

Основы кибернетики и робототехники

Лекция 6 + Лаба 6

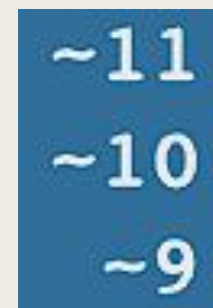
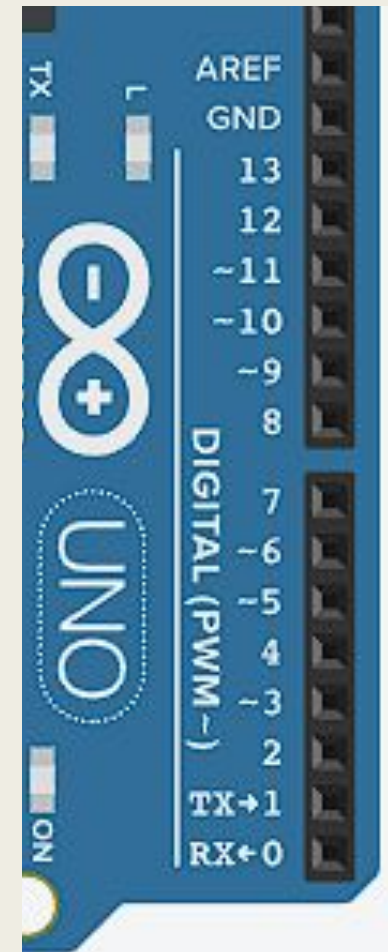
Цифровые выводы с PWM

Продолжим изучение цифровых выводов. Напомню, что есть два типа выводов: цифровые и аналоговые.

Цифровые выводы могут быть двух видов: с PWM и без них.

PWM-выводы помечены символом тильды (~): ~3, ~5, ~6, ~9, ~10, ~11 и используют широтно-импульсную модуляцию (ШИМ) или PWM. Сигнал ШИМ используется для управления устройствами, которым требуется аналоговый сигнал.

Для плавного изменения сигнала используют вызов функции `analogWrite()`, указав в параметре значение от 0 до 255.



Аналоговые

Выводы

На плате UNO есть шесть выводов, которые подписаны от A0 до A5 (у других плат может быть другое число выводов). Они работают с напряжением от 0 до 5V. Благодаря встроенному АЦП (аналого-цифровой преобразователь), данные входы могут считывать напряжение подаваемое на них.



Для чтения показания напряжения есть встроенный метод `analogRead()`, возвращающий значение от 0 до 1023. Значение 0 относится к 0V, а 1023 к 5V. Таким образом, если мы хотим конвертировать значение от 0 до 5, то нужно произвести деление $1023/5 = 204.6$

Кроме того, аналоговые выходы могут работать как цифровые и обозначаются как 14, 15, 16, 17, 18, 19 вместо A0..A5.

И, наоборот, цифровые порты с символом тильды ~ (3, 5, 6, 9, 10, 11) могут работать как аналоговые выходы, используя ШИМ.

Аналоговые выводы, в отличие от цифровых, не нужно объявлять как вход или выход в начале

Задача 1

Сделаем программу для плавного затухания светодиода с использованием цифрового PWM-вывода.

Соберем простую схему микроконтроллера, светодиода и резистора. Добавим следующий код.

```
int led = 9; // контакт 9
int b = 0; // минимальная яркость
int f = 15; // величина изменения яркости за один шаг

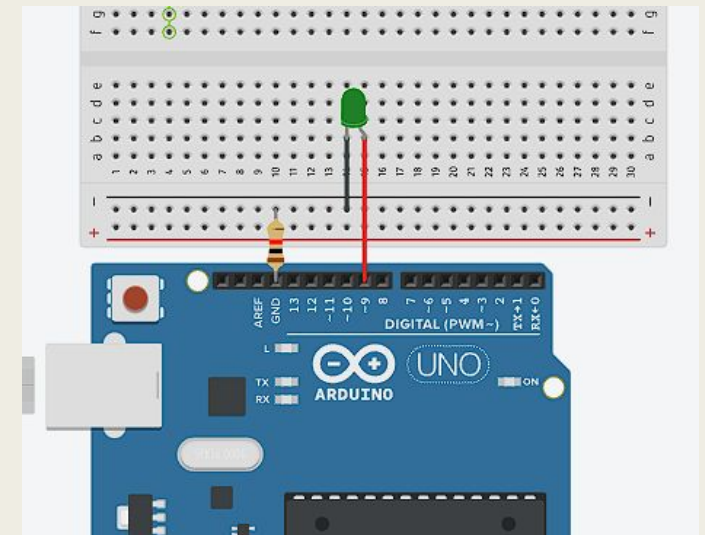
void setup() {
  // устанавливаем контакт на вывод
  pinMode(led, OUTPUT);
}

void loop() {
  // устанавливаем у контакта минимальное значение яркости
  analogWrite(led, b);

  // увеличиваем яркость на значение переменной f
  b = b + f;

  /*меняем порядок затухания на обратный,
  если достигнута максимальная или минимальная яркость */
  if (b == 0 || b == 255) {
    f = -f ;
  }

  // делаем паузу
  delay(30);
}
```



Задача 1.2

Сделаем эту же программу, но уже с использованием цикла for.

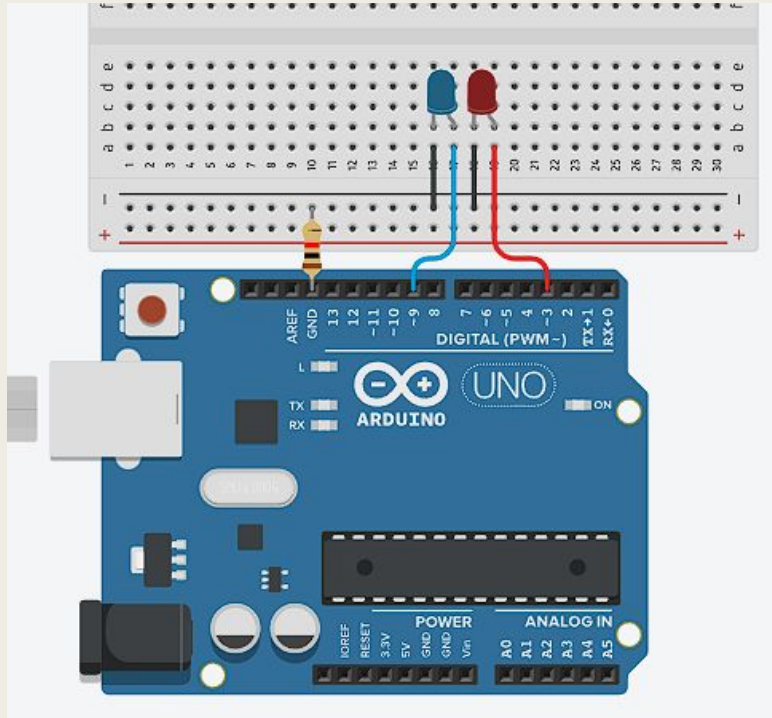
```
int led = 9; // светодиод на 9 пине

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
  // увеличиваем яркость от 0 до максимального значения по 10 единиц
  for (int i=0; i<=255; i+=10)
  {
    analogWrite (led, i);
    // делаем паузу
    delay (30);
  }
  //уменьшаем яркость от 255 до 0 по 10 единиц
  for (int i=255; i>=0; i-=10)
  {
    analogWrite (led, i);
    delay (30);
  }
}
```

Задача 2

Усложним макет. Возьмём ещё один светодиод другого цвета и соберём аналогичную схему на этой же макетной плате, подключив новый светодиод к выводу номер 3.



```
int led_9 = 9; // контакт 9
int led_3 = 3; // контакт 3
int b = 0; // минимальная яркость
int f = 15; // величина изменения яркости за один шаг

void setup() {
  // два контакта в режиме вывода
  pinMode(led_9, OUTPUT);
  pinMode(led_3, OUTPUT);
}

void loop() {
  /* устанавливаем у первого контакта минимальное значение,
  у второго - максимальное */
  analogWrite(led_9, b);
  analogWrite(led_3, 255 - b);

  // остальной код без изменений
  // увеличиваем яркость на значение переменной f
  b = b + f;

  /*меняем порядок затухания на обратный,
  если достигнута максимальная или минимальная яркость */
  if (b == 0 || b == 255) {
    f = -f;
  }

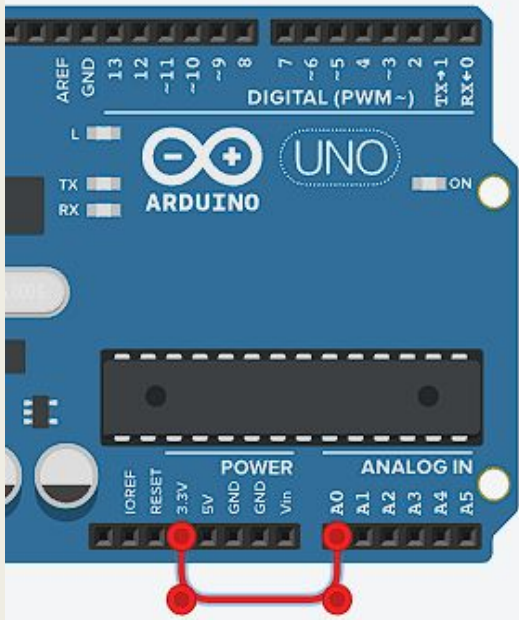
  // делаем паузу
  delay(30);
}
```

Задача 2.2

Сделайте эту же программу, но уже с использованием цикла for.

Задача 3.

Изучим простой пример с одним проводом и аналоговым выводом. Соединим проводом порты A0 и 3.3V. Напишем скетч, который будет измерять и выводить нам напряжение.



The image shows an Arduino Uno R3 board with a red wire connecting the A0 pin to the 3.3V pin. The board is blue with a black PCB. The digital pins are labeled 0-13, and the analog pins are labeled A0-A5. The power pins are labeled 5V, GND, and Vin. The board is connected to a computer via a USB cable.

```
Текст
```

```
1 int analogPin = A0;
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8   int rawReading = analogRead(analogPin);
9   float volts = rawReading / 204.6;
10  Serial.println(volts);
11  delay(1000);
12 }
```

Монитор последовательного интерфейса

```
0.00
0.00
0.00
3.30
3.30
3.30
3.30
3.30
3.30
```

Отпр. Очист.

Вольтме

Задача 4.

Если мы можем снимать значения из аналоговых портов, то можем использовать микроконтроллер как вольтметр. Достаточно вставить провода в выводы GND и A0 и соединить их с контактами на батарее (минус и плюс соответственно).

```
const int analogIn = A0;

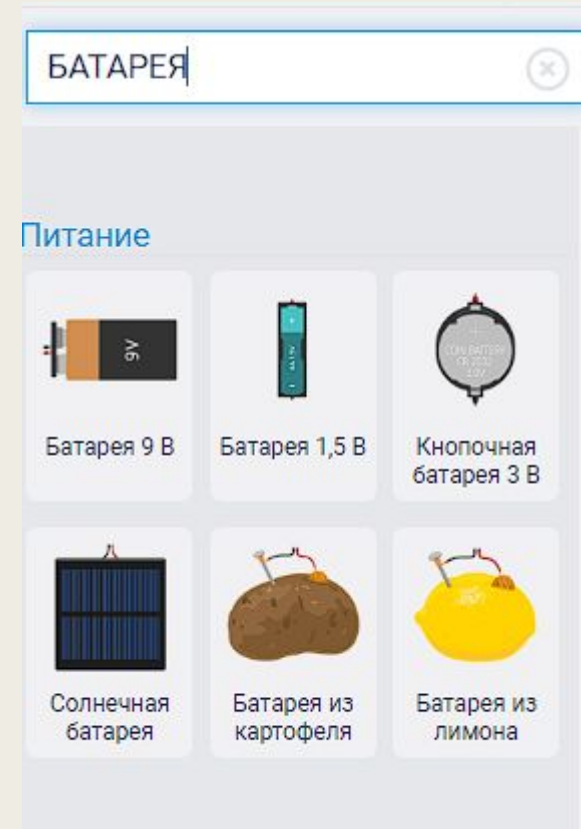
int raw = 0;
float voltage = 0;

void setup() {
  pinMode(analogIn, INPUT);
  Serial.begin(9600);
}

void loop() {
  raw = analogRead(analogIn);
  voltage = (raw * 5.0) / 1024.0;
  Serial.print("Raw Value = ");
  Serial.print(raw);
  Serial.print("\t Voltage = ");
  Serial.println(voltage, 3); // 3 цифры после запятой
  delay(500);
}
```

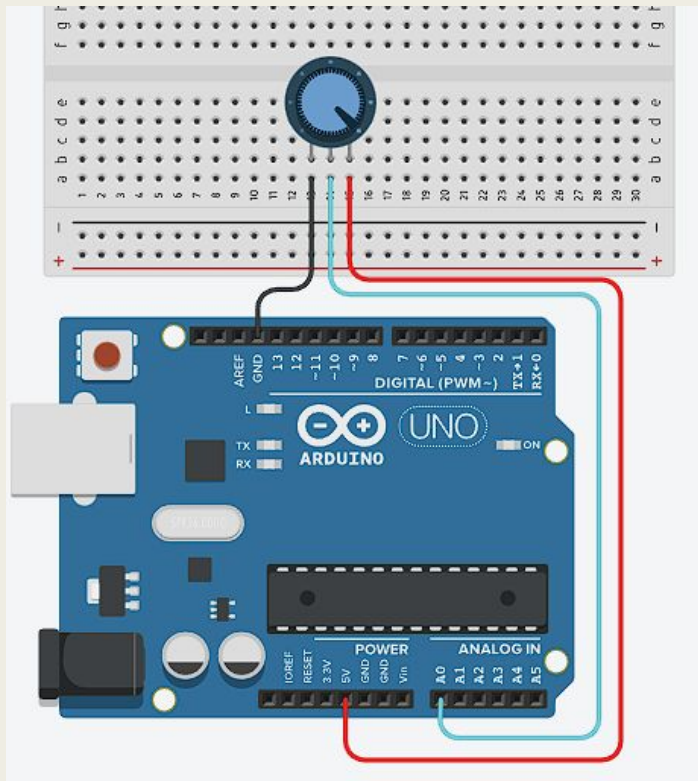
Задача 4.1

Попробуйте измерить данные всех батарей:



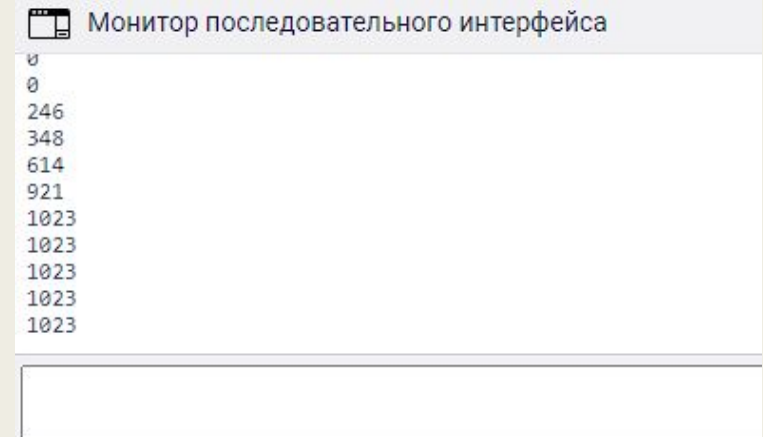
Чтение аналоговых выводов через потенциометр

Создадим скетч, который будет плавно изменять напряжение и подавать его на аналоговый вывод, чтобы получить с него текущее значение напряжения.

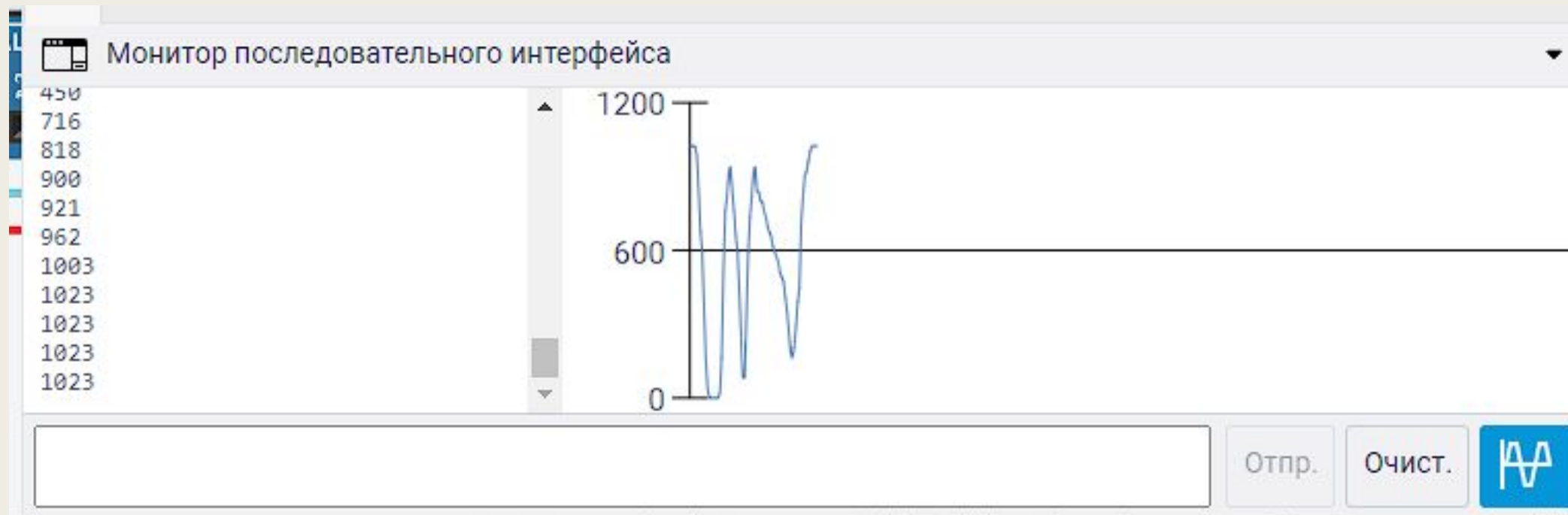


```
void setup() {  
  // инициализируем соединение на скорости 9600 бит в секунду:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // считываем данные с аналогового вывода A0  
  int i = analogRead(A0); // получаем текущее значение  
  Serial.println(i); // выводим результат на монитор  
  delay(250); // небольшая задержка для стабильности вывода результатов  
}
```

Запустите программу, а также откройте окно последовательного монитора. Вращая регулятором потенциометра, вы можете наблюдать, как в окне будут меняться значения от 0 до 1023.

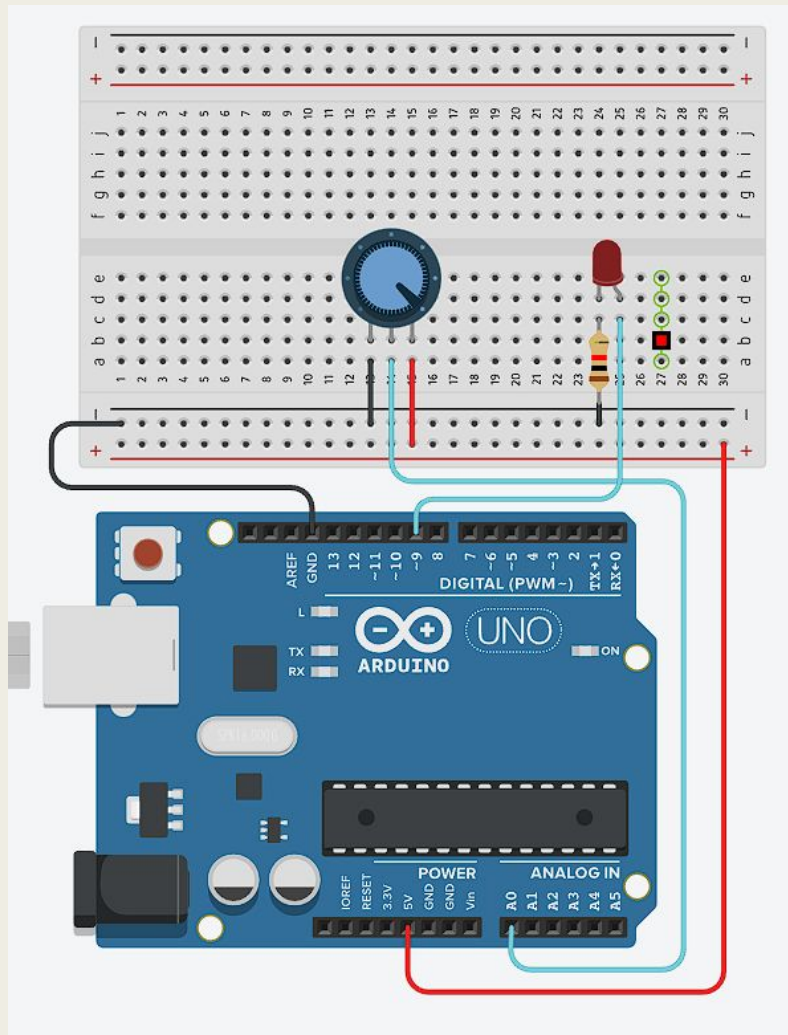


В настройках так же есть инструмент Plotter (Tools | Serial Plotter), позволяющий выводить простенький график. Вращая ручку потенциометра, можем наблюдать изменение графика.



Светодиод с плавной регулировкой

6.



```
int pin_A0 = A0;
int pin_9 = 9;

void setup() {
  // порт для светодиода на выход
  pinMode(pin_9, OUTPUT);

  // пин с потенциометром - вход
  // мы хотим считывать напряжение,
  // выдаваемое им
  pinMode(pin_A0, INPUT);
  Serial.begin(9600);
}

void loop() {
  // объявим переменную для значения напряжения с потенциометра
  int r;
  // а так же для значение яркости
  int b;

  // считываем напряжение с потенциометра от 0 до 1023 и присваем его в переменную
  r = analogRead(pin_A0);

  /* в b записываем полученное ранее значение r, делённое на 4.
  Дробная часть от деления будет отброшена. В итоге
  мы получим целое число от 0 до 255. */
  b = r / 4;

  // выдаём результат переменной b на светодиод
  analogWrite(pin_9, b);

  // выводим результат в Serial Monitor
  Serial.println(b);
  delay(30); // задержка для стабильности
}
```