Electronic Prototyping

Analog to digital converter and actuators

Lesson 3

Analog to digital converter

Example of A/D Applications

- Microphones
  - Take your voice pressure in the air and convert them into an electrical signal
- Strain gauges
  - Determine the displacement of an object (change in dimensions) when a stress is applied
- Thermocouple
  - Converts temperature difference into an electrical potential

What is an ADC?

An ADC (Analog-to-Digital-Converter) is a circuit which gets an analog voltage signal and provides (to software) a variable proportional to the input signal.

An ADC is characterised by:

- The **range** (in volts) of the input signal (typical [0,5V] or [0, 3.3V]).
- The **resolution** (in **bits**) of the converter

Example

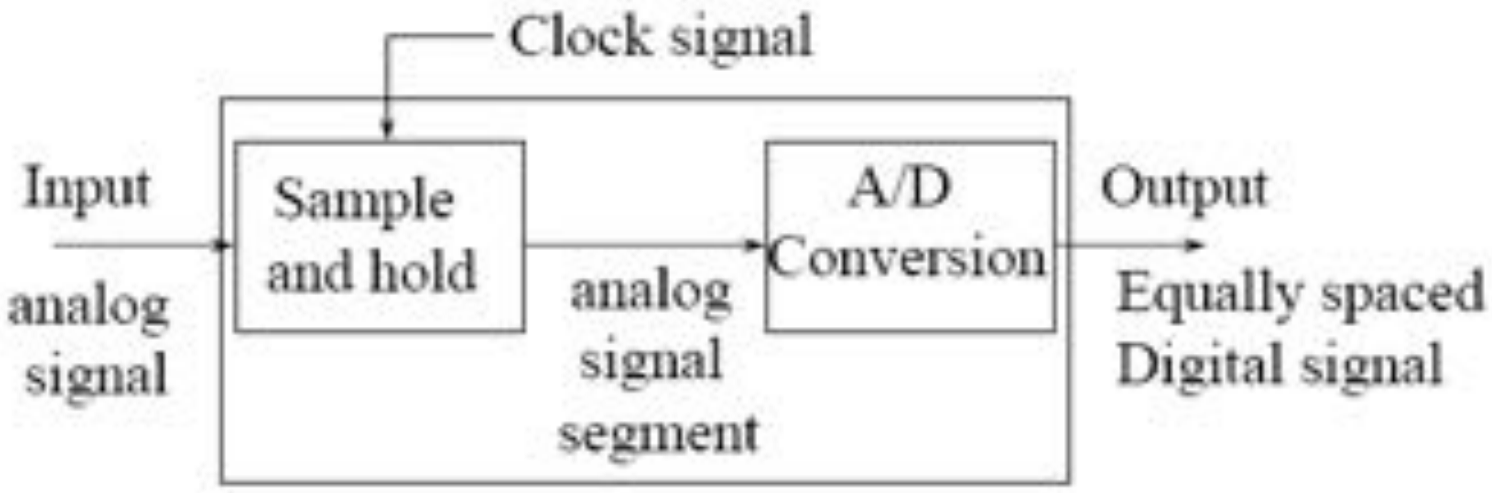
- **Range** = [0,5V]
- **Resolution** = 12 bits

  Results in range [0, 2^{12}- 1] = [0, 4095]

  0V → 0 2.5V → 2048 5V → 4095

Just what does an A/D converter do?

- Converts analog signal into binary words



Analog → Digital Conversion 2 Step process:

- Quantizing
  - Breaking down analog value is a set of finite states
- Encoding
  - Assigning a digital word or number to each state and matching it to the input signal

Step 1: Quantizing (1/2)

Example

You have 0-10V signals. Separate them into a set of discrete states with 1.25V increments.

(How did we get 1.25V?

See next slide…)

Output States	Discrete Voltage Ranges (V)
0	0.00-1.25
1	1.25-2.50
2	2.50-3.75
3	3.75-5.00
4	5.00-6.25
5	6.25-7.50
6	7.50-8.75
7	8.75-10.0

Step 1: Quantizing (2/2)

The number of possible states that the converter can output is:

$$N=2^n$$

where n is the number of bits in the AD converter

Example: For a 3 bit A/D converter, $N=2^3=8$.

Analog quantization size:

$$Q=(V_{max}-V_{min})/N = (10V - 0V)/8 = 1.25V$$

Step 2 – Encoding

Here we assign the digital value (binary number) to each state for the computer to read

Output States	Output Binary Equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Accuracy of A/D Conversion

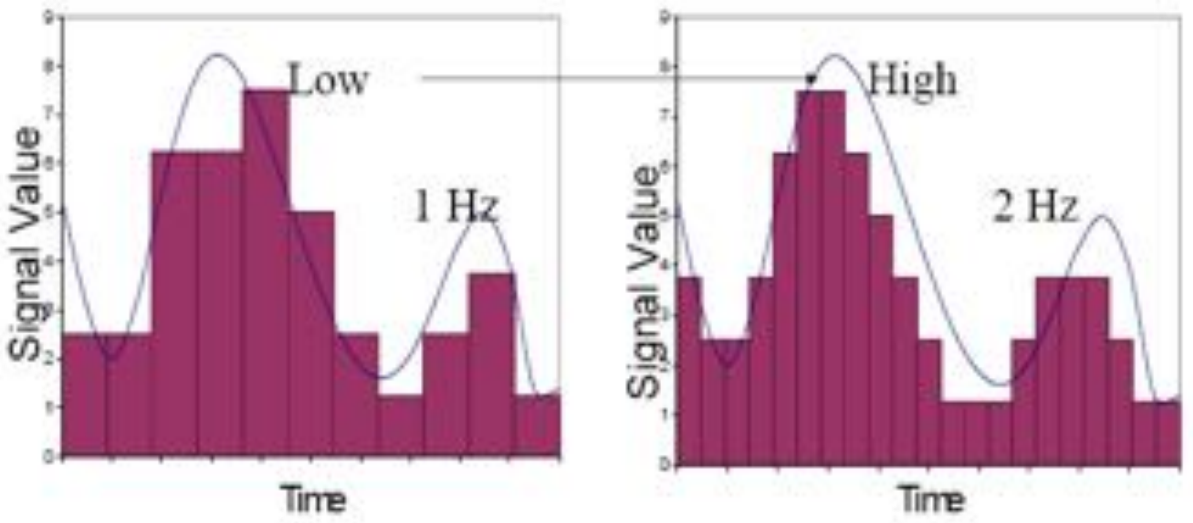
There are two ways to best improve accuracy of A/D conversion:

- increasing the resolution which improves the accuracy in measuring the amplitude of the analog signal.
- increasing the sampling rate which increases the maximum frequency that can be measured.

Resolution

- Resolution
  - (number of discrete values the converter can produce) = Analog Quantization size (Q)
  - (Q) = Vrange / 2^n, where Vrange is the range of analog voltages which can be represented
- limited by signal-to-noise ratio (should be around 6dB)
- In our previous example:
  - Q = 1.25V, this is a high resolution.
  - A lower resolution would be if we used a 2-bit converter, then the resolution would be $10/2^2 = 2.50$V.

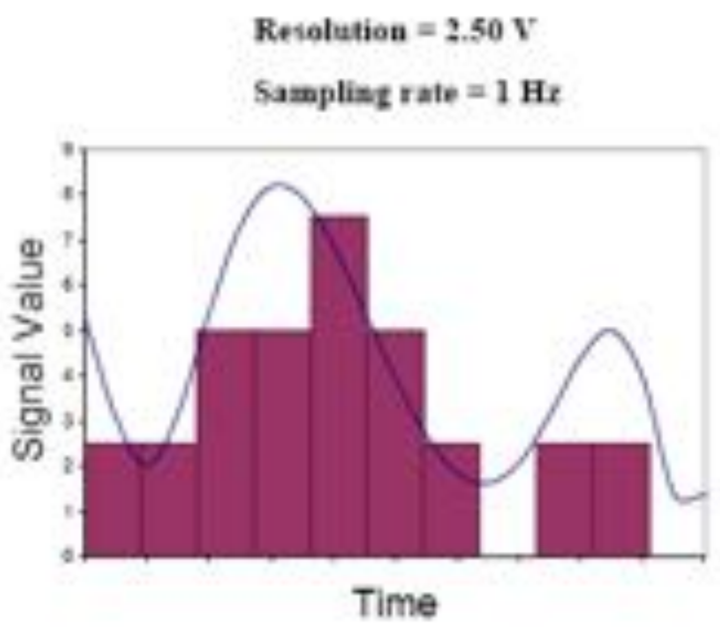
Sampling rate

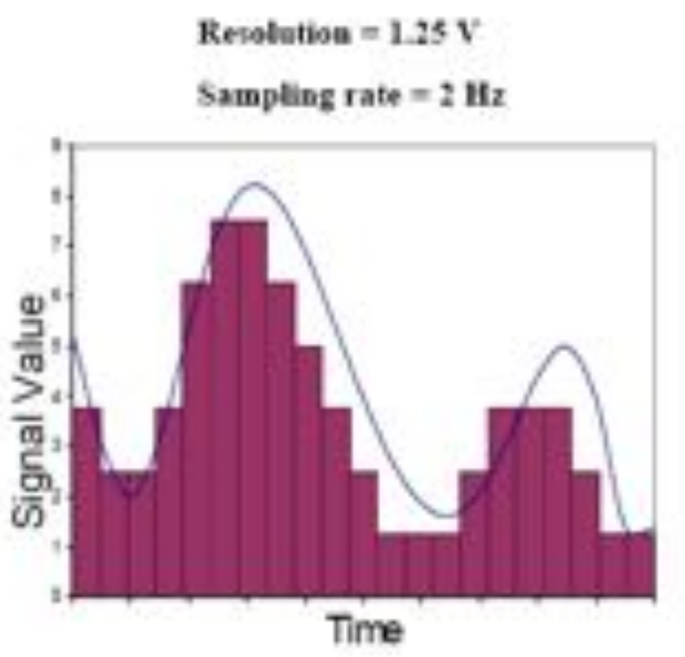


Frequency at which ADC evaluates analog signal. As we see in the second picture, evaluating the signal more often more accurately depicts the ADC signal.

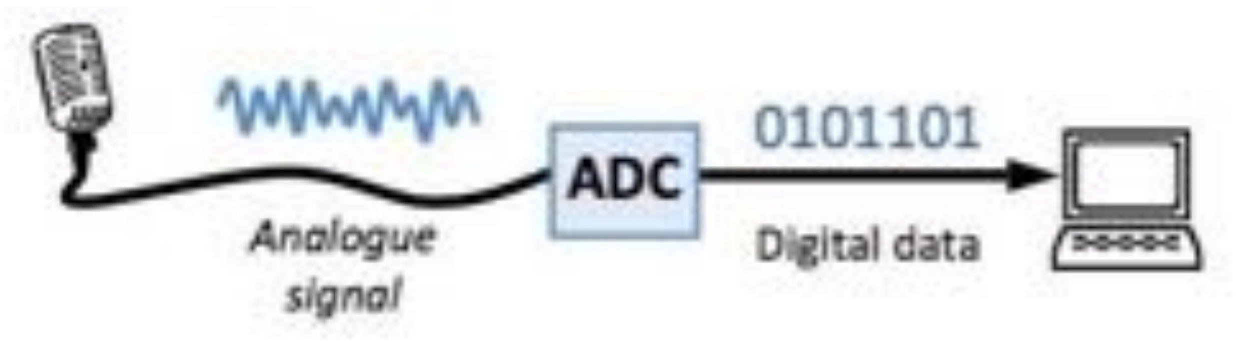
Overall Better Accuracy

Increasing both the sampling rate and the resolution you can obtain better accuracy in your AD signals.





Analog to digital converter



The Analog to Digital Conversion Process (1/2)

- Sounds are analog - they are made of waves that travel through matter. People hear sounds when these waves physically vibrate their eardrums.
- Since Computers are digital devices, they cannot understand these continuous pressure variable analog signals, so they communicate digitally, using electrical impulses that represent 0s and 1s (i.e., through Binary).

Binary Notations:

- One binary digit (0 or 1) is referred to as a bit, which is short for binary digit. One bit can only be used to represent 2 different values: 0 and 1.

The Analog to Digital Conversion Process (2/2)

- To represent more than two values, we need to use multiple bits.
- Two bits combined can be used to represent 4 different values:
  - 0 0, 0 1, 1 0, and 1 1.
- Three bits can be used to represent 8 different values:
  - 0 0 0, 0 0 1, 0 1 0, 1 0 0, 0 1 1, 1 0 1, 1 1 0 & 1 1 1.
- In general, 'n' bits can be used to represent 2^n different values.

2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096

Arduino Due ADC

- The **Due** has the following hardware capabilities:
    - Resolution is defaults to 10 bits (returns values between 0-1023)
        - 2 pins with 12-bit DAC (Digital-to-Analog Converter)
    - The Due boards have 12-bit ADC capabilities that can be accessed by changing the resolution to 12.
        - This will return values from analogRead() between 0 and 4095.

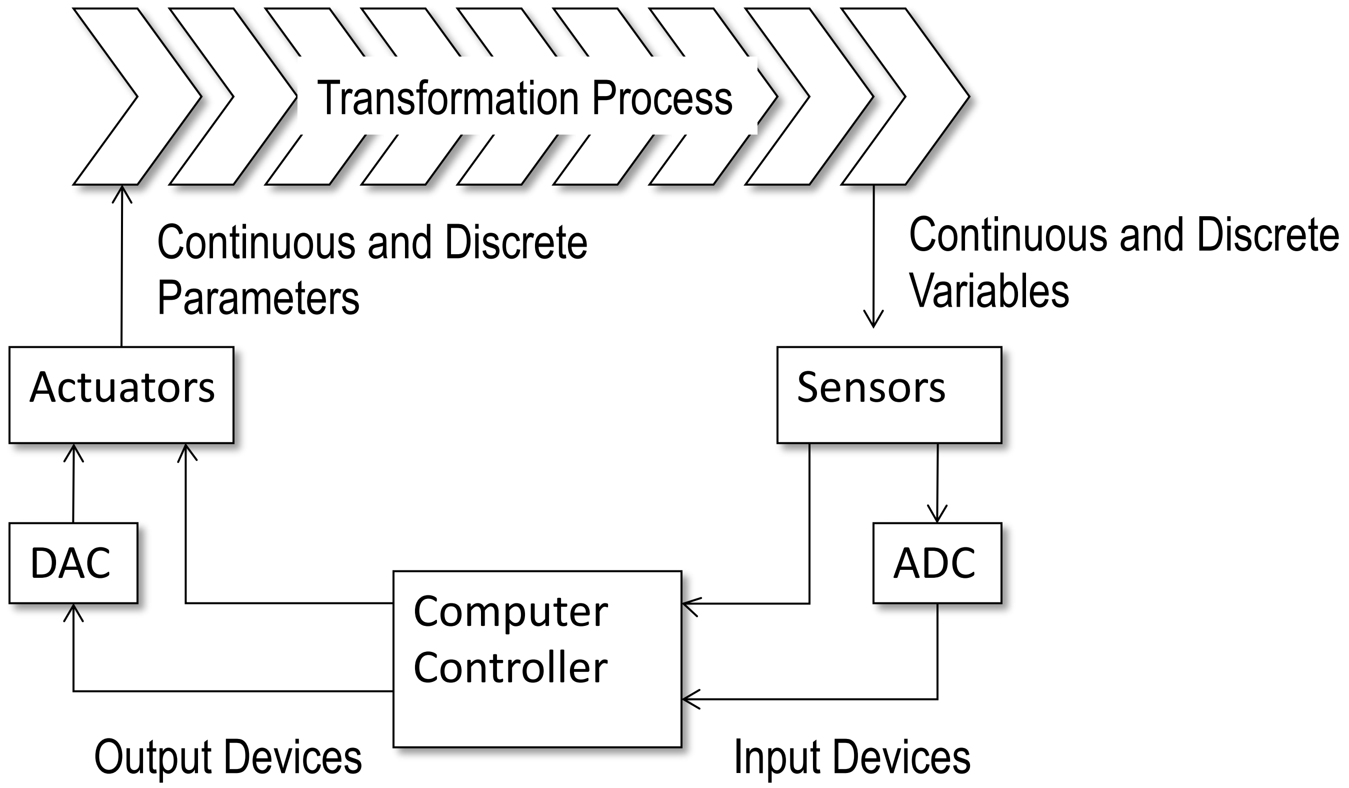
Example 1: Potentiometer

- Simulation of analog sensor using a potentiometer and conversion in millivolts

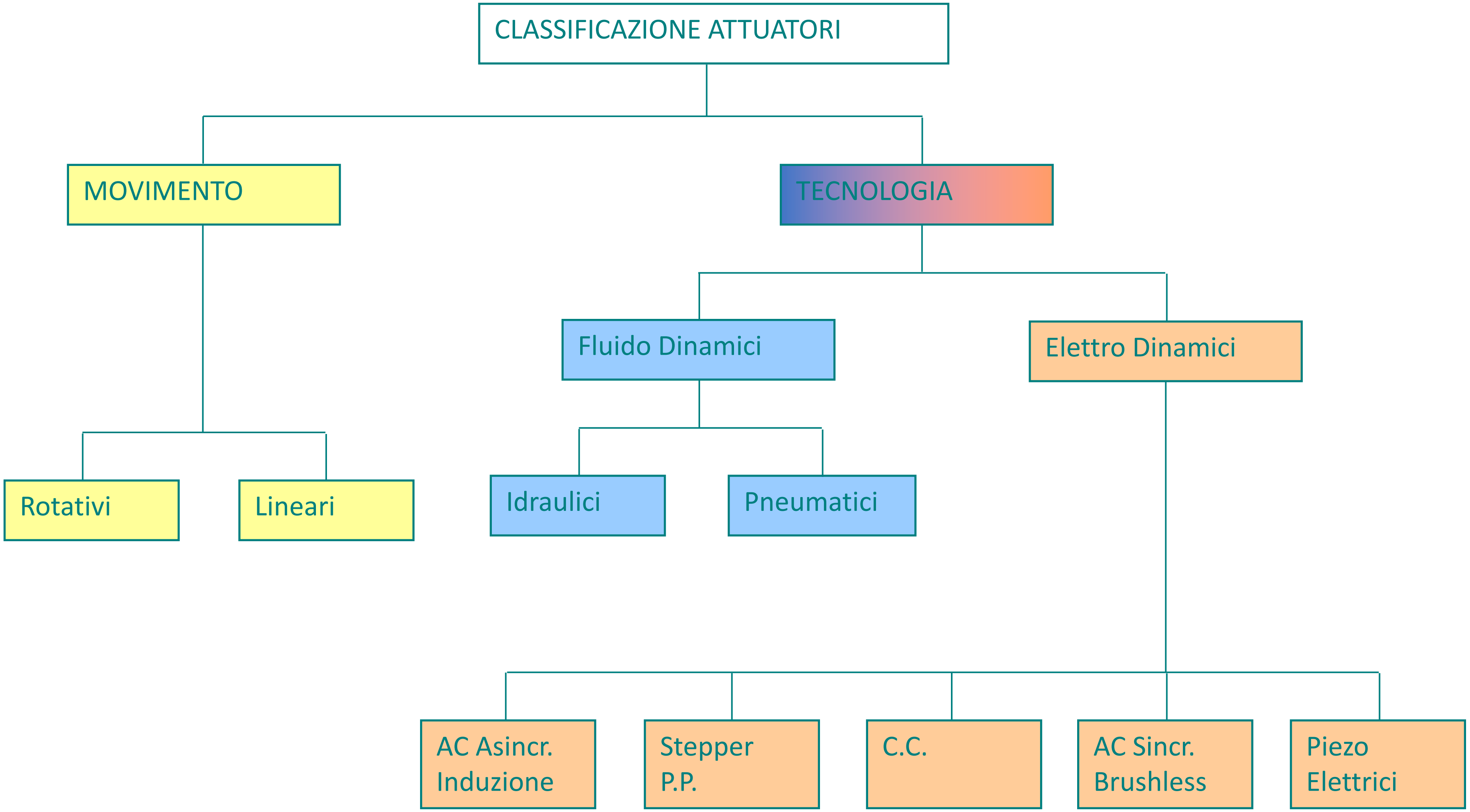
Open Example 1 in the shared folder

Actuators and movements (Mechatronics)

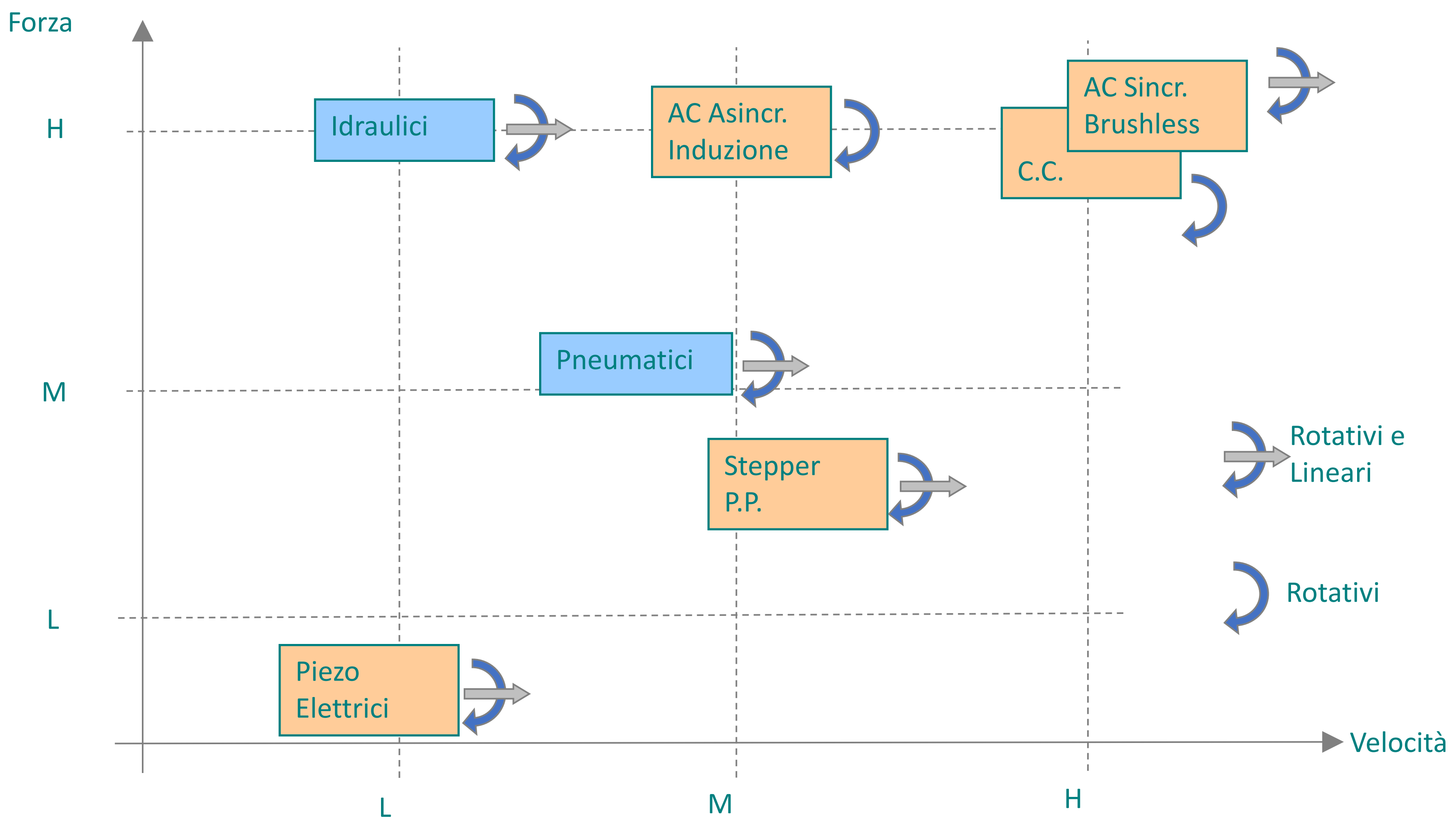
Basic of mechatronics



Actuators

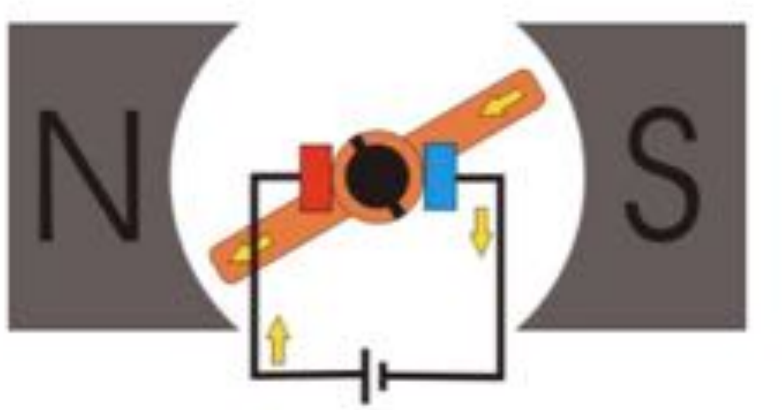

CLASSIFICAZIONE ATTUATORI
MOVIMENTO
TECNOLOGIA
Rotativi
Lineari
Fluido Dinamici
Elettro Dinamici
Idraulici
Pneumatici
AC Asincr. Induzione
Stepper P.P.
C.C.
AC Sincr. Brushless
Piezo Elettrici

Actuators map – Force vs velocity



Electrical Motors

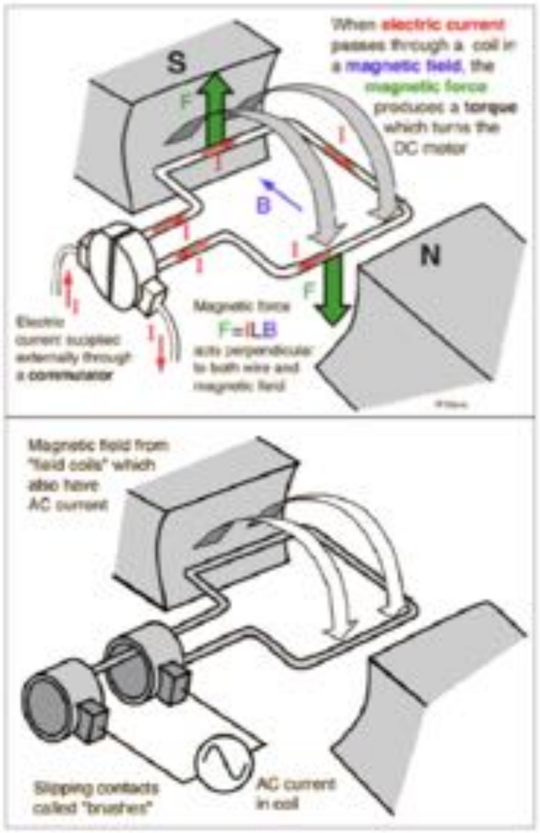
- A motor is an electro-mechanical device that converts electrical energy to mechanical energy.
- The very basic principal of functioning of an electrical motor lies on the fact that force is experienced in the direction perpendicular to magnetic field and the current, when field and current are made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



Types of motors

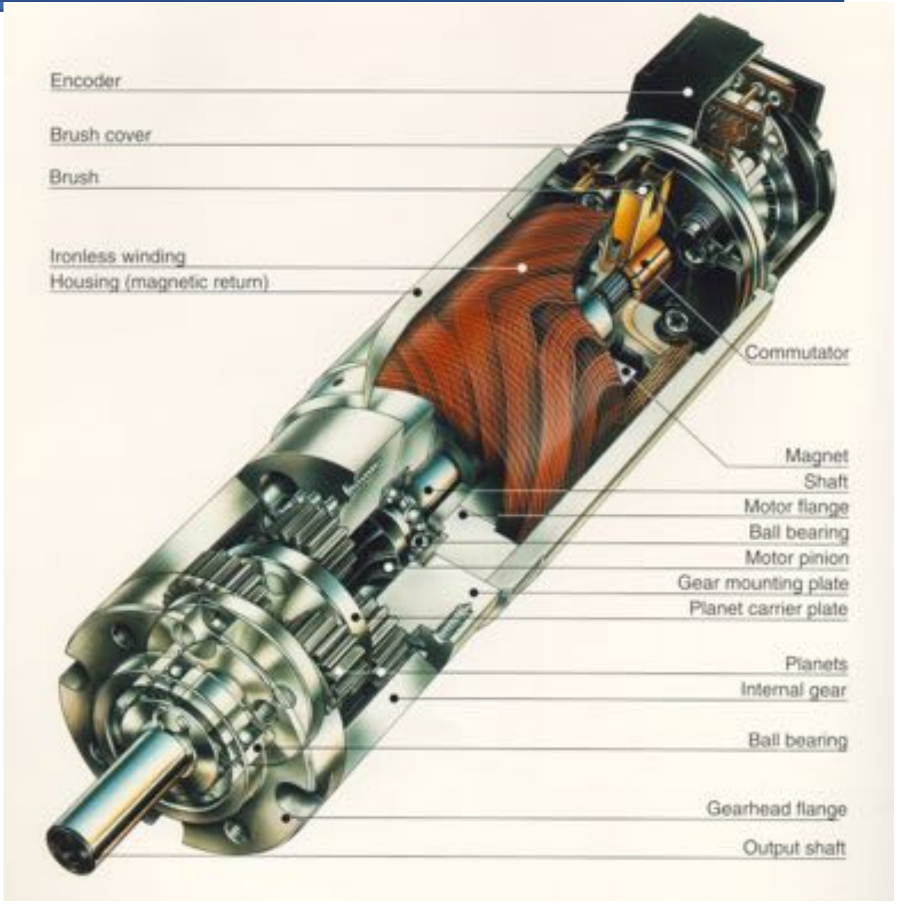
The DC motor as the name suggests, is the only one that is driven by **direct current.**
It's the most primitive version of the electric motor where **rotating torque is produced due to flow of current through the conductor inside a magnetic field.**

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).


When electric current passes through a coil in a magnetic field, the magnetic force produces a torque which turns the DC motor
S
N
F
I
B
Electric current supplied externally through a commutator
Magnetic force
F=ILB
acts perpendicular to both wire and magnetic field
Magnetic field from "field coils" which also have AC current
Slipping contacts called "brushes"
AC current in coil

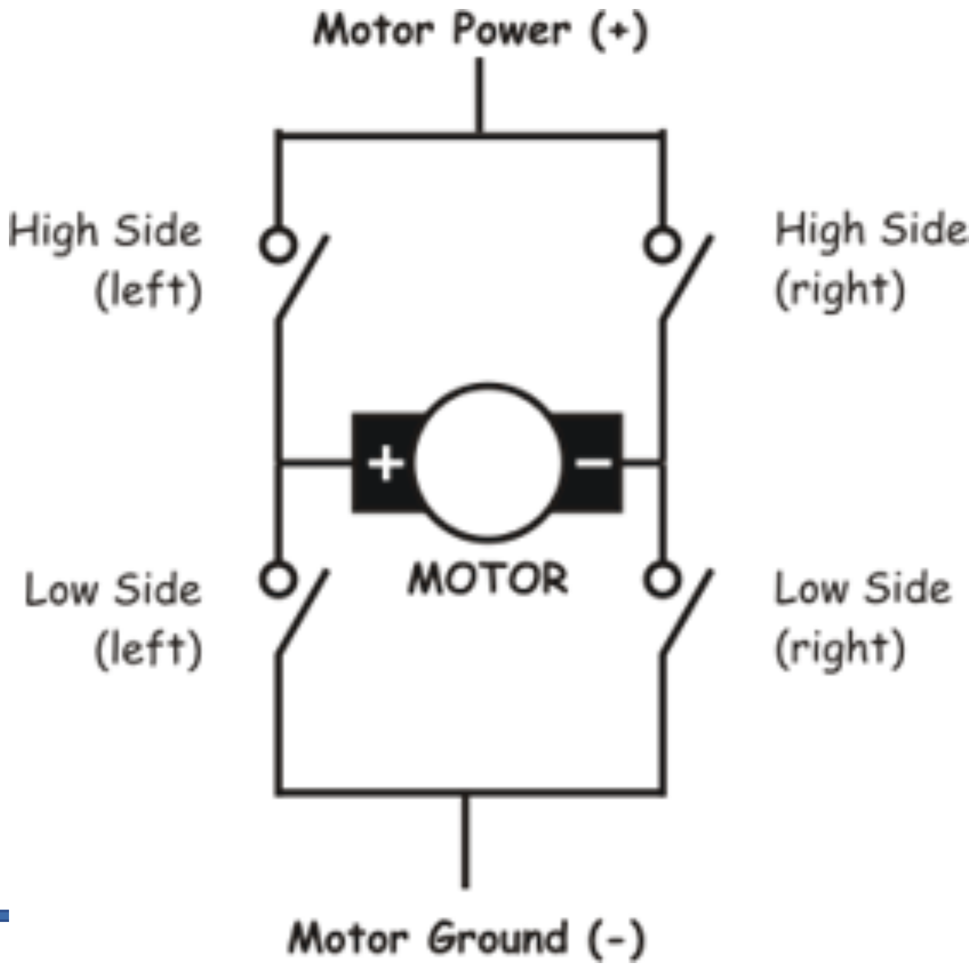
DC motor

- simple, cheap
- 1W - 1kW
- can be overloaded
- brushes wear
- limited overloading on high speeds



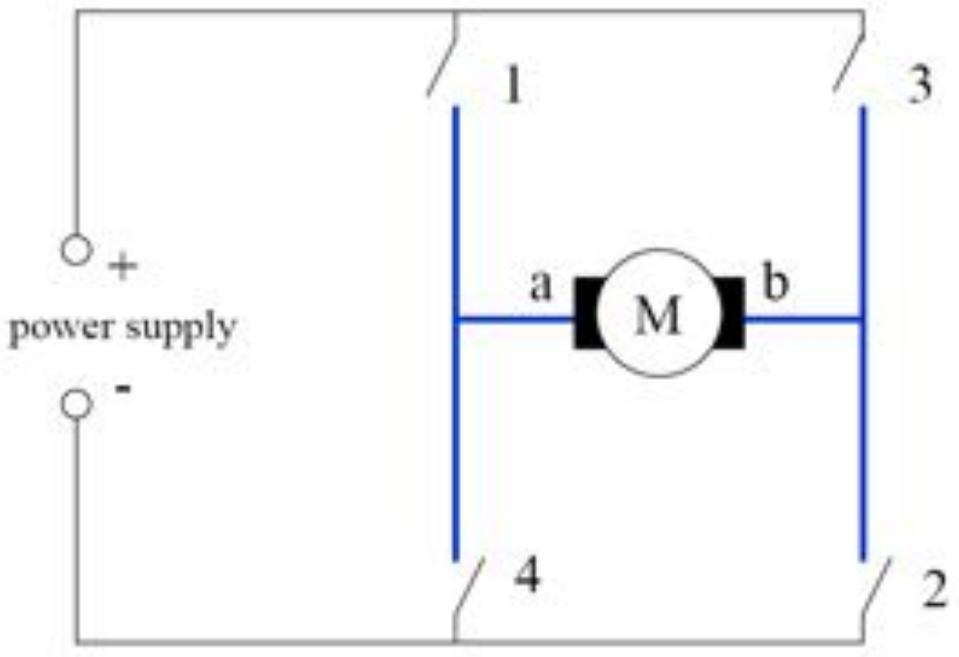
DC motor

- Controller + H-bridge
- PWM-control
- Speed control by controlling motor current=torque
- Efficient small components
- PID control



H - bridge

Allows a motor to be driven in both directions



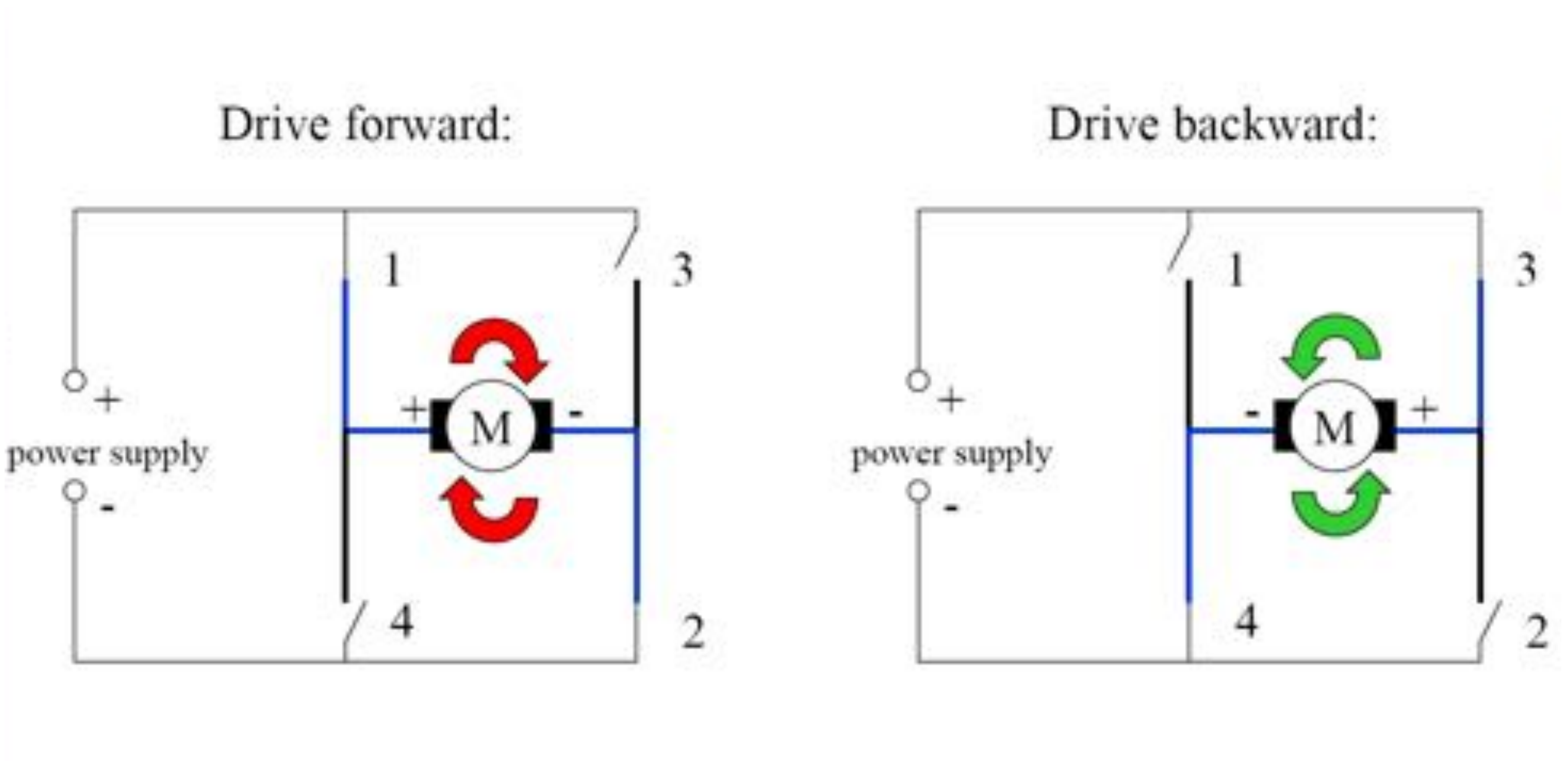
Drive forward:

- Close 1 and 2

Drive backward:

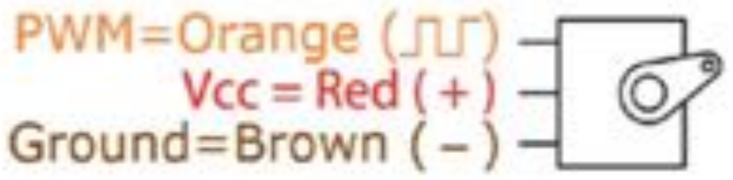
- Close 3 and 4

H - bridge

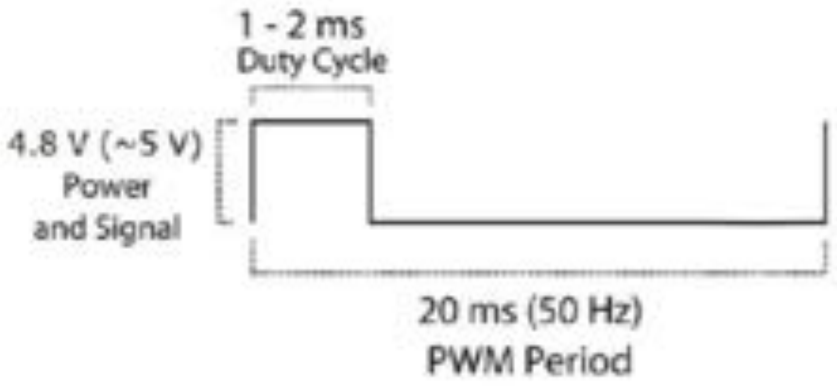


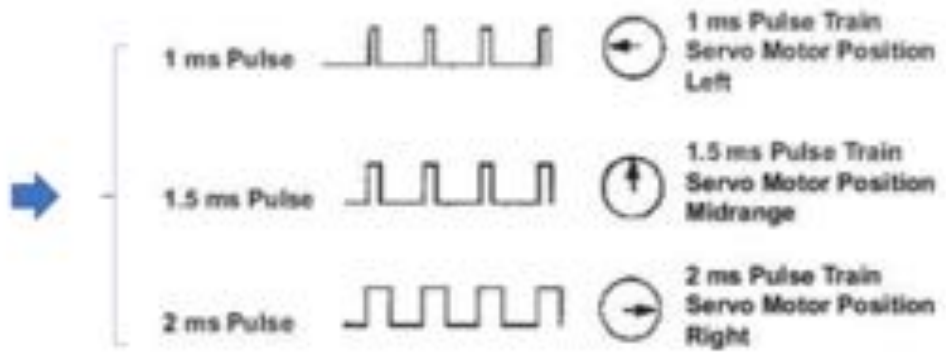
SERVO Motor SG90

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.





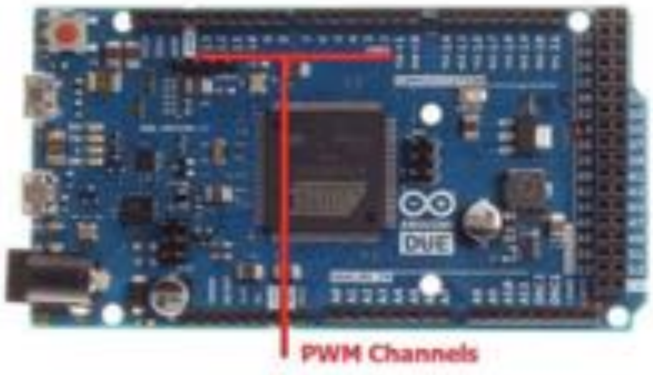
PWM with Arduino Due

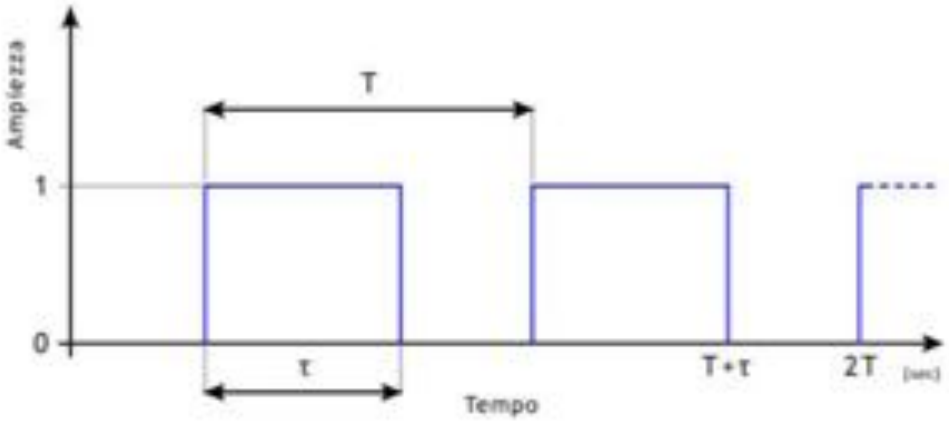
There are **12 PWM Channels** (Pin 2 to Pin 13)

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.





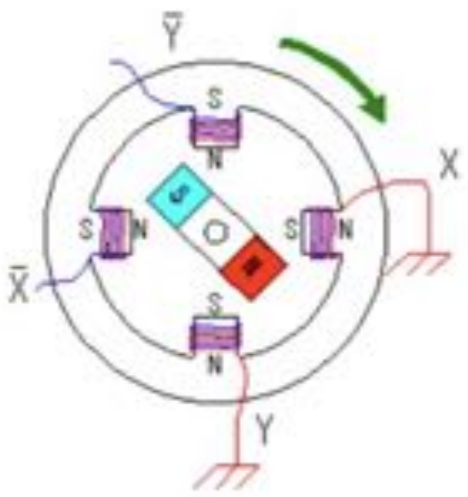
Duty cicle: $d = \frac{\tau}{T}$

Stepper motor

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

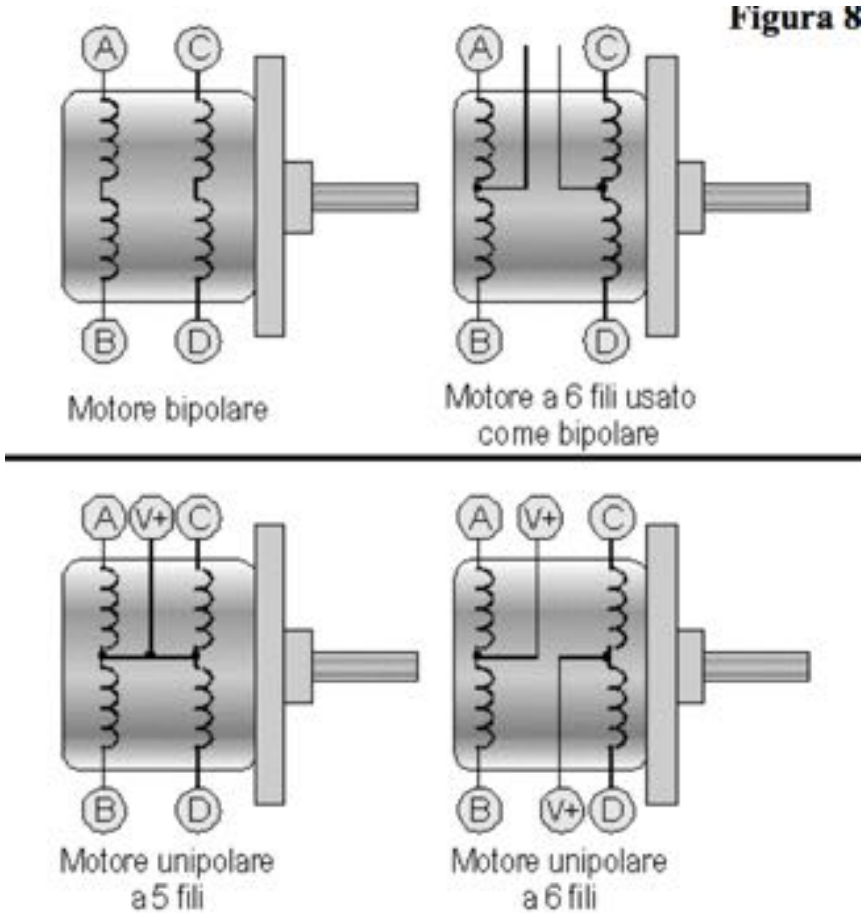
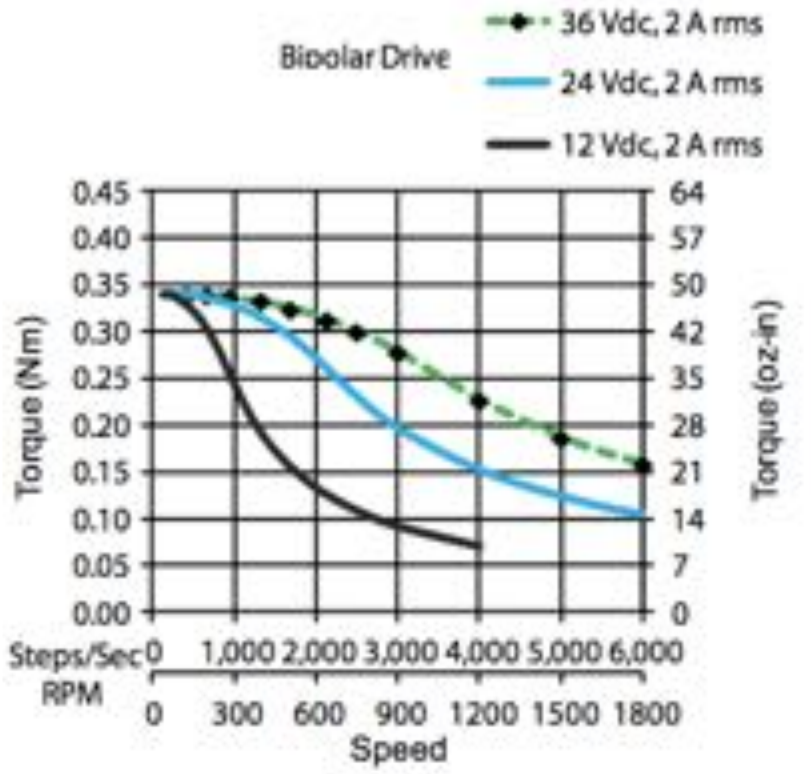


Figura 8
A
C
B
D
Motore bipolare
A
C
B
D
Motore a 6 fili usato come bipolare
A
V+
C
B
D
Motore unipolare a 5 fili
A
V+
C
B
V+
D
Motore unipolare a 6 fili

Stepper motor

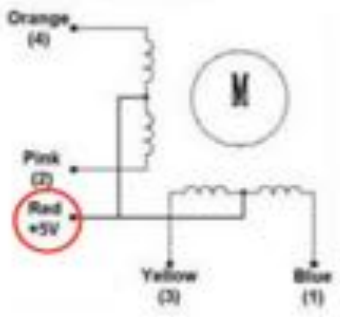


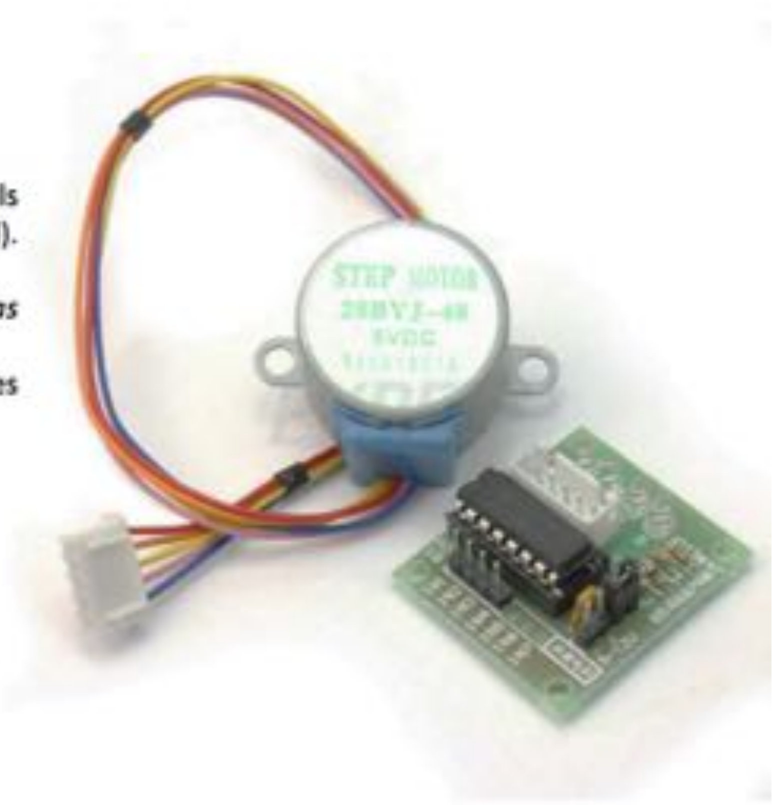
Stepper motor

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.

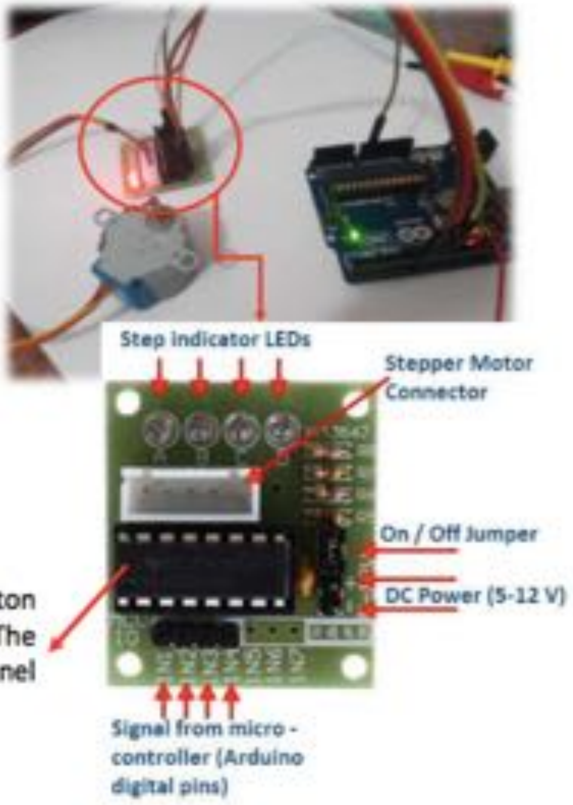




Stepper motor

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.



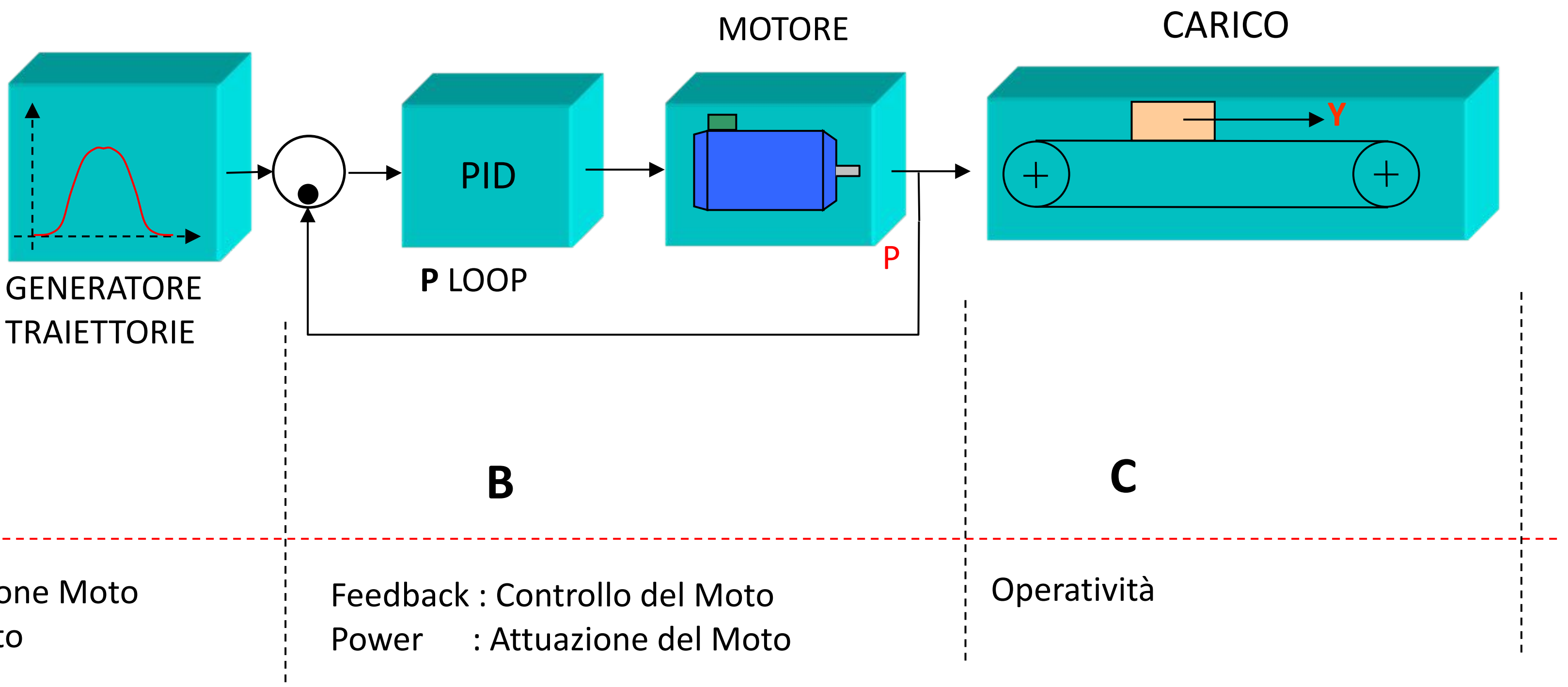
The ULN2003A contains seven darlington transistor drivers all in one package. The *ULN2003A can pass up to 500 mA* per channel and has an internal voltage drop of about 1V.

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.

How to choose the right actuator(s)

System approach



A

Generazione Moto Desiderato

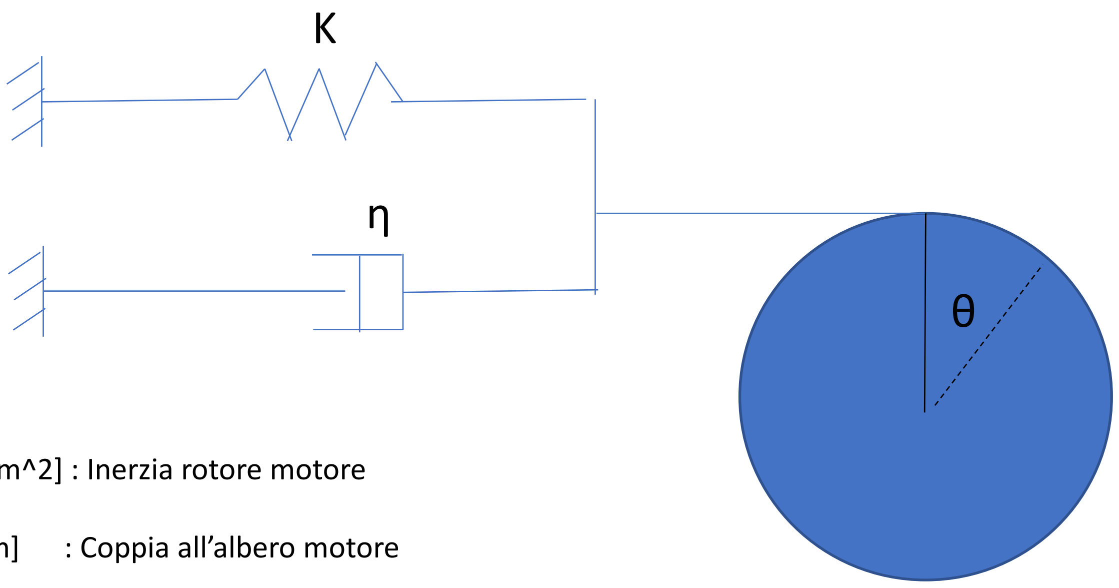
B

Feedback : Controllo del Moto
Power : Attuazione del Moto

C

Operatività

A simple system



Jm	[Kgm^2]	: Inerzia rotore motore
Fm	[Nm]	: Coppia all'albero motore
θ	[rad]	: Posizione angolare albero motore
$\dot{\theta}$	[rad/s]	: Velocità angolare albero motore
$\ddot{\theta}$	[rad/s^2]	: Accel. angolare albero motore

Per dimensionare correttamente il motore servono tre indicatori ricavabili dalla analisi dinamica precedente :

- Coppia efficace o rms (root mean square) *Fm rms* sul periodo *T*[s] del ciclo di lavoro

$$Fm\ rms = \sqrt{\frac{1}{T} Fm(t)^2}$$

- Coppia di picco *Fm pk*

$$Fm\ pk = \left|\max(Fm(t))\right|$$

- Velocità di picco *Vel pk*

$$Vel\ pk = \left|\max(\dot{\theta}(t))\right|$$

Criterio Dimensionamento

Fm rms < M nom

Fm pk < M max

Vel pk < Vel max

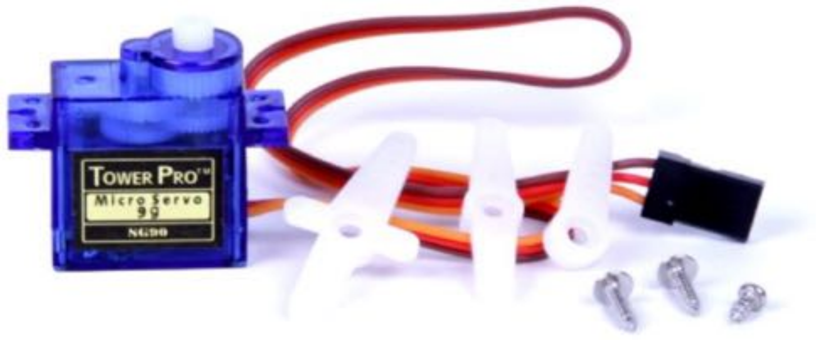
i quali vanno rispettivamente confrontati con i tre parametri caratteristici del motore riportati sul data sheet motore

- M nom : Coppia Nominale
- M max : Coppia massima
- Vel max : Velocità massima

Exercise 2: positioning of Servo motor using a potentiometer

Components

Use analog read values from potentiometer (0 to 1023) to position servo motor from 0 to 180 degrees



#include <Servo.h>

FIND DATASHEET

Exercise 3: change stepper speed using a potentiometer

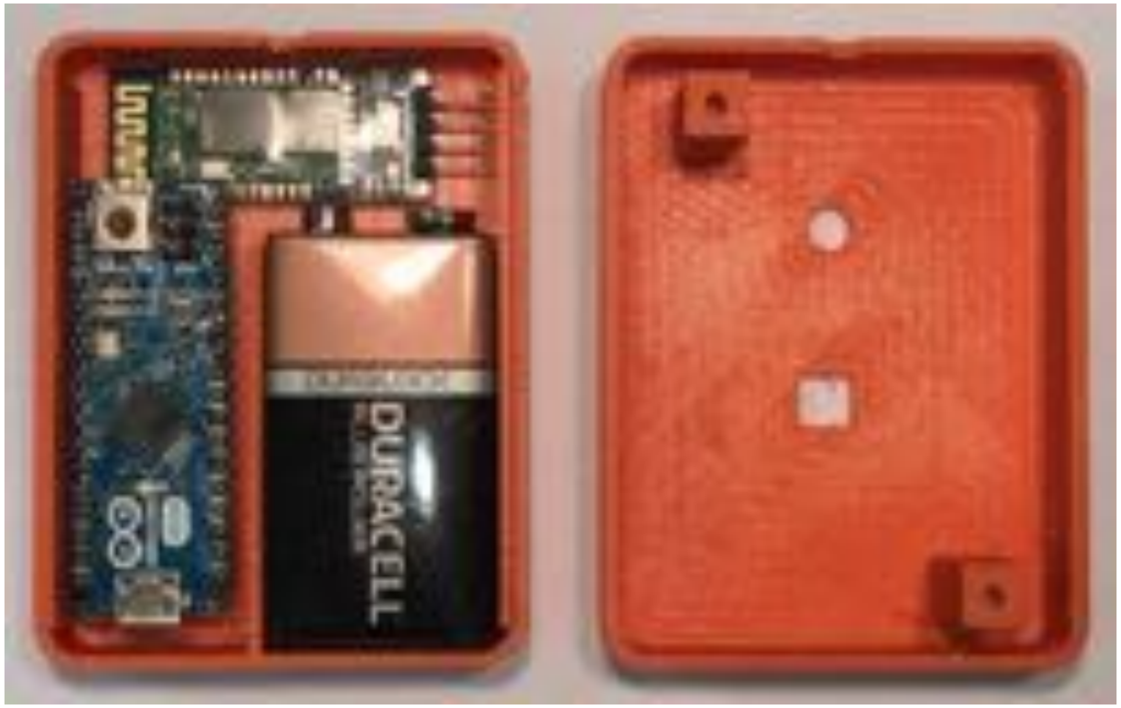
Components

#include <Stepper.h>

FIND DATASHEET

Batteries dimensioning

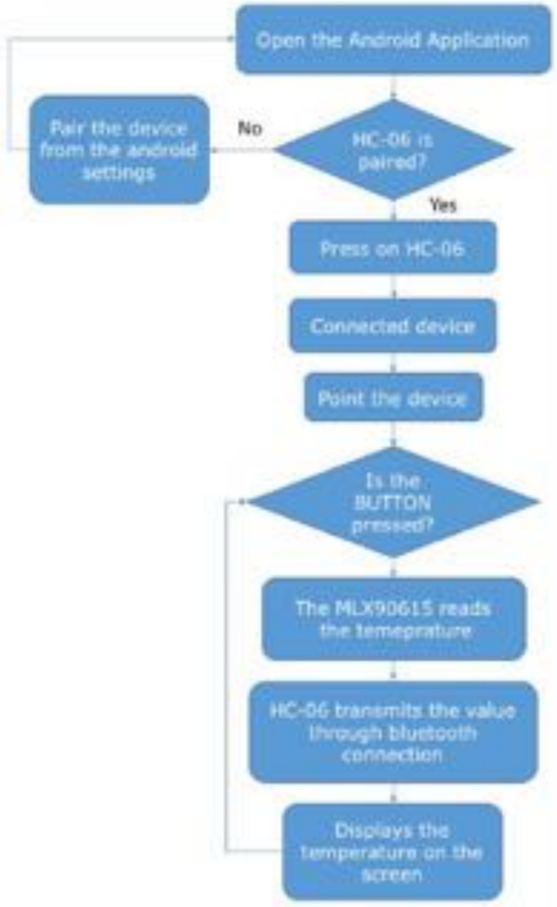
Example: contactless thermometer

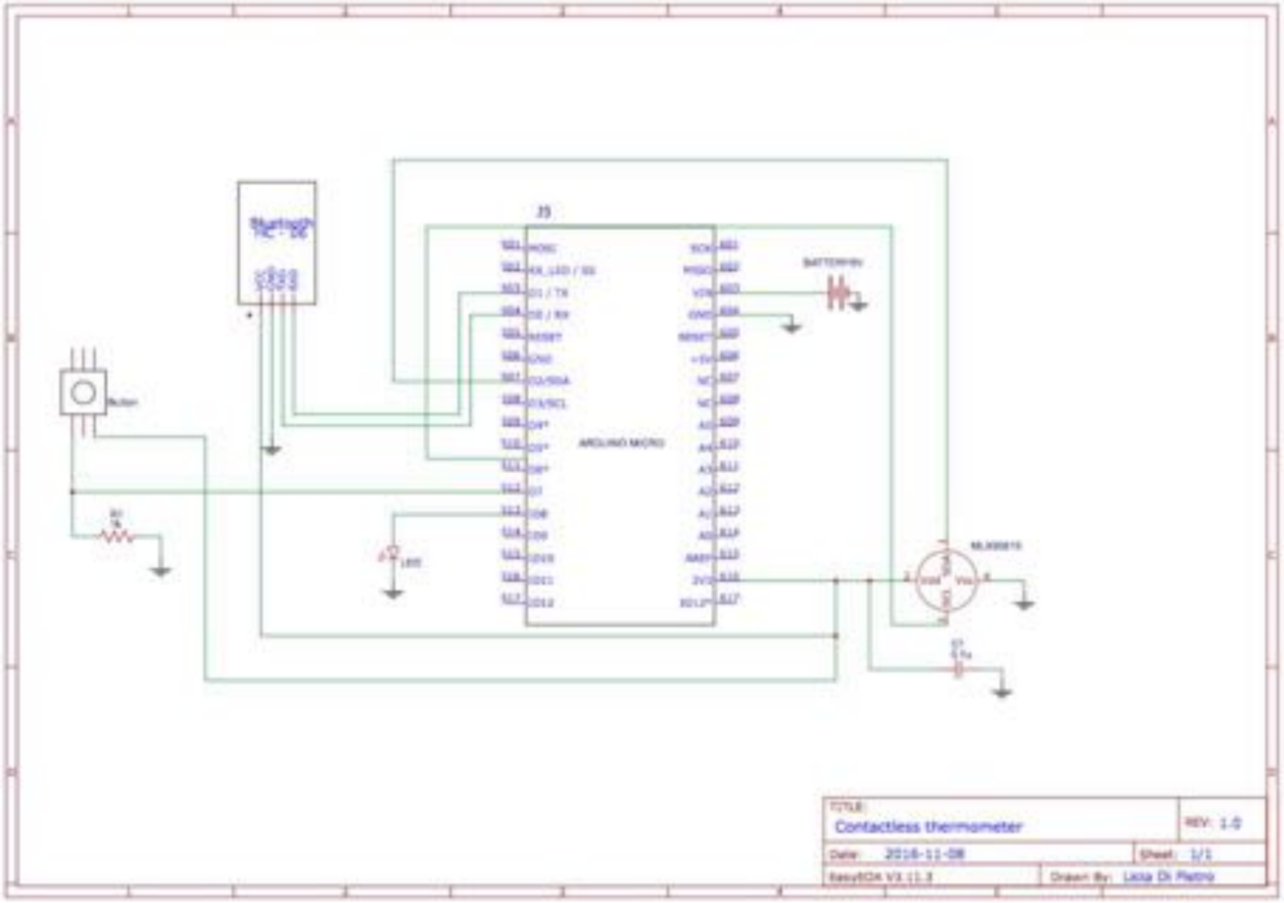


MLX90615 by Melexis

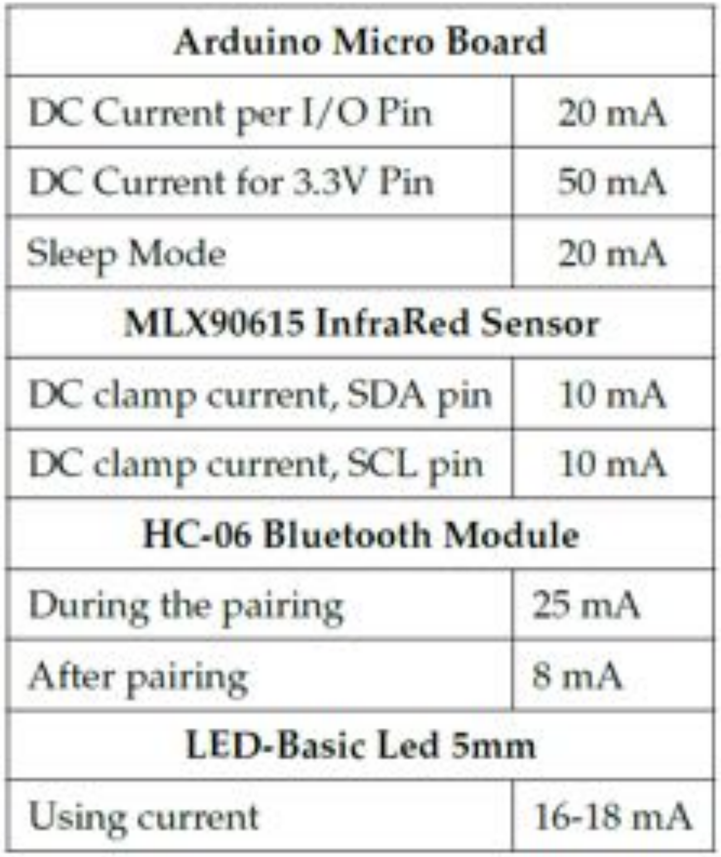
Battery capacity 550 mAh @9 V

Contactless thermometer


Open the Android Application
HC-06 is paired?
No
Pair the device from the android settings
Yes
Press on HC-06
Connected device
Point the device
Is the BUTTON pressed?
The MLX90615 reads the temeprature
HC-06 transmits the value through bluetooth connection
Displays the temperature on the screen

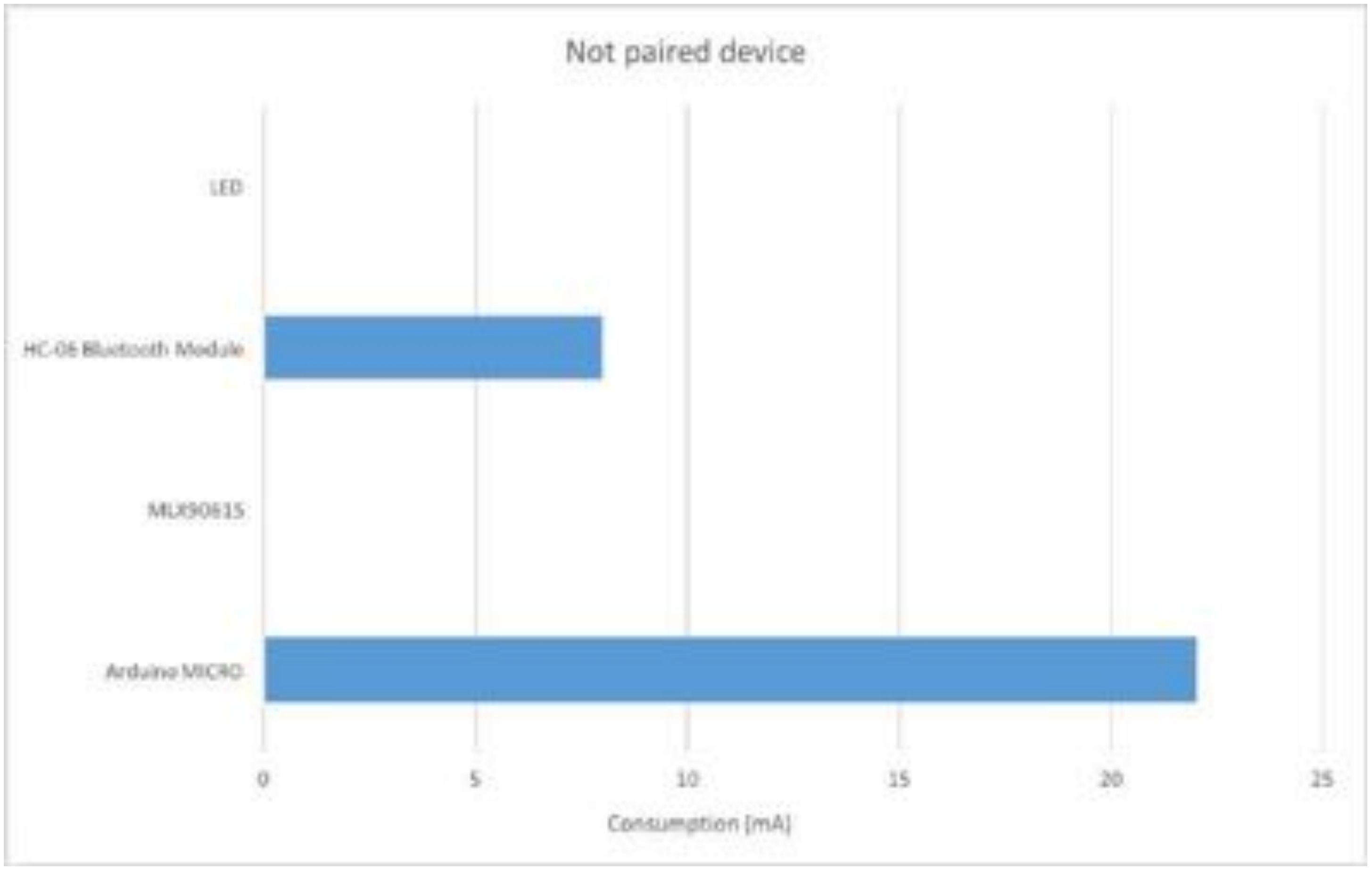


Current consumption

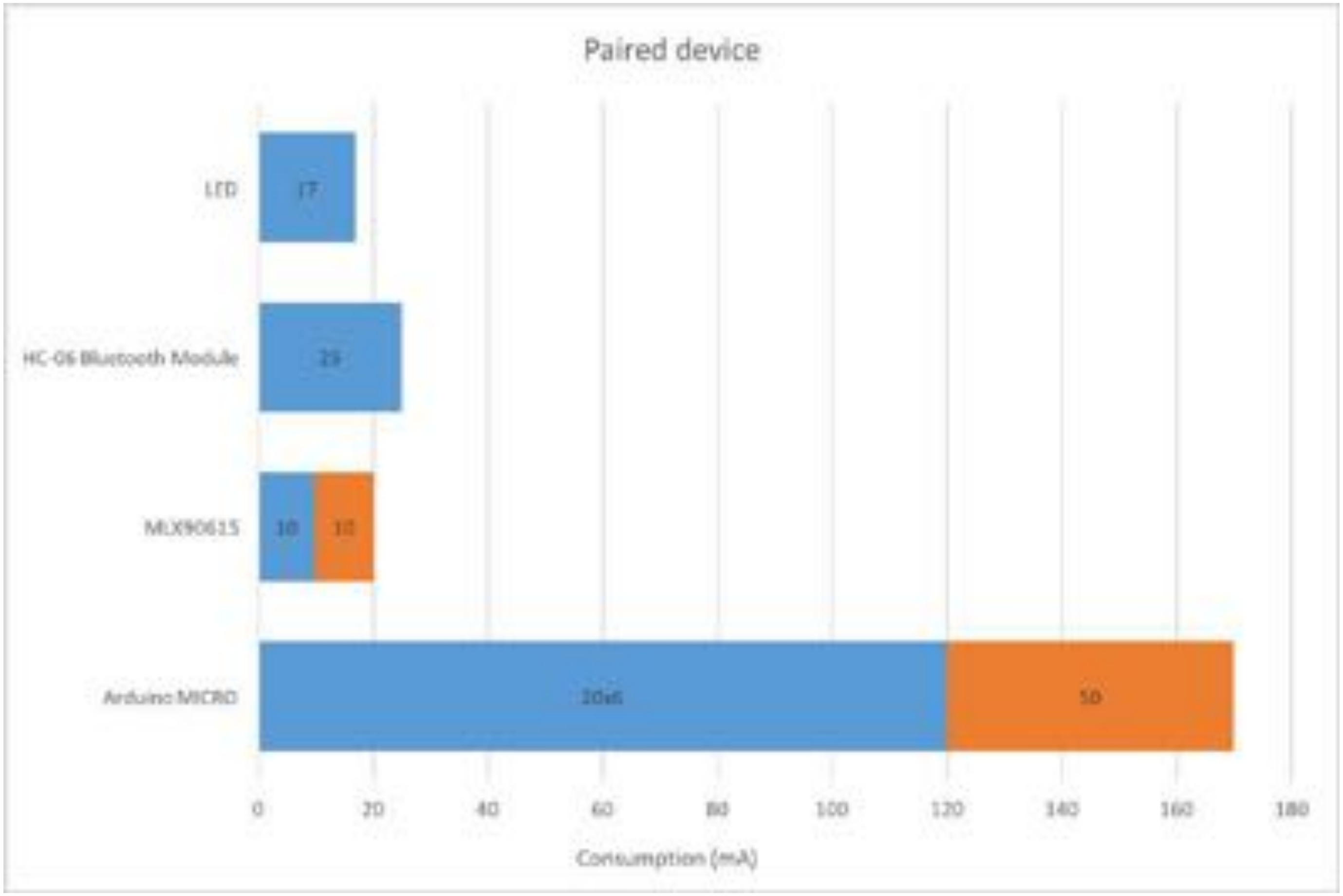


Arduino Micro Board	
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Sleep Mode	20 mA
MLX90615 InfraRed Sensor	
DC clamp current, SDA pin	10 mA
DC clamp current, SCL pin	10 mA
HC-06 Bluetooth Module	
During the pairing	25 mA
After pairing	8 mA
LED-Basic Led 5mm	
Using current	16-18 mA

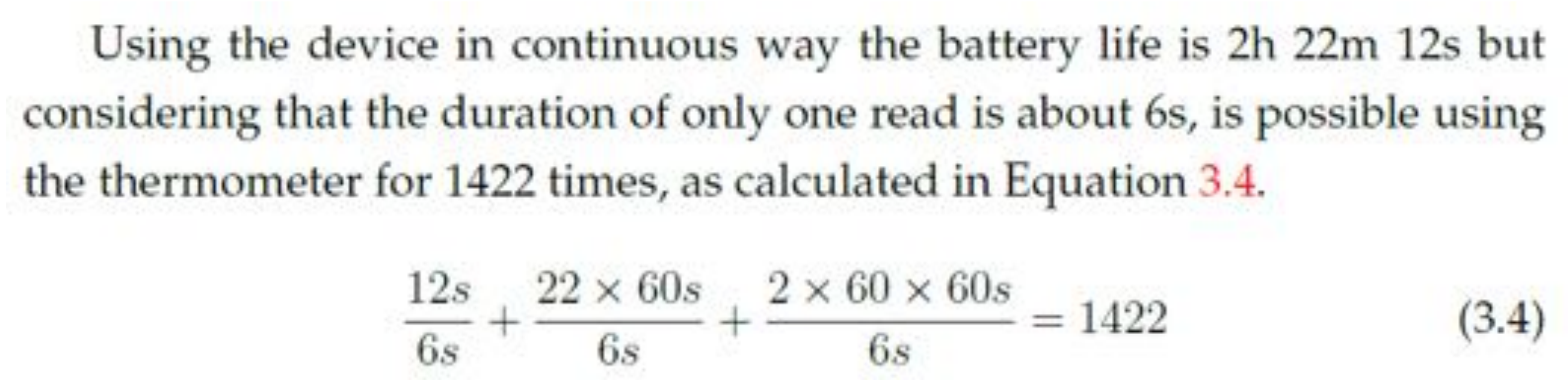
Current consumption



Current consumption



Current consumption



Using the device in continuous way the battery life is 2h 22m 12s but considering that the duration of only one read is about 6s, is possible using the thermometer for 1422 times, as calculated in Equation 3.4.

$$\frac{12s}{6s} + \frac{22 \times 60s}{6s} + \frac{2 \times 60 \times 60s}{6s} = 1422 \qquad (3.4)$$