

Практическое занятие №

Тема: «Датчик освещенности на базе фоторезистора»

Цель работы: приобрести практические навыки по подключению и программированию фоторезистора LDR 50mm на платформе Arduino.

Последовательность выполнения работы:

- Собрать схемы на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно заданию в примере.
- Выполнить задание для самостоятельной работы.

Содержание отчета:

- Название практического занятия, его цель.
- Фото или скриншоты собранной схемы.
- Написанный программный код вставить текстом, Courier New, 12 кегль, одинарный отступ без абзацев.
- Вывод о проделанной работе.
- Файл Fritzing с принципиальной и монтажной схемой.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Фоторезистор LDR 50мм (Light Dependent Resistor) - это крупногабаритный фоторезистор диаметром 50 миллиметров, обладающий повышенной чувствительностью к свету благодаря большой площади светочувствительной поверхности.

Фоторезистор работает на основе фотопроводимости - явления изменения электрического сопротивления материала под воздействием светового излучения.

В темноте: Полупроводниковый материал имеет мало свободных носителей заряда → высокое сопротивление (до 1-10 МОм)

На свету: Фотоны света выбивают электроны → появляются свободные носители заряда → сопротивление резко уменьшается (до 100-1000 Ом)



Рисунок 1 – Фоторезистор 50 мм

Основные характеристики LDR 50мм:

- диаметр: 50 мм (большая площадь чувствительности)
- сопротивление в темноте: 1-10 МОм
- сопротивление при освещении: 100-2000 Ом
- максимальное напряжение: 150-200В
- мощность рассеяния: 100-200 мВт
- спектральная чувствительность: 400-700 нм (видимый свет)
- время спада: 2.5-5 секунд
- время восстановления: 10-20 секунд

Области применения

1. Системы автоматического освещения
2. Метеорологические станции
 - измерение солнечной радиации
 - обнаружение облачности
3. Системы безопасности
 - обнаружение вторжения по изменению освещенности
 - контроль работы осветительных приборов
4. Фотографическая техника
 - экспонометры
 - автоматическая регулировка диафрагмы

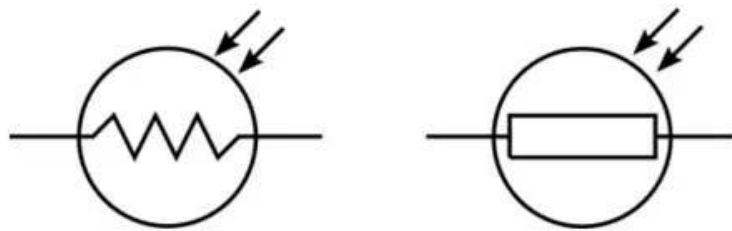


Рисунок 2 – Изображение на принципиальной схеме

ЗАДАНИЕ

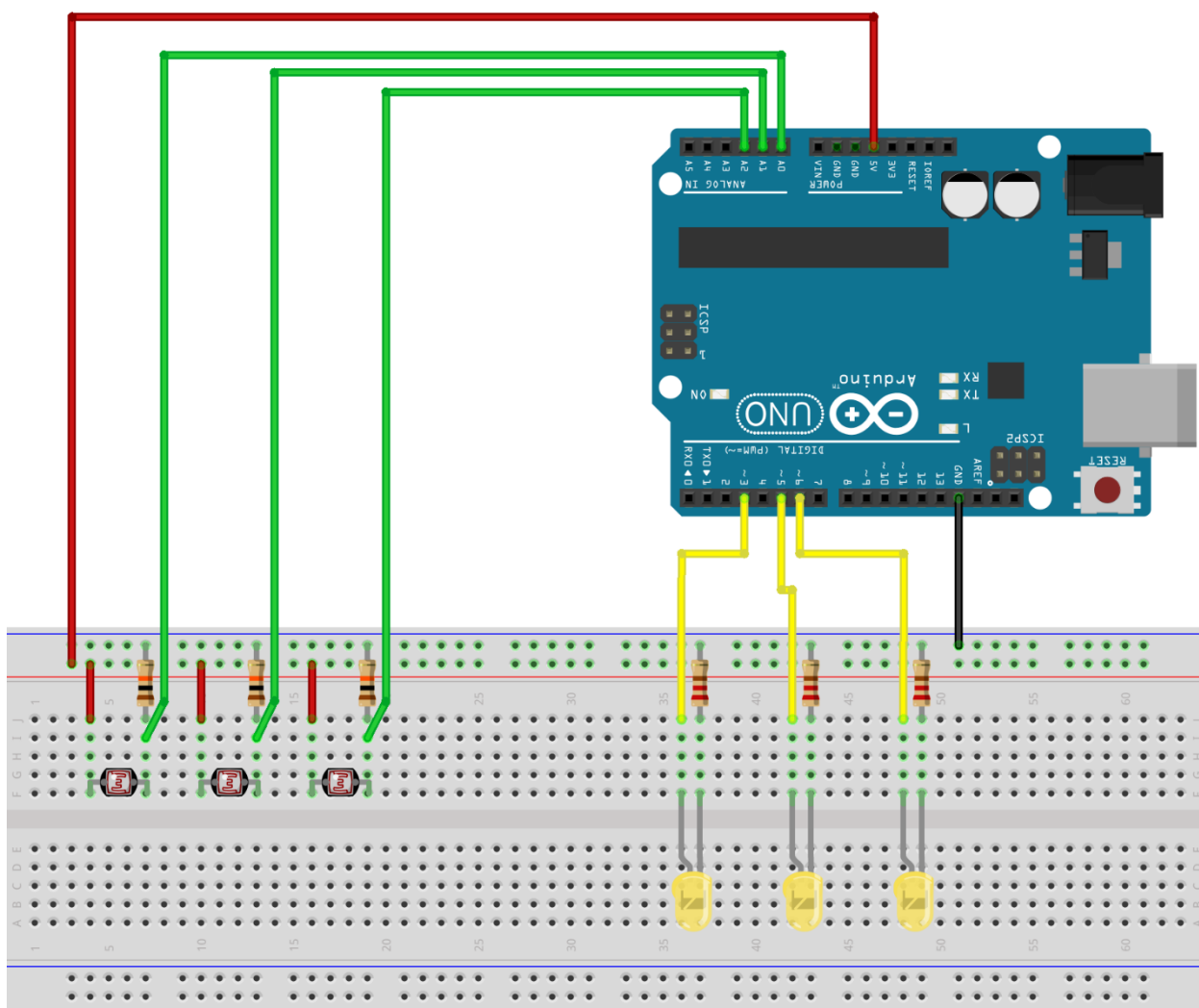


Рисунок 3 – Схема подключения

Программный код №1. Включение света по значению:

```
// Пины для фоторезисторов
#define PHOTO_RESISTOR_1 A0
#define PHOTO_RESISTOR_2 A1
#define PHOTO_RESISTOR_3 A2

// Пины для светодиодов
#define LED_1 3
#define LED_2 5
#define LED_3 6

// Пороговые значения освещенности (можно настроить под ваше
освещение)
int threshold1 = 500; // Порог для первого датчика
int threshold2 = 500; // Порог для второго датчика
int threshold3 = 500; // Порог для третьего датчика

// Переменные для хранения значений с датчиков
int lightValue1 = 0;
int lightValue2 = 0;
int lightValue3 = 0;

void setup() {
  // Настройка пинов светодиодов как выходы
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);
  pinMode(LED_3, OUTPUT);

  // Настройка последовательного порта для отладки
  Serial.begin(9600);

  Serial.println("Система контроля освещенности запущена!");
  Serial.println("=====");
}

void loop() {
  // Чтение значений с фоторезисторов
  lightValue1 = analogRead(PHOTO_RESISTOR_1);
  lightValue2 = analogRead(PHOTO_RESISTOR_2);
  lightValue3 = analogRead(PHOTO_RESISTOR_3);
```

```
// Управление первым светодиодом
if (lightValue1 < threshold1) {
    digitalWrite(LED_1, HIGH); // Включить светодиод при
недостаточном освещении
} else {
    digitalWrite(LED_1, LOW); // Выключить светодиод при
достаточном освещении
}

// Управление вторым светодиодом
if (lightValue2 < threshold2) {
    digitalWrite(LED_2, HIGH);
} else {
    digitalWrite(LED_2, LOW);
}

// Управление третьим светодиодом
if (lightValue3 < threshold3) {
    digitalWrite(LED_3, HIGH);
} else {
    digitalWrite(LED_3, LOW);
}

// Вывод значений в монитор порта для отладки
Serial.print("Датчик 1: ");
Serial.print(lightValue1);
Serial.print(" | Датчик 2: ");
Serial.print(lightValue2);
Serial.print(" | Датчик 3: ");
Serial.println(lightValue3);

// Небольшая задержка для стабильности
delay(200);
}

// Функция для калибровки порогов (вызвать один раз в setup
если нужно)
void calibrateThresholds() {
    Serial.println("Калибровка... Замерьте освещение в
нормальных условиях");
    delay(3000);
}
```

```

threshold1 = analogRead(PHOTO_RESISTOR_1);
threshold2 = analogRead(PHOTO_RESISTOR_2);
threshold3 = analogRead(PHOTO_RESISTOR_3);

Serial.println("Калибровка завершена!");
Serial.print("Новые пороги: ");
Serial.print(threshold1);
Serial.print(", ");
Serial.print(threshold2);
Serial.print(", ");
Serial.println(threshold3);
}

```

Программный код №2. Динамическое освещение:

```

// Пины для фоторезисторов
#define PHOTO_RESISTOR_1 A0
#define PHOTO_RESISTOR_2 A1
#define PHOTO_RESISTOR_3 A2

// Пины для светодиодов (должны поддерживать ШИМ ~)
#define LED_1 3
#define LED_2 5
#define LED_3 6

// Минимальные и максимальные значения освещенности
int minLight = 200; // Минимальное значение (темно)
int maxLight = 800; // Максимальное значение (светло)

void setup() {
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);
  pinMode(LED_3, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // Чтение значений с датчиков
  int light1 = analogRead(PHOTO_RESISTOR_1);
  int light2 = analogRead(PHOTO_RESISTOR_2);
  int light3 = analogRead(PHOTO_RESISTOR_3);
}

```

```
// Преобразование значений освещенности в яркость
светодиодов
// Чем темнее - тем ярче светодиод
int brightness1 = map(light1, minLight, maxLight, 255, 0);
int brightness2 = map(light2, minLight, maxLight, 255, 0);
int brightness3 = map(light3, minLight, maxLight, 255, 0);

// Ограничение значений яркости
brightness1 = constrain(brightness1, 0, 255);
brightness2 = constrain(brightness2, 0, 255);
brightness3 = constrain(brightness3, 0, 255);

// Установка яркости светодиодов
analogWrite(LED_1, brightness1);
analogWrite(LED_2, brightness2);
analogWrite(LED_3, brightness3);

// Вывод отладочной информации
Serial.print("Яркости: ");
Serial.print(brightness1);
Serial.print(", ");
Serial.print(brightness2);
Serial.print(", ");
Serial.println(brightness3);

delay(100);
}
```