

Опыт 2: Использование кнопок

Часть 1: Светодиод, индикация состояния нажатия кнопки

Цифровые пины работают с двумя уровнями сигнала: HIGH (+3.3 В) и LOW (0 В), что делает их удобными для подключения кнопок – устройств с двумя состояниями: «нажата» и «не нажата».

Кнопку подключают так: один контакт – к земле (GND), другой – к цифровому пину. При нажатии пин замыкается на землю, и плата считывает сигнал как LOW. Однако в отпущенном состоянии пин оказывается «в воздухе» – в таком плавающем состоянии микроконтроллер может ошибочно считать его HIGH или LOW. Чтобы избежать неопределённости, к пину подключают подтягивающий резистор (1-10 кОм) к шине +3.3 В. Тогда при отпущенной кнопке на пине гарантированно будет HIGH, а при нажатии – LOW. Светодиод (LED) – это маленькая лампочка, которая загорается, когда через неё идёт ток.

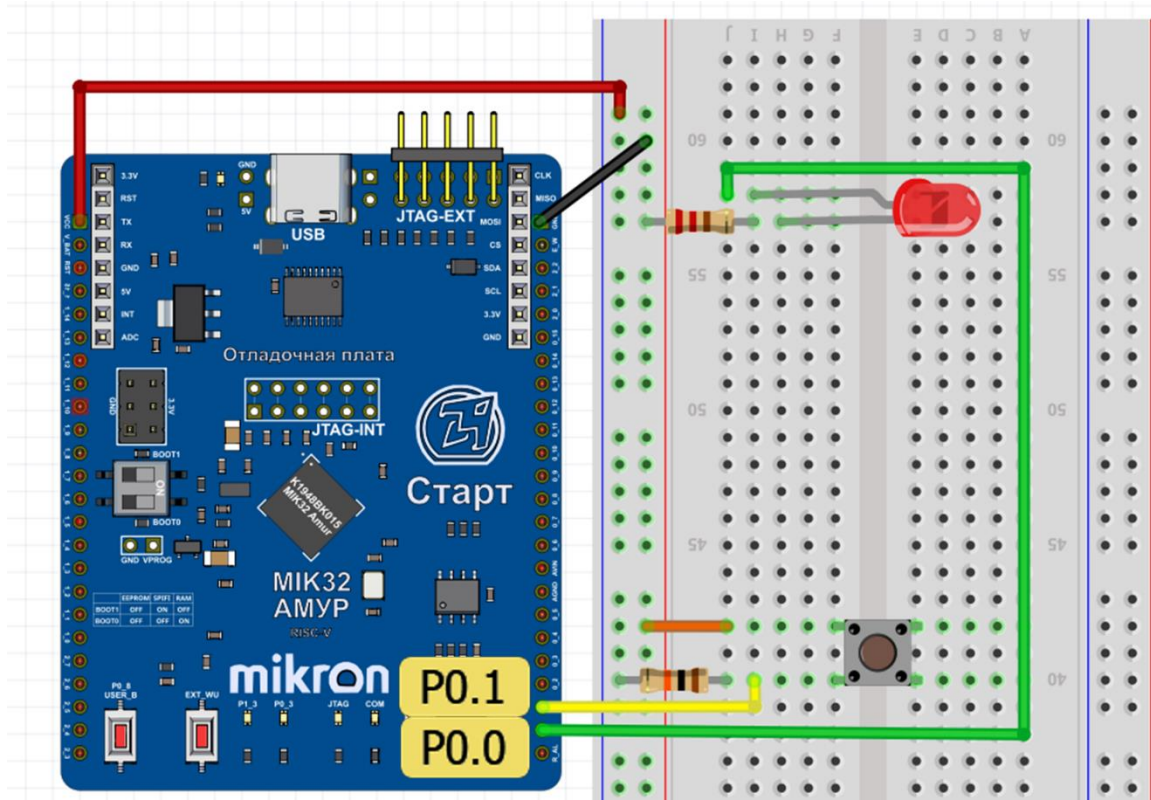
В этом опыте заставим светодиод включаться по нажатию кнопки.

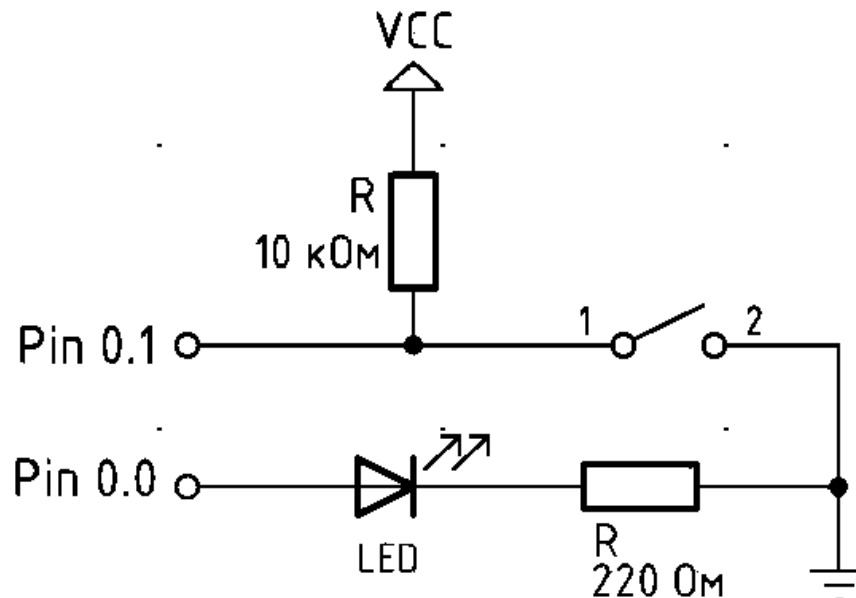
Компоненты:

- 1 кнопки
- 1 светодиод
- 5 проводов
- 1 резистор 10 кОм
- 1 резистор 220 Ом



Схема:





Программа: Открой скетч меню File => Examples => StartGuide => Circuit_2_Button.

Circuit_2_Button.ino

```

1 // Пин встроенного светодиода на плате Arduino START
2 uint8_t ledPin = P0_0;
3
4 // Пин встроенной кнопки на плате Arduino START
5 uint8_t btn_pin = P0_1;
6
7 // Переменная для хранения состояния кнопки (0 – нажата, 1 – отпущена)
8 int buttonS = 0;
9
10 // Функция setup() выполняется один раз при запуске или перезагрузке
11 void setup() {
12   pinMode(ledPin, OUTPUT); // Настраиваем светодиод как выход
13   pinMode(btn_pin, INPUT); // Настраиваем кнопку как вход
14 }
15
16 // Функция loop() выполняется бесконечно
17 void loop() {
18   buttonS = digitalRead(btn_pin); // Считываем состояние кнопки
19
20   if (buttonS == 0) {           // Если кнопка нажата (сигнал LOW)
21     digitalWrite(ledPin, HIGH); // Включаем светодиод
22   } else {                       // Если кнопка отпущена
23     digitalWrite(ledPin, LOW);  // Выключаем светодиод
24   }
25 }

```

Попробуй сам!

Задание 1. Добавь еще один светодиод и сделай так, чтобы если кнопка нажата - горел один светодиод, если не нажата - другой. Не забудь, что второму диоду тоже необходим подтягивающий резистор.

Часть 2: Индикация состояния двух кнопок

В этой схеме в качестве датчика используются кнопки – простой способ подать сигнал на плату. Кнопки подключены между «Землей» и платой START: при нажатии вход становится LOW («0»). Чтобы избежать ложных срабатываний, в схеме используются подтягивающие резисторы.

Компоненты:

- 2 кнопки
- 1 светодиод
- 7 проводов
- 2 резистора 10 кОм
- 1 резистор 220 Ом

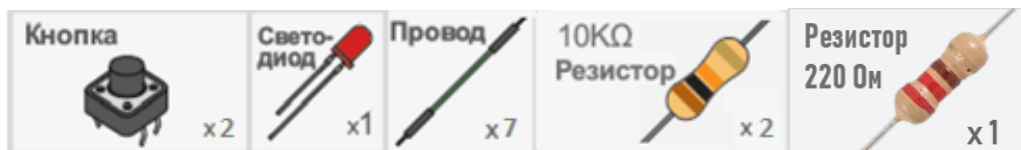
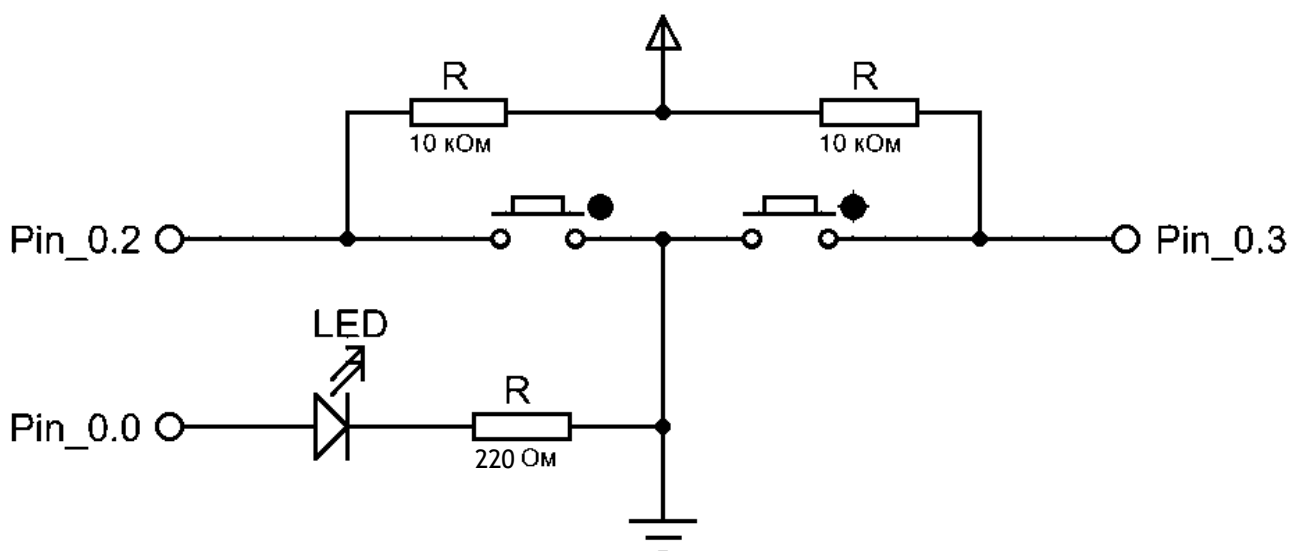
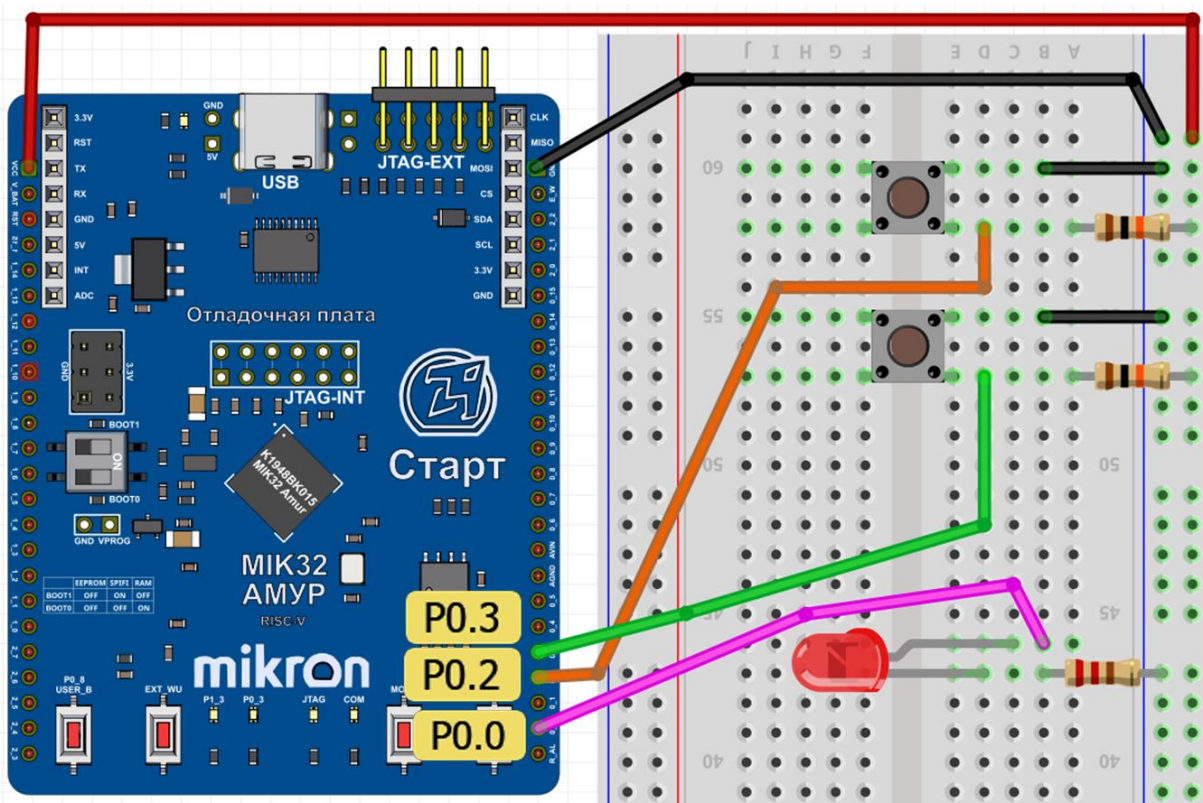


Схема:



Программа: Открой скетч меню File => Examples => StartGuide => Circuit_3_Buttons.

Circuit_3_Buttons.ino

```
1 // Пин встроенного светодиода на плате Arduino START
2 uint8_t ledPin = P0_0;
3
4 // Пины двух внешних кнопок на плате Arduino START
5 const int button1Pin = P0_2; // Кнопка №1 подключена к P0_2
6 const int button2Pin = P0_3; // Кнопка №2 подключена к P0_3
7
8 // Функция setup() выполняется один раз при запуске
9 void setup() {
10     pinMode(button1Pin, INPUT); // Настраиваем P0_2 как вход для кнопки №1
11     pinMode(button2Pin, INPUT); // Настраиваем P0_3 как вход для кнопки №2
12     pinMode(ledPin, OUTPUT); // Настраиваем P0_0 как выход для светодиода
13 }
14
15 // Функция loop() выполняется бесконечно
16 void loop() {
17     // Считываем текущее состояние обеих кнопок
18     int button1State = digitalRead(button1Pin);
19     int button2State = digitalRead(button2Pin);
20
21     // Включаем светодиод, только если нажата ровно одна кнопка
22     if (((button1State == LOW) || (button2State == LOW)) &&
23         !((button1State == LOW) && (button2State == LOW))) {
24         digitalWrite(ledPin, HIGH);
25     } else {
26         digitalWrite(ledPin, LOW);
27     }
28 }
```

Результат: светодиод горит, когда нажата только одна кнопка из двух, и не горит, когда нажаты обе кнопки.

Попробуй сам!

Задание 2. Создай «секретный сигнал»: светодиод должен мигнуть три раза подряд, только если сначала быстро нажать первую кнопку, а потом – вторую в течение 2 секунд.

Если порядок неправильный (например, сначала вторую) – ничего не происходит.

💡 Подсказка: тебе понадобятся переменные для хранения состояния и millis() для отсчёта времени (или упрощённый вариант с delay() и проверкой поочерёдности).

Задание 3. Доработай программу так, чтобы каждое нажатие на любую из двух кнопок увеличивало счётчик, а светодиод мигал столько раз, сколько кнопок было нажато всего с начала работы программы. (Например: Первое нажатие (любой кнопки) → светодиод мигнёт 1 раз, второе нажатие → мигнёт 2 раза подряд, третье → 3 раза подряд, и так далее.)

💡 Подсказка: Используй переменную-счётчик (например, int count = 0;). При нажатии любой кнопки увеличивай её: count++. Затем запусти цикл for, чтобы мигнуть count раз.

Чтобы не считалось несколько раз при удержании – добавь простую «антидребезговую» задержку (delay(200) после нажатия).