах микросхемы ЖКИ WM-C1602N

Схема подключения к контроллеру Arduino LCD-дисплея 1602 показана на рисунке 7.4.

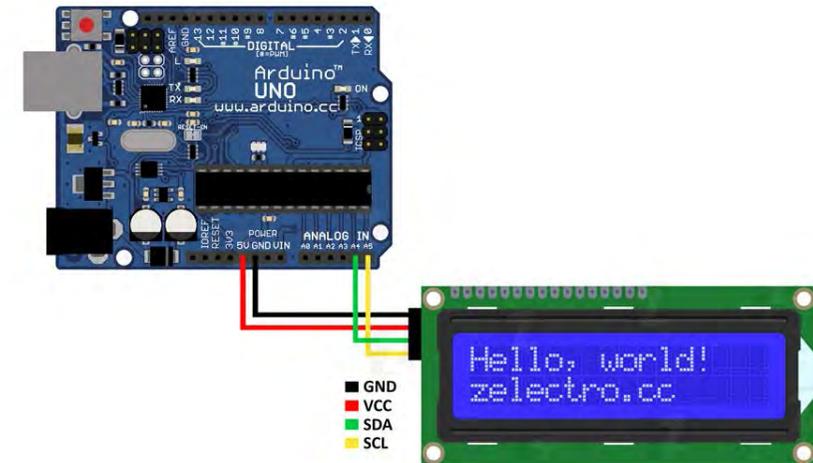


Рисунок 7.4 – Схема подключения к контроллеру Arduino LCD-дисплея 1602

Модуль оборудован четырехпиновым разъемом, в котором:

- SCL – последовательная линия тактирования (SerialClock);
- SDA – последовательная линия данных (SerialData);
- VCC – «+» питания;
- GND – «-» питания.

Интерфейс управления:

- GND – общий;
- VCC – питание;
- DIO – вход данных;
- CLK – вход стробирования данных.

Схема подключения к контроллеру Arduino Nano 4-разрядного LED-дисплея с I2C драйвером TM1637 показана на рисунке 7.2.

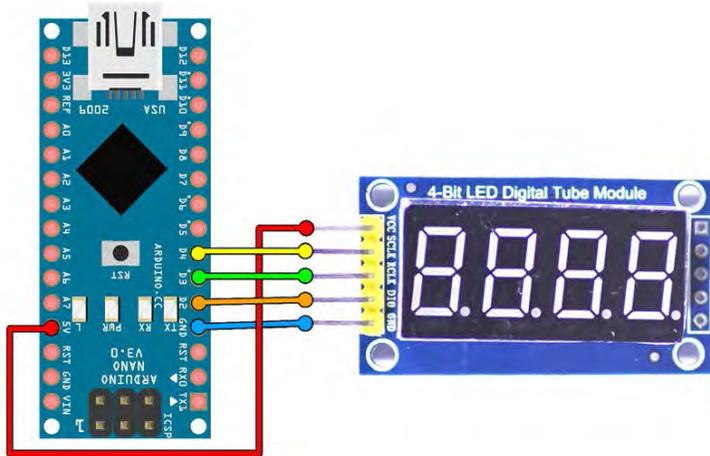


Рисунок 7.2 – Схема подключения к контроллеру Arduino Nano 4-разрядного LED-дисплея с I2C драйвером TM1637

На рисунке 7.3 показана микросхема жидкокристаллического индикатора (ЖКИ) WM-C1602N. Она имеет две строки по 16 символов. Индикатор работает под управлением встроенного контроллера. Назначение выводов у этой микросхемы следующее:

- VCC – питание +5 В;
- GND – общий вывод;
- Contrast – вход для регулировки контрастности изображения символов на индикаторе;
- RS – вход выбора регистра: 0 – регистр команд; 1 – регистр данных;
- RW – выбор режима записи или чтения;
- E – вход разрешения чтения/записи;
- DB0...DB7 – выводы для передачи данных.

включить последовательную передачу данных между ПК и Arduino со скоростью 9600 бит в секунду следующей строкой:

```
Serial.begin(9600);
```

Далее инициализируйте вывод D4. Он должен читать состояние кнопки, чтобы быть настроенным как вход:

```
pinMode(2,INPUT);
```

Далее, когда установки завершены, следует перейти к основному циклу кода. Когда кнопка нажата, 5 В будут свободно проходить через цепь, а когда она не нажата, вход будет соединен с землей через резистор 10 кОм. Это цифровой вход, следовательно, он может быть только во включенном (читается Arduino как 1 или HIGH) или в выключенном состоянии (читается Arduino как 0 или LOW). Промежуточных состояний не существует.

Первое, что нужно сделать в основном цикле, – это создать переменную для хранения информации, поступающей от кнопки. Поскольку сигнал, приходящий от кнопки, может иметь только значения 1 или 0, можно использовать целочисленный тип данных int. Назовите эту переменную sensorValue и присвойте ей значение, равное значению, считанному с входа 2. Это можно сделать с помощью следующей строки:

```
int sensorValue = digitalRead(4);
```

Как только Arduino считывает значение на входе, передайте его в ПК и выведите на экран как десятичное значение. Это можно сделать, используя команду Serial.println(), как показано в следующей строке:

```
Serial.println(sensorValue);
```

При открытии монитора последовательной передачи (Serial Monitor в среде Arduino) можно увидеть последовательность нулей, когда кнопка разомкнута, и последовательность единиц, когда кнопка замкнута.

Полный текст программы DigitalReadSerial

```
/* Читает состояние цифрового входа 4, выводит результат на монитор */
// цифровой вход 4 присоединен к кнопке. Назовем его:

int pushButton = 4;
// проведем необходимые установки:
void setup()
{
  // инициализируем последовательную передачу данных со скоростью
  9600 бит в секунду:

  Serial.begin(9600);
  // назначим вывод 2 входом:
  pinMode(pushButton, INPUT);
}
// основной цикл:
void loop()
```

```

{
// читаем значение на входе:
int buttonState = digitalRead(pushButton);
// выводим значение на монитор:
Serial.println(buttonState);
delay(10); // задержка для стабильного считывания
}

```

2.2.4 Нажимая или отпуская кнопку, наблюдайте на экране дисплея ПК появление 0 или 1, соответствующих положению кнопки.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему подключения кнопки к Arduino, выводы по результатам экспериментальных исследований.

Контрольные вопросы

- 1 Какое напряжение соответствует логическому 0?
- 2 Какое напряжение соответствует логической 1?
- 3 Зачем на схеме, приведенной на рисунке 2.1, используется резистор R1?

3 Лабораторная работа № 3. Моделирование и исследование работы порогового устройства на микроконтроллере

Цель работы: исследовать работу порогового устройства на микроконтроллере с аналоговым датчиком

3.1 Основные теоретические положения

Пороговое устройство реализуется на основе контроллера, показанного на рисунке 1.1.

В качестве источника сигнала, значение которого будет сравниваться с заданной пороговой величиной, используется аналоговый датчик потенциометрического типа, рассмотренный в лабораторной работе № 1.

3.2 Порядок выполнения работы

3.2.1 Присоедините три провода, припаянные к потенциометру, к плате Arduino, как показано на рисунках 1.3 и 1.4 (см. лабораторную работу № 1) в соответствии со схемой, приведенной на рисунке 3.1. Один из крайних выводов потенциометра соедините с землей (GND) Arduino, второй – с 5-вольтовым выходом Arduino. Третьим соедините аналоговый вход A0 и средний вывод потенциометра.

Контрольные вопросы

- 1 Какое напряжение соответствует логическому 0?
- 2 Какое напряжение соответствует логической 1?
- 3 Зачем на схеме, приведенной на рисунке 6.1, используется резистор R1?
- 4 Как программно изменяется временная задержка?
- 5 Как программно изменить частоту сигнала, подаваемого на динамик?
- 6 Опишите, каким образом с помощью светодиода определяется нажатое состояние кнопки.

7 Лабораторная работа № 7. Моделирование и исследование работы средств отображения информации цифрового прибора на микроконтроллере

Цель работы: исследовать работу средств отображения информации, используемых в цифровых приборах на микроконтроллере.

7.1 Основные теоретические положения

На рисунке 7.1 показан 4-разрядный семисегментный LED-индикатор с I2C драйвером TM1637.

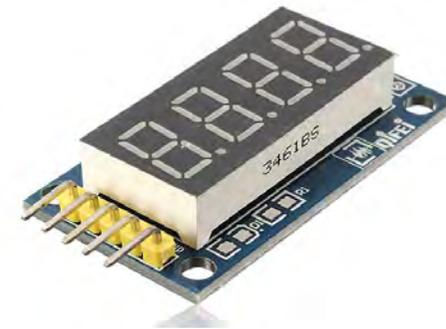


Рисунок 7.1 – Внешний вид 4-разрядного LED-дисплея с I2C драйвером TM1637

Особенности рассматриваемого индикатора:

- четыре семисегментные цифры и разделительные точки с общим анодом;
- всего четыре контакта подключения;
- регулируемая яркость – до восьми уровней яркости;
- логические уровни – 5 или 3,3 В;
- ток потребления – до 80 мА.

```
// К контакту D11 подключен светодиод
// Дадим ему имя:
```

```
int led = 11;
```

```
// Процедура установки запускается один раз при нажатии кнопки сброса:
```

```
void setup()
```

```
{
// инициализируем цифровой вывод D11 как выход:
```

```
pinMode(led, OUTPUT);
```

```
}
```

```
// основной цикл:
```

```
void loop()
```

```
{
```

```
digitalWrite(led, HIGH); // включить светодиод (HIGH означает высокий
уровень //напряжения)
```

```
delay(1000); // ждать секунду
```

```
digitalWrite(led, LOW); // выключить светодиод (LOW означает низкий
уровень //напряжения)
```

```
delay(1000); //ждать секунду
```

```
}
```

6.2.4 Изменяя 3...4 раза число в операторе delay(1000), наблюдайте за работой светодиода.

6.2.5 Добавьте к макету кнопку, подключив её так, как показано на рисунке 6.2 (см. лабораторную работу № 2).

6.2.6 Измените программу Indikator1 (светодиод) так, чтобы светодиод светился при нажатой кнопке и погасал при отпускании кнопки.

6.2.7 Запишите новую программу Indikator2 (светодиод-кнопка) в память контроллера Arduino, переслав её из ПК, и исследуйте работу светодиода, используемого в качестве индикатора состояния дискретного датчика (кнопки).

6.2.8 Запишите программу Indikator3 (динамик-светодиод) в память контроллера Arduino, переслав её из ПК, и исследуйте работу динамика, используемого в качестве звукового индикатора. Для этого измените в операторе **tone()** параметры, устанавливающие длительность временной задержки и паузы.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему подключения светодиода, динамика и кнопки к Arduino, программу Indikator1 и Dinamik, выводы по результатам экспериментальных исследований.

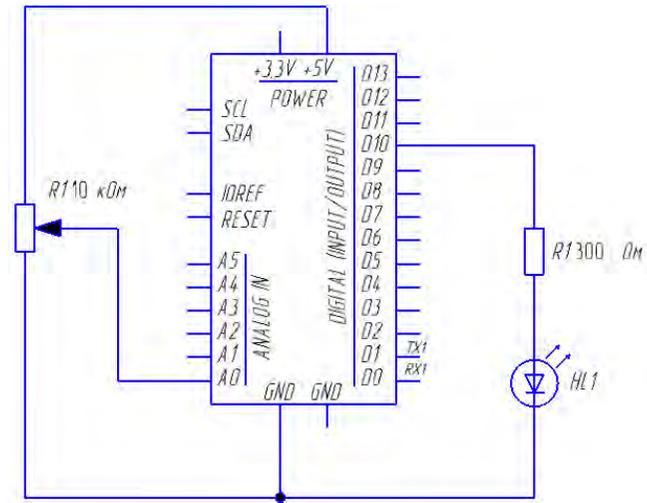


Рисунок 3.1 – Схема подключения потенциометра и светодиода к контроллеру Arduino

Если ручка потенциометра повернута до упора в одну сторону, то на вход АЦП подается 0 В и результат преобразования равен 0. Если ручка потенциометра повернута до упора в другую сторону, то на вход АЦП подается 5 В и результат преобразования равен 1023. В промежуточных положениях ручки потенциометра АЦП возвращает число между 0 и 1023, которое пропорционально напряжению на его среднем выводе.

Если пороговая величина задается напряжением, с которым сравнивается выходное напряжение аналогового датчика, то в программе необходимо предусмотреть преобразование цифрового кода, формируемого АЦП, в напряжение, эквивалентное входному. В программе также необходимо предусмотреть слежение за текущим значением входного сигнала АЦП в режиме реального времени и формирование сигнала, например, логической 1, при его совпадении с заданным пороговым значением.

3.2.2 Подключите плату Arduino к ПК через USB-порт.

3.2.3 Разработайте программу CompareVoltage1. В данной программе необходимо задать пороговое значение напряжения, обеспечить сравнение текущего значения напряжения, поступающего от аналогового датчика, с пороговым и вывод текущих значений на монитор ПК. При их совпадении и превышении текущего значения над пороговым на экране ПК должны отображаться пороговое и текущее значение напряжения.

Полный текст программы CompareVoltage1

/* Текст программы «CompareVoltage1» к лабораторной работе № 3:

Считывает значения напряжения с аналогового датчика (с вывода *A0*), сравнивает с пороговым значением и выводит текущее значение напряжения на монитор до достижения порогового значения, затем значение порога и текущего значения напряжения.*/*

```
float porog = 2.58; //задаем значение установки (порогового значения
напряжения)
void setup()
{
  // инициализируем последовательную передачу данных со скоростью
9600 бит в секунду:
  Serial.begin(9600);
}
// основной цикл:
void loop()
{
  // читаем значение на аналоговом входе 0:
  int sensorValue = analogRead(A0);
  // преобразуем считанное значение (которое меняется от 0 до 1023) в
напряжение (0–5 В):
  float voltage = sensorValue * (5.0 / 1023.0);
  if (voltage < porog)
  {
    Serial.print("voltage = "); // выводим на монитор текст
    Serial.println(voltage); // выводим на монитор считанное текущее
значение напряжения
    delay(100); // задержка в промежутке между считываниями для
стабильности:
  }
  else
  {
    Serial.print("porog = "); // выводим на монитор текст
    Serial.println(porog); // выводим на монитор пороговое значение
    Serial.print("\t\t"); // делаем табуляцию
    Serial.print("voltage = "); // выводим на монитор текст
    Serial.println(voltage); //выводим на монитор считанное текущее
значение напряжения
    delay(100); // задержка в промежутке между считываниями для
стабильности:
  }
}
```

3.2.4 Вторую программу CompareVoltage2 разработайте самостоятельно, предусмотрев в ней при совпадении напряжений включение светодиода.

3.2.5 Запишите разработанную программу CompareVoltage2 в память контроллера Arduino, переслав её из ПК.

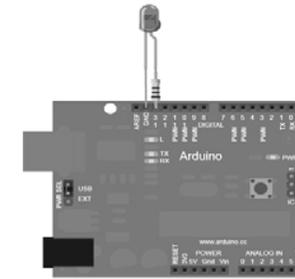


Рисунок 6.2 – Внешний вид подключения светодиода к плате Arduino

6.2 Порядок выполнения работы

6.2.1 Чтобы построить электрическую цепь, соедините первый вывод резистора $R1 = 300 \text{ Ом}$ с цифровым выводом *D11* на Arduino, второй – с длинным выводом светодиода (анодом). Короткий вывод светодиода (катод) соедините с контактом GND Arduino. Динамик подключите одним выводом к выводу *D12* Arduino, другим – к общей шине (см. рисунок 6.1).

6.2.2 Подключите плату Arduino к ПК через USB-порт.

6.2.3 Запишите программу Indikator1 в память контроллера Arduino, переслав её из ПК.

В данной программе в первую очередь необходимо инициализировать вывод *D12* как выход строкой

```
pinMode(11, OUTPUT);
```

В главном цикле включаем светодиод строкой

```
digitalWrite(11, HIGH);
```

Это подаст 5 В на контакт *D11*. На выводах светодиода появится разность потенциалов, и он засветится. Для того чтобы выключить светодиод, следует написать следующую строку:

```
digitalWrite(11, LOW);
```

Это возвратит контакт *D11* в состояние 0 В и выключит светодиод. Для того чтобы человеческий глаз мог видеть изменения, между включением и выключением должно пройти некоторое время. Командой delay() можно приказать контроллеру ничего не делать определенное время (в данном случае 1000 мс или 1 с).

Полный текст программы Indikator1

```
/*
Включает светодиод на 1 с, затем выключает на 1 с.
*/
```

6 Лабораторная работа № 6. Моделирование и исследование работы индикаторных устройств цифрового прибора на микроконтроллере

Цель работы: исследовать работу индикаторных устройств цифрового прибора на микроконтроллере.

6.1 Основные теоретические положения

Макет цифрового прибора с индикаторными устройствами собирается на макетной плате на основе контроллера Arduino Leonardo.

В качестве индикаторных устройств используется светодиод HL1 и динамик BA1. Схема их подключения к Arduino приведена на рисунке 6.1.

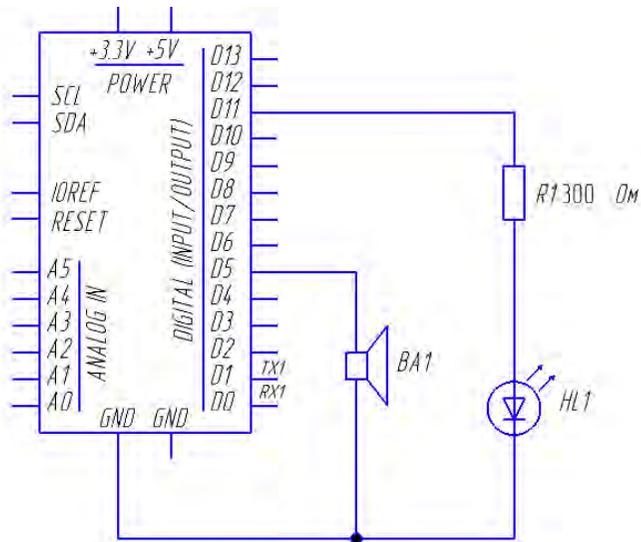


Рисунок 6.1 – Схема подключения светодиода и динамика к микроконтроллеру Arduino

Внешний вид макета цифрового прибора на микроконтроллере со светодиодом в качестве индикаторного устройства показан на рисунке 6.2.

3.2.6 Вращая ручку потенциометра, наблюдайте на экране дисплея ПК изменение значений напряжения, а также срабатывание светодиода при достижении входного напряжения заданного порогового значения.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему подключения потенциометра к Arduino, разработанную программу, выводы по результатам экспериментальных исследований.

Контрольные вопросы

- 1 Какова разрядность встроенного в контроллер Arduino АЦП?
- 2 В каком диапазоне изменяется напряжение на выходе потенциометра?
- 3 В каком диапазоне формируется результат преобразования на выходе АЦП?
- 4 В каком диапазоне выводится значение напряжения на дисплей ПК и почему?
- 5 На какую величину должно измениться напряжение на входе АЦП, чтобы результат преобразования на выходе АЦП изменился на единицу младшего разряда?

4 Лабораторная работа № 4. Моделирование и исследование работы генератора прямоугольных импульсов на микроконтроллере

Цель работы: исследовать работу генератора прямоугольных импульсов на микроконтроллере.

4.1 Основные теоретические положения

Макет генератора прямоугольных импульсов собирается на макетной плате на основе контроллера Arduino Leonardo.

В качестве индикаторного устройства используется светодиод. Схема его подключения к Arduino приведена на рисунке 4.1.

Внешний вид макета генератора на микроконтроллере со светодиодом в качестве индикаторного устройства показан на рисунке 4.2.

Можно воспользоваться встроенным светодиодом, присоединенным к выводу D13 и расположенным непосредственно на плате Arduino.

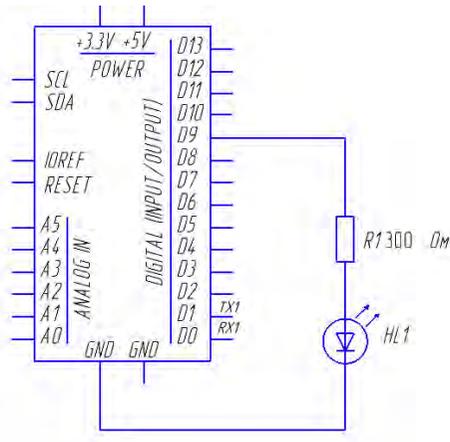


Рисунок 4.1 – Схема подключения светодиода к контроллеру Arduino

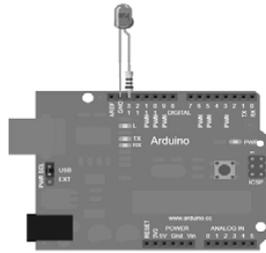


Рисунок 4.2 – Внешний вид подключения светодиода к плате Arduino

4.2 Порядок выполнения работы

4.2.1 Чтобы построить электрическую цепь, соедините первый вывод резистора $R1 = 300 \text{ Ом}$ с цифровым выводом $D9$ на Arduino, второй – с длинным выводом светодиода (анодом). Короткий вывод светодиода (катод) соедините с контактом GND Arduino.

4.2.2 Подключите плату Arduino к ПК через USB-порт.

4.2.3 Запишите программу Generator в память контроллера Arduino, переслав её из ПК.

В данной программе в первую очередь необходимо инициализировать вывод $D9$ как выход строкой

```
pinMode(9, OUTPUT);
```

В главном цикле включайте светодиод строкой **digitalWrite(9, HIGH);**

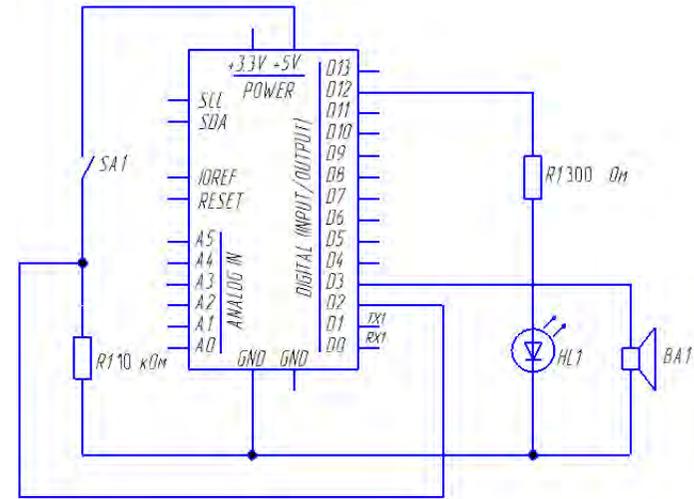


Рисунок 5.3 – Схема подключения кнопки, светодиода и динамика к Arduino

5.2.9 Запишите программу Zaderjka3 в память контроллера Arduino, переслав её из ПК, и исследуйте её выполнение.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему подключения светодиода и кнопки к Arduino, программы временных задержек, выводы по результатам экспериментальных исследований.

Контрольные вопросы

- 1 Какое напряжение соответствует логическому 0?
- 2 Какое напряжение соответствует логической 1?
- 3 Зачем на схеме, приведенной на рисунке 5.1, используется резистор $R1$?
- 4 Как программно изменяется временная задержка?
- 5 Как определить временную задержку с помощью осциллографа?
- 6 Как можно реализовать аппаратно-программную задержку?
- 7 Каким оператором задается звуковой сигнал в программе Zaderjka3 и как устанавливаются его параметры?

// Дадим ему имя:

```
int led = 11;
```

// Процедура установки запускается один раз при нажатии кнопки сброса:

```
void setup()
```

```
{
```

// инициализируем цифровой вывод D11 как выход:

```
pinMode(led, OUTPUT);
```

```
}
```

// основной цикл:

```
void loop()
```

```
{
```

```
  digitalWrite(led, HIGH); // включить светодиод (HIGH означает высокий
уровень //напряжения )
```

```
  delay(1000); // ждать секунду
```

```
  digitalWrite(led, LOW); // выключить светодиод (LOW означает низкий
уровень //напряжения )
```

```
  M1:
```

```
    delay(1000); //ждать секунду
```

```
  goto M1;
```

```
}
```

5.2.3 Подключите к выводу D11 Arduino осциллограф.

5.2.4 Изменяя 4...5 раз число в операторе delay(1000), наблюдайте за работой генератора временной задержки с помощью осциллографа и светодиодного индикатора.

5.2.5 Аппаратно-программную задержку можно реализовать, если к устройству, изображенному на рисунке 5.1, добавить кнопку (рисунок 5.3).

5.2.6 Разработайте программу Zaderjka2, в которой временная задержка начинает вырабатываться после нажатия кнопки SA1. Длительность временного интервала задается преподавателем, а его величина должна индексироваться свечением светодиода HL1.

5.2.7 Запишите программу Zaderjka2 в память контроллера Arduino, переслав её из ПК, и исследуйте её выполнение.

5.2.8 Разработайте программу Zaderjka3, в которой временная задержка начинает вырабатываться после загрузки программы или, если программа загружена, после подачи на контроллера Arduino напряжения питания. Длительность временного интервала, в течение которого генерируется звук, и длительность паузы задается преподавателем. Генерация звука должна сопровождаться свечением светодиода HL1. Для генерации звука к контроллеру Arduino следует подключить динамик, как показано на рисунке 5.3.

Это подаст 5 В на контакт D9. На выводах светодиода появится разность потенциалов, и он засветится. Для того чтобы выключить светодиод, следует написать следующую строку:

```
digitalWrite(9, LOW);
```

Это возвратит контакт D9 в состояние 0 В и выключит светодиод. Для того чтобы человеческий глаз мог видеть изменения, между включением и выключением должно пройти некоторое время. Командой delay() можно приказывать контроллеру ничего не делать определенное число времени (в данном случае 1000 мс или 1 с).

Полный текст программы Generator

```
/*
```

```
Включает светодиод на 1 с, затем выключает на 1 с.
```

```
*/
```

```
// К контакту D9 подключен светодиод
```

```
// Дадим ему имя:
```

```
int led = 9;
```

// Процедура установки запускается один раз при нажатии кнопки сброса:

```
void setup()
```

```
{
```

// инициализируем цифровой вывод D9 как выход:

```
pinMode(led, OUTPUT);
```

```
}
```

// основной цикл:

```
void loop()
```

```
{
```

```
  digitalWrite(led, HIGH); // включить светодиод (HIGH означает высокий
уровень //напряжения )
```

```
  delay(1000); // ждать секунду
```

```
  digitalWrite(led, LOW); // выключить светодиод (LOW означает низкий
уровень //напряжения )
```

```
  delay(1000); //ждать секунду
```

```
}
```

4.2.4 Подключите к выводу D9 Arduino осциллограф.

4.2.5 Изменяя 4...5 раз число в операторе delay(1000), наблюдайте за работой генератора на экране осциллографа и с помощью светодиодного индикатора.



Содержание отчета

Отчет о проделанной работе должен содержать: название работы, цель работы, схему подключения светодиода и кнопки к Arduino, программу Generator, выводы по результатам экспериментальных исследований.

Контрольные вопросы

- 1 Какое напряжение соответствует логическому 0?
- 2 Какое напряжение соответствует логической 1?
- 3 Зачем на схеме, приведенной на рисунке 4.1, используется резистор R1?
- 4 Как программно изменяется временная задержка?
- 5 Как определить период следования импульсов с помощью осциллографа?

5 Лабораторная работа № 5. Моделирование и исследование работы генератора временных интервалов на микроконтроллере

Цель работы: исследовать работу генератора временных интервалов на микроконтроллере.

5.1 Основные теоретические положения

Макет генератора временных интервалов собирается на макетной плате на основе контроллера Arduino Leonardo.

В качестве индикаторного устройства используется светодиод. Схема его подключения к Arduino приведена на рисунке 5.1.

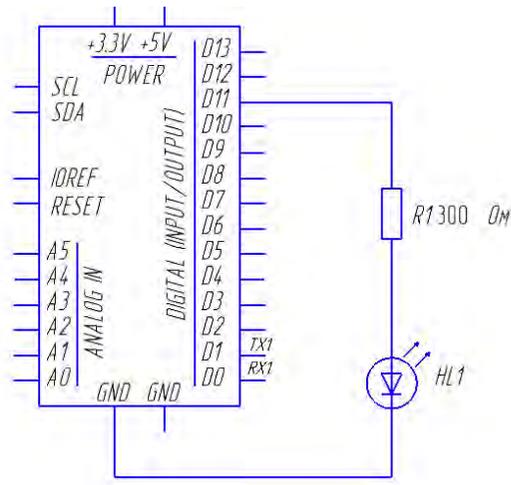


Рисунок 5.1 – Схема подключения светодиода к Arduino

Внешний вид макета цифрового прибора на микроконтроллере со светодиодом в качестве индикаторного устройства показан на рисунке 5.2.

Чтобы построить электрическую цепь, соедините первый вывод резистора R1= 300 Ом с цифровым выводом D11 на Arduino, второй – с длинным выводом светодиода (анодом). Короткий вывод светодиода (катод) соедините с контактом GND Arduino.

Можно воспользоваться встроенным светодиодом, присоединенным к выводу D13.

5.2 Порядок выполнения работы

5.2.1 Подключите плату Arduino к ПК через USB-порт.

5.2.2 Запишите программу Zaderjka1 в память контроллера Arduino, переслав её из ПК, и исследуйте её выполнение.

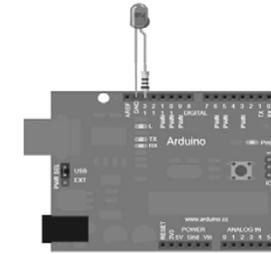


Рисунок 5.2 – Внешний вид подключения светодиода к плате Arduino

В данной программе в первую очередь необходимо инициализировать вывод D11 как выход строкой

pinMode(11, OUTPUT);

В главном цикле включаем светодиод строкой

digitalWrite(11, HIGH);

Это подаст 5 В на контакт D11. На выводах светодиода появится разность потенциалов, и он засветится. Для того чтобы выключить светодиод, следует написать следующую строку:

digitalWrite(11, LOW);

Это возвратит контакт D11 в состояние 0 В и выключит светодиод. Для того чтобы сформировать временной интервал заданной длительности, величина которого будет индицироваться свечением светодиода, воспользуйтесь оператором Zaderjka1 delay().

Полный текст программы Zaderjka1

```
/*
Включает светодиод на время, указанное в операторе delay().
*/
// К контакту D10 подключен светодиод
```