

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Невинномысский технологический институт (филиал)**

Методические указания для проведения лабораторных работ
по дисциплине «Программное обеспечение микропроцессорных систем управления»

Направление подготовки 15.03.04 Автоматизация технологических процессов и произ-
водств
Квалификация выпускника – бакалавр

Методические указания предназначены для проведения лабораторных работ по дисциплине «Программное обеспечение микропроцессорных систем управления» для студентов направления подготовки 15.03.04 Автоматизация технологических процессов и производств и соответствуют требованиям ФГОС ВО направления подготовки магистров.

Составитель: старший преподаватель кафедры ИСЭА Д.В.Самойленко

Содержание

Лабораторная работа №1.....	4
Лабораторная работа №2.....	6
Лабораторная работа №3.....	8
Лабораторная работа №4.....	10
Лабораторная работа №5.....	12
Лабораторная работа №6.....	14
Лабораторная работа №7.....	16

Лабораторная работа №1

Использование цифрового вывода

Цель работы: ознакомиться с функциями управления выводом.

Краткие сведения из теории

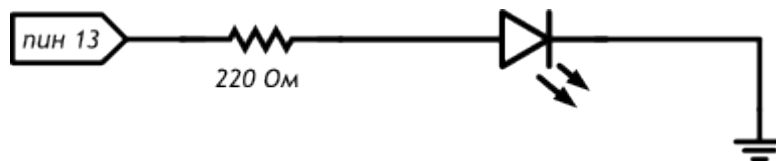


Рисунок 1.1 – Принципиальная схема

Листинг программы:

```
void setup()
{
  // настраиваем пин №13 в режим выхода,
  // т.е. в режим источника напряжения
  pinMode(13, OUTPUT);
}

void loop()
{
  // подаём на пин 13 «высокий сигнал» (англ. «high»), т.е.
  // выдаём 5 вольт. Через светодиод побежит ток.
  // Это заставит его светиться
  digitalWrite(13, HIGH);

  // задерживаем (англ. «delay») микроконтроллер в этом
  // состоянии на 100 миллисекунд
  delay(100);

  // подаём на пин 13 «низкий сигнал» (англ. «low»), т.е.
  // выдаём 0 вольт или, точнее, приравниваем пин 13 к земле.
  // В результате светодиод погаснет
  digitalWrite(13, LOW);

  // замираем в этом состоянии на 900 миллисекунд
  delay(900);

  // после «размораживания» loop сразу же начнёт исполняться
  // вновь, и со стороны это будет выглядеть так, будто
  // светодиод мигает раз в 100 мс + 900 мс = 1000 мс = 1 сек
}
```

Задание на лабораторную работу

Светодиод должен светиться полсекунды, а пауза между вспышками равна одной секунде.

Измените код примера так, чтобы светодиод включался на три секунды после запуска устройства, а затем мигал в стандартном режиме.

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. Что будет, если подключить к земле анод светодиода вместо катода?
2. Что будет, если подключить светодиод с резистором большого номинала (например, 10 кОм)?
3. Что будет, если подключить светодиод без резистора?
4. Зачем нужна встроенная функция `pinMode`? Какие параметры она принимает?
5. Зачем нужна встроенная функция `digitalWrite`? Какие параметры она принимает?
6. С помощью какой встроенной функции можно заставить микроконтроллер ничего не делать?
7. В каких единицах задается длительность паузы для этой функции?

Лабораторная работа №2

Использование ШИМ

Цель работы: ознакомиться с функциями управления выводом с ШИМ.

Краткие сведения из теории

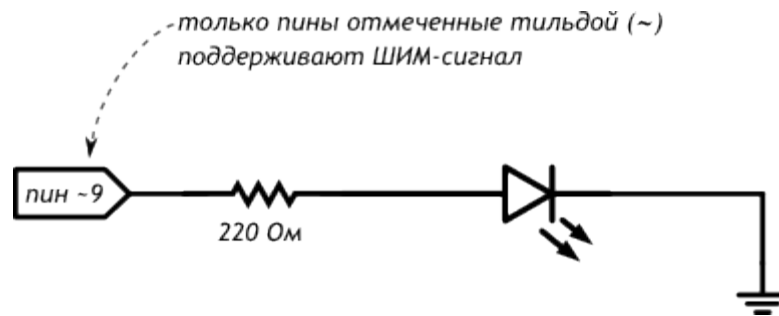


Рисунок 2.1 – Принципиальная схема

Листинг программы:

```
// даём разумное имя для пина №9 со светодиодом
// (англ. Light Emitting Diode или просто «LED»)
// Так нам не нужно постоянно вспоминать куда он подключён
#define LED_PIN 9

void setup()
{
  // настраиваем пин со светодиодом в режим выхода,
  // как и раньше
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  // выдаём неполное напряжение на светодиод
  // (он же ШИМ-сигнал, он же PWM-сигнал).
  // Микроконтроллер переводит число от 0 до 255 к напряжению
  // от 0 до 5 В. Например, 85 — это 1/3 от 255,
  // т.е. 1/3 от 5 В, т.е. 1,66 В.
  analogWrite(LED_PIN, 85);
  // держим такую яркость 250 миллисекунд
  delay(250);

  // выдаём 170, т.е. 2/3 от 255, или иными словами — 3,33 В.
  // Больше напряжение — выше яркость!
  analogWrite(LED_PIN, 170);
  delay(250);

  // все 5 В — полный накал!
  analogWrite(LED_PIN, 255);
  // ждём ещё немного перед тем, как начать всё заново
  delay(250);
}
```

Задание на лабораторную работу

Отключите питание, отключите светодиод от 9-го порта и подключите к 11-му. Измените программу так, чтобы схема снова заработала

Измените код программы так, чтобы в течение секунды на светодиод последовательно подавалось усреднённое напряжение 0, 1, 2, 3, 4, 5 В

Возьмите еще один светодиод, резистор на 220 Ом и соберите аналогичную схему на этой же макетке, подключив светодиод к пину номер 3 и другому входу GND, измените программу так, чтобы светодиоды мигали в противофазу: первый выключен, второй горит максимально ярко и до противоположного состояния

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. Какие из следующих идентификаторов корректны и не вызовут ошибку?
13pin
MOTOR_1
контакт_светодиода
sensor value
leftServo
my-var
distance_eval2
2. Что произойдет, если создать директиву #define HIGH LOW?
3. Почему мы не сможем регулировать яркость светодиода, подключенного к порту 7?
4. Какое усреднённое напряжение мы получим на пине 6, если вызовем функцию analogWrite(6, 153)?
5. Какое значение параметра value нужно передать функции analogWrite, чтобы получить усреднённое напряжение 2 В?

Лабораторная работа №3

Светильник с управляемой яркостью

Цель работы: ознакомиться с принципами аналогового ввода.

Краткие сведения из теории

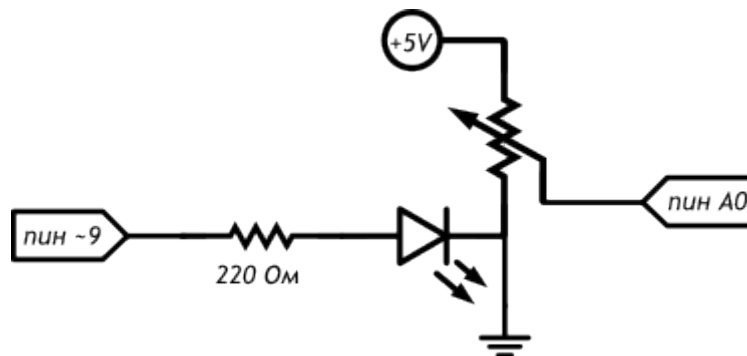


Рисунок 3.1 – Принципиальная схема

Листинг программы:

```
// даём разумные имена для пинов со светодиодом
// и потенциометром (англ potentiometer или просто «pot»)
#define LED_PIN 9
#define POT_PIN A0

void setup()
{
  // пин со светодиодом — выход, как и раньше...
  pinMode(LED_PIN, OUTPUT);

  // ...а вот пин с потенциометром должен быть входом
  // (англ. «input»): мы хотим считывать напряжение,
  // выдаваемое им
  pinMode(POT_PIN, INPUT);
}

void loop()
{
  // заявляем, что далее мы будем использовать 2 переменные с
  // именами rotation и brightness, и что хранить в них будем
  // целые числа (англ. «integer», сокращённо просто «int»)
  int rotation, brightness;

  // считываем в rotation напряжение с потенциометра:
  // микроконтроллер выдаст число от 0 до 1023
  // пропорциональное углу поворота ручки
  rotation = analogRead(POT_PIN);

  // в brightness записываем полученное ранее значение rotation
  // делённое на 4. Поскольку в переменных мы пожелали хранить
  // целые значения, дробная часть от деления будет отброшена.
  // В итоге мы получим целое число от 0 до 255
  brightness = rotation / 4;

  // выдаём результат на светодиод
  analogWrite(LED_PIN, brightness);
}
```

Задание на лабораторную работу

Отключите питание платы, подключите к порту 5 еще один светодиод. Измените код таким образом, чтобы второй светодиод светился на 1/8 от яркости первого

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. Можем ли мы при сборке схемы подключить светодиод и потенциометр напрямую к разным входам GND микроконтроллера?
2. В какую сторону нужно крутить переменный резистор для увеличения яркости светодиода?
3. Что будет, если стереть из программы строчку `pinMode(LED_PIN, OUTPUT)`? строчку `pinMode(POT_PIN, INPUT)`?
4. Зачем мы делим значение, полученное с аналогового входа перед тем, как задать яркость светодиода? что будет, если этого не сделать?

Лабораторная работа №4

Управление вводом/выводом

Цель работы: ознакомиться с принципами ввода/вывода.

Краткие сведения из теории

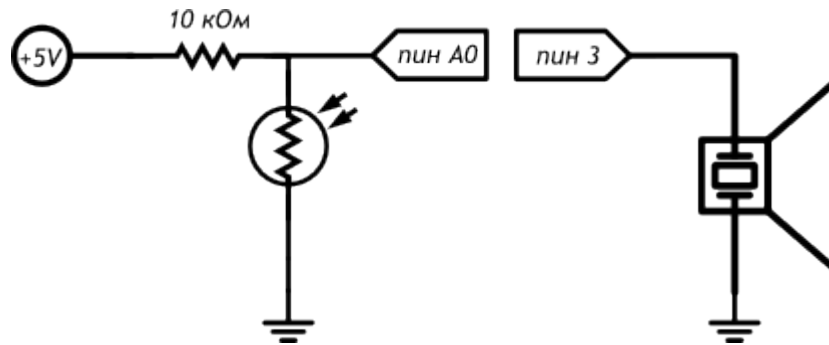


Рисунок 4.1 – Принципиальная схема

Листинг программы:

```
// даём имена для пинов с пьезопищалкой (англ. buzzer) и фото-
// резистором (англ. Light Dependent Resistor или просто LDR)
#define BUZZER_PIN 3
#define LDR_PIN A0

void setup()
{
  // пин с пьезопищалкой — выход...
  pinMode(BUZZER_PIN, OUTPUT);

  // ...а все остальные пины являются входами изначально,
  // всякий раз при подаче питания или сбросе микроконтроллера.
  // Поэтому, на самом деле, нам совершенно необязательно
  // настраивать LDR_PIN в режим входа: он и так им является
}

void loop()
{
  int val, frequency;

  // считываем уровень освещённости так же, как для
  // потенциометра: в виде значения от 0 до 1023.
  val = analogRead(LDR_PIN);

  // рассчитываем частоту звучания пищалки в герцах (ноту),
  // используя функцию проекции (англ. map). Она отображает
  // значение из одного диапазона на другой, строя пропорцию.
  // В нашем случае [0; 1023] -> [3500; 4500]. Так мы получим
  // частоту от 3,5 до 4,5 кГц.
  frequency = map(val, 0, 1023, 3500, 4500);

  // заставляем пин с пищалкой «вибрировать», т.е. звучать
  // (англ. tone) на заданной частоте 20 миллисекунд. При
  // следующих проходах loop, tone будет вызван снова и снова,
  // и на деле мы услышим непрерывный звук тональностью, которая
  // зависит от количества света, попадающего на фоторезистор
  tone(BUZZER_PIN, frequency, 20);
}
```

Задание на лабораторную работу

Уберите из программы чтение датчика освещенности и пропишите азбукой Морзе позывной SOS: три точки, три тире, три точки

Измените код программы так, чтобы с падением освещенности звук становился ниже (например, падал от 5 кГц до 2,5 кГц)

Измените код программы так, чтобы звук терменвокса раздавался не непрерывно, а 10 раз в секунду с различными паузами

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. Каким сопротивлением должен обладать фоторезистор, чтобы на аналоговый вход было подано напряжение 1 В?
2. Можем ли мы регулировать яркость светодиода, подключенного к 11-му порту, во время звучания пьезопищалки?
3. Что изменится в работе терменвокса, если заменить резистор на 10 кОм резистором на 100 кОм? Попробуйте ответить без эксперимента. Затем отключите питание, замените резистор и проверьте.
4. Каков будет результат вызова `map(30,0,90,90,-90)`?
5. Как будет работать вызов `tone` без указания длительности звучания?
6. Можно ли устроить полифоническое звучание с помощью функции `tone`?

Лабораторная работа №5

Управление выводом по событию на вводе

Цель работы: ознакомиться с принципами ввода/вывода.

Краткие сведения из теории

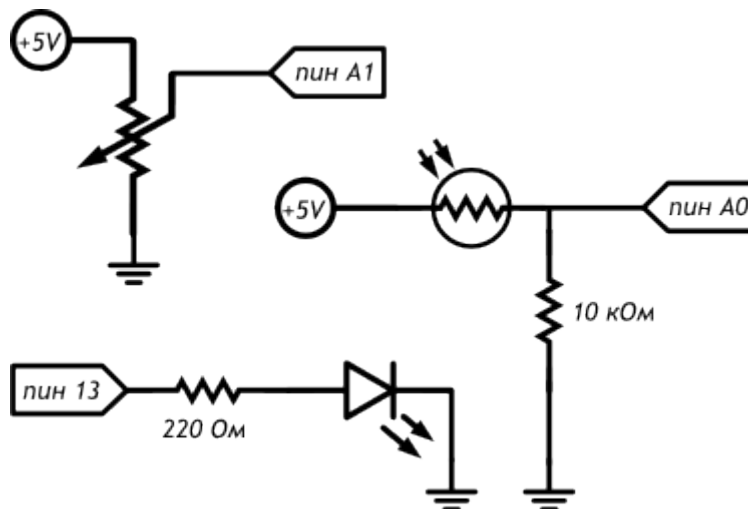


Рисунок 5.1 – Принципиальная схема

Листинг программы:

```
#define LED_PIN 13
#define LDR_PIN A0
#define POT_PIN A1
void setup()
{
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  // считываем уровень освещённости. Кстати, объявлять
  // переменную и присваивать ей значение можно разом
  int lightness = analogRead(LDR_PIN);
  // считываем значение с потенциометра, которым мы регулируем
  // пороговое значение между условными темнотой и светом
  int threshold = analogRead(POT_PIN);
  // объявляем логическую переменную и назначаем ей значение
  // «темно ли сейчас». Логические переменные, в отличие от
  // целочисленных, могут содержать лишь одно из двух значений:
  // истину (англ. true) или ложь (англ. false). Такие значения
  // ещё называют булевыми (англ. boolean).
  boolean tooDark = (lightness < threshold);
  // используем ветвление программы: процессор исполнит один из
  // двух блоков кода в зависимости от исполнения условия.
  // Если (англ. «if») слишком темно...
  if (tooDark) {
    // ...включаем освещение
    digitalWrite(LED_PIN, HIGH);
  } else {
    // ...иначе свет не нужен — выключаем его
    digitalWrite(LED_PIN, LOW);
  }
}
```

Задание на лабораторную работу

Перепишите программу без использования переменной `tooDark` с сохранением функционала устройства.

Добавьте в схему еще один светодиод. Дополните программу так, чтобы при падении освещенности ниже порогового значения включался один светодиод, а при падении освещенности ниже половины от порогового значения включались оба светодиода.

Измените схему и программу так, чтобы светодиоды включались по прежнему принципу, но светились тем сильнее, чем меньше света падает на фоторезистор.

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. Если мы установим фоторезистор между аналоговым входом и землей, наше устройство будет работать наоборот: светодиод будет включаться при увеличении количества света. Почему?
2. Какой результат работы устройства мы получим, если свет от светодиода будет падать на фоторезистор?
3. Если мы все же установили фоторезистор так, как сказано в предыдущем вопросе, как нам нужно изменить программу, чтобы устройство работало верно?
4. Допустим, у нас есть код `if (условие) {действие;}`. В каких случаях будет выполнено действие?
5. При каких значениях у выражение $x + y > 0$ будет истинным, если $x > 0$?
6. Обязательно ли указывать, какие инструкции выполнять, если условие в операторе `if` ложно?
7. Чем отличается оператор `==` от оператора `=`?
8. Если мы используем конструкцию `if(условие) действие1; else действие2;`, может ли быть ситуация, когда ни одно из действий не выполнится? Почему?

Лабораторная работа №6

Управление таймером с выводом на семисегментный индикатор

Цель работы: ознакомиться с принципами вывода.

Краткие сведения из теории

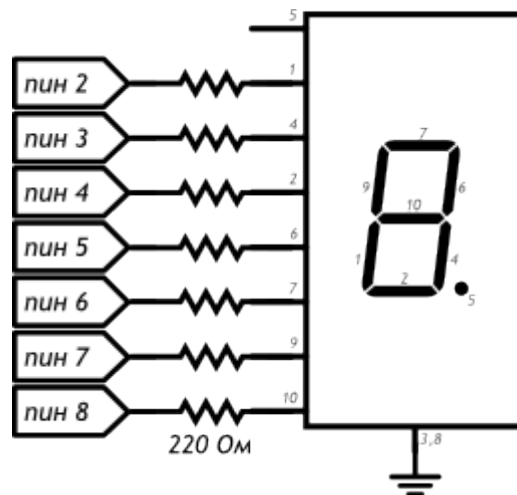


Рисунок 6.1 – Принципиальная схема

Листинг программы:

```
#define FIRST_SEGMENT_PIN 2
#define SEGMENT_COUNT 7
// префикс «0b» означает, что целое число за ним записано в
// в двоичном коде. Единицами мы обозначим номера сегментов
// индикатора, которые должны быть включены для отображения
// арабской цифры. Всего цифр 10, поэтому в массиве 10 чисел.
// Нам достаточно всего байта (англ. byte, 8 бит) для хранения
// комбинации сегментов для каждой из цифр.
byte numberSegments[10] = {
  0b00111111, 0b00001010, 0b01011101, 0b01011110, 0b01101010,
  0b01110110, 0b01110111, 0b00011010, 0b01111111, 0b01111110,
};
void setup()
{
  for (int i = 0; i < SEGMENT_COUNT; ++i)
    pinMode(i + FIRST_SEGMENT_PIN, OUTPUT);
}
void loop()
{
  // определяем число, которое собираемся отображать. Пусть им
  // будет номер текущей секунды, зацикленный на десятке
  int number = (millis() / 1000) % 10;
  // получаем код, в котором зашифрована арабская цифра
  int mask = numberSegments[number];
  // для каждого из 7 сегментов индикатора...
  for (int i = 0; i < SEGMENT_COUNT; ++i) {
    // ...определяем: должен ли он быть включён. Для этого
    // считываем бит (англ. read bit), соответствующий текущему
    // сегменту «i». Истина — он установлен (1), ложь — нет (0)
    boolean enableSegment = bitRead(mask, i);
    // включаем/выключаем сегмент на основе полученного значения
    digitalWrite(i + FIRST_SEGMENT_PIN, enableSegment);
  }
}
```

Задание на лабораторную работу

Измените код, чтобы индикатор отсчитывал десятые секунды.

Поменяйте программу так, чтобы вместо символа «0» отображался символ «A».

Дополните схему и программу таким образом, чтобы сегмент-точка включался при прохождении четных чисел и выключался на нечетных

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. К какой ножке нашего семисегментного индикатора нужно подключать землю?
2. Как мы храним закодированные символы цифр?
3. Каким образом мы выводим символ на индикатор?

Лабораторная работа №7

Управление аналоговым выводом с помощью цифрового ввода

Цель работы: ознакомиться с принципами ввода и вывода.

Краткие сведения из теории

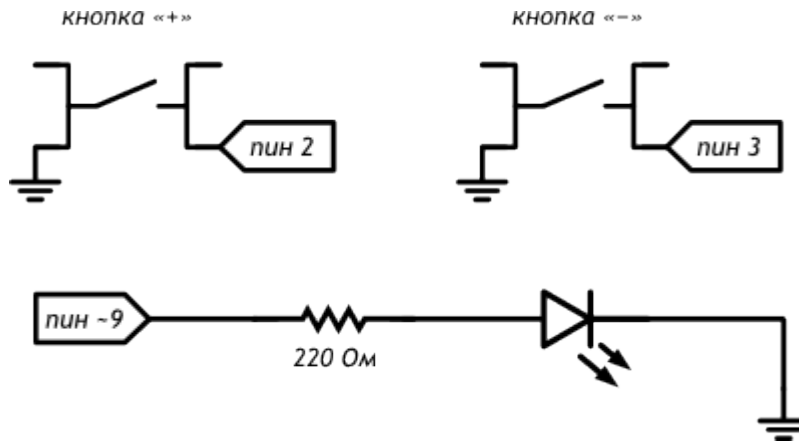


Рисунок 7.1 – Принципиальная схема

Листинг программы:

```
#define PLUS_BUTTON_PIN 2
#define MINUS_BUTTON_PIN 3
#define LED_PIN 9
int brightness = 100;
boolean plusUp = true;
boolean minusUp = true;
void setup()
{
  pinMode(LED_PIN, OUTPUT);
  pinMode(PLUS_BUTTON_PIN, INPUT_PULLUP);
  pinMode(MINUS_BUTTON_PIN, INPUT_PULLUP);
}
void loop()
{
  analogWrite(LED_PIN, brightness);
  // реагируем на нажатия с помощью функции, написанной нами
  plusUp = handleClick(PLUS_BUTTON_PIN, plusUp, +35);
  minusUp = handleClick(MINUS_BUTTON_PIN, minusUp, -35);
}
// Собственная функция с 3 параметрами: номером пина с кнопкой
// (buttonPin), состоянием до проверки (wasUp) и градацией
// яркости при клике на кнопку (delta). Функция возвращает
// (англ. return) обратно новое, текущее состояние кнопки
boolean handleClick(int buttonPin, boolean wasUp, int delta)
{
  boolean isUp = digitalRead(buttonPin);
  if (wasUp && !isUp) {
    delay(10);
    isUp = digitalRead(buttonPin);
    // если был клик, меняем яркость в пределах от 0 до 255
    if (!isUp)
      brightness = constrain(brightness + delta, 0, 255);
  }
  return isUp; // возвращаем значение обратно, в вызывающий код
}
```

Задание на лабораторную работу

Доработайте код таким образом, чтобы шаг изменения яркости настраивался в одном месте.

Создайте еще одну функцию и переделайте код так, чтобы одна функция отвечала за отслеживание нажатий, а другая — за вычисление яркости светодиода и возвращала его в `analogWrite`.

Содержание отчета

В отчете указать цель работы, привести разработанные схему и листинг программы.

Контрольные вопросы

1. Что необходимо для определения собственной функции?
2. Что означает ключевое слово `void`?
3. Как ведет себя программа при упоминании одной переменной с разных сторон от оператора присваивания `=`?