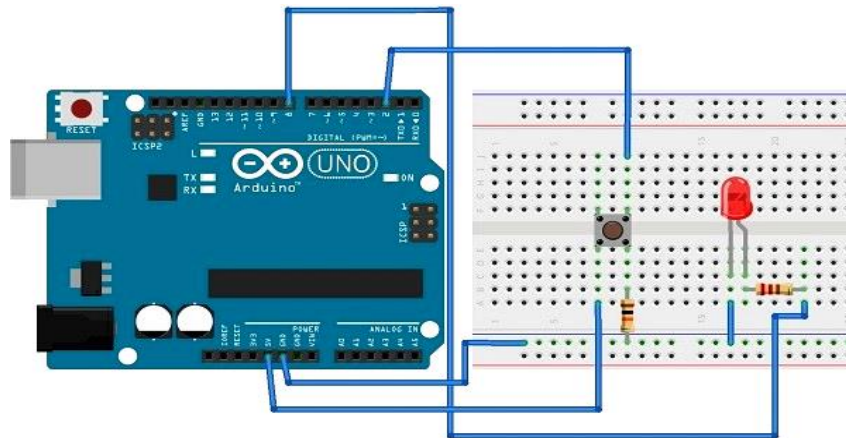


ВКЛЮЧАЕМ УП МК АРДУИНО КНОПКОЙ

Подключаем к ардуино кнопку и светодиод (при нажатой кнопке светодиод будет гореть, при отжатой — не гореть). Это одна из базовых схем, которая неоднократно пригодится вам в будущем и может использоваться для управления роботом на Arduino.

В будущем по нажатию кнопки могут происходить самые разные действия. Для сборки модели нам потребуется:

- плата Arduino
- Breadboard
- 5 проводов и/или перемычек «папа-папа»
- светодиод
- кнопка
- резисторы на 10 кОм и 220 Ом.



Для работы этой модели подойдет следующая программа:

```
int button = 2;
int led = 8;

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

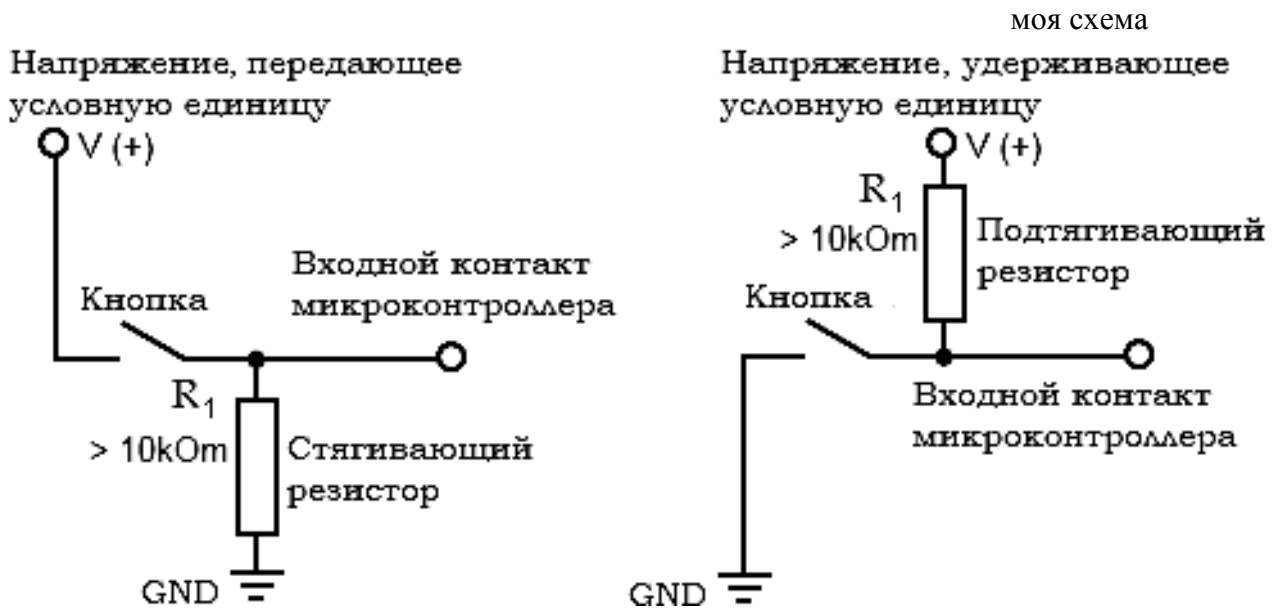
void loop(){
  if (digitalRead(button) == HIGH) {
    digitalWrite(led, HIGH);
  }

  else {
    digitalWrite(led, LOW);
  }
}
```

Примечание:

Если просто соединить выводы кнопки через резистор с пином контроллера и землей, то это не работает. При нажатии кнопки пин соединяется с землей, и digitalRead будет считывать логический 0, но при отпущенной кнопке пин не будет

соединен ни с какой линией и будет «висеть в воздухе», и программа будет считать с вывода и 0 и 1 случайным образом. Правильное подключение предполагает, что в разомкнутом состоянии пин должен быть соединен через резистор с шиной питания, а в замкнутом — с землей, либо наоборот.

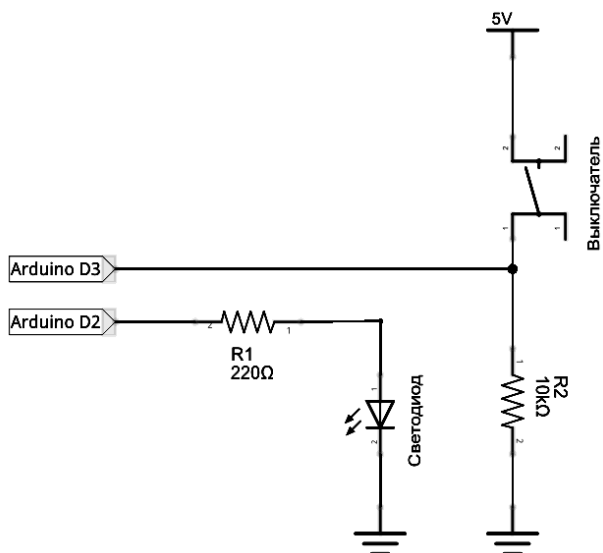


Внутри микроконтроллера есть подтягивающие резисторы. Но они могут быть программно включены или выключены. Попробуйте включить и выключить подтягивающие резисторы программно.

После того, как пин установлен в режим входа, чтобы включить на нем подтягивающий резистор нужно «записать в него 1» с помощью `digitalWrite`.

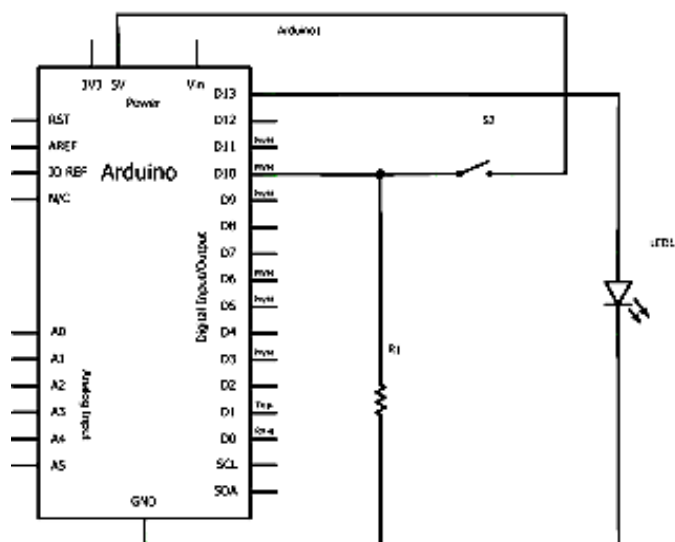
```
void setup() {
  pinMode(8, OUTPUT);
  pinMode(2, INPUT);
  digitalWrite(2, HIGH); //включаем подтягивающий резистор
}
```

```
void loop() {
  digitalWrite(8, !digitalRead(2));
}
```



Кнопками включаем-выключаем

соберем схему:



Включаем светодиод при нажатой кнопке, отключаем при отпущенной кнопке.

Пример 1.

```
int LedPin = 13;           // Подключаем светодиод к порту 13
```

```
int ButPin = 10;          // Подключаем кнопку к выходу 10
```

```
void setup()
```

```
{
  pinMode(LedPin, OUTPUT); // Инициализируем порт "ledPin" как выходной
}
```

```
void loop()
```

```
{
  if(digitalRead(ButPin) == HIGH) // если кнопка нажата
  {
    digitalWrite(LedPin, HIGH); // то включаем светодиод
  }
  else // иначе
```

```
{
  digitalWrite(LedPin,LOW);    // светодиод отключаем
}
}
```

Включаем светодиод при нажатии, Отключаем при повторном нажатии.

Следующим шагом введем в код задержку на подавление дребезга контактов, а также выполним переключение состояния светодиода за счет инвертирования состояния пина с подключенным к нему светодиодом:

Пример 2

```
int LedPin = 13;          // Подключаем светодиод к порту 13
int ButPin = 10;         // Подключаем кнопку к выходу 10

void setup()
{
  pinMode(LedPin, OUTPUT); // Инициализируем порт "ledPin" как выходной
}

void loop()
{
  if(digitalRead(ButPin) == HIGH)    // если кнопка нажата
  {
    digitalWrite(LedPin, !digitalRead(LedPin)); // то переключаем состояние LedPin
    delay(500);
  }
}
```

Включаем светодиод при долгом нажатии на кнопку

Пример 3

```
int LedPin = 13;          // Подключаем светодиод к порту 13
int ButPin = 10;         // Подключаем кнопку к выходу 10
int valBut = 0;          // Задаем переменную, для счетчика
long previousMillis = 0; // Зададим начальное значение для счетчика millis
long TimePush = 500;     // Время нажатия на кнопку
int LedVal = LOW;        // устанавливаем начальное состояние светодиода

void setup()
{
  pinMode(LedPin, OUTPUT); // Инициализируем порт "ledPin" как выходной
}

void loop()
{
  if(digitalRead(ButPin) == HIGH) // если кнопка нажата
```

```

{
    //
    if(millis() - previousMillis >= 1) // начинаем считать время нажатия
    {
        //
        previousMillis = millis(); //
        valBut++; // с каждой миллисекундой увеличиваем значение valBut
    }
}

else // если кнопку отпустили, то valBut становится равным 0
{
    // это необходимо для защиты от срабатывания при частых
    valBut = 0; // многократных нажатиях
}

if(valBut >= TimePush) // как только значение valLed становится равным TimePush
{
    //
    digitalWrite(LedPin, LedVal != LedVal); // то инвертируем состояние LedPin
    valBut = 0; // и устанавливаем valBut = 0
}
}

```

Включение/отключение светодиода при однократном нажатии за счет переключения "флага" состояния кнопки.

Данная программа в отличие от предыдущих, которые циклично повторяли действие при удержании, выполняет всего одно действие при нажатии и удержании.

Пример 4

```

int LedPin = 13; // Подключаем светодиод к порту 13
int ButPin = 10; // Подключаем кнопку к выходу 10
int flag = 0; // флаг состояния

void setup()
{
    pinMode(LedPin, OUTPUT); // Инициализируем порт "ledPin" как выходной
}

void loop()
{
    if(digitalRead(ButPin) == HIGH && flag == 0)
    {
        digitalWrite(LedPin, !digitalRead(LedPin));
        flag = 1;
    }

    if(digitalRead(ButPin) == LOW && flag == 1)
    {
        flag = 0;
    }
}

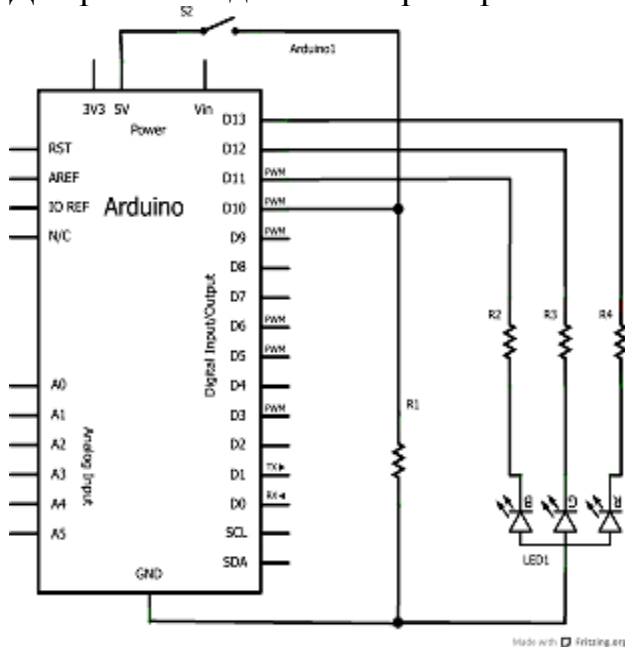
```

Переключаем режимы работы светодиодов по одной кнопке.

Алгоритм работы: По нажатию одной кнопки последовательно переключаем RGB светодиод в режимах: **OFF, R, G, B, RG, GB, RB, RGB**

Переключение режимов работы будет происходить за счет изменения переменной `regim`, значение которой будет изменяться при каждом нажатии на кнопку.

Для работы с данным примером необходимо изменить электрическую схему:



В этом случае скетч будет следующим:

Пример 5

```
int R_LedPin = 13;           // Подключаем красный светодиод к порту 13
int G_LedPin = 12;           // Подключаем зеленый светодиод к порту 12
int B_LedPin = 11;           // Подключаем синий светодиод к порту 11
int ButPin = 10;              // Подключаем кнопку к выходу 10
int flag = 0;                 // флаг состояния
int regim = 0;                // Переключалка
```

```
void setup()
```

```
{
  pinMode(R_LedPin, OUTPUT);
  pinMode(G_LedPin, OUTPUT);
  pinMode(B_LedPin, OUTPUT);
}
```

```
void loop()
```

```
{
  if(digitalRead(ButPin) == HIGH && flag == 0)
  {
```

```

regim ++;
flag = 1;

if(regim > 7)           // Если номер режима превышает требуемого
{                       // то отсчет начинается с нуля
    regim = 0;
}
}

if(digitalRead(ButPin) == LOW && flag == 1)
{
    flag = 0;
}

// ===== Вполняем задачу при выборе режима =====
// РЕЖИМ 0: OFF
if(regim == 0)
{
    digitalWrite(R_LedPin, LOW);
    digitalWrite(G_LedPin, LOW);
    digitalWrite(B_LedPin, LOW);
}

// РЕЖИМ 1: R
if(regim == 1)
{
    digitalWrite(R_LedPin, HIGH);
    digitalWrite(G_LedPin, LOW);
    digitalWrite(B_LedPin, LOW);
}

// РЕЖИМ 2: G
if(regim == 2)
{
    digitalWrite(R_LedPin, LOW);
    digitalWrite(G_LedPin, HIGH);
    digitalWrite(B_LedPin, LOW);
}

// РЕЖИМ 3: B
if(regim == 3)
{
    digitalWrite(R_LedPin, LOW);
    digitalWrite(G_LedPin, LOW);
    digitalWrite(B_LedPin, HIGH);
}

// РЕЖИМ 4: RG
if(regim == 4)

```

```
{  
    digitalWrite(R_LedPin, HIGH);  
    digitalWrite(G_LedPin, HIGH);  
    digitalWrite(B_LedPin, LOW);  
}
```

```
// РЕЖИМ 5: GB
```

```
if(regim == 5)  
{  
    digitalWrite(R_LedPin, LOW);  
    digitalWrite(G_LedPin, HIGH);  
    digitalWrite(B_LedPin, HIGH);  
}
```

```
// РЕЖИМ 6: RB
```

```
if(regim == 6)  
{  
    digitalWrite(R_LedPin, HIGH);  
    digitalWrite(G_LedPin, LOW);  
    digitalWrite(B_LedPin, HIGH);  
}
```

```
// РЕЖИМ 7: RGB
```

```
if(regim == 7)  
{  
    digitalWrite(R_LedPin, HIGH);  
    digitalWrite(G_LedPin, HIGH);  
    digitalWrite(B_LedPin, HIGH);  
}  
}
```