

Практическое занятие №

Тема: «Подключение и программирование нескольких кнопок на аналоговом входе»

Цель работы: приобрести практические навыки по подключению и программированию кнопочных массивов на платформе Arduino.

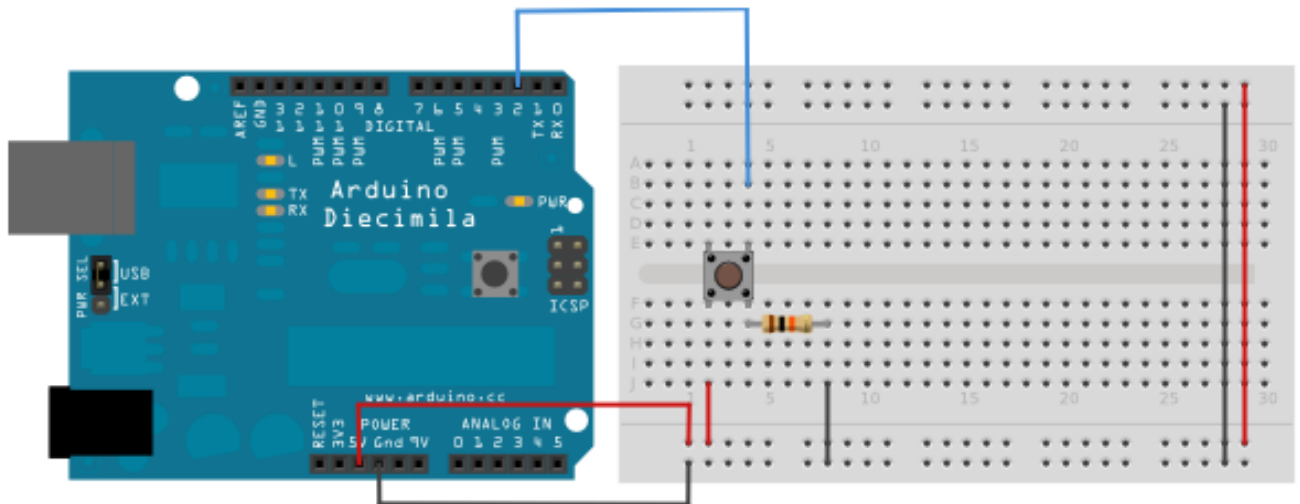
Последовательность выполнения работы:

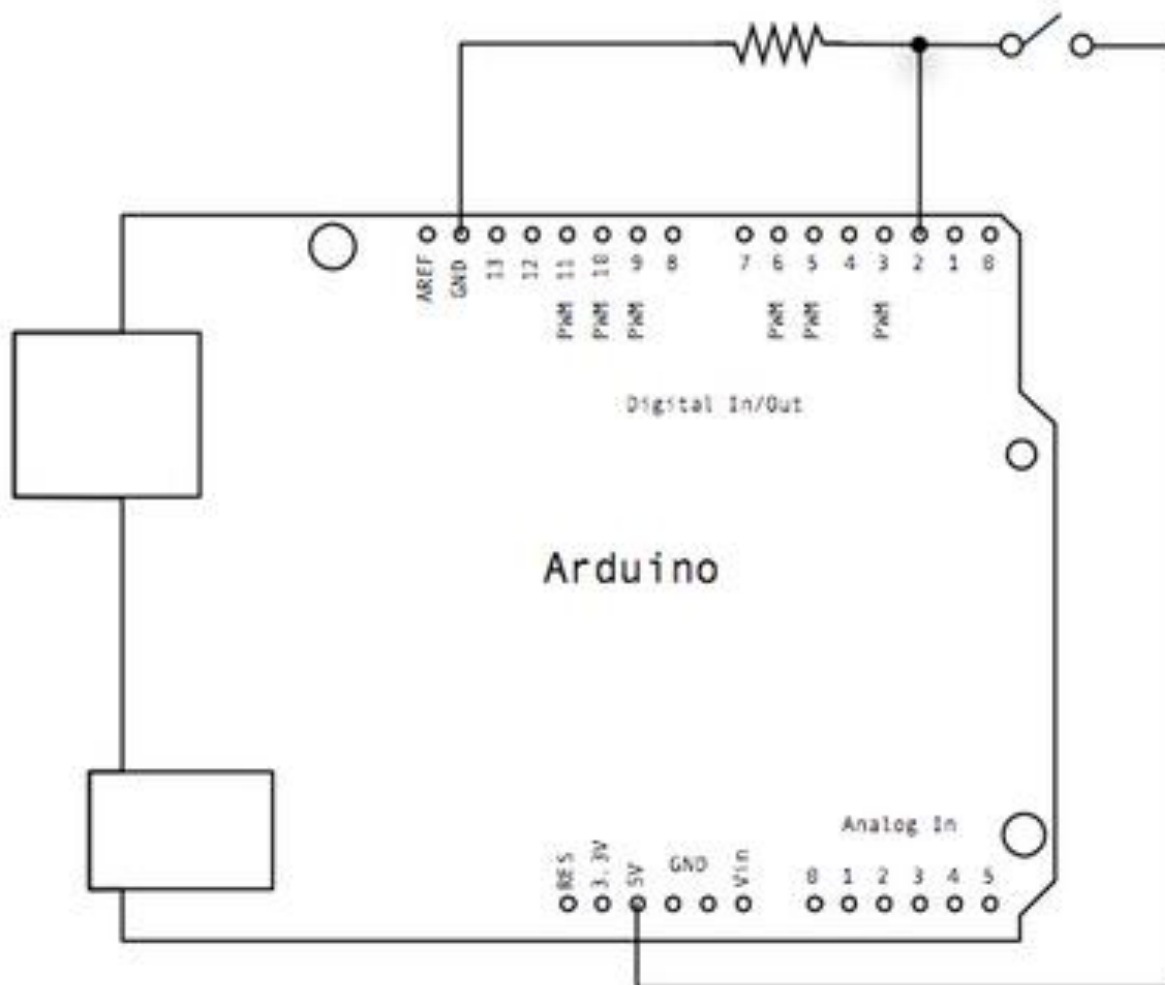
- Собрать схемы на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно заданию в примере.

Содержание отчета:

- название практического занятия, его цель;
- фото или скриншоты собранной схемы;
- написанный программный код вставить текстом;
- вывод о проделанной работе;
- файл Fritzing с принципиальной и монтажной схемой.

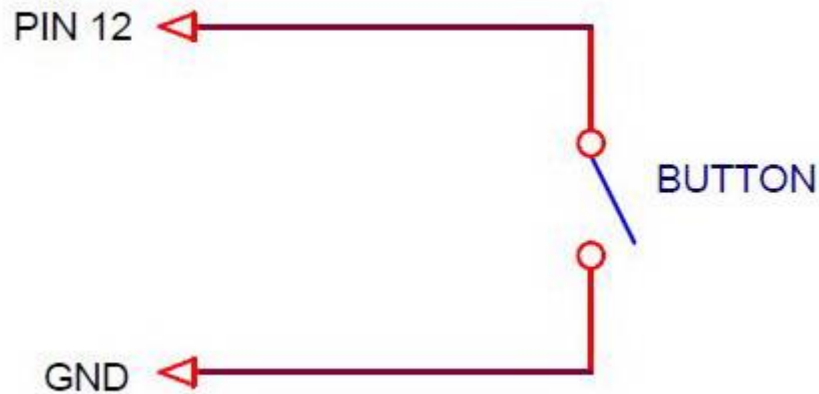
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ





При отжатой кнопке резистор формирует на выводе 5 В, а при нажатой – вход замыкается на землю. Сейчас предложу другой вариант. Все выводы платы имеют внутри контроллера резисторы, подключенные к 5 В. Их можно программно включать или отключать от выводов. Сопротивление этих резисторов порядка 20-50 кОм. Слишком много для реальных схем, но для нашей программы и кнопки, установленной вблизи контроллера, вполне допустимо.

В итоге схема подключения будет выглядеть так.



Для работы с цифровыми выводами в системе Ардуино есть 3 встроенные функции. Они позволяют установить режим вывода, считать или установить вывод в определенное состояние. Для определения состояния выводов в этих функциях используются константы HIGH и LOW, которые соответствуют высокому и низкому уровню сигнала.

`pinMode(pin, mode)`

Устанавливает режим вывода (вход или выход).

Аргументы: `pin` и `mode`.

`pin` – номер вывода;

`mode` – режим вывода.

<code>mode = INPUT</code>	вывод определен как вход, подтягивающий резистор отключен
<code>mode = INPUT_PULLUP</code>	вывод определен как вход, подтягивающий резистор подключен
<code>mode = OUTPUT</code>	вывод определен как выход

Функция не возвращает ничего.

`digitalWrite(pin, value)`

Устанавливает состояние выхода (высокое или низкое).

Аргументы `pin` и `value`:

`pin` – номер вывода;

`value` – состояние выхода.

<code>value = LOW</code>	устанавливает выход в низкое состояние
<code>value = HIGH</code>	устанавливает выход в высокое состояние

Функция не возвращает ничего.

`digitalRead(pin)`

Считывает состояние входа.

Аргументы: `pin` - номер вывода.

Возвращает состояние входа:

<code>digitalRead(pin) = LOW</code>	низкий уровень на входе
<code>digitalRead(pin) = HIGH</code>	высокий уровень на входе

Пример

Программа устанавливает на выводе 13 тот же уровень сигнала, что и на выводе 7.

```
int ledPin = 13; // светодиод подсоединен к цифровому выводу 13
int inPin = 7;   // кнопка подсоединена к цифровому выводу 7
int val = 0;     // переменная для хранения считанного значения

void setup()
{
  pinMode(ledPin, OUTPUT); // конфигурируем цифровой вывод 13 как выход
  pinMode(inPin, INPUT);   // конфигурируем цифровой вывод 13 как вход
}

void loop()
{
  val = digitalRead(inPin); // считываем значение со входа
  digitalWrite(ledPin, val); // выводим на светодиод уровень сигнала на кнопке
}
```

ЗАДАНИЕ

Построить на макетной плате, построить монтажную, принципиальную схему и печатную плату во Fritzing:

! От питания подключен резистор 10кОм, между подключением аналогового входа и кнопкой резисторы 1кОм.

! Перед подключением подключите пустую плату Ардуино

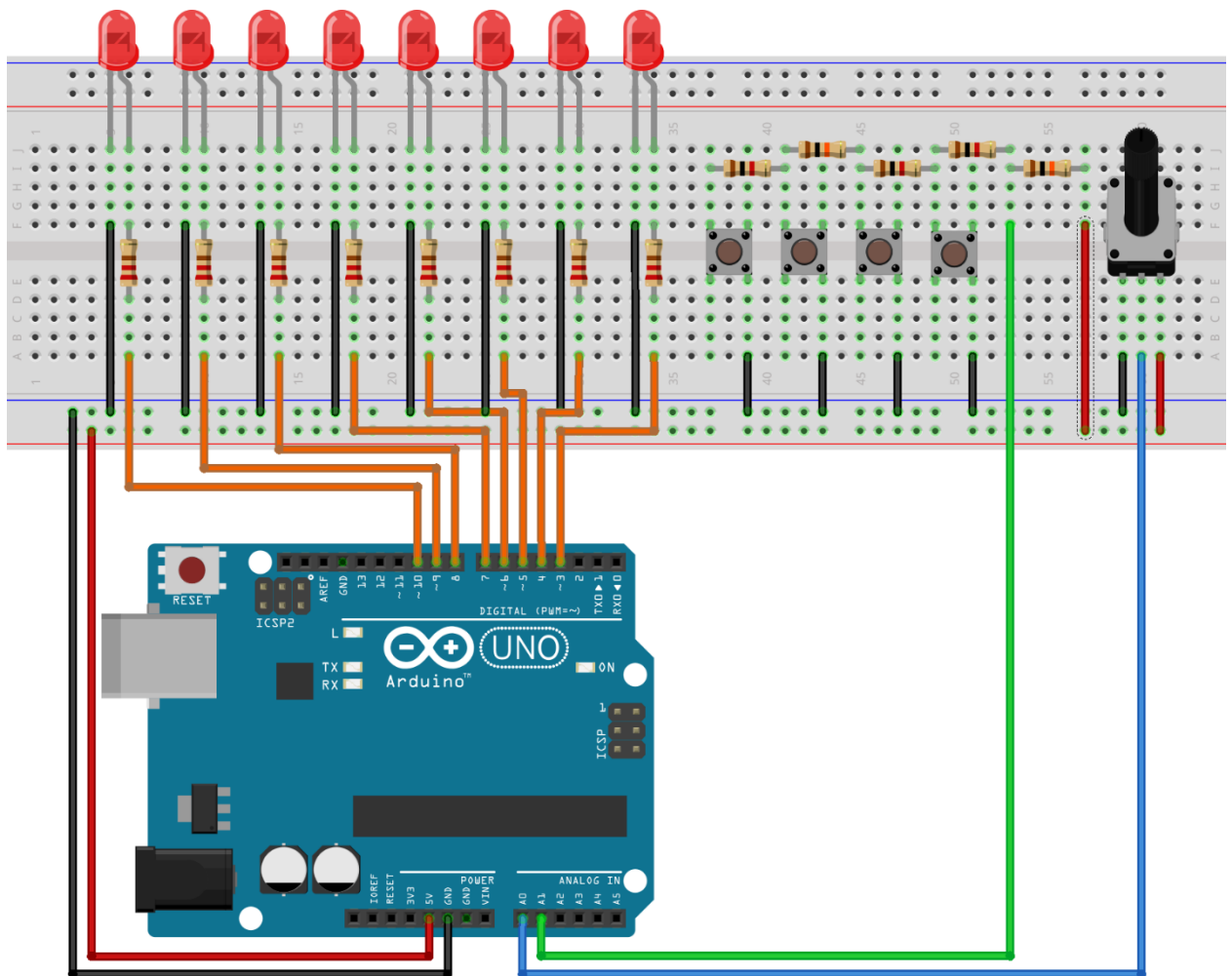


Рисунок 2 – Схема кодключения

Написать программу:

```
// Настройки для калибровки кнопок
const int analogButtonPin = A0; // Аналоговый пин для чтения
значений кнопок
const int analogPotPin = A1; // Аналоговый пин для чтения
значений кнопок

// Пины для светодиодов
const int led1Pin = 3;
const int led2Pin = 4;
const int led3Pin = 5;
const int led4Pin = 6;
const int led5Pin = 7;
const int led6Pin = 8;
const int led7Pin = 9;
const int led8Pin = 10;

// Состояния светодиодов (флаги)
bool led1State = false;
bool led2State = false;
bool led3State = false;
bool led4State = false;
bool led5State = false;
bool led6State = false;
bool led7State = false;
bool led8State = false;

// Значения АЦП для каждой кнопки (ЗНАЧЕНИЯ ТРЕБУЮТ
КАЛИБРОВКИ!)
const int btn1Value = 290; // Кнопка 1 (~1кОм)
const int btn2Value = 234; // Кнопка 2 (~2кОм)
const int btn3Value = 174; // Кнопка 3 (~3кОм)
const int btn4Value = 90; // Кнопка 4 (~4кОм)
const int noPressValue = 1023; // Ничего не нажато (10кОм к
5V)

const int pot1Value = 1000; // Значения потенциометра 1
const int pot2Value = 750; // Значения потенциометра 2
const int pot3Value = 500; // Значения потенциометра 3
const int pot4Value = 250; // Значения потенциометра 4
const int noPressPot = 0; // Ничего не нажато (10кОм к 5V)

// Допуск для сравнения значений
```

```

const int tolerance = 25;

// Переменные для обработки дребезга
int lastButtonState = 0; // 0 - ничего не нажато, 1-4 -
номер кнопки
unsigned long lastDebounceTime = 0;
const unsigned long debounceDelay = 50; // время подавления
дребезга в мс

void setup() {
  Serial.begin(9600); // Инициализация последовательного
порта для отладки

  // Настройка пинов светодиодов как выходов
  for (int i = 3; i < 11; i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }

  Serial.println("Система инициализирована. Готов к
работе!");
  Serial.println("Нажмите кнопки 1-4 или поверните вал
потенциометра для переключения светодиодов");
}

void loop() {
  int buttonValue = analogRead(analogButtonPin); // Читаем
значение с A0 (0-1023)
  int potValue = analogRead(analogPotPin); // Читаем
значение с A1 (0-1023)

  int currentButtonState = getPressedButton(buttonValue); //
Определяем, какая кнопка нажата
  int currentPotState = getdrivePot(potValue); //
Определяем, какая кнопка нажата

  // Вывод для калибровки и отладки
  Serial.println(analogRead(analogButtonPin));
  Serial.println(analogRead(analogPotPin));
  Serial.println(getPressedButton(buttonValue));
  Serial.println(getdrivePot(potValue));

  // Обработка дребезга контактов
  if (currentButtonState != lastButtonState) {

```

```

    lastDebounceTime = millis(); // Сбрасываем таймер при
изменении состояния
}

// Переключаем соответствующий светодиод кнопкой
switch (currentButtonState) {
  case 1:
    led1State = !led1State; // Инвертируем состояние
    digitalWrite(led1Pin, led1State ? HIGH : LOW);
    Serial.print("Кнопка 1: Светодиод 1 ");
    Serial.println(led1State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
    break;
  case 2:
    led2State = !led2State; // Инвертируем состояние
    digitalWrite(led2Pin, led2State ? HIGH : LOW);
    Serial.print("Кнопка 2: Светодиод 2 ");
    Serial.println(led2State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
    break;
  case 3:
    led3State = !led3State; // Инвертируем состояние
    digitalWrite(led3Pin, led3State ? HIGH : LOW);
    Serial.print("Кнопка 3: Светодиод 3 ");
    Serial.println(led3State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
    break;
  case 4:
    led4State = !led4State; // Инвертируем состояние
    digitalWrite(led4Pin, led4State ? HIGH : LOW);
    Serial.print("Кнопка 4: Светодиод 4 ");
    Serial.println(led4State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
    break;
}

// Переключаем соответствующий светодиод кнопкой
switch (currentPotState) {
  case 1:
    led5State = !led5State; // Инвертируем состояние
    digitalWrite(led5Pin, led5State ? HIGH : LOW);
    Serial.print("Кнопка 5: Светодиод 5 ");
    Serial.println(led5State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
    break;
  case 2:
    led6State = !led6State; // Инвертируем состояние
    digitalWrite(led6Pin, led6State ? HIGH : LOW);
    Serial.print("Кнопка 6: Светодиод 6 ");

```

```

        Serial.println(led6State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
        break;
    case 3:
        led7State = !led7State; // Инвертируем состояние
        digitalWrite(led7Pin, led7State ? HIGH : LOW);
        Serial.print("Кнопка 7: Светодиод 7 ");
        Serial.println(led7State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
        break;
    case 4:
        led8State = !led8State; // Инвертируем состояние
        digitalWrite(led8Pin, led8State ? HIGH : LOW);
        Serial.print("Кнопка 8: Светодиод 8 ");
        Serial.println(led8State ? "ВКЛЮЧЕН" : "ВЫКЛЮЧЕН");
        break;
}

// Выводим общее состояние всех светодиодов
printAllLEDsStatus();

delay(500); // Задержка для стабильности, после калибровки
изменить на значение 10-20
}

// Функция для определения нажатой кнопки по аналоговому
значению
int getPressedButton(int analogVal) {
    // Проверяем, попадает ли значение в диапазон для каждой
    кнопки
    if (analogVal >= (btn1Value - tolerance) && analogVal <=
(btn1Value + tolerance)) {
        return 1;
    } else if (analogVal >= (btn2Value - tolerance) &&
analogVal <= (btn2Value + tolerance)) {
        return 2;
    } else if (analogVal >= (btn3Value - tolerance) &&
analogVal <= (btn3Value + tolerance)) {
        return 3;
    } else if (analogVal >= (btn4Value - tolerance) &&
analogVal <= (btn4Value + tolerance)) {
        return 4;
    } else if (analogVal >= (noPressValue - tolerance)) {
        return 0; // Ни одна кнопка не нажата
    } else {
        return -1; // Значение не распознано (шум или ошибка)
    }
}

```

```

    }
}

// Функция для определения аналогового значения
потенциометра при прокрутке
int getdrivePot(int analogVal) {
    // Проверяем, попадает ли значение в диапазон для каждой
    кнопки
    if (analogVal >= (pot1Value - tolerance) && analogVal <=
        (pot1Value + tolerance)) {
        return 1;
    } else if (analogVal >= (pot2Value - tolerance) &&
        analogVal <= (pot2Value + tolerance)) {
        return 2;
    } else if (analogVal >= (pot3Value - tolerance) &&
        analogVal <= (pot3Value + tolerance)) {
        return 3;
    } else if (analogVal >= (pot4Value - tolerance) &&
        analogVal <= (pot4Value + tolerance)) {
        return 4;
    } else if (analogVal >= (noPressPot - tolerance)) {
        return 0; // Ни одна кнопка не нажата
    } else {
        return -1; // Значение не распознано (шум или ошибка)
    }
}
}

```

```

// Функция для вывода статуса всех светодиодов
void printAllLEDsStatus() {
    Serial.println("--- Текущее состояние светодиодов ---");
    Serial.print("LED1: ");
    Serial.print(led1State ? "ВКЛ" : "ВЫКЛ");
    Serial.print(" | LED2: ");
    Serial.print(led2State ? "ВКЛ" : "ВЫКЛ");
    Serial.print(" | LED3: ");
    Serial.print(led3State ? "ВКЛ" : "ВЫКЛ");
    Serial.print(" | LED4: ");
    Serial.println(led4State ? "ВКЛ" : "ВЫКЛ");
    Serial.print("LED5: ");
    Serial.print(led5State ? "ВКЛ" : "ВЫКЛ");
    Serial.print(" | LED6: ");
    Serial.print(led6State ? "ВКЛ" : "ВЫКЛ");
    Serial.print(" | LED7: ");
    Serial.print(led7State ? "ВКЛ" : "ВЫКЛ");
}

```

```
Serial.print(" | LED8: ");  
Serial.println(led8State ? "ВКЛ" : "ВЫКЛ");  
Serial.println("-----");  
}
```