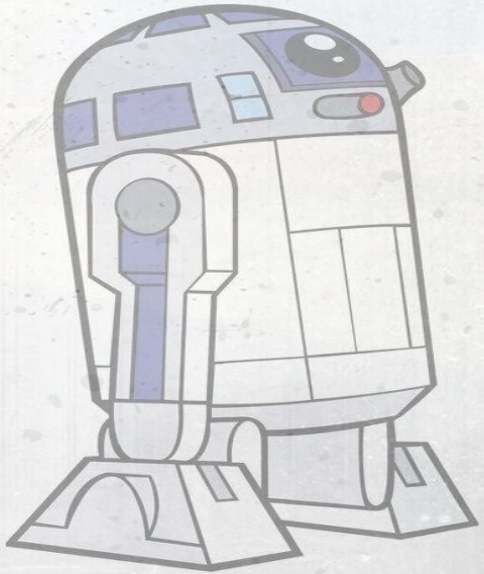
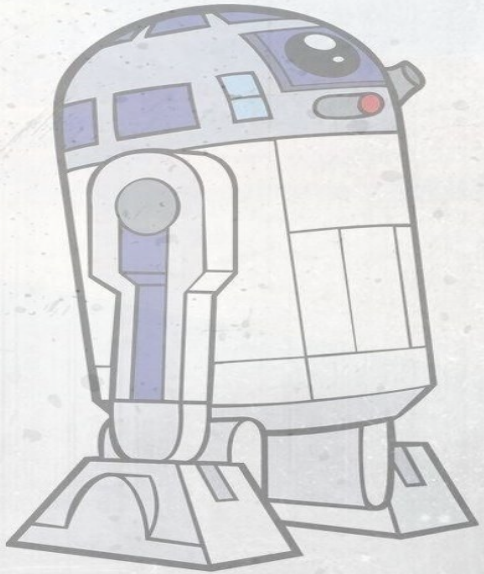
Подключение тактовой кнопки



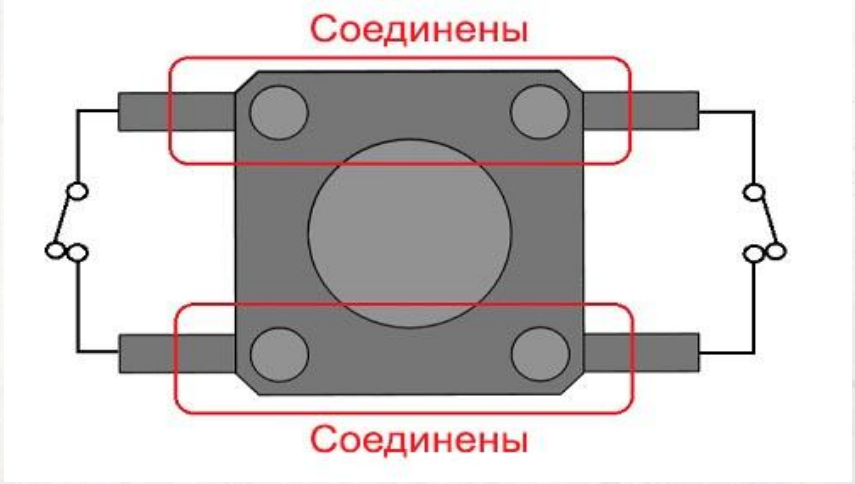
В этом примере мы рассмотрим подключение тактовой кнопки, с помощью которой мы будем осуществлять управление светодиодом

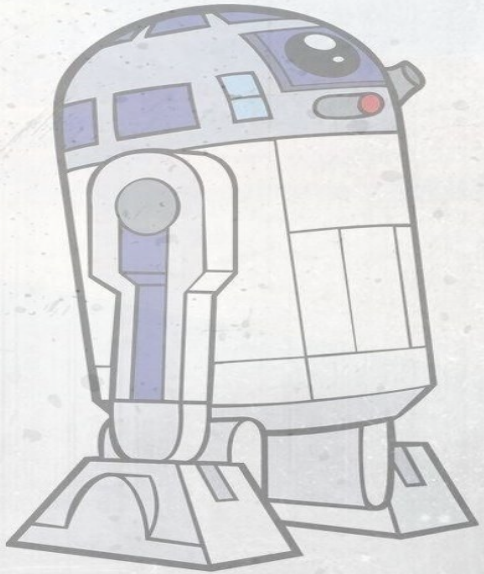
Необходимые компоненты

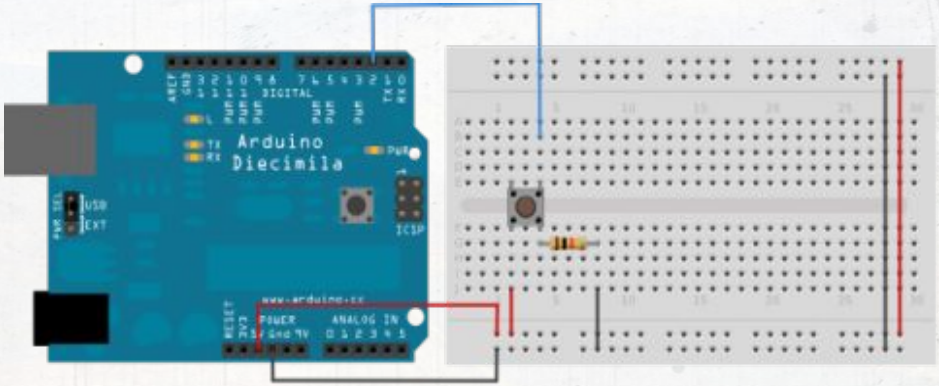
- контроллер Arduino
- тактовая кнопка
- 10кОм резистор
- контактная макетная плата
- соединительные провода

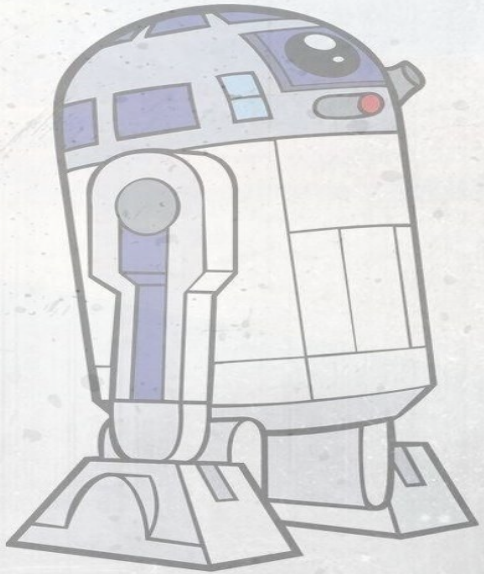


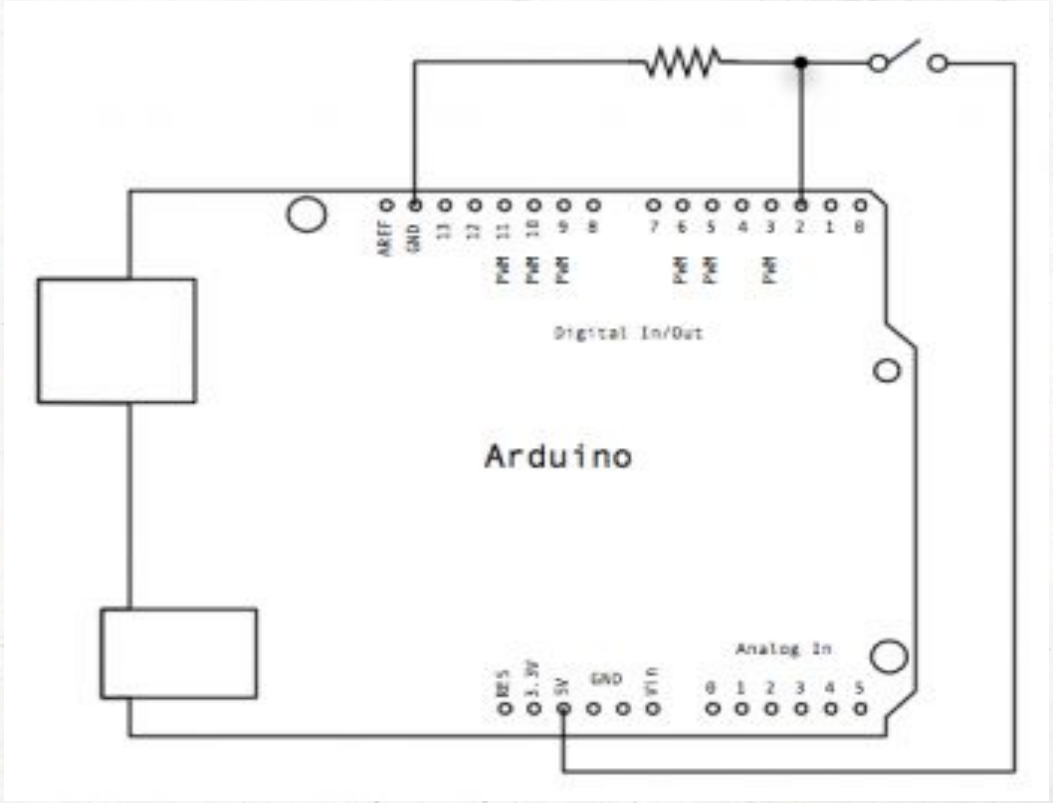
- Кнопка является простейшим устройством, при помощи которого можно управлять ходом программы на микроконтроллере, но физически она выполняет очень простую функцию: замыкает и размыкает контакт. Кнопки бывают нескольких типов:
- С фиксацией – кнопка остаётся нажатой после отпускания, без фиксации – отключается обратно.
- Нормально разомкнутая (*Normal Open, NO*) – при нажатии замыкает контакты. Нормально замкнутая (*Normal Closed, NC*) – при нажатии размыкает контакты.

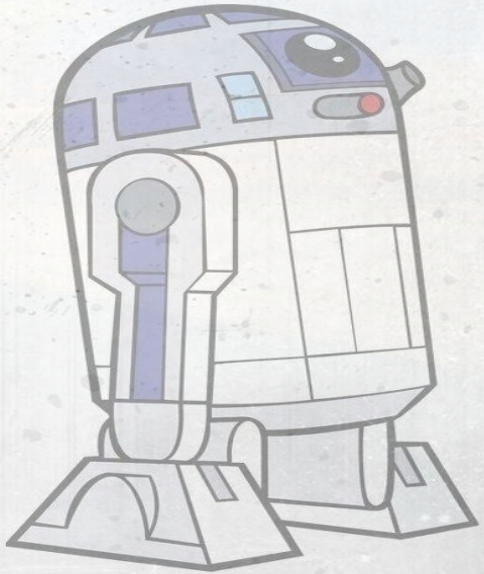




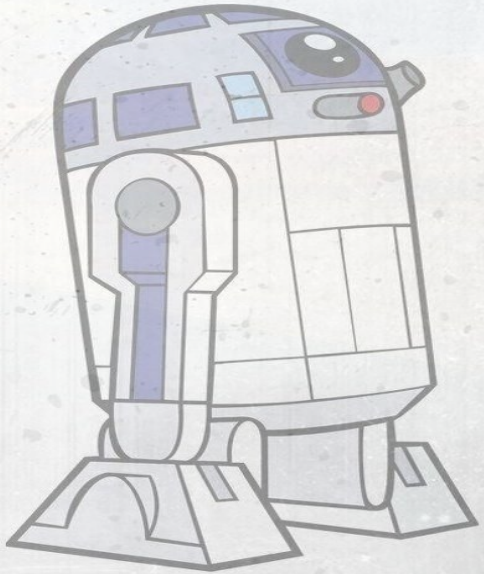

Arduino
Diecimila
DIGITAL
POWER
ANALOG IN
USB
EXT
ICSP




AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
PWM
PWM
PWM
PWM
PWM
PWM
Digital In/Out
Arduino
RES
3.3V
5V
GND
Vin
Analog In
0 1 2 3 4 5

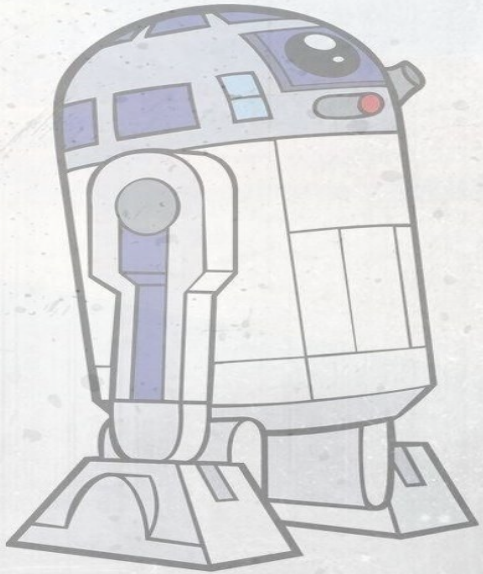


- Подключаем выход питания (5V) и землю (Gnd), красным и черным проводом соответственно к макетной плате. Обычно на макетных платах для питания и земли используют крайние ряды контактов. Третьим, синим, проводом мы соединяем цифровой пин 2 контроллера Arduino к контакту тактовой кнопки. К этому же контакту, либо к контакту, постоянно соединенному с ней в 4х штырковом исполнении, подключаем подтягивающий резистор 10 кОм, который в свою очередь соединяем с землей. Другой выход кнопки соединяем с питанием 5 В.



- Когда тактовая кнопка не нажата, выход 2 подключен только к земле через подтягивающий резистор и на этом входе будет считываться LOW. А когда кнопка нажата появляется контакт между входом 2 и питанием 5В, и считываться будет HIGH.

- Если вход оставить неподключенным, то на входе будет считываться HIGH или LOW случайным образом. Именно поэтому мы используем подтягивающий резистор, чтобы задать определенное значение при ненажатой кнопке.



```
// задаем переменные
const int buttonPin = 2;     // номер входа, подключенный к кнопке
const int ledPin =  13;      // номер выхода светодиода

// переменные
int buttonState = 0;         // переменная для хранения состояния кнопки

void setup() {
  // инициализируем пин, подключенный к светодиоду, как выход
  pinMode(ledPin, OUTPUT);
  // инициализируем пин, подключенный к кнопке, как вход
  pinMode(buttonPin, INPUT);
}

void loop(){
  // считываем значения с входа кнопки
  buttonState = digitalRead(buttonPin);

  // проверяем нажата ли кнопка
  // если нажата, то buttonState будет HIGH:
  if (buttonState == HIGH) {
    // включаем светодиод
    digitalWrite(ledPin, HIGH);
  }
  else {
    // выключаем светодиод
    digitalWrite(ledPin, LOW);
  }
}
```

pinMode(ledPin,OUTPUT):

Описание

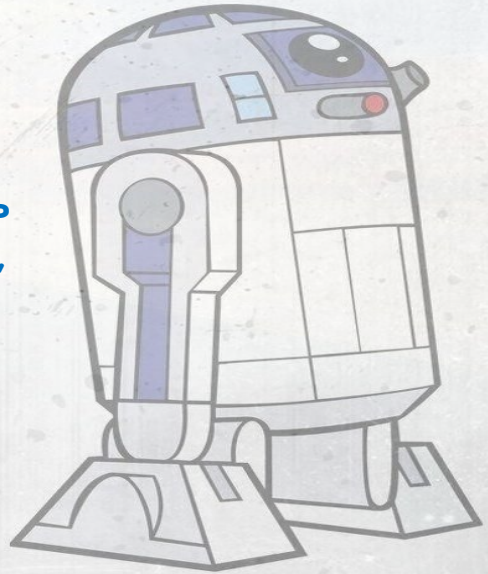
- Устанавливает режим работы заданного вход/выхода(pin) как входа или как выхода. Подробнее про цифровые вход/выходы(pins).

Синтаксис

- pinMode(pin, mode)

Параметры

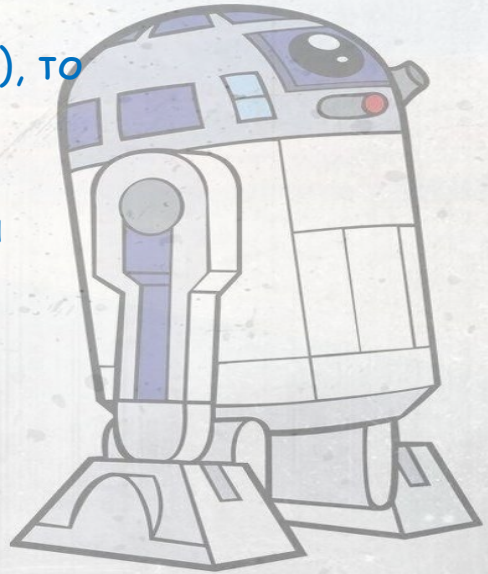
- pin: номер вход/выхода(pin), который Вы хотите установить
- mode: режим одно из двух значение - INPUT или OUTPUT, устанавливает на вход или выход соответственно.



digitalWrite (ledPin, HIGH);

Описание

- Подает HIGH или LOW значение на цифровой вход/выход (pin).
- Если вход/выход (pin) был установлен в режим выход (OUTPUT) функцией pinMode(), то для значение HIGH напряжение на соответствующем вход/выходе (pin) будет 5В (3.3В для 3.3V плат), и 0В(земля) для LOW.
- Если вход/выход (pin) был установлен в режим вход (INPUT), то функция digitalWrite со значением HIGH будет активировать внутренний 20K нагрузочный резистор. Подача LOW в свою очередь отключает этот резистор. Нагрузочного резистра достаточно чтобы светодиод, подключенный к входу, светил тускло. Если вдруг светодиод работает, но очень тускло, возможно необходимо установить режим выход (OUTPUT) функцией pinMode().



Синтаксис

- digitalWrite(pin, value)

Параметры

digitalRead (buttonPin);

Описание

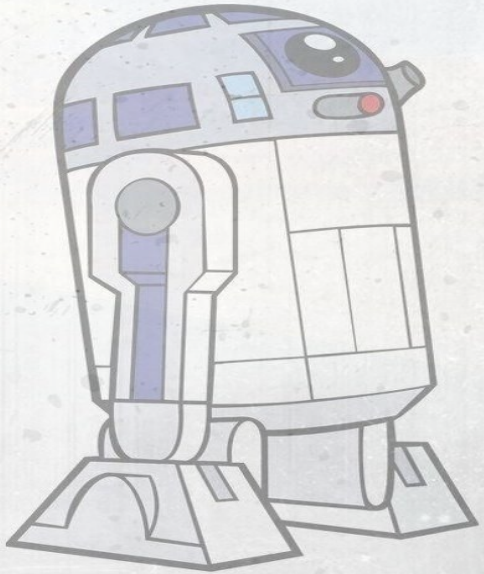
- Функция считывает значение с заданного входа - HIGH или LOW.

Синтаксис

- digitalRead(pin)

Параметры

- pin: номер вход/выхода(pin) который Вы хотите считать



Комментарии

- Комментарии – это строки в программе, которые используются для информирования вас самих или других о том, как работает программа. Они игнорируются компилятором и не экспортируются в процессор, таким образом, они не занимают место в памяти микроконтроллера Atmega.
- Комментарии предназначены только для того, чтобы помочь вам понять (или вспомнить), как работает ваша программа или объяснить это другим. Есть два способа пометить строку как комментарий:
- Пример
- x = 5; // Это комментарий в одной строке. Все после двойного слэша – комментарий
- // до конца строки
- /* это многострочный комментарий – используйте его для закоментирования целых кусков кода
- if (gwb == 0){ // комментарий в строке допустим внутри многострочного комментария
- // но не другой многострочный комментарий
- }
- // не забывайте «закрывать» комментарии – они должны быть парными!
- */
- Подсказка
- Во время экспериментов с кодом, «закомментирование» частей программы – подходящий способ удаления строк, в которых могут быть ошибки. Так строки в коде остаются, но превращаются в комментарии, и компилятор просто игнорирует их. Это может быть особенно полезно при локализации проблемы, или когда не получается скомпилировать программу, а сообщение об ошибке при компиляции скрыто или бесполезно.