

Кокшарова Е.А.

к.п.н., доцент кафедры информационных технологий

Филиал РГППУ в г.Нижнем Тагиле

г. Нижний Тагил, Россия

Колганов Е.А.

студент кафедры информационных технологий

Филиал РГППУ в г.Нижнем Тагиле

г. Нижний Тагил, Россия

МЕТОДИЧЕСКИЕ АСПЕКТЫ РАБОТЫ С ИСПОЛЬЗОВАНИЕМ КОНСТРУКТОРА ARDUINO

Аннотация

Огромную роль в образовательном процессе играет практическое применение знаний. Ведь это позволяет не только закрепить полученные теоретические знания, но и еще больше заинтересовать в получении новых знаний. Одним из способов организации практических занятий по робототехнике и электротехнике, является использование конструктора Arduino. Ведь для успешного выполнения практического задания с использованием Arduino необходимо проявить знания в области электроники и программирования, уметь работать с датчиками, моторами, и другими компонентами. Что позволяет разнообразить учебный процесс.

Ключевые слова: Arduino, схема, среда разработки.

Koksharova E.A.

Ph.D., Associate Professor of the Department of Information Technology

Branch of RGPPU in Nizhny Tagil

Nizhny Tagil, Russia

Kolganov E.A.

Student of the Department of Information Technology

Branch of RGPPU in Nizhny Tagil

Nizhny Tagil, Russia

PRACTICAL APPLICATION OF KNOWLEDGE USING THE ARDUINO CONSTRUCTOR

Annotation

A huge role in the educational process is played by the practical application of knowledge. After all, this allows not only to consolidate the theoretical knowledge gained, but also to interest even more in obtaining new knowledge. One of the ways to organize practical classes in robotics and electrical engineering is to use the Arduino constructor. After all, in order to successfully complete a practical task using Arduino, it is necessary to show knowledge in the field of electronics and programming, be able to work with sensors, motors, and other components. Which allows you to diversify the learning process.

Keywords: Arduino, circuit, development environment.

Конструктор Arduino - это среда разработки, предназначенная для программирования микроконтроллеров. Среда разработки состоит из интерфейса для написания и отладки кода, а так же загрузки его в плату для управления аппаратными компонентами. Программный код Arduino пишется на языке C++.

В образовании Arduino можно использовать для:

Изучения основ электроники и программирования: конструктор Arduino представляет собой плату и различные компоненты к ним, которые соединяются проводами и управляются командами с платы.

Практическая деятельность: сборка и программирование электронных схем на практике, позволяет закрепить полученные теоретические знания.

Проектная деятельность: конструктор предоставляет огромные возможности для создания различных проектов связанных с электроникой.

Arduino удобно использовать в образовательной деятельности, так как он обладает следующими преимуществами:

Масштабируемость — к конструктору создано множество различных

датчиков и контроллеров, которыми можно дополнять уже готовый проект.

Наличие готовых библиотек к датчикам — для подключения различных компонентов разработаны специальные библиотеки, упрощающие написание кода.

Обширная документация — официальные руководства, примеры кода и справочные материалы помогают разработчикам разобраться с платформой и ее возможностями.

Для приобретения начального опыта, мы предлагаем к сборке схемы, которые разобраны ниже.

Светофор

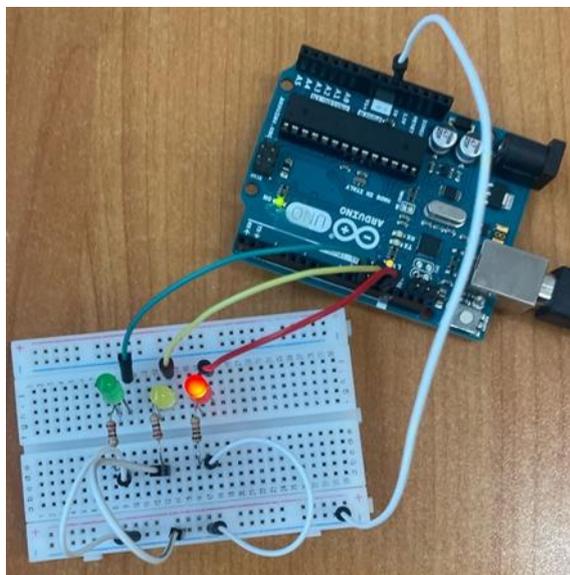


Рис. 1. Сборка схемы светофора

Для сборки светофора понадобятся:

- Набор проводов
- Три резистора
- Три светодиода (красного, желтого, зеленого цвета)
- Плата ARDUINO
- Макетная плата

После подготовки необходимо собрать все компоненты, таким образом, как представлено на рис.1.

После того как схема будет собрана, нужно написать код представленный

в листинге 1 в специальной программе ARDUINO IDE и загрузить код в плату.

Листинг 1 Код для светофора

```
// C++ code
//
void setup()
{
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}
void loop()
{
  digitalWrite(11, HIGH);
  delay(5000);
  digitalWrite(11, LOW);
  digitalWrite(10, HIGH);
  delay(3000);
  digitalWrite(10, LOW);
  digitalWrite(9, HIGH);
  delay(5000);
  digitalWrite(9, LOW);
}
```

После завершения всех шагов получаем сборку, в которой поочередно зажигаются три светодиода, имитируя работу светофора.

Световая дорожка

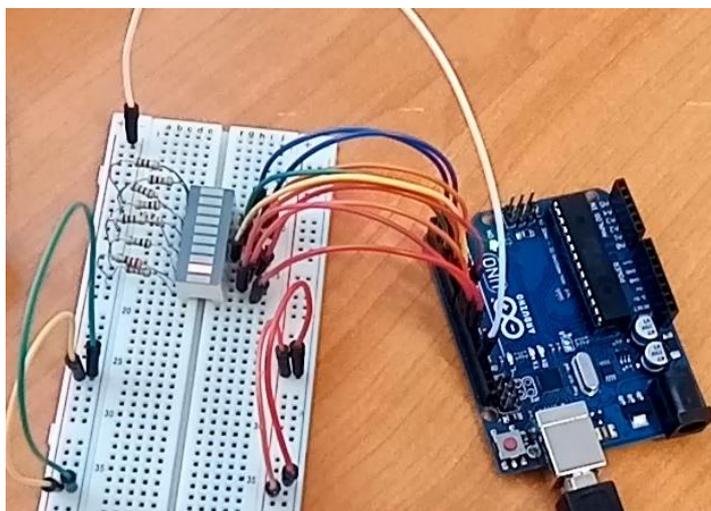


Рис. 2. Сборка схемы световая дорожка

Для данной сборки необходимы следующие компоненты:

- Набор проводов
- Восемь резистора
- Светодиодная шкала
- Плата ARDUINO
- Макетная плата

Собираем все компоненты как показано на рис.2.

Загружаем код, представленный в листинге 2 в плату.

Листинг 2 Код для световой дорожки

```
// переменные верхнего уровня, т.е. объявленные вне функций,  
// называют глобальными. Их значения сохраняются всё время,  
// пока работает микроконтроллер  
int brightness = 0;  
void setup()  
{  
  pinMode(CONTROL_PIN, OUTPUT);  
}  
void loop()  
{  
  // увеличиваем значение яркости на единицу, чтобы нарастить  
  // яркость. Однако яркость не должна быть более 255, поэтому  
  // используем операцию остатка от деления, чтобы при  
  // достижении значения 255, следующим значением снова стал 0  
  // Y % X – это остаток от деления Y на X;  
  // плюс, минус, делить, умножить, скобки – как в алгебре.  
  brightness = (brightness + 1) % 256;  
  // подаём вычисленный ШИМ-сигнал яркости на пин с базой  
  // управляющего транзистора  
  analogWrite(CONTROL_PIN, brightness);  
  // ждём 10 мс перед следующим наращиванием яркости. Таким  
  // образом, полный накал будет происходить в течение  
  // 256×10 = 2560 мс  
  delay(10);  
}
```

В итоге получаем световую дорожку, в которой каждый сегмент загорается поочередно, со временем увеличивая яркость горения отдельного диода.

Датчик уровня воды

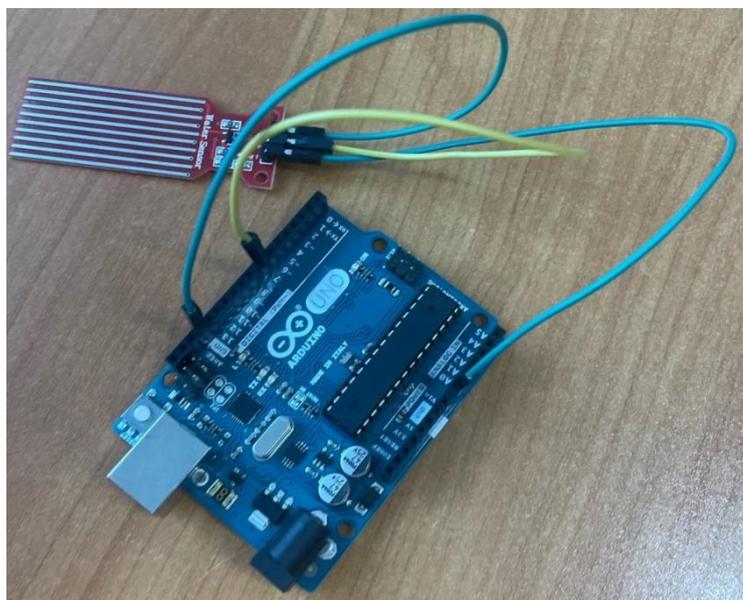


Рис. 3. Сборка датчика уровня воды

Необходимые компоненты:

- Плата ARDUINO
- Набор проводов
- Датчик уровня воды

Сборка схемы проста и не составит проблемы (см рис.3).

Загружаем код, представленный в листинге 3.

Листинг 3 Код для датчика уровня воды

```
// Выводы, подключенные к датчику
#define sensorPower 7
#define sensorPin A0
// Переменная для хранения значения уровня воды
int val = 0;
void setup() {
  // Настраиваем D7 на выход
  pinMode(sensorPower, OUTPUT);
  // Устанавливаем низкий уровень, чтобы на датчик не подавалось питание
  digitalWrite(sensorPower, LOW);
  Serial.begin(9600);
}
void loop()
{
  // получить показания из функции ниже и напечатать его
  int level = readSensor();
  Serial.print("Water level: ");
  Serial.println(level);
  delay(1000);
}
// Данная функция используется для получения показаний int readSensor()
{
  digitalWrite(sensorPower, HIGH);           // Включить датчик
  delay(10);                                 // Ждать 10 миллисекунд
  int val = analogRead(sensorPin);           // Прочитать аналоговое значение
от датчика
  digitalWrite(sensorPower, LOW);           // Выключить датчик
  return val;                                 // Вернуть текущее показание
}
```

Для демонстрации работы датчика уровня воды, необходимо провести небольшой эксперимент. Для него вам понадобятся несколько сосудов, в которые будет налито различное количество воды. Открываем в ARDUINO IDE консоль и погружаем датчик в сосуды с водой. Погружая датчик в сосуды, на консоль будут выдаваться различные значения, что наглядно демонстрирует принцип работы датчика уровня воды.

Метеостанция

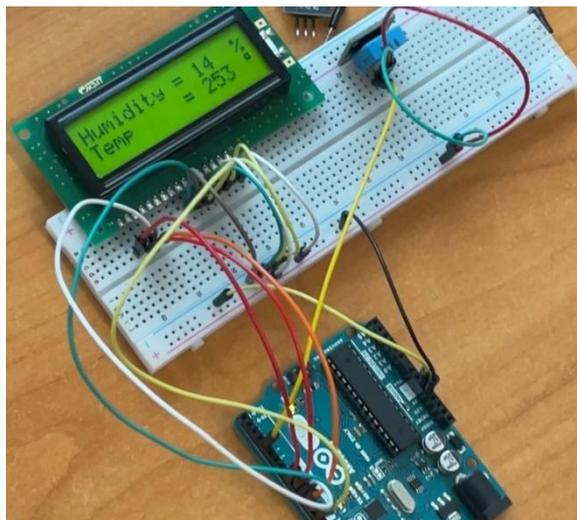


Рис. 4. Сборка метеостанции

Необходимые компоненты:

- Плата ARDUINO
- Набор проводов
- Жидкокристаллический дисплей
- Датчик температуры

Итоговый вид метеостанции представлен на рис.4.

Загружаем код, представленный в листинге 4.

1. Код для метеостанции

```
#include <DHT11.h>
#include <LiquidCrystal.h>
DHT11 dht11(4);
#define DHT11_PIN 4
LiquidCrystal lcd(13,12,11,10,9,8);
byte degree[8] =
{
    B00111,
    B00101,
    B00111,
    B00000,
    B00000,
    B00000
};
void setup() {
    lcd.begin(16,2);
    lcd.createChar(1, degree);
}
void loop() {
    int chk;
    Serial.print(dht11.readHumidity(), 1);
    Serial.print(",\t");
    Serial.println(dht11.readTemperature(), 1);
    lcd.setCursor(0,0);
    lcd.print("Humidity =      % ");
```

```
lcd.setCursor(11, 0);  
lcd.print(dht11.readHumidity(), 1);  
lcd.setCursor(0, 1);  
lcd.print("Temp      =      \1C ");  
}
```

Следует обратить внимание, что в данном коде используется библиотека DHT11.h, которая не является встроенной библиотекой, поэтому ее необходимо скачать с официального сайта ARDIUNO: <https://www.arduino.cc/> и добавить в ARDIUNO IDE при помощи панели «инструменты» - «добавить библиотеку».

После запуска кода на дисплей будет выводиться температура и влажность окружающей среды.

СПИСОК ЛИТЕРАТУРЫ:

1. Гребнева Д. М. РАЗРАБОТКА ИНТЕГРИРОВАННЫХ УРОКОВ ПО ИНФОРМАТИКЕ И ФИЗИКЕ С ИСПОЛЬЗОВАНИЕМ КОНСТРУКТОРА ARDUINO // Наука и перспективы. – 2019. – № 4; URL: nip.esrae.ru/ru/31-254 (дата обращения: 23.11.2023).

2. Григорьев, Е. К. Разработка систем анализа и обработки информации на базе Arduino : учебно-методическое пособие / Е. К. Григорьев, В. А. Ненашев, А. М. Сергеев. — Санкт-Петербург : ГУАП, 2022. — 63 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/263945> (дата обращения: 23.11.2023). — Режим доступа: для авториз. пользователей.