

**Задание**

Овладеть вводом/выводом аналогового сигнала в системе Arduino

1. Исследование ввода/вывода аналогового сигнала в Arduino IDE.

2. Запустить программу Arduino IDE.

3. Объявить данные в Arduino IDE.

4. Задать разрешенные величины данных в Arduino IDE.

5. Запустить отладку программы в Arduino IDE.

6. Вывести значения данных с потенциометра на монитор.

7. Преобразовать значения с потенциометра для вывода в аналоговый порт.

8. Выводить значения потенциометра 0, ½ и 1 на светодиоды, подключенные к выводам 9, 10 и 11 (разные цвета).

9. Оформить отчет по лабораторной работе по правилам, установленным в ГУАП.

10. В отчете должна быть приведена разработанная и откомпилированная программа.

**Ход работы**

В рамках лабораторной работы была изучена работа с аналоговым вводом и выводом сигналов в системе Arduino. Для этого использовался потенциометр, подключенный к аналоговому входу A0, и три светодиода, подключенные к цифровым выходам 9, 10 и 11.

В начале программы были объявлены пины для светодиодов и потенциометра:

const int ledPin1 = 9; // Светодиод 1 (красный)

const int ledPin2 = 10; // Светодиод 2 (зеленый)

const int ledPin3 = 11; // Светодиод 3 (синий)

const int potPin = A0; // Пин для потенциометра

В функции setup() произведена настройка пинов светодиодов как выходных и инициализация последовательного порта для вывода данных:

void setup() {

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

pinMode(ledPin3, OUTPUT);

Serial.begin(9600);

}

В основном цикле программы (loop) реализовано чтение аналогового сигнала с потенциометра с помощью функции analogRead(potPin). Значение с потенциометра варьируется от 0 до 1023 и выводится на монитор последовательного порта:

int potValue = analogRead(potPin);

Serial.print("Потенциометр: ");

Serial.println(potValue);

Для управления светодиодами значение с потенциометра было разделено на три сектора: 0-341, 342-682 и 683-1023. В зависимости от сектора, включался соответствующий светодиод: красный (пин 9) для значений около 0, зеленый (пин 10) для значений около ½ и синий (пин 11) для значений около 1:

int maxled = 1023 / 3;

int ledpin = potValue / maxled;

if (ledpin == 0) {

digitalWrite(ledPin1, HIGH);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin3, LOW);

} else if (ledpin == 1) {

digitalWrite(ledPin2, HIGH);

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin3, LOW);

} else {

digitalWrite(ledPin3, HIGH);

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, LOW);

}

Для стабилизации работы программы добавлена задержка в 50 мс:

delay(50);

Полный код программы:

// Пины для подключения светодиодов

const int ledPin1 = 9; // Светодиод 1 (красный)

const int ledPin2 = 10; // Светодиод 2 (зеленый)

const int ledPin3 = 11; // Светодиод 3 (синий)

// Пин для потенциометра

const int potPin = A0;

void setup() {

// Инициализация пинов для светодиодов как выходных

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

pinMode(ledPin3, OUTPUT);

// Инициализация последовательного порта для вывода данных

Serial.begin(9600);

}

void loop() {

// Чтение значения с потенциометра (диапазон от 0 до 1023)

int potValue = analogRead(potPin);

// Вывод значения на монитор порта

Serial.print("Потенциометр: ");

Serial.println(potValue);

// Задаём ширину каждого сектора потенциометра

int maxled = 1023 / 3;

// Выбор пина в зависимости от значения потенциометра

int ledpin = potValue / maxled;

// Управление светодиодами в зависимости от значения потенциометра

if (ledpin == 0) { // Значение около 0

digitalWrite(ledPin1, HIGH); // Включаем красный

digitalWrite(ledPin2, LOW); // Выключаем зеленый

digitalWrite(ledPin3, LOW); // Выключаем синий

} else if (ledpin == 1) { // Значение около ½

digitalWrite(ledPin2, HIGH); // Включаем зеленый

digitalWrite(ledPin1, LOW); // Выключаем красный

digitalWrite(ledPin3, LOW); // Выключаем синий

} else { // Значение около 1

digitalWrite(ledPin3, HIGH); // Включаем синий

digitalWrite(ledPin1, LOW); // Выключаем красный

digitalWrite(ledPin2, LOW); // Выключаем зеленый

}

delay(50); // Пауза для стабилизации

}

**Результаты**

В ходе работы была успешно реализована система управления светодиодами на основе аналогового сигнала с потенциометра. Программа корректно считывала значения с потенциометра, преобразовывала их и управляла светодиодами в зависимости от положения потенциометра. Значения выводились на монитор последовательного порта, что подтвердило правильность работы программы.

**Список литературы**

1. Официальная документация Arduino (https://docs.arduino.cc/)
2. Блум, Дж. Изучаем Arduino: инструменты и методы технического волшебства. – СПб.: БХВ-Петербург, 2015.
3. Монк, С. Программируем Arduino: основы работы со скетчами. – М.: Вильямс, 2016.
4. Evans, M. Arduino Programming Notebook. – 2007.
5. Arduino Cookbook by Michael Margolis. – O'Reilly Media, 2012.
6. Петин, В. А. Проекты с использованием контроллера Arduino. - 2-е изд. перераб. и доп. - СПБ.: БХВ-Петербург, 2015. - 464 с.: ил. - (Электроника)