

**Цель:** формирование навыков измерения характеристик цепи; создание и использование измерительных приборов.

Основным применением аналоговых входов Arduino является чтение сигналов датчиков. Также аналоговые входы могут использоваться как цифровые входы/выходы, при необходимости подключения большого количества модулей.



Благодаря встроенному АЦП (аналого-цифровой преобразователь), данные входы могут считывать напряжение подаваемое на них. Микроконтроллеры Atmega 328, используемые в Arduino UNO, содержат шестиканальный АЦП, разрешение которого составляет 10 бит. Это позволяет на выходе получать значения от 0 до 1023 (всего 1024 градаций).



*Функции для работы с аналоговыми сигналами:*

- **analogRead()** Функция считывает значение с указанного аналогового входа. Большинство плат Arduino имеют 6 каналов с 10-битным аналого-цифровым преобразователем (АЦП). Напряжение поданное на аналоговый вход, обычно от 0 до 5 вольт будет преобразовано в значение от 0 до 1023, это 1024 шага с разрешением 0.0049 Вольт. Считывание значение с аналогового входа занимает примерно 100 микросекунд (0.0001 сек), т.е. максимальная частота считывания приблизительно 10,000 раз в секунду.

- **analogReference(type)** Функция определяет опорное напряжение относительно которого происходят аналоговые измерения. Параметр type: определяет используемое опорное напряжение (DEFAULT, INTERNAL или EXTERNAL).

- DEFAULT: стандартное опорное напряжение 5 В (на платформах с напряжением питания 5 В) или 3.3 В (на платформах с напряжением питания 3.3 В)
- INTERNAL: встроенное опорное напряжение 1.1 В на микроконтроллерах ATmega168 и ATmega328, и 2.56 В на ATmega8.
- INTERNAL1V1: встроенное опорное напряжение 1.1 В (Arduino Mega)
- INTERNAL2V56: встроенное опорное напряжение 2.56 (Arduino Mega)
- EXTERNAL: внешний источник опорного напряжения, подключенный к выводу AREF

## Оборудование



Кабель USB



Макетная плата



Платформа Arduino



Резистор номиналом  
220 Ом (1 шт)



Светодиод

## Упр.1 Измерение постоянного напряжения (0..20V)

∞ Создадим и загрузим в Arduino скетч для измерения при помощи аналогового входа A0 напряжения до 5В.

- Запустите среду программирования Arduino IDE.
- Создайте и сохраните новый проект под названием **Volt5.ino**.
- Объявим переменные **u** и **u\_in**.

```
int u = 0; // переменная для числа преобразования 0-1023
float u_in = 0.0; // переменная для напряжения 0.0-5.0
```

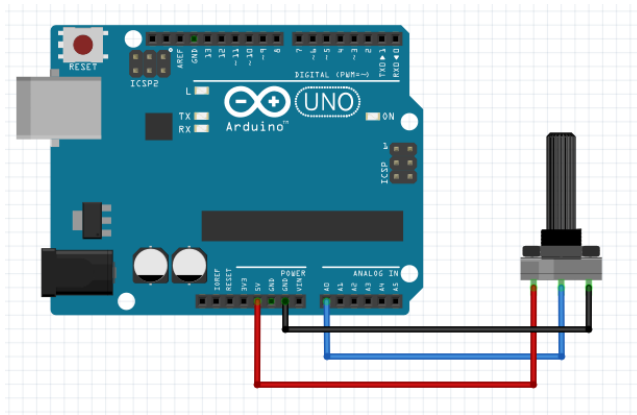
- В области описания функции **Setup()** инициализируем последовательный порт, для передачи данных с платы на компьютер и определим аналоговых вход A0 . Чтобы установить параметры передачи, мы используем функцию **Serial.begin()**, которая принимает один или два параметра. Первым параметром является скорость передачи данных, второй параметр – параметр порта.

```
void setup()
{
  Serial.begin(9600);
  pinMode(A0,INPUT);// A0 - аналоговый вход
}
```

- Подсчитаем и выведем на монитор порта значения напряжения. Впишите в область описания функции **Loop()** следующий код:

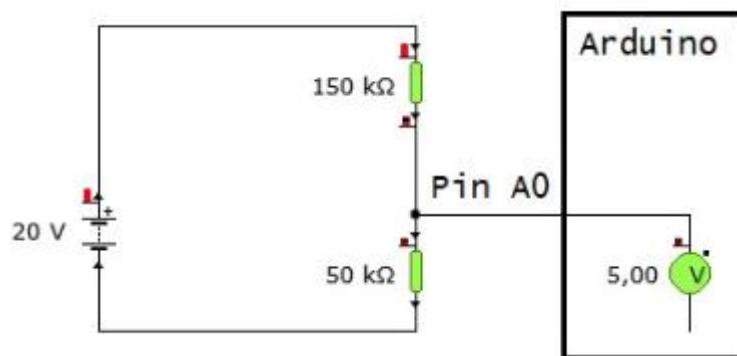
```
void loop() {
  u = analogRead(A0);
  u_in = (u*4.77) / 1023; // Пересчет значений на входе A0
  Serial.println("Напряжение на входе A0 : ",u_in);
  delay(500);
}
```

- Для загрузки программы в плату, используем комбинацию клавиш **Ctrl+U**.
- Проверим работу написанной программы, измеряя напряжение подаваемое на вход питания платы Arduino. Подключим резистор переменного сопротивления ко входу питания **Vin** и аналоговому входу **A0**.



- Откроем монитор последовательного порта (Ctrl+Shift+M) и вращением ручки потенциометра проследим за изменением напряжения..

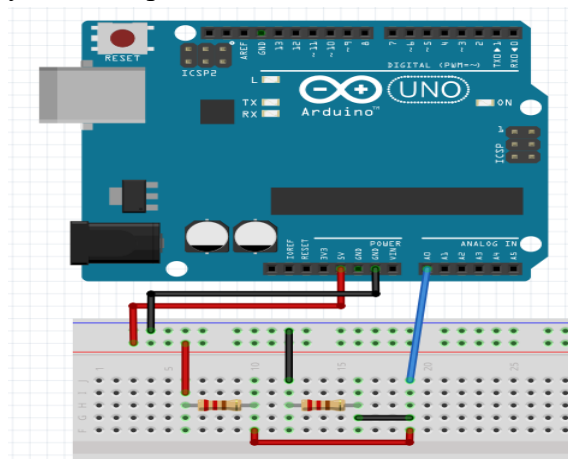
При помощи делителя напряжения на двух резисторах можно повысить предел измеряемого напряжения до 20V.



## Упр.2 Измерение сопротивления при помощи аналогового входа Arduino

Известно, что падение напряжения на резисторе зависит от его сопротивления. Основываясь на этом построим модель омметра на основе делителя напряжения.

- Соберем схему делителя напряжения с двумя резисторами. Левый резистор 10кОм, а сопротивление правого будем измерять.



- Запустите среду программирования Arduino IDE.
- Создайте и сохраните новый проект под названием **OmmMetr.ino**.
- Запишем код программы для вычисления сопротивления правого резистора.

```

OmmMetr | Arduino 1.8.9 (Windows Store 1.8.21.0)
Файл Правка Скетч Инструменты Помощь

float Vout = 0; // Переменная для хранения значения напряжения в средней точки делителя (0-5.0)

float R2 = 0; // Переменная для хранения значения резистора R2

void setup()
{
  Serial.begin(9600); // Подготовка Serial Monitor для вывода информации
}

void loop()
{
  Vout = (5.0 / 1023.0) * analogRead(analogPin); // Вычисляем напряжение в средней точки делителя
  R2 = 10000 / ((5.0 / Vout) - 1); // Вычисляем сопротивление R2 (10000 это значение R1 10 кОм)
  Serial.print("Voltage: "); //
  Serial.println(Vout); // Напряжения в средней точки делителя (0-5.0) для справки
  Serial.print("R2: "); //
  Serial.println(R2); // Значение сопротивления R2

  delay(1000); // Пауза 1 сек
}

```

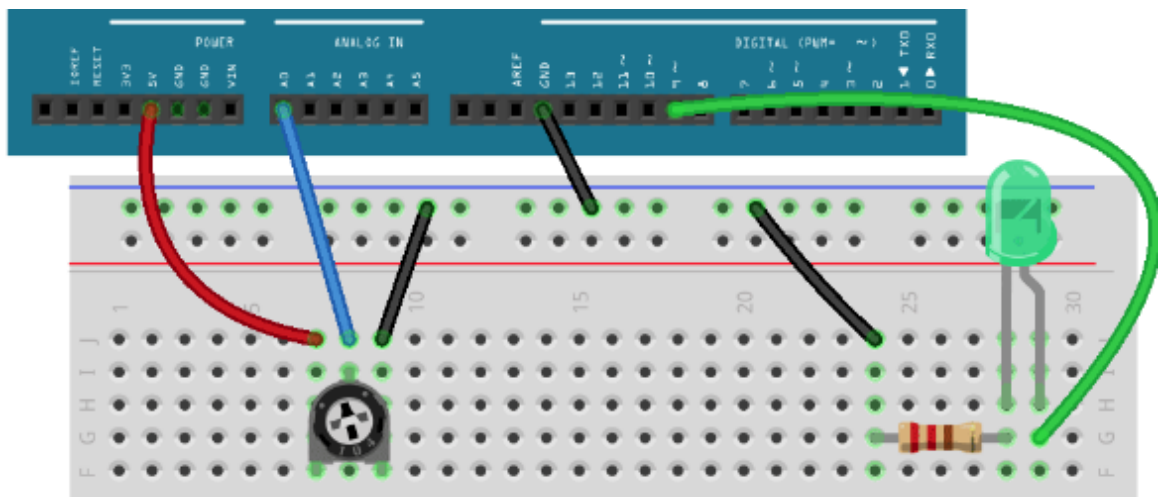
Сохранено.

- Загрузите программу в плату Arduino и измерьте сопротивление предложенных резисторов с заклеенной маркировкой.

### Упр.3 Вывод аналогового сигнала.

∞ В данном упражнении, мы считаем данные с потенциометра и распределим их в интервале от 0 до 255 с помощью функции **map()**. Данный интервал удобно использовать для выводов с PWM (ШИМ). Применим получаемые значения для управления яркостью светодиода, а также будем выводить информацию в **Serial Monitor**.

- Соберем схему представленную на рисунке



- Создадим новую программу с названием AnalogOut.ino.
- Напишите следующий код программы. В данной программе используется функция **map(value, fromLow, fromHigh, toLow, toHigh)**, которая переносит значение (value) из текущего диапазона значений (fromLow .. fromHigh) в новый диапазон (toLow .. toHigh), заданный параметрами. Следует отметить, что функция map не ограничивает значение рамками диапазона, для этого используется функция constrain().

```
const int analogInPin = A0; // потенциометр к A0
const int analogOutPin = 9; // светодиод на выводе 9

int sensorValue = 0; // значения от потенциометра
int outputValue = 0; // значения для PWM

void setup() {
  Serial.begin(9600);
}

void loop() {
  // считываем данные из потенциометра
  sensorValue = analogRead(analogInPin);
  // распределяем данные из интервала 0-1023 в интервал 0-255
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // подаём сигнал на светодиод:
  analogWrite(analogOutPin, outputValue);

  // печатаем результаты в Serial Monitor
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // ждём 2 миллисекунды
  delay(2);
}
```

- Загрузите программу в плату и протестируйте результат.

### Задания для самостоятельного выполнения

- На основе Упр 1. и Упр 2. соберите прибор измеряющий силу тока на участке цепи.
- Соберите схему с эмитирующую бегучую волну оставляющую медленно затухающий след. используйте потенциометр для изменения скорости волны. Сохраните результат в файле Volna.ino
- Дополните предыдущее задание, используя потенциометр для изменения скорости волны. Сохраните результат в том же файле.



Пройдите тест №2 Чтение аналоговых сигналов на странице «Тесты».