Non-commercial joint-stock company

ALMATY UNIVERSITY POWER AND COMMUNICATIONS

Department of "Electronics and Robotics"

MICROPROCESSOR CONTROL AND MONITORING SYSTEMS

Guidelines for the implementation of laboratory works
for students majoring 5B071600 – Instrument making

Almaty 2019

COMPONENTS: Baikenov B.S., Ayazbay A.E. Microprocessor control and monitoring systems. Guidelines for the implementation of laboratory works for students of the specialty 5B071600 - "Instrument Engineering". - Almaty: AUPET, 2019. - 18 p.

Guidelines are devoted to the study of Arduino microcontroller boards, which are the main element for creating digital devices and automatic control systems and control systems.

Considered installing drivers and software Arduino, packages and applications, a description of the necessary tools and components for laboratory work.

The presentation of the material is accompanied by examples on the development of various devices, instruments and systems. For each project, a list of necessary components, wiring diagram and program listing in C language are given.

Methodical instructions are drawn up in order to consolidate the lecture material and are intended to students of the specialty 5B071600 - “Instrument Engineering”.

Il - 7, bibliogr – 5.

Reviewer: Senior Lecturer Kim E.S.

Reprinted according to the plan of publishing a non-commercial joint-stock company “Almaty University of Power and Communications” for 2019.

Introduction

The guidelines are devoted to modeling microprocessor devices and systems in the Proteus software environment based on Arduino microcontroller boards, as well as studying the programming of microcontrollers in the Arduino IDE environment and the use of microcontrollers to communicate with external systems in automation and robotics projects. Technical capabilities, features of connection and interaction of various sensors and actuators are described.

Guidelines are compiled to acquire practical skills and competencies in the design and creation of real digital devices and systems.

1 Laboratory work №1. Traffic light control scheme in Proteus

Purpose of work: to acquire the skills of assembling circuits in the Proteus program, as well as learn how to develop codes for the Arduino microcontroller for controlling traffic lights and polling the state of a button.

1.1 Summary of the theory

Arduino is an Atmega microcontroller, it consists of a microprocessor with memory and various peripheral devices, implemented on a single chip.

Arduino has digital and analog outputs. On digital outputs there can be only two values: logical "1" (TRUE, from 3 to 5 volts) or logical "0" (FALSE, from 0 to 1.5 volts), and on analog outputs - a continuous signal from 0 to 5V .

LED is an electronic device that generates optical radiation when electric current is passed through it in the forward direction.

Each LED is connected through a resistor to a separate Arduino pin. Red - to pin 2, yellow - to pin 3, green - to pin 4. If you submit a single signal to pin 2, then the red LED will light up, if to 3 - yellow, if to 4 - green.

Power connection in the Proteus environment is not necessary, but it is necessary when assembling the circuit on a real board.

1.2 The order of the work

1.2.1 Traffic light.

1.2.1.1 Assemble the traffic light control circuit in Proteus (figure 1.1).

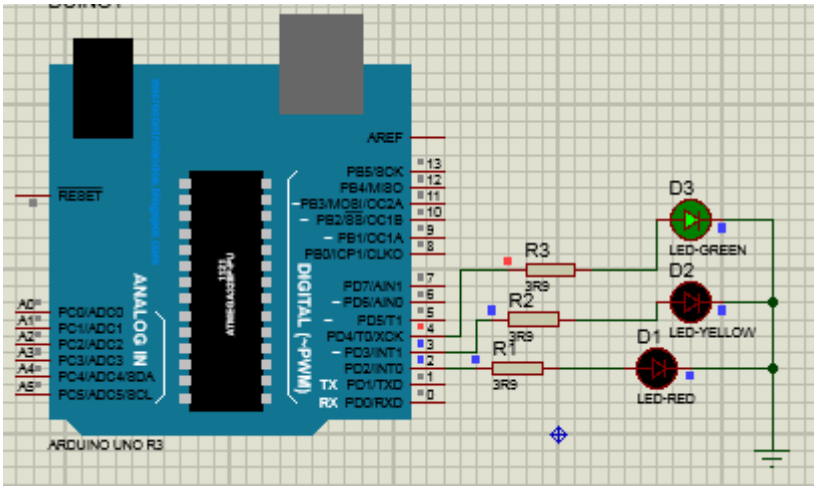

Figure 1.1 - Traffic light control diagram in Proteus

1.2.1.2 Listing of program code.

```
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 Button.

Connecting a button to an Arduino: connect one free button conductor to power or ground, and the other to an Arduino digital output. But, at moments when the button is not closed, electromagnetic pickups will appear on the Arduino digital

output, and because of this, false alarms are possible.

To avoid interference, the digital output is usually connected through a sufficiently large resistor (10 kOhm) to either ground or power. The Proteus environment can be called ideal, so it does not require a resistor.

The principle of operation of the circuit: when the button is pressed, the LED goes out, when the button is released, it lights up.

1.2.2.1 Assemble the button connection diagram in Proteus (figure 1.2).

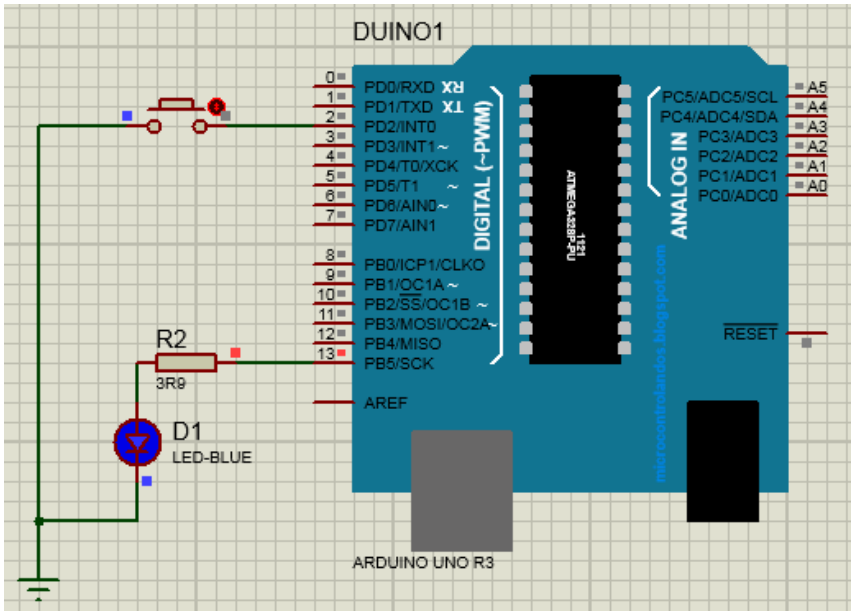

Figure 1.2 - Button connection diagram in Proteus

1.2.2.2 Listing button status polling code.

```
const int buttonPin = 2;
const int ledPin =  13;
int buttonState = 0;
void setup() {
pinMode(ledPin, OUTPUT);
pinMode(buttonPin, INPUT);
}
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
  digitalWrite(ledPin, HIGH);
// no
  } else {
digitalWrite(ledPin, LOW);
  }
}
```

1.3 Report Content

1.3.1 Purpose of the work.
1.3.2 Screenshots of traffic patterns and button connections.
1.3.3 Code listings.
1.3.4 Real scheme on Arduino and LEDs (photo).
1.3.5 Conclusions.

1.4 Test questions

1.4.1 What does the structure of the Arduino sketch look like?
1.4.2 Why are variable and data types declared?
1.4.3 How are I / O ports configured?
1.4.4 How to change the burning time of traffic lights?
1.4.5 How to change the scheme and program to provide a power button?

2 Laboratory work №2. Proteus Stepper Motor Control Diagram

Purpose of work: acquiring skills in the Proteus program, creating a stepper motor circuit, as well as developing the appropriate code for the microcontroller of the Arduino Uno board.

2.1 Summary of the theory

A stepper motor is a motor that moves its shaft depending on the steps and directions specified in the microcontroller program. Such devices are most often used in robotics, printers, manipulators, various machines and other electronic devices. The big advantage of stepper motors over DC motors is the provision of precise angular positioning of the rotor. Also in stepper motors there is the possibility of a quick start, stop, reverse.

A driver is a device that connects a controller and a stepper motor.

In the central shaft there are a number of magnets and several coils. When power is supplied from the microcontroller through the driver to the engine, a magnetic field is created that acts on the magnets and causes the shaft to rotate.

2.2 The order of the work

2.2.1 Assemble the control circuit of the stepper biconductor in Proteus (figure 2.1).

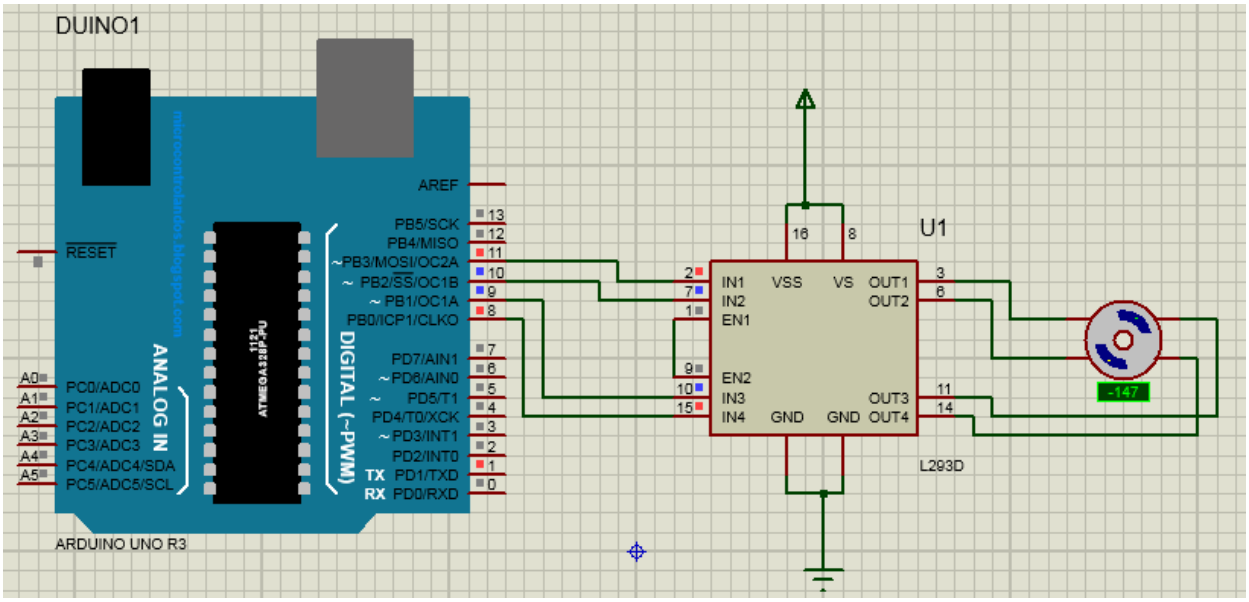

Figure 2.1 - Control scheme of a stepper motor in Proteus

2.2.2 Listing of program code.

To control stepper motors, the Arduino IDE has a standard library that provides only a full-step switching mode.

```
#include <Stepper.h>
const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
void setup() {
  myStepper.setSpeed(60);
  Serial.begin(9600);
}
void loop() {
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);
  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```

2.3 Report Content

2.3.1 Purpose of work.
2.3.2 Screenshot of the control circuit in Proteus.
2.3.3 Code Listing.
2.3.4 Real scheme on Arduino and BD (photo).
2.3.5 Conclusions.

2.4 Test Questions

2.4.1 What function does the SD L293D driver perform?
2.4.2 Why is the ST control sequence called half-step?
2.4.3 Why is the operation mode of the stepper motor called full-step?
2.4.4 How does the SD reverse?
2.4.5 What are the advantages of SD?

3 Laboratory work №3. LCD control circuit in Proteus

Purpose of work: modeling the LCD in the Proteus program and creating a display control code for the Arduino Uno microcontroller.

3.1 Summary of theory

LCD - an electronic device designed to display current information. Two types of this device are used: graphic and symbolic. Depending on the screen size, there are:

- 16x2 character - 16 columns and 2 lines;
- 20x4 characters - 20 columns and 4 lines.

The findings are labeled:

1 (VSS) - power to minus for the controller;
2 (VDD) - power on plus for the controller;
3 (VO) - contrast control settings;
4 (RS) - selection for the register;
5 (R / W) - reading and writing, in particular, writing when connected to the ground;
6 (E) - activation (enable);
7–10 (DB0-DB3) - low bits from the eight-bit interface;
11-14 (DB4-DB7) - high bits from the interface;
15 (A) - positive anode for backlight power;
16 (K) - negative cathode for backlight power.

3.2 The order of the work

3.2.1 Assemble the circuit in the Proteus environment (figure 3.1).

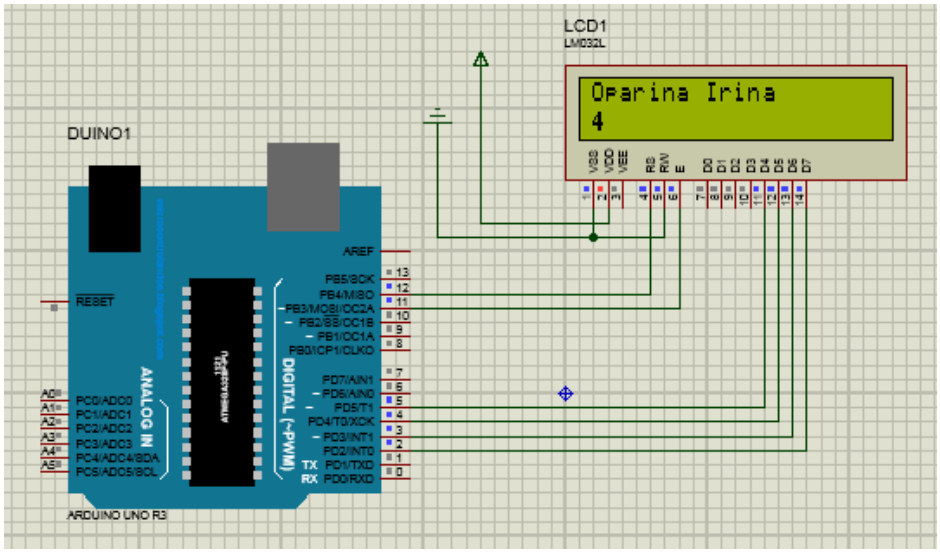

Figure 3.1 – Assembling diagram in Proteus

3.3.2 Listing of program code.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  lcd.begin(16, 2);
}
void loop() {
  lcd.setCursor(0,0);
  lcd.print("Oparina Irina");
  lcd.setCursor(0,1);
  lcd.print(millis ()/1000);
}
```

The actual LCD control circuit with I2C pins is shown in Figure 3.2.

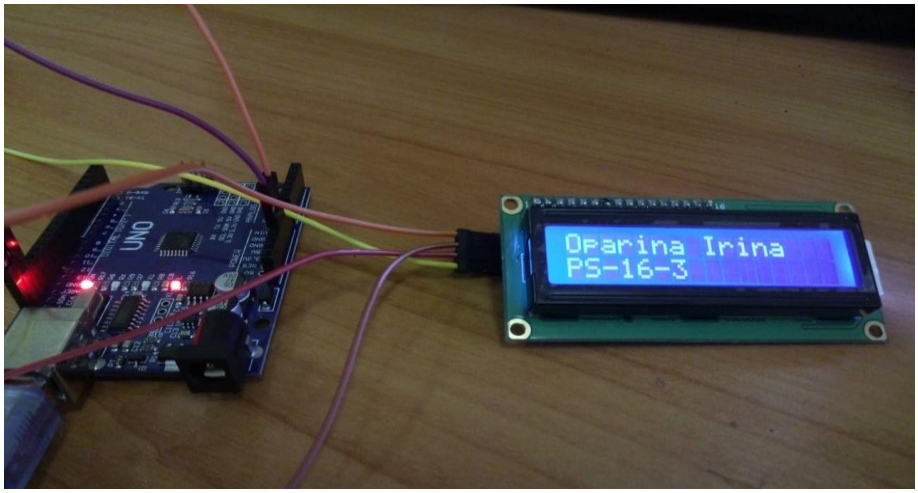

Figure 3.2 – Outward appearance LCD on Arduino

3.3 Report Content

3.3.1 Purpose of work.
3.3.2 Screenshot of the circuit in the Proteus environment.
3.3.3 code listing.
3.3.4 Real scheme on Arduino and LCD (photo).
3.3.5 Conclusions.

3.4 Test questions

3.4.1 How are comments highlighted in Arduino code?
3.4.2 Why is the R / W LCD terminal grounded?
3.4.3 Why is the VSS LCD pin grounded?
3.4.4 What number does the account end?
3.4.5 What type of LCD is used in this program?

4 Laboratory work №4. Simulation of a DHT11 temperature and humidity sensor in Proteus

Objective: to create a digital thermometer and humidity meter circuit in the Proteus environment and a program in the Arduino IDE environment.

4.1 Summary of theory

DHT11 is a digital sensor consisting of a thermistor and a capacitive humidity sensor. Food 3,5-5V, determination of temperature from 0 to 50 degrees with an accuracy of 2 degrees, determination of humidity from 20% to 95% with 5% accuracy.

A thermistor is a thermal resistor whose resistance varies with temperature, i.e. an increase in temperature leads to a drop in its resistance.

A capacitive humidity sensor is a capacitor with a variable capacity and is enclosed in a sealed case, on top of which a moisture-absorbing layer is located. When water particles get on this layer, its dielectric constant changes, which leads to a change in the capacitance of the capacitor.

4.2 The order of the work

4.2.1 Assemble the circuit shown in figure 4.1.

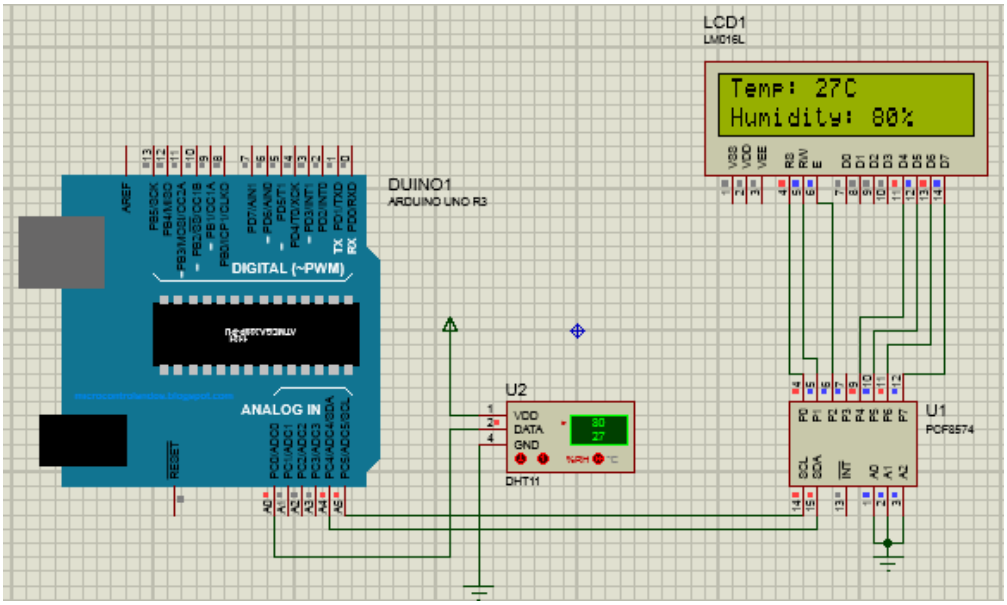

Figure 4.1 - Diagram of a digital thermometer in Proteus

4.2.2 Listing of program code.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#define DHTPIN A0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE); // create object "dht"
LiquidCrystal_I2C lcd(0x20,16,2);  // address and dimension lcd
void setup()
{
 lcd.begin(16, 2);
dht.begin();
}
void loop() {
  int h = dht.readHumidity();
  int t = dht.readTemperature();
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(t);
  lcd.print("C");
  lcd.setCursor(0,1);
  lcd.print("Humidity: ");
  lcd.print(h);
  lcd.print("%");
  delay (200);
}
```

4.3 Report content

4.3.1 Purpose of work.
4.3.2 Screenshot of the circuit in the Proteus environment.
4.3.3 Code Listing.
4.3.4 Real scheme on Arduino (photo).
4.3.5 Conclusions.

4.4 Test questions

4.4.1 How does a thermistor work?
4.4.2 How is humidity measured?
4.4.3 What is the number of moisture measurement ends?
4.4.4 What does the temperature measurement end?
4.4.5 What type of LCD is used in this program?

5 Laboratory work №5. Simulation of a digital voltmeter in Proteus

Objective: to study the principle of operation and design features of a digital voltmeter in the Proteus environment.

5.1 Summary of the theory

A voltmeter is a device for measuring voltage or EMF in electrical circuits. It is connected in parallel to a load or source of electrical energy. The greater the internal resistance in a real voltmeter, the less influence the device has on the measured object and the higher the accuracy.

5.2 The order of the work

5.2.1 Assemble the model diagram of a digital voltmeter in Proteus, shown in figure 5.1.

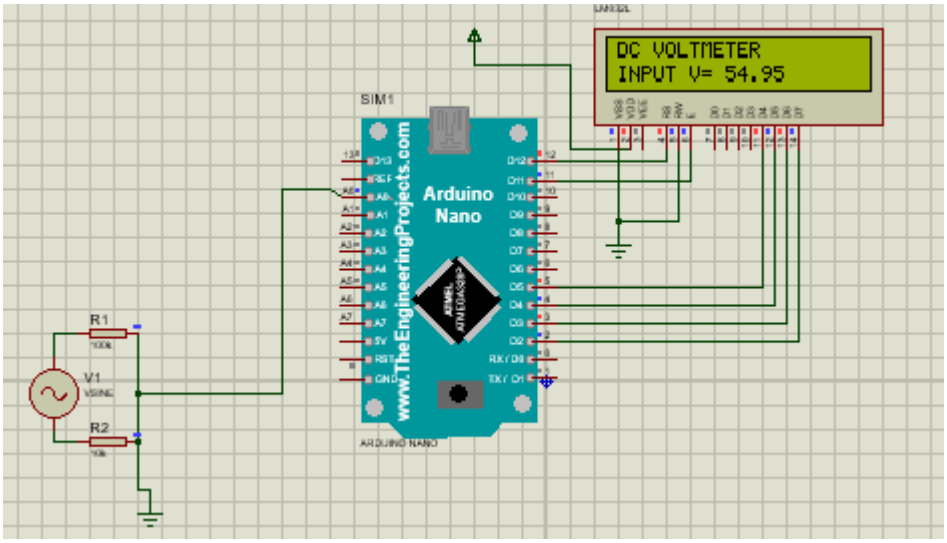

Figure 5.1 - Diagram of a model of a voltmeter in Proteus

By changing the voltage values at the output of voltage source V1 (VSINE), the measured voltage value is displayed on the LCD screen. The input of the voltmeter is the upper end of R1 and ground, i.e. The 2 conductors to V1 are a voltmeter probe.

5.2.2 Listing of program code.

```
#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; // R1 (100K)
float R2 = 10000.0; // R2 (10K)
int value = 0;
void setup(){
  pinMode(analogInput, INPUT);
  lcd.begin(16, 2);
  lcd.print("DC VOLTMETER");
}
void loop(){
  value = analogRead(analogInput);
  vout = (value * 5.0) / 1024.0;
  vin = vout / (R2/(R1+R2));
  if (vin<0.09) {
  vin=0.0;
}
lcd.setCursor(0, 1);
```

```
lcd.print("INPUT V= ");
lcd.print(vin);
delay(500);
}
```

5.3 Report content

5.3.1 Purpose of the work.
5.3.2 Screenshot of the circuit in the Proteus environment.
5.3.3 Code Listing.
5.3.4 Real scheme on Arduino (photo).
5.3.5 Conclusions.

5.4 Test questions

5.4.1 How does the ADC work?
5.4.2 What does the accuracy of measuring a digital voltmeter depend on?
5.4.3 What function do the resistors R1 and R2 perform in the circuit?
5.4.4 From what voltage value does the measurement begin?
5.4.5 What type of LCD is used in this program?

6 Laboratory work №6. Modeling a Bluetooth module in Proteus

Purpose of work: creating a circuit using the Arduino UNO board in the Proteus environment and the Bluetooth module control program in the Arduino IDE software environment.

6.1 Summary of the theory

The main advantages of Bluetooth include good broadband immunity and ease of implementation. Bluetooth protocol is necessary for fast data transmission over short distances. It will be convenient for managing objects from a smartphone if you download it with the corresponding application.

6.2 The order of the work

6.2.1 Assemble the LED control circuit via the Bluetooth channel and download the application for controlling the LED via the Bluetooth module of the smartphone (figure 6.1).

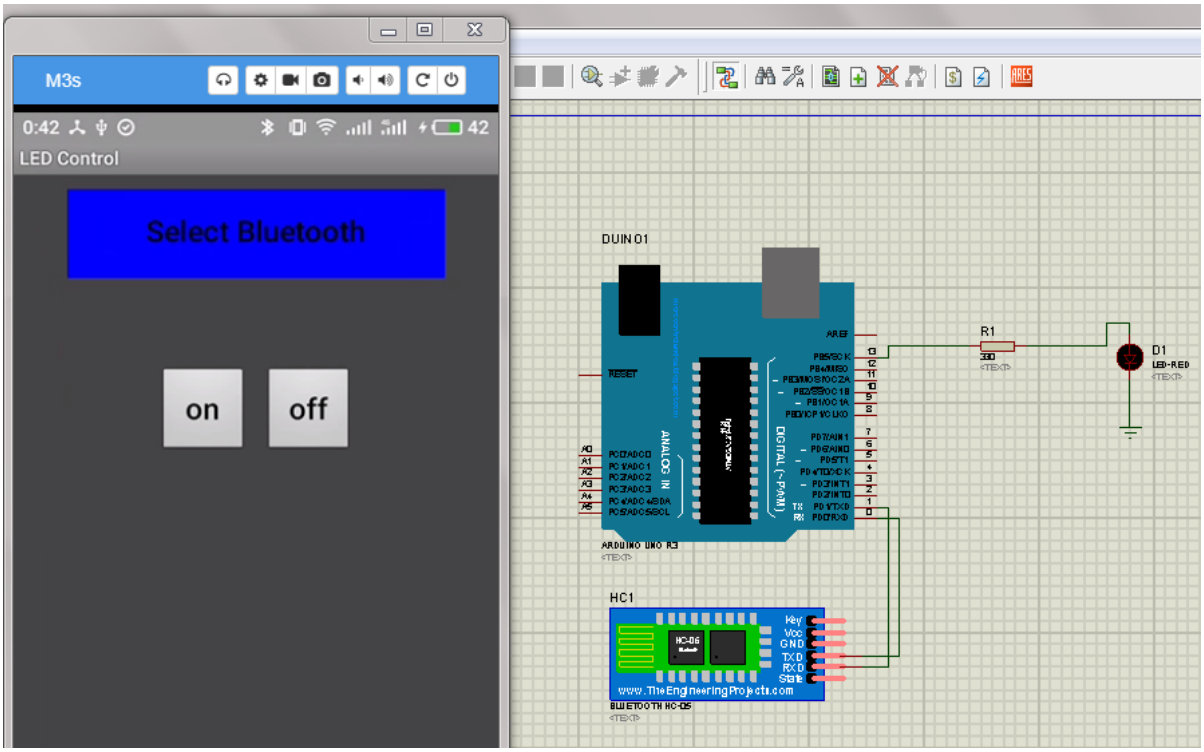

Figure 6.1 - LED control circuit via the Bluetooth module

6.2.2 Listing of program code.

```
String voice;
void setup() {
pinMode(13, OUTPUT);
Serial.begin(9600);
}
void loop() {
  while (Serial.available()) {
  delay(10);
  char c = Serial.read();
  voice += c;
  }
  if (voice.length() > 0)
  {
    Serial.println(voice);
       if(voice == "on")
       {
         digitalWrite(13, HIGH);
        }
  if(voice == "off")
  {
   digitalWrite(13, LOW);
   }
```

```
voice="";
   }
}
```

6.3 Report content

6.3.1 Purpose of work.
6.3.2 Screenshot of the circuit in the Proteus environment.
6.3.3 Code Listing.
6.3.4 Real scheme on Arduino (photo).
6.3.5 Conclusions.

6.4 Test questions

6.4.1 How does the ADC work?
6.4.2 What is the maximum distance to turn on the LED?
6.4.3 What function in the circuit is performed by the resistor R1?
6.4.4 Do external obstacles and the angle of the smartphone affect the operation of the LED?
6.4.5 What transfer protocol is used between the Bluetooth module and the Arduino board?

Bibliography

1 Igo T. Arduino, sensors and networks for communication devices: Trans. from English - SPb .: BHV-Petersburg, 2016. - 544 p.

2 Petin V.A. Arduino and Raspberry Pi in Internet of Things projects. - SPb.: BHV-Petersburg, 2016. - 464 p.

3 Petin V.A. - Microcomputers Raspberry Pi. A practical guide. SPb .: BHV-Petersburg, 2015. - 240 p.

4 Petin V.A. Projects using the Arduino controller - SPb .: BHV-Petersburg, 2016. - 464 p.

5 Prokhorenok N.A. HTML, JavaScript, PHP and MySQL. Dzhenlmensky set of Web-masters. - SPb .: BHV-Petersburg, 2019. - 912 p.

Content

Master plan 2019 г., poz.33

Baikenov Bakhytzhan Sergeevich
Ayazbay Abu-Alim Erikuly

MICROPROCESSOR CONTROL AND MONITORING SYSTEMS

Guidelines for the implementation of laboratory works
for students majoring 5B071600 – Instrument making

Editor M.D. Kurmanbekova
Standardization Specialist G. Mukhametsarieva

Signed to print 10.09.19
Circulation of 20 copies
Volume 1.1 publishing sheets

Format 60x84 1/16
Typographical paper № 1
Order _____Price 600 tg.

Copying office non-commercial joint stock company
"Almaty University of Power and Communications"
050013, Almaty, Baitursynov, 126/1