

Опыт 3: Управление включением светодиода. Потенциометр

В этом опыте ты научишься управлять яркостью или частотой мигания светодиода с помощью потенциометра – переменного резистора, который работает как регулируемый делитель напряжения. При повороте его ручки напряжение на среднем выводе меняется от 0 до 1.25 В, и плата использует это значение для управления светодиодом.

Обрати внимание: У выпрямительного диода тоже есть полярность - серебряная полоска на корпусе – это «минус»!

Компоненты:

- 1 потенциометр
- 1 светодиод
- 6 проводов
- 1 резистор 220 Ом

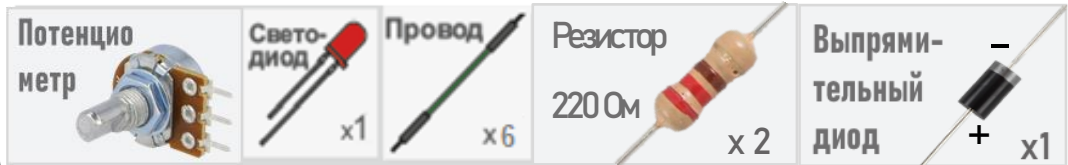
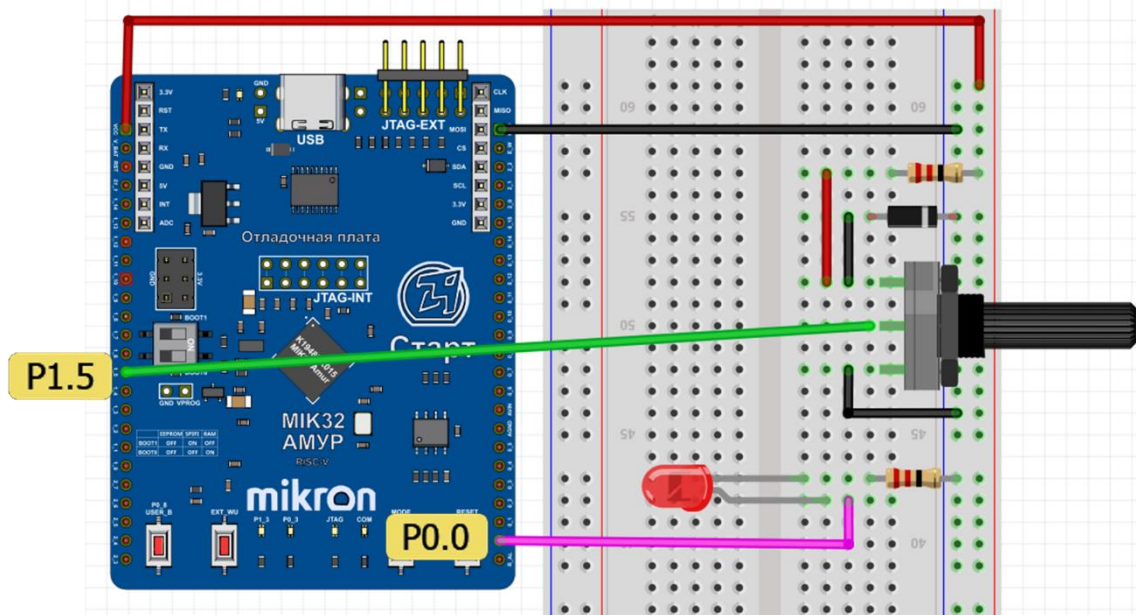


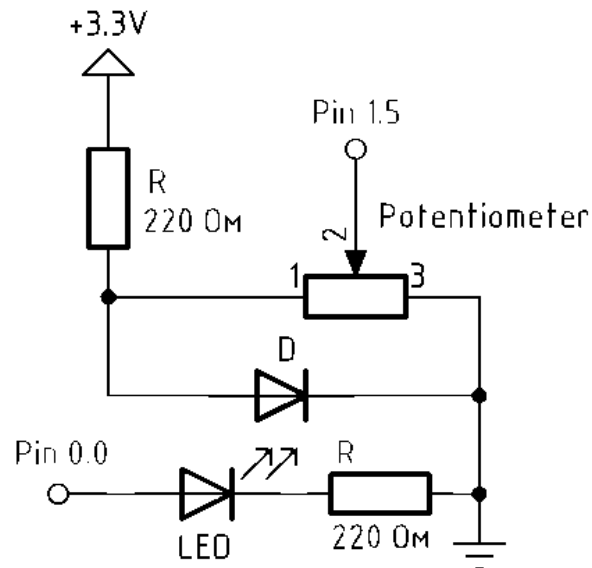
Схема:



Программа: Открой скетч меню File => Examples => StartGuide => Circuit_4_Potentiometer.

Circuit_4_Potentiometer.ino

```
1 // Пин встроенного светодиода на плате Arduino START
2 uint8_t ledPin = P0_0;
3
4 // Пин для подключения среднего вывода потенциометра.
5 // На Arduino START используется аналоговый вход P1_5 (поддерживает АЦП).
6 const int sensorPin = P1_5;
7
8 // Переменная для хранения значения, считанного с потенциометра
9 int sensorValue = 0;
10
11 // Функция setup() выполняется один раз при запуске платы
12 void setup() {
13     // Настраиваем пин светодиода как выход
14     pinMode(ledPin, OUTPUT);
15 }
16
17 // Функция loop() выполняется бесконечно
18 void loop() {
19     sensorValue = analogRead(sensorPin); // Считываем аналоговое значение с потенциометра (от 0 до 1023)
20     digitalWrite(ledPin, HIGH); // Включаем светодиод
21     delay(sensorValue); // Ждём столько миллисекунд, сколько показывает потенциометр
22     digitalWrite(ledPin, LOW); // Выключаем светодиод
23     delay(sensorValue); // Снова ждём то же время
24 }
```



Результат: мигающий светодиод, который мигает быстрее, или медленнее, в зависимости от поворота ручки потенциометра.

Что-то не так?

Неравномерная работа? Может быть из-за ненадежного контакта ножек потенциометра. Проверь надежность, слегка придавив его в макетную плату.

Не работает? Убедись, что случайно не соединил среднюю ножку потенциометра с цифровым портом, вместо аналогового.

Попробуй сам!

Задание 1.

Отключи питание платы, подключи к порту P0_1 еще один светодиод. Измени исходную схему и следующий код таким образом, чтобы второй светодиод светился на 1/8 от яркости первого.

Исходный код: Открой скетч меню File => Examples => StartGuide => Circuit_5_Brightness.

Circuit_5_Brightness.ino

```

1 // Пин встроенного светодиода на плате Arduino START
2 uint8_t ledPin = P0_0;
3
4 // Пин для подключения среднего вывода потенциометра.
5 // На Arduino START используется аналоговый вход P1_5 (поддерживает АЦП).
6 const int sensorPin = P1_5;
7
8 // Функция setup() выполняется один раз при запуске
9 void setup() {
10   pinMode(ledPin, OUTPUT); // Настраиваем пин светодиода как выход
11 }
12
13 // Функция loop() выполняется бесконечно
14 void loop() {
15   int rotation, brightness; //Переменные для хранения значения потенциометра и яркости светодиода
16   // Считываем значение с потенциометра: от 0 (минимум) до 1023 (максимум)
17   rotation = analogRead(sensorPin);
18
19   // Преобразуем значение в диапазон 0-255, подходящий для analogWrite
20   // (1023 / 4 ≈ 255 – целочисленное деление)
21   brightness = rotation / 4;
22
23   // Подаем на светодиод напряжение, пропорциональное положению ручки потенциометра
24   analogWrite(ledPin, brightness);
25 }

```

Задание 2.

Подключи RGB-светодиод (или три отдельных – красный, зелёный, синий) и сделай так, чтобы цвет менялся от красного к синему при повороте ручки потенциометра:

- в начале диапазона – горит только красный,
- в середине – только зелёный,
- в конце – только синий.

💡 Подсказка: раздели диапазон показаний потенциометра (`analogRead`) на три части и используй `if`-условия, чтобы управлять каждым цветом.

Задание 3.

Создай эффект «дыхания»: светодиод плавно разгорается и гаснет, как будто дышит.

Скорость этого «дыхания» должна регулироваться потенциометром:

- при одном крайнем положении – дыхание очень медленное,
- при другом – быстрое, почти как мигание.

💡 Подсказка: используй цикл `for` с `analogWrite` и регулируй задержку (`delay`) внутри него на основе значения с потенциометра.