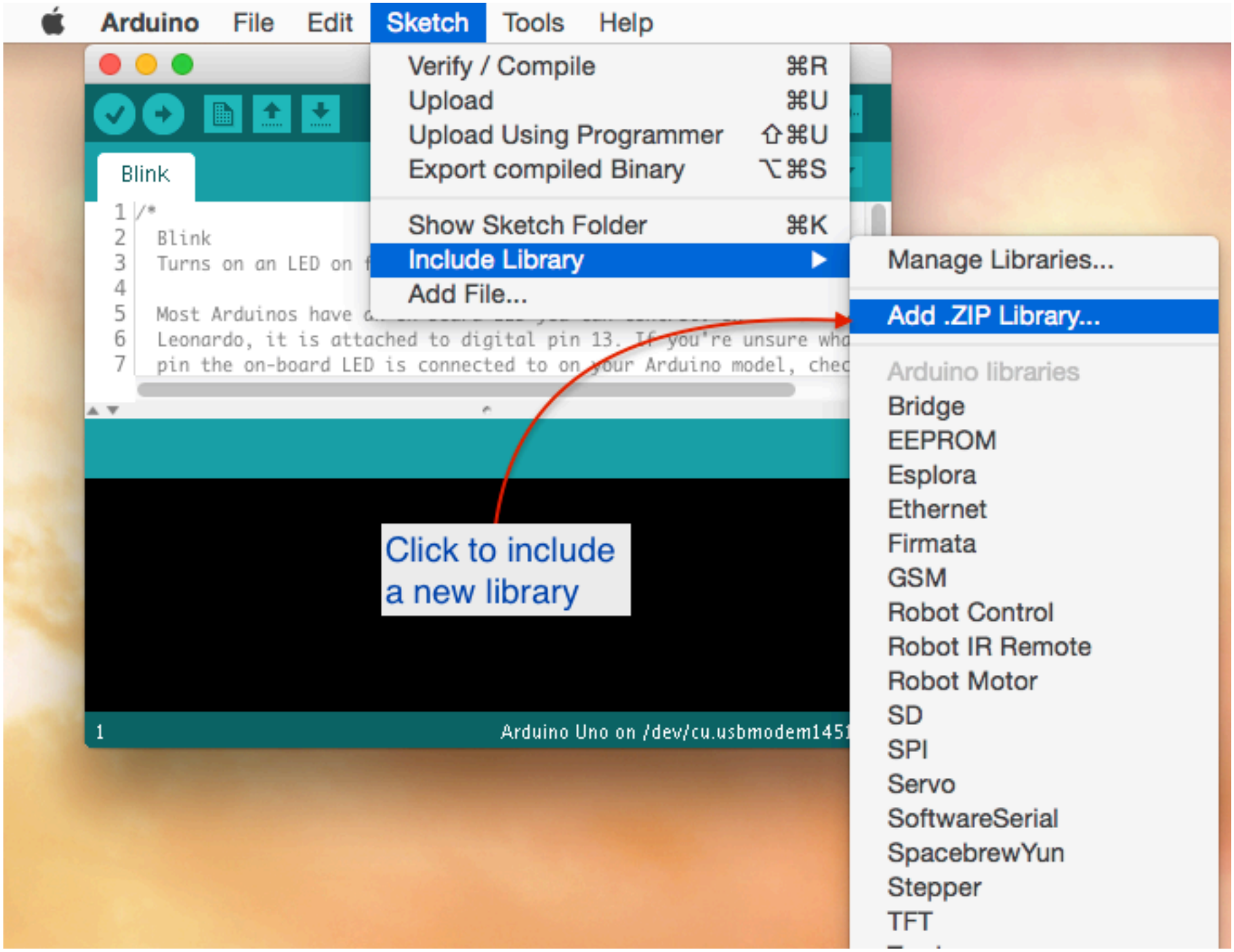
PROGRAMMING WITH ARDUINO

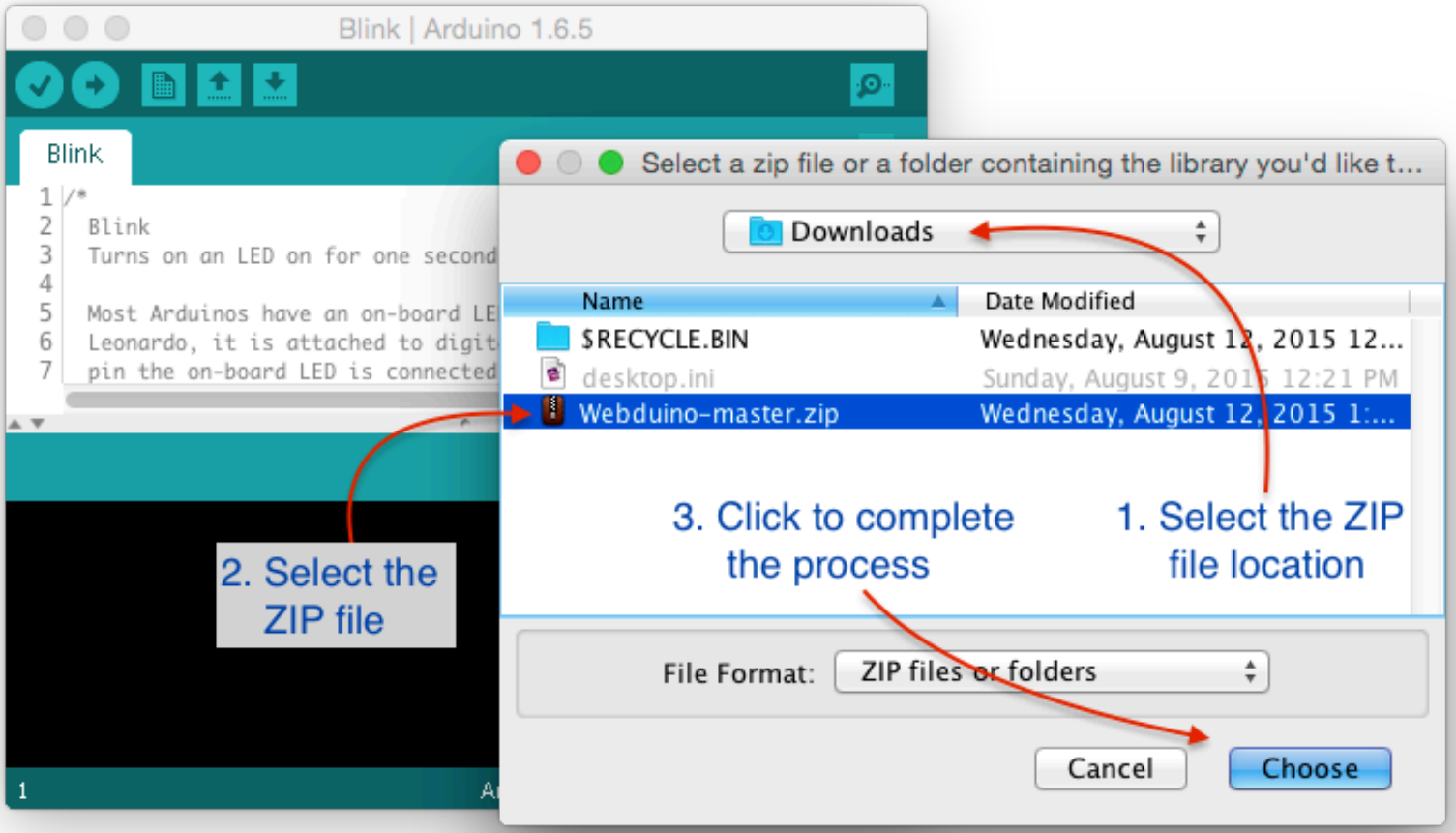
Libraries

Arduino Libraries

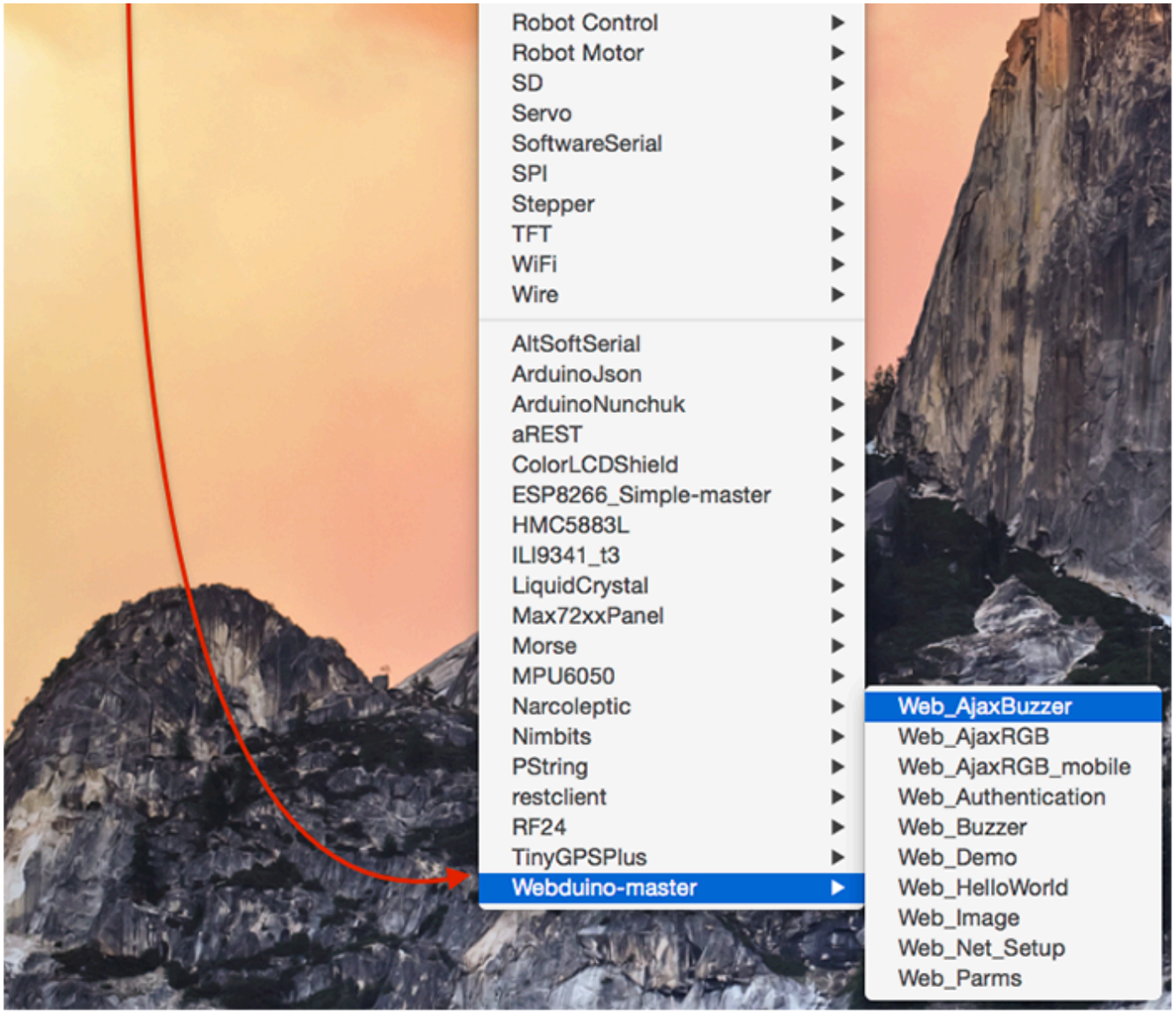
- If there is a library that you need but is not included with the IDE, you can install it. Let's look at an example.
- Download the ZIP file on your computer. It doesn't matter what platform you are on; the libraries work the same regardless of whether you are on Windows, Mac or Linux.
- Also, don't worry about extracting the files from the ZIP archive. The newer versions of the Arduino IDE have an easy library installer that takes care of extracting the library from the ZIP file and copying the files to the right location.
- Assuming the library ZIP file is in your Downloads folder, start the Arduino IDE. Then click on "Sketch → Include Library → Add .ZIP Library…", like this:


Arduino File Edit Sketch Tools Help
Verify / Compile ⌘R
Upload ⌘U
Upload Using Programmer ⇧⌘U
Export compiled Binary ⌥⌘S
Show Sketch Folder ⌘K
Include Library
Add File...
Manage Libraries...
Add .ZIP Library...
Arduino libraries
Bridge
EEPROM
Esplora
Ethernet
Firmata
GSM
Robot Control
Robot IR Remote
Robot Motor
SD
SPI
Servo
SoftwareSerial
SpacebrewYun
Stepper
TFT
Blink
Click to include a new library
Arduino Uno on /dev/cu.usbmodem1451

A new dialogue box will pop up. Browse to the location of the ZIP file, select it, and click on Choose to complete the process:



- When you click on “Choose”, the dialogue box will disappear, but nothing else is going to happen. No confirmation, no sound… To make sure that the Webuino library was actually installed, you can look for the example sketches that most libraries include.
- Go to File → Examples, and look at the bottom of the list for your new library:


Robot Control
Robot Motor
SD
Servo
SoftwareSerial
SPI
Stepper
TFT
WiFi
Wire
AltSoftSerial
ArduinoJson
ArduinoNunchuk
aREST
ColorLCDShield
ESP8266_Simple-master
HMC5883L
ILI9341_t3
LiquidCrystal
Max72xxPanel
Morse
MPU6050
Narcoleptic
Nimbits
PString
restclient
RF24
TinyGPSPlus
Webduino-master
Web_AjaxBuzzer
Web_AjaxRGB
Web_AjaxRGB_mobile
Web_Authentication
Web_Buzzer
Web_Demo
Web_HelloWorld
Web_Image
Web_Net_Setup
Web_Parms

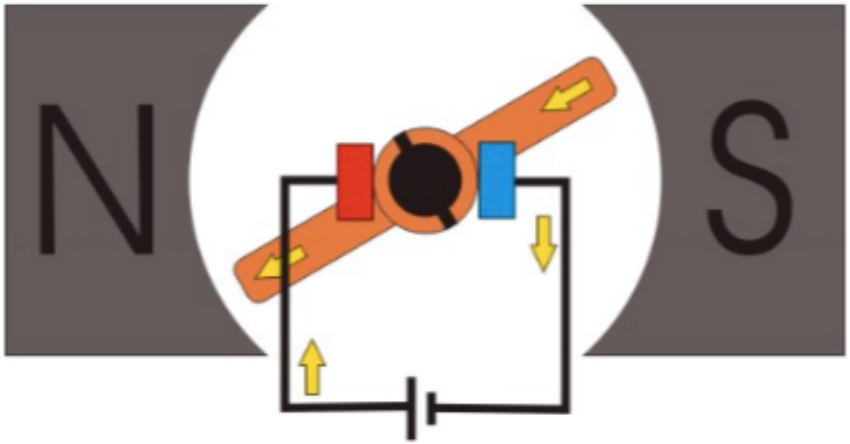
How to include a new library

#include <Name_of_library.h>

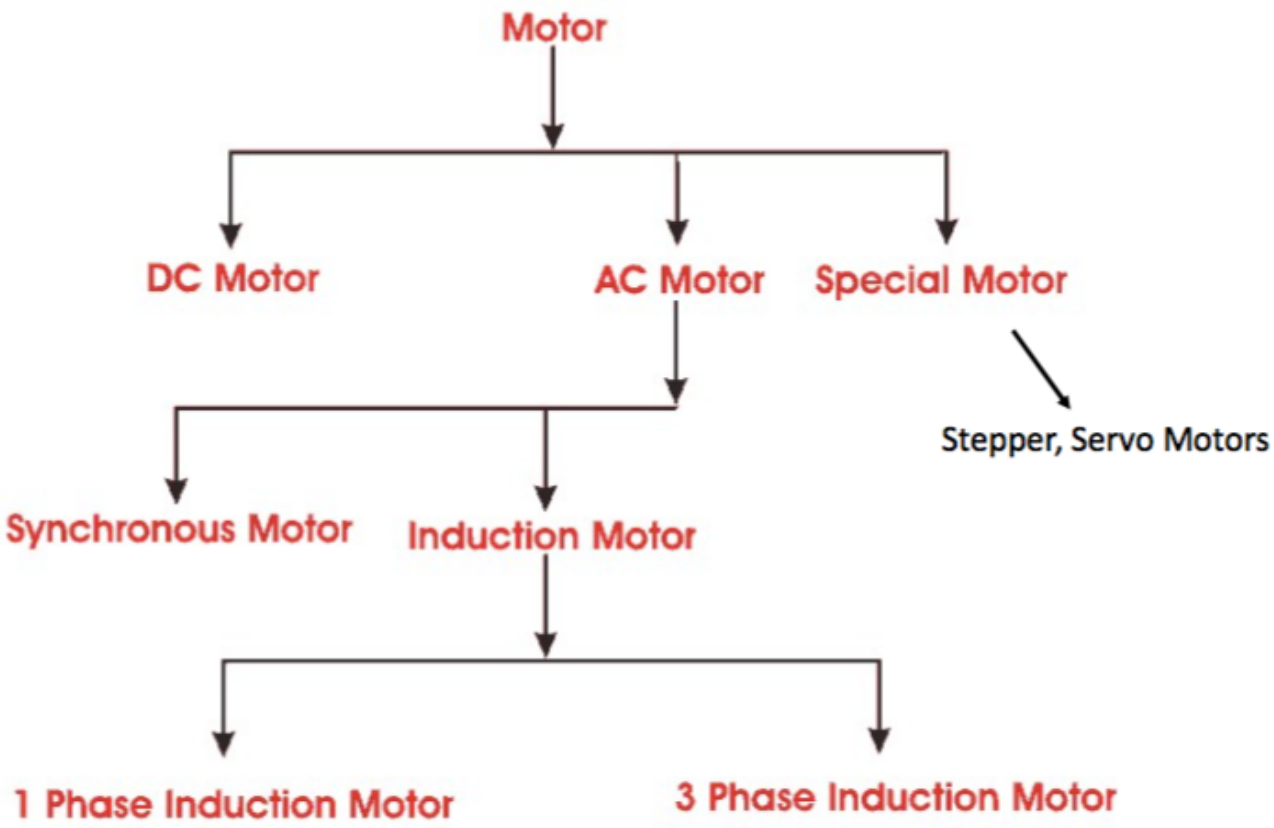
Motors

Motors

- A motor is an electro-mechanical device that converts electrical energy to mechanical energy.
- The very basic principal of functioning of an electrical motor lies on the fact that force is experienced in the direction perpendicular to magnetic field and the current, when field and current are made to interact with each other.



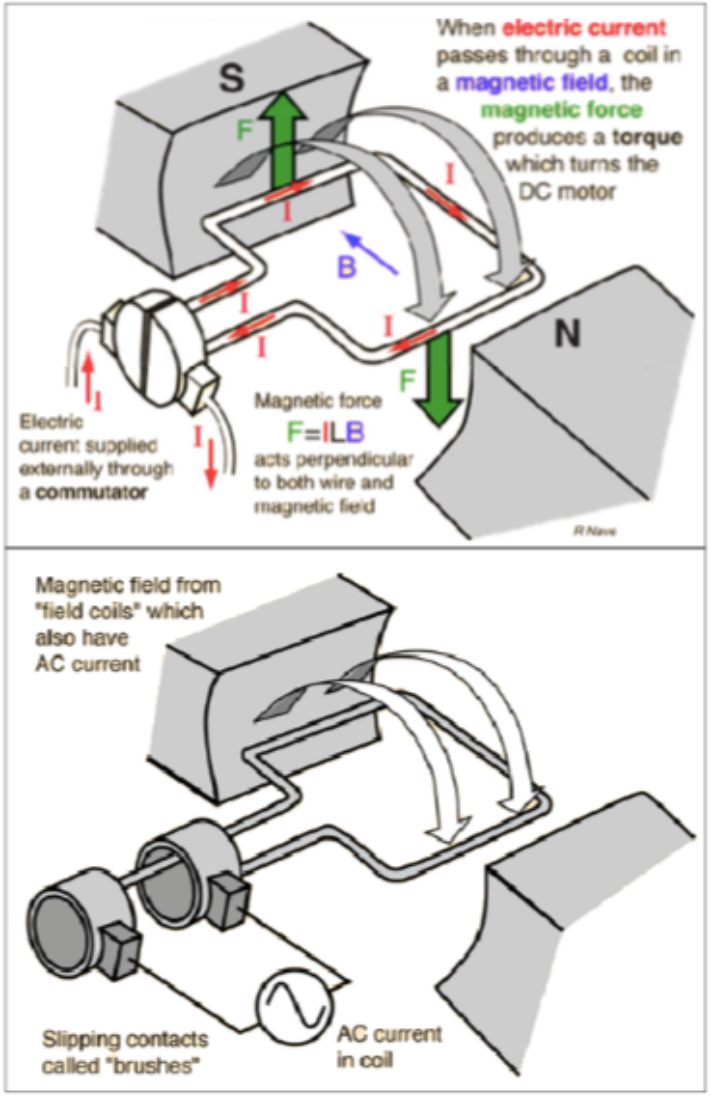
Motors


Motor
DC Motor
AC Motor
Special Motor
Stepper, Servo Motors
Synchronous Motor
Induction Motor
1 Phase Induction Motor
3 Phase Induction Motor

Types of Motors

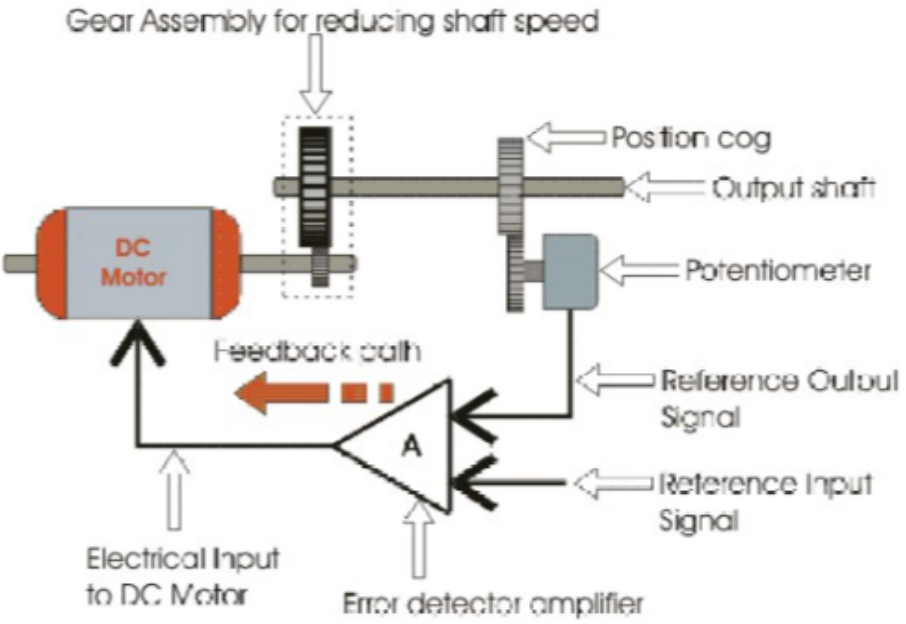
The DC motor as the name suggests, is the only one that is driven by **direct current**.
It's the most primitive version of the electric motor where **rotating torque is produced due to flow of current through the conductor inside a magnetic field**.

AC motors are driven by **alternating current.**
Here the rotor **is magnetically locked with stator rotating magnetic field and rotates with it.**
The speed of these machines are varied by varying the frequency (f) and number of poles (P).



Types of Motor 4 - Servo

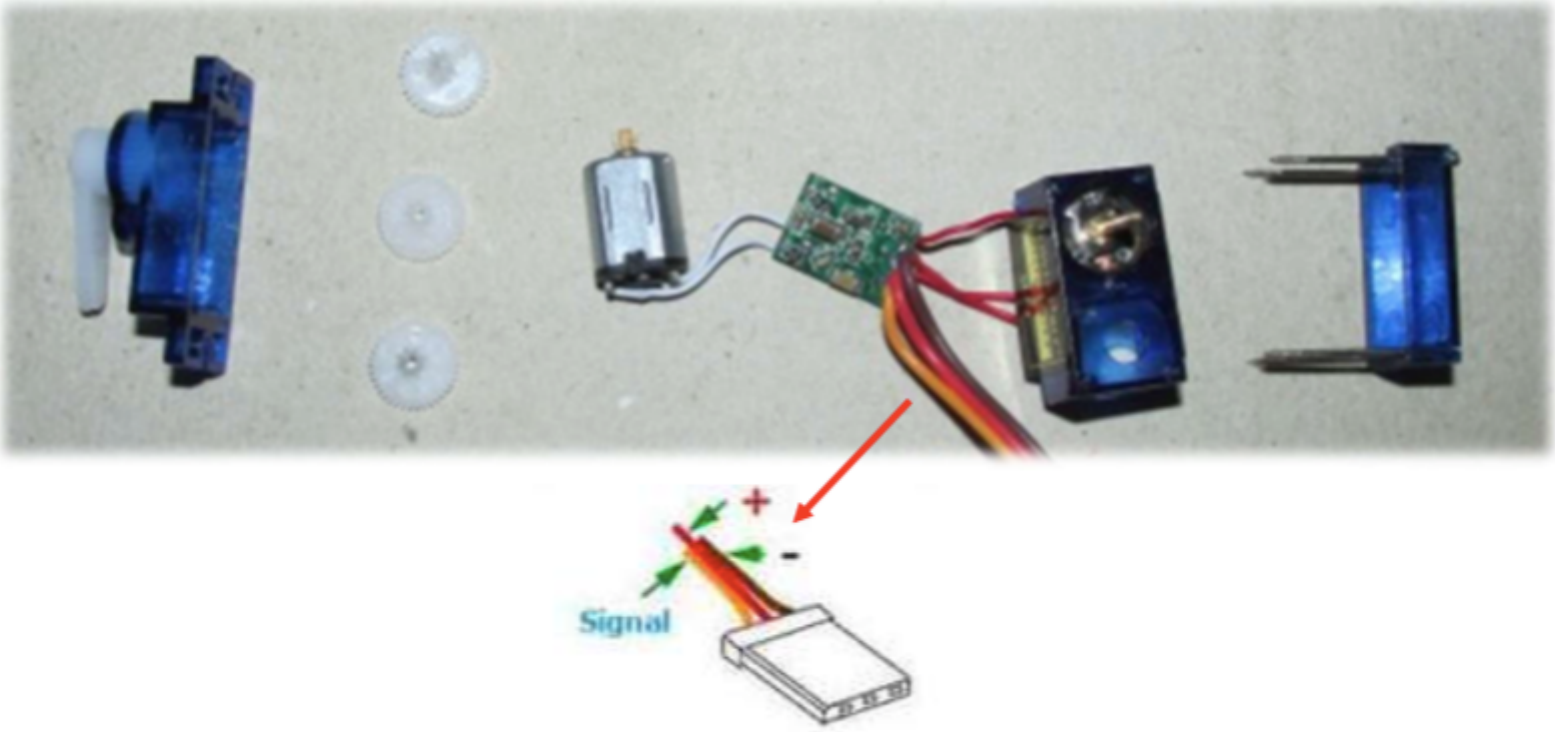
A servo system mainly consists of a **small DC motor**, a **potentiometer, gear arrangement** and a **feedback system.**



- The device is controlled by a feedback signal generated by comparing output signal and reference input signal. Hence, the primary task of a servomechanism is to **maintain the output of a system at the desired value** in the presence of disturbances.
- *During rotation of the shaft, the knob of the potentiometer also rotates* and creates an **varying electrical potential that is taken to the error detector feedback amplifier** along with the input reference commands i.e. input signal voltage.
- The gear mechanism is used to **step down the high rpm of the motor shaft to low rpm at the output shaft** of the servo system (small DC motor will rotate with high speed but the torque generated by its rotation will not be enough to move even a light load).

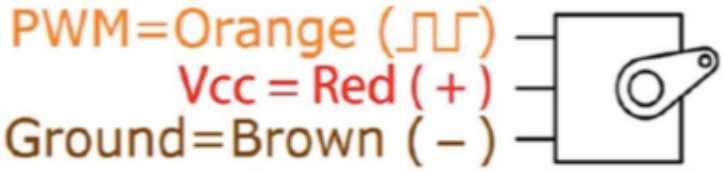
Servo: 2G90

The Tower Pro SG90 servo is one of the cheapest servo motors that you can find on the market. (Even if it is cheap, don't try to rotate the servo motor by hand because this may damage the motor!!)

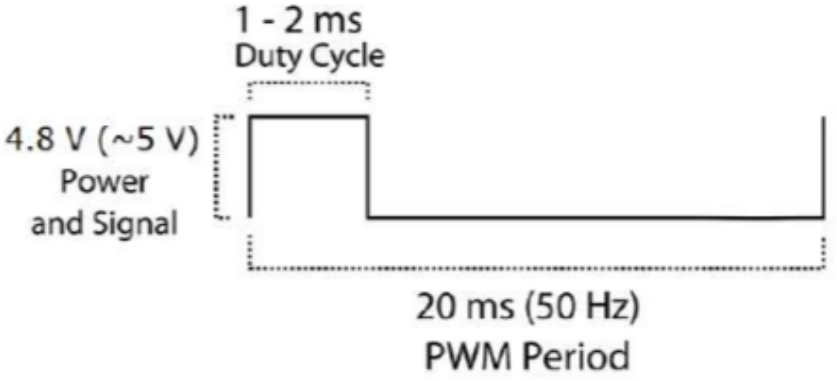


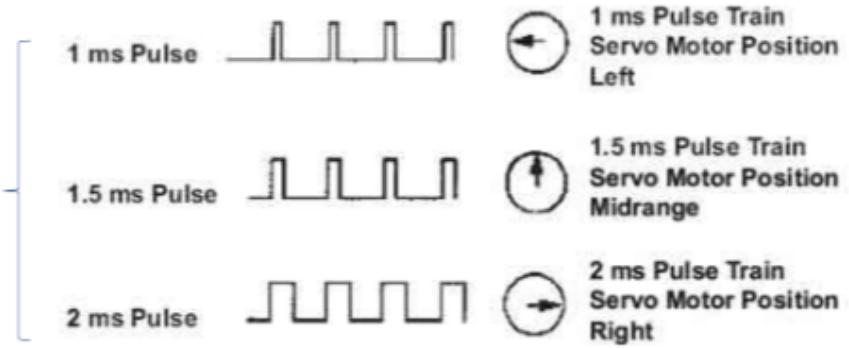
Servo Motor - Control

The servo motor has three terminals.
1. Position signal (PWM Pulses)
2. V_{cc} (From Power Supply)
3. Ground



The **servo motor angular position is controlled by applying PWM** (Pulse Width Modulation) **pulses** of specific width.
The *duration* of pulse varies from about 1 ms for 0 degree rotation to 2 ms for 180 degree rotation.
The pulses need to be given at *frequencies* of about 50Hz to 60Hz.





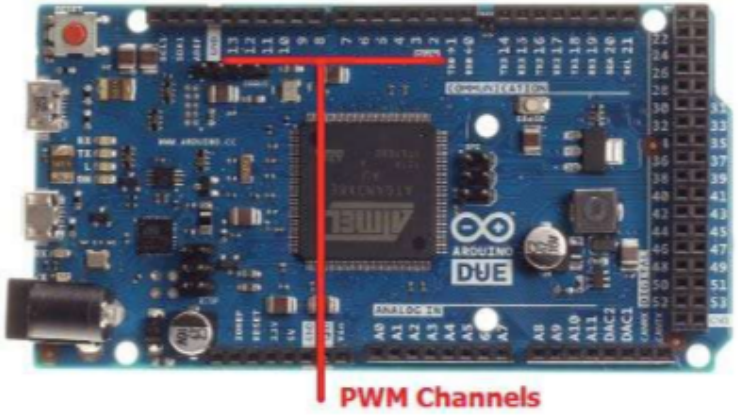
PWM with Arduino Due

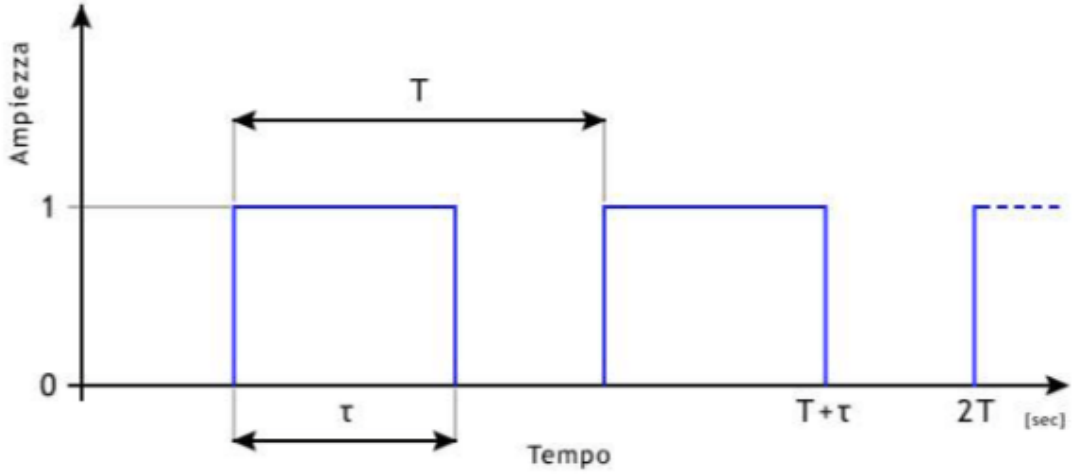
There are **12 PWM Channels** (Pin 2 to Pin 13)

pinMode(pin, OUTPUT)
analogWrite(pin, value)

d

The default PWM resolution is to 8-bit, which can be changed to 12-bit resolution using the *analogWriteResolution()* function.



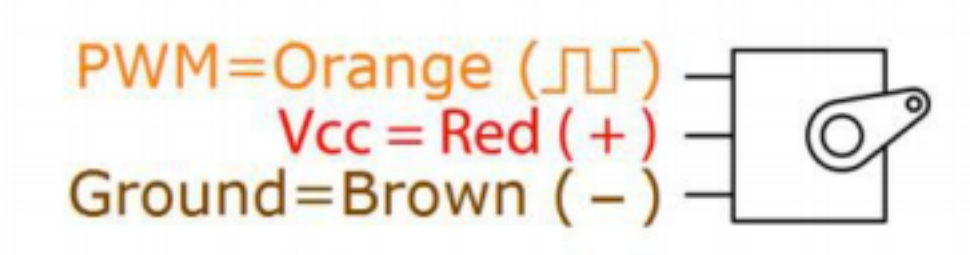


Duty cicle: $d = \frac{\tau}{T}$

Third Program

Micro Servo 9g

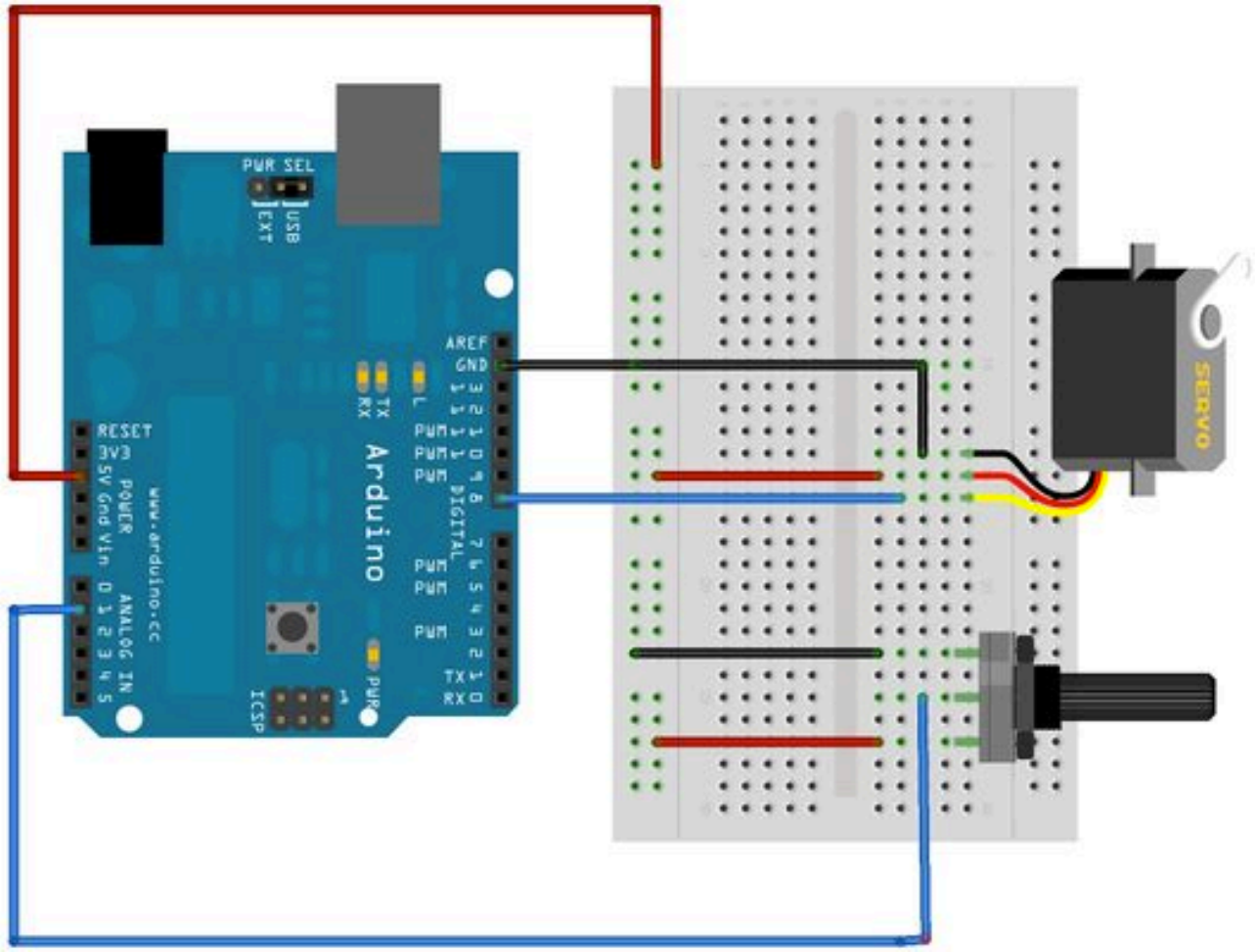
Micro Servo 9g SG90



Library: http://playground.arduino.cc/ComponentLib/Servo

Datasheet: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

Servo Motor Circuit (example)



Servo Motor Sketch

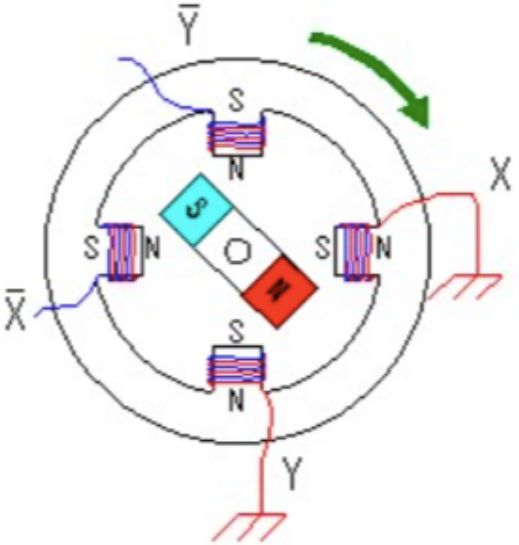
```
#include <Servo.h> // include la Libreria Servo.h

Servo servoMotor;  // Crea l'oggetto di tipo Servo, servoMotor sarà l'oggetto su
 cui opererai.
int valore;        // Inizializza una variabile di tipo intero "valore" il cui v
alore sarà la posizione da impartire al servo.

void setup() {
   servoMotor.attach(8);  // Lega l'oggetto servoMotor al pin a cui abbiamo coll
egato il nostro servo, in questo caso il pin 8.
}

void loop()
 {
   valore = analogRead(A0);  // Legge il valore analogico del potenziometro sul
 pin A0
   valore = map(valore, 0, 1023, 0, 180);  // "Mappa" i valori di una lettura an
alogica (che vanno quindi da 0 a 1023) a valori che vanno da 0 a 180.

   servoMotor.write(valore);  // con il metodo write() passi all'oggetto servoMo
tor la posizione che deve raggiungere.
   delay(15);
  }
```

Stepper

A stepper motor is a **type of DC motor that rotates in steps**.
When electrical signal is applied to it, the motor rotates in steps:

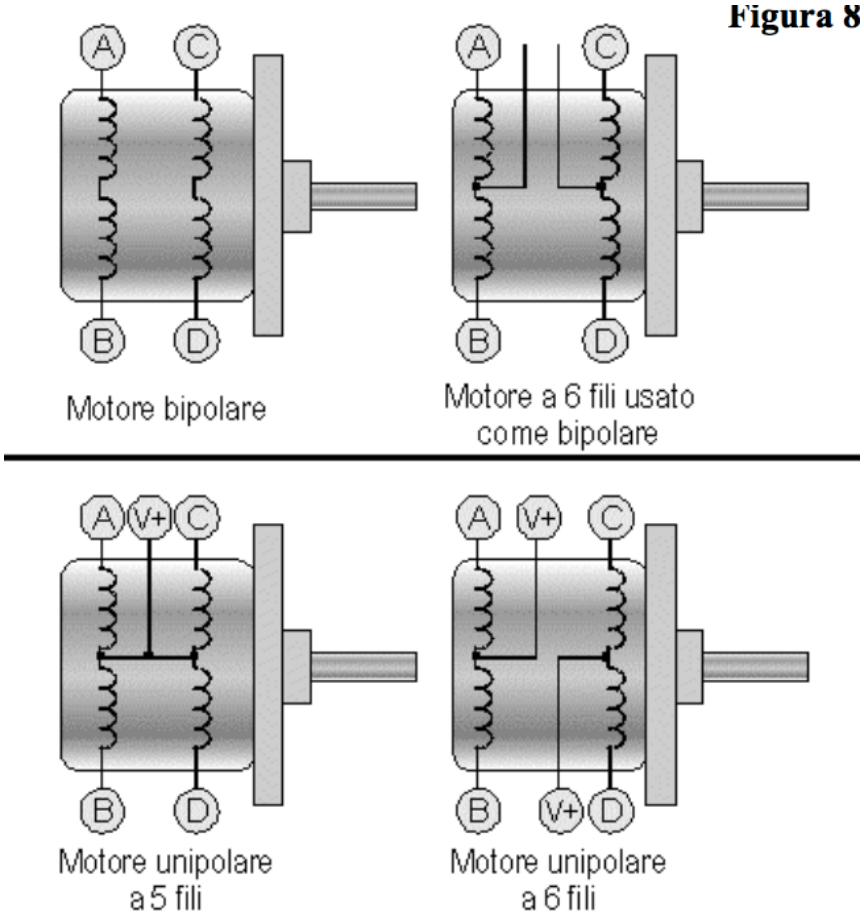
- The **speed of rotation** depends on the *rate at which the electrical signals are applied*;
- The **direction of rotation** is dependent on the *pattern of pulses* that is followed.

A stepper motor is made up of a **rotor**, which is normally a *permanent magnet*. A **stator** is another part which is in the form of *winding*.
The magnetic property of the stator changes and it will selectively attract and repel the rotor, thereby resulting in a stepping motion for the motor.
In order to get correct motion of the motor, a **stepping sequence** has to be followed. This stepping sequence gives the *voltage that must be applied to the stator phase*.
Normally a 4 step sequence is followed. When the sequence is followed from step 1 to 4, we get a **clock wise rotation** and when it is followed from step 4 to 1, we get a **counter clockwise rotation**.

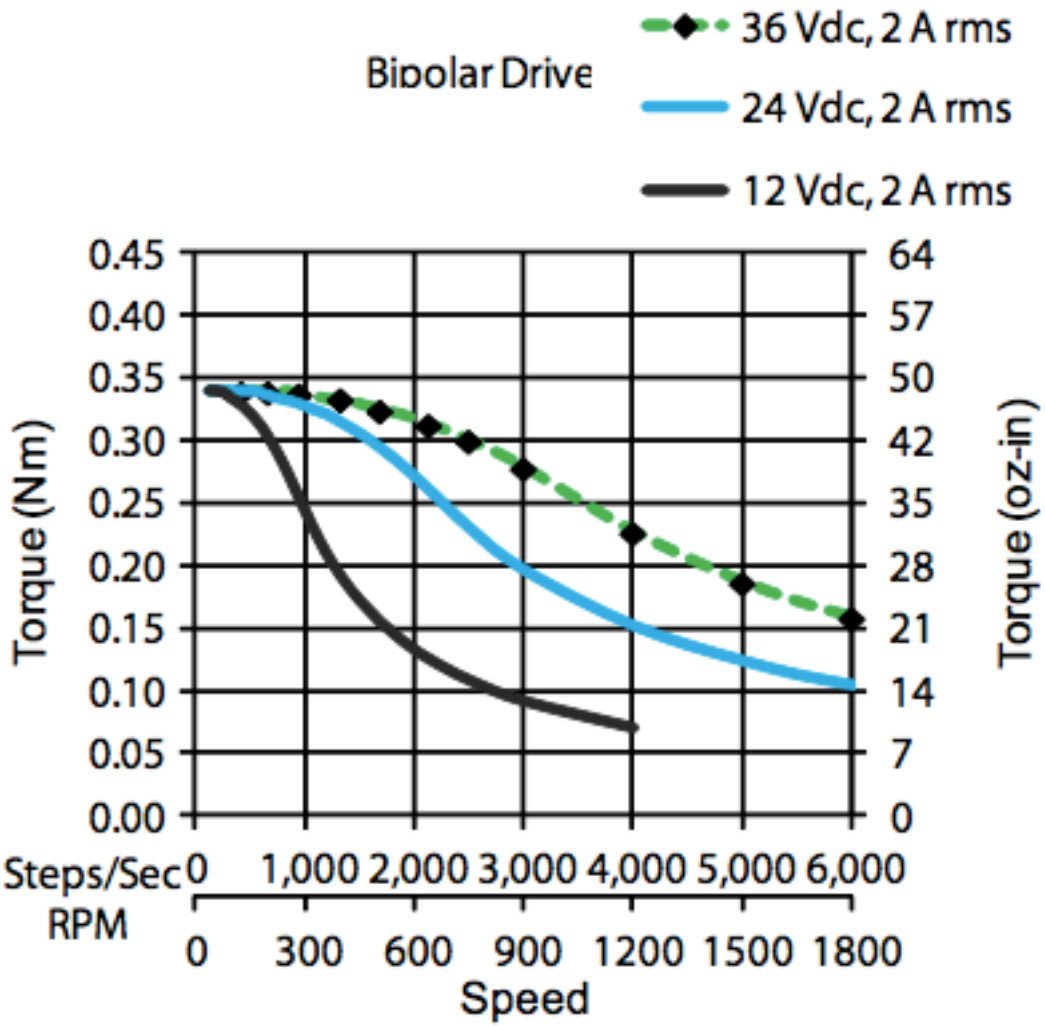


X	$\bar{X}$	Y	$\bar{Y}$
0	1	0	1
1	0	0	1
1	0	1	0
0	1	1	0

Stepper motor



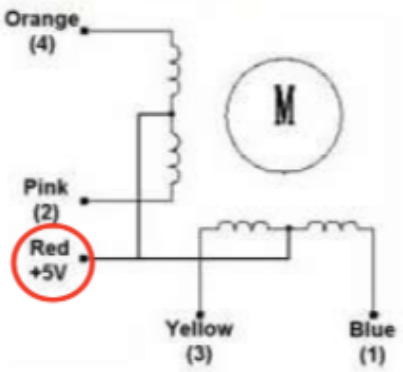
Stepper motor

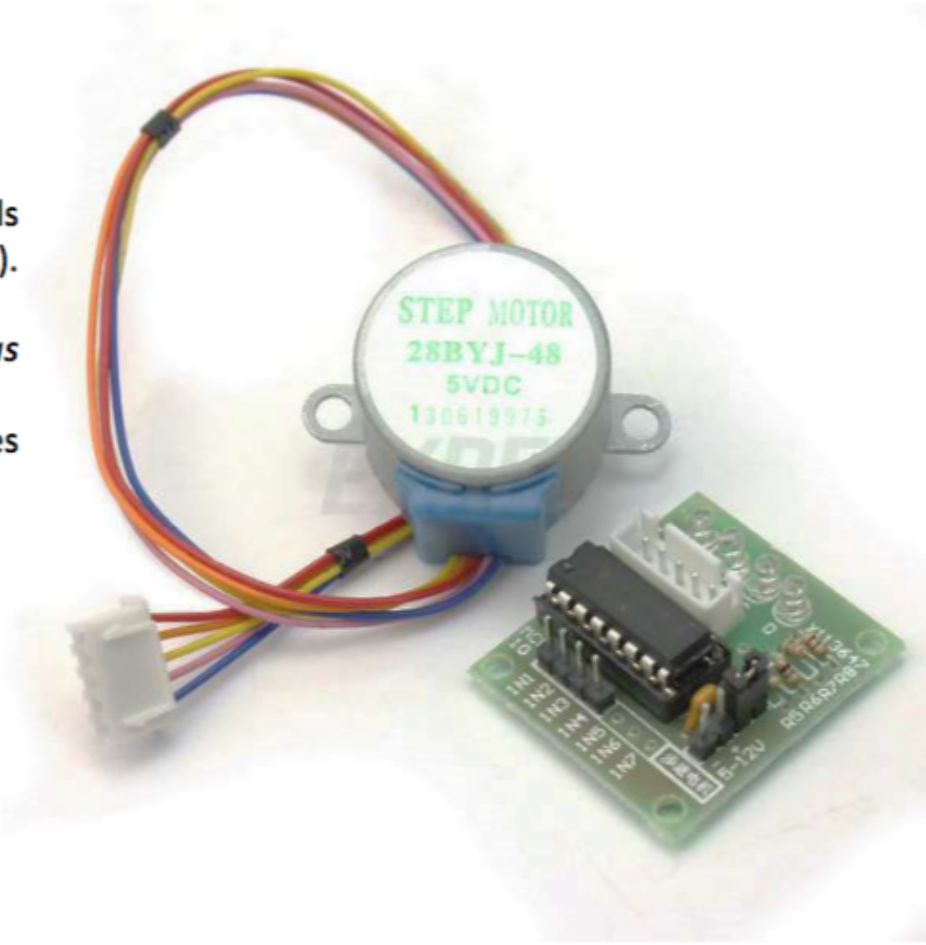


Stepper: 28BYJ-48

28-BYJ48 is an **Unipolar Stepper Motor**
The unipolar stepper motor has five or six wires and four coils (actually two coils divided by center connections on each coil).

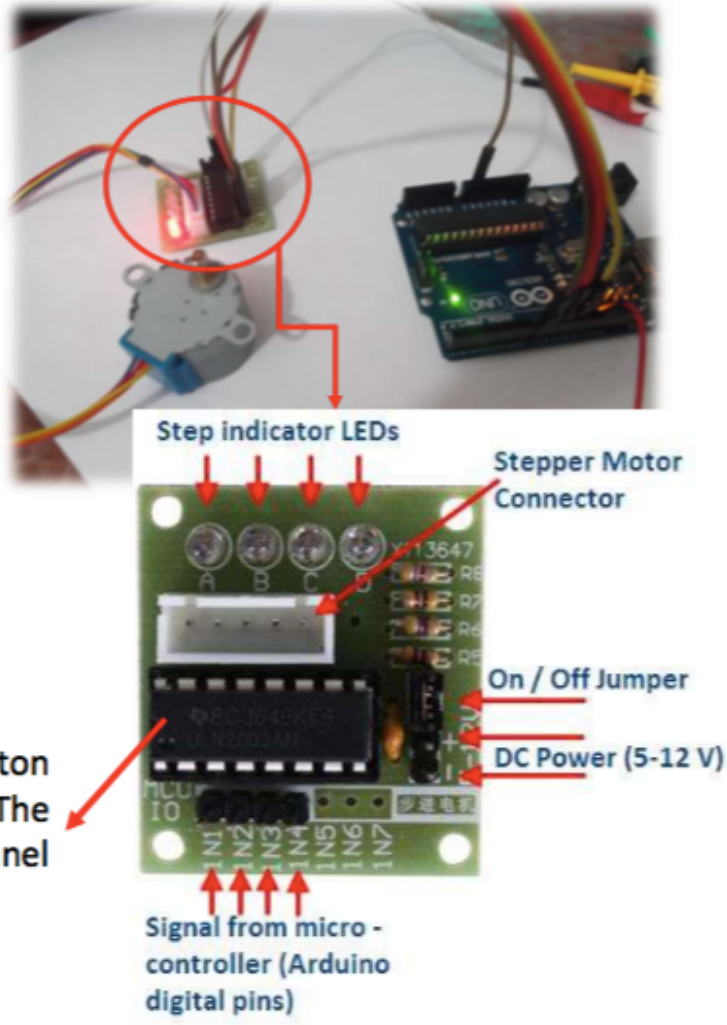
The center connections of the coils are tied together and used as the power connection.
They are called unipolar steppers because power always comes in on this one pole.





Stepper - Driver

A stepper motor driver is a **circuit which is used to drive a stepper motor**.
Driver IC's (i.e. chip) are available at reasonable costs and are easier to implement in terms of assembling. The *drivers must be selected to suit the motor ratings in terms of current and voltages*.
A stepper motor may run at voltages varying from 5 V to 12 V and similarly the current draw will be somewhere in the range of 100 mA to 400 mA.



The ULN2003A contains seven darlington transistor drivers all in one package. The *ULN2003A can pass up to 500 mA* per channel and has an internal voltage drop of about 1V.

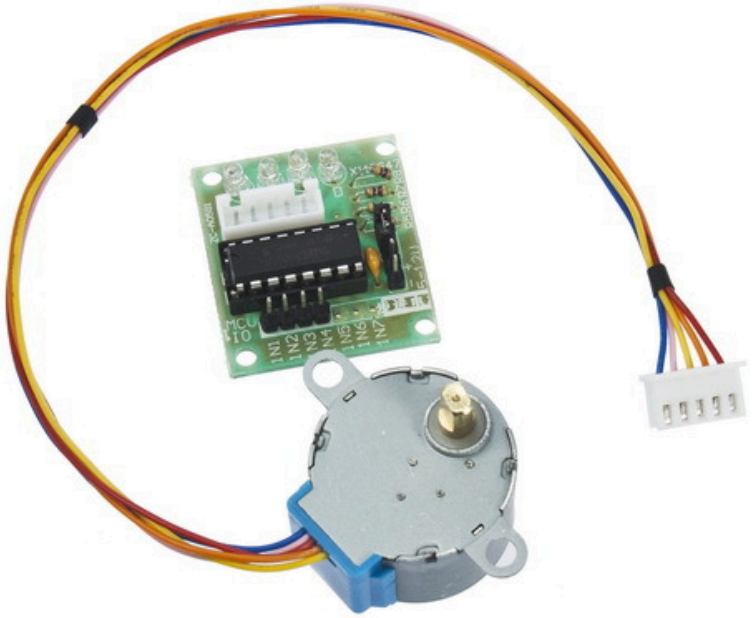
Stepper vs servo motor

- Both the stepper motor and servo motor are used primarily in position control applications, but there lies a difference in their working and construction.
  - In the stepper motor, the position is controlled thanks to the magnetic attraction of the rotor with the magnetized coil of the stator. In a servo motor the position is controlled by the specialized circuit and the feedback mechanism, which generates an error signal to move the motor shaft.
  - Servos are usually limited to a 0-180 degree range, while a stepper motor can rotate continuously.

Fourth Program

Step Motor 5V DC

Stepper motor 28YBJ-48 and related driver



In figure the stepper motor 28YBJ 48, with reducer. Features:

- 4-phase motor
- 5-12 volt power supply, from a source that can be external to Arduino
- consumption: 320 mA
- reduction ratio 1/64

Library: https://www.arduino.cc/en/reference/stepper

Datasheet: http://robocraft.ru/files/datasheet/28BYJ-48.pdf

Step Motor Sketch

```
/* Stepper Motor Control - speed control
  This program drives a unipolar or bipolar stepper motor.
  The motor is attached to digital pins 8 - 11 of the Arduino.
  A potentiometer is connected to analog input 0.
  The motor will rotate in a clockwise direction.
  The higher the potentiometer value, the faster the motor speed.
  Because setSpeed() sets the delay between steps, you may notice the motor
  is less responsive to changes in the sensor value at low speeds.*/

#include <Stepper.h>

const int stepsPerRevolution  = 4096;   // change this to fit the number of steps per revolution

Stepper myStepper(stepsPerRevolution, 8,10,11,9);      // initialize the stepper library on pins 8 through 11:
int stepCount = 0;  // number of steps the motor has taken

void setup() {}  // nothing to do inside the setup
void loop() {
  int sensorReading = analogRead(A0);   // read the sensor value:
  int motorSpeed = map(sensorReading, 0, 1023, 0, 100);  // map it to a range from 0 to 100
  if (motorSpeed > 0)                                    // set the motor speed:
  { myStepper.setSpeed(motorSpeed);
    myStepper.step(stepsPerRevolution/100);     // step 1/100 of a revolution:
  }
}
```