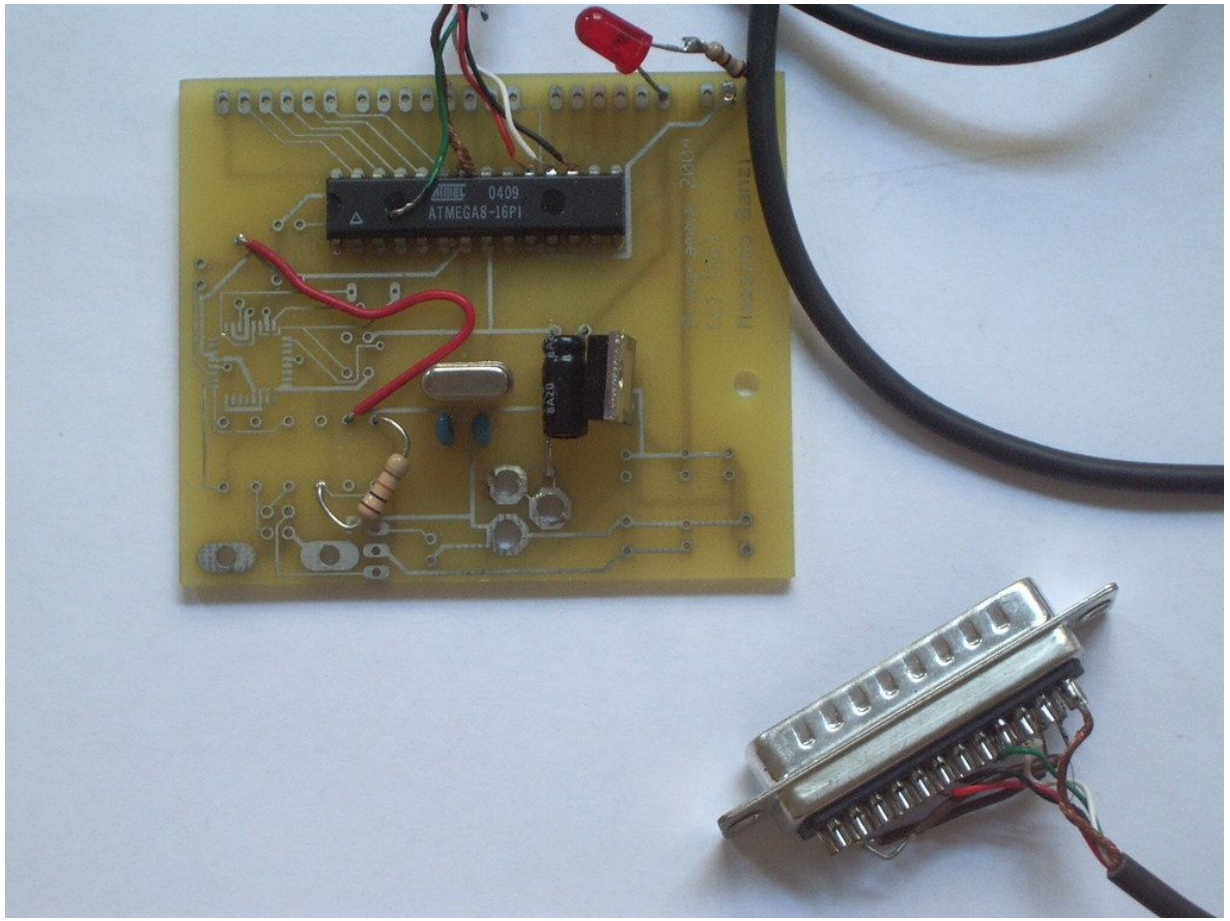
Arduino

Arduino – это инструмент для проектирования электронных устройств (электронный конструктор) более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Это платформа, предназначенная для «physical computing» с открытым программным кодом, построенная на простой печатной плате с современной средой для написания программного обеспечения.

Ядро команды Arduino

Ядро команды Arduino (слева направо): Дэвид Куар-тилльз (David Cuartielles), Джанлука Мартино(Gianluka Martino), Том Иго (Tom Igoe), Дэвид Мелис (David Mellis), и Массимо Банци (Massimo Banzi) на конференции Maker Faire в Нью-Йорке

Первая плата прототипа



Сделанная в 2005 году, имела простейший дизайн и еще не называлась Arduino.
Немного позже, в том же году, Массимо Банци придумал ей имя
(Фото: Массимо Банци)

Для чего?

Arduino применяется для создания электронных устройств с возможностью приема сигналов от различных цифровых и аналоговых датчиков, которые могут быть подключены к нему, и управления различными исполнительными устройствами. Проекты устройств, основанные на Arduino, могут работать самостоятельно или взаимодействовать с программным обеспечением на компьютере. Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Среда разработки программ с открытым исходным текстом доступна для *бесплатного* скачивания.

Язык программирования

- Язык программирования Arduino является реализацией Wiring, схожей платформы для «physical computing», основанной на мультимедийной среде программирования Processing.
- *Processing* представляет собой программное приложение, которое позволяет создавать, изменять, компилировать и запускать *Java*-код. Это *Java*-подобный язык программирования, созданный в *MIT Media Lab* с открытым исходным кодом и одновременно среда разработки.*Processing* позволяет очень быстро создавать визуальные интерактивные интерфейсы пользователей.
- PC – это подход к изучению общения в системе человек-компьютер-компьютер-человек, в основе которого лежит попытка понимания способов физического самовыражения людей.

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

• void loop()
• {
•   digitalWrite(13, 1);
•   delay(500);
•   digitalWrite(13, 0);
•   delay(500);
• }
```

->

```
#include "WProgram.h"
void setup();
void loop();
void setup() {
pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, 1);
  delay(500);
  digitalWrite(13, 0);
  delay(500);
}
int main(void)
{
        init();
        setup();
        for (;;)
                loop();

        return 0;
}
```

Достоинства

Arduino упрощает процесс работы с микроконтроллерами и имеет ряд преимуществ перед другими устройствами:

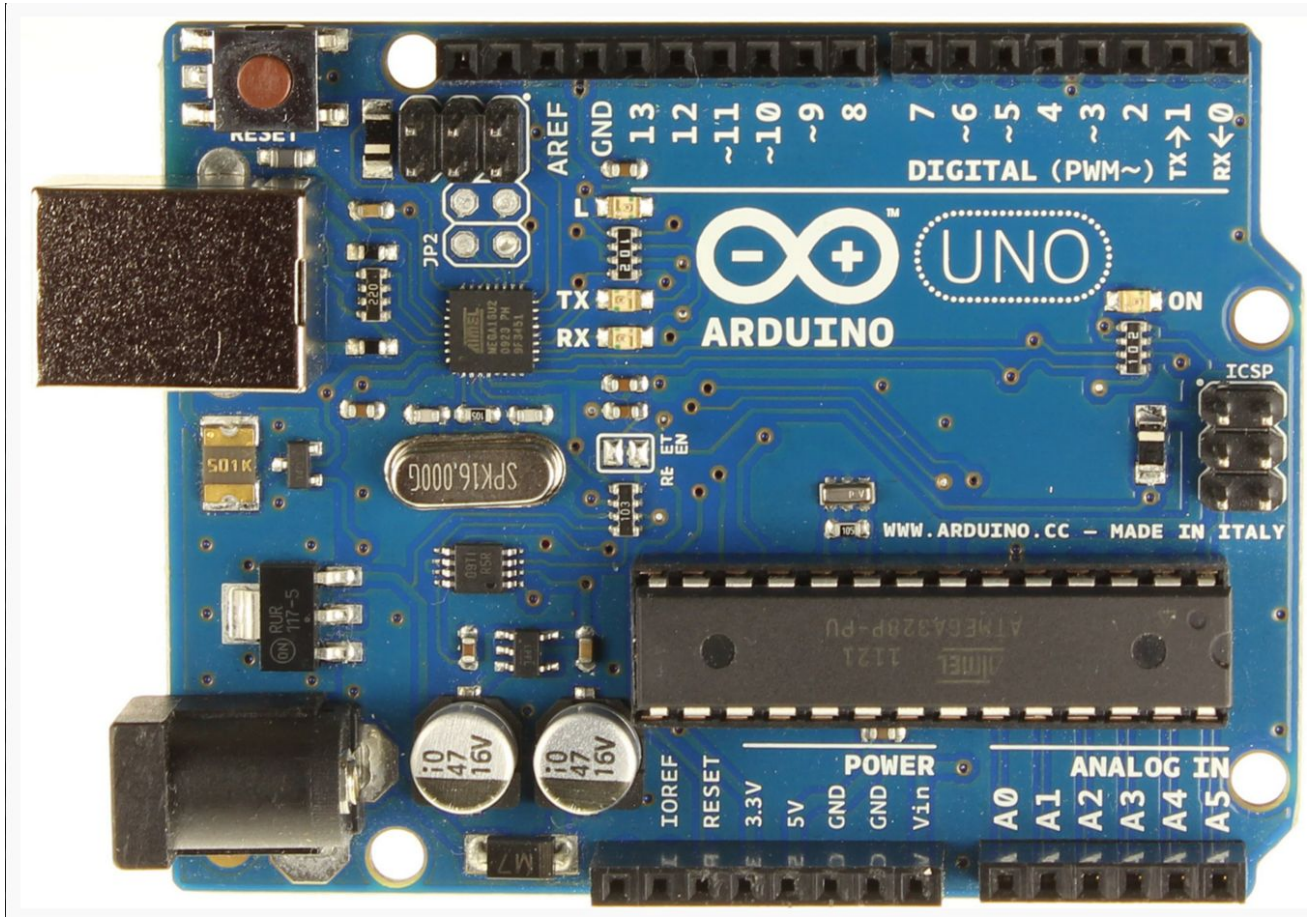
- Низкая стоимость – платы Arduino относительно дешевы по сравнению с другими платформами. Самая недорогая версия модуля Arduino может быть собрана в ручную, а некоторые даже готовые модули стоят меньше 10 долларов.

- Кросс-платформенность – программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows.

Достоинства

- Простая и понятная среда программирования – среда Arduino подходит как для начинающих пользователей, так и для опытных. Arduino основана на среде программирования Processing.
- Программное обеспечение с возможностью расширения и открытым исходным текстом – ПО Arduino выпускается как инструмент, который может быть дополнен опытными пользователями. Язык может дополняться библиотеками C++. Пользователи, желающие понять технические нюансы, имеют возможность перейти на язык AVR C на котором основан C++. Соответственно, имеется возможность добавить код из среды AVR-C в программу Arduino.
- Аппаратные средства с возможностью расширения и открытыми принципиальными схемами – микроконтроллеры ATMEGA8 и ATMEGA168 являются основой Arduino. Схемы модулей выпускаются с лицензией Creative Commons, а значит, опытные инженеры имеют возможность создания собственных версий модулей, расширяя и дополняя их. Даже обычные пользователи могут разработать опытные образцы с целью экономии средств и понимания работы.

Arduino UNO


RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
UNO
ARDUINO
ON
TX
RX
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Технические характеристики

- Arduino Uno контроллер построен на ATmega328. Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

- В отличие от всех предыдущих плат, использовавших FTDI USB микроконтроллер для связи по USB, новый Ардуино Uno использует микроконтроллер ATmega8U2.

- "Uno" переводится как один с итальянского и разработчики тем самым намекают на грядущий выход Arduino 1.0. Новая плата стала флагманом линейки плат Ардуино. Для сравнения с предыдущими версиями можно обратиться к полному списку плат Arduino.

Основные параметры

Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактовая частота	16 МГц

Питание

- Arduino Uno может получать питание через подключение USB или от внешнего источника питания. Источник питания выбирается автоматически.
- Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с центральным положительным полюсом. Провода от батареи подключаются к выводам Gnd и Vin разъема питания.
- Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Питание

Выводы питания:

- VIN. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод.
- 5V. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В.
- 3V3. Напряжение на выводе 3.3 В генерируемое встроенным регулятором на плате. Максимальное потребление тока 50 мА.
- GND. Выводы заземления.

Память

Микроконтроллер ATmega328 располагает 32 кБ флэш памяти, из которых 0.5 кБ используется для хранения загрузчика, а также 2 кБ ОЗУ (SRAM) и 1 Кб EEPROM.(которая читается и записывается с помощью библиотеки EEPROM).

Назначение контактов

- Каждый из 14 цифровых выводов Uno может настроен как вход или выход, используя функции pinMode(), digitalWrite(), и digitalRead(), . Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (по умолчанию отключен) 20-50 кОм и может пропускать до 40 мА. Некоторые выводы имеют особые функции:
- Последовательная шина: 0 (RX) и 1 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины ATmega8U2 USB-to-TTL.
- Внешнее прерывание: 2 и 3. Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции attachInterrupt().

Назначение контактов

- ШИМ: 3, 5, 6, 9, 10, и 11. Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции analogWrite().
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Посредством данных выводов осуществляется связь SPI, для чего используется библиотека SPI.
- LED: 13. Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.
- На платформе Uno установлены 6 аналоговых входов (обозначенных как A0 .. A5), каждый разрешением 10 бит (т.е. может принимать 1024 различных значения). Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее имеется возможность изменить верхний предел посредством вывода AREF и функции analogReference(). Некоторые выводы имеют дополнительные функции:
- I2C: 4 (SDA) и 5 (SCL). Посредством выводов осуществляется связь I2C (TWI), для создания которой используется библиотека Wire.

Назначение контактов

Дополнительная пара выводов платформы:

- **AREF**. Опорное напряжение для аналоговых входов. Используется с функцией analogReference().
- **Reset**. Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на плате расширения, закрывающей доступ к кнопке на самой плате Arduino.

Связь

- На платформе Arduino Uno установлено несколько устройств для осуществления связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega328 поддерживают последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема ATmega8U2 направляет данный интерфейс через USB, программы на стороне компьютера "общаются" с платой через виртуальный COM порт. Прошивка ATmega8U2 использует стандартные драйвера USB COM, никаких сторонних драйверов не требуется, но на Windows для подключения потребуется файл ArduinoUNO.inf. Мониторинг последовательной шины (Serial Monitor) программы Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Светодиоды RX и TX на платформе будут мигать при передаче данных через микросхему FTDI или USB подключение (но не при использовании последовательной передачи через выводы 0 и 1).
- Библиотекой SoftwareSerial возможно создать последовательную передачу данных через любой из цифровых выводов Uno.
- ATmega328 поддерживает интерфейсы I2C (TWI) и SPI. В Arduino включена библиотека Wire для удобства использования шины I2C.

Связь

- Платформа программируется посредством ПО Arduino. Из меню Tools > Board выбирается «Arduino Uno» (согласно установленному микроконтроллеру). Подробная информация находится в справочнике и инструкциях.

- Микроконтроллер ATmega328 поставляется с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500.

- Имеется возможность не использовать загрузчик и запрограммировать микроконтроллер через выводы ICSP (внутрисхемное программирование).

Особенности языка программирования

Структура

Каждая программа Arduino (часто называемая «скетч») имеет две обязательные функции (также называемые подпрограммами).

```
void setup(){  }
```

Все команды, заключенные между фигурными скобками, выполняются только один раз, при первом запуске программы.

```
void loop(){  }
```

Эта подпрограмма выполняется циклически вплоть до отключения питания, после завершения подпрограммы setup().

Особенности языка программирования

Синтаксис

Требования к форматированию в языке С вызывают некоторые затруднения у начинающих (с другой стороны, благодаря своей структуре, язык С обладает большими возможностями). Если Вы запомните следующие правила, этого будет вполне достаточно.

// (однострочный комментарий)
Часто используется для размещения в тексте программы комментариев. Можно пояснять, что значит каждая строка программы. Все что размещается после двойной черты и до конца строки будет игнорироваться компилятором.

{ } (фигурные скобки)
Используются для определения начала и конца блока команд (используются в функциях и циклах).

/* */ (многострочный комментарий). Вы можете использовать эту структуру, если Вам надо создать подробный комментарий на нескольких строках. Все находящееся между этими символами будет игнорироваться компилятором.

; (точка с запятой)
Каждая команда должна заканчиваться этим символом (потерянная точка с запятой — наиболее распространенная ошибка, приводящая к невозможности компиляции).

Особенности языка программирования

Переменные

Любая программа всего лишь определенным образом манипулирует числами. Переменные помогают жонглировать цифрами.

int (целочисленная)
Основная рабочая лошадка, хранится в памяти с использованием двух байт (16 бит). Может содержать целое число в диапазоне -32 768 … 32 767.

long (длинная)
Используется в том случае, когда не хватает емкости int. Занимает в памяти 4 байта (32 бита) и имеет диапазон -2 147 483 648 … 2 147 483 647.

boolean (двоичная)
Простой тип переменной типа True/False. Занимает только один бит в памяти.

float (с плавающей запятой)
Используется для вычислений с плавающей запятой. Занимает в памяти 4 байта (32 бита) и имеет диапазон -3.4028235E+38.

char (символ)Хранит один символ, используюя кодировку ASCII (например «A» =65). Использует один байт памяти (8 бит). Arduino оперирует со строками как с массивами символов.

Особенности языка программирования

Математические операторы

Операторы используются для преобразования чисел.

=	(присвоение) делает что-то равным чему-то(например x=10*2 записывает в переменную x число 20).
%	остаток от деления). Например 12%10 дает результат 2.
+	(сложение)
-	(вычитание)
*	(умножение)
/	(деление)

Особенности языка программирования

Операторы сравнения

Операторы, используемые для логического сравнения.

== (равно) (Например 12==10 не верно (FALSE), 5==5 верно(TRUE).)

!= (не равно) (Например 12!=10 верно (TRUE), 5!=5 не верно (FALSE).)

< (меньше) (Например 12<10 не верно (FALSE), 12<12 не верно (FALSE), 12<14 верно (TRUE).)

> (больше) (Например 12>10 верно (TRUE), 12>12 не верно (FALSE), 12>14 не верно (FALSE).)

Особенности языка программирования

Управляющие структуры

Для определения порядка выполнения команд (блоков команд) служат управляющие структуры. Здесь приведены только основные структуры. Более подробно можете ознакомиться на сайте Arduino.

```
if (условие 1) {}
else if (условие 2) {}
else {}
```

Если условие 1 верно (TRUE) выполняются команды в первых фигурных скобках. Если условие 1 не верно (FALSE) то проверяется условие 2. Если условие 2 верно, то выполняются команды во вторых фигурных скобках, в противном случае выполняются команды в третьих фигурных скобках.

```
for (int i=0;
i<число повторов;
i++) {}
```

Эта структура используется для определения цикла. Цикл повторяется заданное число раз. Переменная i может увеличиваться или уменьшаться.

Особенности языка программирования

Цифровые сигналы

`digitalWrite(pin, value);`
Если порт установлен в режим OUTPUT, в него можно записать HIGH (логическую единицу, +5В) или LOW (логический ноль, GND).

`pinMode(pin, mode);`
Используется, чтобы определить режим работы соответствующего порта. Вы можете использовать адреса портов 0...19 (номера с 14 по 19 используются для описания аналоговых портов 0...5). Режим может быть или INPUT (вход) или OUTPUT (выход).

`digitalRead(pin);`
Если порт установлен в режим INPUT эта команда возвращает значение сигнала на входе HIGH или LOW.

Особенности языка программирования

Аналоговые сигналы

Arduino - цифровое устройство, но может работать и с аналоговыми сигналами при помощи следующих двух команд:

`analogWrite(pin,value);`

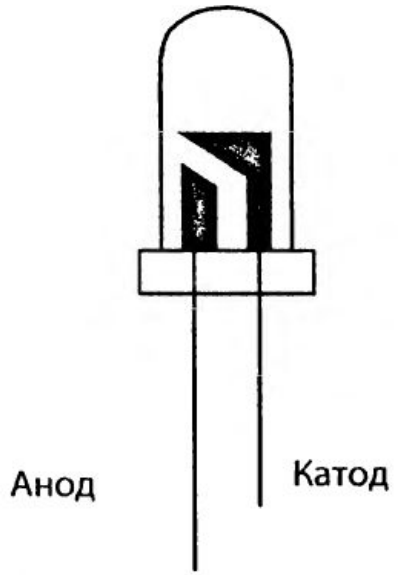
Некоторые порты Arduino (3,5,6,9,10,11) поддерживают режим ШИМ (широтно-импульсной модуляции). В этом режиме в порт посылаются логические единицы и нули с очень большой скоростью. Таким образом среднее напряжение зависит от баланса между количеством единиц и нулей и может изменяться в пределах от 0 (0В) до 255 (+5В).

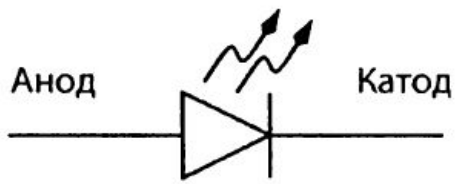
`analogRead(pin);`

Если аналоговый порт настроен в режим INPUT, то можно измерить напряжение на нем. Может принимать значения от 0 (0В) до 1024 (+5В).

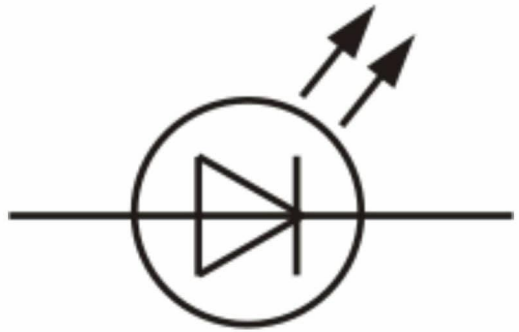
Компонентная база. Светодиоды

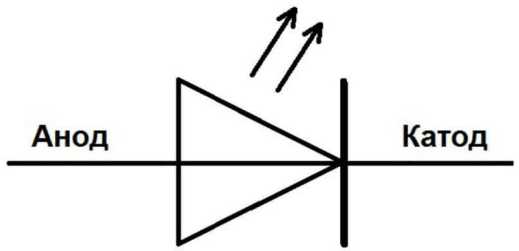
Светодиоды. Обозначение.



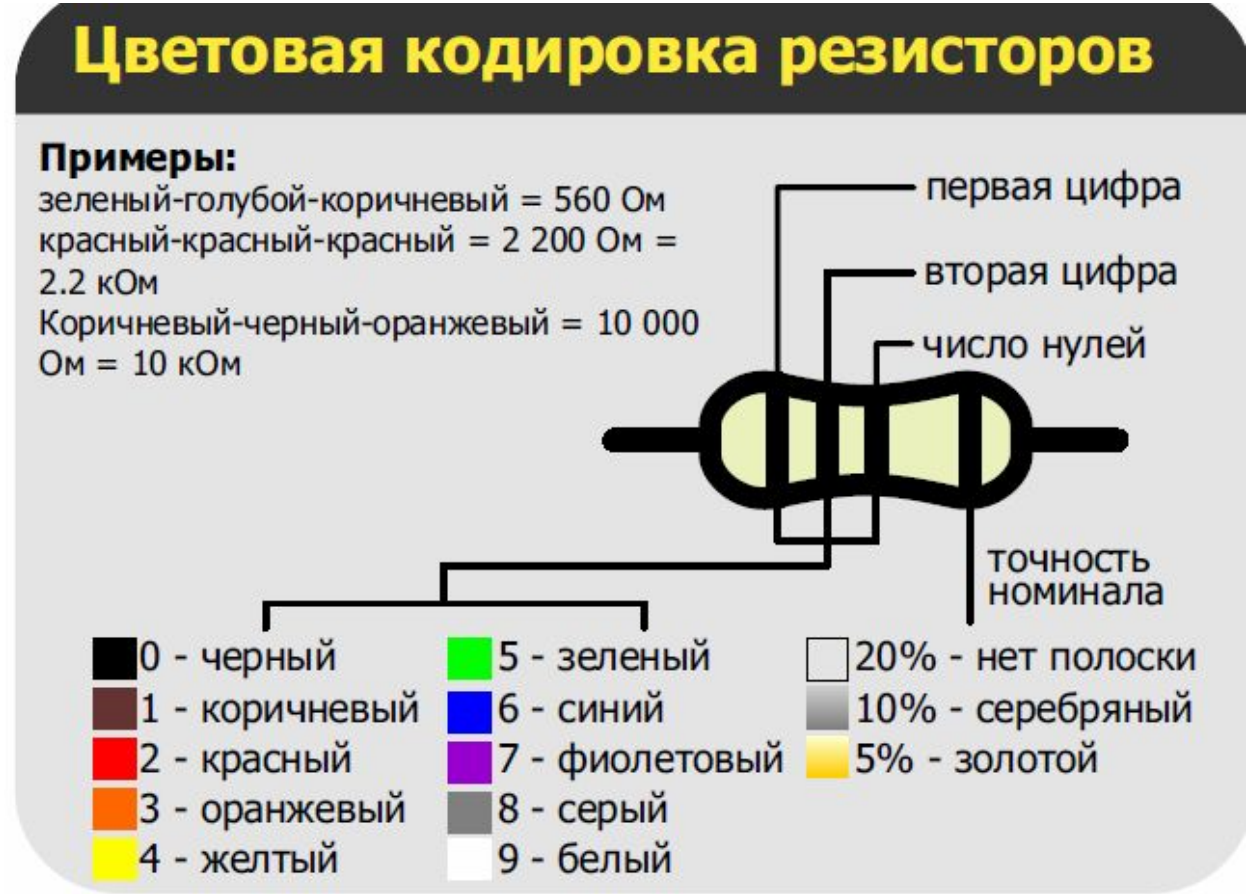


Светодиоды. Обозначение.





Резисторы


Цветовая кодировка резисторов
Примеры:
зеленый-голубой-коричневый = 560 Ом
красный-красный-красный = 2 200 Ом = 2.2 кОм
Коричневый-черный-оранжевый = 10 000 Ом = 10 кОм
первая цифра
вторая цифра
число нулей
точность номинала
0 - черный
1 - коричневый
2 - красный
3 - оранжевый
4 - желтый
5 - зеленый
6 - синий
7 - фиолетовый
8 - серый
9 - белый
20% - нет полоски
10% - серебряный
5% - золотой

Резисторы

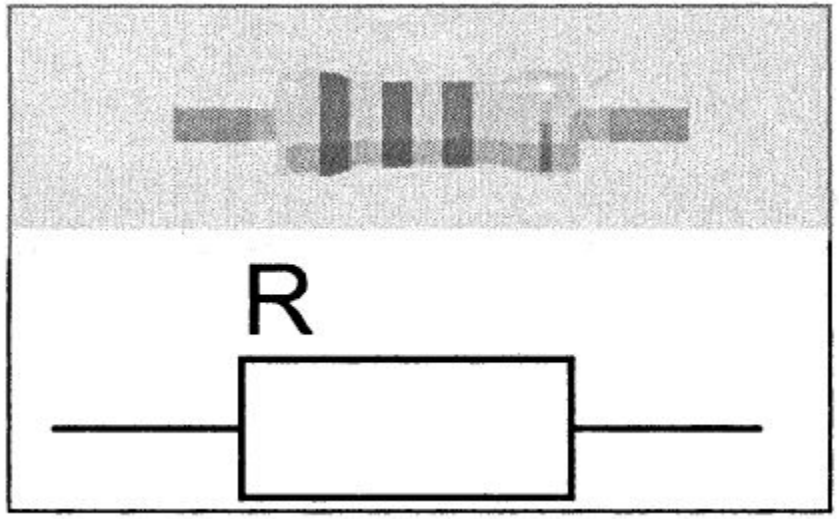


Таблица номиналов резисторов

Цвет	Кольцо 1	Кольцо 2	Кольцо 3 (множитель)	Кольцо 4 (допуск)
Серебристый	–	–	1×10^{-2} = 0,01 Ом	±10%
Золотистый	–	–	1×10^{-1} = 0,1 Ом	±5%
Черный	0	0	1×10^{0} = 1 Ом	–
Коричневый	1	1	1×10^{1} = 10 Ом	±1%
Красный	2	2	1×10^{2} = 100 Ом	±2%
Оранжевый	3	3	1×10^{3} = 1 кОм	–
Желтый	4	4	1×10^{4} = 10 кОм	–
Зеленый	5	5	1×10^{5} = 100 Ом	±0,5%
Голубой	6	6	1×10^{6} = 1 МОм	±0,25%
Фиолетовый	7	7	1×10^{7} = 10 МОм	±0,1%
Серый	8	8	1×10^{8} = 100 МОм	–
Белый	9	9	1×10^{9} = 1000 МОм	–

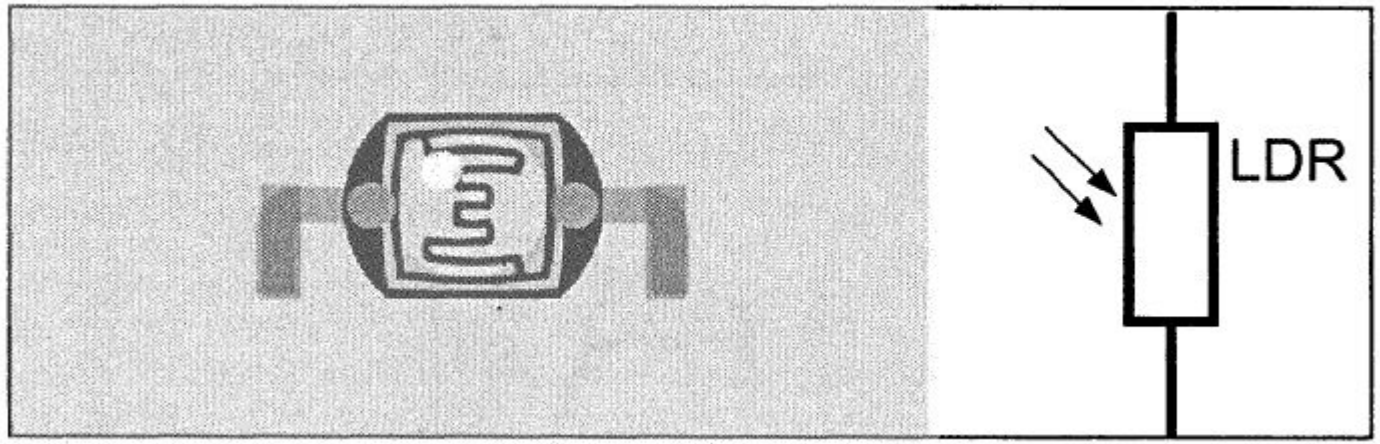
Ряд E24

Значения сопротивления резисторов с допуском ±5% принадлежат ряду E24, причем каждая декада содержит 24 равноотстоящих значения.

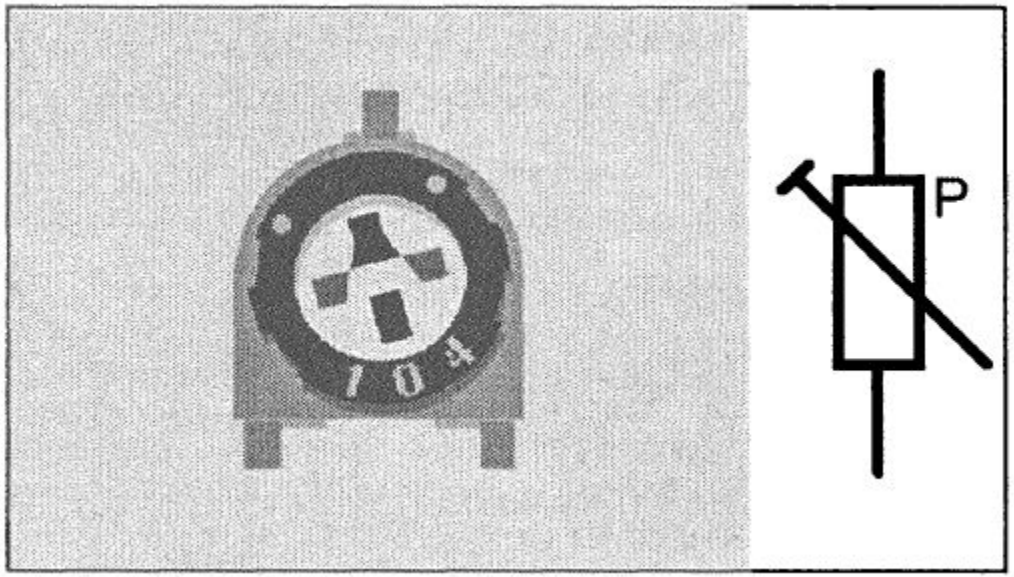
Вот значения нормального ряда E24:

1,0 / 1,1 / 1,2 / 1,3 / 1,5 / 1,6 / 1,8 / 2,0 / 2,2 / 2,4 / 2,7 / 3,0 / 3,3 / 3,6 / 3,9 / 4,3 / 4,7 / 5,1 / 5,6 / 6,2 / 6,8 / 7,5 / 8,2 / 9,1 .

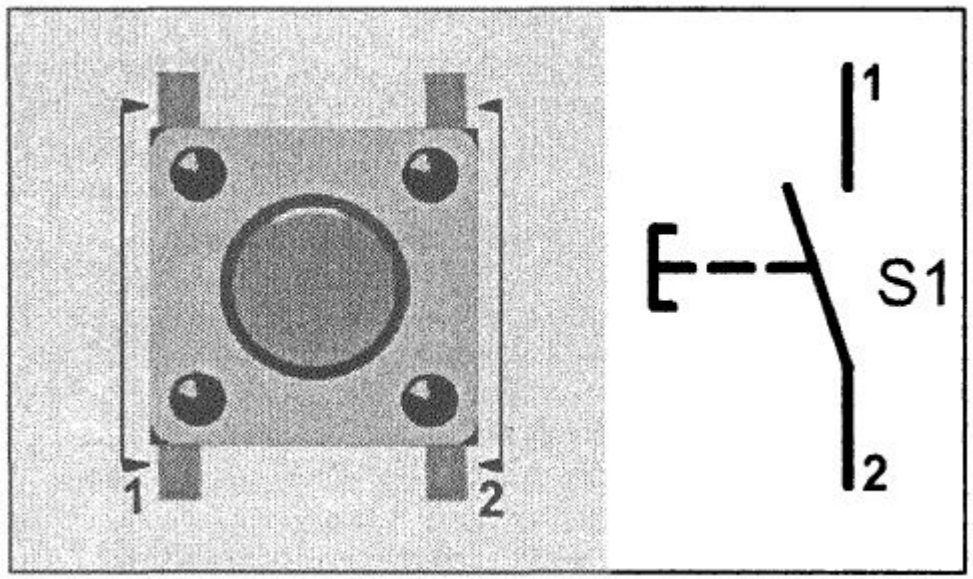
Фоторезистор



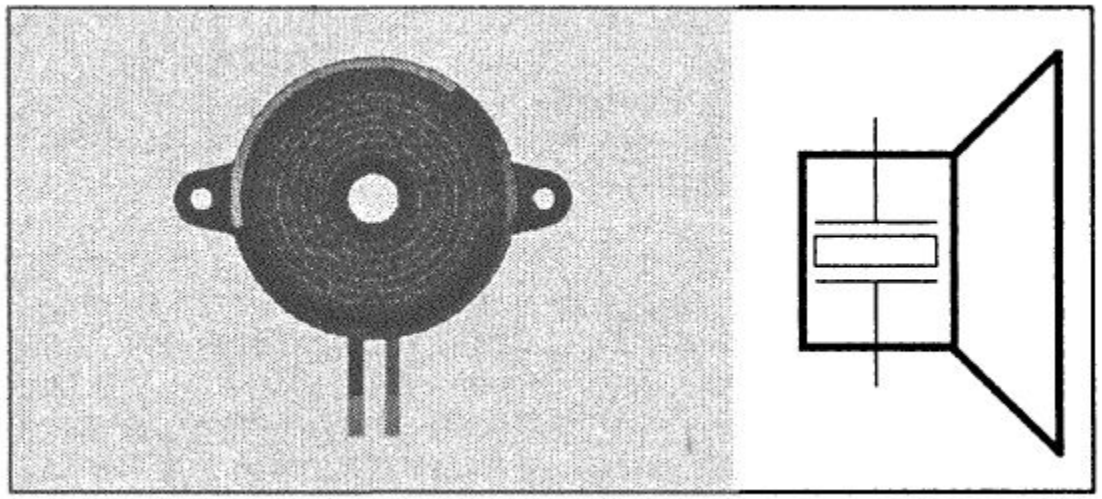
Потенциометр



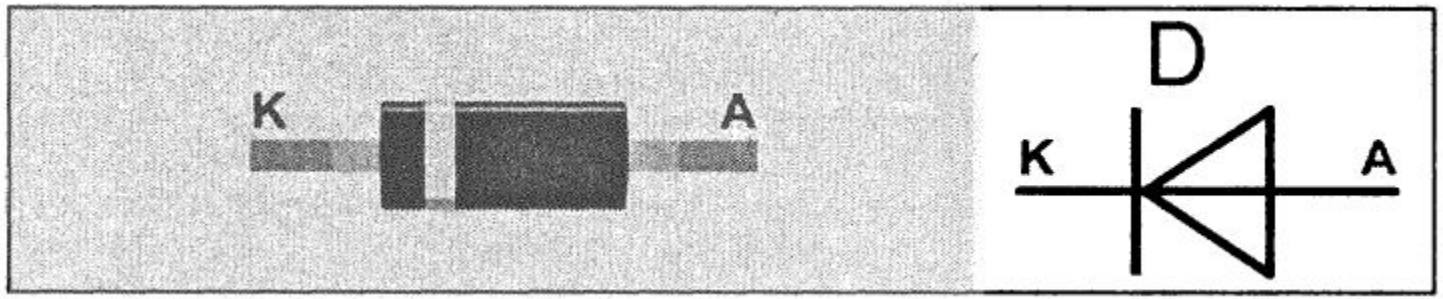
Кнопка



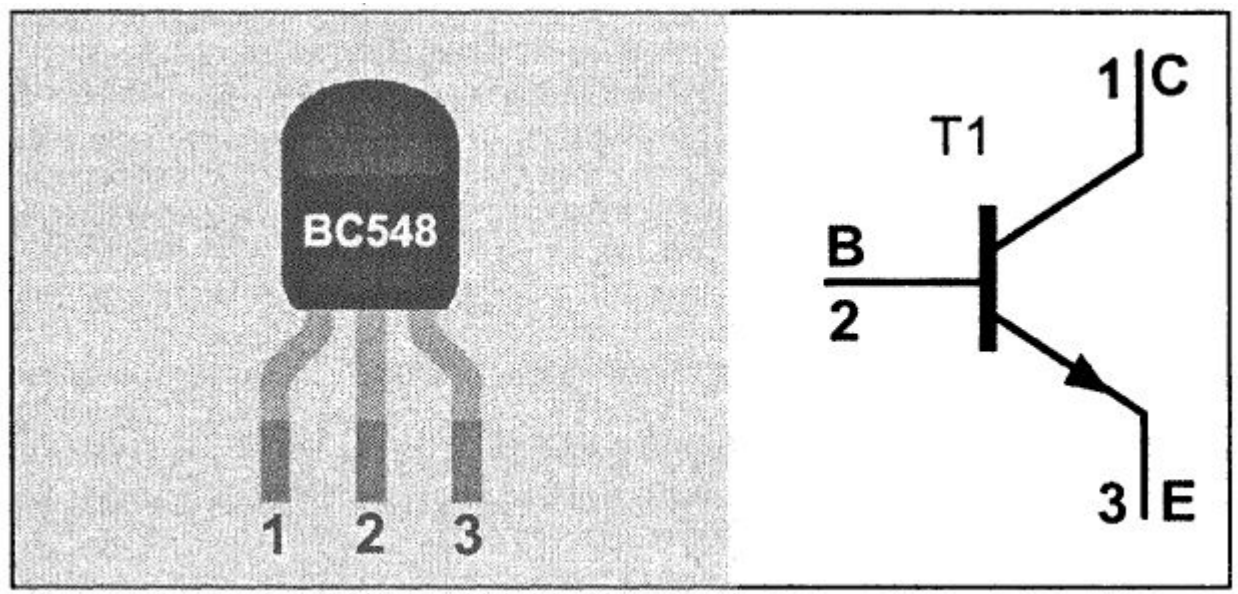
Акустический пьезопреобразователь



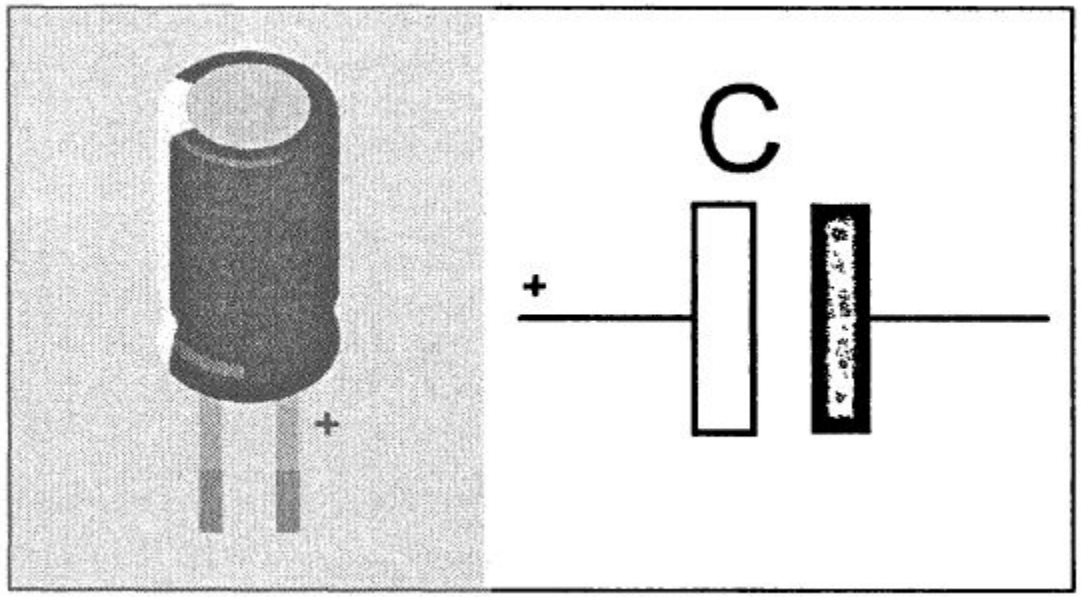
Диод



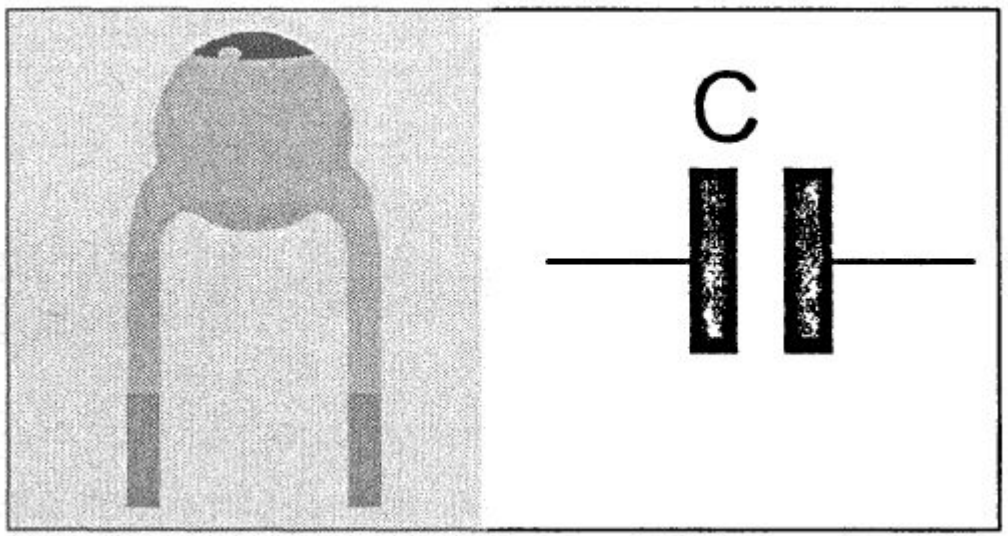
Транзистор



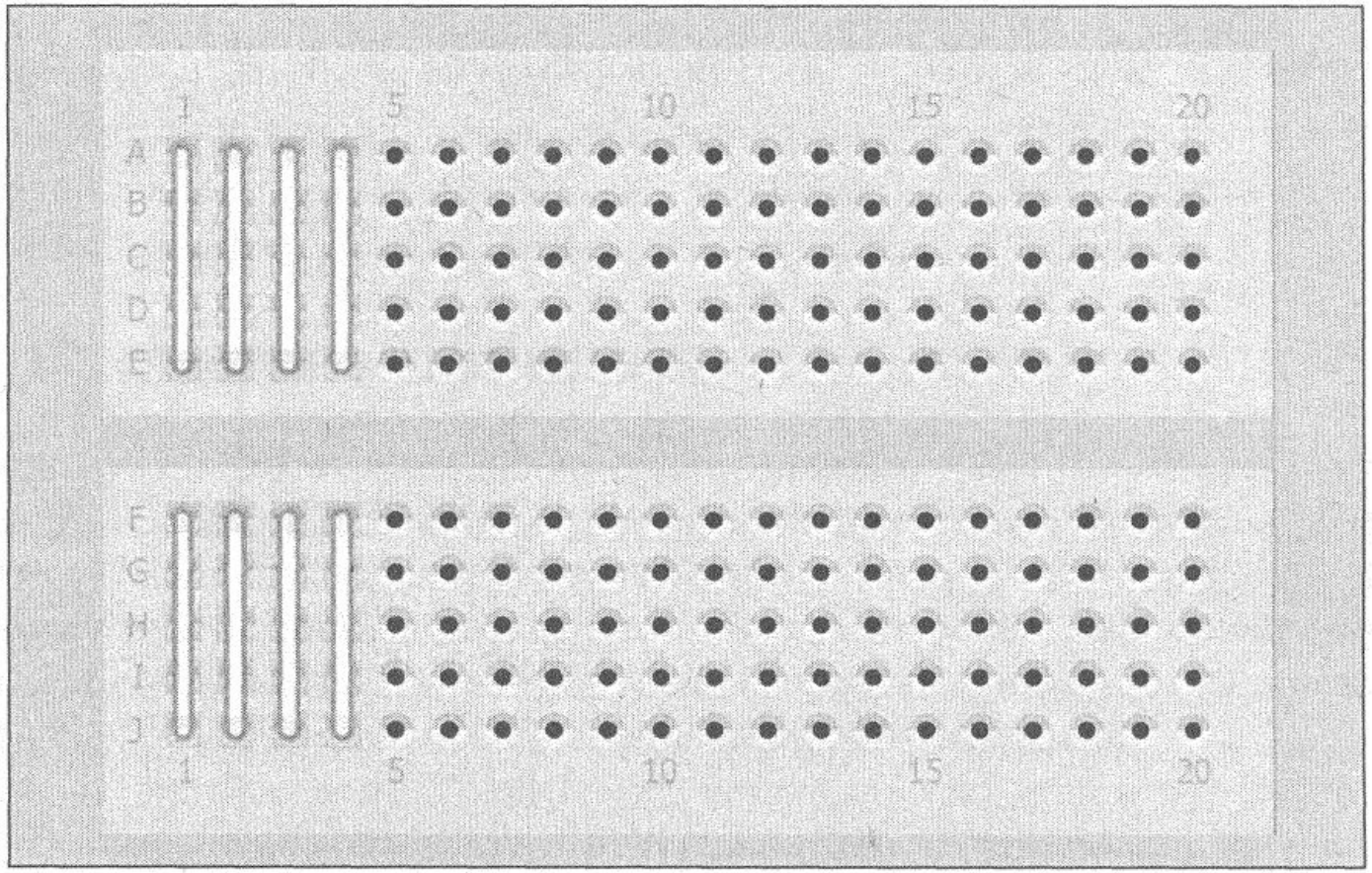
Электролитический конденсатор



Керамический конденсатор



Макетные платы



Пример