

## Практическое занятие №

### Тема: Подключение и программирование электромагнитного реле

**Цель работы:** приобрести практические навыки по подключению и программированию электромагнитного реле 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

### *Электромагнитное реле*

Модуль реле (рисунок 1), входящий в состав ARDUINO SENSOR KIT, предназначен для управления с помощью микроконтроллера цепью с мощной нагрузкой или переменным током. Модуль реле SRD-05VDC-SL-C позволяет управлять электрическими цепями с переменным током до 250 Вольт и нагрузкой до 10 Ампер.

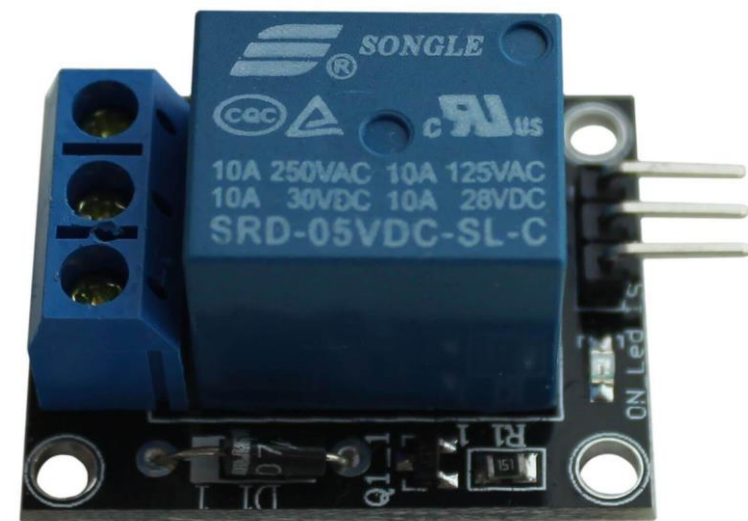


Рисунок 1 – Модуль реле

Таблица 1 – Технические характеристики реле SRD-05VDC-SL-C.

Параметр	Значение
Номинальное рабочее напряжение	5 В
Номинальный рабочий ток	71 мА
Сопротивление	70 Ом
Потребляемая мощность	0.36 Вт
Максимальный коммутируемый ток	10 А
Максимальное коммутируемое напряжение (постоянный ток)	28 В
Максимальное коммутируемое напряжение (переменный ток)	250 В
Рабочая температура	от -25 °С до +70 °С

Контакты модуля реле изображены на рисунке 2.



Рисунок 2 – Контакты модуля реле

*Для его подключения потребуются:*

- плата Arduino Uno / Arduino Nano / Arduino Mega;
- блок питания 12 Вольт;
- провода типа «папа-папа» и «папа-мама»;
- бытовой прибор (лампа накаливания / светодиодная лента и т.п.)

Схема подключения модуля реле с использованием бытового прибора (лампы накаливания) представлена на рисунке 3.

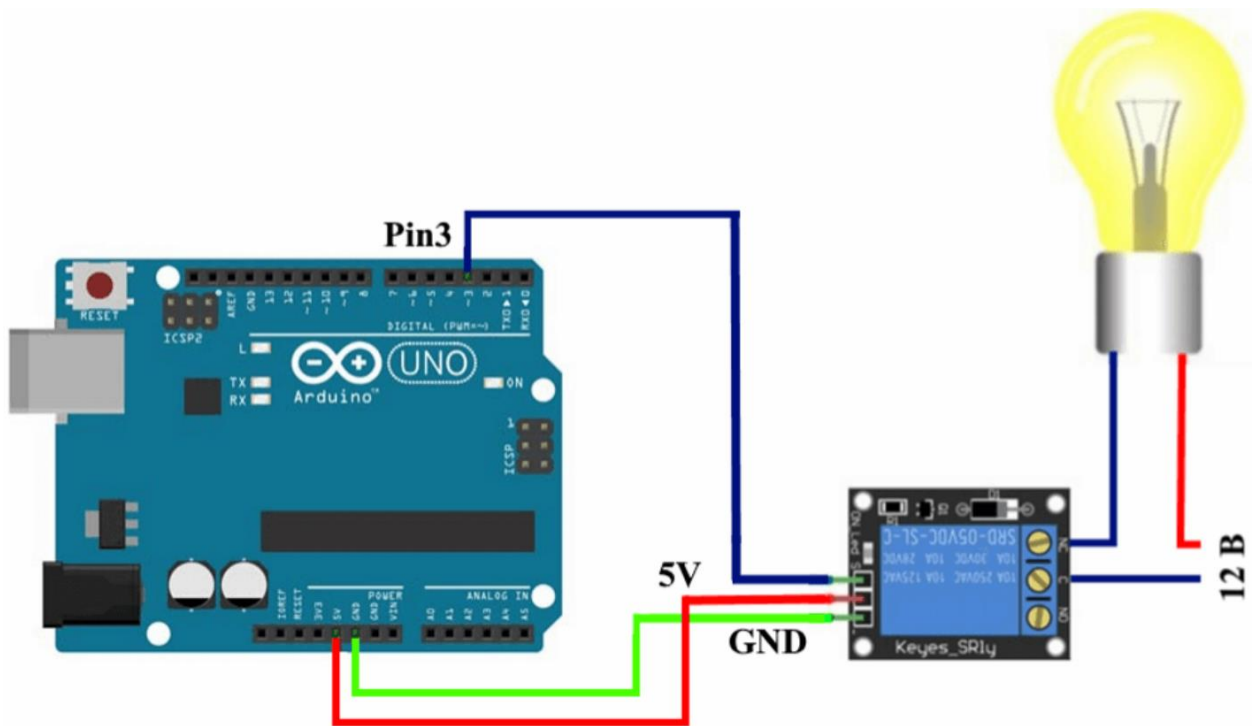


Рисунок 3 – Подключение прибора (лампы накаливания 12В) с помощью модуля реле к Arduino UNO.

## ЗАДАНИЯ

### ЗАДАНИЕ 1: ВКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ РЕЛЕ

**Описание:** Реле циклически включается и выключается с интервалом 30 секунд.

**Схема:**

Реле: Pin 3 — IN

GND — GND

VCC — 5V

```
const int relayPin = 3;           // Пин управления реле
bool relayState = false;         // Текущее состояние реле
unsigned long previousMillis = 0; // Время последнего переключения
const long interval = 30000;     // Интервал 30 секунд (в миллисекундах)

void setup() {
  pinMode(relayPin, OUTPUT);     // Настраиваем пин реле как ВЫХОД
  digitalWrite(relayPin, LOW);   // Изначально выключаем реле
}

void loop() {
```

```

unsigned long currentMillis = millis(); // Текущее время

// Проверяем, прошло ли 30 секунд
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; // Сбрасываем таймер
    relayState = !relayState;      // Меняем состояние реле
    digitalWrite(relayPin, relayState ? HIGH : LOW); //
Управляем реле
}
}

```

## ЗАДАНИЕ 2: УПРАВЛЕНИЕ РЕЛЕ КНОПКОЙ С ЗАДЕРЖКАМИ

**Описание:** При нажатии кнопки: реле включается через 3 секунды на 1 минуту. При зажатии кнопки на 3 секунды: реле выключается через 3 секунды.

### Схема:

Реле: Pin 3 — IN

Кнопка: Pin 2 — GND

```

const int relayPin = 3;          // Пин реле
const int buttonPin = 2;        // Пин кнопки

bool relayState = false;        // Состояние реле
bool buttonActive = false;      // Факт нажатия кнопки
bool longPressActive = false;   // Флаг длительного нажатия

unsigned long timerStart = 0;    // Время начала таймера
const long delayTime = 3000;    // Задержка 3 секунды
const long onTime = 60000;      // Время включения 1 минута

void setup() {
    pinMode(relayPin, OUTPUT);
    pinMode(buttonPin, INPUT_PULLUP); // Подтяжка к +5V
    digitalWrite(relayPin, LOW);
}

void loop() {
    unsigned long currentMillis = millis();

    // Обработка нажатия кнопки
    if (digitalRead(buttonPin) == LOW) { // Кнопка нажата
        if (!buttonActive) {
            buttonActive = true;
            timerStart = currentMillis; // Запуск таймера
        }

        // Проверка длительного нажатия (3 секунды)
        if (buttonActive && !longPressActive && (currentMillis -
timerStart >= delayTime)) {

```

```

        longPressActive = true;
        timerStart = currentMillis; // Сброс таймера для
ВЫКЛЮЧЕНИЯ
    }
} else { // Кнопка отпущена
    if (buttonActive) {
        // Короткое нажатие: запуск таймера включения
        if (!longPressActive) {
            timerStart = currentMillis;
            relayState = true; // Реле включится через 3 секунды
        }
        buttonActive = false;
        longPressActive = false;
    }
}

// Логика управления реле
if (relayState) {
    // Включение реле через 3 секунды
    if (currentMillis - timerStart >= delayTime) {
        digitalWrite(relayPin, HIGH);
        // Автовыключение через 1 минуту
        if (currentMillis - timerStart >= onTime) {
            relayState = false;
            digitalWrite(relayPin, LOW);
        }
    }
}

// Принудительное выключение при длительном нажатии
if (longPressActive && (currentMillis - timerStart >=
delayTime)) {
    relayState = false;
    digitalWrite(relayPin, LOW);
    longPressActive = false;
    buttonActive = false;
}
}
}

```

### ЗАДАНИЕ 3: РЕЛЕ С ЦИФРОВОЙ ИНДИКАЦИЕЙ

**Описание:** Добавляем визуализацию задержек на 7-сегментном индикаторе с точкой.

**Схема:**

Индикатор (общий катод): Пины a-g → Arduino 4-10, точка → 11.

Реле и кнопка как в задании 2.

```

// Пины реле и кнопки
const int relayPin = 3;
const int buttonPin = 2;

```

```

// Пины индикатора (a, b, c, d, e, f, g, dp)
const int segPins[] = {4, 5, 6, 7, 8, 9, 10, 11};

// Цифры 0-9 и символы для анимации
byte digits[10] = {
  B11111100, // 0
  B01100000, // 1
  B11011010, // 2
  B11110010, // 3
  B01100110, // 4
  B10110110, // 5
  B10111110, // 6
  B11100000, // 7
  B11111110, // 8
  B11110110 // 9
};

bool relayState = false;
bool buttonActive = false;
bool longPressActive = false;

unsigned long timerStart = 0;
const long delayTime = 3000;
const long onTime = 60000;

void displayDigit(int digit, bool showDot) {
  // Вывод цифры на индикатор
  byte pattern = digits[digit];
  for (int i = 0; i < 7; i++) {
    digitalWrite(segPins[i], bitRead(pattern, 7 - i));
  }
  digitalWrite(segPins[7], showDot ? HIGH : LOW); //
  Управление точкой
}

void clearDisplay() {
  for (int i = 0; i < 8; i++) {
    digitalWrite(segPins[i], LOW);
  }
}

void setup() {
  pinMode(relayPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  for (int i = 0; i < 8; i++) {
    pinMode(segPins[i], OUTPUT);
  }
  digitalWrite(relayPin, LOW);
  clearDisplay();
}

void loop() {
  unsigned long currentMillis = millis();

```

```

unsigned long elapsedTime = currentMillis - timerStart;

// Обработка кнопки
if (digitalRead(buttonPin) == LOW) {
    if (!buttonActive) {
        buttonActive = true;
        timerStart = currentMillis;
    }
    if (buttonActive && !longPressActive && elapsedTime >=
delayTime) {
        longPressActive = true;
        timerStart = currentMillis;
    }
    else {
        if (buttonActive) {
            if (!longPressActive) {
                timerStart = currentMillis;
                relayState = true;
            }
            buttonActive = false;
            longPressActive = false;
        }
    }
}

// Визуализация задержек
if (relayState && elapsedTime < delayTime) {
    // Обратный отсчет до включения: 3, 2, 1
    int countdown = 3 - (elapsedTime / 1000);
    displayDigit(countdown, true); // Показываем цифру с
точкой
} else if (longPressActive && elapsedTime < delayTime) {
    // Обратный отсчет до выключения
    int countdown = 3 - (elapsedTime / 1000);
    displayDigit(countdown, true);
} else {
    clearDisplay();
}

// Управление реле
if (relayState) {
    if (elapsedTime >= delayTime) {
        digitalWrite(relayPin, HIGH);
        if (elapsedTime >= onTime) {
            relayState = false;
            digitalWrite(relayPin, LOW);
            clearDisplay();
        }
    }
}

if (longPressActive && elapsedTime >= delayTime) {
    relayState = false;
    digitalWrite(relayPin, LOW);
}

```

```
    longPressActive = false;
    buttonActive = false;
    clearDisplay();
  }
}
```