ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Физические методы контроля»

РАДИОЭЛЕКТРОННЫЕ УСТРОЙСТВА В БИОСИСТЕМАХ

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



Могилев 2018

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. физ.-мат. наук, доц. Ф. М. Трухачев

B методических рекомендациях кратко изложены изучаемые теоретические сведения, приведен порядок выполнения экспериментальных исследований, указана структура отчета о выполненной работе и изложен список контрольных вопросов для самопроверки по каждой теме. Составлены в соответствии с рабочей программой по дисциплине «Радиоэлектронные устройства В биосистемах» для студентов направления подготовки 12.03.04 «Биотехнические системы и технологии» дневной формы обучения.

Учебно-методическое издание

РАДИОЭЛЕКТРОННЫЕ УСТРОЙСТВА В БИОСИСТЕМАХ

Ответственный за выпуск

С. С. Сергеев

А. А. Подошевко

Е.С. Лустенкова

Технический редактор

Компьютерная верстка

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс. . 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

	-	_	1111	
-		-		
	-	_		
	-			
				7

Содержание

3

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	8
3 Лабораторная работа № 3. Моделирование и исследование работы	
порогового устройства на микроконтроллере	11
4 Лабораторная работа № 4. Моделирование и исследование работы	
генератора прямоугольных импульсов на микроконтроллере	14
5 Лабораторная работа № 5. Моделирование и исследование работы	
генератора временных интервалов на микроконтроллере	17
6 Лабораторная работа № 6. Моделирование и исследование работы	
индикаторных устройств цифрового прибора на микроконтроллере	21
7 Лабораторная работа № 7. Моделирование и исследование работы	
средств отображения информации цифрового прибора на микроконтроллере	24
8 Лабораторная работа № 8. Моделирование и исследование работы	
исполнительного устройства цифрового прибора на микроконтроллере	28
Список литературы	32

YHUBE PCUTET

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 на16 МГц, разъем микроUSB, 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_{gbix} от 0 до + 5 В, которое затем подается на аналоговый вход Arduino (рисунок 1.2).



Рисунок 1.2 – Схема подключения потенциометра к источнику питания

1.2 Порядок выполнения работы

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, третьим соедините аналоговый вход А0 и средний вывод потенциометра.



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.

2 Подключите плату Arduino к ПК через USB-порт.

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

Serial.begin(9600);

Далее в основном цикле программы следует создать переменную для хранения значения напряжения (которая может изменятся 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. Это можно сделать с помощью команды 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()
{
 // читаем значение на аналоговом входе 0:
 int sensorValue = analogRead(A3);
 // выводим на монитор считанное значение:
```

Serial.println(sensorValue);

delay(1); промежутке // задержка В между считываниями ДЛЯ стабильности: }

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 формируется преобразования результат каком диапазоне

на выходе АЦП?

4 На какую величину должно измениться напряжение на входе АЦП, чтобы результат преобразования на выходе АЦП изменился на единицу младшего разряда?

8

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

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

2.1 Основные теоретические положения

В качестве дискретного датчика будем использовать кнопку. Схема её подключения к Arduino приведена на рисунке 2.1.





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



Рисунок 2.2 - Внешний вид подключения кнопки к плате Arduino

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 D2 подается сигнал LOW, или логического нуля. При нажатии на кнопку устанавливается соединение между двумя выводами кнопки и на нижний вывод подается 5 В, он переходит в состояние HIGH, или логической 2.

Если не использовать резистор *R*1 и оставить нижний вывод кнопки неподключенным к общей шине, то напряжение на нем будет изменяться хаотично. Это приведет к случайному формированию на нижнем выводе напряжения высокого или низкого уровня.

2.2 Порядок выполнения работы

1 Соедините три провода, идущие от кнопки, с Arduino. Первые два присоедините к двум длинным вертикальным рядам на краю макетной платы для получения доступа к 5 В питания и общей шине (GND). Третьим проводом соедините цифровой вывод D4 Arduino с одним из выводов кнопки. Этот же вывод кнопки подсоедините через резистор R1 (10 кОм) к общей шине (GND). Второй вывод кнопки подсоедините к 5 В питания.

2 Подключите плату Arduino к ПК через USB-порт.

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

Теперь, когда установки завершены, переходите 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, поэтому 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 считает 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 D4, выводит результат на монитор цифровой вход D4 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 D4 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 }

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

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 А0 и средний вывод потенциометра.





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.

2 Подключите плату Arduino к ПК через USB-порт.

3 Разработайте программу CompaerVoltage1. 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 CompaerVoltage1

/* 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 А0:

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)</pre>

{

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:

}

}

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

5 Запишите разработанную программу CompaerVoltage2 в память контроллера Arduino, переслав её из ПК.

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.



Рисунок 4.1 - Схема подключения светодиода к контроллеру Arduino

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

Можно воспользоваться встроенным светодиодом, присоединенным к выводу *D*13 и расположенным непосредственно на плате Arduino.



Рисунок 4.2 – Внешний вид подключения светодиода к плате Arduino

4.2 Порядок выполнения работы

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

2 Подключите плату Arduino к ПК через USB-порт.

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

В данной программе в первую очередь необходимо инициализировать вывод *D*9 как выход строкой

pinMode(9, OUTPUT);

В главном цикле включается светодиод строкой

digitalWrite(9, HIGH);

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 приказать контроллеру ничего не делать определенное время (в данном случае 1000 мс или 1 с).

Полный текст программы Generator

/*

Включает светодиод на одну секунду, затем выключает на одну секунду. */

```
// 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 Подключите к выводу D9 Arduino осциллограф.

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.



Рисунок 5.2 – Внешний вид подключения светодиода к плате Arduino

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

- 1 Подключите плату Arduino к ПК через USB-порт.
- 2 Запишите программу Zaderjka1 в память контроллера Arduino, переслав

её из ПК, и исследуйте её выполнение.

В данной программе в первую очередь необходимо инициализировать вывод *D*11 как выход строкой

pinMode(11, OUTPUT);

В главном цикле включается светодиод строкой:

digitalWrite(10, HIGH);

Это подаст 5 В на контакт *D*11. На выводах светодиода появится разность потенциалов, и он засветится. Для того чтобы выключить светодиод, следует написать следующую строку:

digitalWrite(10, 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, следует воспользоваться оператором Zaderjka1 delay().

Полный текст программы Zaderjka1

```
/*
Включает светодиод на время, указанное в операторе delay().
*/
// 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 *D*11 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
goto M1;
}
```

3 Подключите к выводу D11 Arduino осциллограф.

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

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

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

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

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 задаются преподавателем. Генерация звука должна сопровождаться свечением светодиода HL1. Для генерации звука к контроллеру Arduino следует подключить динамик, как показано на рисунке 5.3.



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

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 *R*1?

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?

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.

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



Рисунок 6.1 – Схема подключения светодиода и динамика к микроконтроллеру Arduino



Рисунок 6.2 – Внешний вид подключения светодиода к плате Arduino

6.2 Порядок выполнения работы

1 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. Динамик подключите одним выводом к выводу D5 Arduino, другим – к общей шине (см. рисунок 6.1).

2 Подключите плату Arduino к ПК через USB-порт.

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

В данной программе в первую очередь необходимо инициализировать вывод *D*11 как выход строкой

pinMode(11, OUTPUT);

В главном цикле включается 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 одну секунду, затем выключает на одну секунду. */ // 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 *D*11 как выход:

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

```
// основной цикл:
void loop()
```

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

// ждать секунду delay(1000);

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

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

```
}
```

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

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

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

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).

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.

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

1 На рисунке 7.1 показана микросхема трёхразрядного цифро-буквенного светодиодного семисегментного индикатора с разъединёнными катодами ТОТ-3361АН-IN. В корпусе одной микросхемы все одноименные сегменты соединены между собой.

В одном корпусе располагаются три индикатора. Питание на каждый индикатор подаётся раздельно и поочередно, что обеспечивает их работу в динамическом режиме. Прямой номинальный ток через каждый сегмент – 10 мА, прямое напряжение на сегменте (при номинальном токе) – 2 В.



Рисунок 7.1 – Конструкция, условное обозначение и электрическая схема микросхемы цифро-буквенного светодиодного индикатора ТОТ-3361АН-IN

2 На рисунке 7.2 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;

- Е – вход разрешения чтения/записи;

– DB0...DB7 – выводы для передачи данных.



Рисунок 7.2 – Условное обозначение на схемах микросхемы ЖКИ WM-C1602N

3 На рисунке 7.3 показано подключение двух микросхем E40561 (HG1, HG2) к микропроцессору DD1. Каждая микросхема содержит по четыре светодиодных семисегментных индикатора с разъединёнными катодами. В корпусе одной микросхемы все одноименные сегменты соединены между собой и сигналы, управляющие их работой, подаются с выходов регистра DD3 через токоограничивающие резисторы R11...R18. Питание на каждый индикатор подаётся раздельно через транзисторные ключи VT1...VT8 от источника Vcc = +5 В на выводы 6, 8, 9 и 12 микросхем HG1 и HG2. Индикаторы работают в динамическом режиме, т. е. включаются поочерёдно.

4 Схема подключения к контроллеру Arduino Leonardo микросхемы трёхразрядного цифро-буквенного светодиодного семисегментного индикатора с разъединёнными катодами ТОТ-3361АН-IN показана на рисунке 7.4.

Порядок выполнения работы

1 Подключите плату Arduino к ПК через USB-порт.

2 Запишите программу D-led-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 Leonardo микросхемы трёхразрядного цифро-буквенного светодиодного индикатора TOT-3361AH-IN, результаты экспериментальных исследований, выводы.

26





Рисунок 7.3 – Схема подключения к микроконтроллеру MCS светодиодных семисегментных индикаторов E40561 (HG1, HG2)



Рисунок 7.4 – Схема подключения к контроллеру Arduino Leonardo микросхемы трёхразрядного цифро-буквенного светодиодного семисегментного индикатора с разъединёнными катодами TOT-3361AH-IN

Контрольные вопросы

1 Опишите устройство и принцип работы цифро-буквенного светодиодного индикатора ТОТ-3361АН-IN.

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 ЖКИ-индикатора.

5 Опишите работу всех элементов, обеспечивающих вывод информации на цифровые индикаторы E40561 (HG1, HG2) (см. рисунок 7.3).

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

1 Для создания одно- или многокоординатных систем сканирования в приборах неразрушающего контроля используются специальные электроприводы. Если от привода требуется повышенная точность позиционирования вала и высокий момент удержания при нулевой скорости и не требуется повышенная максимальная скорость вращения, то целесообразно использовать шаговый привод вращения. Параметры блока управления определяются числом обмоток, потребляемым током и порядком коммутации обмоток шагового двигателя (ШД). Технический уровень блока управления определяется в основном примененной в них элементной базой.

Частота вращения и суммарный угол поворота вала ШД пропорциональны частоте и числу поданных импульсов управления соответственно. При отсутствии управляющих импульсов ШД находится в режиме фиксированной стоянки и сохраняет конечные результаты предыдущих перемещений. Привод с ШД сочетает возможности глубокого регулирования частоты вращения с возможностью числового задания и надежной фиксации конечных координат.

По конструкции и принципу работы ШД подразделяются на двигатели механического и электромашинного типов.

Основными характеристиками ШД являются предельная динамическая и предельная механическая характеристики, которые выражают взаимосвязь между приемистостью и вращающим моментом ШД.

Помимо самого ШД, необходимо также специальное электронное устройство для управления его работой, которое обеспечивает преобразование унитарной последовательности импульсов в *m*-фазную систему напряжений, питающих обмотки ШД (рисунок 8.1) через усилители мощности.

Такие устройства управления разрабатываются на основе микроконтроллеров. Подключение ШД к микроконтроллерной системе осуществляют следующим образом (рисунок 8.2).



Рисунок 8.1 – Электрическая схема ШД



Рисунок 8.2 – Схема подключения к микроконтроллеру МСЅ шагового двигателя

2 Чтобы обеспечить реверсивное вращение вала ШД, надо на его обмотки подавать последовательность импульсов тока определенной величины. Временные диаграммы такой последовательности импульсов показаны на рисунке 8.3.

Электронная библиотека Белорусско-Российского университета http://e.biblio.bru.by/



Рисунок 8.3 – Порядок чередования импульсов на выводах ШД

Порядок выполнения работы

1 Подключите стенд к ЭВМ.

2 Подключите стенд с помощью адаптера к сети 220 В.

3 Загрузите в стенд программу, инициализирующую работу стенда.

4 Загрузите в стенд программу диагностики работы стенда.

5 Исследуйте работу в различных режимах подключенного к стенду шагового двигателя.

6 Исследуйте работу шагового двигателя с концевым датчиком.

Содержание отчета

Отчет о проделанной работе должен содержать название работы, цель работы, схему включения ШД, результаты экспериментальных исследований, выводы.

Контрольные вопросы

1 Опишите устройство и принцип работы шагового двигателя.

2 Опишите принцип управления работой шагового двигателя с помощью микроконтроллера.

3 Как осуществляется регулирование частотой и направлением вращения вала шагового двигателя?

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 c.