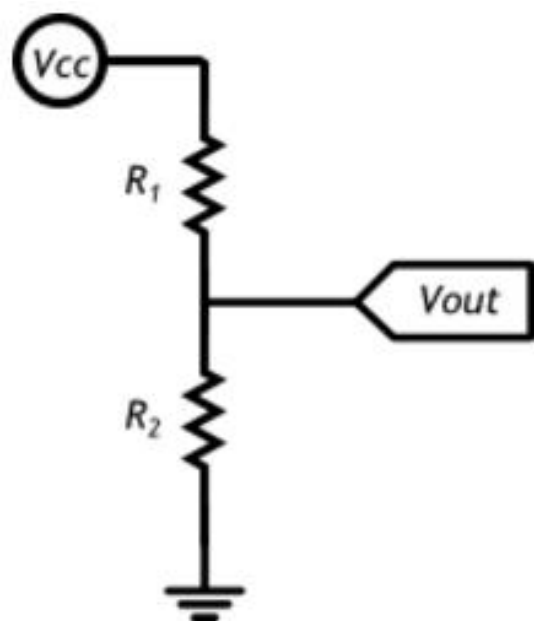


# Делитель напряжения

# Делитель напряжения



$$V_{out} = \frac{R_2 V_{cc}}{R_1 + R_2}$$

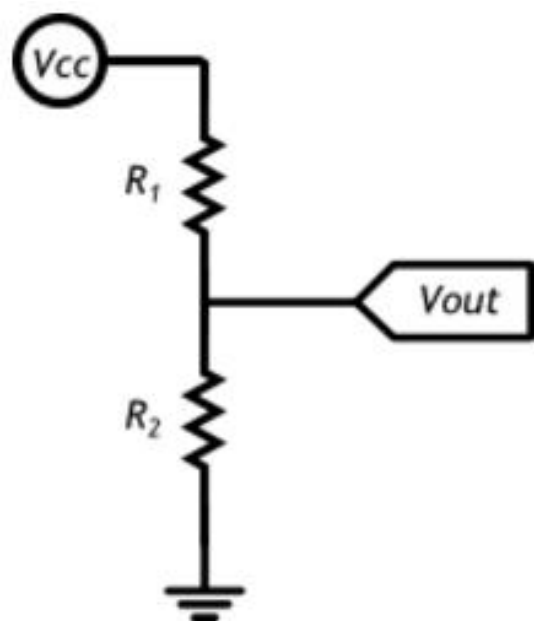
- $V_{out}$  — напряжение, которое считывает аналоговый порт

Пример 1:

$$V_{cc} = 5 \text{ В}; R_1 = 100 \text{ Ом}; R_2 = 100 \text{ Ом};$$

$$V_{out} = \frac{100 \text{ Ом} * 5 \text{ В}}{100 \text{ Ом} + 100 \text{ Ом}} = \frac{1}{2} * 5 \text{ В} = 2,5 \text{ В}$$

# Делитель напряжения



$$V_{out} = \frac{R_2 V_{cc}}{R_1 + R_2}$$

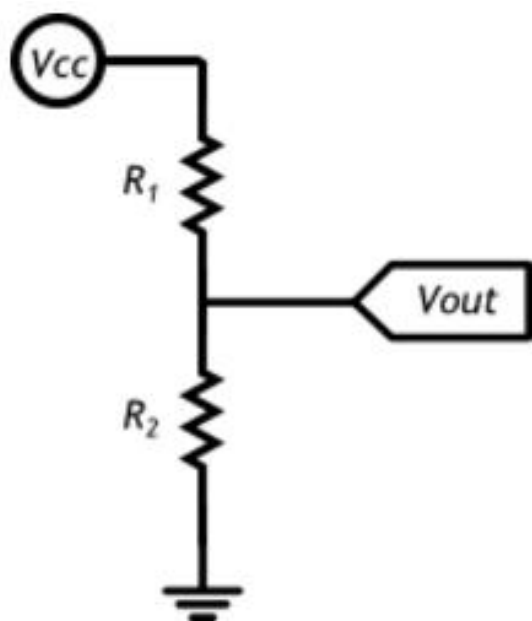
- $V_{out}$  — напряжение, которое считывает аналоговый порт

Пример 2:

$$V_{cc} = 5 \text{ В}; R_1 = 100 \text{ Ом}; R_2 = 200 \text{ Ом};$$

$$V_{out} = \frac{200 \text{ Ом} * 5 \text{ В}}{100 \text{ Ом} + 200 \text{ Ом}} = \frac{2}{3} * 5 \text{ В} \approx 3,3 \text{ В}$$

# Делитель напряжения



$$V_{out} = \frac{R_2 V_{cc}}{R_1 + R_2}$$

- $V_{out}$  — напряжение, которое считывает аналоговый порт

Пример 3:

$V_{cc} = 5 \text{ В}; R_1 = 100 \text{ Ом}; R_2 = 10 \text{ Ом};$

$$V_{out} = \frac{10 \text{ Ом} * 5 \text{ В}}{100 \text{ Ом} + 10 \text{ Ом}} = \frac{1}{11} * 5 \text{ В} \approx 0,45 \text{ В}$$

# Пример программы

level02

```
1 #define LED1 2
2 #define LED10 11
3
4 int led_on = 100;
5
6
7 void setup() {
8     // put your setup code here, to run once:
9     for(int i = LED1; i<= LED10; i++)
10         pinMode(i, OUTPUT);
11
12 }
```

# Пример программы

```
#define LED1 2
```

Создание константы с именем LED1 и значением 2.

```
int led_on = 100;
```

Создание глобальной переменной. Вместо int может быть bool, float, char.

```
void setup() {  
  // put your setup code here, to run once:  
}
```

Функция setup() — функция инициализации, выполняется только один раз

# Пример программы

```
for(int i = 0; i<= N; i = i + 1)
{
    }
}
```

Цикл for, где  $i$  — счетчик, 0 — начальное значение,  $i \leq N$  — условие выхода из цикла,  $i = i + 1$  - шаг цикла

```
pinMode(<номер порта>, <режим работы порта>);
```

Настраивает режим работы порта.

<номер порта> - значение типа int

<режим работы порта> - INPUT (порт генерирует сигнал), OUTPUT (порт принимает сигнал)

```
pinMode(5, OUTPUT);
```

```
pinMode(3, INPUT);
```

# Пример программы

```
14 void loop() {
15     // put your main code here, to run repeatedly:
16
17     for(int i = LED1; i<= LED10; i=i+1)
18     {
19         digitalWrite(i, HIGH);
20         delay(led_on);
21
22     }
23
24     for(int i = LED10; i>= LED1; i=i-1)
25     {
26         digitalWrite(i, LOW);
27         delay(led_on);
28     }
29
30 }
```



# Пример программы

```
void loop()  
{  
  // put your main code here, to run repeatedly:  
}
```

Функция `loop` — главная функция в программе, выполняется бесконечное количество раз

```
digitalWrite(<номер порта>, <напряжение на порте>);
```

Устанавливает напряжение на ЦИФРОВОМ порте.

<номер порта> - значение типа `int`

<напряжение на порте> - `HIGH` (устанавливает напряжение в 5 В),  
`LOW` (устанавливает напряжение в 0 В)

```
digitalWrite(5, HIGH);
```

```
digitalWrite(LOW, HIGH);
```

# Пример программы

```
delay(<время в миллисекундах>);
```

Останавливает работу функции loop() на указанное время

<время в миллисекундах> - значение типа int

```
delay(1000);
```

```
Serial.begin(9600);
```

Используется в функции setup(). Устанавливает скорость соединения между платой и компьютером. Используется при запуске монитора порта.

```
i = analogRead(<номер аналогового порта>)
```

Функция возвращает значение типа int, полученное на аналоговом порте.

<номер аналогового порта> - A0, A1, A2, A3, ... (номера аналоговых портов)

```
i = analogRead(A0)
```

# Пример программы

```
Serial.println(<значение>);
```

Выводит значение в монитор порта. Перед использованием необходимо запустить Serial.begin.

```
Serial.println(analogRead(A0));
```

```
i = map(<значение>, <начало диапазона 1>, <конец диапазона 1>, <начало диапазона 2>, <конец диапазона 2>);
```

Возвращает отображение <значения> из диапазона 1 в диапазон 2

```
i = map(sens, 0, 1023, 2, 11);
```

```
if (<условие>)
```

```
{  
    <операторы 1>
```

```
}  
else
```

```
{  
    <операторы 2>
```

```
}
```