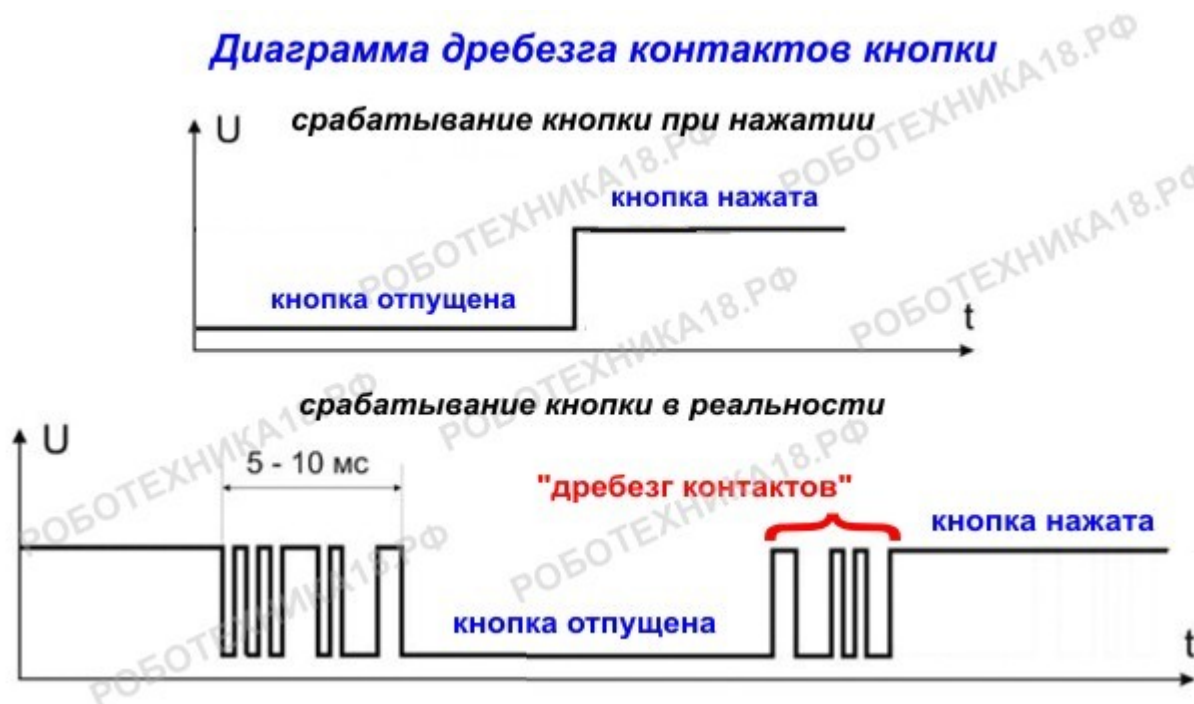


Как подключить кнопку к Ардуино без дребезга

Как подключить тактовую кнопку к цифровым и аналоговым входам на Arduino Uno правильно. Расскажем, как избежать «дребезг контактов», рассмотрим схему подключения одной и нескольких кнопок к плате Ардуино. Рассмотрим функции `digitalRead` и `analogRead` в языке программирования C++, которые часто требуется использовать при создании различных проектов на Ардуино для начинающих.

Работа с тактовыми кнопками на Ардуино

Главная проблема использования кнопок для управления Arduino заключается в «дребезге контактов». Дело в том, что механические контакты в тактовых кнопках никогда не замыкаются и размыкаются мгновенно. В течении нескольких миллисекунд происходит многократное замыкание и размыкание контактов — в итоге на микроконтроллер поступает не единичный сигнал, а серия импульсов.

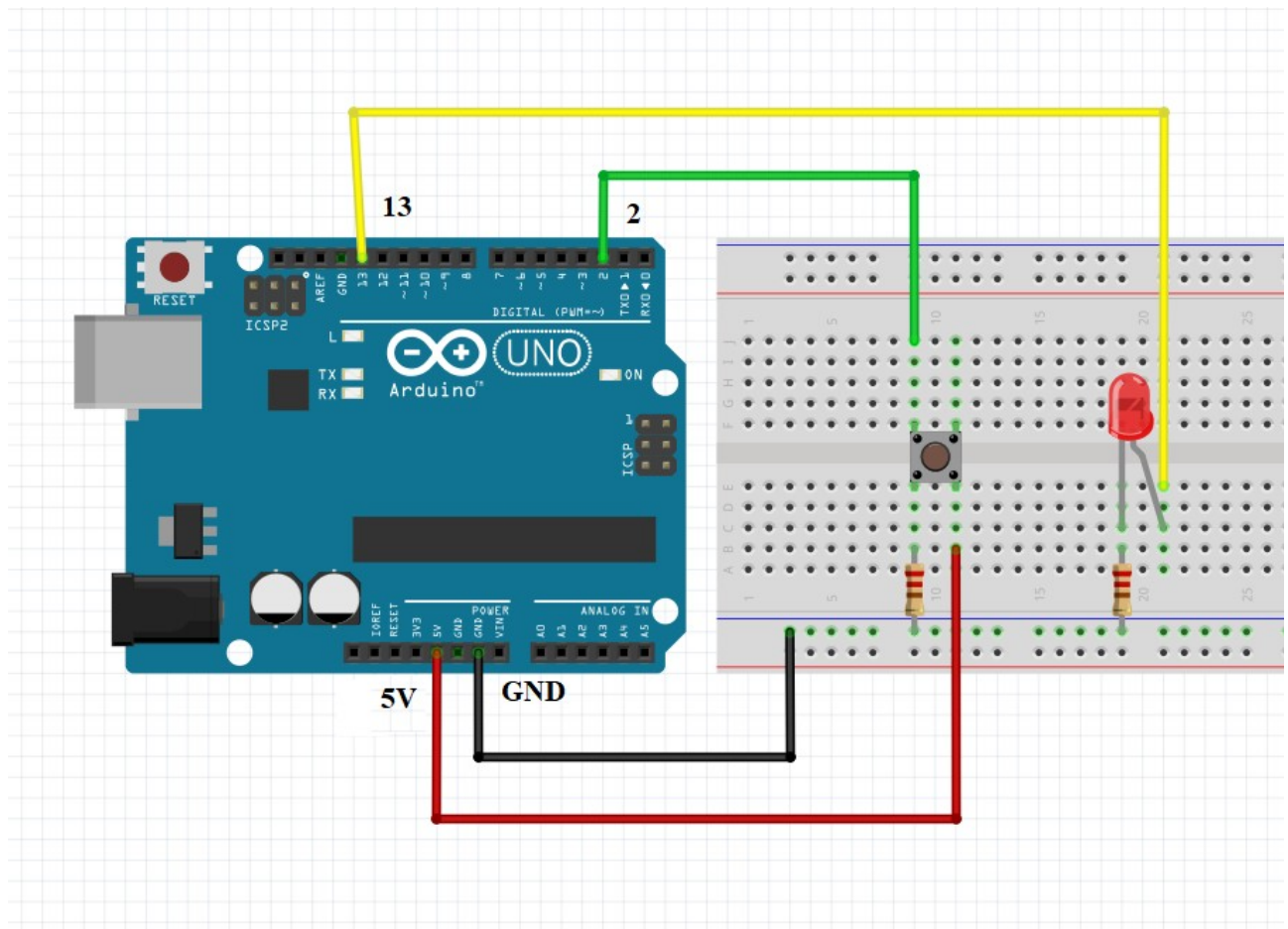


Для того, чтобы исключить на микроконтроллере Arduino дребезг кнопки используют различные электрические схемы с триггерами и конденсаторами. Но намного удобнее и проще использовать программный способ борьбы с возможным дребезгом тактовой кнопки — для этого применяют задержку на несколько миллисекунд или используют библиотеку `Bounce2.h` для борьбы с дребезгом контактов для Arduino.

Для этого занятия потребуется:

- Arduino Uno / Arduino Nano / Arduino Mega;
- макетная плата;
- 1 светодиод;
- резисторы на 220 Ом и 10 кОм;

- 1 тактовая кнопка;
- провода «папа-папа» и «папа-мама».



Используем цифровые порты на плате для подключения тактовой кнопки и команду `digitalRead` для считывания данных. Соберите схему, как на рисунке выше и загрузите скетч. Обратите внимание, что при отпущенной кнопке на Pin2 поступает логический «0». С помощью кнопки будем выключать и включать встроенный светодиод, подключенный к цифровому 13 порту на плате микроконтроллера Ардуино.

Скетч Ардуино подключение кнопки к цифровому входу

```
void setup() {
  pinMode(13, OUTPUT); // объявляем пин 13 как выход
  pinMode(2, INPUT); // объявляем пин 2 как вход
}

void loop() {
  if (digitalRead(2) == HIGH) // когда на пин 2 поступает высокий сигнал
  {
    digitalWrite(13, HIGH); // зажигаем светодиод
  }

  if (digitalRead(2) == LOW) // когда на пин 2 поступает низкий сигнал
  {
```

```
    digitalWrite(13, LOW); // выключаем светодиод
  }
}
```

Пояснения к коду:

1. процедура `setup` выполняется один раз, используется процедура для конфигурации портов микроконтроллера (назначение режима работы портов);
2. процедуры `setup` и `loop` должны присутствовать в любой программе (скетче);
3. использованные константы: `INPUT`, `OUTPUT`, `LOW`, `HIGH`, пишутся заглавными буквами, иначе компилятор их не распознает и выдаст ошибку.

Задание: При нажатии кнопки светодиод горит, а при следующем нажатии выключается

```
boolean buttonWasUp = true;
boolean ledEnabled = false;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}

void loop() {
  // узнаем, отпущена ли кнопка сейчас
  boolean buttonIsUp = digitalRead(2);

  // если кнопка была отпущена и не отпущена сейчас
  if (buttonWasUp && !buttonIsUp) {

    // исключаемдребезг контактов тактовой кнопки
    delay(10);

    // и считываем сигнал с кнопки снова
    buttonIsUp = digitalRead(2);

    // если кнопка нажата, то переворачиваем сигнал светодиода
    if (!buttonIsUp) {
      ledEnabled = !ledEnabled;
      digitalWrite(13, ledEnabled);
    }
  }

  // запоминаем состояние кнопки для новой итерации
  buttonWasUp = buttonIsUp;
}
```

