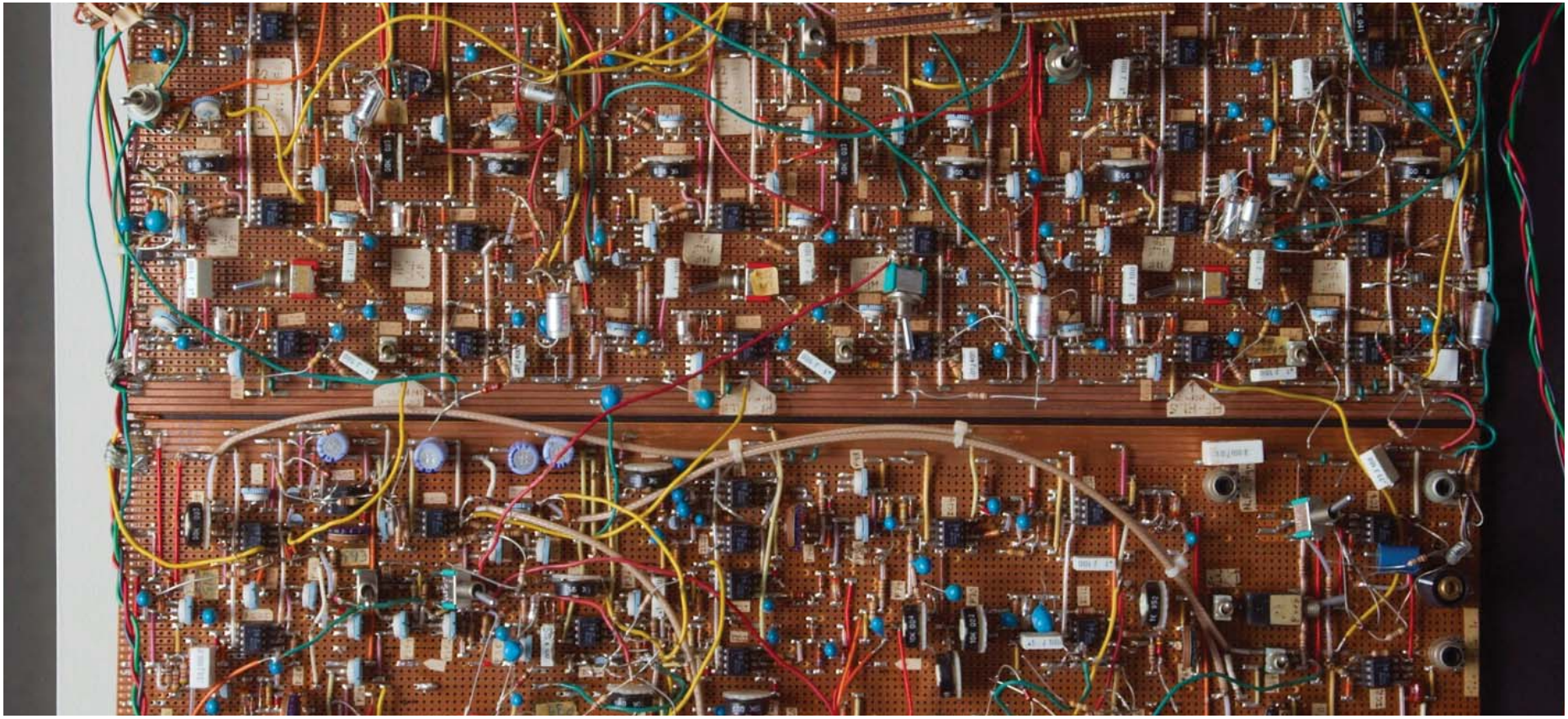


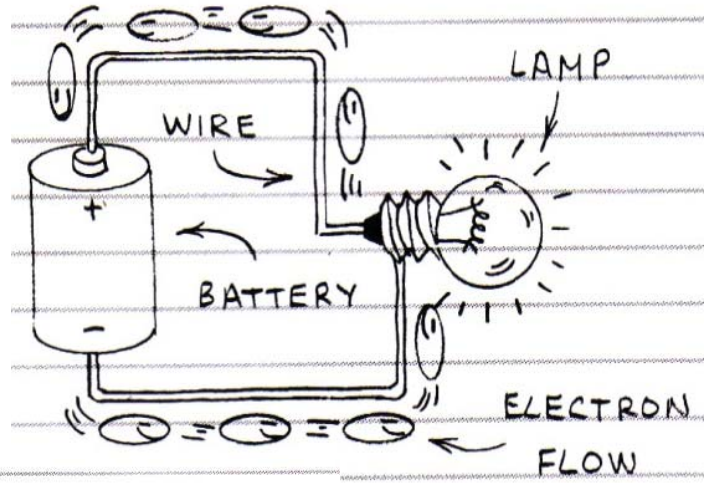
INTRODUCTION TO ELECTRONICS

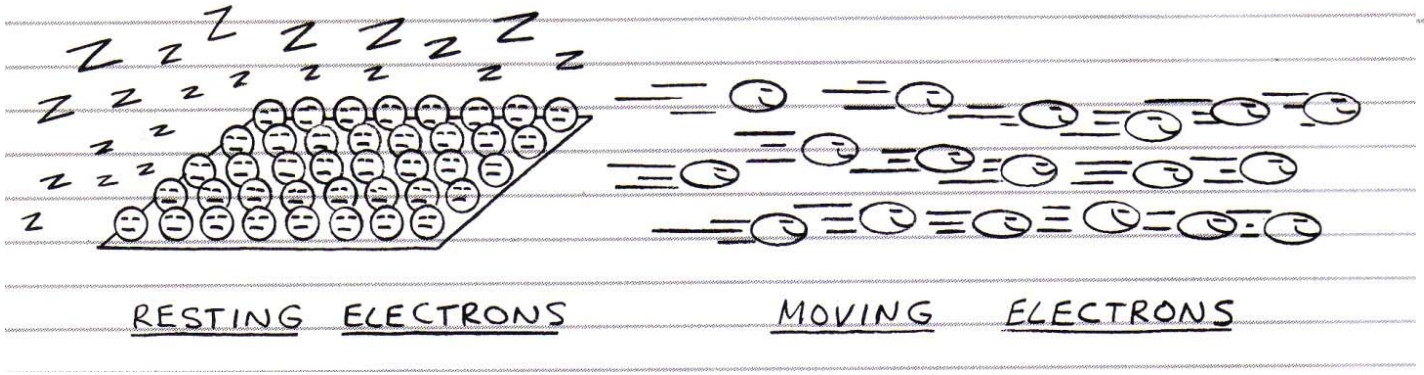
* Electricity, electronics, transduction, circuits, Ohm's Law
* Definition of components
* Using a multimeter
* Reading schematics
* Soldering

ELECTRICITY

_ Form of energy resulting from the existence of charged particles (such as electrons or protons), either statically as an accumulation of charge or dynamically as a current.

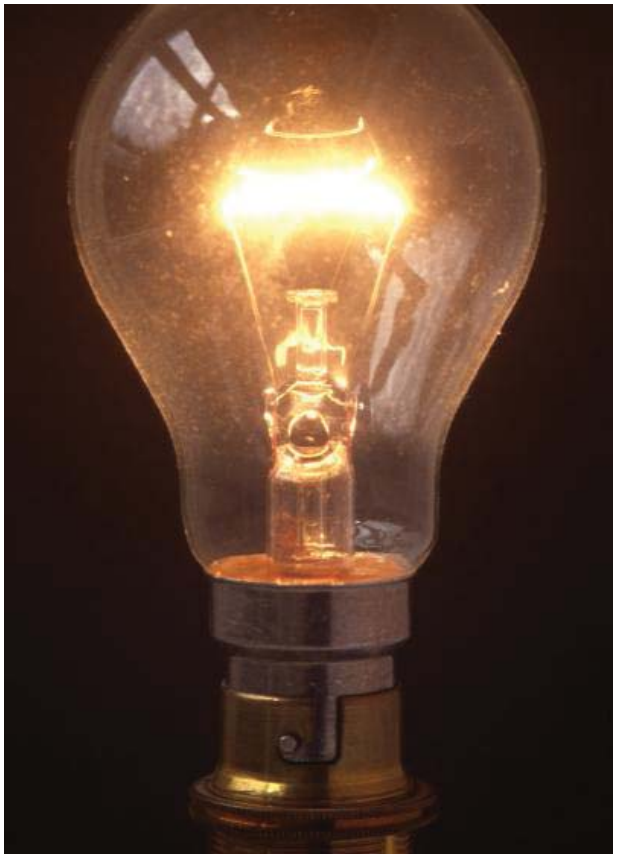
_Is the presence or flow of an electric charge





TRANSDUCTION

_ Conversion of one energy type to another energy type



Electricity ---------> Light

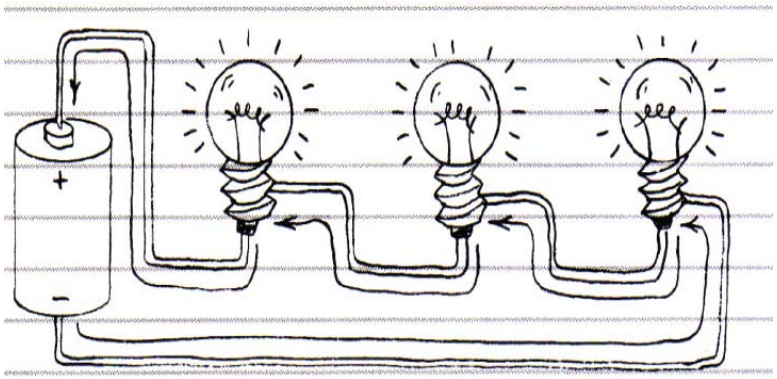
Electricity ---------> Heat

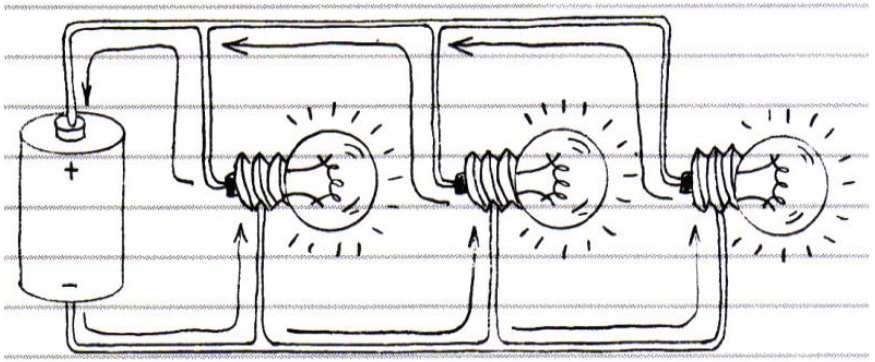
Electricity --> Rotational

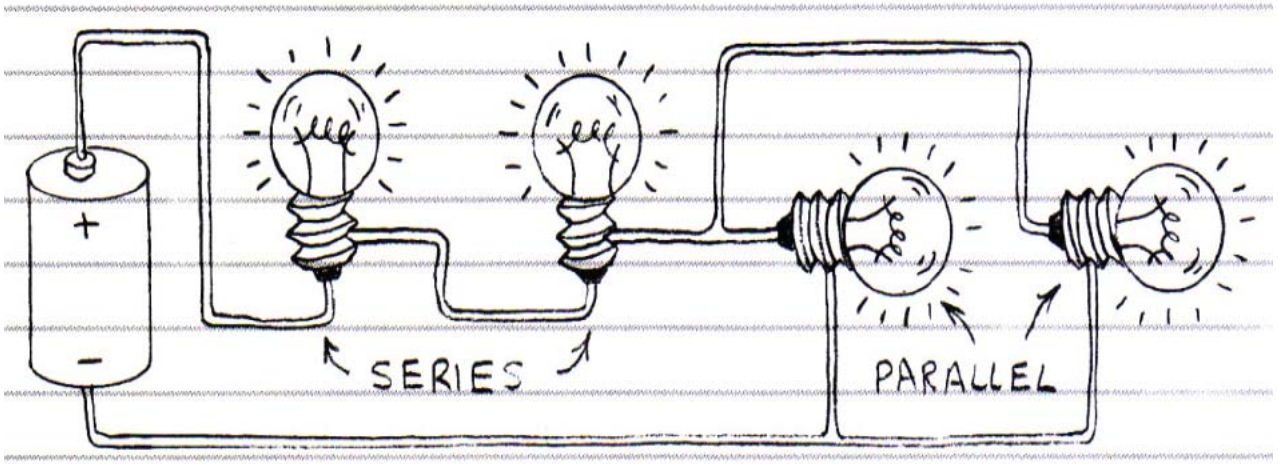
CIRCUITS

_ Devices that work by controlling the flow of electrons through electrical components

_A closed path formed by interaction of electrical components through which the electric current can flow.

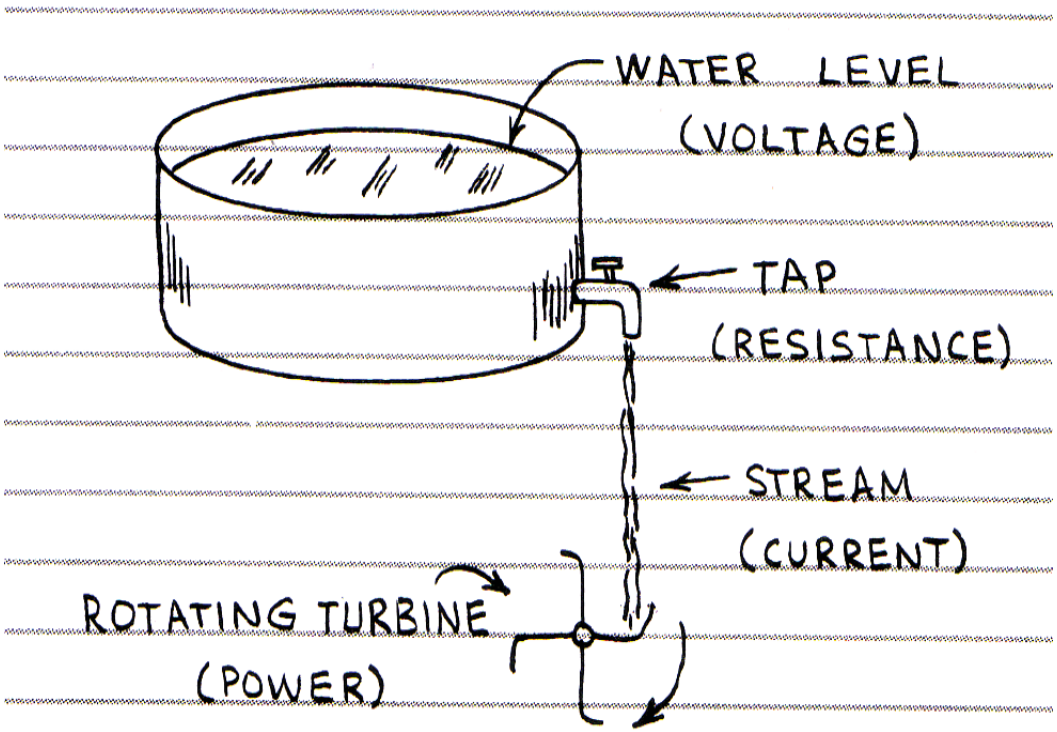






KEY TERMS

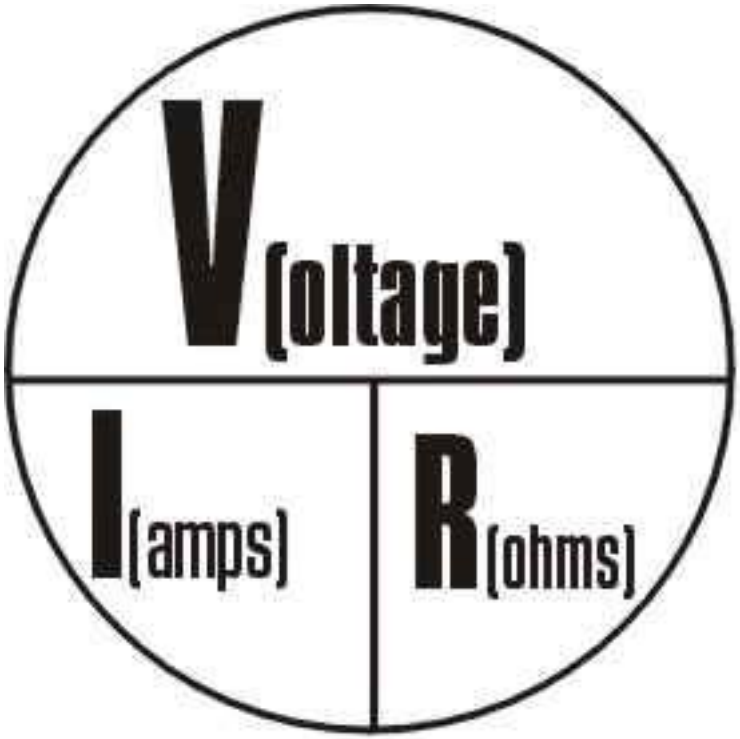
Voltage(V) unit: Volts(V) - potential difference in the circuit
Resistance(R) unit: Ohm(Ω)- a measure of the degree to which an object opposes an electric current through it
Current (I) unit: Ampere(A) - Quantity of electrons passing a given point
Power (P) unit:Watt (W) - Work done by electrical current



OHM'S LAW

_ For a given resistance, voltage across the two points is directly proportional to the current between them.

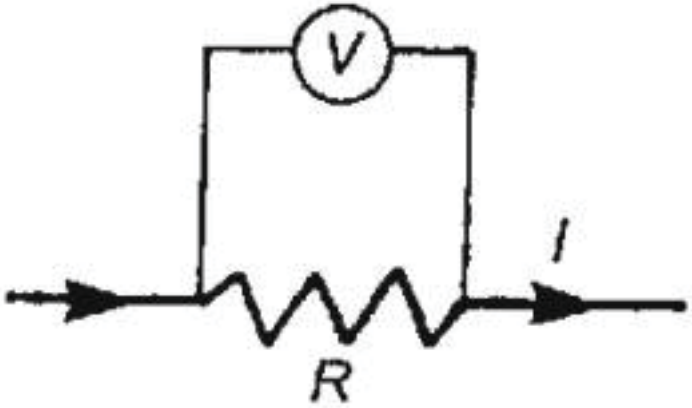
http://falstad.com/circuit/e-ohms.html



$$I \times R = V$$

$$V / R = I$$

$$V / I = R$$

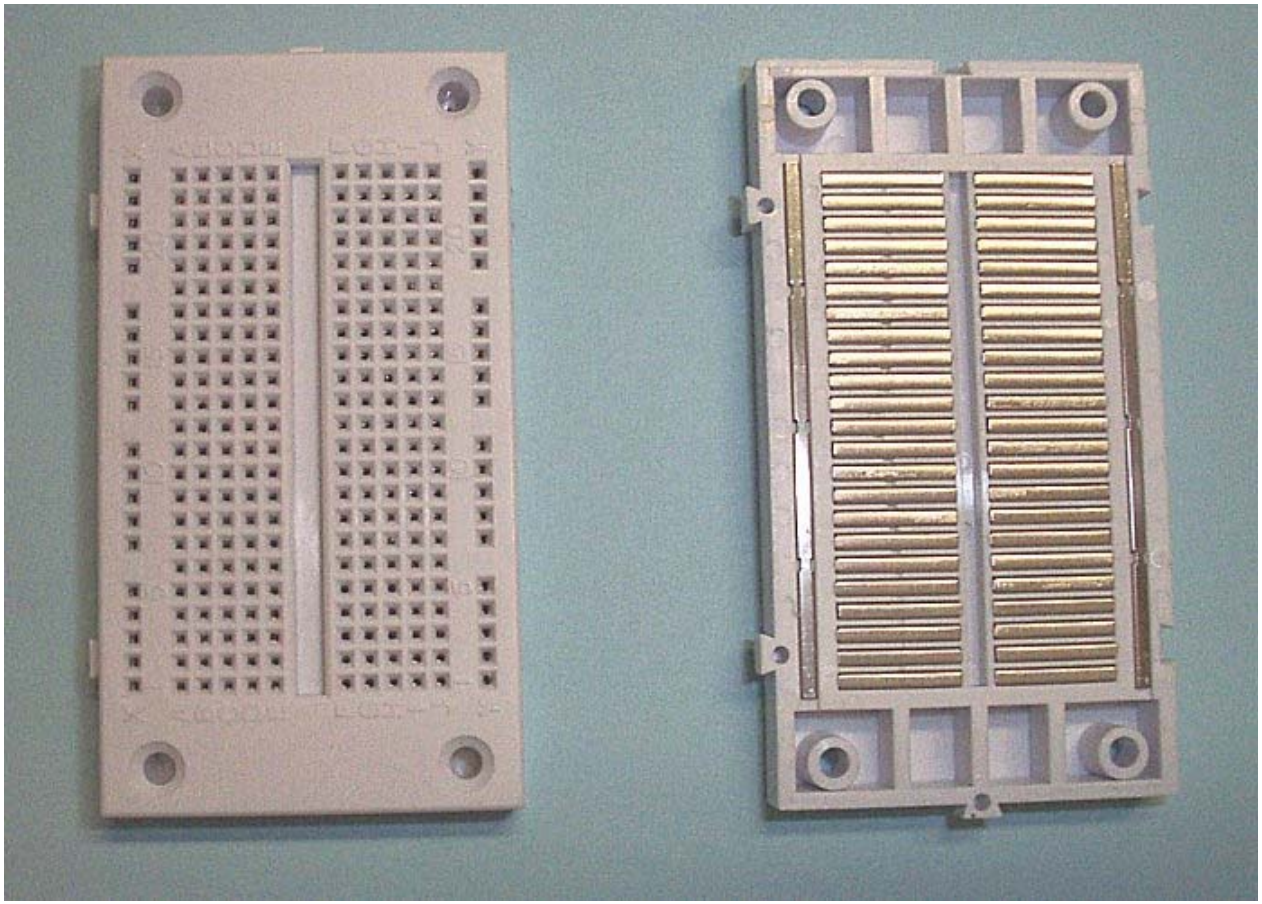


DEFINITIONS OF COMPONENTS

CONNECTORS:

Hookup Wire 22AWG

Alligator Clips

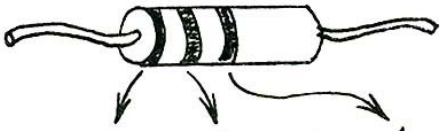


Breadboard

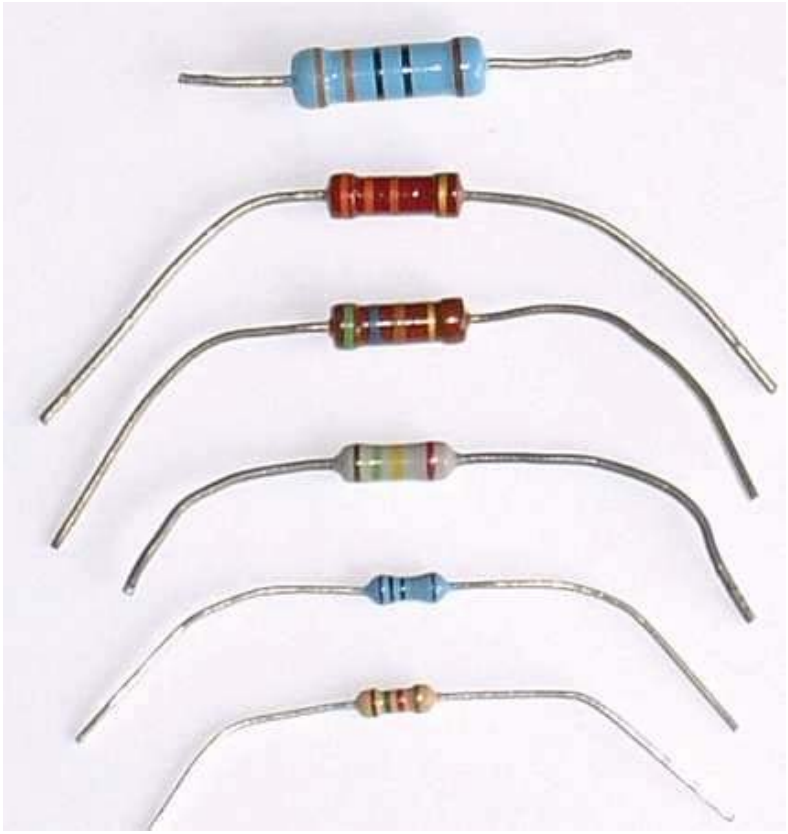
DEFINITIONS OF COMPONENTS

RESISTORS:

RESISTOR COLOR CODE:



BLACK	0	0	×	1
BROWN	1	1	×	10
RED	2	2	×	100
ORANGE	3	3	×	1,000
YELLOW	4	4	×	10,000
GREEN	5	5	×	100,000
BLUE	6	6	×	1,000,000
VIOLET	7	7	×	10,000,000
GRAY	8	8	×	100,000,000
WHITE	9	9		

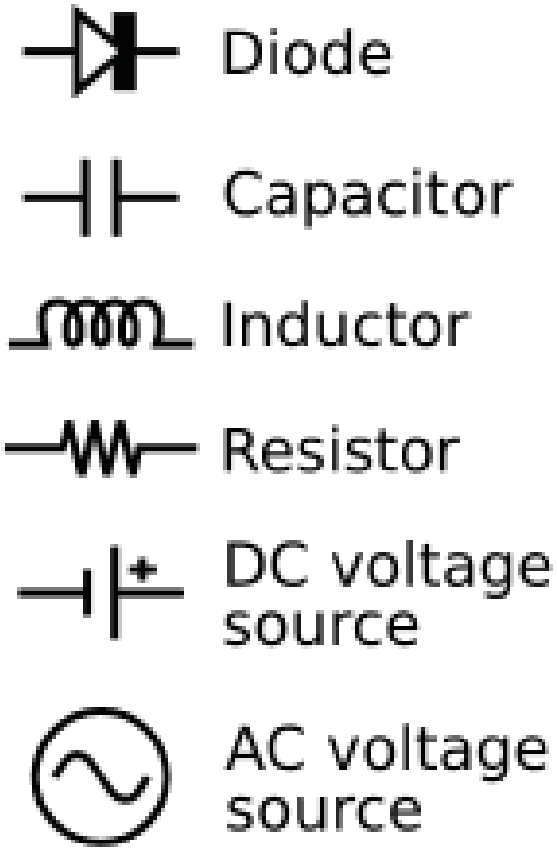


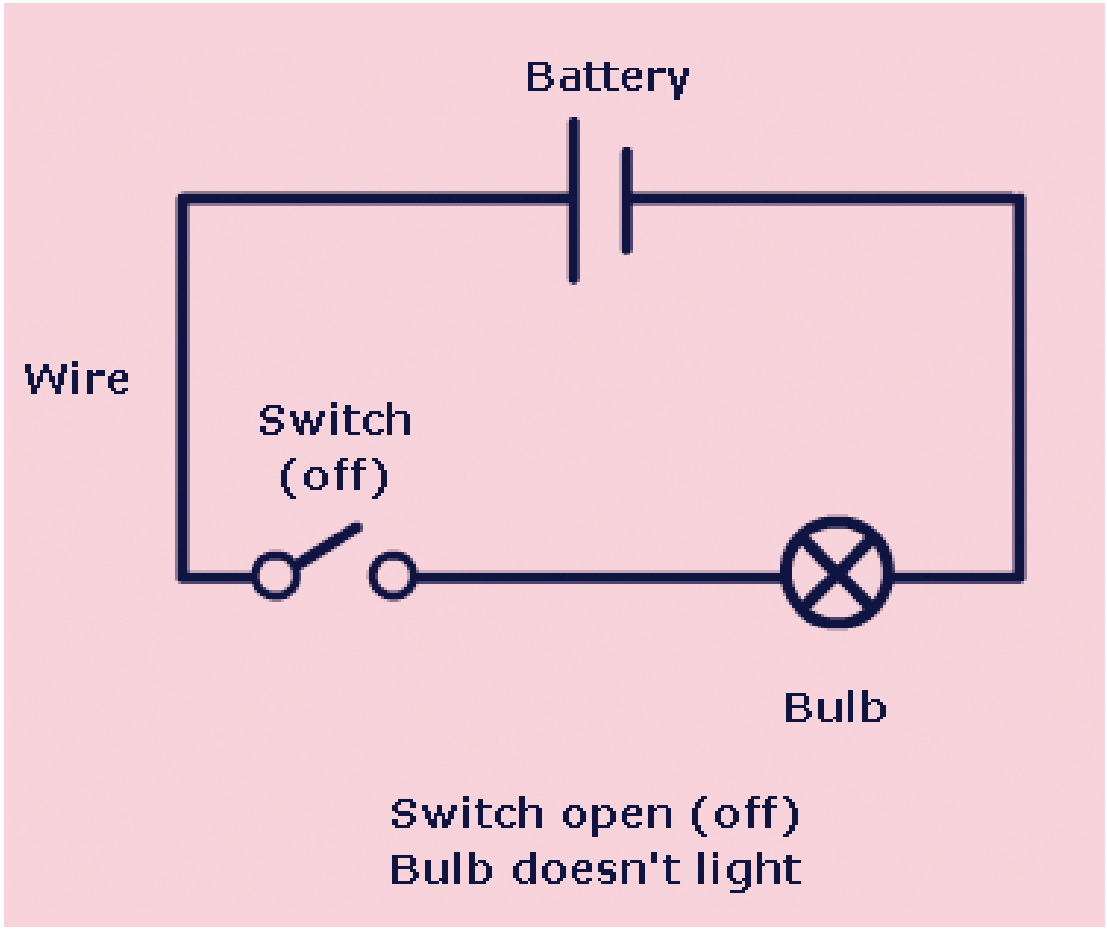
BB ROY of Great Britain had a Very Good Wife

http://www.dannyg.com/examples/res2/resistor.htm
http://itp.nyu.edu/physcomp/Labs/Components

READING SCHEMATICS

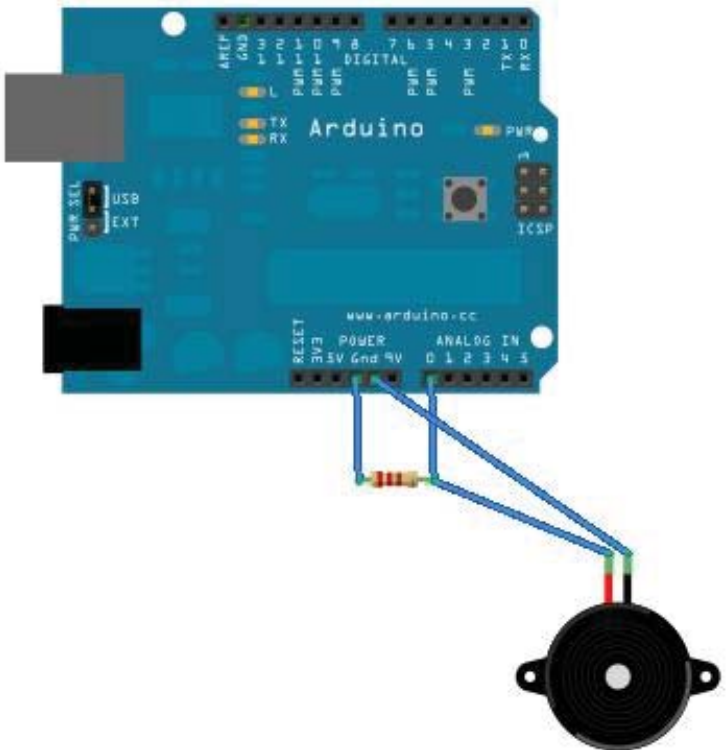
_ A schematic is a diagram representing a circuitin which:Symbols represent components and lines represent wires.

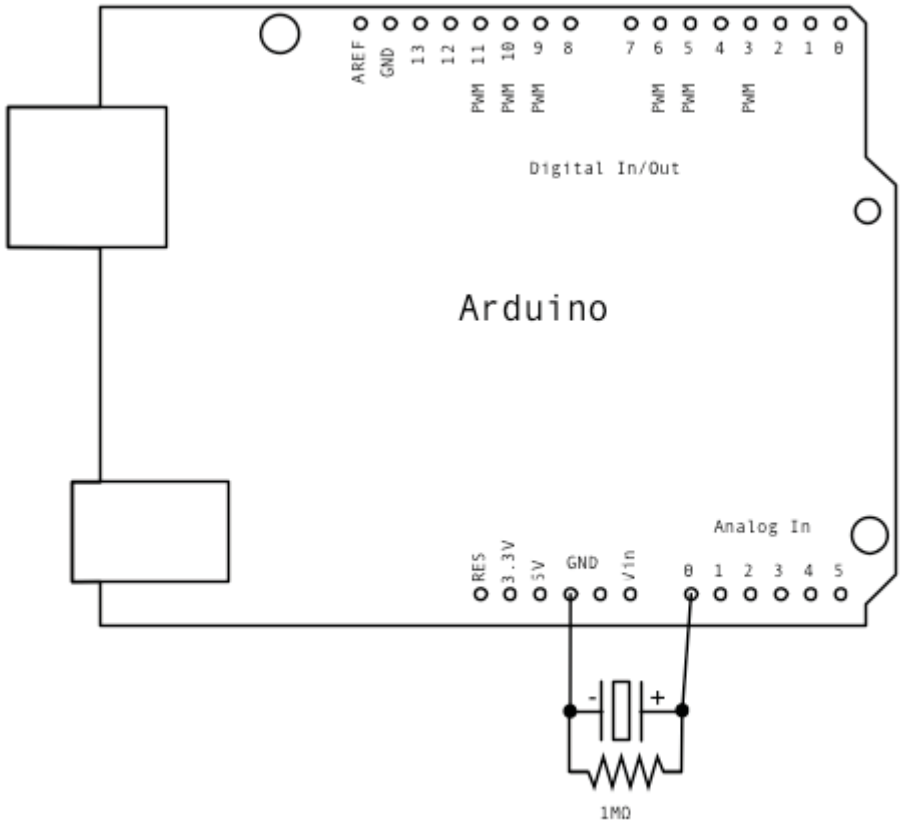




http://www.rapidtables.com/electric/electrical_symbols.htm

READING SCHEMATICS



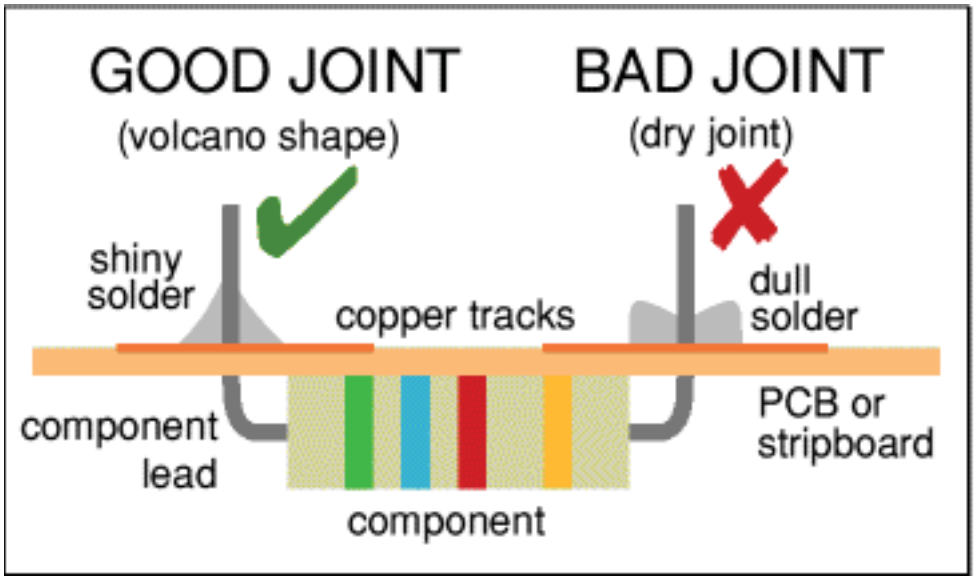


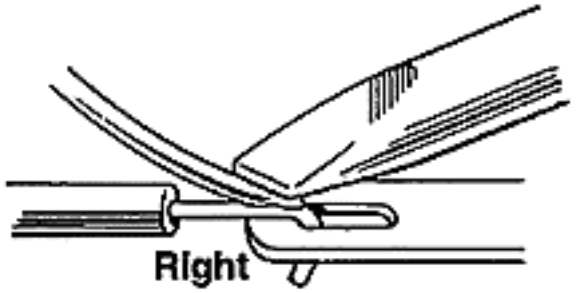
SOLDERING

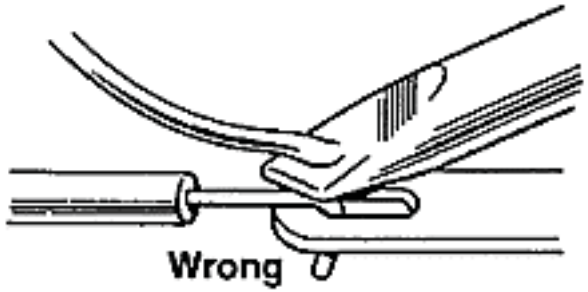
TIPS:

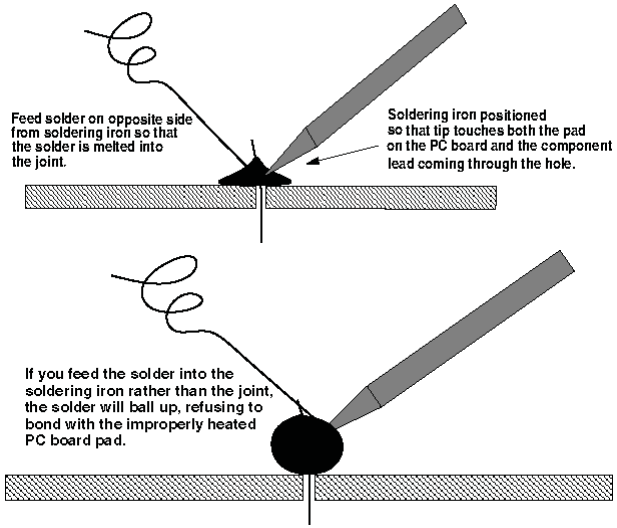
- Never touch the tip of solder iron.
- Always return the iron to the stand once soldering is done.
- Well-ventilated area.
- Wash your hands after using solder.

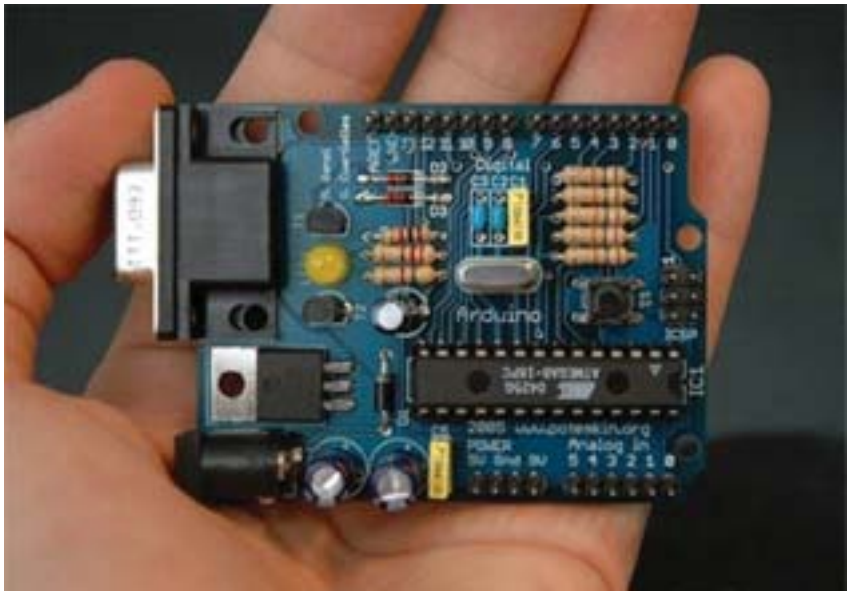
Always feed solder to joint and solder iron to solder.
Good solders are quick joints.









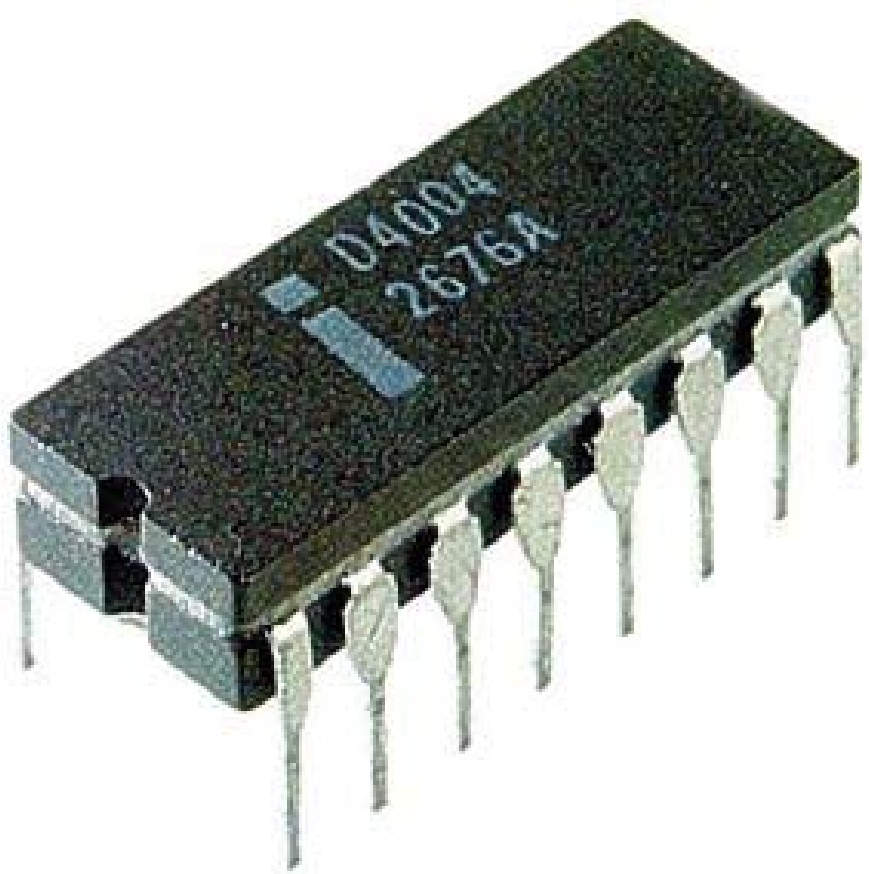


Digital In/Out

* Overview of Microcontrollers
* Intro to Arduino

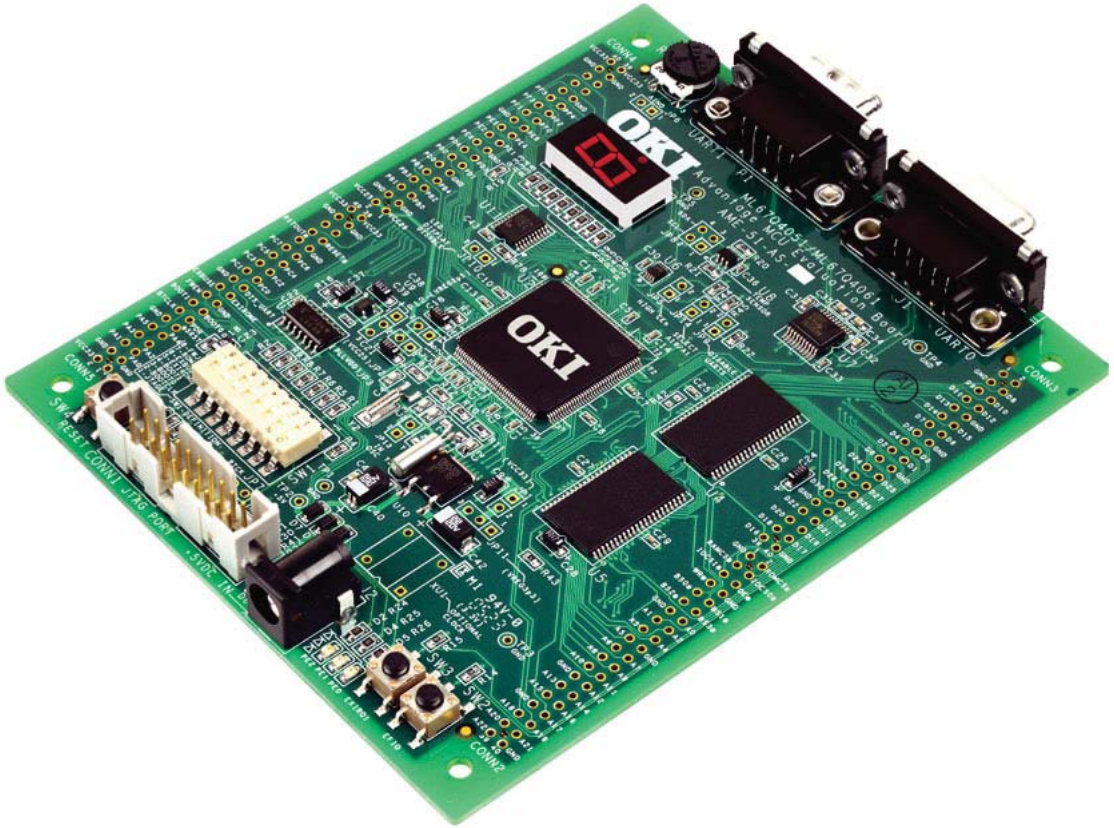
MICROPROCESSOR

_incorporates most or all of the functions of a central processing unit (CPU) on a single integrated circuit (IC).

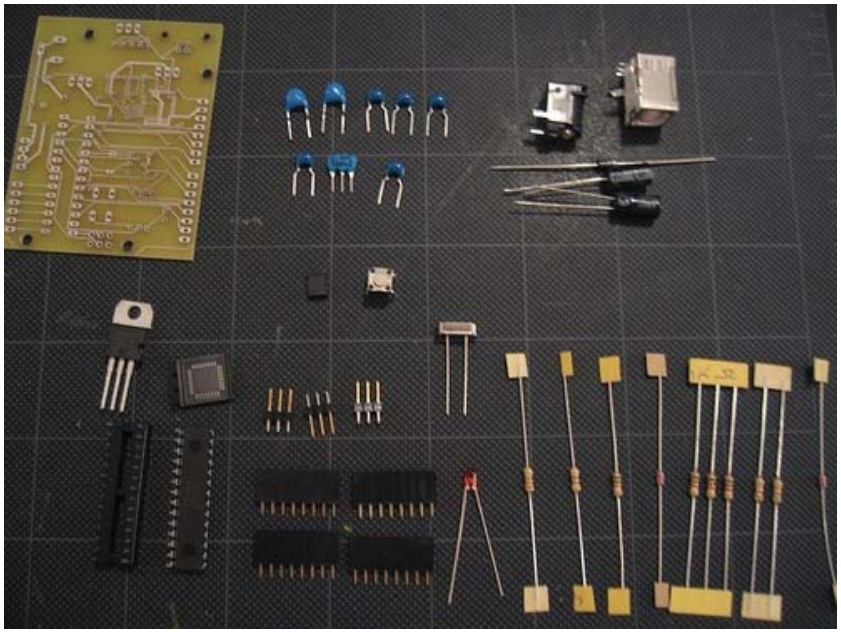


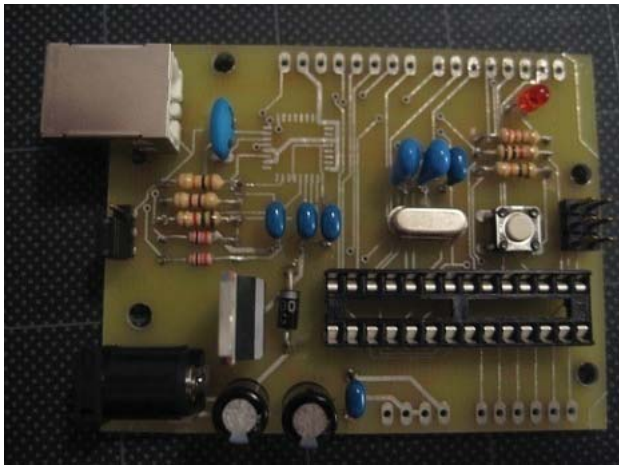
MICROCONTROLLER

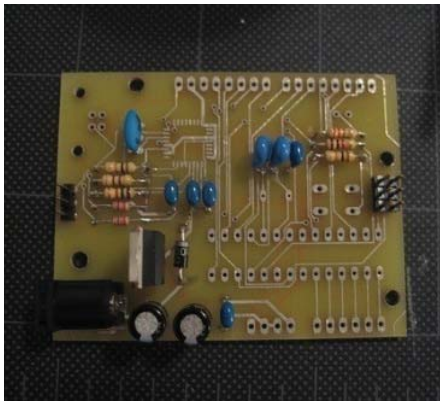
_Consists of microprocessor combined with support functions such as a crystal oscillator, timers, watchdog, small amount of RAM, serial and analog I/O etc.
_Microcontrollers are designed for small or dedicated applications.

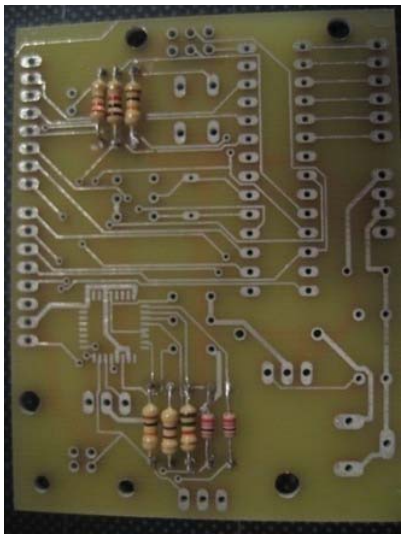


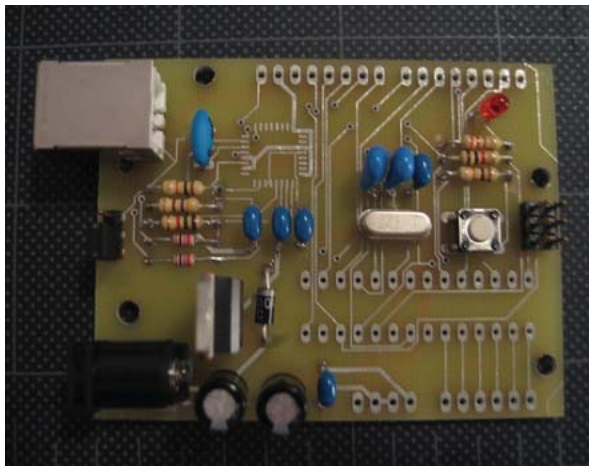
MICROCONTROLLER-DIY

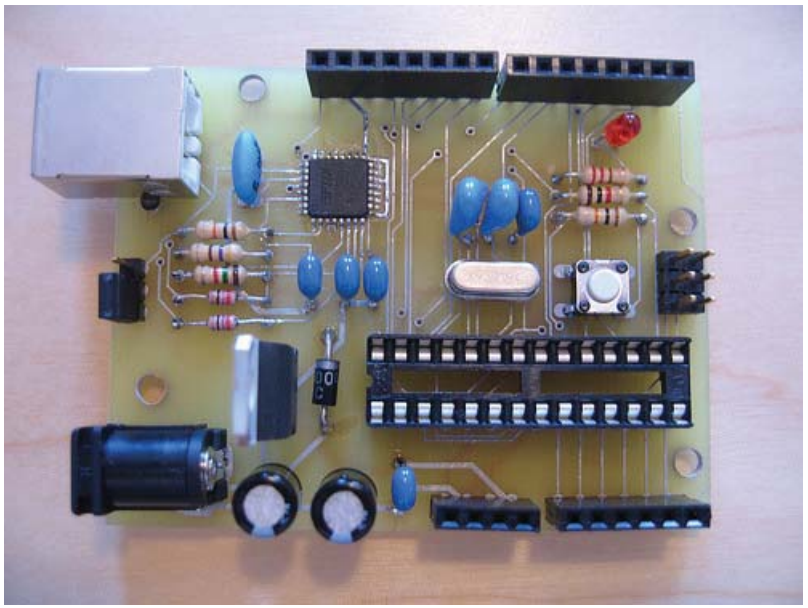










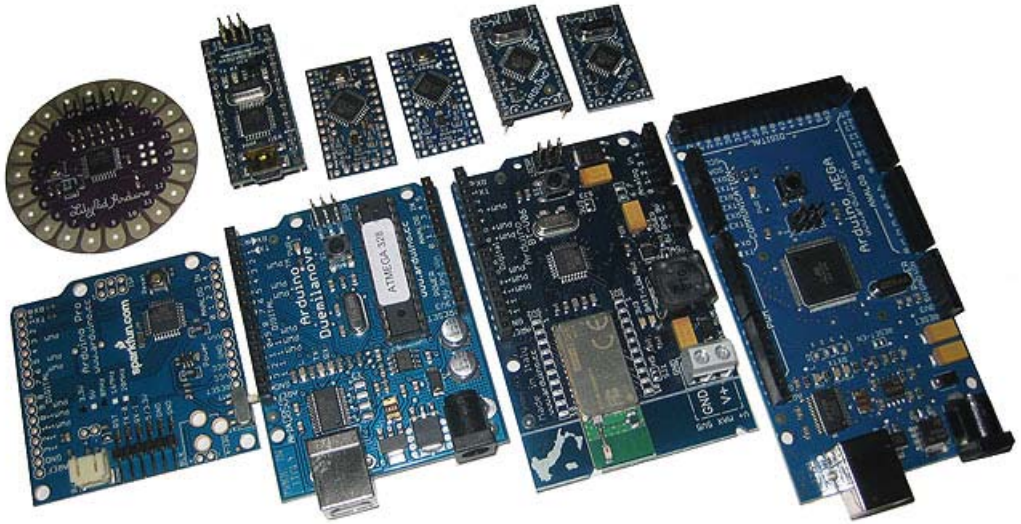


ARDUINO

_ Is a specific type of microcontrollers
_Can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.

ARDUINO TYPES

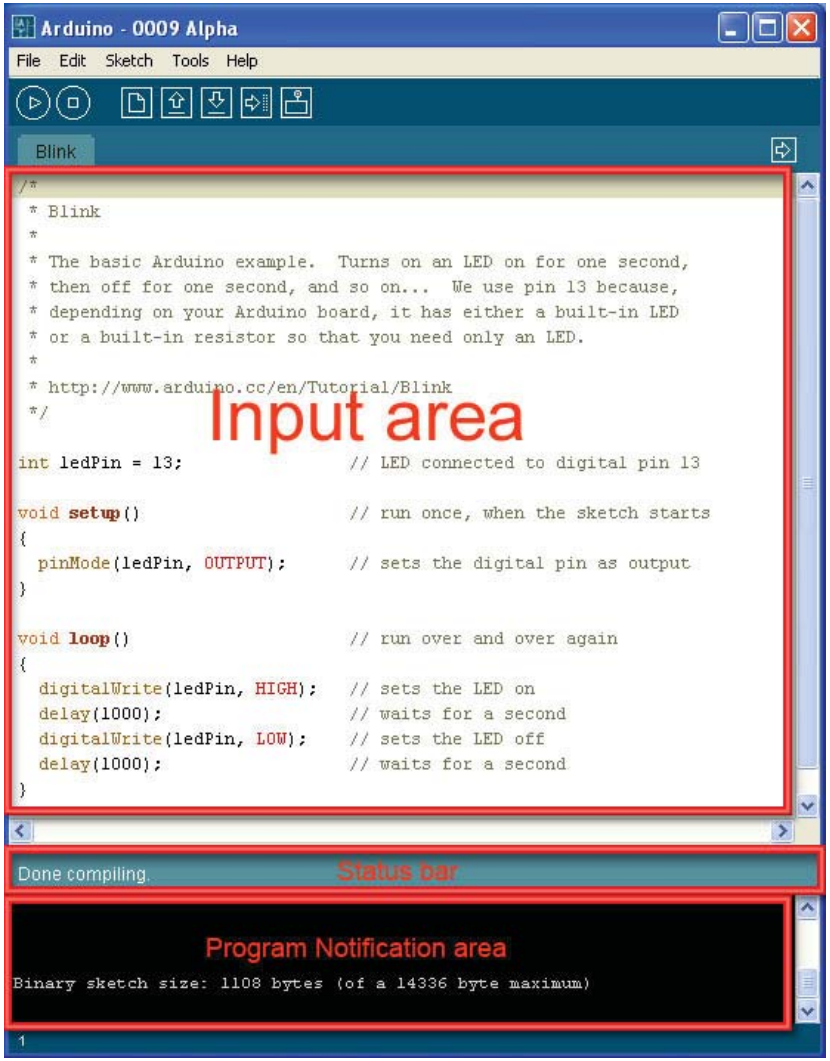
_Manufactured differently which serve different chip size / needs.



Mini light
Pro Mini
Nano
Lylipad
Pro
Diecimila,
Duemilanove
Bluetooth
Mega

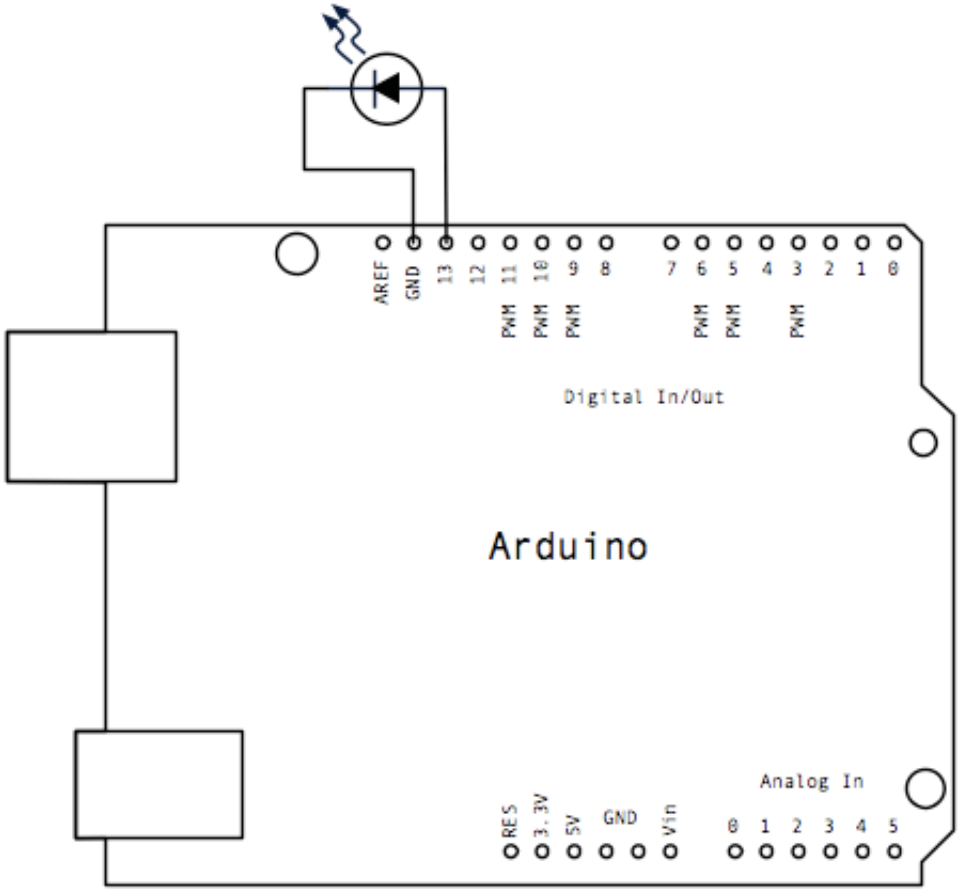
PROCESSOR PROGRAMMING

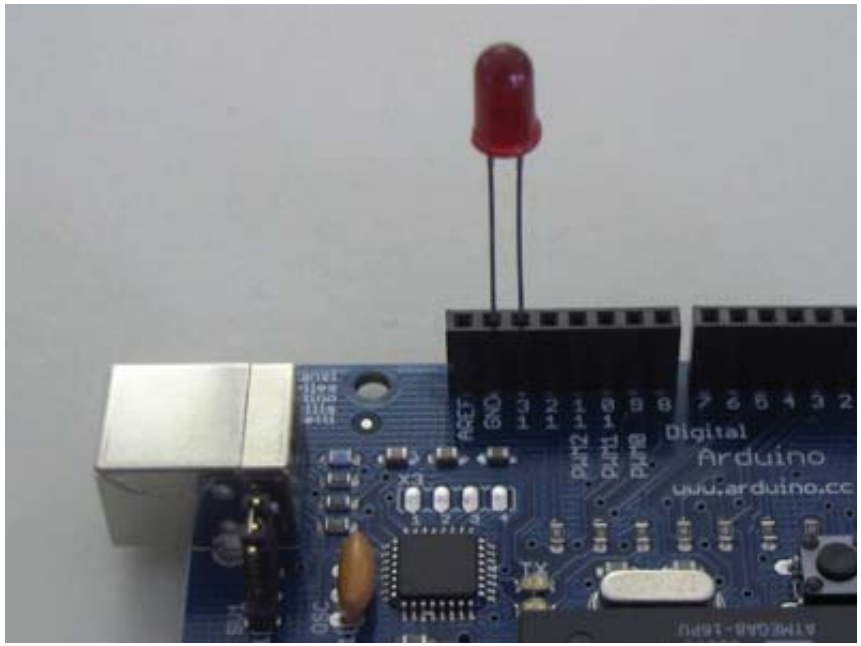
_Programming a processor is basically to assignnew functions to the microprocessor.
_Arduino uses C coding to program its microprocessor
_Arduino developed its own platform (software) to code its processor. www.arduino.cc




```
void setup()
{
    Serial.begin(9600);
}
void loop()
{


}
```

BLINK HARDWARE

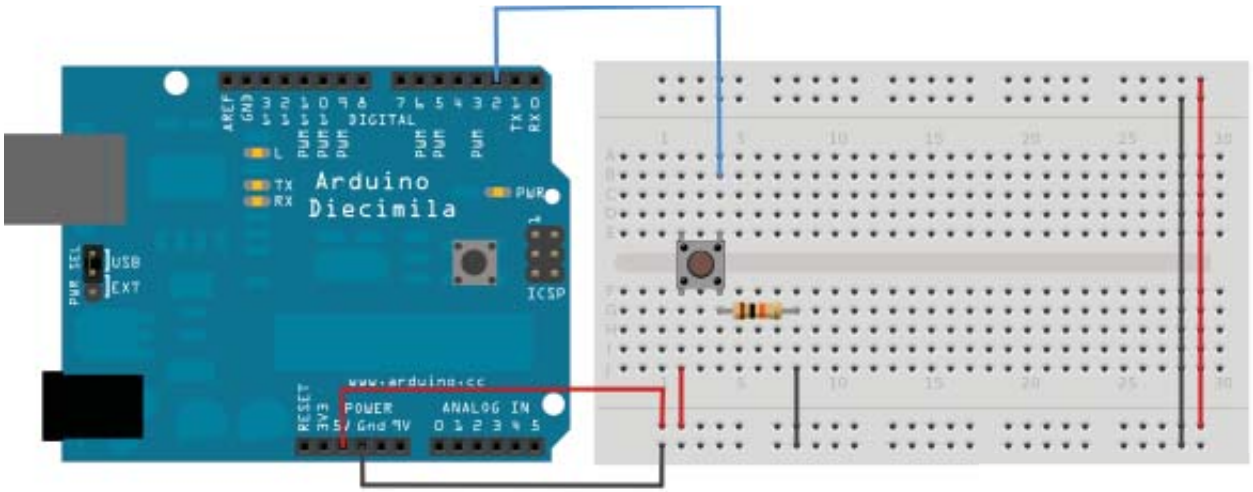


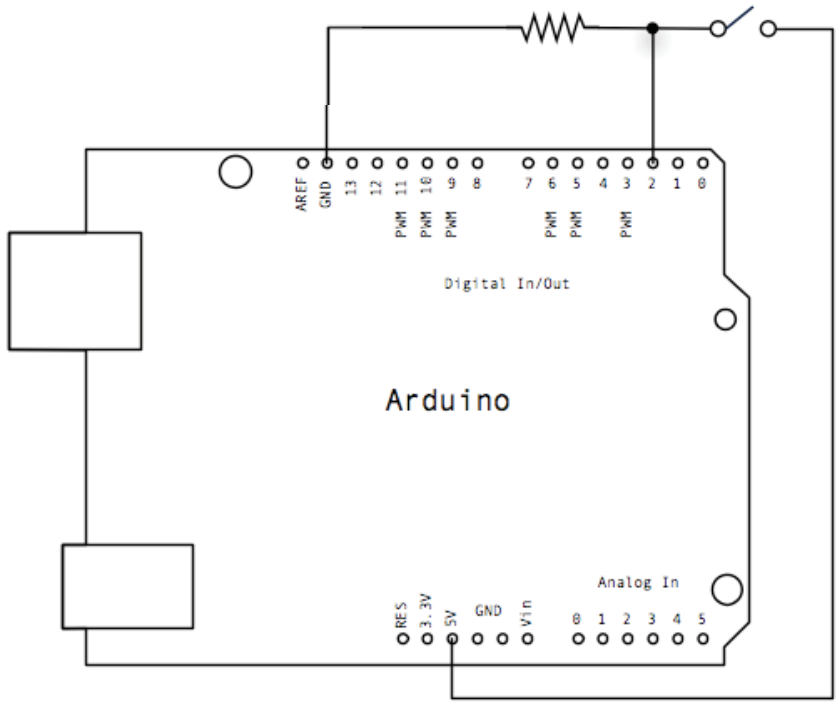


BLINK CODE

```
int ledPin =  13;    // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup()   {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);   // set the LED on
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);                  // wait for a second
}
```

BUTTON HARDWARE





BUTTON HARDWARE

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```