

# Управление цифровыми входами и выходами



# Проверка связи



## Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти



## Поставьте в чат:

-  если меня видно и слышно
-  если нет

# Вспоминаем прошрое занятие

**Вопрос:** Какая микросхема лежит в основе большинства плат Arduino?



# Вспоминаем прошрое занятие

**Вопрос:** Какая микросхема лежит в основе большинства плат Arduino?

**Ответ:** Микроконтроллер AVR



# Вспоминаем прошрое занятие

**Вопрос:** Какие обязательные функции  
должна содержать программа для Arduino?



# Вспоминаем прошрое занятие

**Вопрос:** Какие обязательные функции  
должна содержать программа для Arduino?

**Ответ:** `setup()` и `loop()`



# Вспоминаем прошрое занятие

**Вопрос:** Какая функция останавливает выполнение программы на заданное число миллисекунд?



# Вспоминаем прошрое занятие

**Вопрос:** Какая функция останавливает выполнение программы на заданное число миллисекунд?

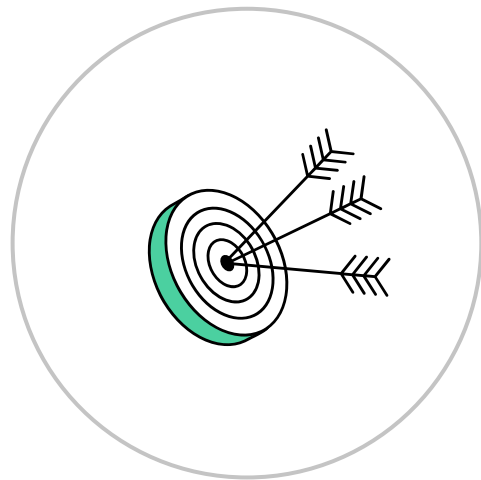
**Ответ:** `delay(ms)`





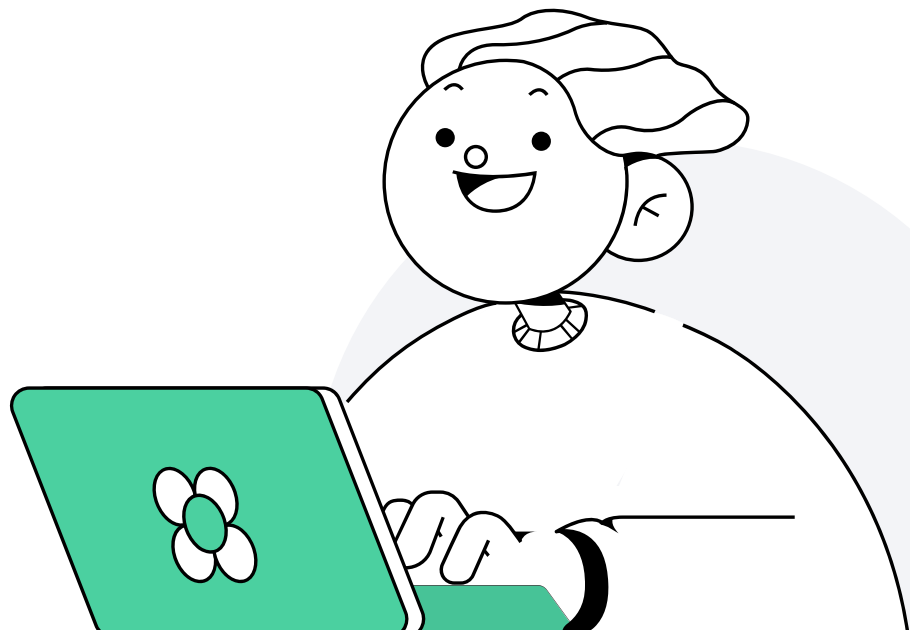
# Цели занятия

- Узнаем, как устроены порты ввода/вывода микроконтроллера
- Познакомимся с эффектомдребезга контактов кнопки
- Узнаем, как управлять яркостью светодиодов с помощью ШИМ
- Научимся создавать собственную библиотеку



# План занятия

- 1 Как устроены порты ввода/вывода в микроконтроллере
- 2 Как обрабатывать сигнал от кнопки
- 3 Как управлять яркостью светодиода
- 4 Как создать свою библиотеку
- 5 Итоги



# Как устроены порты ввода/вывода в микроконтроллере

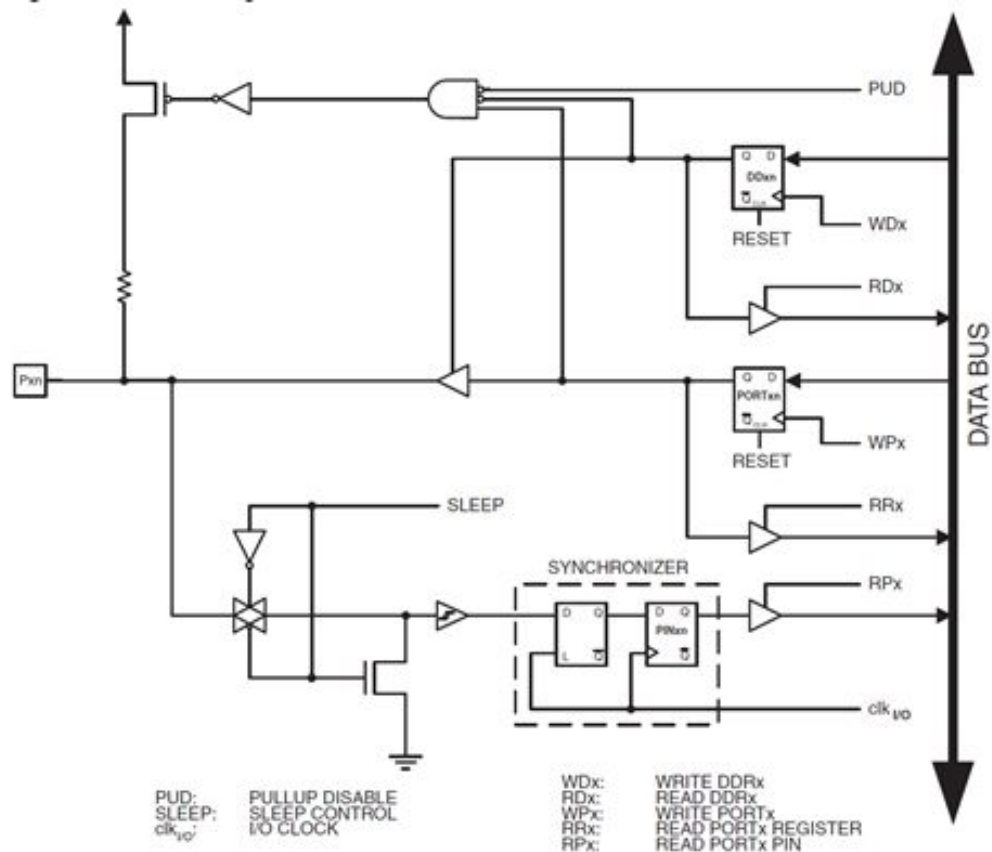


1

# Структура порта ввода/вывода

С внешними элементами микроконтроллер взаимодействует через порты ввода/вывода. Выводы объединяются в группы, которые называют портами. Для управления портами используются регистры (x - имя порта):

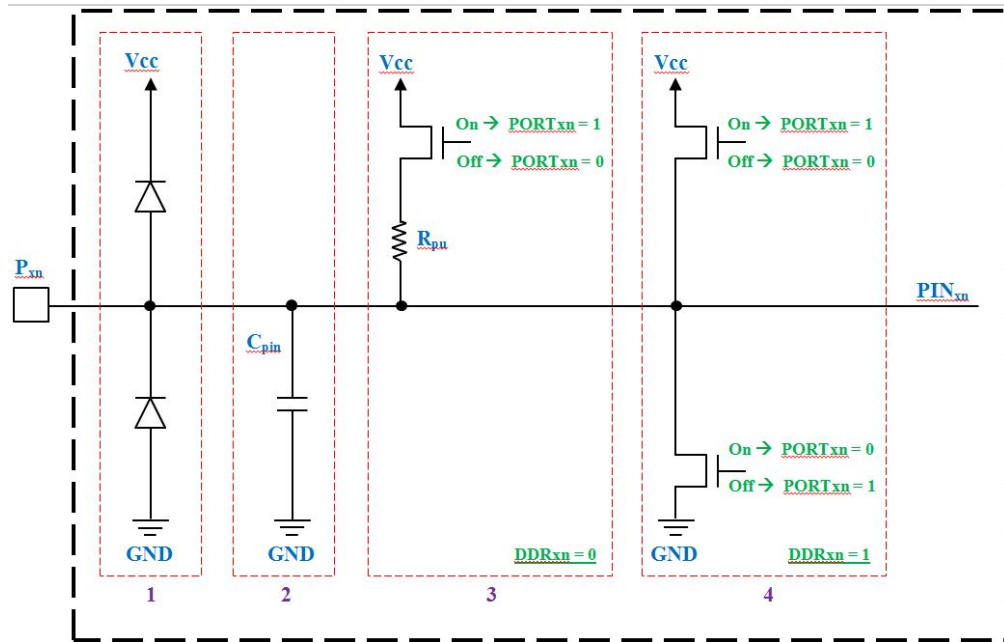
- DDRx - направление данных
- PORTx - управление состоянием вывода
- PINx - состояние на выводе



# Упрощенная структура порта ввода/вывода

Порт содержит:

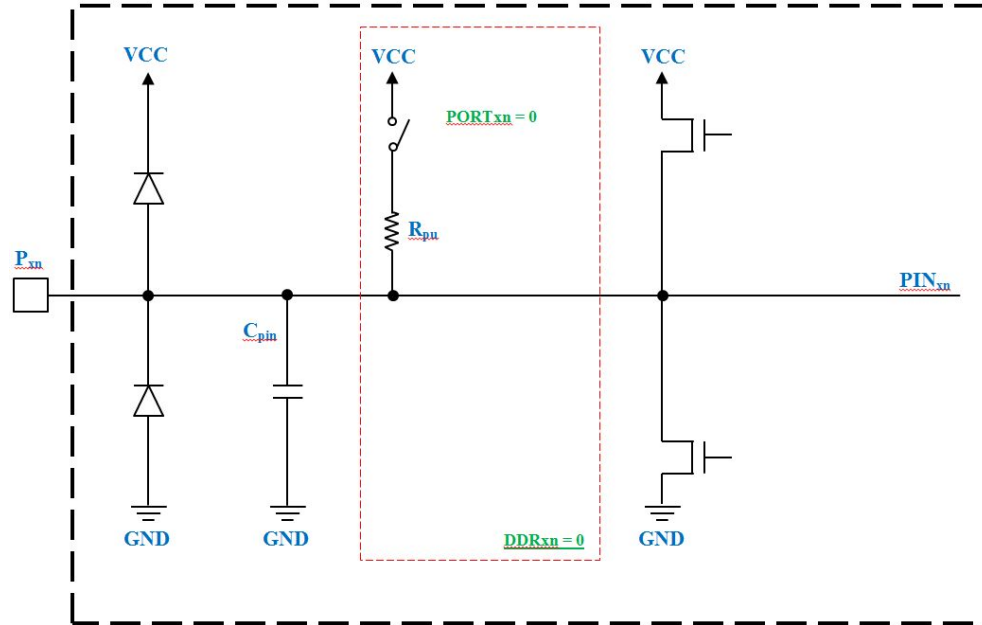
- схему защиты от статического напряжения (1)
- паразитную емкость (2)
- схему включения подтягивающего резистора (3)
- схему задания логического уровня (4)



# Режим высокоимпеданстного входа

Значение регистров (x - имя порта,  
n - номер разряда):

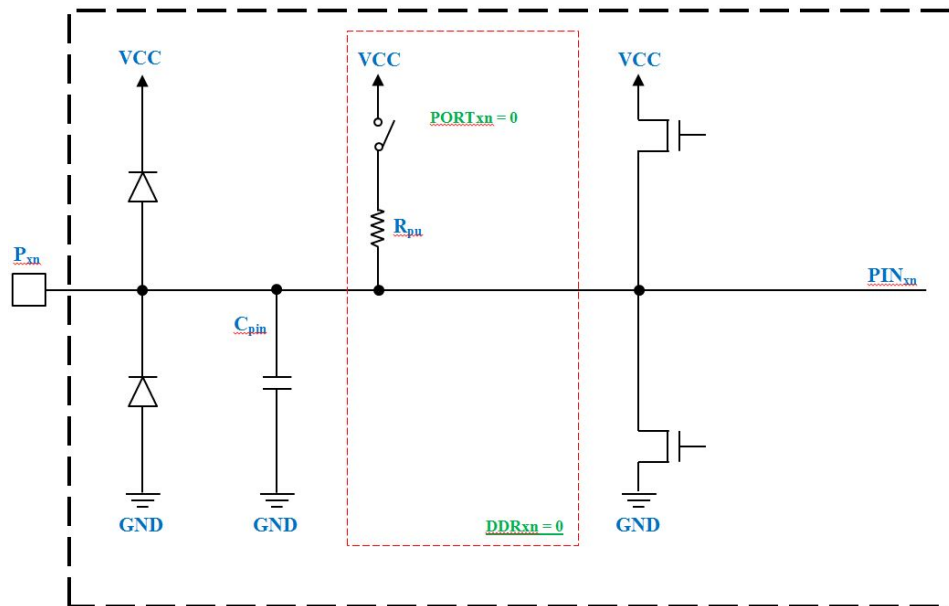
- $DDRx_n = 0$
- $PORTx_n = 0$



# Режим входа с подтяжкой к лог. 1

Значение регистров (x - имя порта, n - номер разряда):

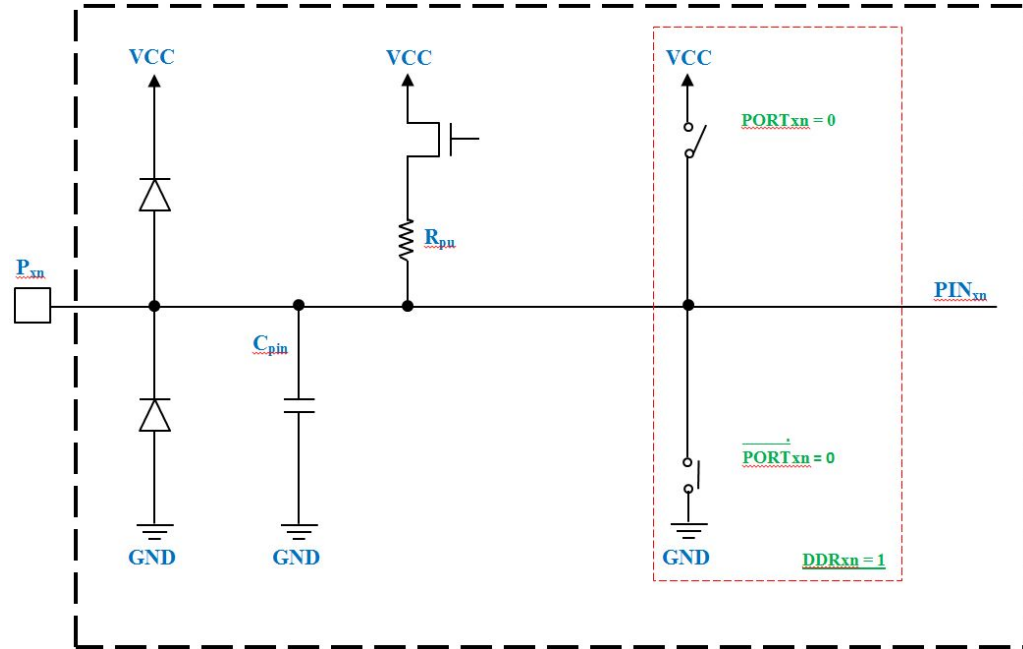
- $DDRx_n = 0$
- $PORTx_n = 1$



# Режим выхода лог. 0

Значение регистров (x - имя порта, n - номер разряда):

- $\text{DDRx}_n = 1$
- $\text{PORTx}_n = 0$

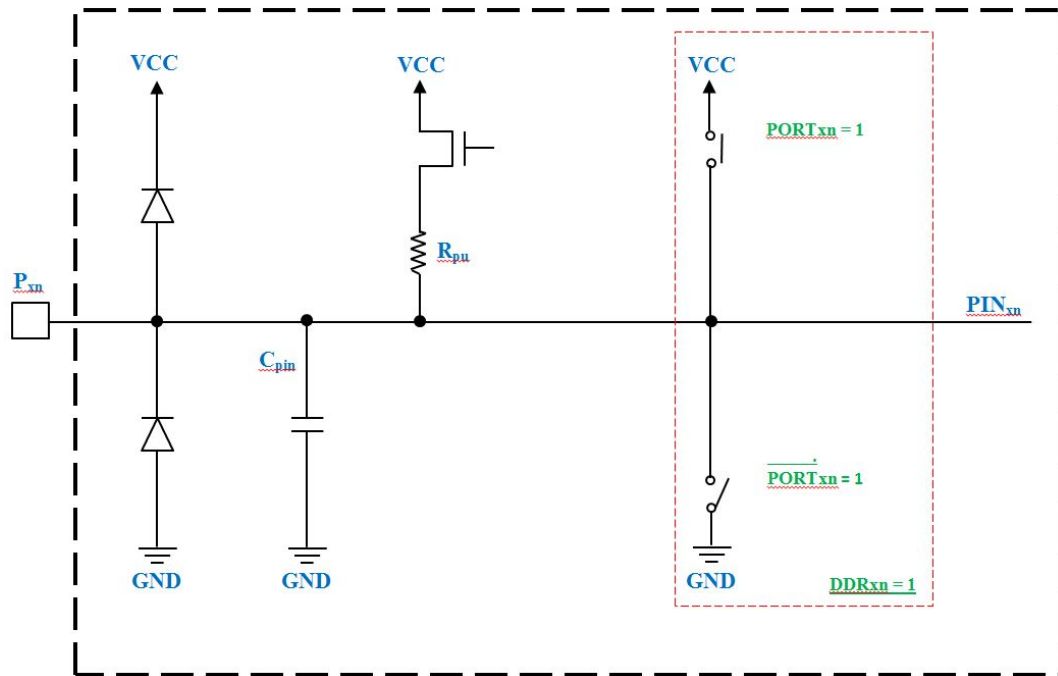




# Режим выхода лог. 1

Значение регистров (x - имя порта, n - номер разряда):

- $\text{DDRx}_n = 1$
- $\text{PORTx}_n = 1$



# Как функции цифрового ввода/вывода связаны с регистрами

`void pinMode(uint8_t pin, uint8_t mode)` — модифицирует значение в соответствующем регистре **DDRx**

`digitalWrite(uint8_t pin, uint8_t value)` — модифицирует значение в соответствующем регистре **PORTx**

`int digitalRead(uint8_t pin)` — считывает значение из соответствующего регистра **PINx**

# Как обрабатывать сигнал от кнопки

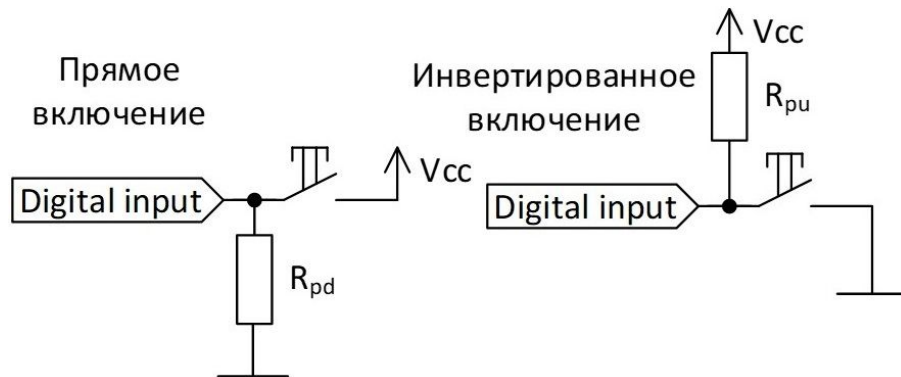
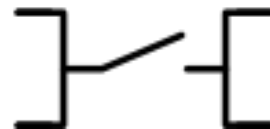


2

# Тактовая кнопка

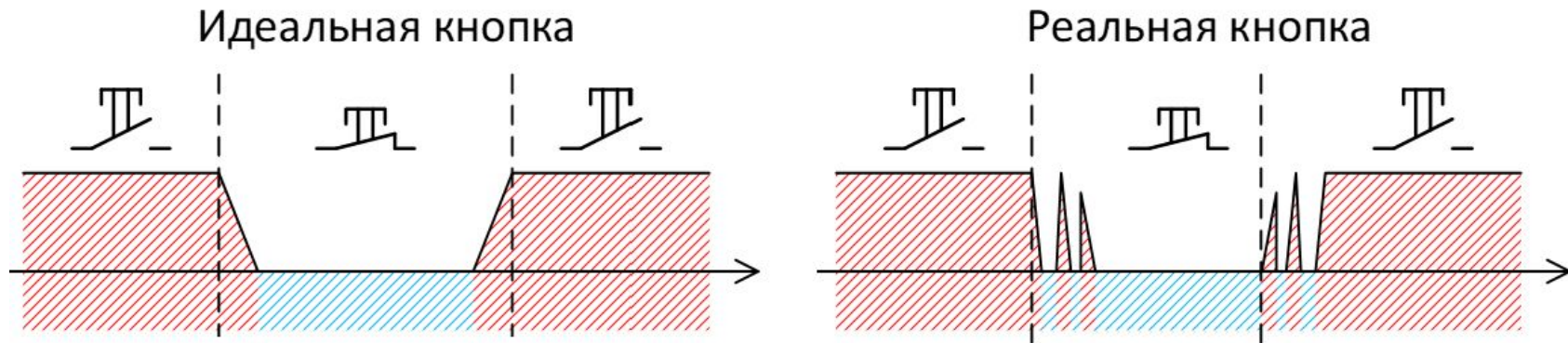
Правильное название: тактильная кнопка (Tactile Button)

Кнопка без фиксации, т.е. после нажатия кнопка возвращается в исходное состояние. Тактильность подразумевает наличие тактильной обратной связи в виде ощущения порога срабатывания кнопки. Также момент срабатывания кнопки может сопровождаться характерным звуковым щелчком.



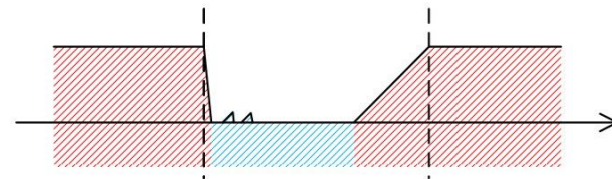
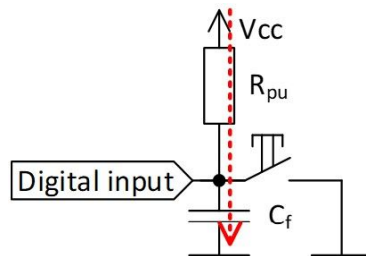
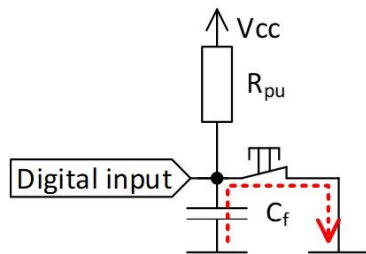
# Дребезг контактов

Сухие контакты реальных кнопок ни когда не замыкаются и не размыкаются мгновенно. Происходит многократные кратковременные замыкания и размыкания



# Аппаратное подавление дребезга контактов

Самый простой способ борьбы с дребезгом контактов, это подключение параллельно кнопки керамического конденсатора обычно до 1 мкФ.



# Программное подавлениедребезга контактов

Принцип программного подавлениядребезга контактов - введение задержки между считыванием состояния линии при первом изменении этого состояния:

```
const int led = 9;           //светодиод подключен к контакту 9
const int button = 2;        //кнопка подключена к контакту 2
int lastButton = LOW;        //предыдущее состояние кнопки
int curButton = LOW;         //Текущее состояние кнопки
int ledOn = LOW;             //Текущее состояние светодиода
```

```
/*Функция подавлениядребезга
```

```
last - предыдущее состояние кнопки*/
```

```
int debounce (int last)
{
    int current = digitalRead(button);
    if(last != current) //если состояние изменилось
    {
        delay(5);
        current = digitalRead(button);
    }
    return current;
}
```

# Программное подавлениедребезга контактов

Программа изменения состояния светодиода по каждому нажатию кнопки:

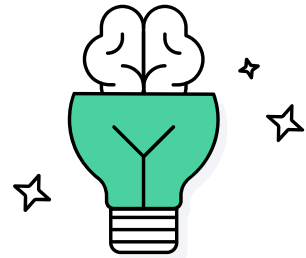
```
void setup()
{
    pinMode(led, OUTPUT);    //контакт светодиода - выход
    digitalWrite(led, LOW);  //на выходе лог. 0

    pinMode(button, INPUT);  //контакт кнопки - вход
    digitalWrite(button, HIGH); //подтягивающий резистор на входе
}

void loop()
{
    curButton = debounce (lastButton);
    if(lastButton == HIGH && curButton == LOW) //условие нажатия
    {
        ledOn = !ledOn;
        digitalWrite(led, ledOn);
    }
    lastButton = curButton;
}
```



# Практическое задание №1



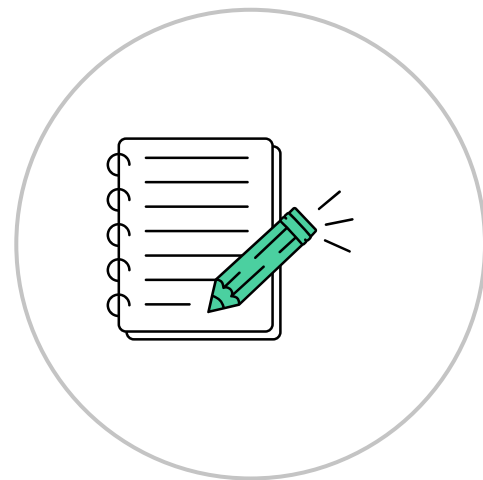
# Практика: подавление дребезга контактов

## Задание:

- 1) соберите схему в симуляторе WOKWI, подключив светодиод к выводу 9, а кнопку - к выводу 2;
- 2) создайте скетч с текстом, приведенным выше;
- 3) проведите моделирование работы

**Как выполнять:** напишите в чат об удачной работе схемы

**Время выполнения:** 5 минут



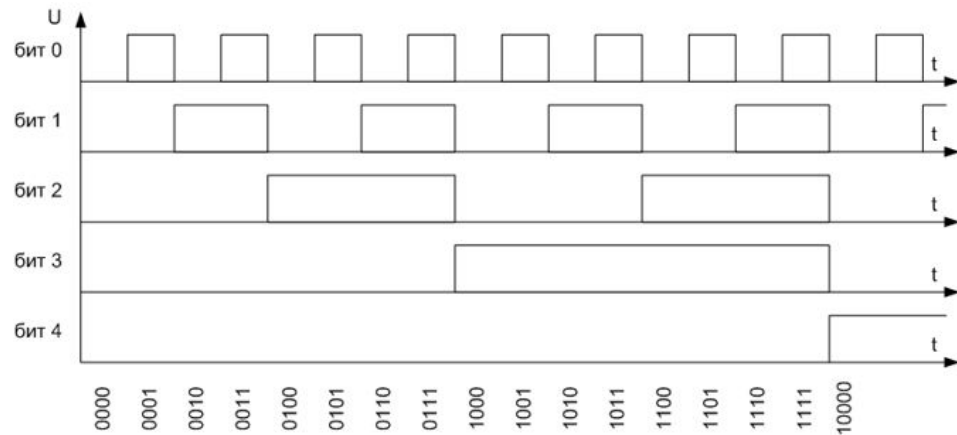
# Как управлять яркостью светодиода



3

# Таймеры микроконтроллера

Таймером называется средство микроконтроллера, служащее для измерения времени и реализации задержек. Основой таймера служит суммирующий счетчик, который считает количество входных импульсов

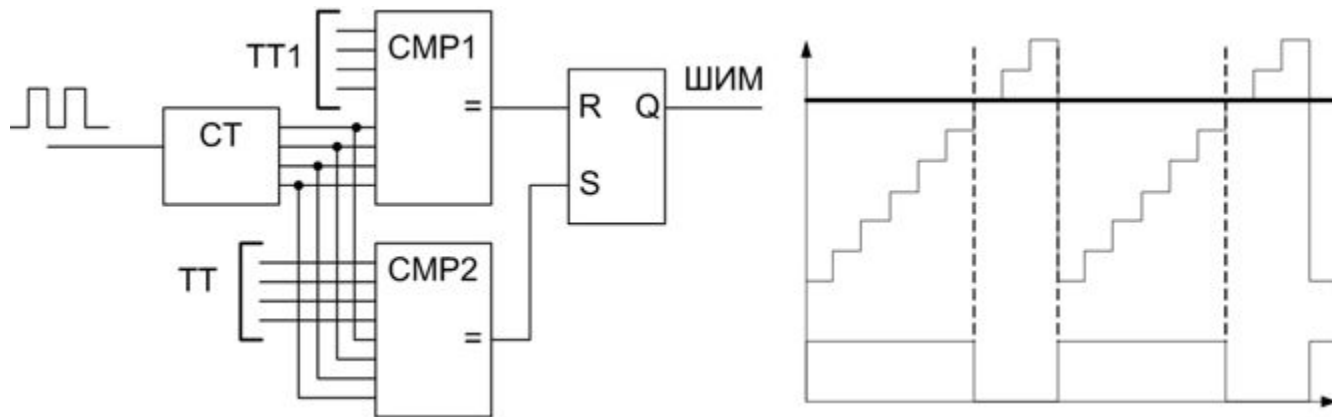


# Широтно-импульсная модуляция

- **Широтно-импульсная модуляция (ШИМ)** – импульсный сигнал постоянной частоты и переменной скважности.
- **Скважность** - отношения периода следования импульса к длительности импульса. С помощью задания скважности (длительности импульсов) можно менять среднее напряжение на выходе ШИМ.
- Обратная величина, то есть отношение длительности импульса к периоду, называется **коэффициентом заполнения**.

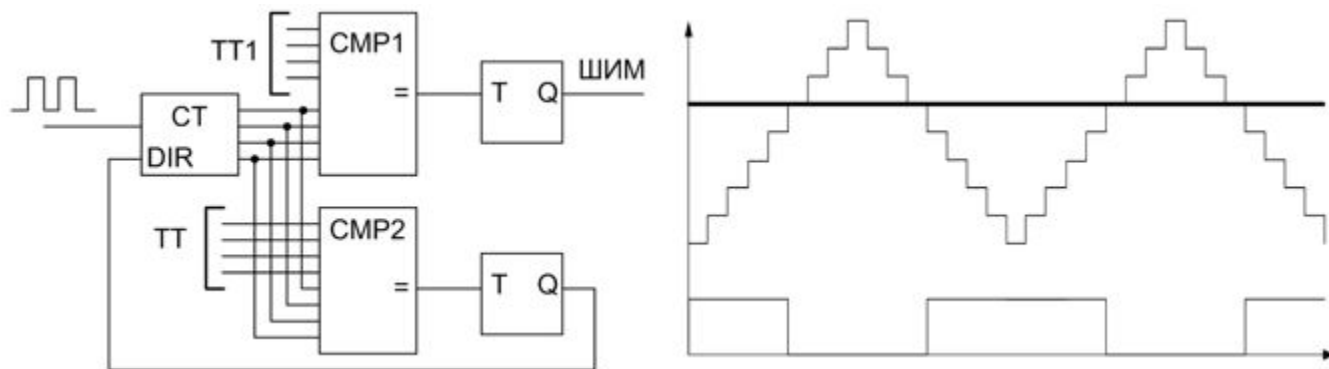
# Быстрый ШИМ

Период ШИМ определяется максимальным значением, до которого считает счетчик. В этот момент ШИМ-сигнал устанавливается в «1». При достижении счетчиком значения, поданного на второй вход цифрового компаратора, осуществляется сброс выходного ШИМ-сигнала.



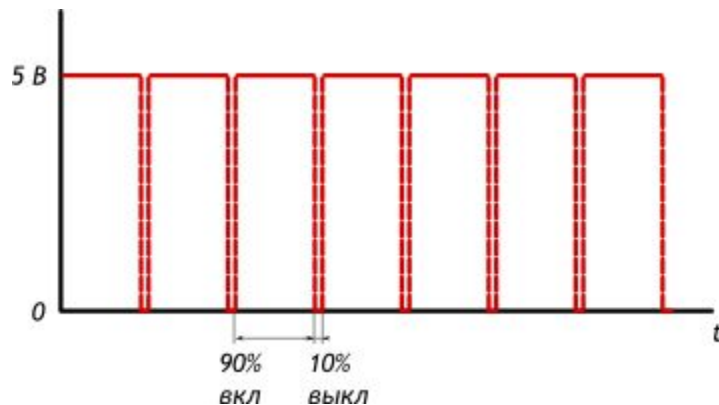
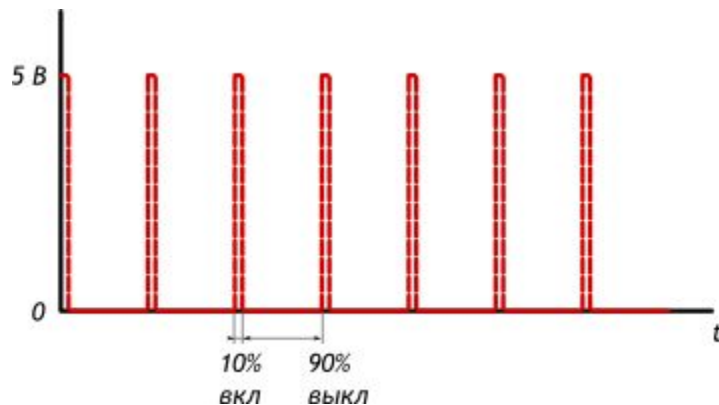
# Фазовый ШИМ

В данном режиме счетчик работает как суммирующий и считает от 0 до максимального значения, а при достижении максимального значения работает как вычитающий, считая до 0.



# Изменение скважности ШИМ сигнала

Если сигнал ШИМ пропустить через фильтр низких частот (ФНЧ), то на выходе фильтра мы получим аналоговый сигнал, напряжение которого пропорционально коэффициенту заполнения ШИМ.





# Функция формирования сигнала с ШИМ

`void analogWrite(uint8_t pin, int val)` — формирует ШИМ сигнал на порте ввода/вывода. После вызова `analogWrite()` на выходе будет генерироваться периодический сигнал с частотой примерно 490 Гц с заданной шириной импульса до следующего вызова `analogWrite()` (или вызова `digitalWrite()` или `digitalRead()` на том же порту вход/выхода)

Параметры:

- `pin`: номер вход/выхода(`pin`), допустимые значения для большинства плат: 3, 5, 6, 9, 10, 11
- `value`: период рабочего цикла значение между 0 (полностью выключено) and 255 (сигнал подан постоянно)

Возвращаемое значение: нет

# Плавное изменение яркости светодиода

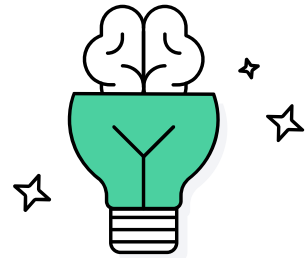
Человеческий глаз выступает как ФНЧ для поступающего на него светового потока, поэтому яркость светодиода можно задавать с помощью ШИМ:

```
const int led = 9;           //светодиод подключен к контакту 9
void setup()
{
    pinMode(led, OUTPUT);    //контакт светодиода - выход
    digitalWrite(led, LOW);  //на выходе лог. 0
}

void loop()
{
    for(int i = 0; i < 256; i++)
    {
        analogWrite(led,i);
        delay(10);
    }

    for(int i = 255; i >= 0; i--)
    {
        analogWrite(led,i);
        delay(10);
    }
}
```

# Практическое задание №2



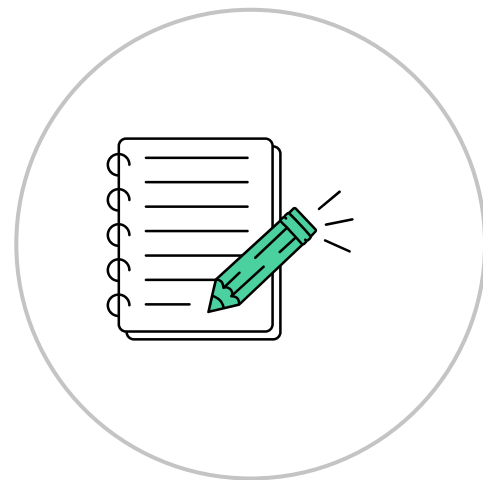
# Практика: подавление дребезга контактов

## Задание:

- 1) соберите схему в симуляторе WOKWI, подключив светодиод к выводу 9;
- 2) создайте скетч с текстом, приведенным выше;
- 3) проведите моделирование работы

**Как выполнять:** напишите в чат об удачной работе схемы

**Время выполнения:** 5 минут



# Как создать свою библиотеку



4



**Библиотека – это набор текстовых файлов с кодом, который можно подключить в свой скетч и использовать имеющиеся там команды.**

Библиотека может иметь несколько файлов или даже папок с файлами, но подключается всегда один – главный заголовочный файл с расширением .h, а он в свою очередь подтягивает остальные необходимые файлы.

# Библиотека для Arduino

В общем случае библиотека имеет такую структуру (название библиотеки testLib):

- testLib – папка библиотеки
  - examples – папка с примерами (необязательно)
  - testLib.h – заголовочный файл
  - testLib.cpp – файл реализации
  - keywords.txt – карта подсветки синтаксиса (необязательно)

# Пример преобразования кода в библиотеку

Скетч, воспроизводящий код Морзе:

```
int pin = 13;      //встроенный светодиод

void setup()
{
    pinMode(pin, OUTPUT);
}

void loop()
{
    dot(); dot(); dot();
    dash(); dash(); dash();
    dot(); dot(); dot();
    delay(3000);
}
```

```
/*Функция формирования точки с помощью светодиода*/
void dot()
{
    digitalWrite(pin, HIGH);
    delay(250);
    digitalWrite(pin, LOW);
    delay(250);
}

/*Функция формирования тире с помощью светодиода*/
void dash()
{
    digitalWrite(pin, HIGH);
    delay(1000);
    digitalWrite(pin, LOW);
    delay(250);
}
```



# Содержимое заголовочного файла

Заголовочный файл содержит класс, в котором объявляются функций и используемые переменные:

```
/*
    Morse.h - Библиотека для формирования кода Морзе на светодиоде
*/
#ifndef Morse_h
#define Morse_h

#include "WProgram.h"

class Morse
{
public:
    Morse(int pin);
    void dot();
    void dash();
private:
    int _pin;
};

#endif
```

# Содержимое файла реализации библиотеки

Файл реализации библиотеки содержит описание конструктора, деструктора (при необходимости) и методов класса:

```
/*
    Morse.cpp - Библиотека для формирования кода Морзе на
    светодиоде
*/

#include "WProgram.h"    //стандартные типы для Arduino
#include "Morse.h"       //заголовочный файл самой библиотеки

Morse::Morse(int pin)
{
    pinMode(pin, OUTPUT);
    _pin = pin;
}
```

```
void Morse::dot()
{
    digitalWrite(_pin, HIGH);
    delay(250);
    digitalWrite(_pin, LOW);
    delay(250);
}

void Morse::dash()
{
    digitalWrite(_pin, HIGH);
    delay(1000);
    digitalWrite(_pin, LOW);
    delay(250);
}
```

# Использование библиотеки

Изначальный скетч, переписанный с использованием созданной библиотеки:

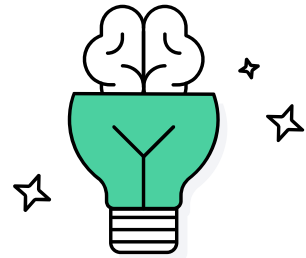
```
#include <Morse.h>

Morse morse(13);

void setup()
{
}

void loop()
{
  morse.dot(); morse.dot(); morse.dot();
  morse.dash(); morse.dash(); morse.dash();
  morse.dot(); morse.dot(); morse.dot();
  delay(3000);
}
```

# Практическое задание №3



# Практика: мигаем светодиодом в симуляторе

## Задание:

- 1) создайте библиотеку Morse в виде двух файлов .h и .cpp, помещенных в папку **Morse**
- 2) разместите эту папку в папке **libraries** Arduino IDE.
- 3) создайте новый скетч с текстом программы, приведенным на предыдущем слайде и выполните его компиляцию.

**Как выполнять:** напишите в чат о результатах компиляции

**Время выполнения:** 10 минут



# Итоги



6

# Итоги занятия

Сегодня мы

- 1 Узнали устройство порта ввода/вывода микроконтроллера
- 2 Научились подавлятьдребезг контактов от кнопки
- 3 Научились управлять яркостью светодиода с помощью ШИМ
- 4 Создали собственную библиотеку



# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи





**Задавайте вопросы  
и пишите отзыв о лекции**

