



**Некоммерческое
акционерное
общество**

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ И СВЯЗИ**

Кафедра «Электроника
и робототехника»

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И КОНТРОЛЯ

Методические указания по выполнению лабораторных работ
для студентов специальности 5В071600 – Приборостроение

Алматы 2019

СОСТАВИТЕЛИ: Байкенов Б.С., Аязбай А.Е. Микропроцессорные системы управления и контроля. Методические указания по выполнению лабораторных работ для студентов специальности 5В071600 – «Приборостроение». – Алматы: АУЭС, 2019. – 18 с.

Методические указания посвящены изучению микроконтроллерных плат Arduino, являющихся основным элементом для создания цифровых приборов и систем автоматического управления и контроля.

Рассмотрены установка драйверов и программного обеспечения Arduino, пакетов и приложений, описание необходимых инструментов и комплектующих для проведения лабораторных работ.

Изложение материала сопровождается примерами по разработке различных устройств, приборов и систем. Для каждого проекта приведен перечень необходимых компонентов, монтажная схема и листинг программ на языке С.

Методические указания составлены в целях закрепления лекционного материала и предназначены для студентов специальности 5В071600 – «Приборостроение».

Ил. - 7, библиогр. – 5.

Рецензент: доцент Курпенев Б.К.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2019 г.

Введение

Методические указания посвящены моделированию микропроцессорных приборов и систем в программной среде Proteus на базе микроконтроллерных плат Arduino, а также изучению программирования микроконтроллеров в среде Arduino IDE и использования микроконтроллеров для связи с внешними системами в проектах автоматизации и робототехники. Описаны технические возможности, особенности подключения и взаимодействия различных датчиков и исполнительных устройств.

Методические указания составлены для приобретения практических навыков и компетенций по проектированию и созданию реальных цифровых устройств и систем.

1 Лабораторная работа №1. Схема управления светофором в среде Proteus

Цель работы: приобрести навыки сборки схем в программе Proteus, а также научиться разрабатывать коды для микроконтроллера Arduino по управлению огнями светофора и опроса состояния кнопки.

1.1 Краткие сведения

Arduino - микроконтроллер семейства Atmega, состоит из микропроцессора с памятью и различных периферийных устройств, реализован на одной микросхеме.

Arduino имеет цифровые и аналоговые выходы. На цифровых выходах может быть только два значения: логическая "1" (TRUE, от 3 до 5 вольт) или логический "0" (FALSE, от 0 до 1,5 вольт), а на аналоговых выходах – непрерывный сигнал от 0 до 5В.

Светодиод – электронный прибор, создающий [оптическое излучение](#) при пропускании через него электрического тока в прямом направлении.

Каждый светодиод подключен через резистор к отдельному выводу Arduino. Красный - к выводу 2, желтый - к выводу 3, зеленый - к выводу 4. Если подать единичный сигнал на вывод 2, то загорится красный светодиод, если на 3 - желтый, если на 4 - зеленый.

Подключение питания в среде Proteus не обязательно, но при сборке схемы на реальной плате необходимо.

1.2 Порядок выполнения работы

1.2.1 Светофор.

1.2.1.1. Собрать схему управления огнями светофора в Proteus (рисунок 1.1).

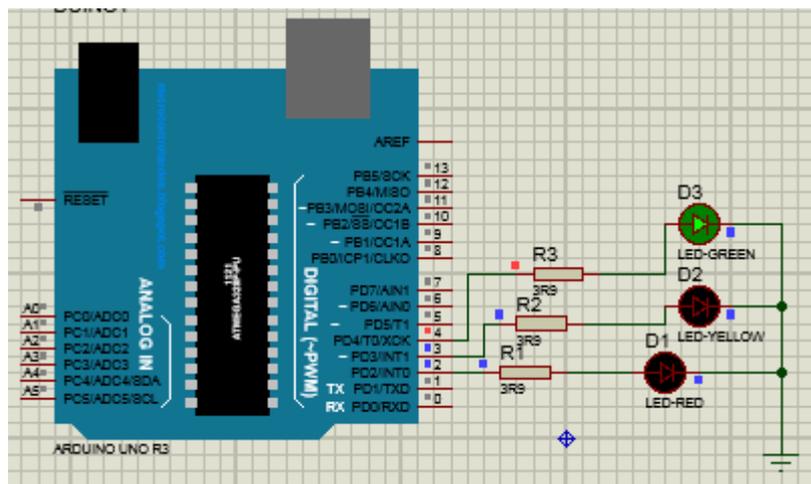


Рисунок 1.1 – Схема управления огнями светофора в Proteus

1.2.1.2 Листинг кода программы.

```
int led_red = 2; // пин подключения 2
int led_yellow = 3; // пин подключения 3
int led_green = 4; // пин подключения 4
void setup() { // конфигурация выводов на выход
  pinMode(led_red, OUTPUT);
  pinMode(led_yellow, OUTPUT);
  pinMode(led_green, OUTPUT);
}
void loop() {
  Led (led_red, HIGH); // красный светодиод загорается
  delay(5000); // задержка 5 секунд
  Led (led_red, LOW); // красный светодиод гаснет
  Led (led_yellow, HIGH); // желтый светодиод загорается
  delay(5000); // задержка 5 секунд
  Led (led_yellow, LOW); // желтый светодиод гаснет
  Led (led_green, HIGH); // зеленый светодиод загорается
  delay(5000); // задержка 5 секунд
  Led (led_green, LOW); // зеленый светодиод гаснет
}
void Led (uint8_t pin, byte status) // замена функции digitalWrite на Led
{
  digitalWrite(pin, status);
}
```

1.2.2 Кнопка.

Подключение кнопки к Arduino: один свободный проводник кнопки соединить с питанием или землёй, другой – с цифровым выводом Arduino. Но, в моменты, когда кнопка не замкнута, на цифровом выводе Ардуино будут появляться электромагнитные наводки, и из-за этого возможны ложные срабатывания.

Для избежания наводок цифровой вывод обычно подключают через резистор 10 кОм к земле или к питанию. В первом случае - схема с подтягивающим резистором, во втором – схема со стягивающим резистором.

В среде Proteus не требуется подключения подтягивающего или стягивающего резистора.

Принцип работы схемы: при нажатии на кнопку светодиод гаснет, при отпускании кнопки – горит.

1.2.2.1 Собрать схему подключения кнопки в Proteus (рисунок 1.2).

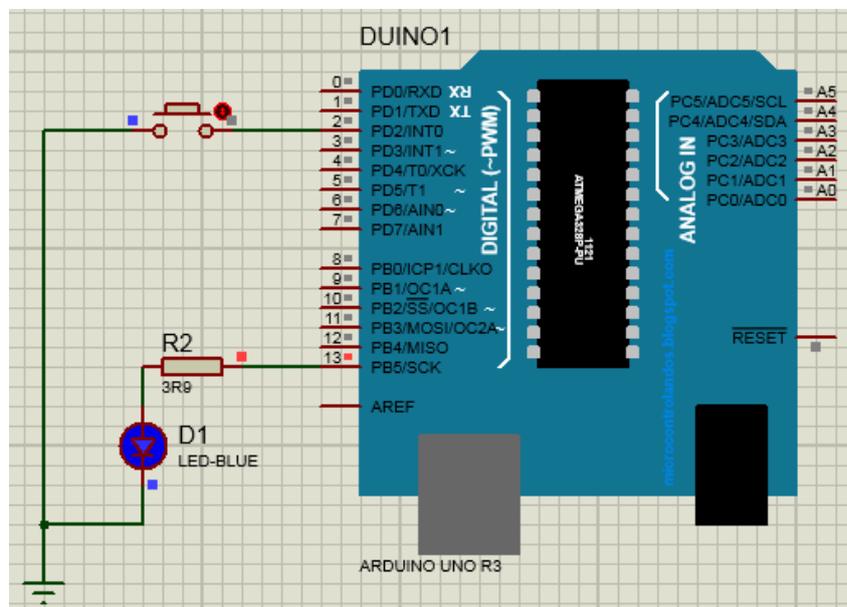


Рисунок 1.2 – Схема подключения кнопки в Proteus

1.2.2.2 Листинг кода опроса состояния кнопки.

```
const int buttonPin = 2; // пин подключения кнопки 2
const int ledPin = 13; // пин подключения светодиода 13
// переменные будут меняться
int buttonState = 0; // переменная для чтения статуса кнопки
void setup() {
    // конфигурация вывода 13 на выход
    pinMode(ledPin, OUTPUT);
    // конфигурация вывода 2 на вход
    pinMode(buttonPin, INPUT);
}
```

```
void loop() {
  // чтение состояния значения кнопки:
  buttonState = digitalRead(buttonPin);
  // кнопка нажата - buttonState is HIGH:
  if (buttonState == HIGH) {
    // включить светодиод:
    digitalWrite(ledPin, HIGH);
  // нет:
  } else {
    // выключить светодиод:
    digitalWrite(ledPin, LOW);
  }
}
```

1.3 Содержание отчета

- 1.3.1 Цель работы.
- 1.3.2 Скриншоты схем светофора и подключения кнопки.
- 1.3.3 Листинги кодов.
- 1.3.4 Реальная схема на Ардуино и светодиодами (фото).
- 1.3.5 Выводы.

1.4 Контрольные вопросы

- 1.4.1 Как выглядит структура скетча Ардуино?
- 1.4.2 Зачем объявляются типы переменных и данных?
- 1.4.3 Как осуществляется конфигурация портов ввода-вывода?
- 1.4.4 Как изменить длительность горения ламп светофора?
- 1.4.5 Как изменить схему и программу, чтобы предусмотреть кнопку включения?

2 Лабораторная работа №2. Схема управления шаговым двигателем в среде Proteus

Цель работы: приобретение навыков работы в программе Proteus, создание схемы шагового двигателя, а также разработка соответствующего кода для микроконтроллера платы Arduino Uno.

2.1 Краткие сведения

Шаговый двигатель – это мотор, перемещающий свой вал в зависимости от заданных в программе микроконтроллера шагов и направления. Подобные устройства чаще всего используются в робототехнике, принтерах, манипуляторах, различных станках и прочих электронных приборах. Большим

преимуществом шаговых двигателей над двигателями постоянного вращения является обеспечение точного углового позиционирования ротора. Также в шаговых двигателях имеется возможность быстрого старта, остановки, реверса.

Драйвер – это устройство, которое связывает контроллер с шаговым двигателем.

В центральном вале имеется ряд магнитов и несколько катушек. При подаче питания от микроконтроллера через драйвер к двигателю создается магнитное поле, которое воздействует на магниты и заставляет вал вращаться.

2.2 Порядок выполнения работы

2.2.1 Собрать схему управления шаговым двигателем в Proteus (рисунок 2.1).

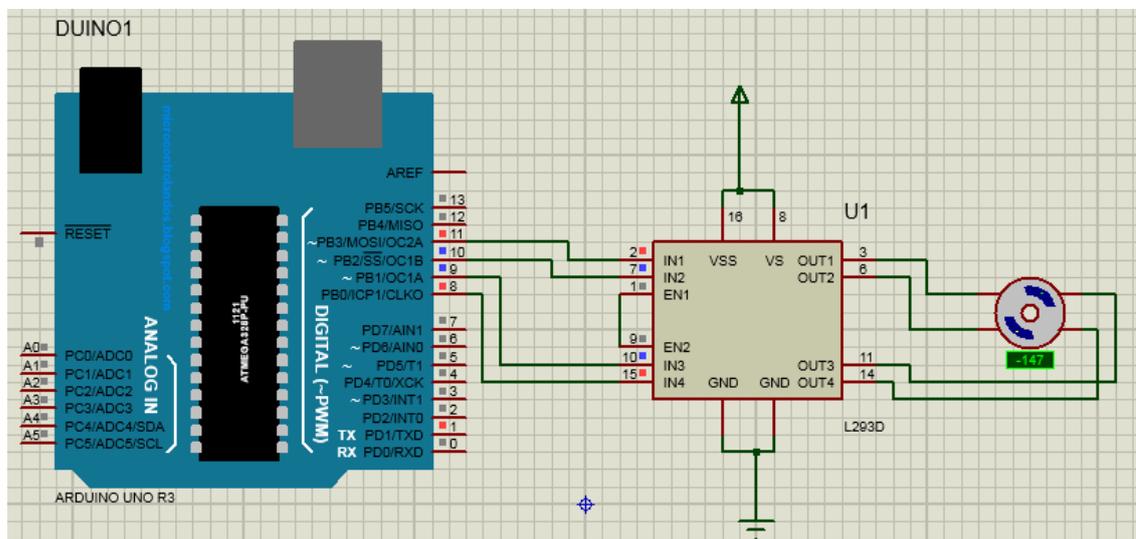


Рисунок 2.1 – Схема управления шагового двигателя в Proteus

2.2.2 Листинг кода программы.

Для управления шаговыми двигателями в Arduino IDE есть стандартная библиотека, которая осуществляет только полношаговый режим коммутации.

```
#include <Stepper.h>
const int stepsPerRevolution = 200; // количество шагов за оборот
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11); // подключение к
пинам 8-11
void setup() {
  myStepper.setSpeed(60); // установка скорости вращения ротора
  Serial.begin(9600); // инициализация последовательного порта
}
```

```
void loop() { // Функция ожидает, пока поступит команда,
преобразовывает текст и подает сигнал на двигатель для его вращения на
указанное число шагов
  Serial.println("clockwise"); // по часовой стрелке
  myStepper.step(stepsPerRevolution);
  delay(500);
  Serial.println("counterclockwise"); // против часовой стрелки
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```

Найденный hex-файл выбрать в поле Program File, расположенном в окне Properties платы Arduino Uno R3.

2.3 Содержание отчета

2.3.1 Цель работы.

2.3.2 Скриншот схемы управления в Протеус.

2.3.3 Листинг кода.

2.3.4 Реальная схема на Ардуино и ШД (фото).

2.3.5 Выводы.

2.4 Контрольные вопросы

2.4.1 Какую функцию выполняет драйвер ШД L293D?

2.4.2 Почему последовательность управления ШД называется полшаговой?

2.4.3 Почему режим работы ШД называется полношаговым?

2.4.4 Как осуществить реверс ШД?

2.4.5 Какие преимущества есть у ШД?

3 Лабораторная работа №3. Схема управления LCD в среде Proteus

Цель работы: моделирование LCD в программе Proteus и создание кода управления дисплеем для микроконтроллера Arduino Uno.

3.1 Краткие сведения

LCD - электронное устройство, предназначенное для отображения текущей информации. Применяются два типа данного устройства: графические и символьные. В зависимости от размера экрана, существуют:

- 16x2 символьный – 16 колонок и 2 строки;

- 20x4 символов – 20 колонок и 4 строки.

Выводы имеют маркировку:

- 1 (VSS) – питание на минус для контроллера;
- 2 (VDD) – питание на плюс для контроллера;
- 3 (VO) – настройки управления контрастом;
- 4 (RS) – выбор для регистра;
- 5 (R/W) – чтение и запись, в частности, запись при соединении с землей;
- 6 (E) – активация (enable);
- 7–10 (DB0-DB3) – младшие биты от восьмибитного интерфейса;
- 11–14 (DB4-DB7) – старшие биты от интерфейса;
- 15 (A) – положительный анод на питание подсветки;
- 16 (K) – отрицательный катод на питание подсветки.

3.2 Порядок выполнения работы

3.2.1 Собрать схему в среде Proteus (рисунок 3.1).

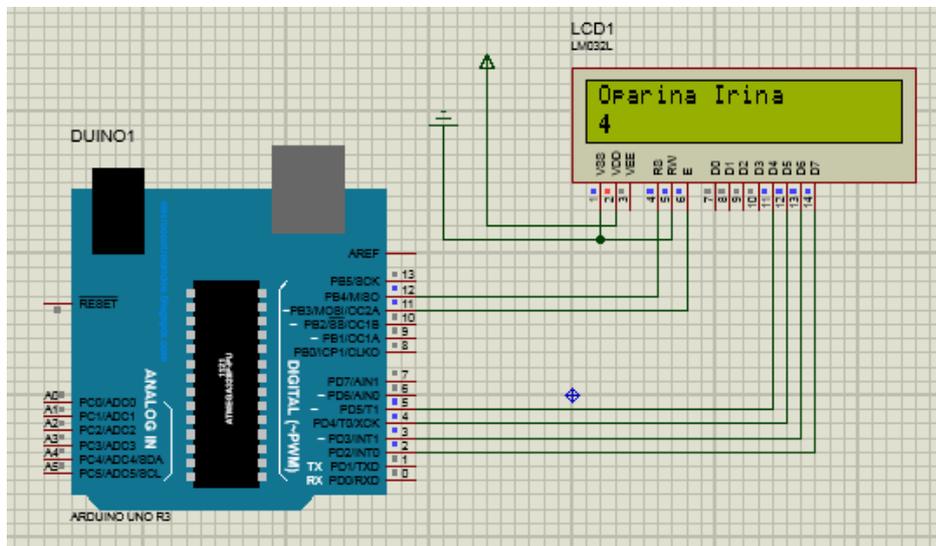


Рисунок 3.1 – Собранный схема в Proteus

3.3.2 Листинг кода программы.

```
#include <LiquidCrystal.h>
// инициализация библиотеки номерами пинов интерфейса
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  // настройка размерности LCD:
  lcd.begin(16, 2);
}
void loop() {
  // Включение дисплея:
  lcd.setCursor(0,0); //1 столбец, 1 строка
  lcd.print("Oparina Irina"); //вывод текста
```

```
lcd.setCursor(0,1); //1 столбец, 2 строка  
lcd.print(millis ()/1000); //вывод скорости  
}
```

3.2. Реальная схема управления LCD с выводами I²C приведена на рисунке

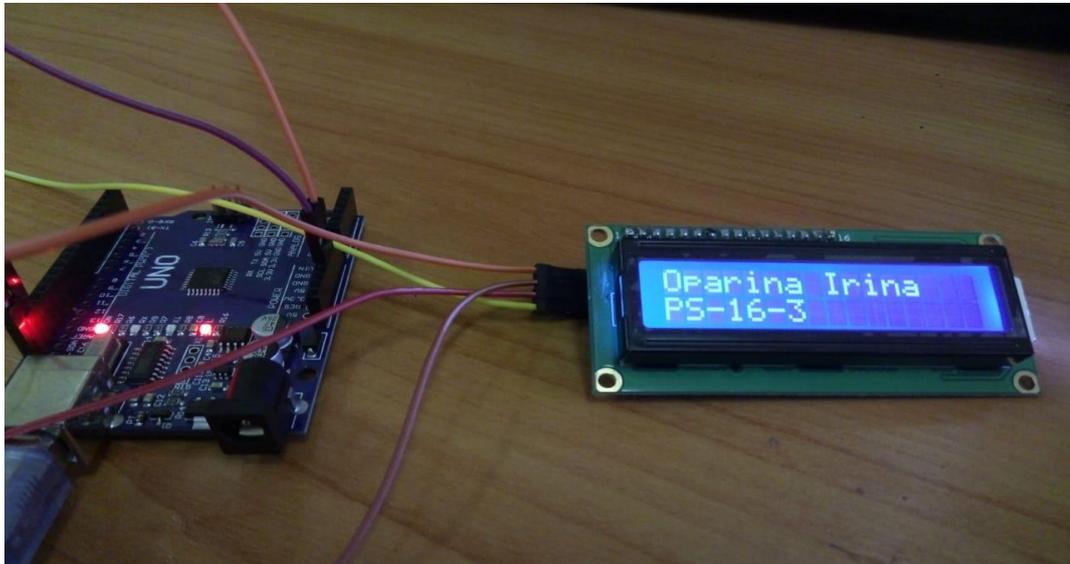


Рисунок 3.2 – Внешний вид LCD на Arduino

3.3 Содержание отчета

- 3.3.1 Цель работы.
- 3.3.2 Скриншот схемы в среде Протеус.
- 3.3.3 Листинг кода.
- 3.3.4 Реальная схема на Ардуино и LCD (фото).
- 3.3.5 Выводы.

3.4 Контрольные вопросы

- 3.4.1 Как выделяются комментарии в коде Ардуино?
- 3.4.2 Почему вывод R/W LCD заземляется?
- 3.4.3 Почему вывод V_{SS} LCD заземляется?
- 3.4.4 Каким числом заканчивается счет?
- 3.4.5 Какой тип LCD применяется в данной программе?

4 Лабораторная работа №4. Моделирование датчика температуры и влажности DHT11 в среде Proteus

Цель работы: создание схемы цифрового термометра и измерителя влажности в среде Proteus и программы в среде Arduino IDE.

4.1 Краткие сведения

DHT11 - это цифровой датчик, состоящий из термистора и емкостного датчика влажности. Питание 3,5-5V, определение температуры от 0 до 50 градусов с точностью 2 град, определение влажности от 20% до 95% с 5% точностью.

Термистор — это термический резистор, сопротивление которого изменяется с температурой, т.е. увеличение температуры приводит к падению его сопротивления.

Емкостной датчик влажности — это конденсатор с переменной емкостью и заключен в герметичный чехол, поверх которого расположен влагопоглощающий слой. При попадании частиц воды на этот слой, меняется его диэлектрическая проницаемость, что приводит к изменению емкости конденсатора.

4.2 Порядок выполнения работы

4.2.1 Собрать схему, приведенную на рисунке 4.1.

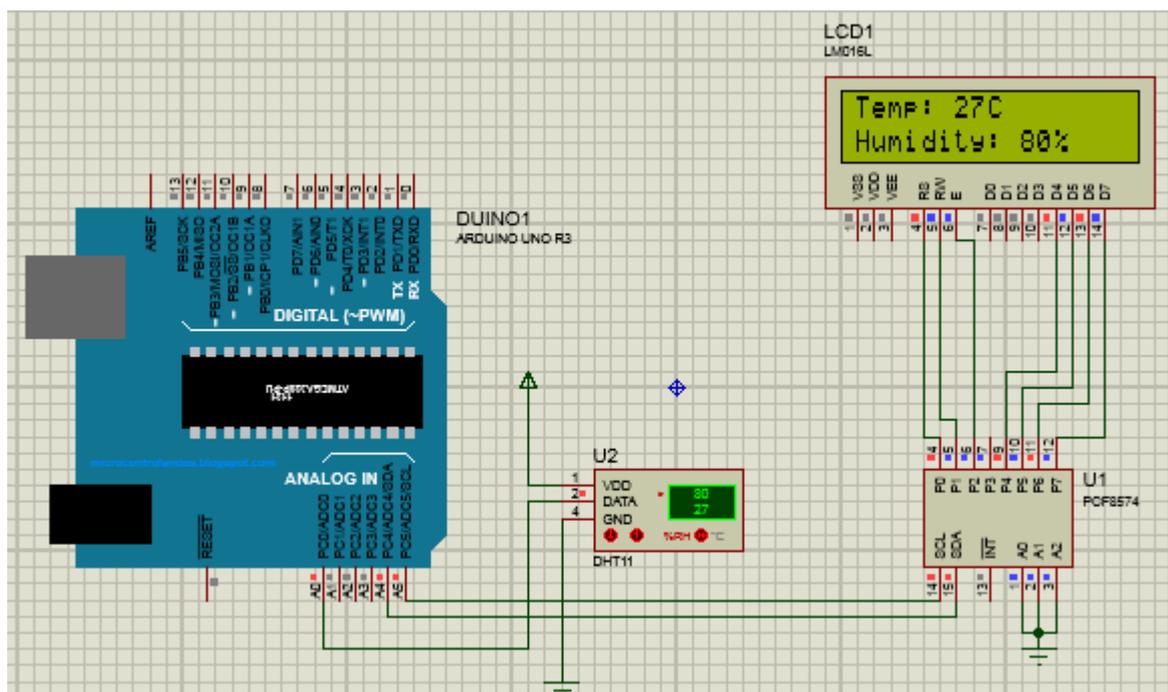


Рисунок 4.1 – Схема цифрового термометра в Proteus

4.2.2 Листинг кода программы.

```
// подключение библиотек
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
```

```

#define DHTPIN A0 // директива, которая позволяет дать имя константе
перед тем как программа будет скомпилирована
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE); // создание и обозначение объекта "dht"
LiquidCrystal_I2C lcd(0x20,16,2); // адрес и размерность lcd
void setup()
{
  lcd.begin(16, 2); // на дисплее будет 16 столбцов и 2 строчки
  dht.begin();      // подготовка датчика DHT11 к работе
}
void loop() {
  int h = dht.readHumidity();
  int t = dht.readTemperature(); // ввод переменных "h" и "t" , в которые
будет считываться значение с датчика DHT11
  lcd.setCursor(0,0); // 1 столбец, 1 строка
  lcd.print("Temp: "); // вывод текста
  lcd.print(t); //
  lcd.print("C");

  lcd.setCursor(0,1); // 1 столбец, 2 строка
  lcd.print("Humidity: "); // вывод текста
  lcd.print(h);
  lcd.print("%");
  delay (200); // ожидание
}

```

4.3 Содержание отчета

- 4.3.1 Цель работы.
- 4.3.2 Скриншот схемы в среде Протеус.
- 4.3.3 Листинг кода.
- 4.3.4 Реальная схема на Ардуино (фото).
- 4.3.5 Выводы.

4.4 Контрольные вопросы

- 4.4.1 Как работает термистр?
- 4.4.2 Как измеряется влажность среды?
- 4.4.3 Каким числом заканчивается измерение влажности?
- 4.4.4 Каким числом заканчивается измерение температуры?
- 4.4.5 Какой тип LCD применяется в данной программе?

5 Лабораторная работа №5. Моделирование цифрового вольтметра в среде Proteus

Цель работы: изучить принцип действия и особенности проектирования цифрового вольтметра в среде Proteus.

5.1 Краткие сведения

Вольтметр - прибор для измерения напряжения или ЭДС в электрических цепях. Подключается параллельно нагрузке или источнику электрической энергии. Идеальный вольтметр должен обладать бесконечно большим внутренним сопротивлением. Поэтому, чем выше внутреннее сопротивление в реальном вольтметре, тем меньше влияния оказывает прибор на измеряемый объект и, следовательно, тем выше точность.

5.2 Порядок выполнения работы

5.2.1 Собрать схему модели цифрового вольтметра в Proteus, приведенную на рисунке 5.1.

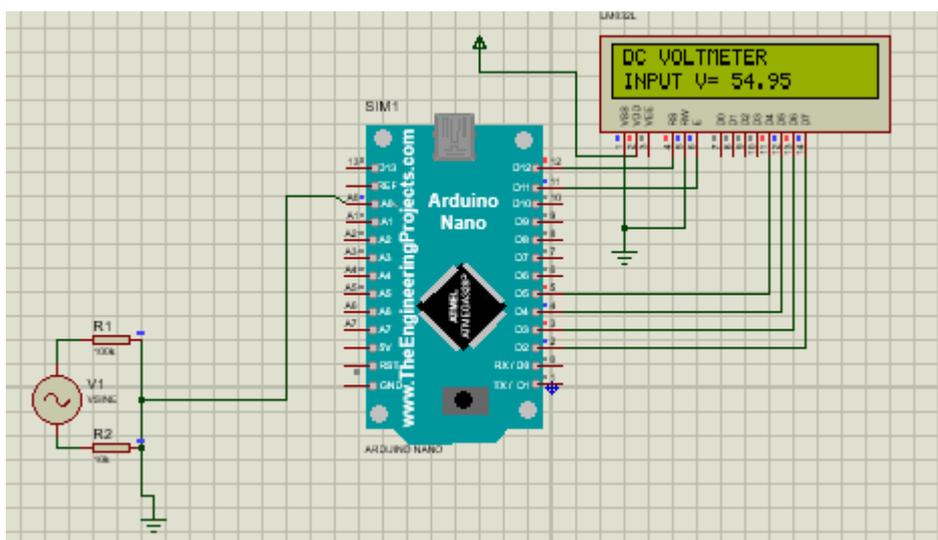


Рисунок 5.1 – Схема модели вольтметра в Proteus

При изменении значения напряжения на выходе источника напряжения V1 (VSINE), на экране LCD отобразится измеренное значение напряжения. Входом вольтметра является верхний конец R1 и земля, т.е. 2 проводника к V1 являются щупом вольтметра.

5.2.2 Листинг кода программы.

```
#include <LiquidCrystal.h>
```

```

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int analogInput = 0;
float vout = 0.0;
float vin = 0.0;
float R1 = 100000.0; // сопротивление R1 (100К)
float R2 = 10000.0; // сопротивление R2 (10К)
int value = 0;
void setup(){
  pinMode(analogInput, INPUT);
  lcd.begin(16, 2);
  lcd.print("DC VOLTMETER");
}
void loop(){
  // считывание аналогового значения
  value = analogRead(analogInput);
  vout = (value * 5.0) / 1024.0;
  vin = vout / (R2/(R1+R2));
  if (vin<0.09) {
    vin=0.0;// обнуляем нежелательное значение
  }
  lcd.setCursor(0, 1);
  lcd.print("INPUT V= ");
  lcd.print(vin);
  delay(500);
}

```

5.3 Содержание отчета

- 5.3.1 Цель работы.
- 5.3.2 Скриншот схемы в среде Протеус.
- 5.3.3 Листинг кода.
- 5.3.4 Реальная схема на Ардуино (фото).
- 5.3.5 Выводы.

5.4 Контрольные вопросы

- 5.4.1 Как работает АЦП?
- 5.4.2 От чего зависит точность измерения цифрового вольтметра?
- 5.4.3 Какую функцию в схеме выполняют резисторы R1 и R2?
- 5.4.4 С какого значения напряжения начинается измерение?
- 5.4.5 Какой тип LCD применяется в данной программе?

6 Лабораторная работа №6. Моделирование Bluetooth модуля в среде Proteus

Цель работы: создание схемы с использованием платы Arduino UNO в среде Proteus и программы управления Bluetooth модулем в программной среде Arduino IDE.

6.1 Краткие сведения

Главными достоинствами Bluetooth можно назвать хорошую устойчивость к широкополосным помехам и простоту реализации. Bluetooth протокол необходим для быстрой передачи данных на небольших расстояниях. Он будет удобен для управления объектами со смартфона, если загрузить его соответствующим приложением.

6.2 Порядок выполнения работы

6.2.1 Собрать схему управления светодиодом по каналу Bluetooth и скачать приложение для управления светодиодом через Bluetooth модуль смартфона (рисунок 6.1).

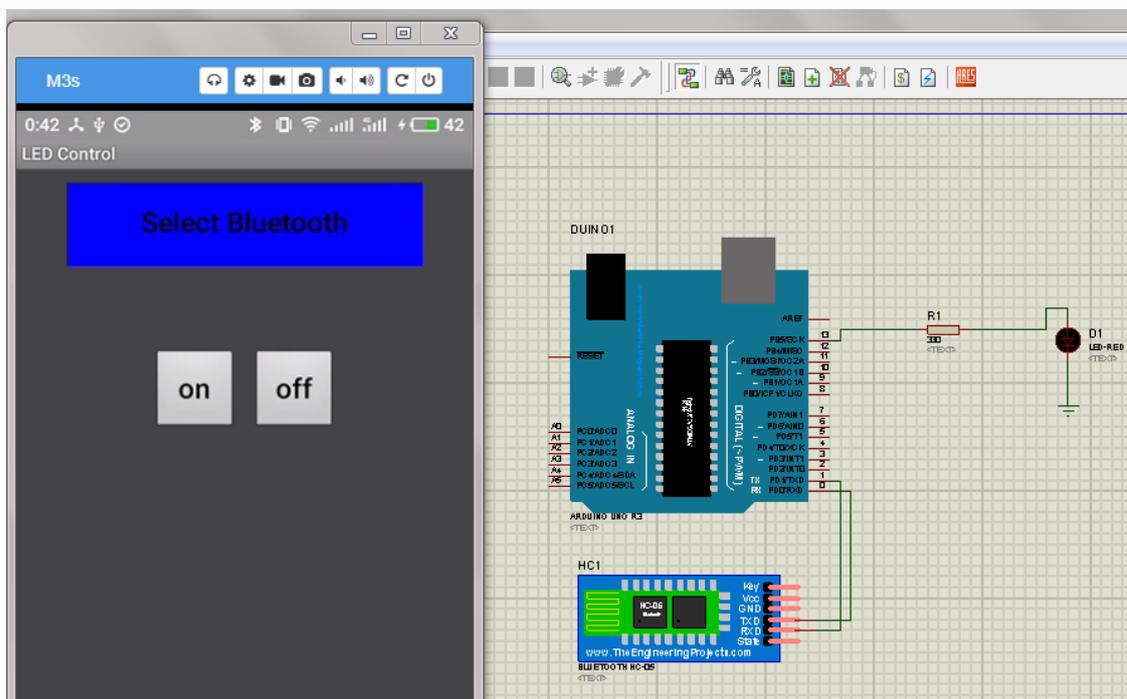


Рисунок 6.1 – Схема управления светодиодом через модуль Bluetooth

6.2.2 Листинг кода программы.

```
String voice;  
void setup() {
```

```

pinMode(13, OUTPUT); // Вывод 13 - выход
Serial.begin(9600); // Установка скорости обмена данными
}
void loop() {
  while (Serial.available()) //пока есть принятые данные
  {
    delay(10);
    char c = Serial.read(); // принятие данных и сохранение в переменной
    voice += c;
  }
  if (voice.length() > 0)
  {
    Serial.println(voice); //получен сигнал
    if(voice == "on")
    {
      digitalWrite(13, HIGH); // включить светодиод
    }
    if(voice == "off")
    {
      digitalWrite(13, LOW); // выключить светодиод
    }
    voice="";
  }
}

```

6.3 Содержание отчета

- 6.3.1 Цель работы.
- 6.3.2 Скриншот схемы в среде Протеус.
- 6.3.3 Листинг кода.
- 6.3.4 Реальная схема на Ардуино (фото).
- 6.3.5 Выводы.

6.4 Контрольные вопросы

- 6.4.1 Как работает АЦП?
- 6.4.2 Какое максимальное расстояние включения светодиода?
- 6.4.3 Какую функцию в схеме выполняют резистор R1?
- 6.4.4 Влияют ли внешние препятствия и угол направления смартфона на срабатывание светодиода?
- 6.4.5 Какой протокол передачи применяется между модулем Bluetooth и платой Ардуино?

Список литературы

- 1 Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. - СПб.: БХВ-Петербург, 2016. - 544 с.
- 2 Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. – СПб.: БХВ-Петербург, 2016. – 464 с.
- 3 Петин В.А. – Микрокомпьютеры Raspberry Pi. Практическое руководство. - СПб.: БХВ-Петербург, 2015. – 240 с.
- 4 Петин В.А. Проекты с использованием контроллера Arduino – СПб.: БХВ-Петербург, 2016. – 464 с.
- 5 Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Дженльменский набор Web-мастера. – СПб.: БХВ-Петербург, 2019. – 912 с.

Содержание

Введение.....	3
1 Лабораторная работа №1.....	3
2 Лабораторная работа №2.....	6
3 Лабораторная работа №3.....	8
4 Лабораторная работа №4.....	10
5 Лабораторная работа №5.....	13
6 Лабораторная работа №6.....	15
Список литературы.....	17

Байкенов Бахытжан Сергеевич
Аязбай Абу-Алим Ерикулы

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И КОНТРОЛЯ

Методические указания по выполнению лабораторных работ
для студентов специальности 5В071600 – Приборостроение

Редактор Л.Т. Сластихина

Специалист по стандартизации Г. Мухаметсариева

Подписано в печать 16.09.19

Тираж 20 экз.

Объем 1,1 уч.-изд. л.

Формат 60x84 1/16

Бумага типографическая № 1

Заказ_____Цена 600 тг.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, ул. Байтурсынова, 126/1