

Практическая работа 12

Кнопочный переключатель

В этом эксперименте мы делаем из тактовой кнопки триггер, борясь с «дребезгом».

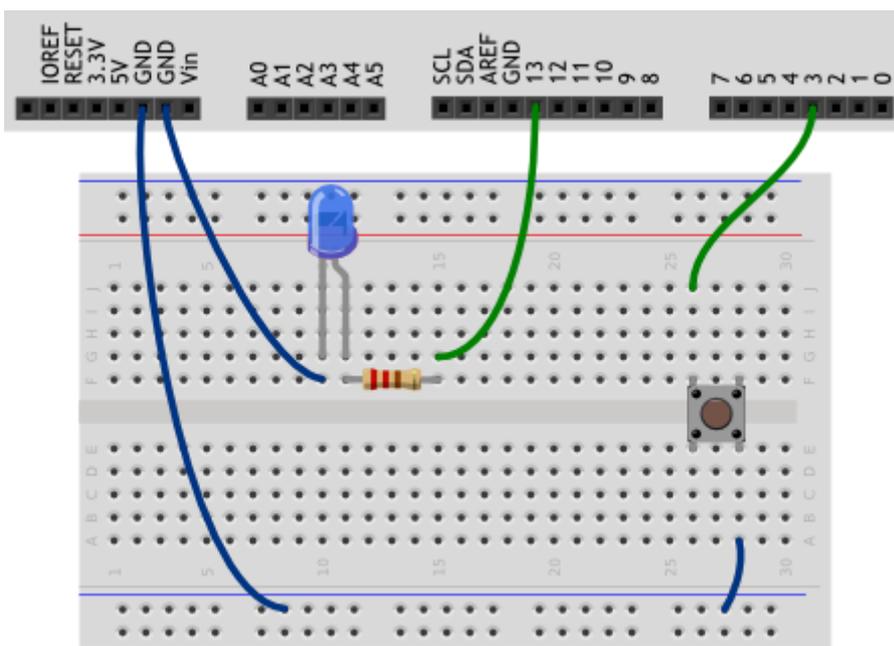
Задание 1. Ответить на вопросы

1. Что такое «дребезг»?
2. Как «дребезг» скажется на светодиоде?
3. Какие способы подключения кнопки Вы знаете? Зарисуйте схемы.

Задание 2. Список деталей для эксперимента

1. 1 плата Arduino Uno
2. 1 беспаячная макетная плата
3. 1 тактовая кнопка
4. 1 резистор 220 Ом
5. 1 светодиод
6. 5 проводов «папа-папа»

Схема на макетной плате:



1. Зарисуйте принципиальную схему установки.

Обратите внимание

✓ Мы могли бы один из контактов кнопки соединить проводом напрямую с одним из входов GND, но мы сначала «раздали» «землю» на длинную рельсу макетки. Если мы работаем с макетной платой, так поступать удобнее, т.к. в схеме могут появляться новые участки, которые тоже нужно будет соединить с «землей»

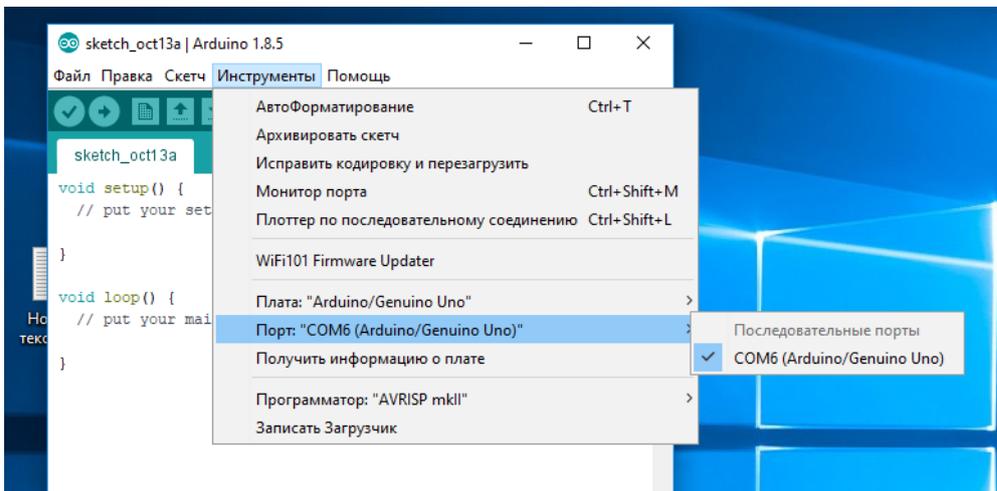
✓ Также полезно руководствоваться соображениями аккуратности изделия, поэтому катод светодиода мы соединяем с другим входом GND отдельным проводом, который не мешает нам работать в середине макетки.

Задание 3. Программирование микроконтроллера

1. Запустите приложение



2. Убедитесь, что выбран нужный порт



3. Наберите в редакторе кода следующий код программы:

```
sketch_nov13a §  
#define BUTTON_PIN 3  
#define LED_PIN 13  
  
boolean buttonWasUp = true; // была ли кнопка отпущена?  
boolean ledEnabled = false; // включен ли свет?  
  
void setup()  
{  
  pinMode(LED_PIN, OUTPUT);  
  pinMode(BUTTON_PIN, INPUT_PULLUP);  
}  
  
void loop()  
{  
  // определить момент «клика» несколько сложнее, чем факт того,  
  // что кнопка сейчас просто нажата. Для определения клика мы  
  // сначала понимаем, отпущена ли кнопка прямо сейчас...  
  boolean buttonIsUp = digitalRead(BUTTON_PIN);  
  
  // ...если «кнопка была отпущена и (&&) не отпущена сейчас»...  
  if (buttonWasUp && !buttonIsUp) {  
    // ...может это «клик», а может и ложный сигнал (дребезг),  
    // возникающий в момент замыкания/размыкания пластин кнопки,  
    // поэтому даём кнопке полностью «успокоиться»...  
    delay(10);  
    // ...и считываем сигнал снова  
    buttonIsUp = digitalRead(BUTTON_PIN);  
    if (!buttonIsUp) { // если она всё ещё нажата...  
      // ...это клик! Переворачиваем сигнал светодиода  
      ledEnabled = !ledEnabled;  
      digitalWrite(LED_PIN, ledEnabled);  
    }  
  }  
  
  // запоминаем последнее состояние кнопки для новой итерации  
  buttonWasUp = buttonIsUp;  
}
```

37 Arduino/Genuino Uno на COM11

Пояснения к коду

Поскольку мы сконфигурировали вход кнопки как INPUT_PULLUP, при нажатии на кнопку на данном входе мы будем получать 0. Поэтому мы получим значение true («истина») в булевой переменной buttonIsUp («кнопка отпущена»), когда кнопка отпущена.

Логический оператор && («и») возвращает значение «истина» только в случае истинности обоих его операндов. Взглянем на так называемую таблицу истинности для выражения buttonWasUp && !buttonIsUp («кнопка была отпущена и кнопка не отпущена»):

buttonWasUp	buttonIsUp	!buttonIsUp	buttonWasUp && !buttonIsUp
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Здесь рассмотрены все возможные сочетания предыдущего и текущего состояний кнопки и мы видим, что наш условный оператор if сработает только в случае, когда кнопка нажата только что: предыдущее состояние 1 («была отпущена»), а текущее 0 («не отпущена»).

Через 10 миллисекунд мы проверяем еще раз, нажата ли кнопка: этот интервал больше, чем длительность «дребезга», но меньше, чем время, за которое человек успел бы дважды нажать на кнопку. Если кнопка всё еще нажата, значит, это был не дребезг.

Мы передаем в digitalWrite не конкретное значение HIGH или LOW, а просто булеву переменную ledEnabled. В зависимости от того, какое значение было для нее вычислено, светодиод будет зажигаться или гаситься.

Последняя инструкция в buttonWasUp = buttonIsUp сохраняет текущее состояние кнопки в переменную предыдущего состояния, ведь на следующей итерации loop текущее состояние уже станет историей.

Задание 4. Ответьте на следующие вопросы

1. В каком случае оператор `&&` возвращает значение «истина»?
2. Что такое «дребезг»?
3. Как мы с ним боремся в программе?
4. Как можно избежать явного указания значения уровня напряжения при вызове `digitalWrite`?

Задание 5. Самостоятельно измените существующую программу и схему

1. Измените код так, чтобы светодиод переключался только после отпускания кнопки.
2. Добавьте в схему еще одну кнопку и доработайте код, чтобы светодиод зажегся только при нажатии обеих кнопок.