МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

Дата сдачи на проверку:

11 февраля 2025 г.

Проверено:

11 февраля 2025 г.

**Отчет по практической работе №4**

**ПЬЕЗОЭЛЕМЕНТ, МОТОР и ТЕКСТОВЫЙ ДИСПЛЕЙ**

по дисциплине

«Информационные технологии»

Вариант 2

Выполнил студент гр. УТБ-1301-02-00 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Борисов И. А/

(Подпись)

Проверил ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Шмакова Н. А/

(Подпись)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

Киров 2025

Цель работы: закрепление основ работы с Arduino.

1.1 Кнопочные ковбои

Индивидуальная задание представлено в таблице 1

Таблица 1 – Индивидуальное задание

|  |  |  |  |
| --- | --- | --- | --- |
| Начальное состояние  светодиодов вкл/выкл | Количество  игроков | После окончания игры  начинаем сначала/остановка | Выигрыш игрока |
| Выкл | 2 | + | пьезо |

1.2 Схема сборки “Кнопочные ковбои” представлена на рисунке 1.1

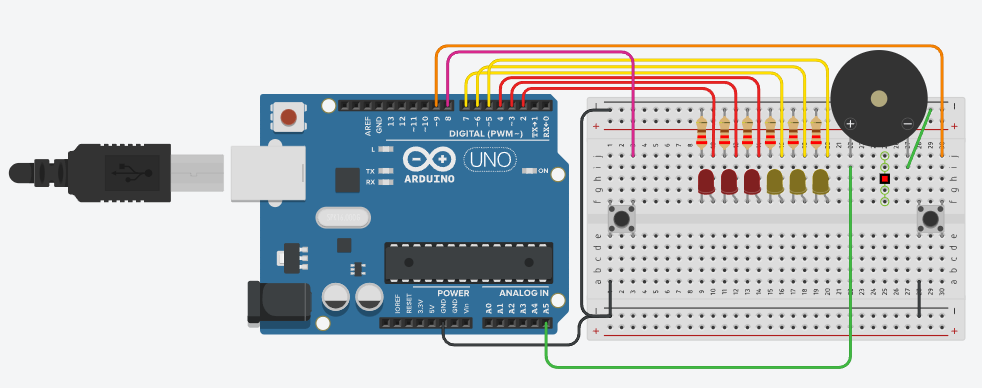


Рисунок 1.1 – Схема сборки на макетной плате

1.3 Принципиальная схема представлена на рисунке 1.2

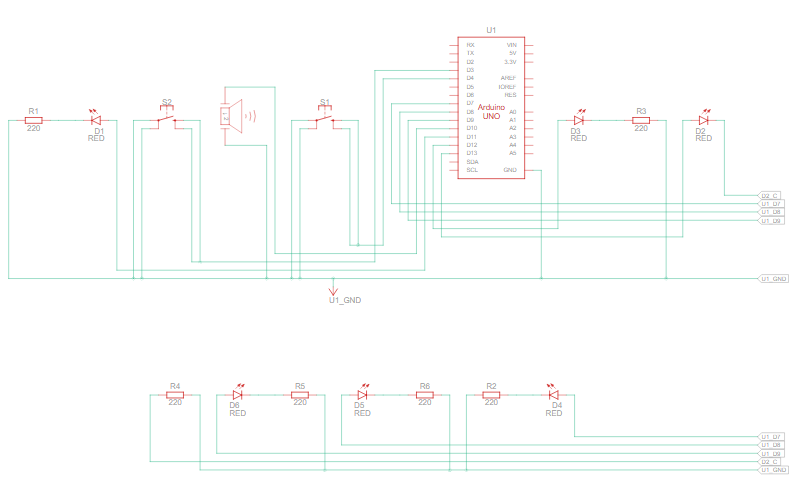


Рисунок 1.2 – Принципиальная схема

1.4 Листинг программного кода

#define BUZZER\_PIN 19 *// пин с пищалкой (англ. «buzzer»)* #define led\_COUNT 3 *// количество ламп каждому* #define PLAYER\_COUNT 2 *// количество игроков-ковбоев*int buttonPins[PLAYER\_COUNT] = {8, 9};  
int ledPins[PLAYER\_COUNT][led\_COUNT] = {{2, 3, 4},{5, 6, 7}};  
int liveCOUNT[PLAYER\_COUNT] = {0, 0};  
bool game = false;  
  
void setup() {  
 pinMode(BUZZER\_PIN, OUTPUT);  
 for (int j = 0; j < PLAYER\_COUNT; j++) {  
 for (int i = 0; i < led\_COUNT; i++) {  
 pinMode(ledPins[j][i], OUTPUT);  
 }  
 pinMode(buttonPins[j], INPUT\_PULLUP);  
 }  
}  
  
int procled(int pl, bool sost, int live) {  
 for (int k = 0; k < live; k++) {  
 digitalWrite(ledPins[pl][k], sost);  
 }  
}  
  
void loop() {  
  
 for (int j = 0; j < 2; j++){  
 procled(j, HIGH, liveCOUNT[j]);  
 }  
 delay(random(2000, 7000));  
 if (digitalRead(buttonPins[0])&& digitalRead(buttonPins[1])){  
 tone(BUZZER\_PIN, 4000, 500);  
 game = true;  
 }  
 if (game)  
 for (int player = 0; ;player = (player+1) % PLAYER\_COUNT) {  
 *// если игрок номер «player» нажал кнопку...* if (!digitalRead(buttonPins[player]) && game) {  
 procled(abs(player - 1), LOW, 3);  
 *// ...включаем его светодиоды и кол-во жизней  
 //добавляем очко победителю* liveCOUNT[player] = liveCOUNT[player] + 1;  
 procled(player, HIGH, liveCOUNT[player]);  
 delay(500);  
 procled(player, LOW, liveCOUNT[player]);  
 delay(500);  
 game = false;  
 *// Проверяем, достиг ли игрок 3 очков* if (liveCOUNT[player] >= 3) {  
  
 procled(player, HIGH, liveCOUNT[player]);  
 delay(500);  
 procled(player, LOW, liveCOUNT[player]);  
 delay(500);  
 *// Обнуляем очки всех игроков* for (int i = 0; i < PLAYER\_COUNT; i++) {  
 liveCOUNT[i] = 0;  
 }  
 }  
 break; *// Есть победитель! Выходим (англ. break) из цикла* }  
 }  
}

1.5 Ссылка на проект

[Кнопочные ковбои](https://www.tinkercad.com/things/aHOOc36nPLt-lab-41/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits%3Fpage%3D1)

2.1 Вывод текста

Индивидуальное задание представлено в таблице 2

Таблица 2 – Индивидуальное задание

|  |  |  |  |
| --- | --- | --- | --- |
| ЖК экран(I2C) | Кнопки/потенциометр | Подсветка дисплея | Отображаем строк |
| ­­­­­­\_ | кнопки | кнопки | 5 |

2.2 Схема сборки “Вывод текста” представлена на рисунке 2.1

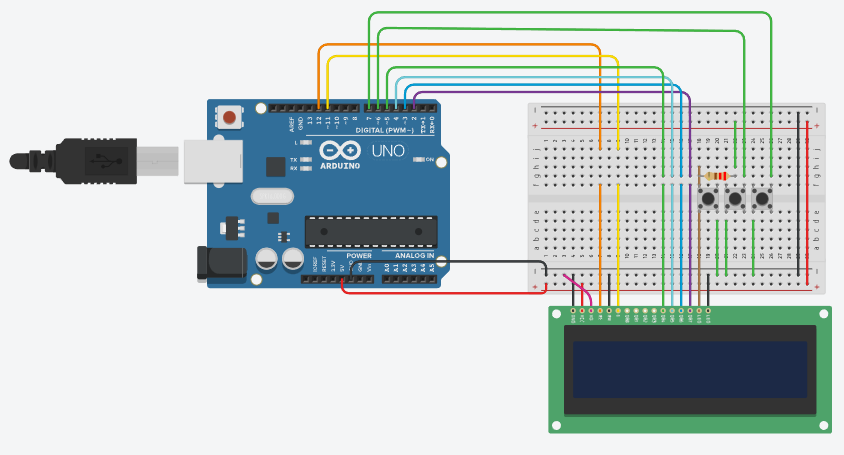


Рисунок 2.1 – Схема сборки на макетной плате

2.3 Принципиальная схема представлена на рисунках 2.2

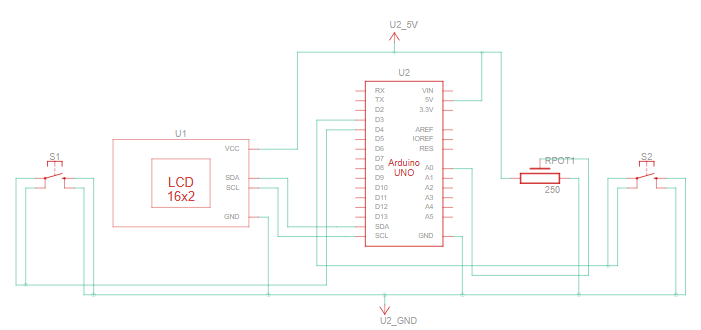


Рисунок 2.2 – Принципиальная схема

2.4 Листинг программного кода

#define button\_down\_pin 6  
 #define button\_up\_pin 7  
 #include <LiquidCrystal.h>  
#include<string.h>  
int seconds = 0;  
String pr = " ";  
String str = "We are trying t0 read the text using button We are trying t0 read the text using We are trying t0 read the text using buttonbutton 123";  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
int len = str.length();  
int size = (int)((len - len%16)/16);  
char output[15][16];  
void setup()  
{  
 lcd.begin(16, 2);  
 pinMode(button\_down\_pin,INPUT\_PULLUP);  
 pinMode(button\_up\_pin,INPUT\_PULLUP);  
 Serial.begin(9600);  
}  
int count = 0;  
void generate() {  
 for (int i = 0; i < (16 \* (size +1) - len); i++) {  
 str = str + pr;}  
 for (int i = 0; i <= size+1; i++) {  
 for (int j = 0; j < 16; j++) {  
 output[i][j] = str[count];  
 count++;  
 }  
 }  
 count = 0;  
}  
int index = 0;  
void loop()  
{  
 if (count == 0) {generate(); count = 1;}  
 if (!digitalRead(button\_down\_pin)) {  
 index = (index + 1);  
 if (index == size) {  
 index = index - 1;  
 }  
 }  
 if (!digitalRead(button\_up\_pin)) {  
 index = (index - 1);  
 if (index<0) {  
 index = index + 1;  
 }  
 }  
 Serial.print(index);  
 Serial.println(size);  
 for (int i=0; i < 16; i++) {  
 lcd.setCursor(i, 0);  
 lcd.print(output[index][i]);  
 lcd.setCursor(i, 1);  
 lcd.print(output[(index+1)][i]);  
 }  
  
 seconds += 1;  
}

2.5 Ссылка на проект

[Вывод текста](https://www.tinkercad.com/things/hxwe2LgPGX7-lr-42/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits%3Fpage%3D1)

3.1 Тестер батареек

Индивидуальное задание представлено в таблице 3

Таблица 1 – Индивидуальное задание

|  |  |  |  |
| --- | --- | --- | --- |
| ЖК экран(I2C) | Движение строки | По кнопке | На второй строке  отображать |
| + | Слева-направо | Смена направления | Время работы программы |

3.2 Схема сборки “ Тестер батареек” представлена на рисунке 3.1

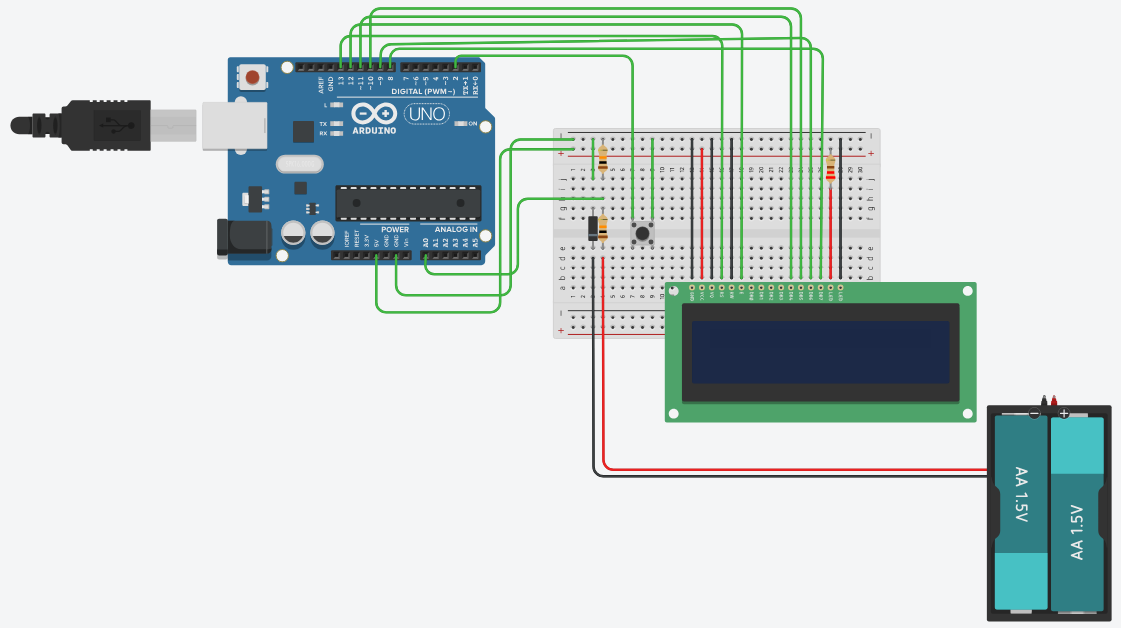


Рисунок 3.1 – Схема сборки на макетной плате

3.3 Принципиальная схема представлена на рисунках 3.2

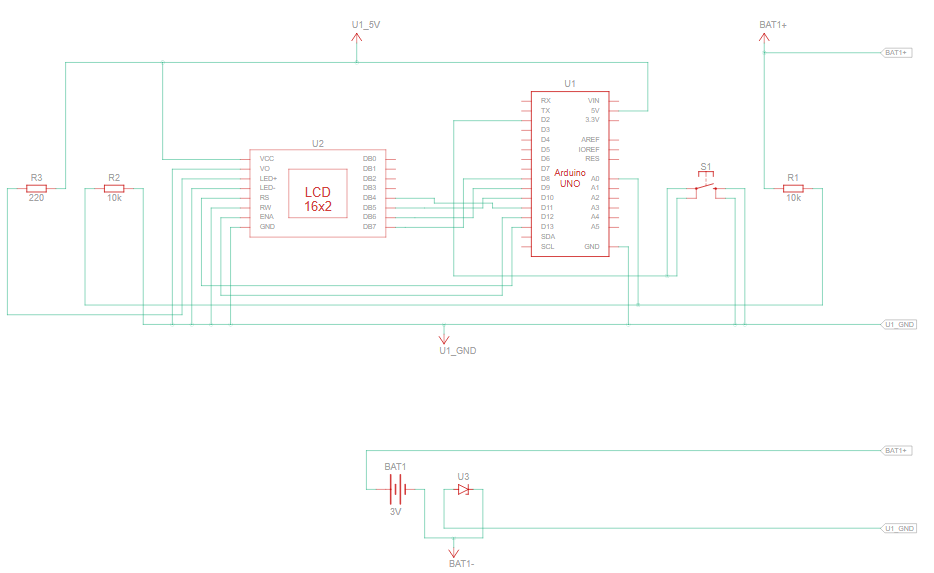


Рисунок 3.2 – Принципиальная схема

3.4 Листинг программного кода

#include <LiquidCrystal.h>  
#define BUTTON\_PIN 2 *// Измените пин кнопки на 2 для использования прерываний*LiquidCrystal lcd(13, 12, 11, 10, 9, 8);  
int position = 0;  
String message1 = " I did this job. ";  
String message2 = " My favorite color is blue ";  
String currentMessage = message1;  
volatile bool buttonPressed = false;  
unsigned long startTime = 0;  
  
void setup() {  
 lcd.begin(16, 2);  
 lcd.setCursor(0, 1);  
 lcd.print(" " + String((millis() - startTime) / 1000) + "s "); *// Время работы программы в секундах* pinMode(BUTTON\_PIN, INPUT\_PULLUP);  
 attachInterrupt(digitalPinToInterrupt(BUTTON\_PIN), changeMessage, FALLING);  
 startTime = millis(); *// Получаем время запуска программы*}  
  
void loop() {  
 lcd.setCursor(0, 1);  
 lcd.print(" " + String((millis() - startTime) / 1000) + "s "); *// Время работы программы в секундах* if (buttonPressed) {  
 if (currentMessage == message1) {  
 currentMessage = message2;  
 } else {  
 currentMessage = message1;  
 }  
 buttonPressed = false;  
 }  
  
 lcd.setCursor(position, 0);  
 lcd.print(currentMessage.substring(0, currentMessage.length() - position));  
 lcd.setCursor(0, 0);  
 lcd.print(currentMessage.substring(currentMessage.length() - position, currentMessage.length()));  
 position = (position + 1) % currentMessage.length();  
 delay(200);  
}  
  
void changeMessage() {  
 buttonPressed = true;  
}

3.5 Ссылка на проект

[Тестер батареек](https://www.tinkercad.com/things/kEdpAOB9PbN-43-metodichka/editel?returnTo=%2Fdashboard%2Fdesigns%2Fcircuits&sharecode=da9Z3Am8Ea8oHi5gmtsVag8T1AulT1eH7chO5P3hkS4%20)

4.1 Задание с мотором

|  |
| --- |
| При срабатывании любых 2х датчиков включать мотор |

4.2 Листинг программного кода

#define leftLinePin A0  
#define leftDirPin 2  
 #define motor 5  
  
 #define rightLinePin A1  
#define rightDirPin 3  
  
 #define tiltLed 8  
 #define tiltSensor A2  
  
int runSpeed = 0;  
  
void setup() {  
 pinMode(leftDirPin, OUTPUT);  
 pinMode(rightDirPin, OUTPUT);  
 pinMode(tiltLed, OUTPUT);  
 pinMode(motor, OUTPUT);  
 Serial.begin(9600);  
}  
  
void loop() {  
 bool whiteLeft = digitalRead(leftLinePin);  
 bool whiteRight = digitalRead(rightLinePin);  
 bool tilt = digitalRead(tiltSensor);  
  
 if (whiteLeft) digitalWrite(leftDirPin, HIGH);  
 else digitalWrite(leftDirPin, LOW);  
 if (whiteRight) digitalWrite(rightDirPin, HIGH);  
 else digitalWrite(rightDirPin, LOW);  
 if (tilt) digitalWrite(tiltLed, HIGH);  
 else digitalWrite(tiltLed, LOW);  
 Serial.println(runSpeed);  
  
 if (whiteLeft || whiteRight) {  
 runSpeed += 10; *// Увеличиваем скорость на 10* Serial.print(runSpeed);  
 delay(1500);  
 if (runSpeed > 250) {  
 runSpeed = 250; *// Ограничиваем максимальную скорость* }  
 } else {  
 runSpeed = 0; *// Устанавливаем скорость в 0, если ни один датчик не сработал* }  
  
 if (tilt) {  
 runSpeed -= 20; *// Уменьшаем скорость на 10* if (runSpeed < 20) {  
 runSpeed = 20; *// Ограничиваем минимальную скорость* }  
 }  
 analogWrite(motor, runSpeed);  
}

5.1 Светодиодная матрица 8x8 (Troyka модуль)

Индивидуальное задание представлено на рисунке 5.1

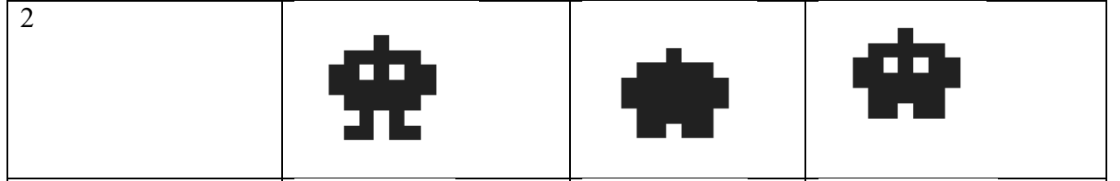


Рисунок 5.1 Индивидуальное задание

5.2 Листинг программного кода

#define FIRST\_SEGMENT\_PIN 0  
 #define LAST\_SEGMENT\_PIN 13  
int count = 0; *// первоначальное значение*int n = 0;  
int pins[7] = {0, 1, 2, 3, 4, 5, 6,};  
byte numberSegments[10] = { 0b0111111, 0b0000110, 0b1011011, 0b1001111, 0b1100110, 0b1101101, 0b1111101, 0b0000111, 0b1111111, 0b1101111 };  
void showDigit(int num, bool digit ) *// num - само число. digit - true если единицы*{  
 int k = 0;  
 if (digit) k = 7;  
 else k = 0;  
 int mask = numberSegments[num];  
 for (int i = 0; i < 7; i++) {  
 boolean enableSegment = bitRead(mask, i);  
 digitalWrite(i + k, !enableSegment);  
 }  
}  
  
int length(int value)  
{  
 int len = 0;  
 do  
 {  
 value /= 10;  
 len++;  
 } while (value);  
 return len;  
}  
int fib(int n)  
{  
 int c = 0, p = 1, q;  
 for (int i = 1; i <= n; ++i)  
 {  
 q = c + p;  
 p = c;  
 c = q;  
 }  
 return c;  
}  
  
void setup()  
{  
 for (int i = 0; i <= LAST\_SEGMENT\_PIN; ++i)  
 pinMode(i + FIRST\_SEGMENT\_PIN, OUTPUT);  
  
 if (length(count) == 2)  
 {  
 showDigit(count / 10, false); *// десятки* showDigit(count % 10, true); *// единицы* }  
 else  
 {  
 showDigit(0, false);  
 showDigit(count % 10, true);  
 }  
 delay(2000);  
}  
  
void loop()  
{  
 count = fib(n);  
 if (count > 100) while (true);  
 if (length(count) == 2)  
 {  
 showDigit(count / 10, false); *// десятки* showDigit(count % 10, true); *// единицы* }  
 else  
 {  
 showDigit(0, false);  
 showDigit(count % 10, true);  
 }  
 delay(2000);  
 n++;  
}

4.6 Светодиодная матрица АЛС340А1

|  |
| --- |
| Вывести на матрицу первую букву своей фамилии и затем номер варианта поочередно. |

6.2 Листинг программного кода

#define COUNT\_LED 12  
int pins[] = {0, 1, 2, 3, 4, 0, 0, 7, 8, 9, 10, 11, 12, 13};  
int anode[] = {5, 1, 8, 14, 13};  
int cathode[] = {2, 12, 3, 4, 11, 10, 9};  
  
void setup()  
{  
 for (int i = 0; i <= 13; ++i)  
 pinMode(i, OUTPUT);  
 *//Serial.begin(9600);* clearIndicator();  
}  
int status = 0;  
void loop() {  
 *// P* int number = (millis() / 1000) % 10;  
 if (number < 5) {  
 digitalWrite(2 - 1, LOW);  
 digitalWrite(5 - 1, HIGH);  
 digitalWrite(1 - 1, HIGH);  
 digitalWrite(8 - 1, HIGH);  
 digitalWrite(14 - 1, HIGH);  
 digitalWrite(13 - 1, HIGH);  
 clearIndicator();  
  
 digitalWrite(5 - 1, HIGH);  
 digitalWrite(2 - 1, LOW);  
 digitalWrite(12 - 1, LOW);  
 digitalWrite(3 - 1, LOW);  
 digitalWrite(4 - 1, LOW);  
 digitalWrite(11 - 1, LOW);  
 digitalWrite(10 - 1, LOW);  
 digitalWrite(9 - 1, LOW);  
 clearIndicator();  
  
 digitalWrite(9 - 1, LOW);  
 digitalWrite(5 - 1, HIGH);  
 digitalWrite(1 - 1, HIGH);  
 digitalWrite(8 - 1, HIGH);  
 digitalWrite(14 - 1, HIGH);  
 digitalWrite(13 - 1, HIGH);  
 clearIndicator();  
 }  
 else  
 {  
 digitalWrite(5 - 1, HIGH);  
 digitalWrite(2 - 1, LOW);  
 digitalWrite(12 - 1, LOW);  
 digitalWrite(3 - 1, LOW);  
 digitalWrite(4 - 1, LOW);  
 digitalWrite(11 - 1, LOW);  
 digitalWrite(10 - 1, LOW);  
 digitalWrite(9 - 1, LOW);  
 clearIndicator();  
  
 digitalWrite(13 - 1, HIGH);  
 digitalWrite(2 - 1, LOW);  
 digitalWrite(12 - 1, LOW);  
 digitalWrite(3 - 1, LOW);  
 digitalWrite(4 - 1, LOW);  
 digitalWrite(11 - 1, HIGH);  
 digitalWrite(10 - 1, HIGH);  
 digitalWrite(9 - 1, LOW);  
 clearIndicator();  
  
 digitalWrite(9 - 1, LOW);  
 digitalWrite(8 - 1, HIGH);  
 digitalWrite(14 - 1, HIGH);  
 clearIndicator();  
  
 digitalWrite(4 - 1, LOW);  
 digitalWrite(8 - 1, HIGH);  
 digitalWrite(14 - 1, HIGH);  
 clearIndicator();  
  
 digitalWrite(2 - 1, LOW);  
 digitalWrite(8 - 1, HIGH);  
 digitalWrite(14 - 1, HIGH);  
 clearIndicator();  
  
 digitalWrite(8 - 1, HIGH);  
 digitalWrite(2 - 1, HIGH);  
 digitalWrite(12 - 1, HIGH);  
 digitalWrite(3 - 1, HIGH);  
 digitalWrite(4 - 1, LOW);  
 digitalWrite(11 - 1, LOW);  
 digitalWrite(10 - 1, LOW);  
 digitalWrite(9 - 1, LOW);  
 clearIndicator();  
 }  
}  
  
void clearIndicator() {  
 for (int i = 0; i <= 6; ++i) digitalWrite(cathode[i]-1, HIGH);  
 for (int i = 0; i <= 4; ++i) digitalWrite(anode[i]-1, LOW);  
}

Вывод

В результате лабораторной работы были изучены новые детали для Arduino IDE такие как (экраны, батарейки, моторчики, датчики наклона и линии).