

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 электронных вычислительных машин

Отчет по лабораторной работе №4
по дисциплине
«Информатика»
«Пьезоэлемент, мотор и текстовый дисплей»
Вариант 6

Выполнил студент гр. ИВТб-1301-05-00	_____ /Макаров С.А./
Руководитель преподаватель	_____ /Шмакова Н.А./

Киров 2025

Цель работы

Цель работы: Закрепление основ работы с Arduino.

Задание

1. Кнопочные ковбои. Начальное состояние светодиодов выключено. 3 игрока, для каждого по два светодиода. После окончания игры она начинается сначала. Выиграш игорка сигнализируется свеодиодом. игрок зарабатывает бонусы за нажатие, по нажатию на кнопку, если игрок нажал первый после звукового сигнала, то он зарабатывает бонус.
2. Вывод текста. Текст задан строкой 50–100 символов. Мы пытаемся прочитать текст с помощью навигации с помощью потенциометра. Дополнительно мы можем включать и выключать подсветку экрана кнопкой.
3. Тестер батареек. Текст задан строкой 20–30 символов. Организовать бегущую строку без использования циклов и встроенной функции scroll.
4. Задание с мотором. Дополнить схему платой расширения для моторов и датчиков. Подключить 2 датчика линии и 1 датчик наклона к плате и 3 светодиода для легкого контроля за датчиками.
5. Светодиодная матрица 8x8 (Тройка модуль). Отобразить на матрице изменение состояний. По желанию можно дополнить или продублировать состояния для плавного перехода или более интересной анимации.
6. Светодиодная матрица АЛС340А1. Вывести на матрицу первую букву своей фамилии и затем номер варианта поочередно.
7. Дополнительное задание. Выполнить задание 5 на матрице АЛС340А1

Решение

Задание 1

Выполнена живая сборка. Исходный код программы:

```
#define SHOOT_LED A5
#define PLAYER_COUNT 3
int buttonPins[PLAYER_COUNT] = {11, 12, 13};
int shootLeds[PLAYER_COUNT] = {8, 9, 10};
int scoreLeds[PLAYER_COUNT * 2] = {2, 3, 4, 5, 6, 7};
int scoreCounts[PLAYER_COUNT] = {0, 0, 0};
void setup() {
    pinMode(SHOOT_LED, OUTPUT);
    for (int i = 0; i < PLAYER_COUNT; i++) {
        pinMode(buttonPins[i], INPUT_PULLUP);
        pinMode(shootLeds[i], OUTPUT);
        pinMode(scoreLeds[i], OUTPUT);
        pinMode(scoreLeds[i + 1], OUTPUT);
        pinMode(scoreLeds[i + 2], OUTPUT);
    }
}
void loop(){
    delay(random(2000, 7000));
    while(!digitalRead(buttonPins[0]) ||
        !digitalRead(buttonPins[1]) ||
        !digitalRead(buttonPins[2])) {}
    digitalWrite(SHOOT_LED, HIGH);
    for (int player = 0; ; player = (player + 1) % PLAYER_COUNT) {
        if (!digitalRead(buttonPins[player])) {
            digitalWrite(shootLeds[player], HIGH);
            scoreCounts[player]++;
            for (int i = 0; i < scoreCounts[player]; i++)
                digitalWrite(scoreLeds[i + player * 2], HIGH);
            if (scoreCounts[player] == 2) {
                delay(4000);
                scoreCounts[0] = 0;
                scoreCounts[1] = 0;
                scoreCounts[2] = 0;
                for (int i = 0; i < PLAYER_COUNT * 2; i++)
                    digitalWrite(scoreLeds[i], LOW);
            } else {
                delay(2000);
            }
            digitalWrite(shootLeds[player], LOW);
            digitalWrite(SHOOT_LED, LOW);
            break;
        }
    }
}
```

Задание 2

Выполнена живая сборка. Исходный код программы:

```
#include <LiquidCrystal.h>
#define POT_PIN A0
#define BUTTON_PIN 3
#define BACKLIGHT_PIN 5
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
String text = "Lorem ipsum dolor sit amet, cons ectetur
    adipiscing elit, sed do eiusmod tempor incididunt ut labore et
    dolore magna aliqua. Lorem ipsum dolor sit amet, cons ectetur
    adipiscing elit, sed do eiusmod tempor incididunt ut labore et
    dolore magna aliqua.";
int rows = (text.length() + 15) / 16;
int scroll = max(0, rows - 2);
volatile boolean isEnabled = true;
void setup() {
    lcd.begin(16, 2);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    pinMode(BACKLIGHT_PIN, OUTPUT);
    attachInterrupt(1, switchDisplay, FALLING);
}
void switchDisplay() {
    isEnabled = !isEnabled;
}
void updateDisplay(int start) {
    lcd.clear();
    for (int i = 0; i < 2; i++) {
        int index = start + i;
        if (index < rows) {
            int position = index * 16;
            String row = text.substring(position,
                min(position + 16, text.length()));
            lcd.setCursor(0, i);
            lcd.print(row);
        }
    }
}
void loop() {
    if (isEnabled) {
        digitalWrite(BACKLIGHT_PIN, HIGH);
    } else {
        digitalWrite(BACKLIGHT_PIN, LOW);
    }
    int potValue = analogRead(POT_PIN);
    int position = map(potValue, 0, 1023, 0, scroll);
    updateDisplay(position);
    delay(200);
}
```

Задание 3

Выполнена живая сборка. Исходный код программы:

```

#include <Wire.h>
#include <hd44780.h>
#include <hd44780ioClass/hd44780_I2Cexp.h>
#define BUTTON_PIN 3
hd44780_I2Cexp lcd(0x27, 16, 2);
String message = "hello computer science ";
volatile boolean direction = true;
int length = message.length();
int position = 0;
void setup() {
    lcd.init();
    lcd.backlight();
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    attachInterrupt(1, changeDirection, FALLING);
}
void changeDirection() {
    direction = !direction;
}
void loop() {
    lcd.clear();
    lcd.setCursor(0, 0);
    String displayMessage = message.substring(position,
        min(position + 16, length));
    lcd.print(displayMessage);
    if (displayMessage.length() < 16 && length > 16)
        lcd.print(message.substring(0, 16 - displayMessage.length()));
    if (direction) {
        position++;
        if (position == length)
            position = 0;
    } else {
        position--;
        if (position == 0)
            position = length;
    }
    lcd.setCursor(0, 1);
    lcd.print(millis() / 1000);
    delay(700);
}

```

Задание 4

Выполнена живая сборка. Исходный код программы:

```

#define LINE_SENSOR_1 A0
#define LINE_SENSOR_2 A1
#define TILT_SENSOR A2
#define DIR_PIN 4
#define SPEED_PIN 5
#define LED_PIN_WHITE 8
#define LED_PIN_BLUE 9
#define LED_PIN_RED 10
void setup() {
    pinMode(DIR_PIN, OUTPUT);
}

```

```

pinMode(SPEED_PIN, OUTPUT);
pinMode(LED_PIN_WHITE, OUTPUT);
pinMode(LED_PIN_BLUE, OUTPUT);
pinMode(LED_PIN_RED, OUTPUT);
Serial.begin(9600);
}
void loop() {
  int valueSensor1 = analogRead(LINE_SENSOR_1);
  int valueSensor2 = analogRead(LINE_SENSOR_2);
  int valueTilted = analogRead(TILT_SENSOR);
  digitalWrite(DIR_PIN, LOW);
  boolean isS1 = valueSensor1 > 500 ? true : false;
  boolean isS2 = valueSensor2 < 500 ? true : false;
  boolean isS3 = valueTilted < 500 ? true : false;
  isS1 ? digitalWrite(LED_PIN_WHITE, HIGH)
    : digitalWrite(LED_PIN_WHITE, LOW);
  isS2 ? digitalWrite(LED_PIN_BLUE, HIGH)
    : digitalWrite(LED_PIN_BLUE, LOW);
  isS3 ? digitalWrite(LED_PIN_RED, HIGH)
    : digitalWrite(LED_PIN_RED, LOW);
  if (isS3) {
    analogWrite(SPEED_PIN, 255);
  } else if (isS1 || isS2) {
    analogWrite(SPEED_PIN, 80);
  } else {
    analogWrite(SPEED_PIN, 0);
  }
}
}

```

Задание 5

Выполнена живая сборка. Исходный код программы:

```

#include <Wire.h>
#include <TroykaLedMatrix.h>
#define SDA_PIN 5
#define SCL_PIN 4
TwoWire customWire = TwoWire(1);
TroykaLedMatrix matrix;
const uint8_t sprite_1[] {
  0b00111100,
  0b00111100,
  0b00111100,
  0b11111111,
  0b01011010,
  0b01111110,
  0b01000010,
  0b00111100,
};
const uint8_t sprite_2[] {
  0b00111100,
  0b00111100,
  0b00111100,
  0b11111111,
};

```

```

    0b01111010,
    0b01111110,
    0b01111000,
    0b00111100,
};
const uint8_t sprite_3[] {
    0b00111100,
    0b00111100,
    0b00111100,
    0b11111111,
    0b01111110,
    0b01111110,
    0b01111110,
    0b00111100,
};
const uint8_t sprite_4[] {
    0b00111100,
    0b00111100,
    0b00111100,
    0b11111111,
    0b01011110,
    0b01111110,
    0b00011110,
    0b00111100,
};
void setup() {
    customWire.begin(SDA_PIN, SCL_PIN, 100000);
    matrix.begin(customWire);
}
void loop() {
    matrix.clear();
    matrix.drawBitmap(sprite_1);
    delay(300);
    matrix.clear();
    matrix.drawBitmap(sprite_2);
    delay(300);
    matrix.clear();
    matrix.drawBitmap(sprite_3);
    delay(300);
    matrix.clear();
    matrix.drawBitmap(sprite_4);
    delay(300);
}

```

Задание 6

Выполнена живая сборка. Исходный код программы:

```

int columnPins[5] = {2, 3, 4, 5, 6};
int rowPins[7] = {7, 8, 9, 10, 11, 12, 13};
byte frame1[7] = {
    0b11111,
    0b10000,
    0b10000,
    0b11111,
    0b10001,

```

```

    0b10001,
    0b11111
};
byte frame2[7] = {
    0b10001,
    0b11011,
    0b10101,
    0b10001,
    0b10001,
    0b10001,
    0b10001
};
int lastSwitch = 0;
int interval = 1000;
boolean toggle = false;
void setup() {
    for (int i = 0; i < 5; i++) {
        pinMode(columnPins[i], OUTPUT);
        digitalWrite(columnPins[i], LOW);
    }
    for (int i = 0; i < 7; i++) {
        pinMode(rowPins[i], OUTPUT);
        digitalWrite(rowPins[i], HIGH);
    }
}
void loop() {
    unsigned long currentTime = millis();
    if (toggle) {
        displayFrame(frame1);
    } else {
        displayFrame(frame2);
    }
    if (currentTime - lastSwitch >= interval) {
        toggle = !toggle;
        lastSwitch = currentTime;
    }
}
void displayFrame(byte frame[7]) {
    for (int row = 0; row < 7; row++) {
        digitalWrite(rowPins[row], LOW);
        for (int col = 0; col < 5; col++) {
            if (bitRead(frame[row], 4 - col)) {
                digitalWrite(columnPins[col], HIGH);
            } else {
                digitalWrite(columnPins[col], LOW);
            }
        }
        for (int col = 0; col < 5; col++) {
            digitalWrite(columnPins[col], LOW);
        }
        digitalWrite(rowPins[row], HIGH);
    }
}
}

```


Дополнительное задание

Выполнена живая сборка. Исходный код программы:

```
int columnPins[5] = {2, 3, 4, 5, 6};
int rowPins[7] = {7, 8, 9, 10, 11, 12, 13};
byte sprite1[] {
    0b011110,
    0b011110,
    0b111111,
    0b10101,
    0b111111,
    0b10001,
    0b011110
};
byte sprite2[7] {
    0b011110,
    0b011110,
    0b111111,
    0b11101,
    0b111111,
    0b11100,
    0b011110
};
byte sprite3[7] {
    0b011110,
    0b011110,
    0b111111,
    0b111111,
    0b111111,
    0b111111,
    0b011110
};
byte sprite4[7] {
    0b011110,
    0b011110,
    0b111111,
    0b10111,
    0b111111,
    0b00111,
    0b011110
};
int lastSwitch = 0;
int interval = 500;
int i = 0;
void setup() {
    for (int i = 0; i < 5; i++) {
        pinMode(columnPins[i], OUTPUT);
        digitalWrite(columnPins[i], LOW);
    }
    for (int i = 0; i < 7; i++) {
        pinMode(rowPins[i], OUTPUT);
        digitalWrite(rowPins[i], HIGH);
    }
}
void loop() {
    int currentTime = millis();
    if (i == 0) {
        displayFrame(sprite1);
```

```

    } else if (i == 1) {
        displayFrame(sprite2);
    } else if (i == 2) {
        displayFrame(sprite3);
    } else if (i == 3) {
        displayFrame(sprite4);
    }
    if (currentTime - lastSwitch >= interval) {
        i = (i + 1) % 4;
        lastSwitch = currentTime;
    }
}
void displayFrame(byte frame[7]) {
    for (int row = 0; row < 7; row++) {
        digitalWrite(rowPins[row], LOW);
        for (int col = 0; col < 5; col++) {
            if (bitRead(frame[row], 4 - col)) {
                digitalWrite(columnPins[col], HIGH);
            } else {
                digitalWrite(columnPins[col], LOW);
            }
        }
        for (int col = 0; col < 5; col++) {
            digitalWrite(columnPins[col], LOW);
        }
        digitalWrite(rowPins[row], HIGH);
    }
}
}

```

Вывод

В ходе лабораторной работы закрепились основы работы в среде Arduino IDE, а также собраны схемы пьезоэлементами, светодиодами, моторами и текстовыми дисплеями.