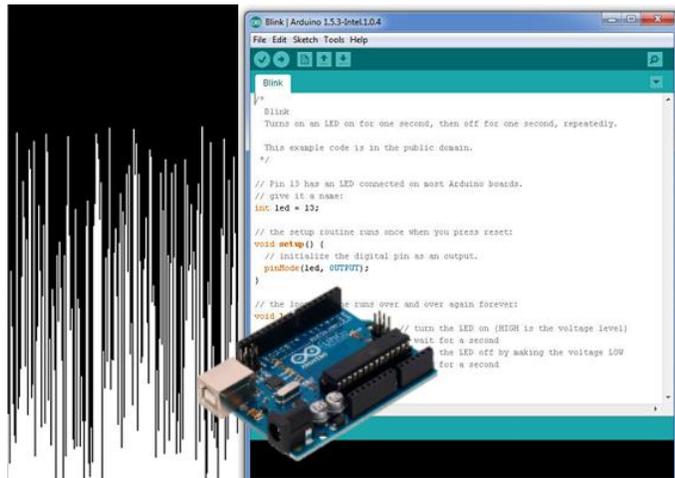




# Программирование микроконтроллеров на C++



Занятие 1

**Основы работы с платформой Arduino.  
Взаимодействие с цифровыми портами.**

## Микроконтроллер Atmega 328p

Тактовая частота: 16МГц

Объём Flash-памяти: 32 кб

Объём SRAM-памяти: 2 кб

Объём EEPROM-памяти: 1 кб

Напряжение питания: 1,8 – 5,5 В

Потребляемый ток в режиме работы: 0,2 мА (1 МГц, 1,8 В)

Потребляемый ток в режиме сна: 0,75 мкА (1 МГц, 1,8 В)

Количество таймеров/счётчиков: 2 восьмибитных, 1 шестнадцатибитный

Общее количество портов: 23

Количество ШИМ (PWM) выходов: 6

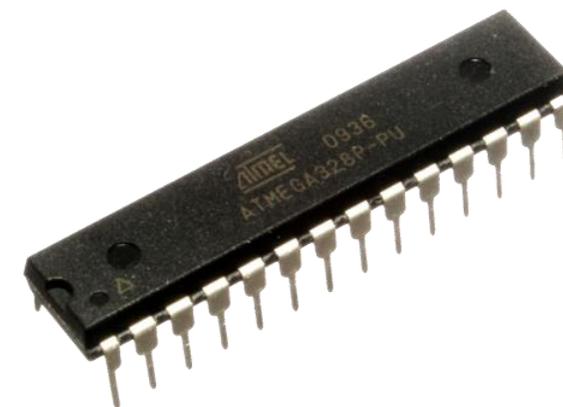
Количество каналов АЦП (аналоговые входы): 6

Количество аппаратных USART (Serial): 1

Количество аппаратных SPI: 1 Master/Slave

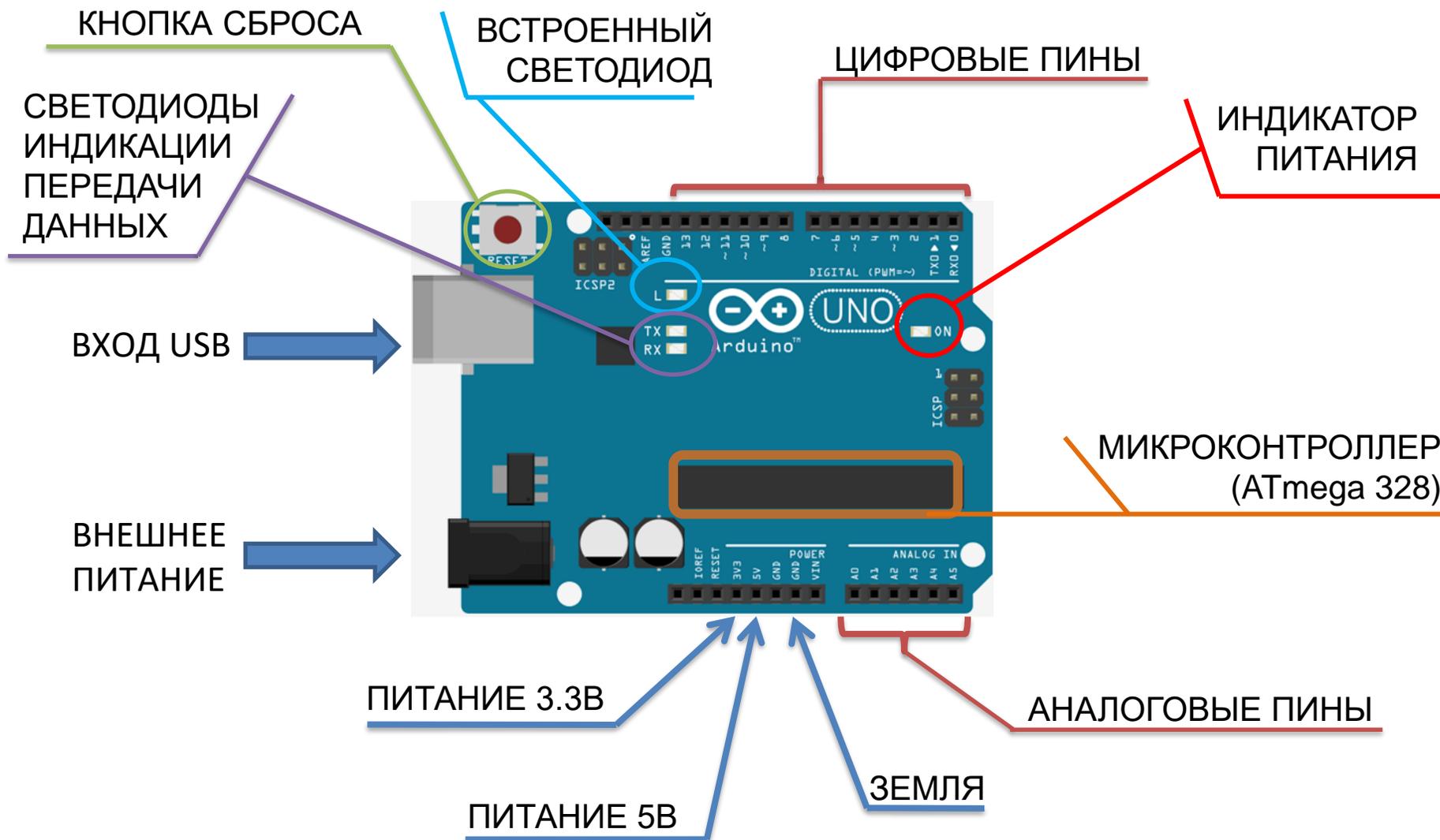
Количество аппаратных I<sup>2</sup>C/SPI: 1

Разрешение АЦП: 10 бит

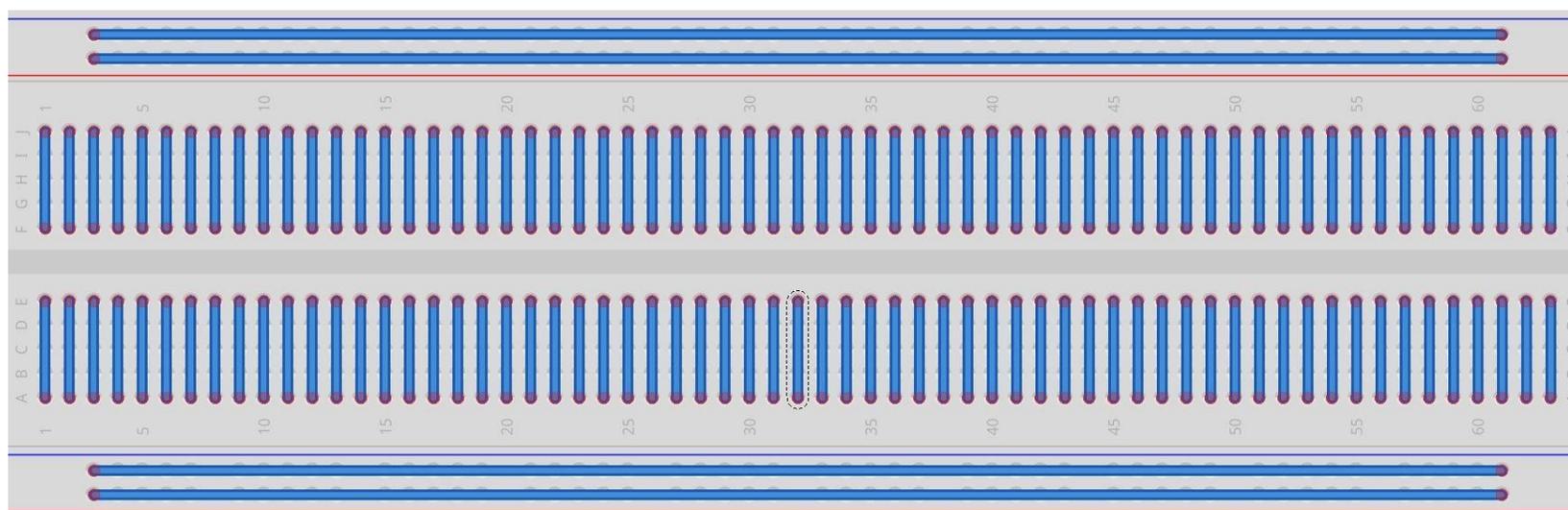
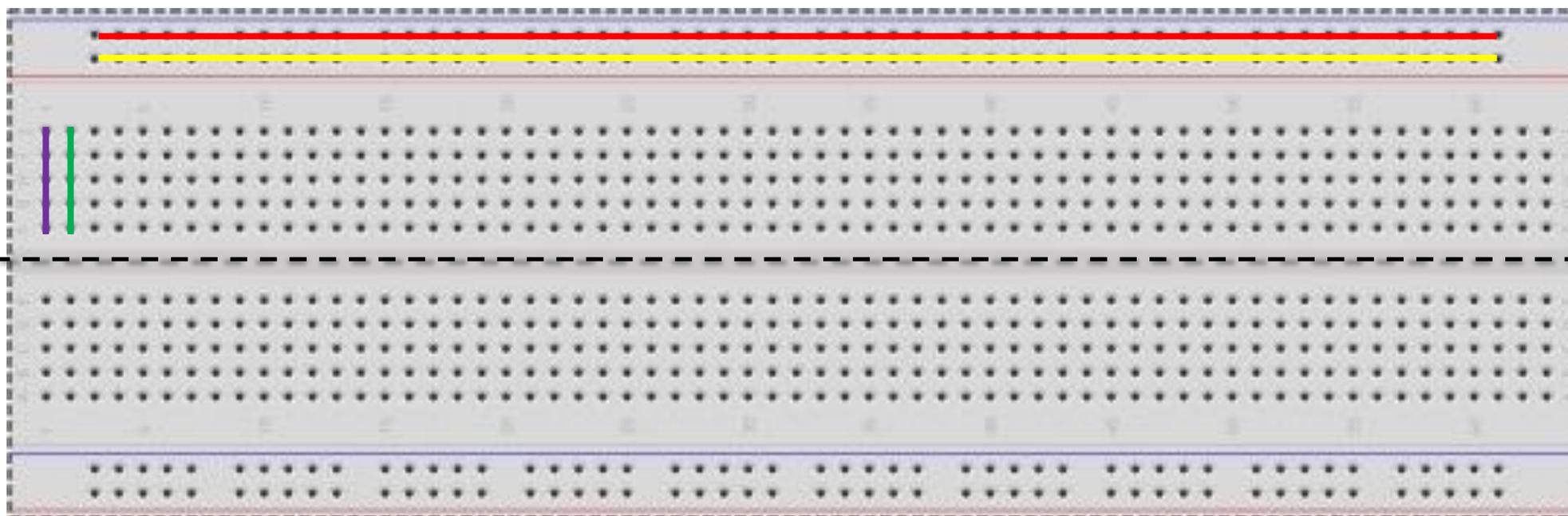




# Выходы платы Arduino UNO



## Устройство макетной платы



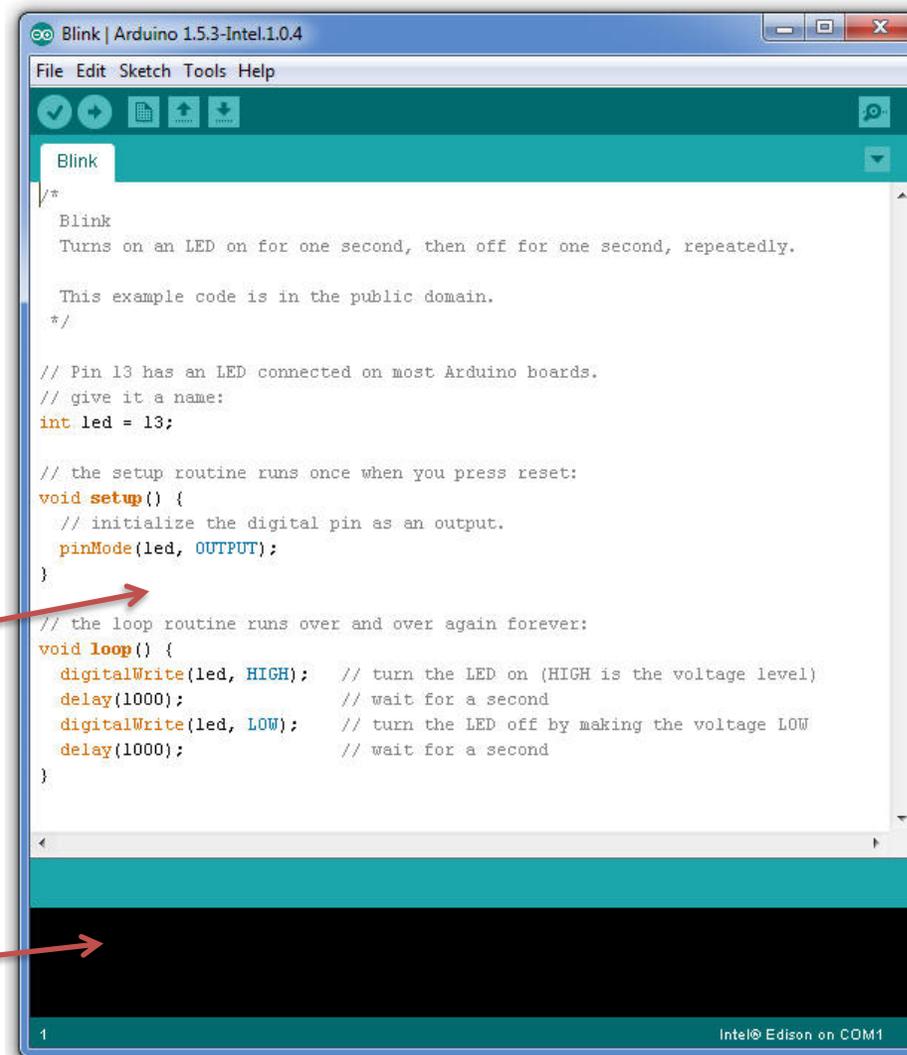
## Редактор программного кода

МЕНЮ

ПАНЕЛЬ  
ИНСТРУМЕНТОВ

РЕДАКТОР  
ПРОГРАММНОГО  
КОДА

ОКНО ВЫВОДА  
СООБЩЕНИЙ  
КОМПИЛЯТОРА И  
ЗАГРУЗЧИКА



## Язык программирования Arduino - C++

Функция вызывается  
всего 1 раз при  
запуске Arduino



```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

Функция вызывается  
постоянно раз за  
разом, пока Arduino  
работает



```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

## Поддерживаемые конструкции

Операторы и функции C++

```
if ... else  
for, while, do ... while  
switch  
break  
continue  
return  
goto
```

+

Математические и логические операторы

+

Типы данных: встроенные и пользовательские

Функции для работы с портами:

```
pinMode()
```

```
digitalRead(), digitalWrite()
```

```
analogRead(), analogWrite()
```

Функции для работы со временем:

```
delay()
```

```
delayMicroseconds()
```

```
millis()
```

```
micros()
```

Макросы:

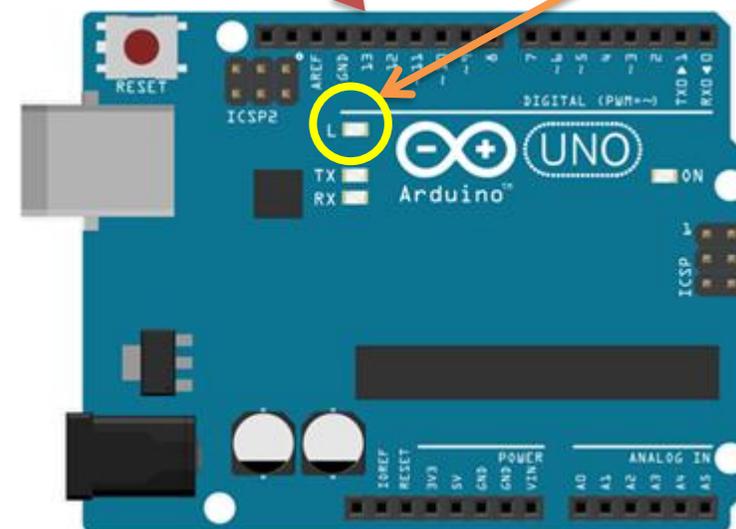
```
INPUT, OUTPUT, HIGH, LOW
```

## Первая программа: мигаем светодиодом

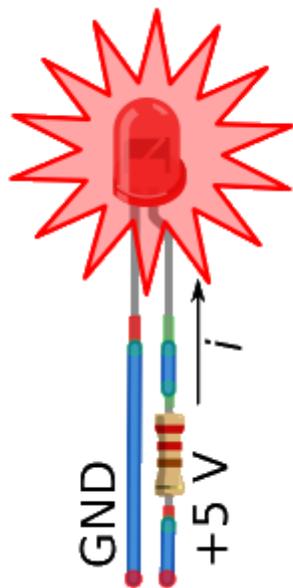
```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

Цифровой пин № 13

Встроенный светодиод



# Расчёт параметров элементов



amperka.ru/product/led-5mm

Москва, Большой Дворянской 6, +7 499 322-13-17  
пн-вс, 9:00-21:00

амперка

ВИКИ ВИДЕО ФОРУМ ДОСТАВКА ОПЛАТА НАПИСАТЬ НАМ

Главная / Радиодетали / Светодиоды / Светодиоды 5 мм (4 шт.)

## Светодиоды 5 мм (4 шт.)

арт AMP-X009-R4

Следующее изображение

Наличными

Visa / MasterCard

Курьер

Самовывоз

Красный

40 Р

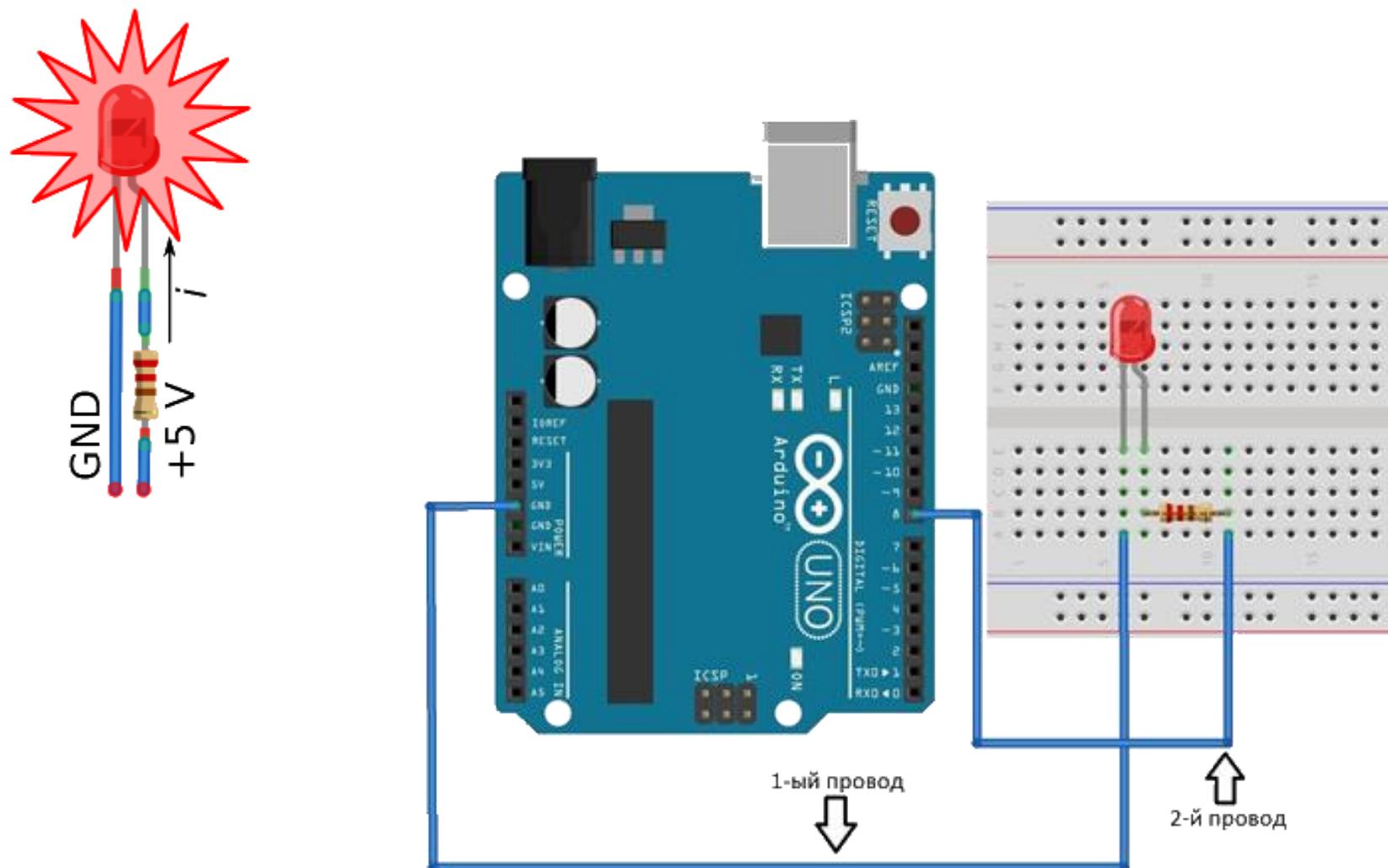
В КОРЗИНУ

В одной упаковке – 4 штуки.

Для одновременного подключения большого количества независимых светодиодов к управляющей электронике, если не хочется занимать большое количество пинов, рассмотрите [выходной сдвиговый регистр](#).

|                             | Красный | Жёлтый | Зелёный | Синий | Белый |
|-----------------------------|---------|--------|---------|-------|-------|
| Макс. прямое напряжение (В) | 2,3     | 2,5    | 2,5     | 3,8   | 3,8   |
| Сила тока (мА)              | 20      | 20     | 20      | 20    | 20    |

## Использование макетных плат



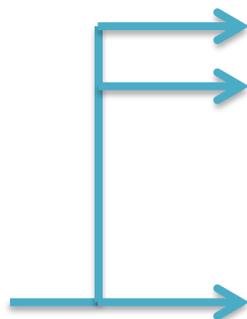
## Вывод данных из Arduino

Инициализация  
порта  
последовательного  
соединения



```
void setup() {  
  Serial.begin(9600);  
  
  delay(1000);
```

Вывод данных в  
последовательный  
порт



```
  Serial.println("Serial is OK");  
  Serial.println("=====");  
}  
  
void loop() {  
  Serial.println("I'm in the loop!");  
  delay(1000);  
}
```

## Программирование портов Atmega 328p

Порты Arduino:

- порт В (цифровые пины с 8 по 13);
- порт С (аналоговые входные пины);
- порт D (цифровые пины с 0 по 7).

Общение с портами напрямую возможно через регистры (на примере порта В):

- регистр **DDRB** - The Port B Data Direction Register - чтение и запись;  
0 – INPUT  
1 – OUTPUT
- регистр **PORTB** - The Port B Data Register - чтение и запись;  
0 – LOW  
1 – HIGH
- регистр **PINB** - The Port B Input Pins Register - только чтение.

```
pinMode(13, OUTPUT);
```



```
DDRB = B00100000;
```

```
digitalWrite(13, HIGH);
```



```
PORTB = B00100000;
```

Плохой вариант!!!



не надо  
так

## Программирование Atmega 328p на уровне портов (1)

Порты Arduino:

- порт В (цифровые пины с 8 по 13);
- порт С (аналоговые входные пины);
- порт D (цифровые пины с 0 по 7).

Общение с портами напрямую возможно через регистры (на примере порта С):

- регистр **DDRB** - The Port B Data Direction Register – чтение и запись;  
0 – INPUT  
1 – OUTPUT
- регистр **PORTB** - The Port B Data Register – чтение и запись;  
0 – LOW  
1 – HIGH
- регистр **PINB** - The Port B Input Pins Register – только чтение.

```
pinMode(13, OUTPUT);
```



```
DDRB = B00100000;
```

```
DDRB = DDRB | B00100000;
```

```
DDRB |= B00100000;
```

```
DDRB |= (1 << 5);
```

## Программирование Atmega 328p на уровне портов (2)

Порты Arduino:

- порт В (цифровые пины с 8 по 13);
- порт С (аналоговые входные пины);
- порт D (цифровые пины с 0 по 7).

Общение с портами напрямую возможно через регистры (на примере порта С):

- регистр **DDRB** - The Port B Data Direction Register - чтение и запись;  
0 – INPUT  
1 – OUTPUT
- регистр **PORTB** - The Port B Data Register - чтение и запись;  
0 – LOW  
1 – HIGH
- регистр **PINB** - The Port B Input Pins Register - только чтение.

```
digitalWrite(13, HIGH);
```



```
PORTB = B00100000;
```

```
PORTB = PORTB | B00100000;
```

```
PORTB |= B00100000;
```

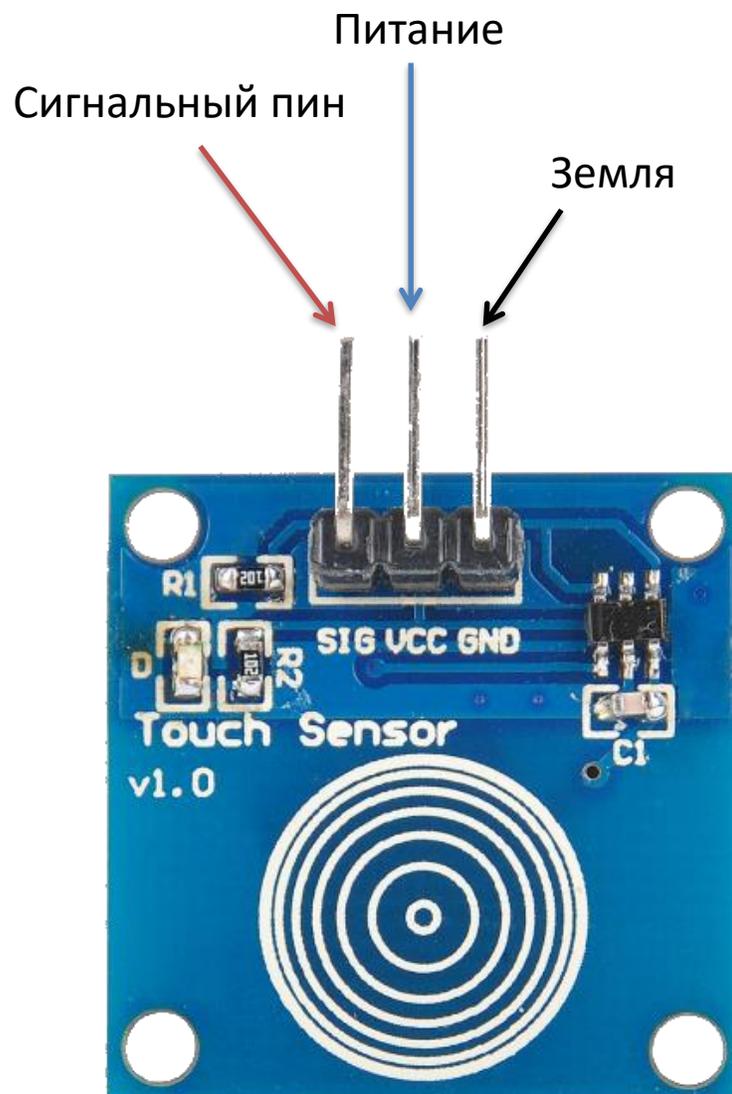
```
PORTB |= (1 << 5);
```

```
digitalWrite(13, LOW);
```



```
PORTB &= B11011111;
```

## Считывание данных с цифровых портов



```
#define PIN_LED 10
```

```
#define PIN_SENSOR 8
```

```
void setup() {  
  pinMode(PIN_LED, OUTPUT);  
  pinMode(PIN_SENSOR, INPUT);  
}
```

```
void loop() {  
  int val = digitalRead(PIN_SENSOR);  
  
  if (val == HIGH)  
    digitalWrite(PIN_LED, HIGH);  
  else  
    digitalWrite(PIN_LED, LOW);  
}
```



## Внешние аппаратные прерывания (1)

```
attachInterrupt(int interrupt, void (*isr)(), int mode)
```

Прерывание (interrupt):

| Плата    | 0 | 1 | 2  | 3  | 4  | 5  |
|----------|---|---|----|----|----|----|
| UNO      | 2 | 3 |    |    |    |    |
| Mega2560 | 2 | 3 | 21 | 20 | 19 | 18 |
| Leonardo | 3 | 2 | 0  | 1  | 7  |    |

Обработчик прерывания (interrupt service routine, ISR):

```
void ISR_func() {  
}
```

Внутри ISR:

1. не работает `delay()`;
2. значения, возвращаемые `millis()`, не изменяются;
3. возможна потеря данных Serial-соединения;
4. переменные, изменяемые в функции, должны быть объявлены как `volatile`.

Режим прерывания:

- LOW
- CHANGE
- RISING
- FALLING

## Внешние аппаратные прерывания (2)



```
#define PIN_SENSOR 2
```



```
void setup() {  
  pinMode(PIN_LED, OUTPUT);  
  pinMode(PIN_SENSOR, INPUT);  
  pinMode(13, OUTPUT);
```



```
  attachInterrupt(0, ISR_func, CHANGE);  
}
```



```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```



```
void ISR_func() {  
  int val = digitalRead(PIN_SENSOR);  
  digitalWrite(PIN_LED, val);  
}
```