

## **Лабораторная работа № 1**

### **“Введение. Цифровые порты ввода-вывода”**

#### **Цель работы**

Знакомство с Arduino. Изучение цифровых портов ввода-вывода.

#### **Задача**

Познакомиться с теоретическим материалом и выполнить задание по вариантам.

#### **Необходимый материал для выполнения лабораторной работы**

1. Arduino UNO
2. Breadboard
3. Светодиоды
4. Тактовые кнопки
5. Резисторы
6. Монтажные провода

#### **Содержание отчёта**

1. Цель работы.
2. Задание на работу по варианту.
3. Собранная схема подключения.
4. Блок-схема алгоритма.
5. Код, написанный на языке программирования C.
6. Выводы по работе.

## Содержание

Теоретическая часть .....	3
Аппаратная часть .....	3
Подключение светодиода и тактовой кнопки .....	6
Программная часть .....	9
Программирование тактовой кнопки и светодиода .....	10
Обработка однократного нажатия кнопки .....	14
Что такоедребезг и как от него избавиться .....	16
Задания .....	18
Полезные ссылки .....	23

## **Теоретическая часть**

### **Аппаратная часть**

Ардуино (Arduino) - аппаратно-программные средства для создания простых электронных систем автоматики и робототехники. Система имеет полностью открытую архитектуру.

*Аппаратная часть* представляет собой электронные платы с микроконтроллером, сопутствующими элементами (стабилизатор питания, кварцевый резонатор, блокировочные конденсаторы и т.п.), портом для связи с персональным компьютером, разъемами для сигналов ввода-вывода и т.п.

В рамках данной дисциплины, лабораторные работы разработаны для отладочной платы Arduino UNO, самой популярной из семейства плат Ардуино, возможности которой наглядно отражены на рисунке ниже.

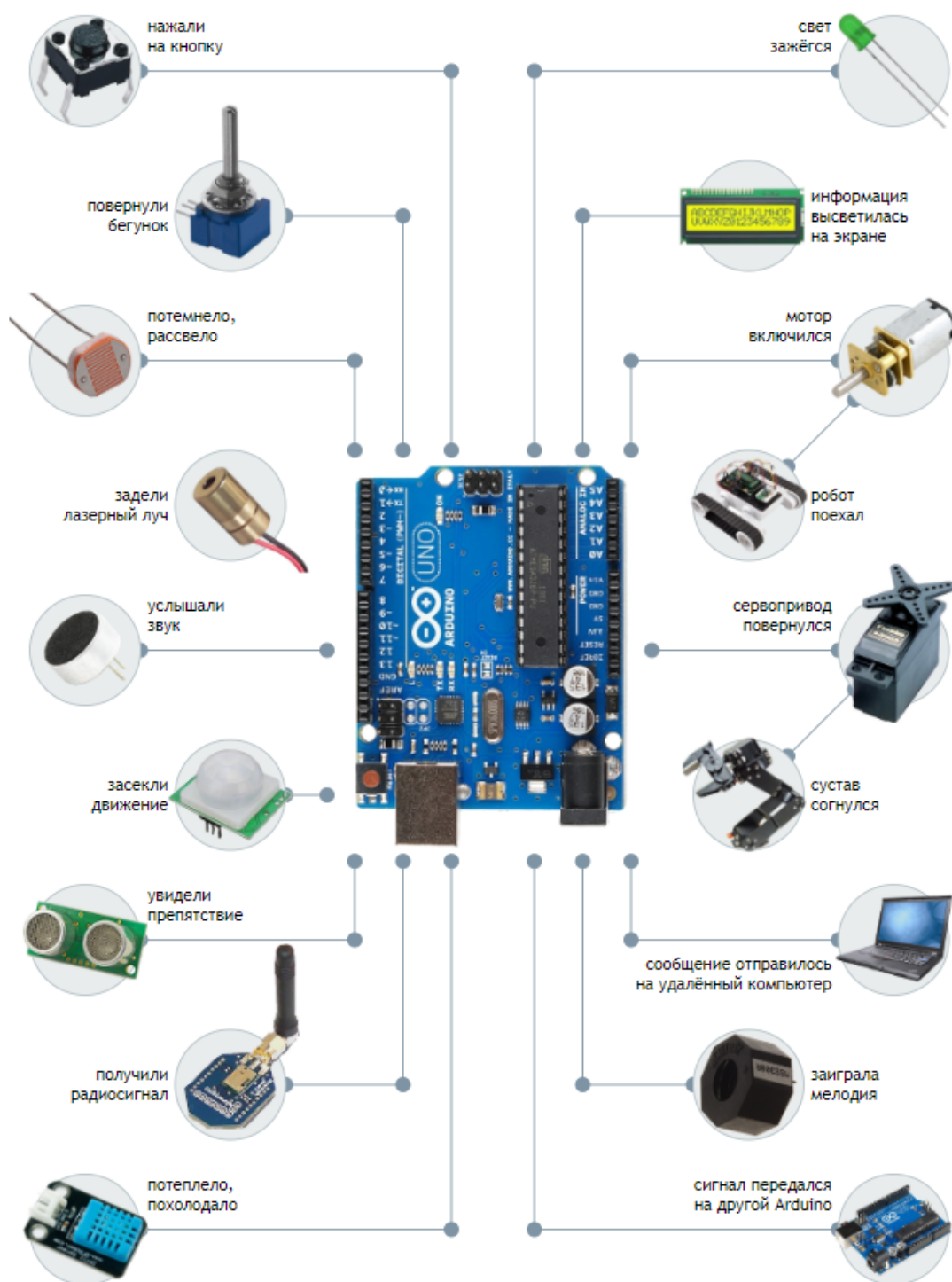


Рисунок 1 - Возможности платы Arduino Uno

Рассмотрим подробнее устройство и распиновку платы. Arduino Uno R3 выполнена на микроконтроллере ATmega328 и содержит:

- 14 цифровых портов входа-выхода ( 6 из них поддерживают режим ШИМ модуляции);

- 6 аналоговых входов;
- частота тактирования 16 МГц;
- USB порт;
- разъем питания;
- разъем внутрисхемного программирования;
- кнопка сброса.

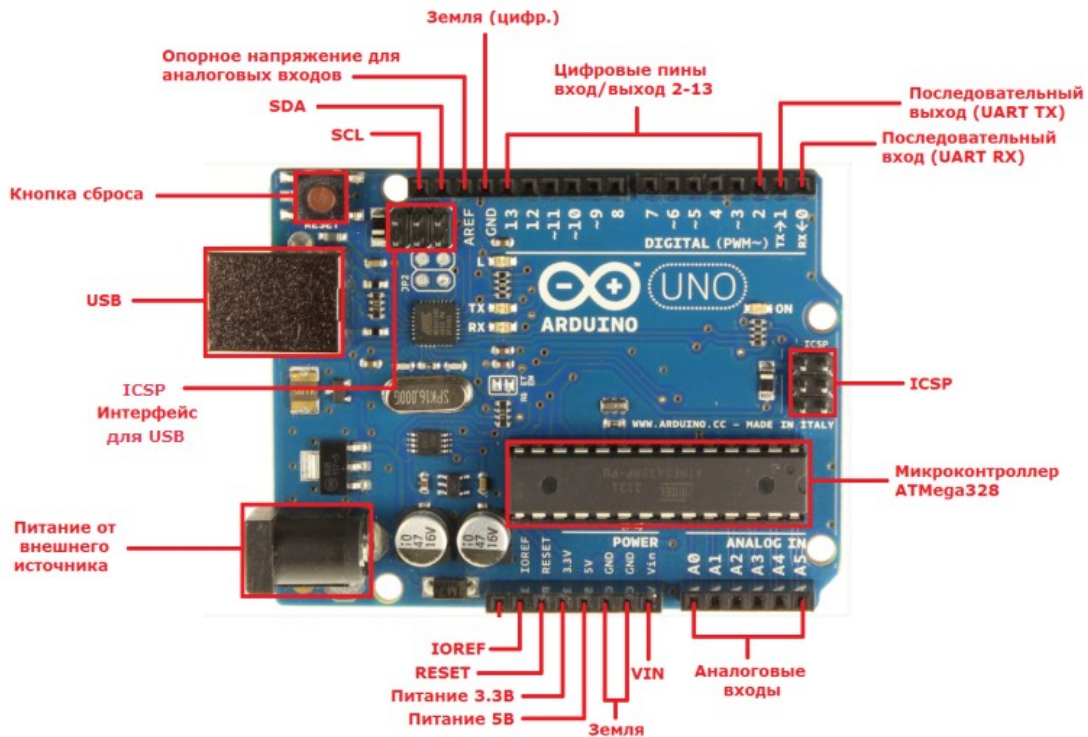
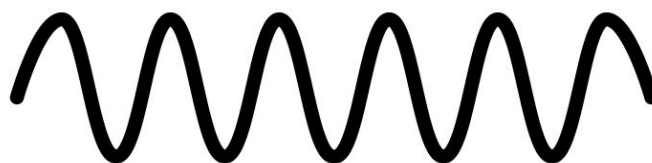


Рисунок 2 - Распиновка платы Arduino Uno

В рамках данной лабораторной работы, *питание и прошивка платы* будет осуществляться за счет подключения к компьютеру через бортовой USB порт.

На рисунке 2 указано, что пины 2-13 являются цифровыми портами ввода-вывода. Разберемся с тем, что такое цифровой сигнал. Цифровой сигнал представлен последовательностью цифровых значений. Чаще всего применяются двоичные цифровые сигналы (“0” и “1”), так как они используются в двоичной электронике и легче кодируются. Так, например, сигнал с тактовой кнопки может быть равен “1”, если кнопка нажата и “0” если не нажата. Помимо цифрового сигнала существует также аналоговый

сигнал, который подробно будет рассмотрен в следующей лабораторной работе.



Аналоговый сигнал



Цифровой сигнал

Рисунок 3 - Изображение аналогового и цифрового сигналов

### **Подключение светодиода и тактовой кнопки**

Так как данная лабораторная работа нацелена на формирование навыка по работе с цифровыми входами-выходами, разберем подключение элементов (светодиода и тактовой кнопки) к пинам, генерирующим цифровой сигнал. Для этого, помимо Arduino Uno и светодиода с тактовой кнопкой понадобится макетная плата (или Breadboard) - универсальная плата для сборки и моделирования прототипов электронных устройств, а также монтажные провода по типу “папа-папа” и резисторы.

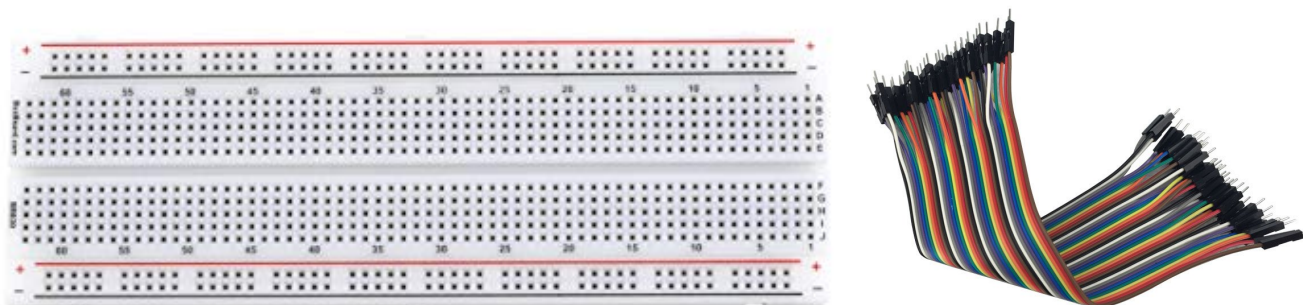


Рисунок 4 - Макетная плата и провода “папа-папа”

Для начала необходимо подать питание с Arduino Uno на шину питания макетной платы, затем, для подключения кнопки, подвести плюс питания на кнопку и через резистор (во избежание короткого замыкания) пустить на землю; для снятия цифрового сигнала (нажата кнопка или нет) необходимо подтянуть цифровой пин в цепь. Если цифровой сигнал снимать между плюсом питания и кнопкой, то при нажатии на кнопку на контроллер придет значение “0”, если наоборот - “1” (Рисунок 5).

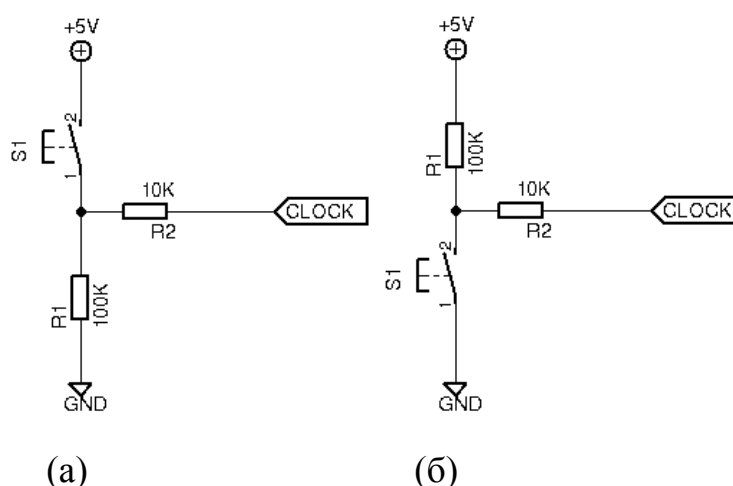


Рисунок 5 - Разница снятия цифрового сигнала в цепи с тактовой кнопкой (на рис. 5 (a) после нажатия на кнопку придет сигнал “1”; на рис. 5 (б) после нажатия на кнопку придет сигнал “0”)

Для управления светодиодом, необходимо подвести пин, генерирующий управляющий сигнал на анод светодиода (рисунок 6), а катод пустить на минус питания через резистор (собственное сопротивление диода мало и без резистора, ограничивающего ток, он перегорит). Схема подключения изображена на рисунке 7.

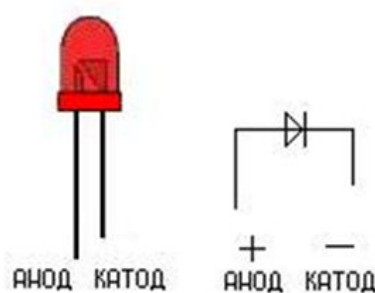


Рисунок 6 - Устройство светодиода

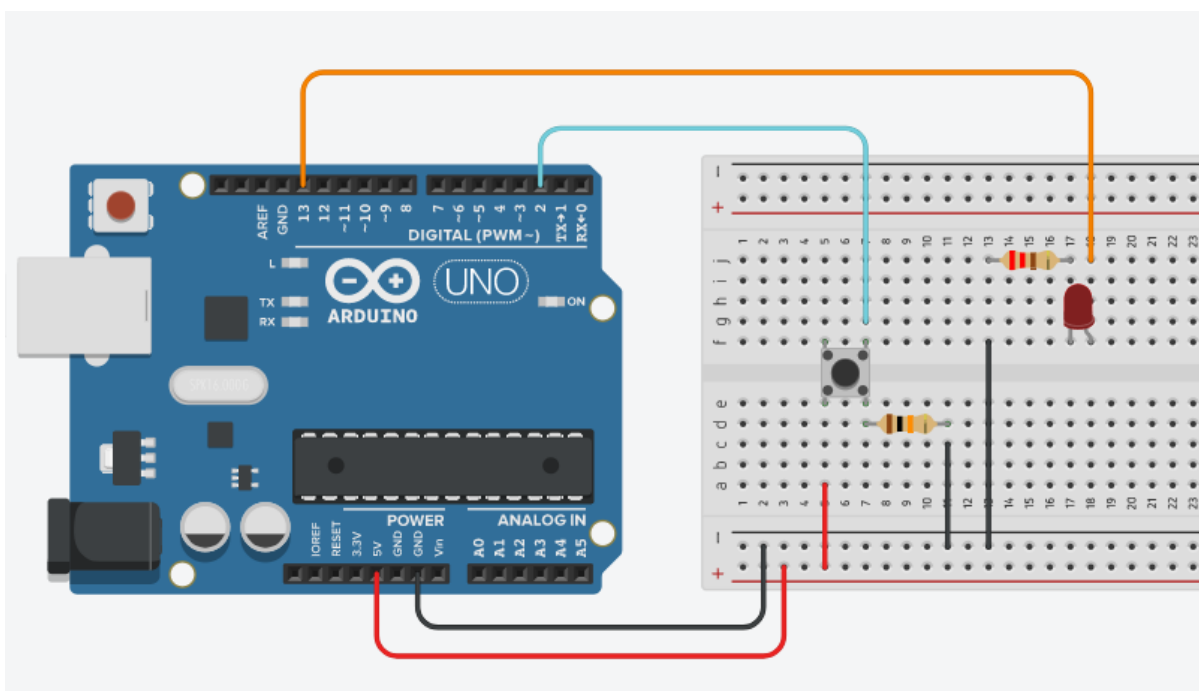


Рисунок 7 - Схема подключения светодиода и тактовой кнопки

Для расчета номинала резисторов применяется закон Ома. Чтобы найти сопротивление резистора, который включен в цепь с кнопкой, необходимо напряжение питания 5 В поделить на значение тока 40 мА. Для расчета сопротивления резистора, включенного в цепь с диодом для начала найдем значение напряжение которое должно упасть на резисторе, чтобы не перегорел светодиод. Напряжение питания равняется 5 В, питающее напряжение светодиода примерно 3 В, следовательно на резисторе должно упасть 2В при токе в 40 мА. Таким образом, значения сопротивлений резисторов равны 125 Ом и 50 Ом соответственно, т.е. для предотвращения короткого замыкания и перегорания светодиода необходимо, чтобы сопротивления выбранных резисторов были больше или равны подсчитанным.

### Программная часть

После подключения исполнительных элементов к Arduino Uno встает задача написания кода и прошивки платы. Для удобства работы с Arduino существует бесплатная официальная среда программирования [Arduino IDE](#),



которая была разработана под самые основные операционные системы: Windows, Mac OS и Linux. Скачать данную среду можно по ссылке 3 из раздела Полезные ссылки.

Итак, разберёмся со средой программирования. После скачивания, успешной установки и открытия программы, на экране появится окно, представленное на рисунке 8.

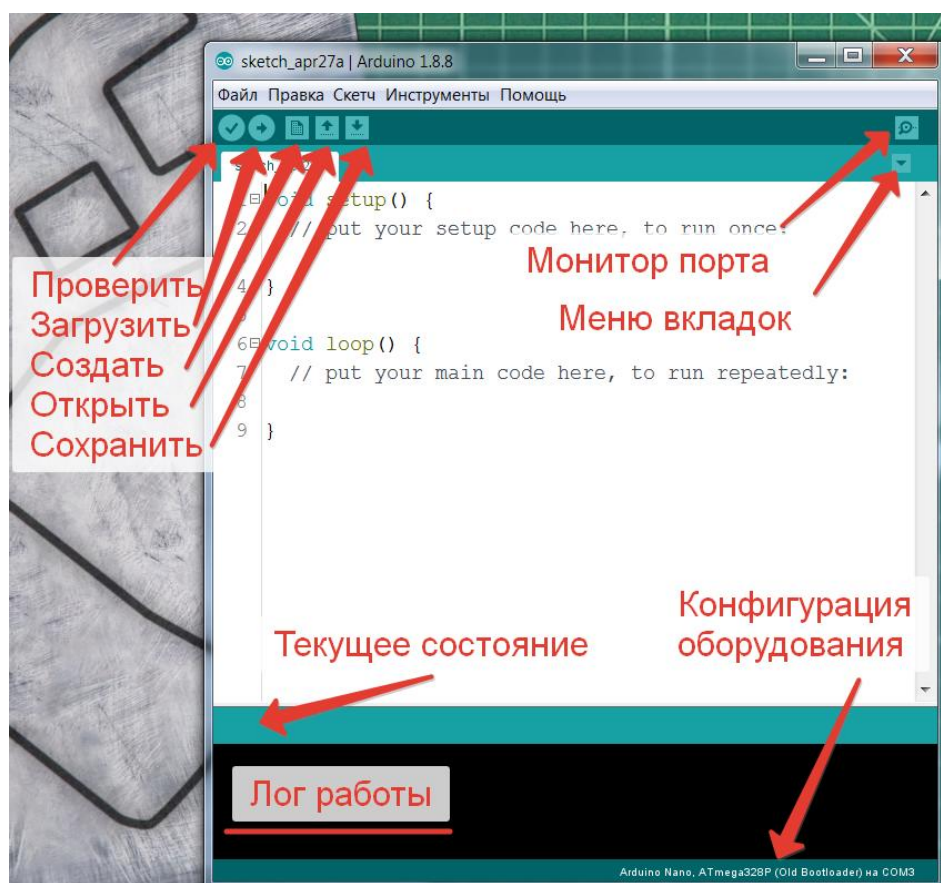


Рисунок 8 - Окно программы Arduino IDE с описанием интерфейса

Для начала загрузим отладочный код для проверки работоспособности платы и IDE. Для этого необходимо подключить плату к компьютеру с помощью USB кабеля (по необходимости произвести установку драйверов). Затем запустить Arduino IDE, в Инструментах выбрать плату Arduino Uno и нужный порт. После этого необходимо зайти в Файл - Примеры - 0.1 Basics - Blink, нажать кнопка Загрузка. После этого светодиод на плате, обозначенный L должен мигать.

## Программирование тактовой кнопки и светодиода

Теперь приступим к написанию кода. Данная программа должна управлять светодиодом с помощью кнопки: при нажатой кнопке светодиод светится, при отжатой кнопке светодиод не светится. Пример программы с комментариями представлен в Листинге 1.

Листинг 1 - Пример программы включения светодиода по нажатию кнопки

```
1. #define button 2    //Обозначение пина к которому подключен кнопка
2. #define led_pin 13  //Обозначение пин к которому подключен
   светодиод
3.
4.
5. void setup() { //Функция инициализации переменных
6.   pinMode(button, INPUT); //Переводим указанный пин в режим
   считывания данных
7.   pinMode(led_pin, OUTPUT); //Переводим указанный порт в режим
   вывода данных
8. }
9.
10. void loop() { //Функция, которая постоянно вызывается. Обычно
   в ней происходят основные действия
11.   digitalWrite(led_pin, digitalRead(button)); //Записываем в
   порт, к которому подключен светодиод, состояние которое приходит
   с кнопки, т.е. 0 или 1
12.   delay(50); //Задержка для нормализации работы кнопки, то
   есть гашениядребезга
13. }
```

Для работы с цифровыми выводами в системе Ардуино есть 3 встроенные функции. Они позволяют установить режим вывода, считать состояние вывода или установить вывод в определенное состояние. Для определения состояния выводов в этих функциях используются константы HIGH и LOW, которые соответствуют высокому и низкому уровню сигнала, помимо этих констант можно использовать цифры “0” и “1”, которые будут расцениваться контролером, как низкий и высокий уровень соответственно.

Разберем подробно код из листинга 1.

Первым делом идёт объявление макросов, для удобной работы с кодом в дальнейшем. Для этого используется директива препроцессора “#define”, которая позволяет дать имя константе перед тем как программа будет

скомпилирована. Определенные этой директивой константы не занимают программной памяти, поскольку компилятор заменяет все обращения к ним их значениями на этапе компиляции, соответственно они служат исключительно для удобства программиста и улучшения читаемости текста программы.

Следующим шагом идёт вызов функции “`setup()`”, которая используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т.д. Функция “`setup()`” запускается только один раз, после каждой подачи питания или сброса платы Arduino. В этой функции запускаются функции, которые определяют режим портов, скорость работы UART и тому подобное.

Последней вызывается функция “`loop()`”, которая делает точь-в-точь то, что означает её название, и крутится в цикле, позволяя вашей программе совершать вычисления и реагировать на них. В традиционном виде её можно сравнить с функцией “`main()`”, отличается только тем, что в традиционной функции “`main()`” для циклического выполнения программы необходимо запускать цикл, который всегда будет выполняться.

Далее перечислим и разберём функции, которые использовались в листинге 1.

Первая функция **`pinMode(pin, mode)`**. Она устанавливает режим вывода (вход или выход).

Аргументы: `pin` и `mode`. Где `pin` – номер вывода; `mode` – режим вывода, всего их 3:

- 1) `INPUT` - вывод определен как вход, подтягивающий резистор отключен.
- 2) `INPUT_PULLUP` - вывод определен как вход, подтягивающий резистор подключен.
- 3) `OUTPUT` - вывод определен как выход.

Следующая функция **`digitalWrite(pin, value)`**. Она устанавливает состояние выхода (высокое или низкое).

Аргументы pin и value: pin – номер вывода; value – состояние выхода. Состояний выходов всего 2, если только это не аналоговый выход.

- 1) HIGH - установление порта в высокое состояние, то есть на его выходе будет 5 вольт. Помимо использования данного слова, можно использовать цифру 1, что также переведёт порт в высокой состояние.
- 2) LOW - установление порта в низкое состояние, то есть 0 вольт. Цифровое обозначение - цифра 0.

Последняя функция **digitalRead(pin)**. Данная команда позволяет считывать состояние входа. Она имеет всего один аргумент “pin” - это номер пина, с которого нужно что-то считать. Данная функция возвращает состояние входа, которое может быть либо 0 либо 1 или HIGH либо LOW.

Для примера разберем следующее задание: “Необходимо при нажатии на первую кнопку первый светодиод включать, второй выключать, при нажатии на вторую кнопку, второй светодиод включать, первый выключать”. В листинге 2 представлен один из возможных вариантов решения этой задачи.

#### Листинг 2 - Обработка нажатия 2-х кнопок

```
1. #define button1 2    //Пин к которому подключен кнопка №1
2. #define button2 3    //Пин к которому подключен кнопка №2
3. #define led1_pin 13   //Пин к которому подключен светодиод №3
4. #define led2_pin 12   //Пин к которому подключен светодиод №4
5.
6. bool flag = 0; //Обозначение булевой переменной - флаг
7.
8. void setup() {
9.     pinMode(button1, INPUT); //Переводим указанный пин в режим
    считывания данных
10.    pinMode(button2, INPUT); //Переводим указанный пин в режим
    считывания данных
11.    pinMode(led1_pin, OUTPUT); //Переводим указанный порт в
    режим вывода данных
12.    pinMode(led2_pin, OUTPUT); //Переводим указанный порт в
    режим вывода данных
13. }
14.
15. void loop() {
16.     if(digitalRead(button1) == 1 && flag == 0){
17.         flag = 1; //Поднимаем флаг
18.         delay(50); //Задержка для нормализации работы кнопки, то
```

```

    есть гашения дребезга
19.     }
20.     if(flag == 1){
21.         digitalWrite(led1_pin, HIGH); //Записываем в порт, к
           которому подключен светодиод №1 значение высокого напряжения
22.         digitalWrite(led2_pin, LOW); //Записываем в порт, к
           которому подключен светодиод №2 значение низкого напряжения
23.     }
24.     if(digitalRead(button2) == 1 && flag == 1){
25.         flag = 0; //Опускаем флаг
26.         delay(50); //Задержка для нормализации работы кнопки, то
           есть гашения дребезга
27.     }
28.     if(flag == 0){
29.         digitalWrite(led1_pin, LOW); //Записываем в порт, к
           которому подключен светодиод №1 значение низкого напряжения
30.         digitalWrite(led2_pin, HIGH); //Записываем в порт, к
           которому подключен светодиод №2 значение высокого напряжения
31.     }
32. }

```

В листинге выше представлен простой пример обработки нажатия двух кнопок. В принципе, этот листинг отличается от 1-го только тем, что у него в основной функции имеются условия с использованием, так называемого “флага”. Принцип работы “флага” прост, изначально необходимо выставить значение флага 0, затем, при нажатии на первую кнопку меняет своё состояние на противоположное и также делает при нажатии на вторую кнопку. Теперь в условиях необходимо проверять не только нажата ли кнопка, но и состояние флага. Внутри условия, в данном случае, происходит переключение светодиодов.

### **Обработка однократного нажатия кнопки**

Очень часто при проектировании систем возникает необходимость ограничить количество нажатий одной кнопки либо, наоборот, обеспечить управление системой с помощью одной кнопки. В связи с этим, необходимо понимать как это происходит. Поскольку при удержании кнопки работать с флагами не получится, так как они будут менять свое состояние пока вы не отпустите кнопку, можно применить метод с запоминанием предыдущего состояния кнопки в конце рабочего цикла обработки кнопки.

Вначале работы необходимо записать текущее состояние кнопки, затем проверить, нажата ли она вообще или нет. Если она нажата, то сравниваем её текущее состояние с предыдущим, которое изначально равняется нулю, иначе ждём, пока кнопка не будет нажата. Если текущее состояние кнопки больше предыдущего, это означает, что кнопка в предыдущий такт не была нажата. В таком случае, мы можем увеличивать переменную счётчика, который будет отвечать за количество нажатий на кнопку. Теперь, проверяя, какое значение сейчас хранит счётчик, можно выполнять какие-либо действия, например зажигать светодиод, который будет соответствовать номеру нажатия на кнопку. После всего этого необходимо в предыдущее состояние кнопки записать нынешнее, чтобы потом можно было сравнивать два состояния, предыдущее и нынешнее.

Для лучшего понимания работы алгоритма на рисунке 9 приведена блок-схема алгоритма.

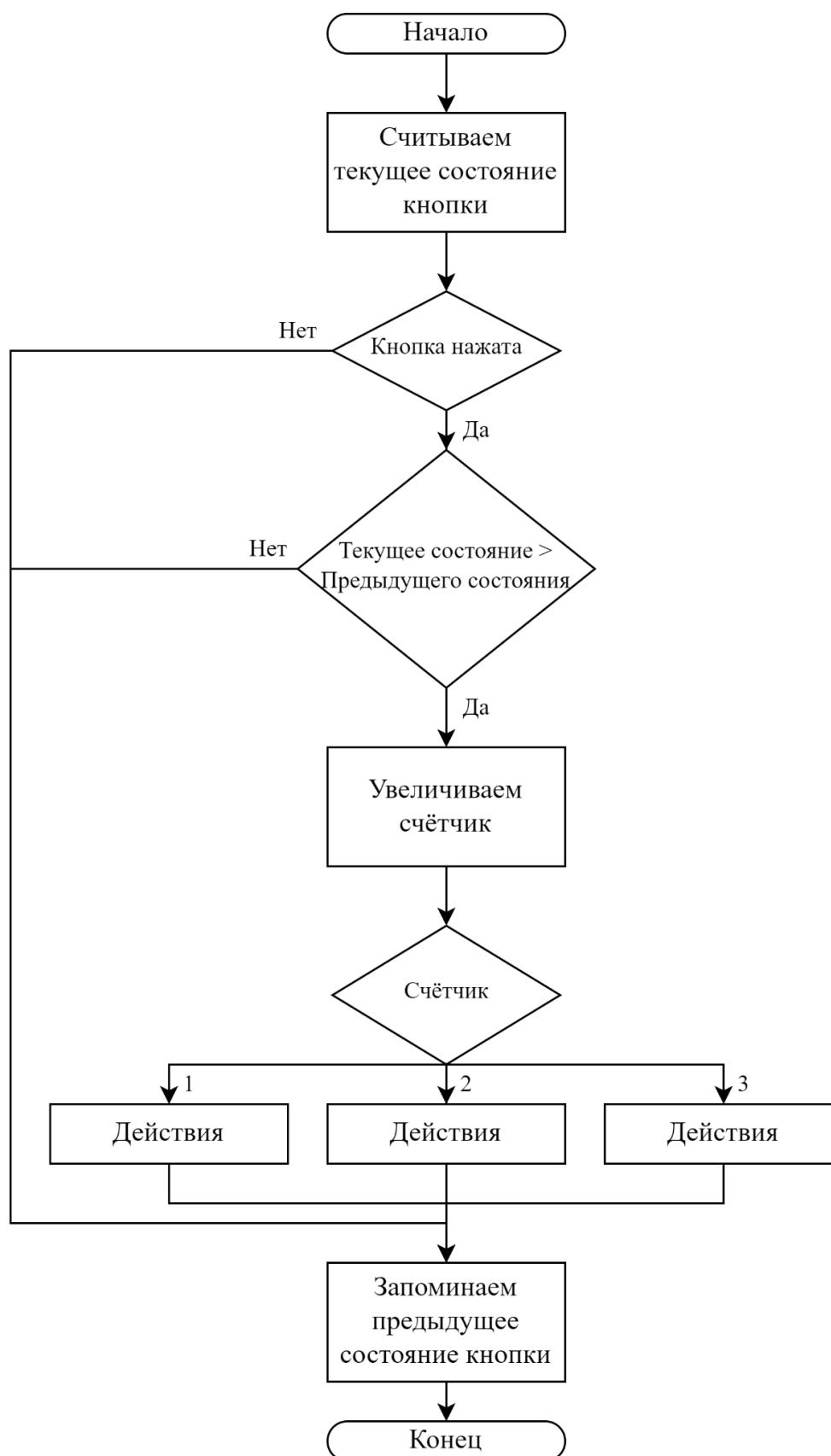


Рисунок 9 - Блок-схема алгоритма работы обработки однократного нажатия кнопки

## Что такое дребезг и как от него избавиться

Когда происходит нажатие на кнопку или на микропереключатель или изменение положения тумблера, два металлических контакта замыкаются. Для пользователя может показаться, что контакт наступил мгновенно, однако это не так. Внутри коммутатора есть движущиеся части. Когда происходит нажатие на коммутатор, он вначале создает контакт между металлическими частями, но только в кратком разрезе микросекунды. Затем он делает контакт немного дольше, а затем еще немного дольше. В конце коммутатор полностью замыкается. Коммутатор скачет (дребезжит) между состояниями наличия и отсутствия контакта. Обычно оборудование работает быстрее, чем дребезг, что приводит к тому, что оборудование думает, что вы нажали на кнопку *несколько раз*.

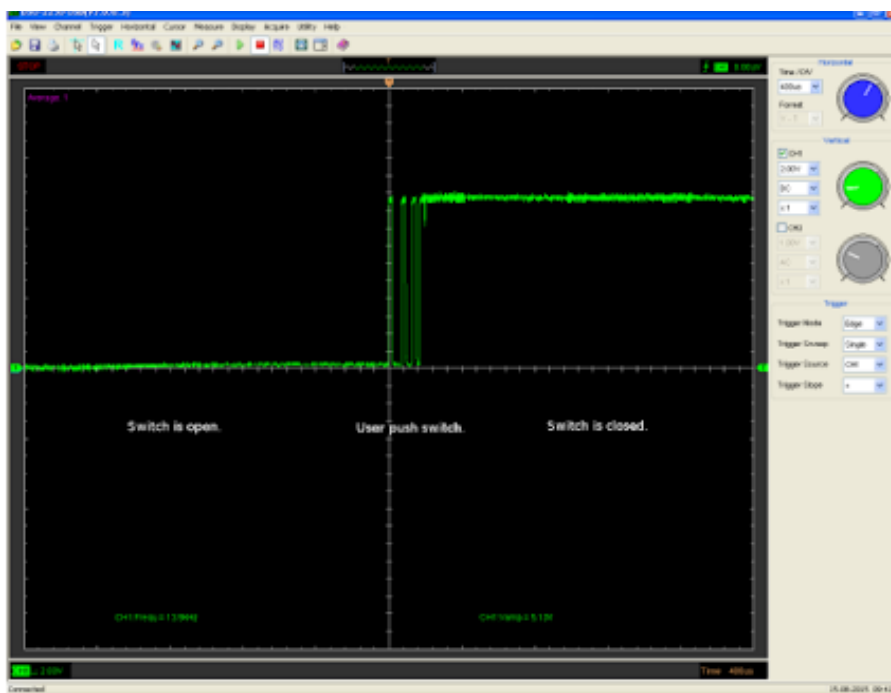


Рисунок 10 - Осциллограмма дребезга контактов

Каждый коммутатор обладает своими собственными характеристиками относительно дребезга. Если сравнивать два одинаковых коммутатора, есть большая вероятность того, что они будут «дребезжать» по-разному.



Существует следующие варианты устранения дребезга, это добавление в цепь конденсатора и программное подавление дребезга, которые подробно описаны по ссылке [4].

## **Задания**

### **1 вариант**

Осуществить управление светодиодами с помощью одной кнопки: при первом нажатии должен загореться 1-ый светодиод, при втором нажатии должен загореться 2-ой светодиод и т.д. После загорания 6-го светодиода, все светодиоды должны погаснуть после нажатия на кнопку. Светодиоды должны быть подключены в один ряд.

### **2 вариант**

Осуществить управление рядом светодиодов, состоящего из 6-ти светодиодов и расположенных в один ряд с помощью 1-ой кнопки. Каждое третье нажатие на кнопку должно зажигать 2 светодиода, остальные нажатия зажигают по 1-му светодиоду.

### **3 вариант**

Осуществить управление светодиодами с помощью двух кнопок: при нажатии на первую кнопку должны светиться 1, 3 и 5 светодиод, при нажатии на вторую кнопку загораются 2, 4 и 6, а первая тройка светодиодов должна погаснуть, при повторном нажатии на первую кнопку должны снова загореться 1,3 и 5 светодиод, а 2,4 и 6 погаснуть и т.д. Светодиоды должны быть подключены в один ряд.

### **4 вариант**

Осуществить управление светодиодами с помощью одной кнопки: изначально все светодиоды горят, при первом нажатии должны погаснуть 1 и 6 светодиоды, при втором нажатии должны погаснуть 2 и 5 светодиоды, затем 3 и 4. После того, как все светодиоды погаснут, они должны все загореться после нажатия кнопки. Светодиоды должны быть подключены в в один ряд.

### **5 вариант**

Осуществить управление светодиодами с помощью одной кнопки: при первом нажатии на кнопку должны светиться 2, 4 и 6 светодиод, при нажатии второй раз загораются 1, 3 и 5, а первая тройка светодиодов должна погаснуть. Светодиоды должны быть подключены в один ряд.

### **6 вариант**

Осуществить управление светодиодами с помощью одной кнопки: при первом нажатии на кнопку должны светиться 2, 4 и 6 светодиод, при втором нажатии на кнопку загораются 1, 3 и 5, а первая тройка светодиодов должна погаснуть. Светодиоды должны быть подключены в два ряда следующим образом:

1 светодиод	2 светодиод	3 светодиод
4 светодиод	5 светодиод	6 светодиод

### **7 вариант**

Осуществить управление рядом светодиодов, состоящего из 6-ти светодиодов и расположенных в один ряд с помощью 1-ой кнопки. Каждое 5-е нажатие на кнопку должно зажигать 2 светодиода, остальные нажатия зажигают по 1-му светодиоду.

### **8 вариант**

Осуществить управление светодиодным рядом из 6-ти светодиодов с помощью 2-х кнопок. При нажатии на одну кнопку светодиоды последовательно загораются, при нажатии на вторую, светодиоды последовательно гаснут по 2 штуки. В том случае, когда происходит нажатие на вторую кнопку во время свечения нескольких светодиодов, они должны гаснуть с того же момента на котором остановилось их зажигание.

### **9 вариант**

Осуществить управление светодиодами с помощью одной кнопки: изначально все светодиоды горят, при первом нажатии должны погаснуть 1 и 6 светодиоды, при втором нажатии должны погаснуть 2 и 5 светодиоды, затем 3 и 4. После того, как все светодиоды погаснут, они должны все загореться после нажатия кнопки. Светодиоды должны быть подключены в два ряда следующим образом:

1 светодиод	2 светодиод	3 светодиод
4 светодиод	5 светодиод	6 светодиод

### **10 вариант**

Осуществить управление светодиодом с помощью двух кнопок: моргание светодиода. При нажатии на первую кнопку частота мигания увеличивается, при нажатии на вторую кнопку частота уменьшается.

### **11 вариант**

Осуществить управление рядом светодиодов, состоящего из 6-ти светодиодов и расположенных в один ряд с помощью 1-ой кнопки. Каждое 6-е нажатие на кнопку должно зажигать 2 светодиода, остальные нажатия зажигают по 1-му светодиоду.

### **12 вариант**

Осуществить управление светодиодами с помощью одной кнопки: при первом нажатии должны загореться 1-ый и 6-ой светодиоды, при втором нажатии должны загореться 2-ой и 5-ый светодиоды, затем 3-ий и 4-ый. После загорания всех светодиодов, все светодиоды должны погаснуть после нажатия кнопки. Светодиоды должны быть подключены в один ряд.

### 13 вариант

Осуществить управление светодиодами с помощью двух кнопок: при нажатии на первую кнопку должны светиться 1, 3 и 5 светодиод, при нажатии на вторую кнопку загораются 2, 4 и 6, а первая тройка светодиодов должна погаснуть. Светодиоды должны быть подключены в два ряда следующим образом:

1 светодиод	2 светодиод	3 светодиод
4 светодиод	5 светодиод	6 светодиод

### 14 вариант

Осуществить управление светодиодами с помощью одной кнопки: при первом нажатии должны загореться 1 и 6 светодиоды, при втором нажатии должны загореться 2 и 5 светодиоды, затем 3 и 4. После загорания всех светодиодов, все светодиоды должны погаснуть после нажатия кнопки. Светодиоды должны быть подключены в два ряда следующим образом:

1 светодиод	2 светодиод	3 светодиод
4 светодиод	5 светодиод	6 светодиод

### 15 вариант

Осуществить управление светодиодами с помощью одной кнопки: изначально все светодиоды горят, при первом нажатии должен погаснуть 1 светодиод, при втором нажатии должен погаснуть 2 светодиод и т.д. После того, как погаснет 6 светодиод, все светодиоды должны загореться после нажатия на кнопку. Светодиоды должны быть подключены в один ряд.

### **16 вариант**

Осуществить управление светодиодами с помощью одной кнопки: при первом нажатии должен загореться 1-ый светодиод, при втором нажатии должен загореться 6-ой светодиод и т.д. до схождения к центру. После загорания 4-го светодиода, все светодиоды должны погаснуть после нажатия на кнопку. Светодиоды должны быть подключены в один ряд.

### **17 вариант**

Осуществить управление светодиодным рядом из 6-ти светодиодов с помощью 2-х кнопок. При нажатии на одну кнопку светодиоды последовательно загораются, при нажатии на вторую, светодиоды последовательно гаснут. В том случае, когда происходит нажатие на вторую кнопку во время свечения нескольких светодиодов, они должны гаснуть с того же момента на котором остановилось их зажигание.

### **18 вариант**

Осуществить управление светодиодным рядом из 6-ти светодиодов с помощью 2-х кнопок. При нажатии на одну кнопку светодиоды последовательно загораются п 2 шт, при нажатии на вторую, светодиоды последовательно гаснут. В том случае, когда происходит нажатие на вторую кнопку во время свечения нескольких светодиодов, они должны гаснуть с того же момента на котором остановилось их зажигание. После того, как все светодиоды загорелись, работа не останавливается, при следующем нажатии на первую кнопку все светодиоды гаснут и загораются заново.

### **Полезные ссылки**

1. Уроки Ардуино - учимся программировать Arduino/ URL:  
[https://alexgyver.ru/arduino\\_lessons/](https://alexgyver.ru/arduino_lessons/);
2. Амперка/ URL: <https://amperka.ru/>;
3. Arduino/ Software/ URL: <https://www.arduino.cc/en/software>;
4. RadioProg/ Дребезг контактов, и как с ним бороться/ URL:  
<https://radioprogram.ru/post/251>;