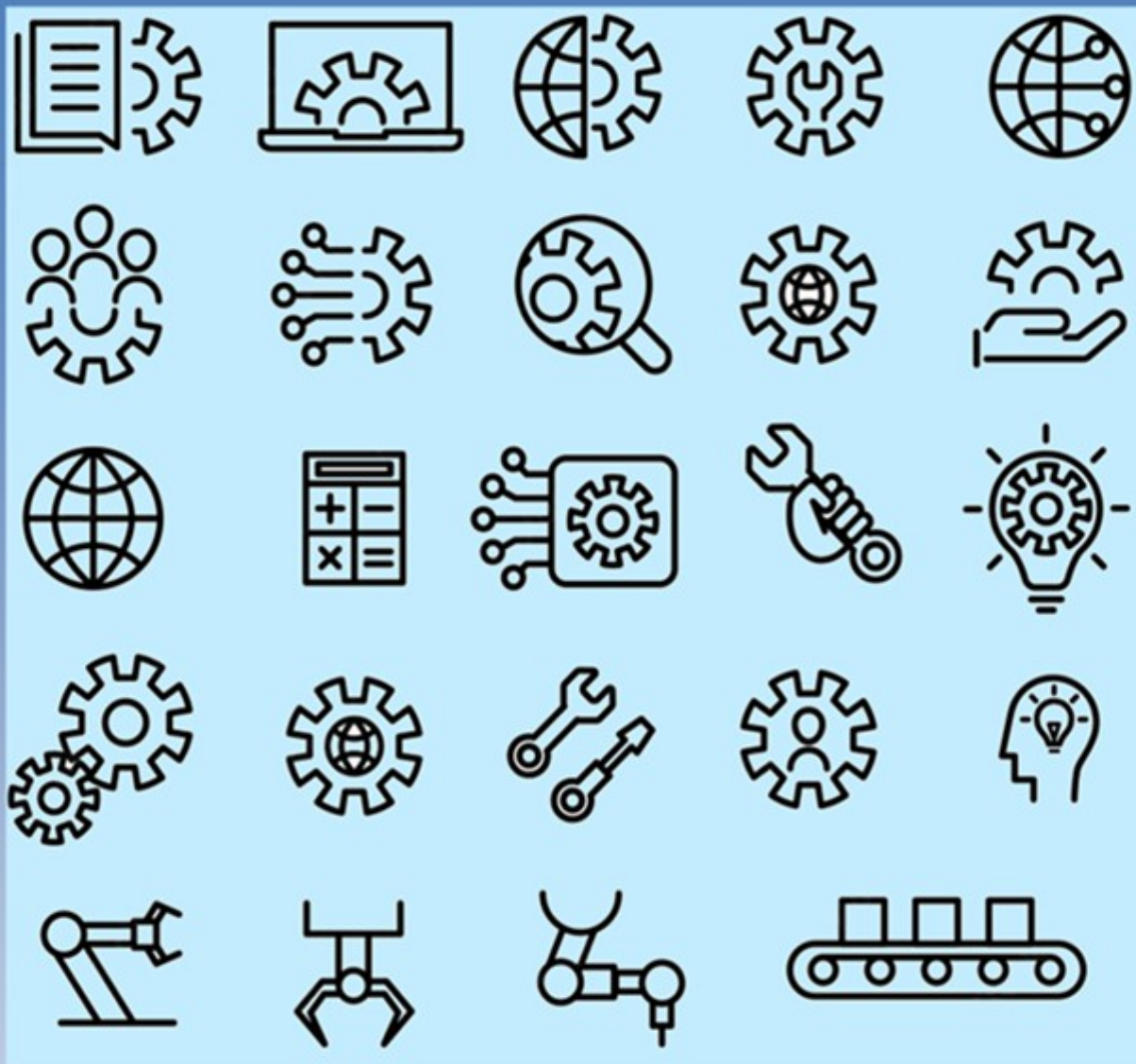


# Automazione Industriale e Meccatronica III



Arduino Simulations  
in TinkerCAD Without Hardware



Excel

WOKWi

Questo testo contiene una serie di lezioni ed esercitazioni che possono essere realizzate con

- il simulatore **ThinkerCad** disponibile sul sito <https://www.tinkercad.com/>
- il simulatore **Wokwi** disponibile sul sito <https://wokwi.com/>
- il simulatore **CircuitJS** disponibile sul sito <https://www.falstad.com/circuit/circuitjs.html>
- il simulatore **SimulIDE** disponibile sul sito <https://www.simulide.com/p/home.html>
- il simulatore **NL5 lite** disponibile sul sito <https://sidelinesoft.com/nl5/index.php?page=download>
- un foglio di calcolo (Excel, Calc ...)
- un kit Arduino R1

I simulatori in oggetto permettono di programmare una scheda Arduino UNO e di risolvere semplici problemi di automazione industriale utilizzando i più comuni componenti elettronici ed una serie di sensori ed attuatori.

La maggior parte delle esercitazioni proposte contiene una breve descrizione dei componenti utilizzati. Per ulteriori dettagli è necessario fare riferimento a testi specifici di elettronica ed automazione.

Una conoscenza di base dell'elettronica e dell'elettrotecnica è necessaria per capire gli schemi proposti.

Quasi tutti gli esercizi presentano una possibile soluzione software da caricare su una scheda Arduino.

Il vantaggio offerto dall'utilizzo di ThinkerCAD, rispetto ad altri software di simulazione, è la possibilità di replicare in modo identico il circuito su una breadboard e di utilizzare lo stesso programma simulato sulla scheda Arduino.

*Questo testo può essere utilizzato liberamente **SOLO PER SCOPI DIDATTICI**.*

*Qualsiasi altro utilizzo deve essere concordato con l'autore e non può essere distribuito su altri siti web.*

*Il testo aggiornato periodicamente è reperibile a questo indirizzo web:*

<http://www.energiazero.org/cartelle.asp?dir=thinkercad>

*Si ringrazia in anticipo per eventuali segnalazioni di errori e/o miglioramenti apportabili al testo alla seguente mail: [energiazero.org@gmail.com](mailto:energiazero.org@gmail.com)*

### **NOTA BENE:**

*Alcuni esempi e immagini sono stati reperiti sul web cercando materiale che non fosse coperto da copyright. Si ringraziano tutti coloro che hanno reso disponibile il materiale.*

*Se per errore fosse stato inserito materiale protetto, cortesemente segnalatelo alla mail sopra citata.*

## SOMMARIO

INTRODUZIONE .....	2
⤴ il microcontrollore .....	9
La scheda Arduino UNO R3 .....	10
I pin di Arduino: digitali, PWM ed analogici .....	12
Microcontrollore industriale .....	13
Modulo relè WiFi ESP32-S3 industriale a 8 canali .....	13
LINGUAGGIO DI PROGRAMMAZIONE DI ARDUINO .....	15
LE VARIABILI .....	15
LE COSTANTI .....	15
LE STRUTTURE PRINCIPALI .....	15
STRUTTURE DI CONTROLLO .....	16
OPERATORI ARITMETICI .....	18
OPERATORI di CONFRONTO e BOOLEANI .....	18
OPERATORI COMPOSTI .....	19
LE FUNZIONI di INPUT E OUTPUT .....	20
FUNZIONI TEMPORALI .....	22
FUNZIONI MATEMATICHE .....	23
FUNZIONI TRIGONOMETRICHE .....	23
NUMERI CASUALI .....	24
COMUNICAZIONE SERIALE .....	24
⤴ AUTOMAZIONE CON ARDUINO .....	25
CARATTERISTICHE E LIMITI .....	25
LIVELLI LOGICI .....	26
TTL (Transistor-Transistor Logic): .....	26
CMOS (Complementary Metal-Oxide-Semiconductor) .....	26
diodo LED .....	27
INGRESSO DIGITALE CON PARTITORE DI TENSIONE .....	28
PULSANTE (PUSH BUTTON) .....	29
INTERRUTTORE (SLIDER) .....	30
INTERRUTTORE e PULSANTE in modalita' PULL-UP (LOGICA INVERSA) .....	31
POTENZIOMETRO .....	32
GESTIONE RELE' CON ARDUINO .....	33
VALUTAZIONE DEL TEMPO TRASCORSO CON ARDUINO millis() .....	34
LETTURA ANALOGICA MEDIATA .....	35
ESERCIZIO millis() .....	36
Finecorsa meccanico .....	37
IL TRANSISTOR .....	39
transistor BJT ( <i>bipolar junction transistor</i> ) .....	40
transistor MOS (MOSFET) .....	40

BJT vs MOS (MOSFET): Caratteristiche Tecniche .....	41
Quando usare i transistor BJT e MOSFET .....	41
IL TRANSISTOR BJT .....	42
ESERCIZIO BJT .....	43
TRANSISTOR PER PILOTARE RELE' DI POTENZA (TENSIONE>5V) .....	44
ESERCIZIO BJT + RELE' .....	45
IL TRANSISTOR DI POTENZA (DARLINGTON) .....	46
TIP120 per attivare ELEMENTO riscaldante resistivo .....	47
IL TRANSISTOR MOSFET .....	48
DIMENSIONAMENTO MOSFET come interruttore .....	49
Controllo motore MOSFET di potenza .....	50
circuito motore CC MOSFET di potenza semplice .....	50
IRF520 MOSFET .....	51
ESERCIZIO CON NMOS .....	52
CONFRONTO FRA TRANSISTOR BJT E NMOS .....	53
SHIELD MOSFET 4 CANALI ROSSA .....	54
SHIELD MOSFET 4 CANALI BLU .....	55
Relay Shield Arduino IMPILABILE .....	56
Layout shield .....	57
TEST DEI 4 RELE' DELLO SHIELD .....	58
Simulazione nastro traspostatore .....	59
ELETTROVALVOLE PNEUMATICHE .....	60
PANNELLO DI ELETTROPNEUMATICA .....	61
Azionamento cilindri pneumatici .....	62
1° ESERCIZIO .....	62
2° ESERCIZIO .....	64
⌠ SENSORI E TRASDUTTORI SISTEMI DI CONTROLLO .....	66
Elenco dei più comuni trasduttori .....	67
Proprietà' Generali dei Sensori .....	67
Tempo di Risposta DEI PRINCIPALI SENSORI .....	68
Sensori Ottici e Fotoelettrici .....	68
Sensori di Pressione .....	68
Sensori di Prossimità (Induttivi e Capacitivi) .....	68
Sensori di Posizione e Distanza .....	68
Sensori di Temperatura (I più lenti) .....	68
SENSORI DI PROSSIMITA' .....	69
Tipi di sensori di prossimità .....	69
1. Sensore di prossimità induttivo .....	69
2. Sensore di prossimità capacitivo .....	70
3. Sensore di prossimità a ultrasuoni .....	71
4. Sensore di prossimità ottico .....	72

5. Sensore di prossimità magnetico .....	72
Configurazioni e applicazioni dei sensori .....	74
Contatto DRY (aSCIUTTO ) e WET (bagnato) .....	75
CABLAGGIO SENSORI 2 e 3 fili CON ALIMENTAZIONE.....	76
SENSORE A 2 FILI .....	76
SENSORE A 3 FILI .....	76
Sensore A INFRAROSSI a TRE fili E18-D80NK (NPN).....	77
Cablaggio sensore di distanza a infrarossi E18-D80NK (NPN).....	79
SENSORE A ULTRASUONI .....	80
Funzionamento del sensore per Arduino.....	80
Sensore di distanza ad ultrasuoni HC SR04 con Arduino .....	82
Principio di funzionamento:.....	82
ESERCIZIO THINKERCAD MISURA DISTANZA CON SENSORE ULTRASUONI .....	84
SISTEMI DI MOVIMENTAZIONE INDUSTRIALE .....	85
IL NASTRO TRASPORTATORE .....	86
Monitorare STATO sensori senza bloccare il codice .....	87
Guida lineare.....	89
SENSORE DI TEMPERATURA TMP36 .....	90
CURVA CARATTERISTICA DEL SENSORE TMP36 .....	91
TERMISTORE NTC (Negative Temperature Coefficient).....	94
MONITORARE TEMPerATURA TRAMITE TERMISTORE NTC .....	95
SISTEMI DI CONTROLLO DELLA TEMPERATURA CON MICROCONTROLLORE .....	96
SISTEMA CONTROLLO TEMPERATURA ON-OFF CON TRANSISTOR.....	97
SISTEMA CONTROLLO TEMPERATURA ON-OFF CON RELE' .....	98
CONTROLLO ON-OFF CON ISTERESI .....	99
CONTROLLO DI TEMPERATURA PROPORZIONALE .....	103
Controllo ON-OFF vs PROPORZIONALE .....	105
TERMORESISTENZE .....	106
Perché utilizzare un sensore al platino .....	106
Differenza tra Pt100 e Pt1000.....	106
Come scegliere il giusto sensore al platino .....	107
Sostituzione delle termoresistenze: nota sulle norme industriali .....	107
Convertire la resistenza Pt100/Pt1000 in temperatura .....	108
Curva caratteristica delle termoresistenze .....	109
Termocoppie .....	110
Digitalizzatore di termocoppia .....	112
Termocoppia Tipo K .....	113
Sonda termocoppia tipo K COMMERCIALE .....	114
Collegamento del modulo MAX6675 a un Arduino .....	115
ESTENSIMETRI INDUSTRIALI .....	116
Resistenza degli estensimetri.....	118

Il ponte di Wheatstone .....	118
Collegamento a quarto di ponte .....	118
Legame deformazione elastica E variazione di resistenza elettrica .....	119
Misura della deformazione E DELLA FORZA assiale .....	119
ESERCIZIO .....	120
FOGLIO DI CALCOLO per valutare deformazioni elastiche .....	121
CELLE DI CARICO .....	122
Scheda elettronica per Cella di carico - HX711 .....	123
Schema di collegamento ad Arduino .....	124
INTERRUPT E CONTEGGIO IMPULSI DA UN TRASDUTTORE .....	125
ENCODER MAGNETICI .....	127
ENCODER OTTICI .....	128
Encoder incrementale .....	128
Encoder incrementale: risoluzione .....	129
Encoder incrementale: esempio d'uso .....	129
Encoder assoluto .....	129
Encoder assoluto: single-turn o multi-turn .....	130
Misura di velocità dal segnale encoder .....	131
Encoder avanzati .....	131
ENCODER OTTICO AD INFRAROSSI .....	132
⏏ ATTUATORI .....	133
MOTORE IN CORRENTE CONTINUA (C.C.) .....	134
775 D SHAFT .....	135
Parti principali del motore C.C. A SPAZZOLE .....	136
Motori CC senza spazzole BLDC (brushless ) .....	137
Parti fondamentali del motore BLDC .....	137
Funzionamento del motore brushless .....	137
PWM (pulse wide modulation): modulazione di larghezza d'impulso .....	138
regolazione velocità di un motore d.c. mantenendo alta la coppia motrice .....	139
Problemi nella regolazione PWM di un motore DC .....	139
ESERCIZIO PWM MOTORE CC .....	141
Disturbi elettromagnetici nei motori CC a spazzole .....	142
Controllo velocità motore CC in PWM con Mosfet .....	143
INVERSIONE VERSO DI ROTAZIONE MOTORE C.C. CON 2 RELE' .....	146
ESERCIZIO VERSO ROTAZIONE MOTORE CON RELE' .....	146
DRIVER L298N H-Bridge .....	148
SERVOMOTORI .....	150
Tipologie di riduttori per l'automazione .....	151
PLANETARIO (EPICICLOIDALE) .....	151
CICLOIDALE .....	152
ARMONICO .....	154

Il gioco NEI RIDUTTORI (backlash).....	155
Comparazione e scelta .....	156
Analisi dei Punti Chiave.....	156
GESTIONE SERVOMOTORE DIRETTA CON ARDUINO .....	157
ESERCIZIO GESTIONE SERVOMOTORE CON ARDUINO E POTENZIOMETRO .....	158
MOTORE STEPPER (PASSO-PASSO) .....	159
DRIVER A4988 .....	161
Utilizzo del driver passo-passo A4988 .....	162
DRIVER DRV8825 contro A4988 .....	164
△ SISTEMI DI REGOLAZIONE.....	165
SISTEMA DI RISCALDAMENTO resistivo .....	165
△ SISTEMI DI CONTROLLO .....	167
SCHEMA A BLOCCHI DI SISTEMA DI CONTROLLO DI TEMPERATURA .....	168
ESEMPIO CONTROLLO PID con transistor .....	169
GENERARE SEGNALI ANALOGICI (DAC) CON ARDUINO .....	170
ESERCIZIO: VARIARE LA LUMINOSITA' DI UN DIODO LED .....	171
COME VARIARE LA VELOCITA' DI UN MOTORE C.C. MANTENENDO ALTA LA COPPIA MOTRICE .....	172
SISTEMA DI CONTROLLO ON-OFF .....	173
SISTEMA DI CONTROLLO PID (proporzionale – integrale – derivativo) .....	174
IMPLEMENTAZIONE NUMERICA PID .....	175
Integrazione numerica dell'errore (metodo dei rettangoli) .....	176
Derivazione numerica dell'errore .....	176
Tempo di campionamento di un sistema automatico .....	177
Cosa succede con un tempo di campionamento errato? .....	177
Simulazione sistema di controllo di temperatura di un locale che disperde calore .....	178
Controllo ON-OFF .....	179
Controllo ON-OFF con isteresi.....	180
Controllo proporzionale P .....	181
Controllo proporzionale-integrale PI .....	182
Regole di Ziegler-Nichols.....	183
△ robotica industriale .....	184
Sistemi robotici .....	185
Tipi di giunto .....	185
Tipi di robot.....	185
VIDEO .....	186
Robot collaborativi (cobot) .....	187
CINEMATICA DEL ROBOT .....	188
CINEMATICA DIRETTA DEL ROBOT PLANARE A 2 LINK.....	188
FOGLIO DI CALCOLO.....	188
CINEMATICA INVERSA DEL ROBOT DEL ROBOT PLANARE .....	190
FOGLIO DI CALCOLO.....	190

ESERCIZIO CINEMATICA INVERSA DEL ROBOT DEL LASER PLANARE .....	191
Servo SG90 .....	192
Individuare la posizione 0 .....	193
Regolazione della velocità .....	194
Servo a rotazione continua .....	194
Servo MG90S.....	195
Dynamixel XM540-W270-T/R .....	196
Esercizio .....	198
Diagramma di flusso .....	199
ROBOT SCARA .....	202
MOVIMENTI E ANGOLI DEL ROBOT SCARA .....	202
Applicazioni tipiche DEL ROBOT SCARA .....	203
END EFFECTOR .....	203
COMANDO SCARA.....	204
ROBOT ANTROPOMORFO .....	208
Gradi di libertà di un robot .....	208
Applicazioni tipiche suddivise per settore: .....	209
Robot seriali e quasi seriali .....	210
Principali produttori di robot industriali .....	211
CINEMATICA DIRETTA ED INVERSA robot a 3 link.....	212
Cinematica robot Antropomorfo con EXCEL .....	213
Scara vs antropomorfo .....	214

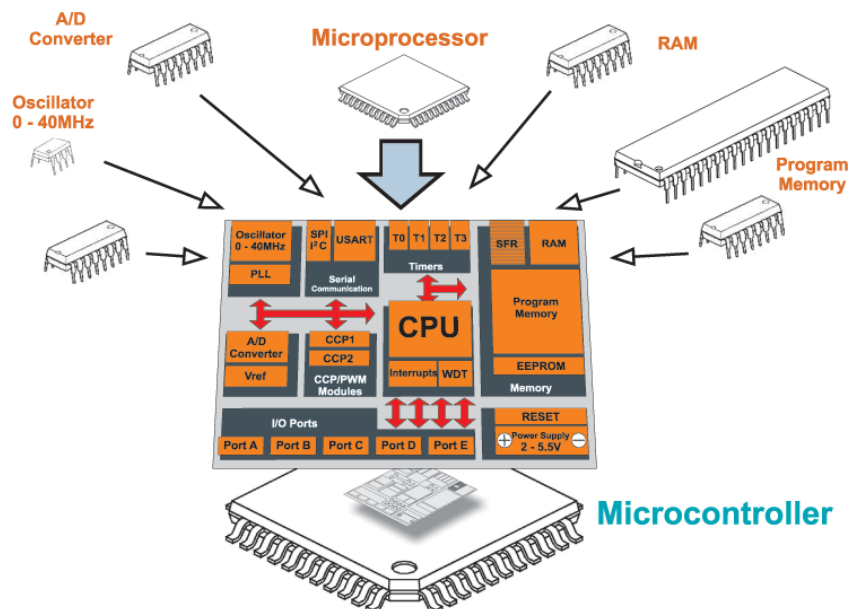
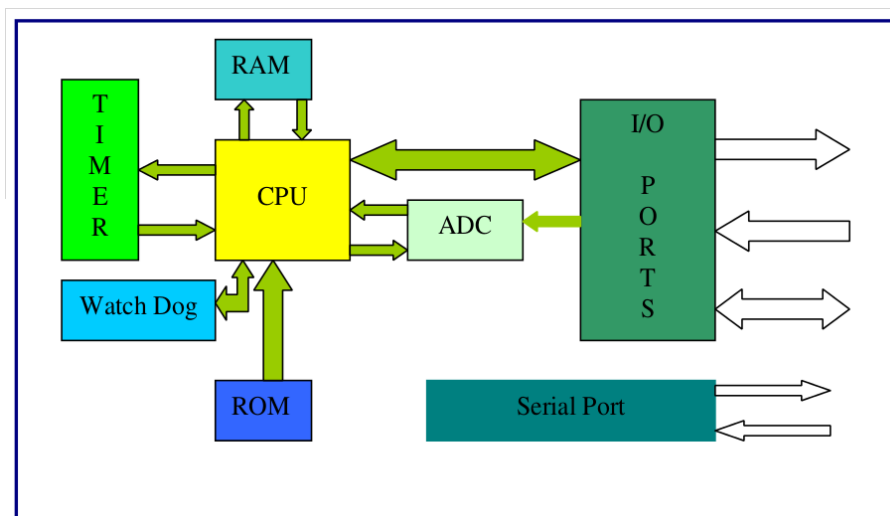


# IL MICROCONTROLLORE

Un microcontrollore (microcontroller o MCU, MicroController Unit) è un single-chip computer, ovvero un microcalcolatore integrato su un singolo chip. Come suggerisce il nome, il microcontrollore è utilizzato principalmente per realizzare sistemi di controllo digitale e, in particolare, nei dispositivi cosiddetti embedded. Si tratta di sistemi elettronici di elaborazione a microprocessore progettati appositamente per una determinata applicazione (special purpose) ovvero non riprogrammabili dall'utente per altri scopi.

Il microcontrollore si differenzia rispetto al microprocessore in quanto al proprio interno contiene normalmente anche una certa quantità di memoria RAM e di EPROM e vari dispositivi periferici integrati, come timer, convertitori AD etc. Si tratta dunque di un vero e proprio computer completo di tutto ciò che occorre per il suo funzionamento.

La figura seguente mostra uno schema della struttura interna di un MCU.

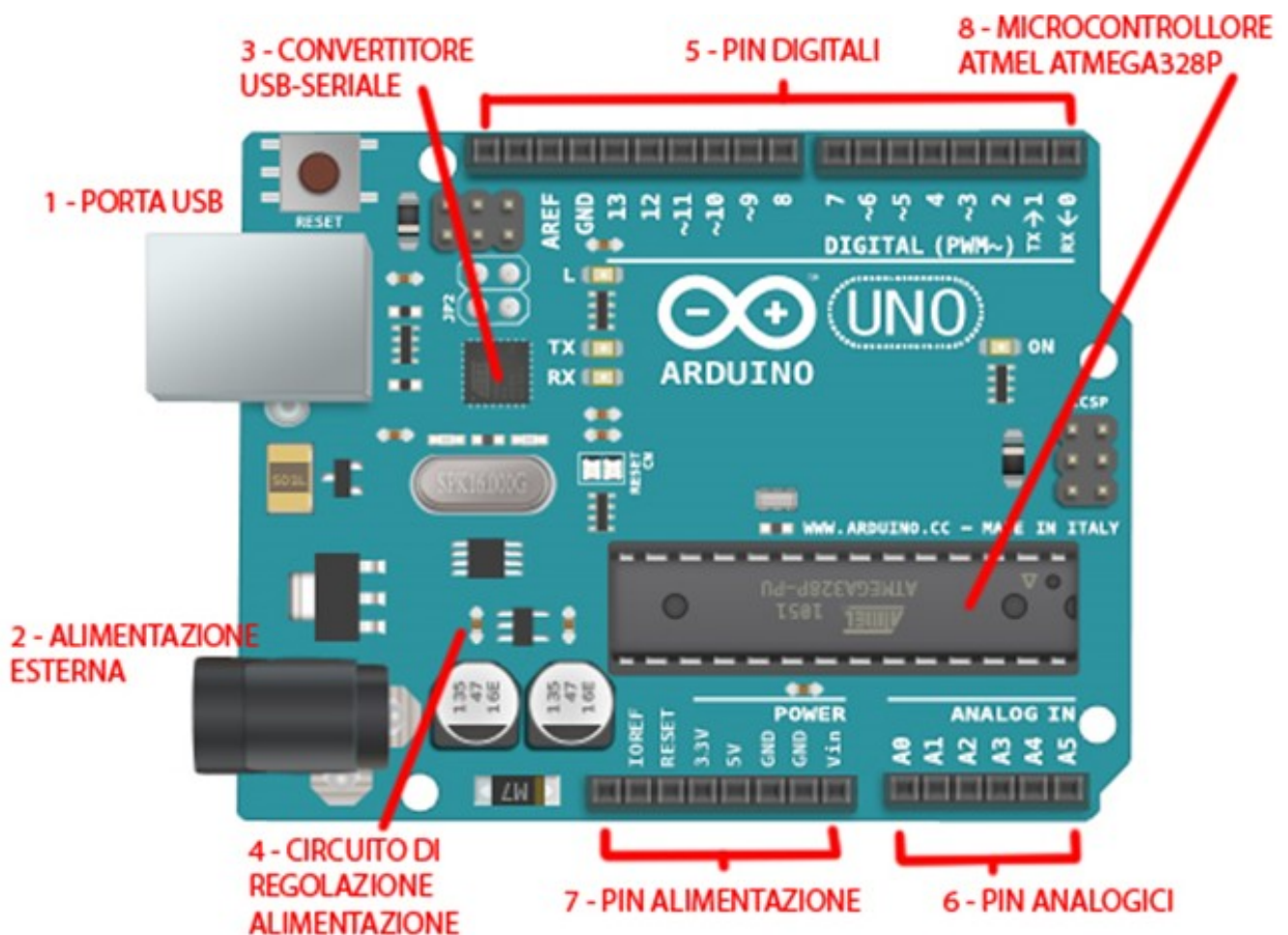


Si noti che generalmente il MCU non viene collegato a chip di memoria esterni (a differenza del microprocessore): l'intero programma di gestione del MCU e i relativi dati devono dunque risiedere sulla memoria interna integrata on chip.

A differenza del Personal Computer (PC), che è un dispositivo general purpose (cioè di applicazione generale, che può eseguire un gran numero di programmi diversi), i microcontrollori hanno una potenza piuttosto limitata e sono utilizzati in applicazioni specifiche, spesso per eseguire sempre lo stesso identico compito.

Arduino è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore. È stata ideata e sviluppata nel 2005 da alcuni membri dell'Interaction Design Institute di Ivrea come strumento per la prototipazione rapida e per scopi hobbistici, didattici e professionali. Il nome della scheda deriva da quello del bar di Ivrea frequentato dai fondatori del progetto, nome che richiama a sua volta quello di Arduino d'Ivrea, Re d'Italia nel 1002.

Con Arduino si possono realizzare in maniera relativamente rapida e semplice piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, automatismi per il controllo della temperatura e dell'umidità e molti altri progetti che utilizzano sensori, attuatori e comunicazione con altri dispositivi. La scheda è abbinata a un semplice ambiente di sviluppo integrato per la programmazione del microcontrollore. Tutto il software a corredo è libero, e gli schemi circuitali sono distribuiti come hardware libero e per questo motivo è molto utilizzato nella didattica educativa.



### 1- PORTA USB

È la porta con cui si collega la scheda al computer tramite cavo apposito. Il suo ruolo, ovviamente, è anche quello di scambiare i dati con il computer permettendo l'upload dello sketch.

Una volta che lo sketch è caricato sulla scheda, questa porta può anche essere utilizzata per alimentare la scheda con un alimentatore esterno da 5 V con uscita USB.

### 2- ALIMENTAZIONE ESTERNA

Questo jack permette l'alimentazione esterna alla scheda.

Si suggerisce di non superare i 12 V onde evitare problemi di stabilità e surriscaldamento della scheda.

### **3- CONVERTITORE USB-SERIALE**

È una parte importantissima della scheda che consente la comunicazione bidirezionale tra il computer e la scheda, in particolare, tra il computer e il microcontrollore, scambiando dati e consentendo l'upload degli sketch.

### **4- CIRCUITO DI REGOLATORE DI ALIMENTAZIONE**

Nel caso siano presenti sia alimentazioni tramite USB che tramite jack, grazie a questo ripartitore, la scheda è in grado di scegliere dove prendere la tensione necessaria.

È considerata fonte primaria quella proveniente dal jack esterno. In ogni caso, come già detto, la tensione proveniente dal jack non dovrebbe mai superare i 12V, ma nemmeno essere inferiore ai 7V. In quest'ultimo caso, infatti, è possibile che il sistema non riesca a fornire alla scheda i 5V nominali di cui ha bisogno per funzionare.

### **5- PIN DIGITALI**

Sono 14 PIN che rispondono ad una logica digitale I/O. Significa che possono essere collegati in lettura a dei sensori o a dei dispositivi esterni a patto che essi funzionino con logica digitale.

La logica digitale è quella booleana, ovvero che prevede solo due stati, 0 e 1 (acceso/spento, on/off, alto/basso) e che, in termini elettrici, associa 0V allo 0 e 5 V all'1.

Se colleghiamo a questo PIN una lampadina potremmo accenderla e spegnerla non modulare, almeno in linea teorica, la sua luminosità.

Una menzione speciale meritano i PIN 3, 5, 6, 9, 10 e 11 che possono essere utilizzati come PIN analogici e impulsi PWM (Pulse Width Modulation che vedremo negli esempi in seguito) utilissimi per la regolazione di attuatori come motori e servomotori.

### **6- PIN ANALOGICI**

Sono 6 PIN che possono leggere e inviare segnali analogici, con valori cioè compresi tra 0V e 5V.

In particolare il microcontrollore legge la tensione presente sul PIN e restituisce un valore compreso tra 0 e 1023 (un numero a 10 bit).

Alcuni sensori provvedono a mappare il valore risultante nella scala desiderata (ad esempio temperatura o distanza), in altri casi, la conversione va effettuata nel codice stesso.

### **7- PIN ALIMENTAZIONE**

Sono i PIN dedicati all'alimentazione dei sensori, degli attuatori o dei circuiti creati e collegati alla scheda. Possono fornire una tensione di 5V e 3,3V con i rispettivi PIN, mentre quelli contrassegnati con GND servono per raccogliere la "massa", il ritorno della corrente dal circuito.

Una piccola menzione per i due PIN RESET e Vin. Il primo serve per resettare il microcontrollore con azione identica a quella del pulsante dedicato installato a bordo macchina.

Il secondo permette di prelevare alimentazione in quantità pari a quella fornita dal jack o di restituire la stessa direttamente al regolatore di tensione di Arduino (non useremo questi due PIN in questo corso).

### **8- MICROCONTROLLORE ATMEL ATMEGA328P**

È il vero cuore della scheda che consente di agire come microcontrollore, controllando cioè dispositivi esterni, integrando quella che è la memoria su cui è salvato il programma.

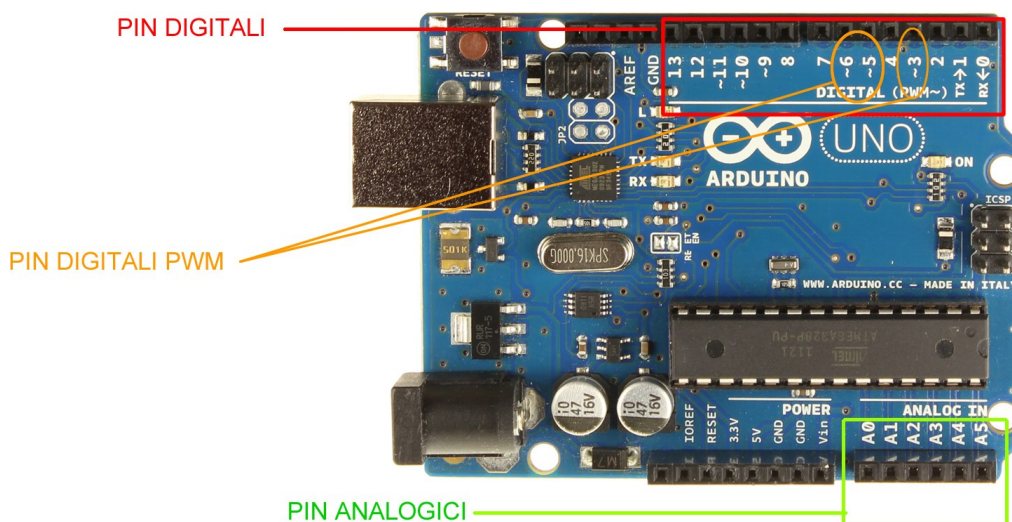
## I PIN DI ARDUINO: DIGITALI, PWM ED ANALOGICI

I PIN DI ARDUINO SONO LE PORTE CHE CONSENTONO ALLA SCHEDA DI COMUNICARE E QUINDI RICEVERE ED EMETTERE INFORMAZIONI VERSO I DISPOSITIVI AD ESSO CONNESSI.

Arduino ha un totale di 19 pin che si dividono in 2 macro categorie: pin analogici (5 pin) e pin digitali (14 pin). I pin digitali sono utilizzabili sia per ricevere segnali e quindi acquisire informazioni (input) che per emettere segnali ossia spedire informazioni (output). I pin digitali si dividono a loro volta in base al supporto o meno della funzione PWM.

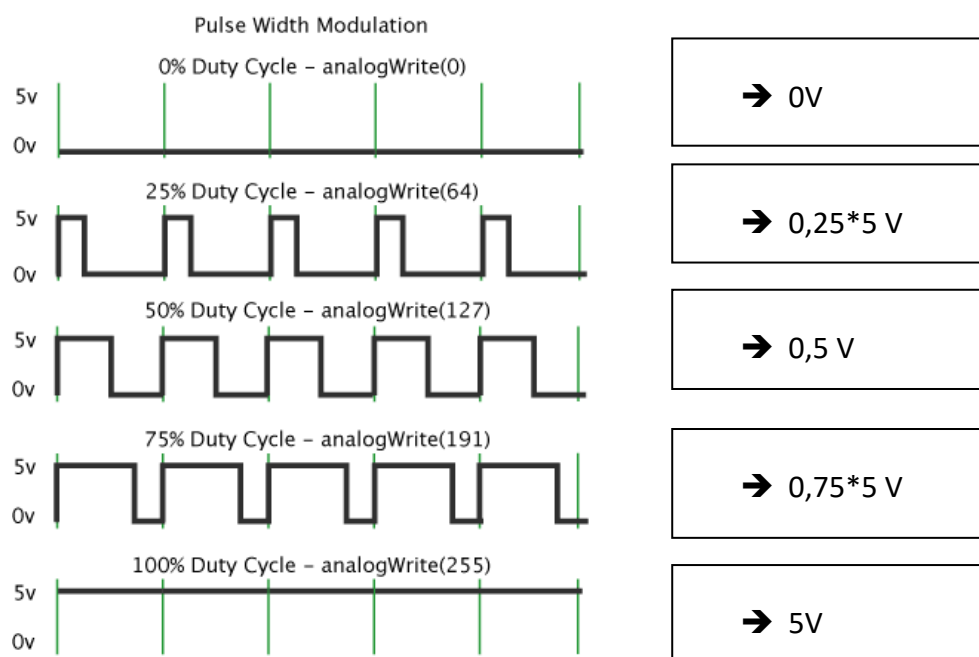
I pin che non hanno PWM sono: 1,2,4,7,8,12,13,15.

Questi pin sono come detto prima in grado di gestire solo segnali 0 e 1 (low/high) il che significa che possono essere utilizzati in situazioni come un relè, un pulsante e tutte quelle situazioni in cui vi è sono fondamentalmente 2 stati o possibilità.



Con un pin PWM è possibile generare in uscita un segnale analogico da 0-5V con una risoluzione di 8 bit ( $5/255$  volt  $\approx 0,02V$ ). Un segnale PWM (pulse wide modulation) è in termini molto semplicistici, un onda quadra 0-5V (ad alta frequenza) con delle durate prestabilite per la parte alta (5V).

Ciò permette di simulare un valore analogico di tensione compreso tra 0-5V con uno digitale con la maggior parte degli attuatori (transistor, relè, motori CC ...).



I pin analogici invece sono in grado solo di ricevere segnali ed hanno un range che va da 0 a 1023. Questa tipologia di pin è utilizzata quindi per leggere tutti quei sensori come trimmer, potenziometri, fotoresistenze, ultrasuoni, IR.

A differenza di un microcontrollore per uso obbistico o domestico, quello industriale deve essere in grado di funzionare anche in ambienti critici e di interfacciarsi facilmente con altri dispositivi industriali che tipicamente funzionano a 24-30V o in tensione alternata.

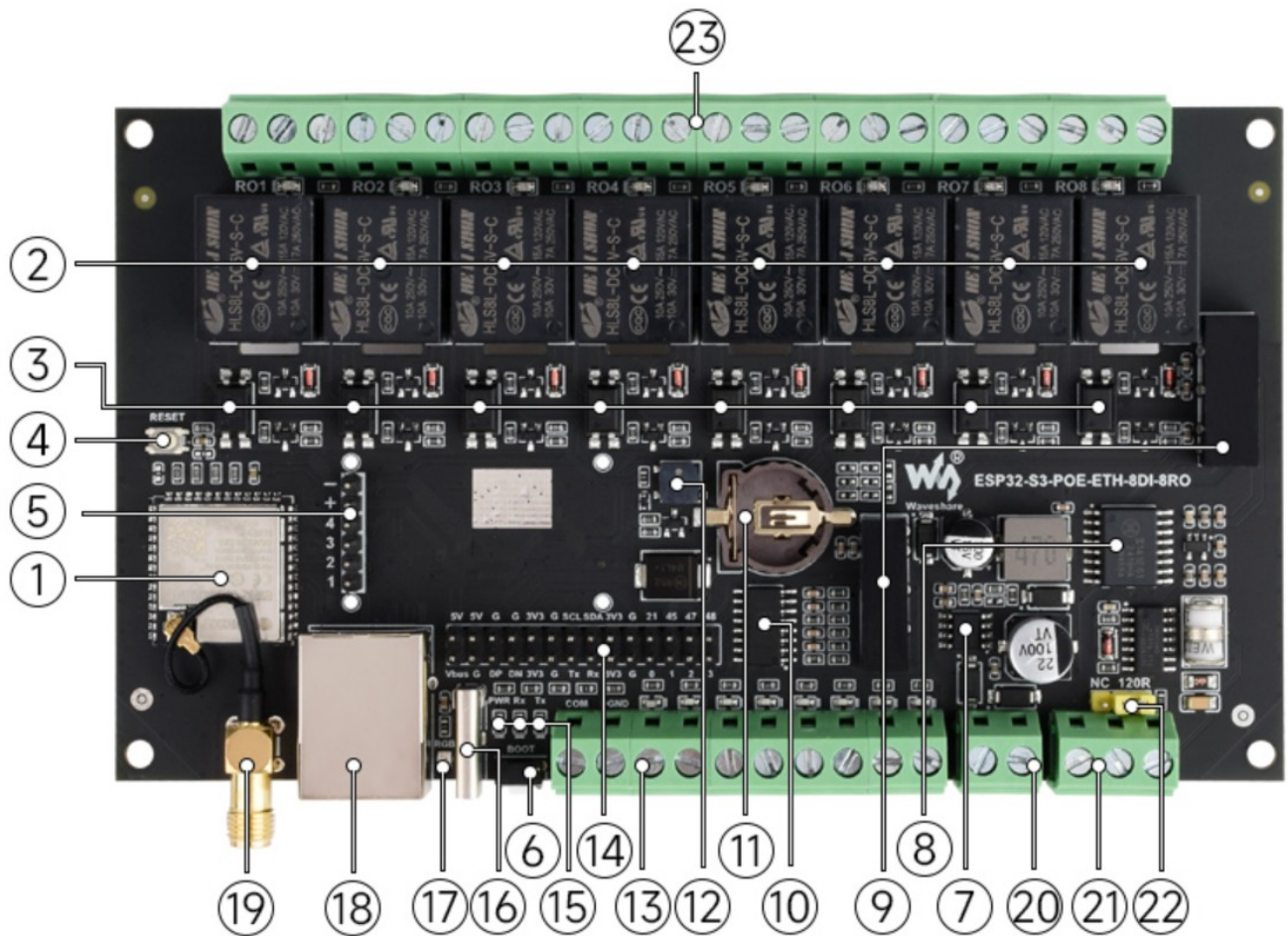
Un esempio di microcontrollore industriale basato sul chip ESP32 è il seguente.

### MODULO RELÈ WIFI ESP32-S3 INDUSTRIALE A 8 CANALI

Basato su ESP32-S3, supporta WiFi / Bluetooth.

Interfacce di ingresso digitale, RS485 e porta Ethernet integrate.

Circuiti di protezione integrati come isolamento di potenza e isolamento optoaccoppiatore, sicuri, stabili e affidabili.

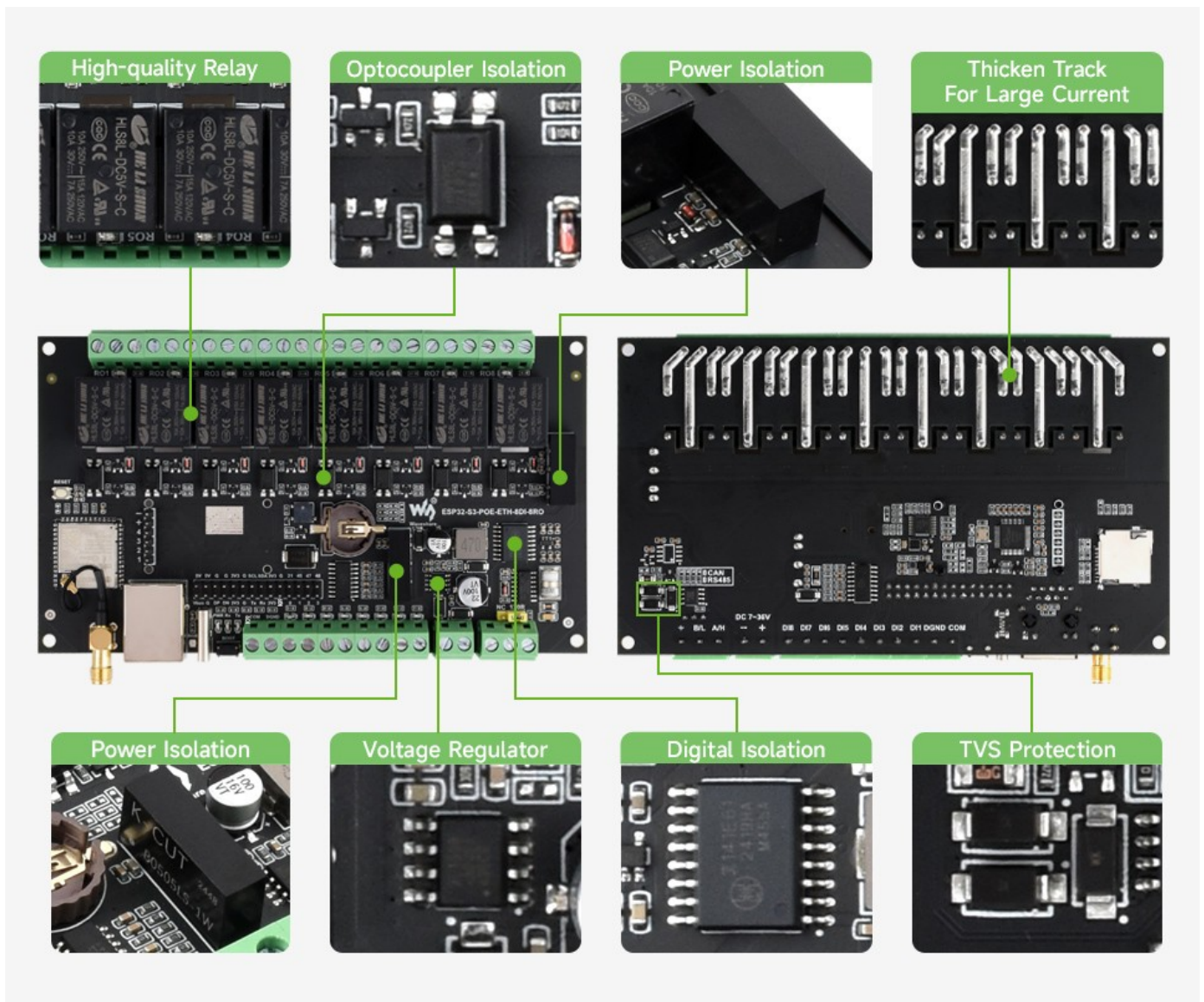


#### Caratteristiche

- Basato sul microcontrollore ESP32-S3 con processore dual-core Xtensa LX7 a 32 bit, in grado di funzionare a 240 MHz
- Comunicazione wireless dual-mode Wi-Fi a 2,4 GHz e Bluetooth LE integrata, con prestazioni RF superiori
- Relè di alta qualità, portata dei contatti:  $\leq 10A$  250V AC / 30V DC
- Supporta ingressi digitali passivi e attivi, con isolamento optoaccoppiatore bidirezionale. Il relè supporta il controllo del collegamento degli ingressi digitali.
- Interfaccia RS485 isolata integrata, per la connessione a vari moduli o sensori industriali RS485 Modbus
- Connettore pin integrato, che consente l'accesso ad altri dispositivi
- Porta USB Type-C integrata per alimentazione, download del firmware e debug
- Morsetto a vite per alimentazione integrata, supporta un ingresso di tensione ampio 7~36 V, adatto per applicazioni industriali
- Chip RTC integrato, supporta attività pianificate

- Chip Ethernet W5500 integrato per estendere la porta di rete 10/100Mbps tramite interfaccia SPI
- Opzionale per la versione con porta di rete PoE, con modulo PoE integrato per funzionalità PoE (conforme allo standard IEEE 802.3af)
- Isolamento dell'optoaccoppiatore integrato per prevenire interferenze dal circuito ad alta tensione esterno collegato al relè
- Isolamento digitale integrato per prevenire interferenze da segnali esterni
- Isolamento dell'alimentatore unibody integrato, che fornisce una tensione isolata stabile, senza bisogno di alimentazione aggiuntiva per il terminale isolato
- Slot per scheda TF integrato per l'archiviazione di immagini e file su scheda TF esterna
- Cicalino integrato, LED RGB, alimentatore e indicatori RS485 TX/RX per il monitoraggio dello stato operativo dei dispositivi
- Custodia in ABS montata su guida, facile da installare, sicura da usare

Circuiti di protezione di isolamento multipli integrati



Supporta RS485 e controllo remoto Bluetooth / WiFi

Relè a 8 canali integrati e ingressi digitali a 8 canali

Portata dei contatti del relè integrato fino a 10 A 250 V CA / 30 V CC

Controllo diretto di elettrodomestici a 220 V CA o dispositivi inferiori a 30 V CC

## LE VARIABILI

Sono dei contenitori di dati, il loro valore viene modificato durante l'esecuzione del programma.

**boolean** - E' una variabile booleana, quindi il suo valore è vero o falso.

**byte** - Contiene un numero tra 0 e 255.

**int** - Contiene un numero intero compreso tra -32'768 e 32'767 (16 bit, 2 byte). Mettendo davanti ad int "const" diventa una costante, quindi dopo che è stata dichiarata non può più cambiare. (viene usata ad esempio per assegnare un nome ad un pin).

**short** - Come "int" ma anche per "Arduini" con architettura Arm (es. Arduino Due, int in queste schede è di 4 byte)

**unsigned int** - Come int ma solo numeri positivi, quindi tra 0 e 65'535.

**word** - Come "unsigned int" ma anche per "Arduini" con architettura Arm.

**long** - Contiene un numero tra -2'147'483'648 e 2'147'483'647 (32 bit, 4 byte).

**unsigned long** - Come long ma solo numeri positivi, quindi da 0 a 4'294'967'295.

**float** - Può memorizzare numeri con la virgola.

**double** - Nelle schede con architettura Arm contiene un numero fino a  $1'7976931348623157 \times 10^{308}$  (8 byte).

**char** - Contiene un singolo carattere di testo (il numero corrispondente nella tabella ASCII).

**string** - Contiene più caratteri di testo. Es: `char Str1[] = "esempio";`

## LE COSTANTI

Le costanti sono le variabili preimpostate nel linguaggio di Arduino

**INPUT** e **OUTPUT** - sono usate per definire se uno specifico Pin deve essere di ingresso o di uscita.

**HIGH** e **LOW** - sono usati per esempio quando si vuole accendere o spegnere un Pin di Arduino.

**true** e **false** - indicano che la condizione può essere vera o falsa.

## LE STRUTTURE PRINCIPALI

La struttura base di un programma Arduino si sviluppa in almeno due parti:

### **void setup ( )**

{Qui mettiamo la parte dello sketch che deve essere eseguita una sola volta (ad esempio dichiarazioni di input e output).}

### **void loop ( )**

{Qui mettiamo la parte dello sketch che deve essere eseguita ciclicamente fino allo spegnimento di Arduino. Le istruzioni vengono eseguite in sequenza dalla prima all'ultima.}

## STRUTTURE DI CONTROLLO

Le strutture di controllo servono a far eseguire al nostro Arduino delle operazioni di logica

**If** - è il "se" di Arduino, tramite questa struttura è possibile prendere delle decisioni all'interno del programma.

*Esempio: se a è maggiore di b accendi "led1", altrimenti il "led1" rimarrà spento.*

```
if (a > b)
{
  digitalWrite(led1, HIGH);
}
```

**If...else** - come if ma se la condizione messa tra parentesi è falsa verrà eseguito tutto il codice che segue else.

*Esempio: se a è maggiore di b accendi "led1". Altrimenti accendi "led2".*

```
if (a > b)
{
  digitalWrite(led1, HIGH);
}
else
{
  digitalWrite(led2, HIGH);
}
```

**for** - Ripete il codice per il numero di volte inserito.

*Esempio: scrivi 3 volte "esempio" sul monitor seriale.*

```
for (int i=0;i<3;i++)
{
  Serial.print("esempio");
}
```

**switch case** - Vengono eseguiti diversi blocchi di programma a seconda del valore della variabile posta tra parentesi.

*Esempio: se il valore di "sensore1" è uguale a 600 accendi "led1", se è uguale a 700 accendi "led2".*

*Se il valore di "sensore1" è diverso sia da 600 che da 700, spegni "led1" e "led2".*

```
switch (sensore1)
{
  case 600:
    digitalWrite(led1, HIGH);
    break;
  case 700:
    digitalWrite(led2, HIGH);
    break;
  default:
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
}
```

**while** - Esegue un blocco di codice infinite volte fino a quando la condizione posta tra le parentesi diventa vera. (se lo è già all'inizio non viene eseguito)

*Esempio: tieni acceso "led1" finchè "sensore1" diventa più piccolo di 600.*

```
while (sensore1 < 600)
{
  digitalWrite (led1, HIGH);
}
```

**do while** - Il ciclo "Do While" funziona nello stesso modo del ciclo While, con l'eccezione che viene provata la condizione solo al termine del ciclo, in questo modo il ciclo "Do While" verrà eseguito sempre almeno una volta.

*Esempio: attendi finchè il valore di un sensore diventa stabile, aspetta 50 millisecondi infinite volte, finchè il valore del sensore diventa più basso di 100.*

```
do
{
  delay(50);
  x = readSensors();
}
while (x < 100);
```

**Break** - Questa struttura serve a bloccare un ciclo "for", "while" o "do". Viene utilizzato anche per separare le varie condizioni nella funzione "switch case".

**Continue** - Questo comando fa saltare il resto del codice all'interno del ciclo, e riavvia il ciclo.

*Esempio: Crea un salto tra l'incremento del valore di "x"*

```
for (x = 0; x < 255; x ++ )
{
  if ( ( x > 120 ) && ( x < 180 ) )
  {
    continue;
  }
  analogWrite ( PWMpin, x );
  delay (50);
}
```

**Return** - Termina una funzione che si sta eseguendo e ne restituisce un risultato.

*Esempio: se la lettura è maggiore di 400 restituisci 1, altrimenti 0*

```
int checkSensor ( )
{
  if ( analogRead(0) > 400 )
  {
    return 1;
  }
  else
  {
    return 0;
  }
}
```

## OPERATORI ARITMETICI

Questi operatori vengono utilizzati per cambiare il valore contenuto in una variabile.

**=** - Assegnazione: assegna ad una variabile un valore.

*Esempio: assegna alla variabile "valoresensore" il valore di tensione presente in ingresso al pin analogico 0*

```
valoresensore = analogRead(0);
```

**+** - Addizione: aggiungi un valore ad una variabile.

*Esempio: somma al valore di "Y" il valore 3.*

```
Y = Y + 3;
```

**-** - Sottrazione: sottrai un valore ad una variabile.

**\*** - Moltiplicazione: moltiplica una variabile per un valore.

**/** - Divisione: dividi una variabile per un valore.

**%** - Modulo: assegna alla variabile il valore del resto di una divisione.

*Esempio: 7 diviso 5 uguale 1 con resto 2. La variabile x ora avrà valore 2.*

```
X = 7 % 5;
```

## OPERATORI DI CONFRONTO E BOOLEANI

Questi operatori vengono usati all'interno degli "if" per testare i valori delle variabili.

**==** - Uguale a

*Esempio: se "variabile1" è uguale a 10, accendi "led1".*

```
if ( variabile1 == 10)
{
    digitalWrite (led1, HIGH);
}
```

**!=** - Diverso da

< - Minore di

> - Maggiore di

<= - Minore o uguale a

>= - Maggiore o uguale a

Se si vogliono testare più condizioni nello stesso "if" si devono usare gli operatori booleani:

**&&** - "and" testa se la condizione 1 e la condizione 2 sono vere

*Esempio: se "valoresensore" è compreso tra 100 e 200, esegui il codice successivo.*

```
if ( valoresensore >= 100) && (valoresensore <= 200)
{
  \\inserire il codice da eseguire
}
```

**||** - "or" testa se la condizione 1 o la condizione 2 sono vere

*Esempio: se "valoresensore1" o "valoresensore2" sono maggiori di "100", esegui il codice successivo.*

```
if ( valoresensore1 > 100) || (valoresensore2 > 100)
{
  \\inserire il codice da eseguire
}
```

**!** - "not" testa se la condizione è falsa

*Esempio: se "x" vale "falso" (quindi zero) esegui il codice successivo.*

```
if ( !x)
{
  \\inserire il codice da eseguire
}
```

## OPERATORI COMPOSTI

Servono a eseguire operazioni come incrementare il valore di una variabile.

**++** - Incremento.

*Esempio: incrementa di uno il valore di "val" (val++ è come scrivere val = val+1)*

```
val++;
```

**--** - Decremento.

**+=** - Addizione composta.

*Esempio: incrementa di "y" il valore "val" (val += y è come scrivere val = val+y)*

```
val += y;
```

**-=** - Sottrazione composta.

**\*=** - Moltiplicazione composta.

**/=** - Divisione composta.

## LE FUNZIONI DI INPUT E OUTPUT

Sono le funzioni necessarie all'utilizzo dei pin I/O di Arduino.

**pinMode ( )** - Serve a definire se intendiamo utilizzare un pin come ingresso o come uscita.

*Esempio: classico esempio di lampeggio di un led, in questo caso definisco il pin "ledPin" come uscita (OUTPUT) perché devo collegarci un led che è un dispositivo di uscita.*

*Se collegassi un pulsante avrei dovuto definirlo come ingresso (INPUT)*

```
int ledPin = 13;
```

```
void setup()
```

```
{  
pinMode ( ledPin, OUTPUT );  
}
```

```
void loop()
```

```
{  
digitalWrite ( ledPin, HIGH );  
delay ( 1000 );  
digitalWrite ( ledPin, LOW );  
delay ( 1000 );  
}
```

**digitalWrite ( )** - Permette di scrivere un valore su un pin digitale. Tipicamente viene usato per portare un pin di uscita a livello alto (5 Volt su Arduino Uno) o basso (0 Volt).

Può anche essere usato per forzare un pin di ingresso a livello alto o basso, tramite le resistenze di pull-up interne ad Arduino.

*Esempio di utilizzo: accende il led interno collegato al pin 13*

```
digitalWrite ( ledPin, HIGH );
```

**digitalRead ( )** - Consente di leggere il valore di un pin, essendo digitale il valore può assumere solo 2 valori: alto (HIGH) o basso (LOW)

*Esempio: leggi lo stato di un pulsante, successivamente accendi un led se il pulsante è premuto, spegnilo se il pulsante è a riposo.*

```
int ledPin = 13;
```

```
int pulsantePin = 3;
```

```
int val = 0;
```

```
void setup()
```

```
{
```

```
pinMode ( ledPin, OUTPUT );  
pinMode ( pulsantePin, INPUT );  
}
```

```
void loop()  
{  
val = digitalRead ( pulsantePin );  
digitalWrite ( ledPin, val );  
}
```

**analogRead ( )** - Legge la tensione applicata su un pin di ingresso analogico.

Il valore di questa tensione può andare da 0 a 5V e viene acquisita da Arduino come un valore su una scala tra 0 e 1023.

**analogWrite ( )** - Manda in uscita una tensione 0 - 5 Volt modulata in PWM. Questa tensione viene vista da molti utilizzatori (ad esempio un led) come una tensione variabile.

Per ottenere ciò dobbiamo scrivere nel campo del valore un numero tra 0 e 255, dove 0 è il led spento e 255 è il led acceso alla massima luminosità.

*Esempio: leggendo il valore analogico di un potenziometro collegato al pin 3 si otterrà un valore con un range da 0 a 1023.*

*Dividiamo questo numero per 4 (quindi circa 255) e portiamolo in uscita ad un led collegato su un pin pwm (ad esempio il 9).*

*Otterremo una regolazione della luminosità del led girando il potenziometro.*

```
int ledPin = 9;  
int potenziometroPin = 3;  
int val = 0;  
  
void setup()  
{  
pinMode ( ledPin, OUTPUT );  
}  
  
void loop()  
{  
val = analogRead ( potenziometroPin );  
analogWrite ( ledPin, val / 4 );  
}
```

## FUNZIONI TEMPORALI

Queste funzioni sono quelle che ci permettono di mettere in pausa il programma o di conoscerne il tempo trascorso dal suo avvio

**millis ( )** - Restituisce il numero in millisecondi trascorsi da quando il programma è partito.

*Esempio: stampa sul monitor seriale il tempo trascorso dall'avvio del programma*

*unsigned long tempo;*

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.print("Time: ");
  tempo = millis();
  Serial.println(tempo); // va a capo
  delay(1000);
}
```

**micros ( )** - Stessa cosa di millis ma in microsecondi.

**delay ( )** - Mette in pausa il programma per il valore (in millisecondi) che inseriamo tra parentesi.

*Esempio: anche qui metto l'esempio di un lampeggio di un led.*

*Il led viene acceso, passano 1000 millisecondi (1 secondo) poi si spegne, passano altri 1000 millisecondi e il loop riparte.*

*int ledPin = 13;*

```
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

**delayMicroseconds ( )** - Stessa cosa di delay ma in microsecondi.

## FUNZIONI MATEMATICHE

**min ( x, y )** - Calcola il valore più piccolo fra x e y.

*Esempio: assegna alla variabile "Val" il valore più piccolo tra 3 e 10*

*Val = min ( 3, 10 );*

**max ( x, y )** - Calcola il valore più grande fra x e y.

**abs ( x )** - Calcola il valore assoluto di x.

**constrain ( x, a, b )** - Restituisce il valore "x" solo se è compreso tra i valori "a" e "b".

Se "x" è più piccolo di "a" restituisce "a" se invece è più grande di "b" restituisce "b".

*Esempio: limita il valore di un sensore "sensVal" tra 10 e 150, se "sensVal" è compreso tra i due valori allora lascia "sensVal" invariato.*

*sensVal = constrain ( sensVal, 10, 150 );*

**map ( value, fromLow, fromHigh, toLow, toHigh )** - Cambia il range di un valore.

*Esempio: converti un valore con range 0 - 1000 in un valore con range 0 - 200.*

*Se "variabile" vale 10, "valore" sarà 2.*

*valore = map ( variabile, 0, 1000, 0, 200 );*

**pow ( base, exponent )** - Indicando la base e l'esponente, esegue l'elevazione a potenza di un numero. Funziona anche con una frazione come esponente.

**sqrt ( x )** - Calcola la radice quadrata del numero x.

## FUNZIONI TRIGONOMETRICHE

Le funzioni trigonometriche di base di Arduino

**sin ( rad )** - Calcola il seno di un angolo (in radianti).

**cos ( rad )** - Calcola il coseno di un angolo (in radianti).

**tan ( rad )** - Calcola il valore della tangente di un angolo (in radianti).

## NUMERI CASUALI

Ottenere numeri "random" da Arduino

### **randomSeed (seed)**

La sequenza di numeri casuali di arduino è una catena di numeri, molto lunga ma limitata. Infatti pur essendo i numeri non collegati tra di loro ha la limitazione che l'ordine di questi numeri casuali è sempre lo stesso. Il comando randomSeed avvia la sequenza in un punto ben preciso, infatti al posto di "seed" va inserito un numero.

Se si vuole partire da un punto casuale basta inserire la lettura di un pin analogico non collegato a nulla, l'instabilità su quel pin genererà ad ogni loop un valore di lettura diverso.

**random ( min, max )** - Questa funzione restituisce un numero intero, di valore compreso fra min e max-1. Se min non è specificato il valore minimo restituito sarà 0.

*Esempio:*

*stampa su monitor seriale un numero casuale, compreso tra 0 e 299.*

```
long randNumber;
void setup()
{
  Serial.begin (9600);
  randomSeed ( analogRead (0) );
}

void loop()
{
  randNumber = random (300);
  Serial.println (randNumber);
  delay(50);
}
```

## COMUNICAZIONE SERIALE

Si utilizzano per inviare e ricevere dati tra Arduino e il PC tramite un cavo USB.

**Serial.begin ( speed )** - Serve ad impostare la velocità della comunicazione tra arduino e pc. Generalmente si usa 9600 bps (bit al secondo) ma si possono impostare anche altre velocità, fino a 115.200 bps.

*Esempio: inizializza la porta seriale a 9600 bps*

```
Serial.begin(9600);
```

**Serial.print ( val, format )** - Invia un valore al pc tramite la comunicazione seriale. Il formato indica il sistema numerico utilizzato.

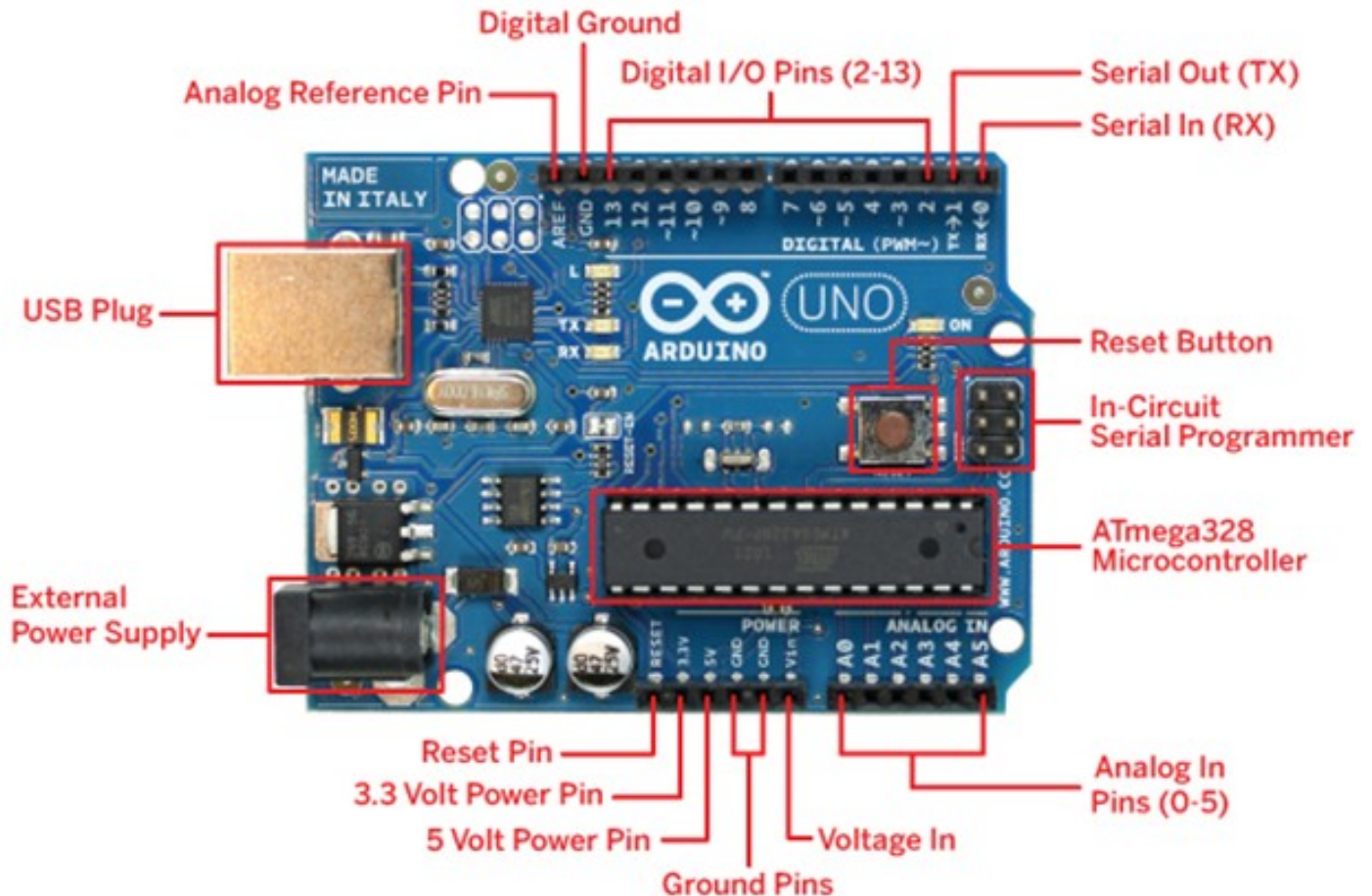
*Esempio:*

```
Serial.print(32); // stampa sul serial monitor 32.
Serial.Print(32, DEC); // stampa 32 in decimale (32)
Serial.Print(32, HEX); // stampa 32 in esadecimale (20)
Serial.Print(32, OCT); // stampa 32 in ottale (40)
Serial.Print(32, BIN); // stampa 32 in binario (100000)
// stampa il valore associato al numero 32 nella tabella ASCII ( carattere spazio)
Serial.Print(32, BYTE);
```

**Serial.println ( val, format )** - Invia un valore al pc con in coda il carattere "A CAPO"



# AUTOMAZIONE CON ARDUINO



## CARATTERISTICHE E LIMITI

I pin digitali possono leggere e generare tensioni di 5V (0V=LOW, 5V=HIGH).

I pin analogici (A0-A5) possono leggere tensioni variabili fra 0-5V con risoluzione 10 bit ( $2^{10} \rightarrow 0-1023$ ).

I pin PWM possono generare *finiti* segnali analogici variabili fra 0-5V con risoluzione 8 bit (0-255).

La corrente erogata dal singolo pin può arrivare a 30-40 mA.

Complessivamente la corrente che si prelevare da una scheda Arduino UNO deve superare I 300-400 mA.

## LIVELLI LOGICI

### TTL (TRANSISTOR-TRANSISTOR LOGIC):

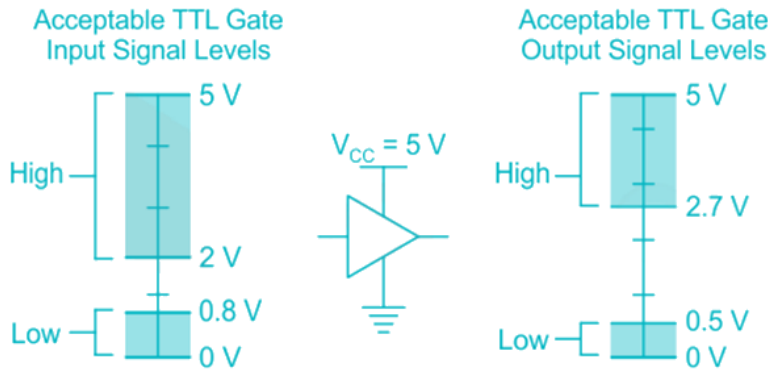
A TTL input signal is defined as "low" when between 0 V and 0.8 V with respect to the ground terminal.

A TTL input signal is defined as "high" when between 2 V and 5 V.

if a voltage signal ranging between 0.8 V and 2.0 V is sent into the input of a TTL gate, there is no certain response from the gate and therefore it is considered "uncertain" (precise logic levels vary slightly between sub-types and by temperature).

TTL outputs are typically restricted to narrower limits of between 0.0 V and 0.4 V for a "low".

TTL outputs are typically restricted to narrower limits of between 2.4 V and 5 V for a "high", providing at least 0.4 V of noise immunity.

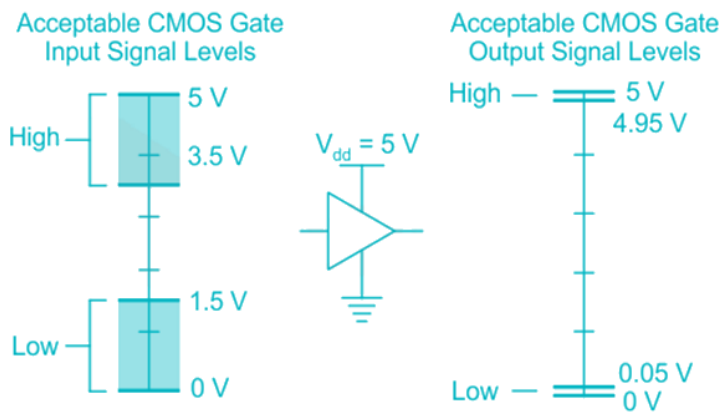


### CMOS (COMPLEMENTARY METAL-OXIDE-SEMICONDUCTOR)

For a CMOS gate operating at a power supply voltage of 5 volts,

The acceptable input signal voltages range from 0 volts to 1.5 volts for a "low" logic state and 3.5 volts to 5 volts for a "high" logic state.

Acceptable output signal voltages range from 0 volts to 0.05 volts for a "low" logic state, and 4.95 volts to 5 volts for a "high" logic state:



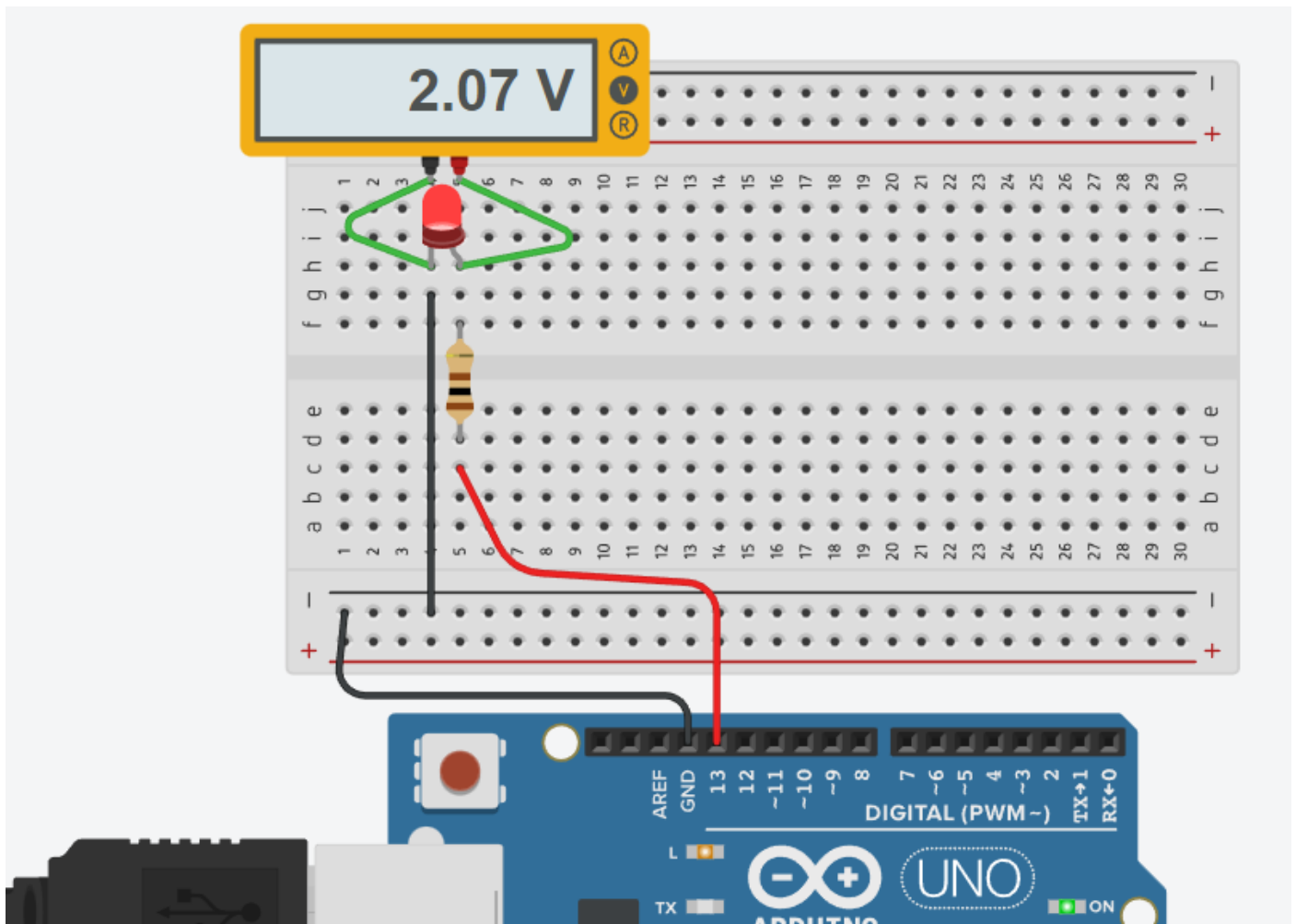
Specifications	TTL	ECL	CMOS
FAN IN	12-14	> 10	> 10
FAN OUT	10	25	50
power dissipation (mW)	10	75	0.001
Noise margin	0.5	0.16(least)	1.5 (highest)
Propagation delay(ns)	10	>3	15
Noise immunity	very good	good	excellent

## DIODO LED

Il Led è un componente elettronico costituito da una giunzione P-N con arseniuro di gallio o con fosforo di gallio che emette luce quando attraversato da una corrente compresa tra 10 e 30mA (dall'anodo al catodo).

La caduta di tensione ai capi del Led è di circa 2V (dipende dal colore del Led).

Il circuito sottostante accende e spegne il Led con una frequenza di 1 sec. .

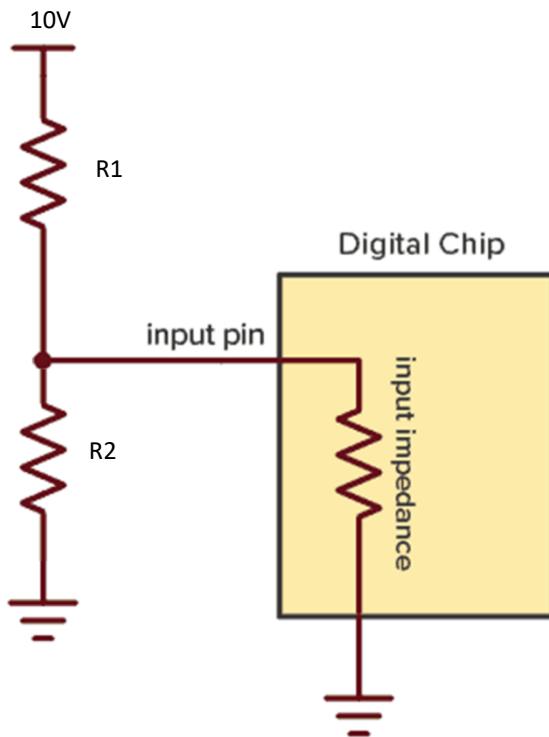


### CODICE

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Ai pin di Arduino non è possibile applicare tensioni superiori a 5V (si danneggerebbero).  
Di conseguenza è necessario ridurre una tensione >5V mediante un partitore di tensione.



I pin dei MCU presentano impedenze oltre a 1M ohm

Fissando  $R1=R2= 10k$  sulla R2 otteniamo 5V. Però in parallelo alla R2 abbiamo l'impedenza del PIN.

### **Cosa cambia?**

In pratica nulla perché 1M in parallelo a 1k danno una resistenza di ancora 1K e quindi all'ingresso del PIN rileviamo (mediante altro circuito interno) 5V.

### **Calcoli:**

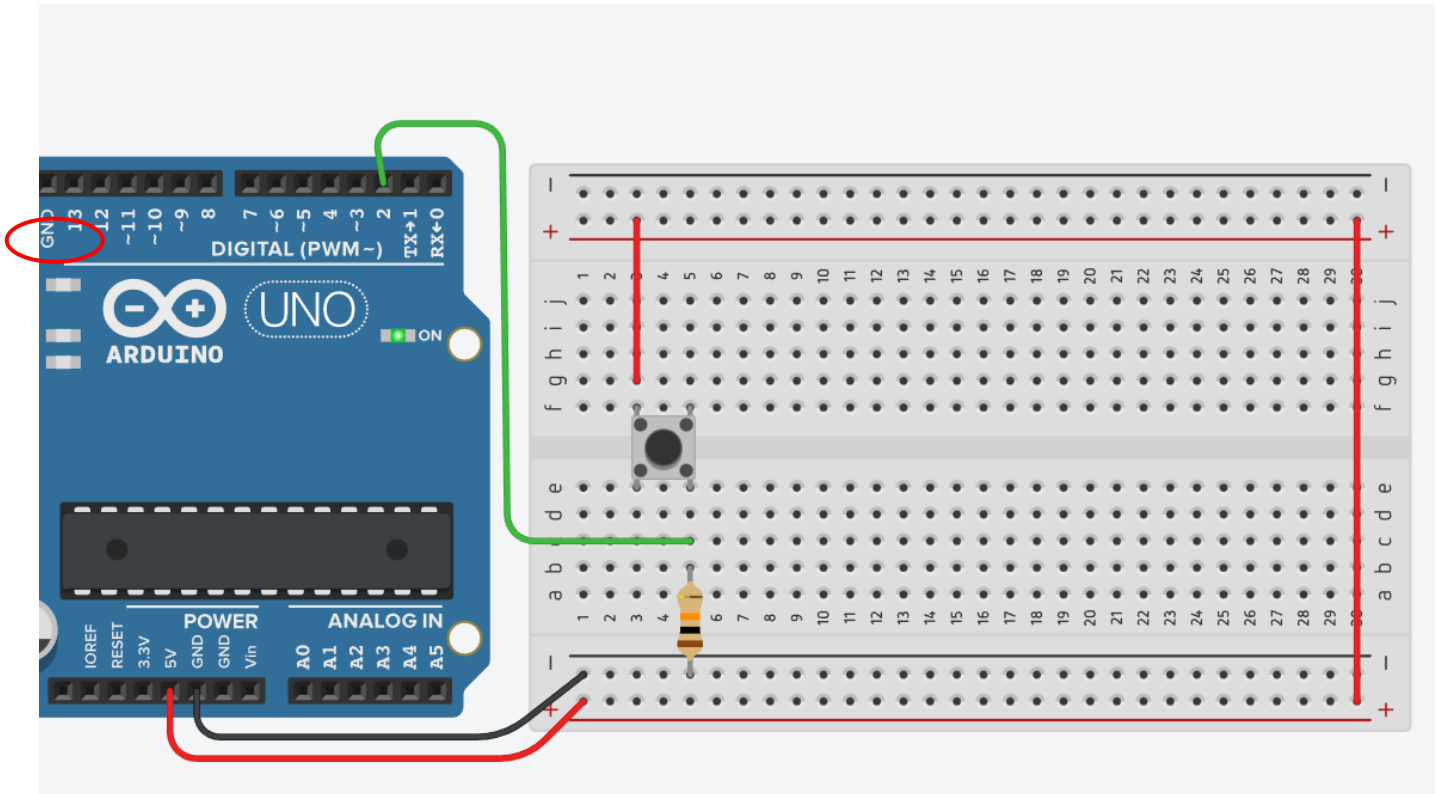
$$R_{eq} = (1/R1 + 1/impedenza)^{-1} = (1/1000 + 1/1000000)^{-1} = 999 \text{ ohm}$$

$$V_{out} = V_{cc} * 999 / (1000 + 999) = 4,997 \text{ V}$$

## PULSANTE (PUSH BUTTON)

*E' un pulsante che non mantiene lo stato se viene rilasciato.*

Lo schema mostra come utilizzare il pulsante in modalità NA (normalmente aperto) per accendere il LED interno di Arduino. La resistenza serve a proteggere il pin 2 nel caso in cui non fosse impostato come INPUT.



### CODICE

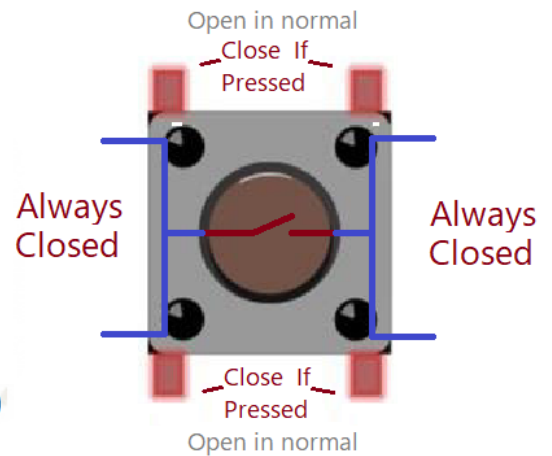
```
int buttonState = 0;

void setup()
{
  pinMode(2, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  // read the state of the pushbutton value
  buttonState = digitalRead(2);
  // check if pushbutton is pressed.
  if (buttonState == HIGH) {
    // turn LED on
    digitalWrite(LED_BUILTIN, HIGH);
  } else {
    // turn LED off
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```



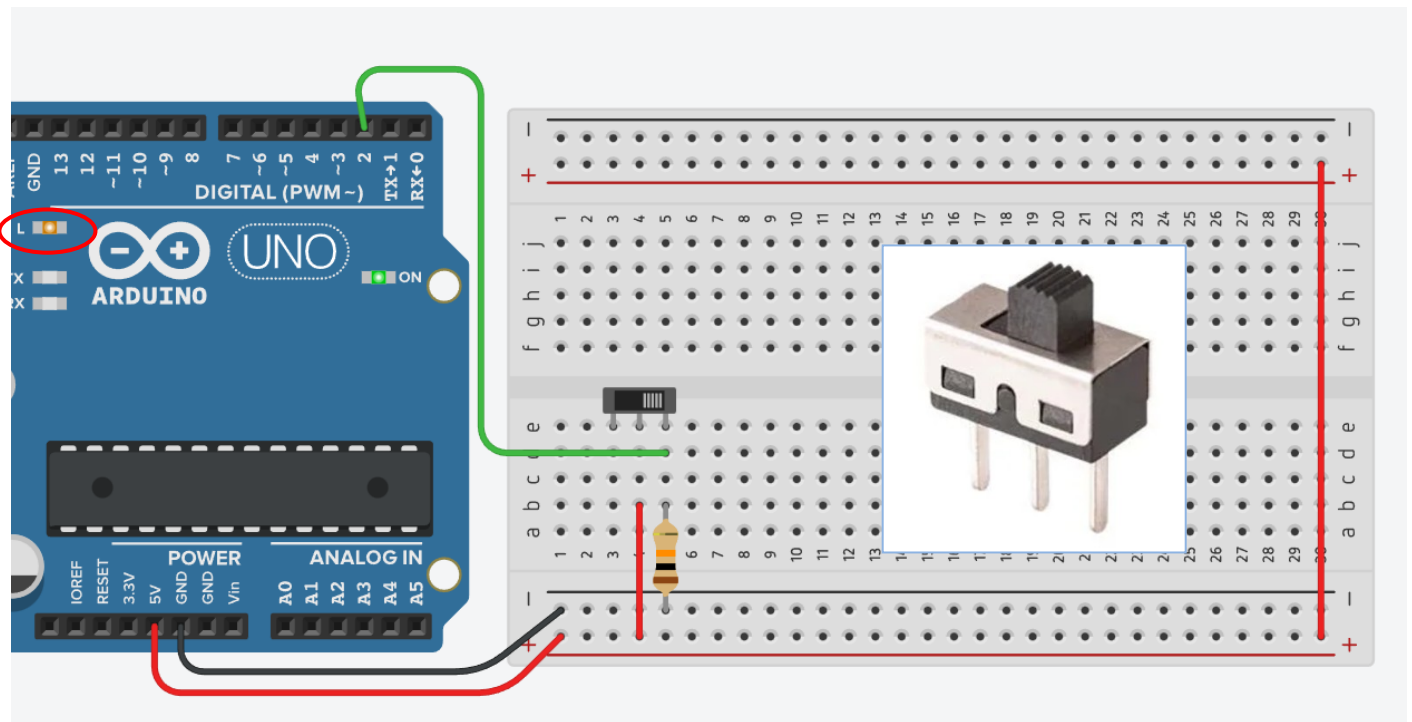
Push Button (4 Pins)



## INTERRUTTORE (SLIDER)

E' un pulsante che mantiene lo stato se viene rilasciato.

Lo schema mostra come utilizzare il pulsante in modalità NA (normalmente aperto) per accendere il LED interno presente sulle schede Arduino. La resistenza in serie all'interruttore è fondamentale per limitare la corrente in uscita da Arduino quando l'interruttore viene chiuso.



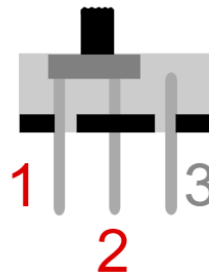
### CODICE

```
int buttonState = 0;

void setup()
{
  pinMode(2, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}

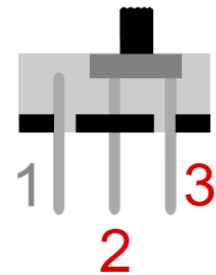
void loop()
{
  // read the state of the pushbutton value
  buttonState = digitalRead(2);
  // check if pushbutton is pressed.
  if (buttonState == HIGH) {
    // turn LED on
    digitalWrite(LED_BUILTIN, HIGH);
  } else {
    // turn LED off
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```

Left  
Position



APERTO  
PIN Arduino  
legge massa

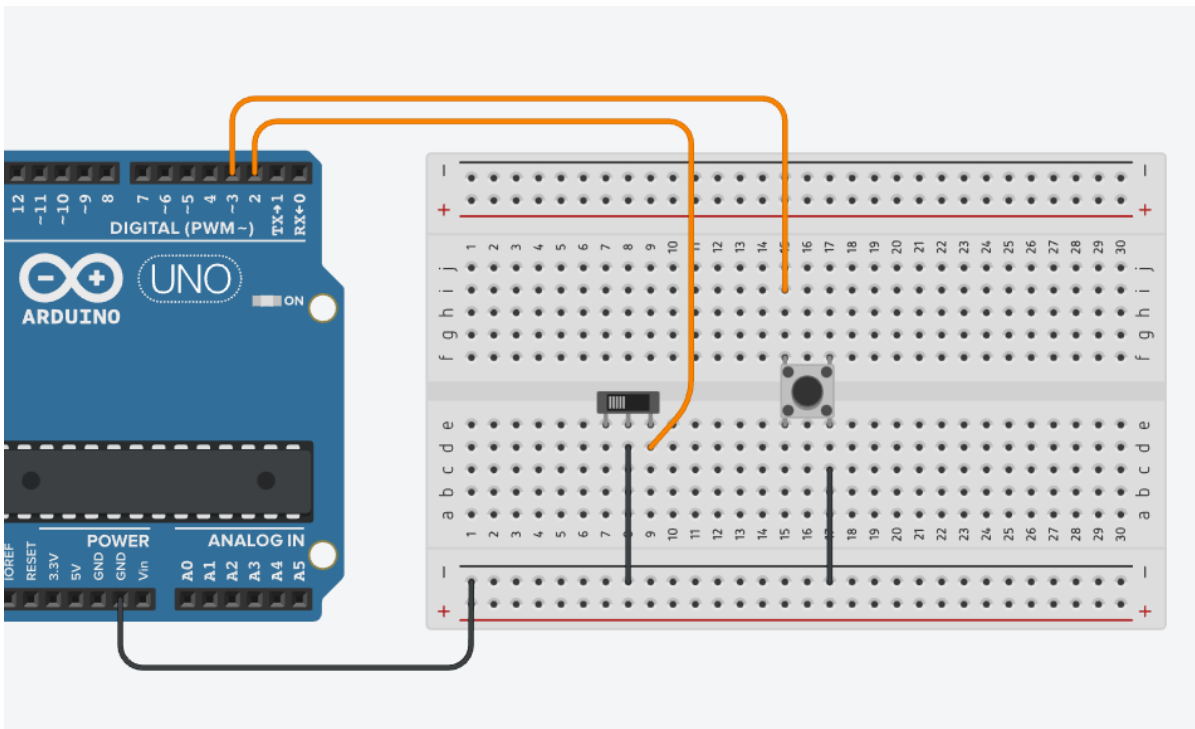
Right  
Position



CHIUSO  
PIN Arduino  
legge 5V

## INTERRUTTORE E PULSANTE IN MODALITA' PULL-UP (LOGICA INVERSA)

Si evita l'utilizzo di resistenze esterne usando quelle interne di Arduino connesse ai PIN digitali.  
Bisogna prestare attenzione a leggere l'ingresso: HIGH quando l'interruttore è aperto e LOW quando è chiuso.



### CODICE

```
void setup() {
  Serial.begin(9600);
  //configure pin 2-3 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(LED_BUILTIN, OUTPUT); // PIN13
}

void loop() {
  int button1 = digitalRead(2);
  Serial.println(button1);

  int button2 = digitalRead(3);
  Serial.println(button2);

  // Con il pullup la logica è invertita, significa il pulsante
  //va HIGH quando è aperto e LOW quando viene premuto.
  // Attivare il pin 13 quando il pulsante è premuto e spento quando non lo è:

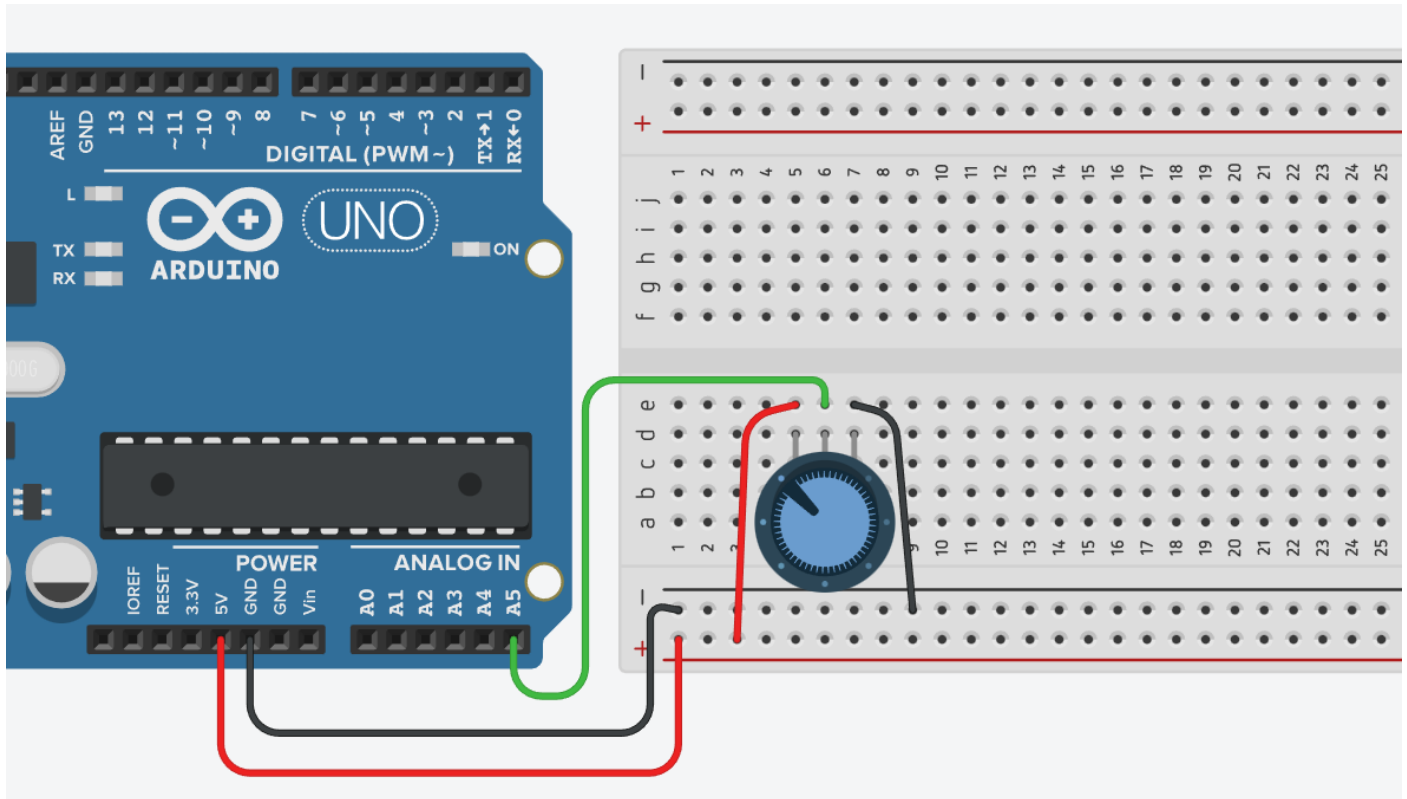
  if (button1 == LOW || button2 == LOW) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else {
    digitalWrite(LED_BUILTIN, LOW);
  }

  delay(100);
}
```

## POTENZIOMETRO

Il potenziometro è un dispositivo elettrico equivalente ad un partitore di tensione resistivo variabile (cioè a due resistori collegati in serie, aventi la somma dei due valori di resistenza costante, ma di cui può variare il valore relativo).

Può essere usato per generare un segnale di controllo analogico (0-5V) per regolare degli attuatori (es. luminosità di un LED, velocità di un motore ecc.).



### CODICE

```
int sensorValue = 0;

void setup()
{
  pinMode(A5, INPUT);
  Serial.begin(9600); // ATTIVA LA COMUNICAZIONE SERIALE
}

void loop()
{
  // read the input on analog pin 0:
  sensorValue = analogRead(A5);
  // print out the value you read:
  Serial.print("Valore analogico: ");
  Serial.println(sensorValue); // VA A CAPO
  Serial.print("Volt: ");
  Serial.println(sensorValue * 5.0/1024.0);

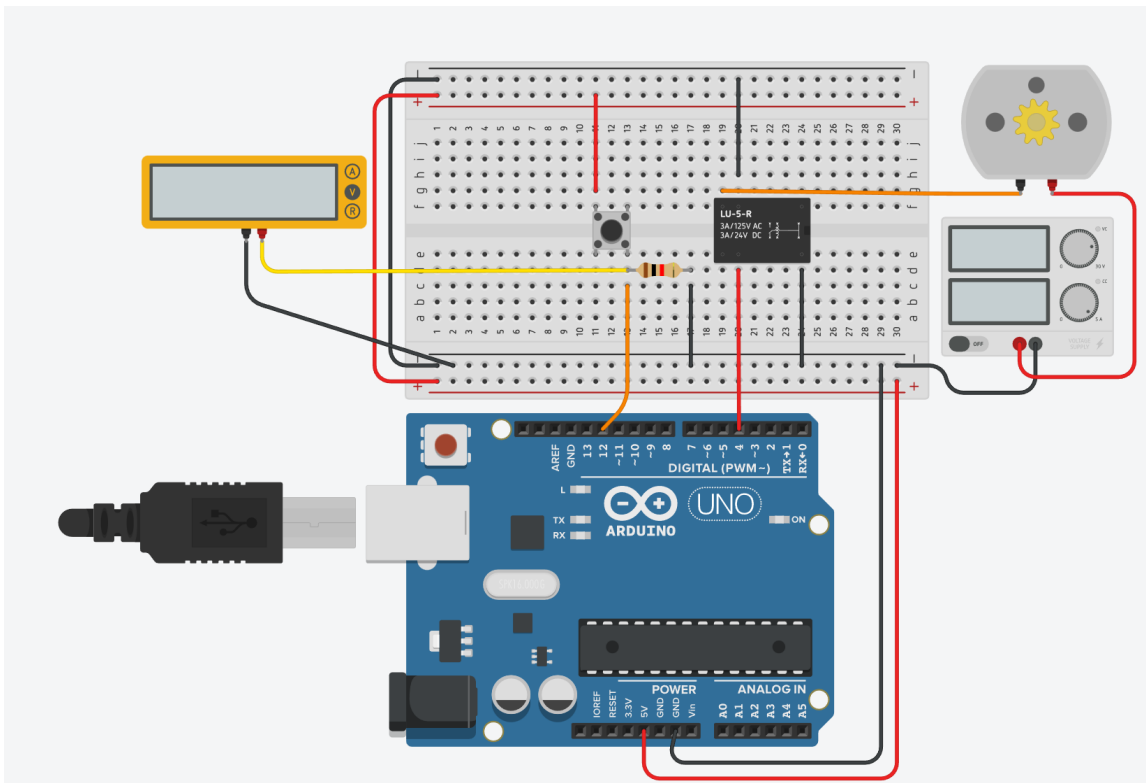
  delay(1000); // Delay a little bit to improve simulation performance
}
```

### MONITOR SERIALE

```
Valore analogico: 1023

Volt: 5.00
Valore analogico: 1023
```

Attivare un motore CC tramite un relè quando viene premuto un pulsante di START.



### CODICE

```
int pinAvvio = 12;
int pinRele= 4;
int stato_bottone=LOW;

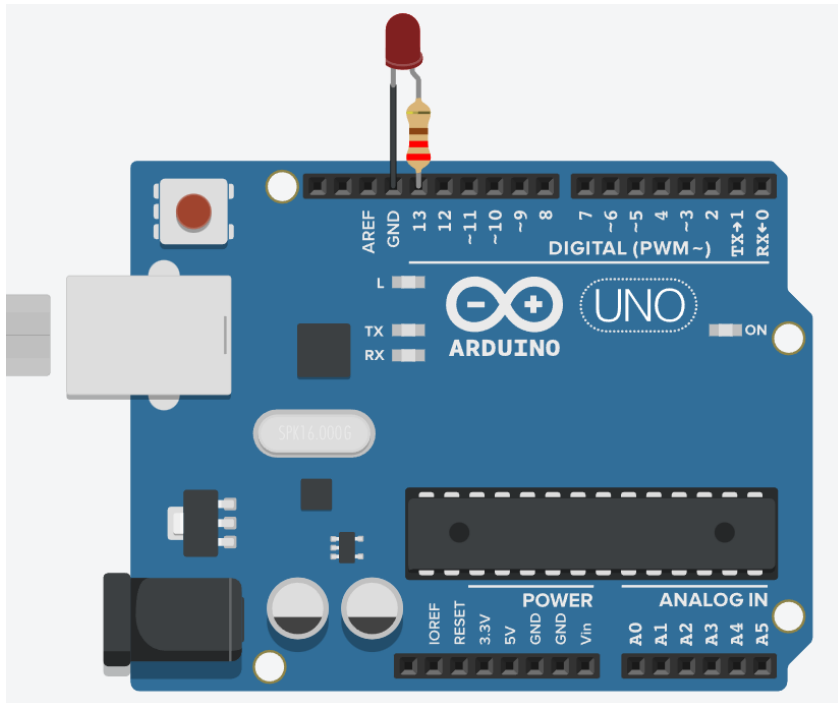
void setup()
{
  pinMode(pinAvvio, INPUT);
  pinMode(pinRele, OUTPUT);
}

void loop()
{
  stato_bottone= digitalRead(pinAvvio);
  if (stato_bottone ==HIGH)
  {
    digitalWrite(pinRele, HIGH);
  }
  else
  {
    digitalWrite(pinRele, LOW);
  }
}
```

## VALUTAZIONE DEL TEMPO TRASCORSO CON ARDUINO MILLISECOND()

Per valutare il tempo trascorso in Arduino si deve impiegare la funzione “millis()” che ritorna il numero di millisecondi trascorsi dall’accensione di arduino. Questo tempo va salvato in una variabile di tipo **long** ( $2^{32}$  → da -2.147.483.648 a 2.147.483.647).

Bisogna prestare attenzione al fatto che “millis()” non torna l’ora attuale ma l’intervallo di tempo trascorso dall’accensione e quindi va usata per valutare se è trascorso un determinato intervallo di tempo!



### CODICE

```
const int ledPin = 13;
int ledState = 0;
long previousMillis = 0; // ultimo tempo di aggiornamento del LED
long dt = 0;

// const indica che non varia
const long interval = 1000; // intervallo di blink (milliseconds)

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  long currentMillis = millis();
  dt= currentMillis – previousMillis;

  if (dt >= interval) {
    previousMillis = currentMillis; // save the last time you blinked the LED

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) { ledState = HIGH; }
    else { ledState = LOW; }

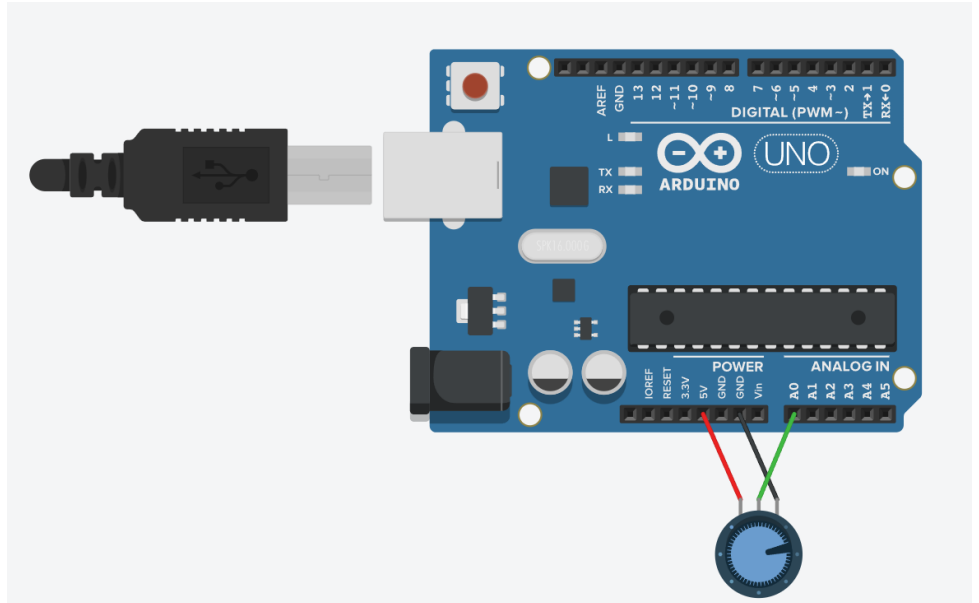
    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }
}
```

In generale quando si deve acquisire un dato analogico da un sensore è preferibile effettuare più letture a breve distanza una dall'altra (il breve dipende dal tempo di risposta del sensore) in modo da mediare eventuali errori di lettura o variazioni indesiderate dovute a disturbi esterni. Anche in questo caso è necessario utilizzare la funzione "millis()".

Nell'esempio sottostante viene effettuata la lettura del potenziometro per 5 volte ad intervalli di 1 sec. Al termine delle 5 letture (si utilizza una variabile contatore "n" per tenere traccia delle letture effettuate) viene calcolata la media che viene stampata a schermo.

Da notare che ad ogni intervallo (dentro il blocco "if ( dt > 1000) { " va incrementato il contatore e aggiornato il tempo della lettura fatta.

Dopo le 5 letture va resettato sia il contatore che il valore della media!



Un potenziometro che permette di regolare la tensione sul pin A0 da 0-5V è il componente adatto a simulare un sensore analogico con uscita in tensione.

### CODICE

```
int sensorValue = 0;
int sensorValueM = 0;
int n=0;
long t0=0;
long dt=0;

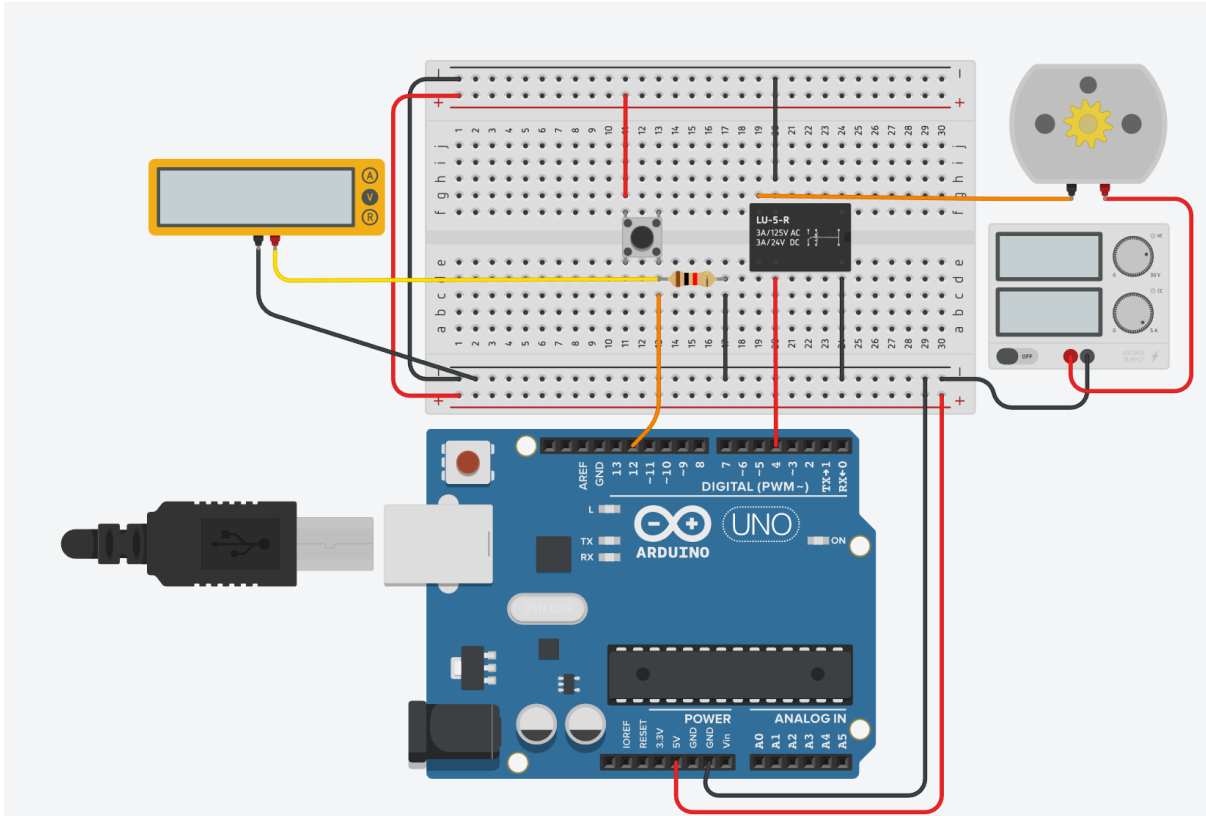
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  dt = millis() - t0;
  // leggo sensore ogni 1 sec
  if ( dt > 1000) {
    sensorValue = analogRead(A0);
    sensorValueM = sensorValueM + sensorValue;
    Serial.print("Sensore "); Serial.println(sensorValue);
    n= n+1; // contatore
    t0= millis(); // tempo attuale
  }

  if (n>=5) {
    sensorValueM = sensorValueM / 5;
    Serial.print("Media "); Serial.println(sensorValueM);
    n=0;
    sensorValueM = 0;
  }
  delay(10); // 10ms
}
```

## ESERCIZIO MILLIS()

Attivare un motore DC tramite un relè quando viene premuto per almeno 3 secondi un pulsante di START.



### CODICE

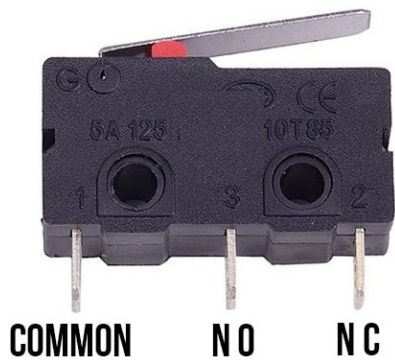
```
int stato_bottone=LOW;
long tempo;
long counter=0;
long delta_t;

void setup()
{
  pinMode(4, OUTPUT); // pin comando relè
  pinMode(12, INPUT); // pin stato bottone
}

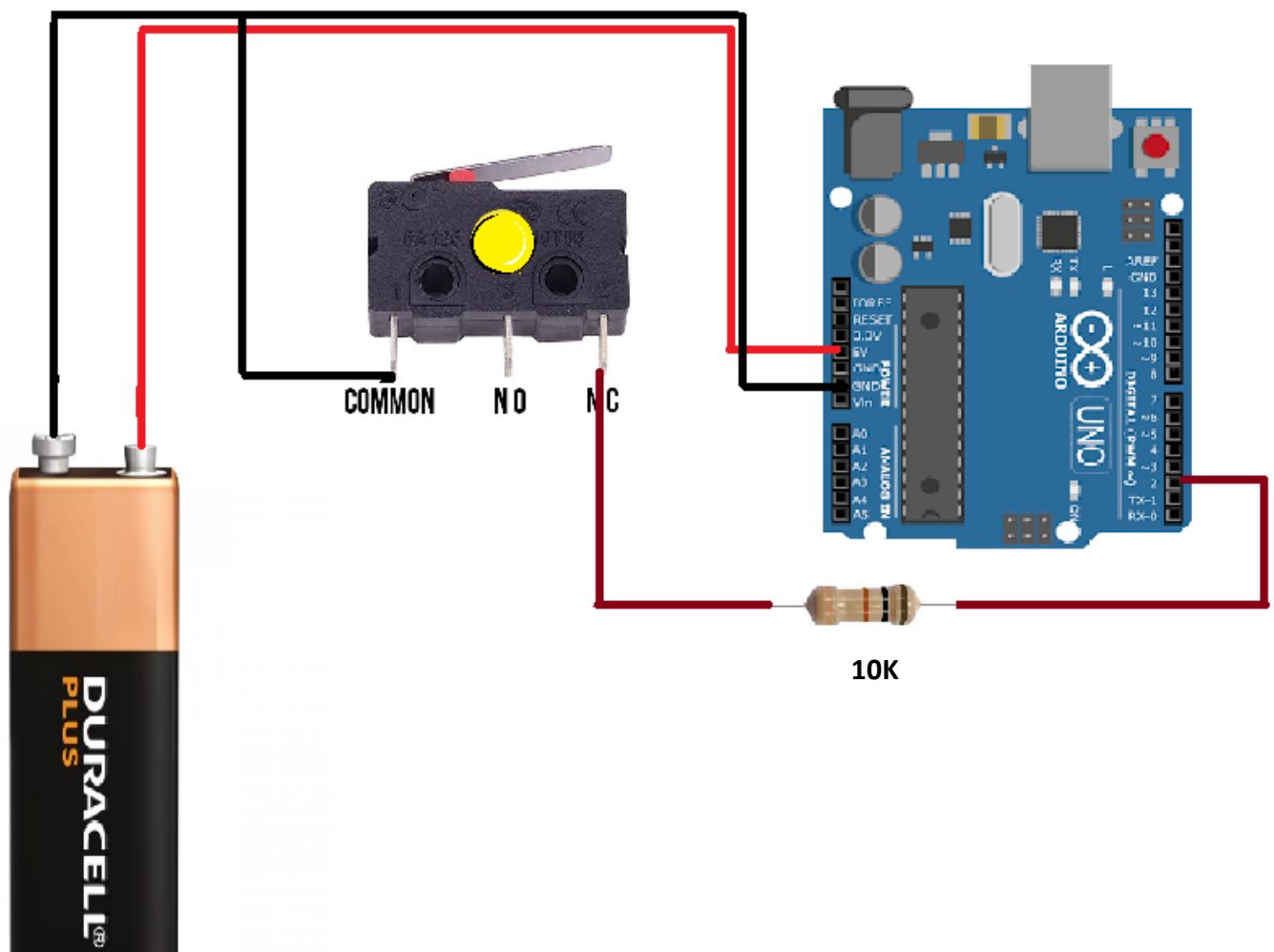
void loop()
{
  stato_bottone= digitalRead(12);
  if (stato_bottone ==HIGH){
    counter++;
    if (counter==1) {tempo = millis();}
    delta_t = millis() - tempo;
    if (delta_t>=3000){ digitalWrite(4, HIGH); }
  }
  else
  {
    tempo = millis();
    digitalWrite(4, LOW);
    counter=0;
  }
}
```

Il finecorsa meccanico ha tre pin:

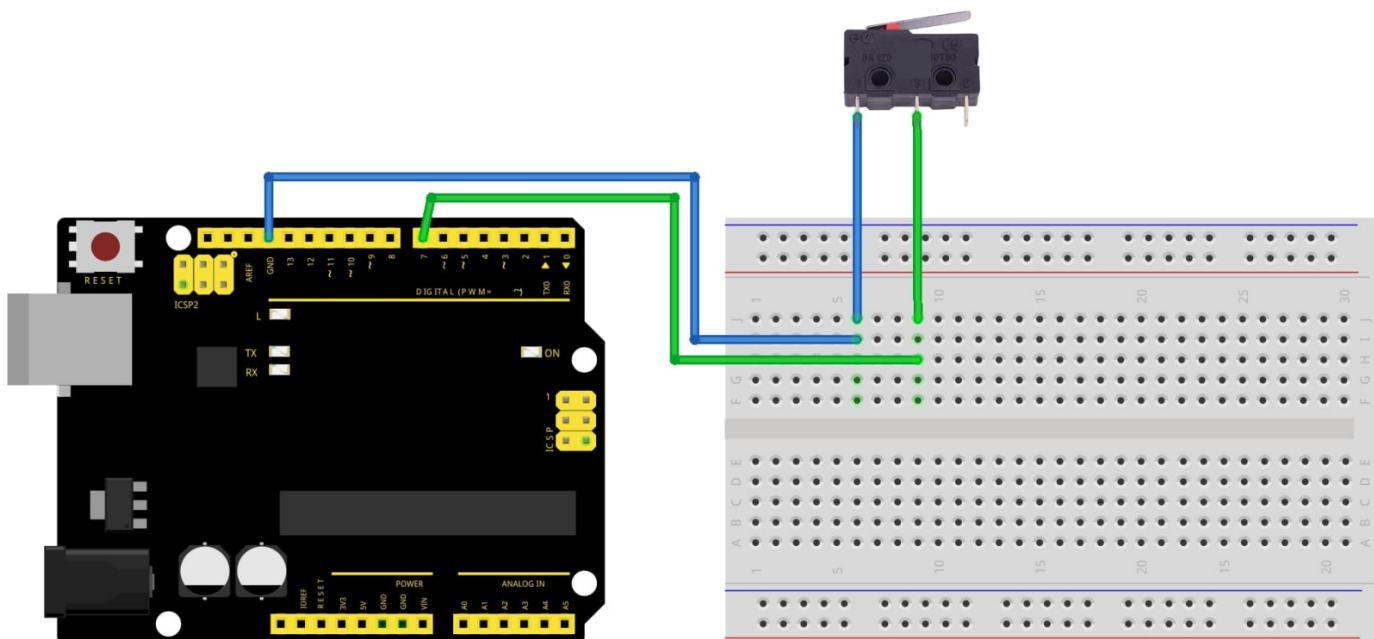
- Comune – Come suggerisce il nome, è il pin comune tra i pin normalmente aperti (NO) e normalmente chiusi (NC).
- Normalmente aperto : normalmente aperto significa che non c'è contatto tra questo pin e il pin comune finché non viene premuto/attivato il finecorsa.
- Normalmente chiuso – Normalmente chiuso significa che c'è sempre contatto tra questo pin e il pin comune. Quando il finecorsa viene premuto/attivato, il contatto viene interrotto.



Il finecorsa è fondamentalmente un interruttore unipolare a due vie (SPDT).



## Schema elettrico dell'interruttore di finecorsa Arduino - Normalmente aperto (è la configurazione più comune).



## Codice Arduino per interruttore di finecorsa normalmente aperto

```
#define LIMIT_SWITCH_PIN 7

void setup() {
  Serial.begin(9600);
  pinMode(LIMIT_SWITCH_PIN, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(LIMIT_SWITCH_PIN) == HIGH)
  {
    Serial.println("Activated!");
  }

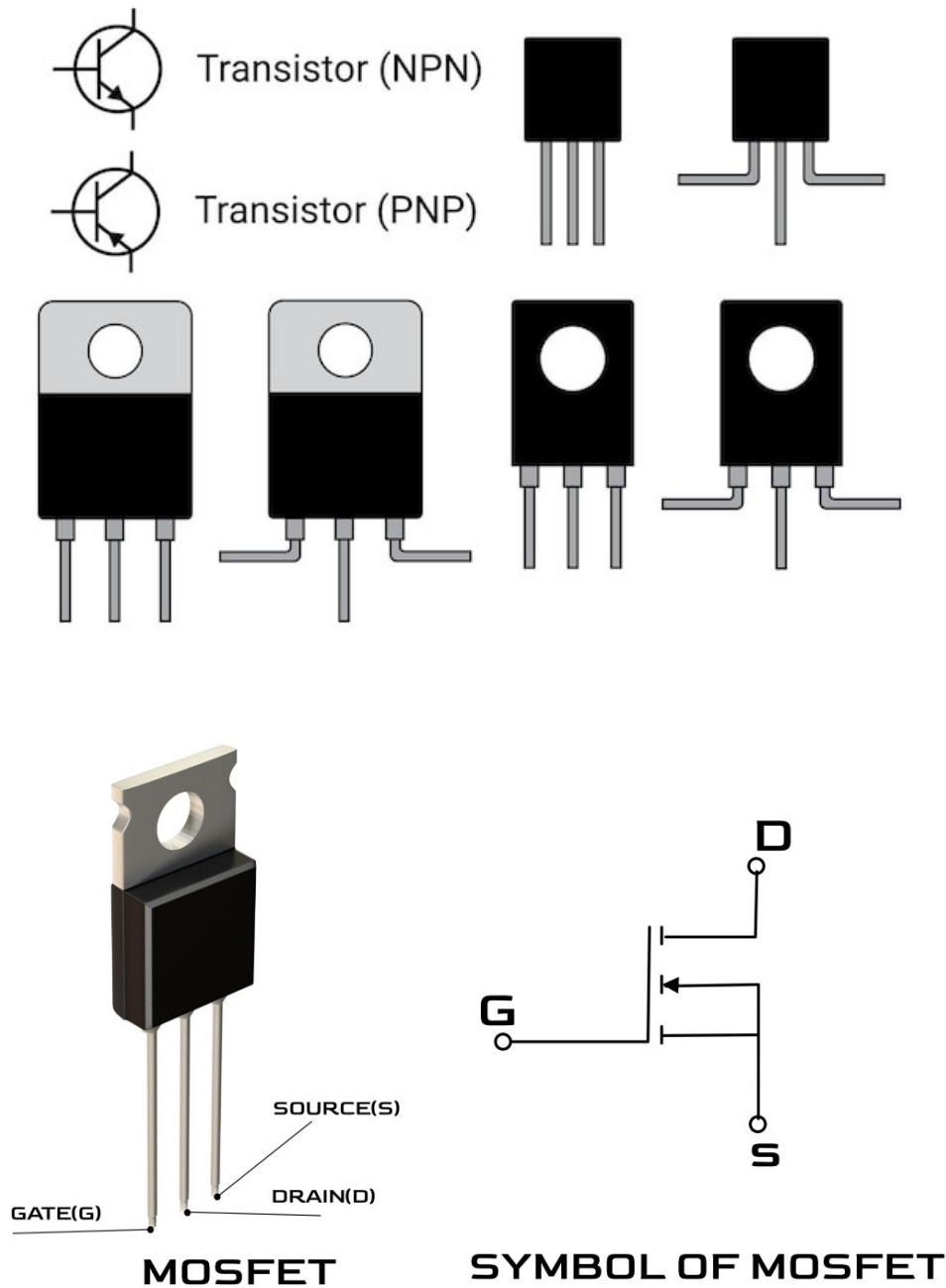
  else
  {
    Serial.println("Not activated.");
  }

  delay(100);
}
```

## IL TRANSISTOR

I transistor si possono dividere in due grandi famiglie:

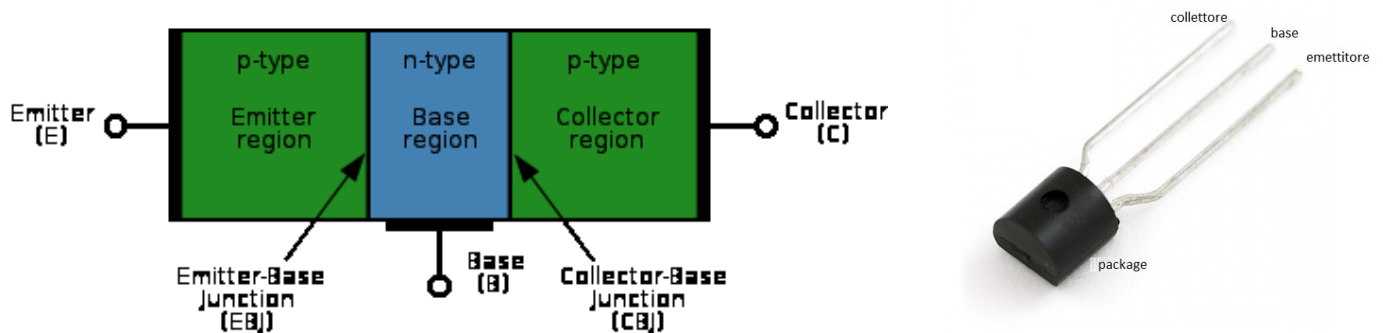
1. la famiglia dei "BJT: bipolar junction transistor"
2. la famiglia dei "MOSFET: metal-oxide-semiconductor-field-effect-transistor" comunemente detto MOS.



## TRANSISTOR BJT (BIPOLAR JUNCTION TRANSISTOR)

La differenza principale tra le due famiglie risiede nella tecnologia con cui vengono realizzati.

Il BJT viene implementato tramite una giunzione bipolare costituita da silicio drogato in maniere differenti in tre zone dette regione di base, regione di collettore e regione di emettitore.

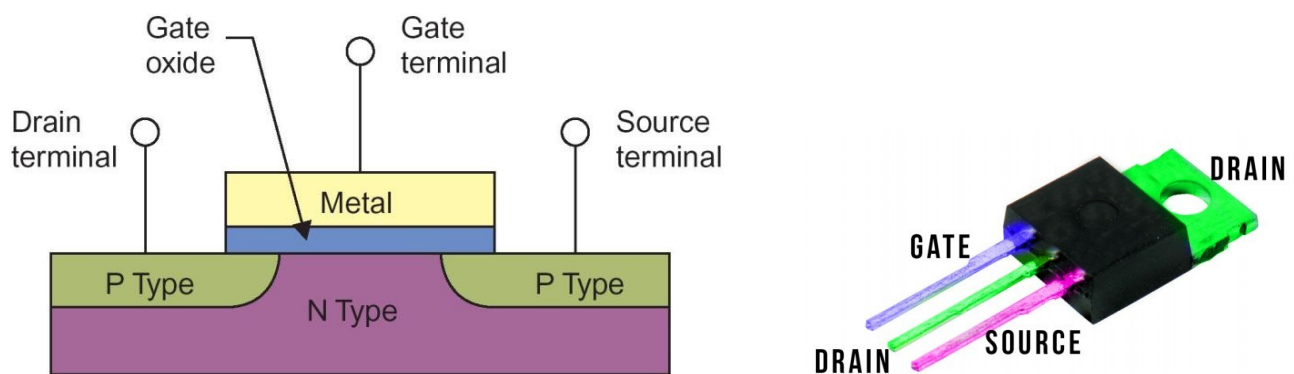


I terminali delle tre regioni vengono metallizzati per creare i contatti che successivamente usciranno dal package del componente per poter essere saldati sulla scheda elettronica.

L'applicazione di tensioni adeguate sui terminali di collettore ed emettitore e l'imposizione di un flusso di corrente nella regione di base fanno funzionare il BJT in una delle sue tre regioni di lavoro: saturazione e interdizione nel caso si voglia farlo lavorare come interruttore, zona lineare nel caso si voglia realizzare un amplificatore.

## TRANSISTOR MOS (MOSFET)

Il MOS (MOSFET) invece è strutturato in maniera differente tramite tre strati: lo strato di metallo, lo strato di ossido e quello di semiconduttore.



Il principio fisico su cui si basa è diverso rispetto a quello del BJT. I terminali sono sempre tre, cambiano solo i nomi: qui abbiamo gate, drain e source.

Come dice il nome stesso del transistor il funzionamento si basa sul cosiddetto effetto di campo che crea un canale in cui possono fluire gli elettroni tra source e drain, quando ai terminali sono applicate le corrette tensioni.

Anche con questo dispositivo le regioni di lavoro sono tre: interdizione e triodo che determinano il funzionamento come interruttore e la saturazione che determina il funzionamento come amplificatore.

## BJT VS MOS (MOSFET): CARATTERISTICHE TECNICHE

Fatta una sintetica panoramica sulle principali tecnologie costruttive, possiamo addentrarci nelle caratteristiche tecniche delle due famiglie di transistor per capire quali siano gli aspetti di maggiore rilievo che fanno pendere l'ago della bilancia dalla parte dei BJT o dalla parte dei MOS.

Nell'ambito digitale, ad esempio dei microprocessori e dei circuiti integrati regna sovrano senza rivali il MOSFET.

La sua caratteristica di essere auto-isolato (auto-isolato: dispositivo che può essere implementato nella stessa regione di silicio assieme ad altri componenti ad esso uguali, senza interferire con il funzionamento dei dispositivi adiacenti, grazie ad un isolamento elettrico dato dal processo produttivo) rende molto facile la connessione in serie o in parallelo di questi dispositivi senza strati di silicio aggiuntivi, operazione che con i BJT non è possibile senza l'aggiunta di strati di silicio tra un transistor e l'altro. Questo riduce notevolmente i costi e la complessità del progetto rendendo il MOS il dispositivo perfetto.

Quando si comincia a parlare di commutazioni, trasferimento di potenza e quindi, di convertitori switching di potenza è opportuno riconsiderare il BJT. Infatti per il trasferimento di potenze maggiori di 1 kW e correnti superiori ai 200 A il MOSFET lascia il posto al BJT (e ad altri componenti ...).

Il BJT infatti regge potenze fino a 2 kW e correnti fino a 500 A. Se per le piccole potenze c'è bisogno di frequenze di commutazione elevate il MOS si rivela un'ottima scelta perché il BJT non sostiene elevate frequenze di commutazione.

Inoltre nei convertitori switching la dimensione dei componenti, come induttori e condensatori, risulta inversamente proporzionale alla frequenza. Per cui se in fase di progetto si decide di mantenere contenute le dimensioni dei componenti aumentando la frequenza di commutazione, con il MOS possiamo andare fino a frequenze di qualche MHz contro i 100 kHz scarsi del BJT.

Altro aspetto in cui il MOSFET vince la battaglia con il BJT è il metodo di controllo.

- nel BJT il circuito di pilotaggio deve essere in grado di dare corrente costante nella base del BJT, operazione non sempre facile soprattutto in fase di commutazione o quando si pilotano carichi che richiedono grandi quantità di corrente.
- il MOS a sua volta deve essere pilotato con una tensione di gate costante molto più facile da ottenere sia in fase di commutazione sia in fase di pilotaggio di carichi che richiedono grandi correnti.

## QUANDO USARE I TRANSISTOR BJT E MOSFET

Nell'elettronica digitale, nei microprocessori e nei circuiti integrati si utilizza il MOSFET.

La sua principale caratteristica è quella di essere auto-isolato: dispositivo che può essere implementato nella stessa regione di silicio assieme ad altri componenti ad esso uguali, senza interferire con il funzionamento dei dispositivi adiacenti, grazie ad un isolamento elettrico dato dal processo produttivo.

Questo facilita la connessione in serie o in parallelo di questi dispositivi senza strati di silicio aggiuntivi, operazione che con i BJT non è possibile senza l'aggiunta di strati di silicio tra un transistor e l'altro. La cosa riduce notevolmente i costi e la complessità del progetto rendendo il MOSFET il transistor utilizzato nella realizzazione dei microchip.

Se abbiamo bisogno di commutazioni, trasferimento di potenza e quindi, di convertitori switch di potenza è ben utilizzare il transistor BJT. Infatti per il trasferimento di potenze maggiori di 1 kW e correnti superiori ai 200 A il transistor BJT è da preferire al transistor MOSFET. Il BJT riesce a mantenere potenze fino a 2 kW e correnti fino a 500 A. Tuttavia considerando che per le piccole potenze c'è bisogno di frequenze di commutazione elevate il MOS è di nuovo un'ottima scelta rispetto al transistor BJT. Inoltre quando spento non permette alla corrente di scorrere, e ciò si traduce nella riduzione della potenza dissipata. Tale dispositivo fornisce un considerevole risparmio energetico e previene il surriscaldamento del circuito, una delle principali problematiche dei circuiti integrati.

Riassumendo:

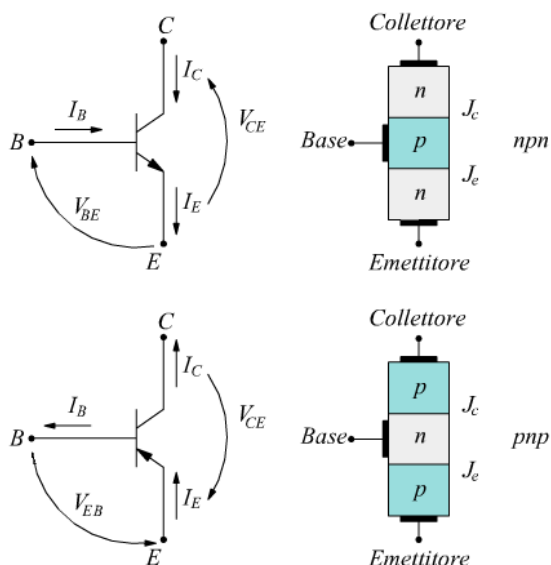
- il MOSFET può gestire i cambi di stato con frequenze nell'ordine dei MHz,
- il BJT riesce a gestire frequenze di commutazione al di sotto dei 100 kHz.

## IL TRANSISTOR BJT

Il transistor a giunzione bipolare (abbreviazione comunemente utilizzata BJT, dall'inglese bipolar junction transistor) è una tipologia di transistor largamente usata nel campo dell'elettronica analogica principalmente come amplificatore di corrente e interruttore elettronico.

Esso è composto da tre strati di materiale semiconduttore drogato (drogaggio: aggiunta al semiconduttore puro ("intrinseco") di piccole percentuali di atomi non facenti parte del semiconduttore stesso, es. fosforo e arsenico per giunzione "n" e boro e alluminio per giunzione "p", allo scopo di modificare le proprietà elettroniche del materiale), solitamente silicio, in cui lo strato centrale ha drogaggio opposto agli altri due, in modo da formare una doppia giunzione p-n.

Ogni strato è un terminale. Quello centrale prende il nome di *base*, quelli esterni sono detti *collettore* ed *emettitore*.



### DIMENSIONAMENTO DI MASSIMA DELLA R<sub>b</sub> PER ATTIVARE IL TRANSISTOR

Parametro fondamentale di un transistor è il suo h<sub>FE</sub>, cioè il guadagno di corrente o fattore di amplificazione:

$$I_c = h_{FE} \cdot I_b$$

Sigla BJT	tipo	hFE(min)	hFE(MAX)	V <sub>ce</sub>	I <sub>c</sub>
BC107	NPN	110	800	5V	2mA
BC107A	NPN	110	220	5V	2mA
BC107B	NPN	200	450	5V	2mA
BC107C	NPN	420	800	5V	2mA
BC337	NPN	100	600	1V	50mA
BC327	PNP	100	600	1V	50mA

Nei dimensionamenti conviene tutelarsi prendendo il valore minimo.

La corrente I<sub>c</sub> gestibile con un semplice BJT è dell'ordine delle centinaia di milli-ampere.

Nota la I<sub>c</sub> necessaria all'utilizzatore (motore, lampada ecc.) si calcola la I<sub>b</sub> attraverso l'h<sub>FE</sub> del transistor scelto (che deve essere in grado di gestire la I<sub>c</sub> richiesta):  $I_b = I_c / h_{FE}$ .

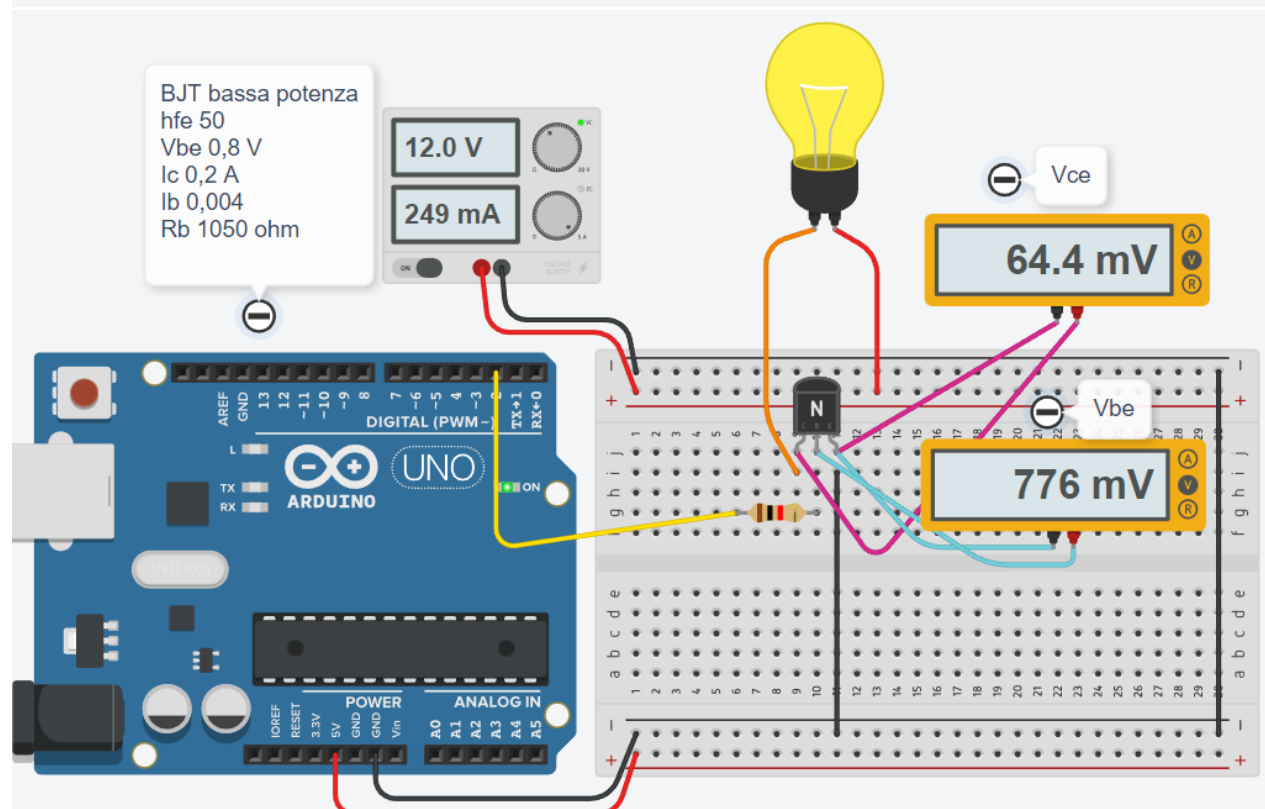
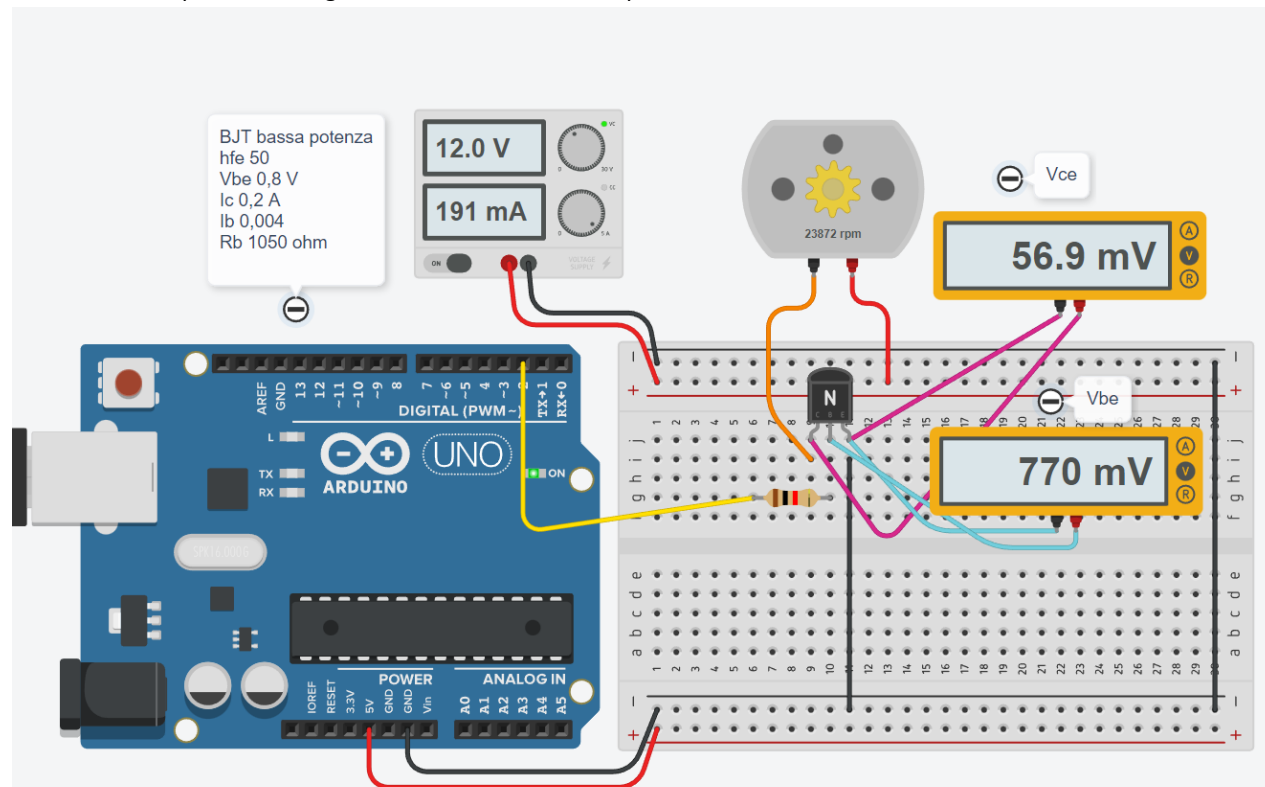
Ipotesizzando una caduta di tensione tipica V<sub>be</sub> del transistor di 0.8V (recuperabile dalla scheda tecnica del BJT) si ricava la R<sub>b</sub> necessaria a limitare la corrente di base:  $R_b = (V_{micro} - 0.8) / I_b$  ohm.

Ad esempio per un motore che assorbe 200mA con un BJT che ha h<sub>FE</sub>=50 la R<sub>b</sub> comandata da Arduino vale:

$$I_b = 0.2 / 50 \quad R_b = (5 - 0.8) / (0.2 / 50) = 1050 \text{ ohm.}$$

## ESERCIZIO BJT

Avviare una lampada di emergenza e un motore di bassa potenza a 12V tramite un transistor BJT.



### COMPITO

1. Modificare il circuito per avviare un motore di bassa potenza a 5V.
2. Modificare il circuito per avviare un motore di bassa potenza a 24V.

## TRANSISTOR PER PILOTARE RELE' DI POTENZA (TENSIONE >5V)

Nel caso in cui l'utilizzatore da attivare sia collegato alla rete elettrica a 220V si può utilizzare un relè di potenza (generalmente con una tensione bobina superiore a 5V) attivato da un transistor pilotato dal microcontrollore a 5V (la corrente  $I_c$  richiesta dalla bobina del relè non può essere fornita direttamente dal microcontrollore).

Il transistor verrà pilotato dal microcontrollore a 5V sulla resistenza di base dove è richiesta una corrente  $I_b$  molto inferiore a quella di collettore necessaria ad attivare il relè ( $I_b = I_c / hFE$ ).

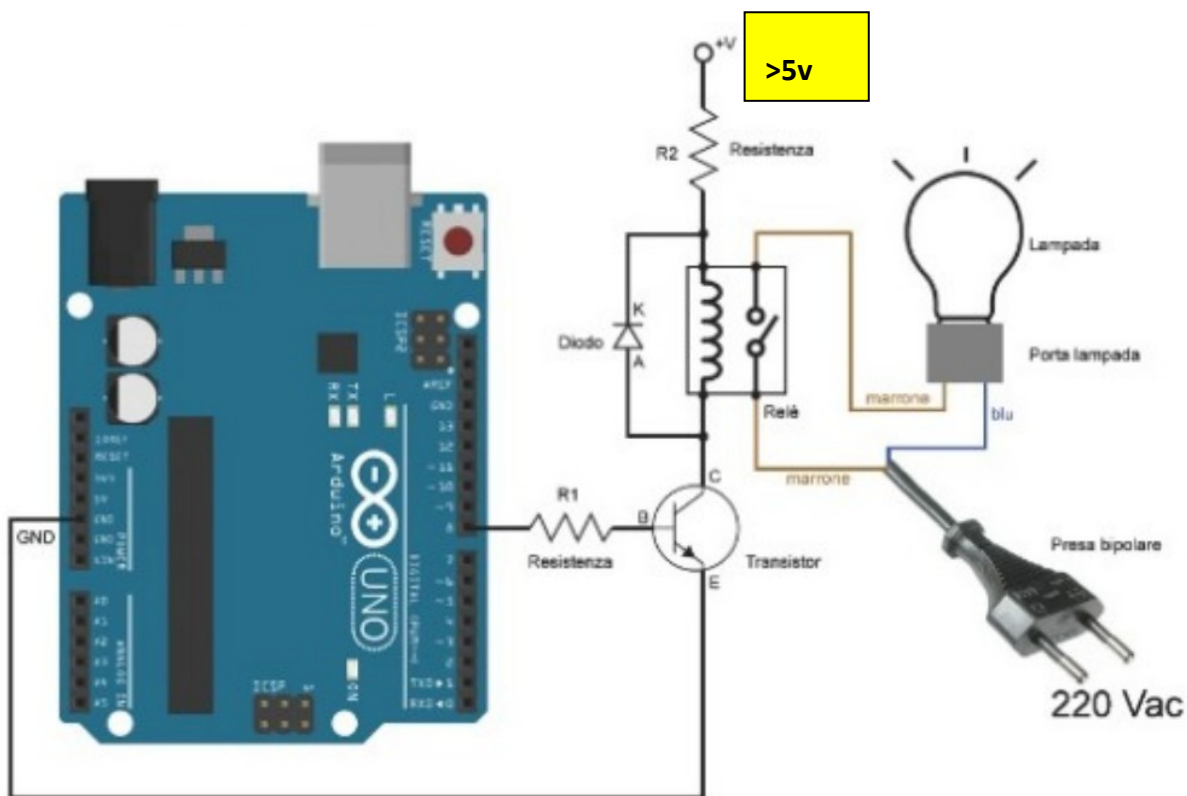
Del relè deve essere nota la corrente minima necessaria ad attivare la bobina e la tensione minima richiesta ai suoi capi.

La  $R_2$  in serie alla bobina del relè serve per limitare correttamente la tensione sulla bobina poiché generalmente l'alimentatore usato per il relè è a 12 o 24V.

Ad esempio se il relè necessita di 12V e 100mA, utilizzando un alimentatore da 24V, sulla  $R_2$  dovremo avere  $24 - 12 = 12V$  (trascurando la  $V_{ce}$  nel transistor).

Di conseguenza per avere una  $I_c = 50mA$  servirà una resistenza  $R_2 = 12 / 0.05 = 240 \text{ ohm}$ .

La potenza che la  $R_2$  deve dissipare è pari a  $Pot. = 0.05 * 12V = 0.6 \text{ watt}$  (→ scegliere R da 1 watt).



## ESERCIZIO BJT + RELÈ'

Si vuole accendere e spegnere una lampada a 220V tramite Arduino.

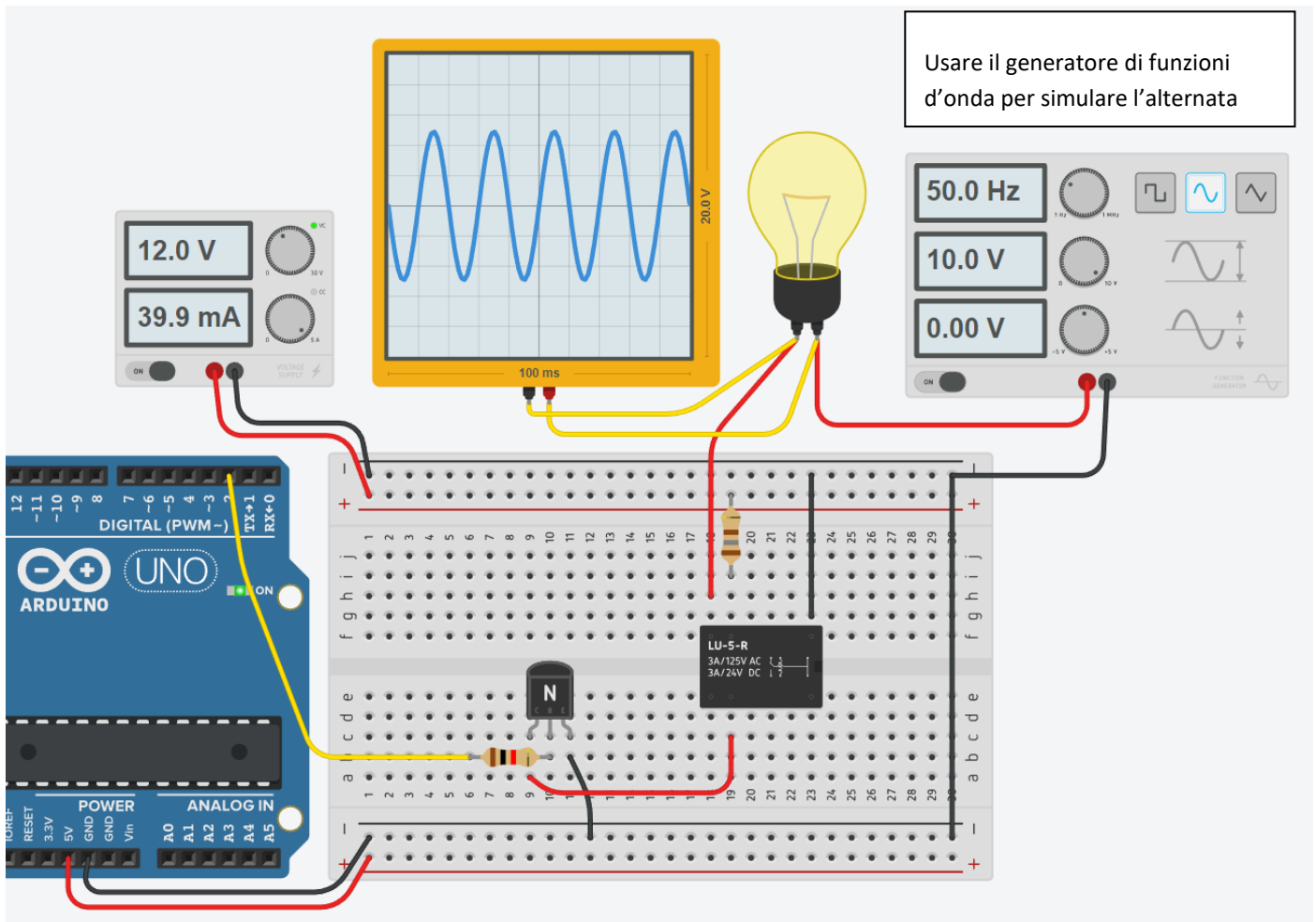
Il relè impiegato ha una bobina che necessita di una tensione di alimentazione di 5V e una corrente di 40 mA.

La sua resistenza vale quindi  $(V/I) = 125 \text{ ohm}$ .

Trascurando la caduta di tensione  $V_{ce}$  sul transistor servirà una resistenza  $R_2$  in serie al relè che abbia una caduta di tensione pari a 7V (12-5).

Quindi la  $R_2$  varrà  $175 \text{ ohm}$  ( $7/0.04$ ) con una potenza da dissipare pari a circa 0.28W (scegliere R da 1 watt!).

La resistenza di base  $R_1$  si calcola noto l' "hfe" del transistor.



## CODICE

```
void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
}
```

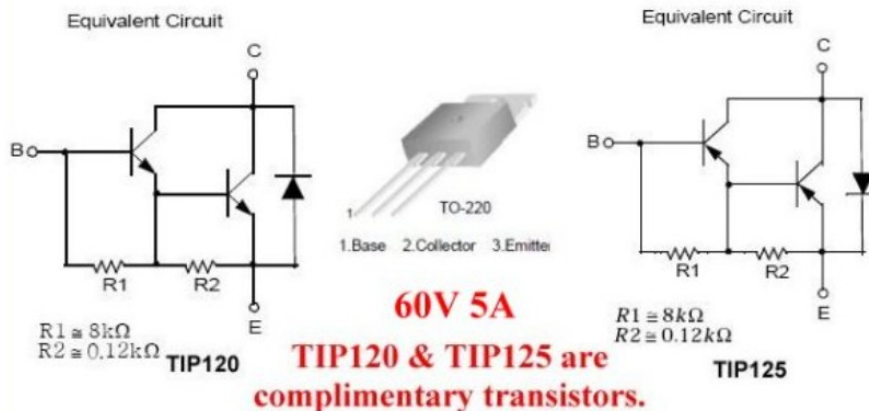
## IL TRANSISTOR DI POTENZA (DARLINGTON)

Un Darlington utilizza almeno due transistor bipolari in cui i collettori sono legati insieme, l'emettitore del transistor più piccolo è legato alla base del transistor più grande, mentre le connessioni del circuito sono fatte all'emettitore dal transistor più grande e la base del transistor più piccolo è l'ingresso.

In questo modo si ottiene un guadagno di potenza maggiore di quello che può fornire un singolo transistor.

Il guadagno di corrente è il prodotto dell' $h_{fe}$  di ogni singolo transistor, mentre la maggior parte della corrente è trasportata dal transistor più grande.

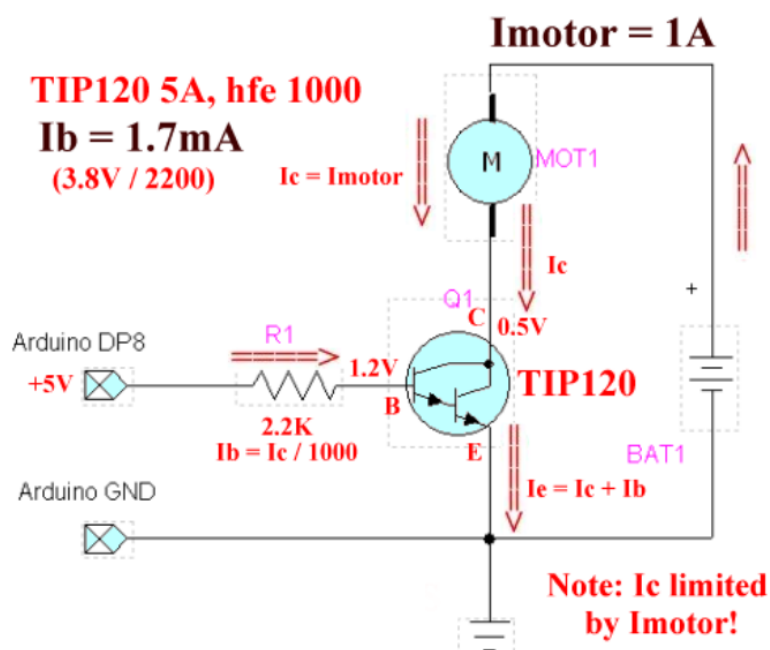
Un classico transistor di potenza è il TIP 120 di cui si allega un estratto del datasheet.



Symbol	Parameter	Test Condition	Min.	Max.	Units
$h_{FE}$	* DC Current Gain	$V_{CE} = 3V, I_C = 0.5A$ $V_{CE} = 3V, I_C = 3A$	1000 1000		
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 3A, I_B = 12mA$ $I_C = 5A, I_B = 20mA$		2.0 4.0	V
$V_{BE(on)}$	* Base-Emitter ON Voltage	$V_{CE} = 3V, I_C = 3A$		2.5	V

Farchild TIP120 (NPN) and TIP125 (PNP) Darlington Transistors

Collegamento tipico del TIP120 per pilotare un motore CC.



## TIP120 PER ATTIVARE ELEMENTO RISCALDANTE RESISTIVO

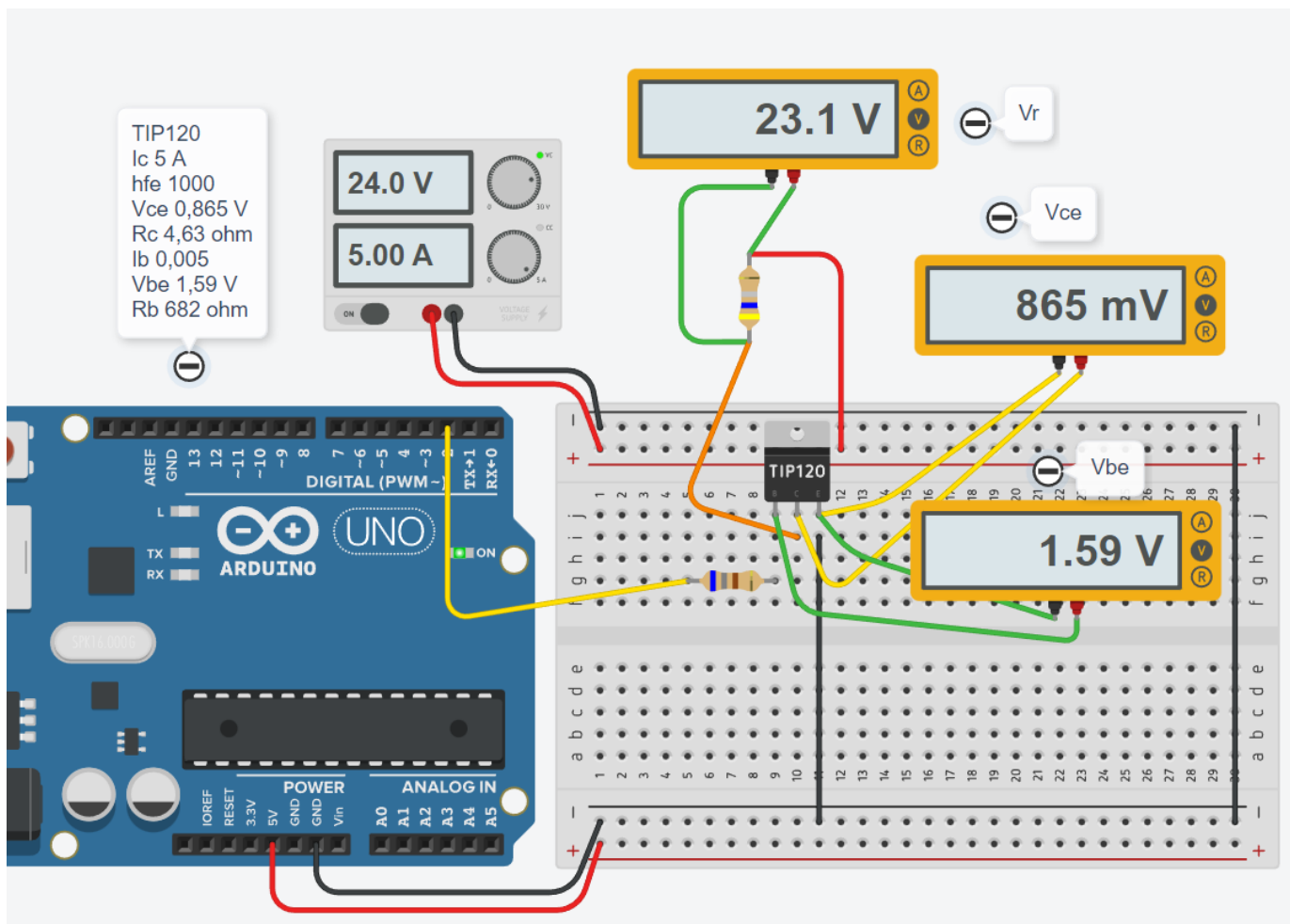
Una resistenza elettrica percorsa da corrente continua genera un potenza termica pari a quella elettrica assorbita (Joule) :

$$Pot = V \cdot I \text{ [watt]} .$$

Tramite un transistor di potenza TIP120 ( $I_c$  max 5A) si vuole generare una potenza termica di circa 115 watt.

Sul simulatore Thinkercad si può osservare la corrente prodotta dal generatore di tensione: 5A.

Il prodotto  $V_{Rxl} = 23.1 \times 5 = 115.5 \text{ watt}$  fornisce la potenza dissipata dalla resistenza.



NOTA:

Regola empirica per calcolare la differenza di temperatura dovuta alla dissipazione termica:

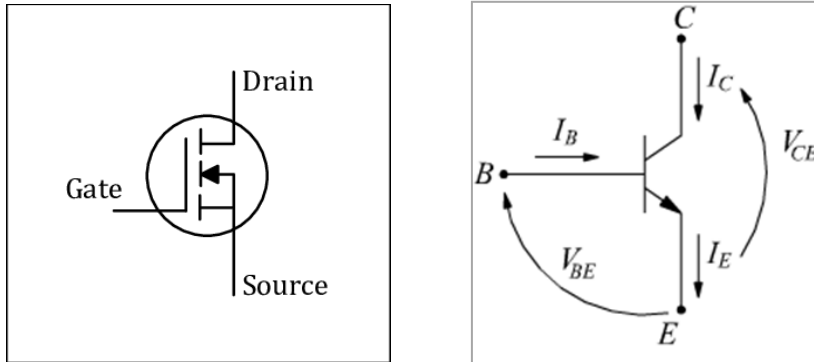
$$\Delta T \approx \frac{R(T) I^2}{80 \div 100 \text{ mW}/^\circ\text{C}}$$

## IL TRANSISTOR MOSFET

Un MOSFET a canale N è un transistor che funziona utilizzando una tensione di ingresso positiva.

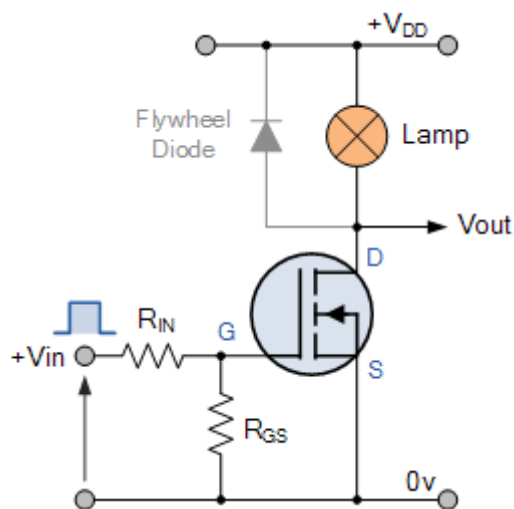
Ha una resistenza di ingresso estremamente elevata (quasi infinita) che consente di utilizzare il MOSFET come interruttore comandato da un microcontrollore (in grado di produrre una tensione positiva sufficiente a portarlo in saturazione). Come per il BJT può anche essere usato come amplificatore di corrente.

La nomenclatura dei pin del MOSFET è diversa da quella del BJT:



Applicando un'opportuna tensione di pilotaggio al gate G di un MOSFET, la resistenza del canale drain-source (D-S),  $R_{DS(on)}$  varia da un valore di molte centinaia di  $k\Omega$  (circuito aperto) ad un valore inferiore a  $1\Omega$  (cortocircuito).

### Un esempio di utilizzo del MOSFET come interruttore



In questo circuito viene utilizzato un MOSFET a canale N per accendere e spegnere una semplice lampada.

La tensione di ingresso del gate  $V_{GS}$  viene portata ad un livello di tensione positivo appropriato (da minimo 2-3V e oltre) per accendere il dispositivo e a un livello di tensione 0 per spegnerlo .

Se il carico resistivo della lampada dovesse essere sostituito da un carico induttivo come una bobina, un solenoide o un relè, sarebbe necessario un "diodo di protezione" in parallelo al carico per proteggere il MOSFET da eventuali correnti di ritorno.

La potenza dissipata nel MOSFET ( $P_D$ ) dipende dalla corrente che scorre attraverso il canale  $I_D$  a saturazione e anche dalla "resistenza" del canale  $R_{DS(on)}$  .

## DIMENSIONAMENTO MOSFET COME INTERRUOTTORE

Supponiamo di dover accendere una lampada di potenza a 6 V (24 W).

Il MOSFET standard impiegato ha un valore di resistenza di attivazione del canale  $R_{DS(on)}$  di 0,1 ohm.

Calcolare la potenza dissipata nel dispositivo di commutazione MOSFET.

La corrente che scorre attraverso la lampada è calcolata come:

$$P = V \times I_D$$

$$\therefore I_D = \frac{P}{V} = \frac{24}{6} = 4.0 \text{ amps}$$

Quindi la potenza dissipata nel MOSFET sarà data come:

$$P = I^2 \cdot R$$

$$P_D = I_D^2 \times R_{DS}$$

$$\therefore P_D = 4^2 \times 0.1 = 1.6 \text{ watts}$$

Quando si utilizza il MOSFET come interruttore per controllare motori CC o carichi elettrici con correnti di spunto elevate, la resistenza del canale "ON" ( $R_{DS(on)}$ ) tra drain D e il source S è molto importante.

Poiché la relazione di potenza di base è:  $P = I^2 R$ , un valore di resistenza del canale  $R_{DS(on)}$  elevato comporterebbe semplicemente la dissipazione e lo spreco di grandi quantità di potenza all'interno del MOSFET stesso con conseguente aumento eccessivo della temperatura, che se non controllato potrebbe causare il riscaldamento e il danneggiamento del MOSFET a causa di un sovraccarico termico.

Un valore  $R_{DS(on)}$  più basso per la resistenza del canale è anche un parametro desiderabile in quanto aiuta a ridurre la tensione di saturazione effettiva del canale ( $V_{DS(sat)} = I_D \cdot R_{DS(on)}$ ) attraverso il MOSFET e quindi funzionerà ad una temperatura più bassa.

I MOSFET di potenza hanno generalmente un valore  $R_{DS(on)}$  inferiore a 0,01Ω che consente loro di funzionare a temperature basse, prolungando la loro durata operativa.

Una delle principali limitazioni quando si utilizza un MOSFET come dispositivo di commutazione è la massima corrente di drenaggio D che può gestire.

Quindi il parametro  $R_{DS(on)}$  è una guida importante per l'efficienza di commutazione del MOSFET ed è semplicemente dato come rapporto di  $V_{DS} / I_D$  quando il transistor è attivo.

Quando si utilizza un MOSFET o qualsiasi tipo di transistor ad effetto di campo come dispositivo di commutazione a stato solido, è sempre consigliabile selezionare quelli che hanno un valore  $R_{DS(on)}$  molto basso e dotarli di un dissipatore di calore adatto per aiutare ridurre qualsiasi fuga termica e danni.

*I MOSFET di potenza utilizzati come interruttore generalmente hanno una protezione da sovracorrente integrata nel loro design, ma per applicazioni ad alta corrente il transistor a giunzione bipolare BJT è una scelta migliore.*

## CONTROLLO MOTORE MOSFET DI POTENZA

L'elevatissima resistenza di ingresso (o di gate) del MOSFET, la velocità di commutazione molto elevata e la facilità con cui possono essere pilotati li rendono ideali per interfacciarsi con amplificatori operazionali, porte logiche standard e microcontrollori.

Tuttavia, è necessario prestare attenzione per garantire che la tensione di ingresso gate-source (G-S) sia scelta correttamente perché quando si utilizza il MOSFET come interruttore, il dispositivo deve presentare una bassa resistenza del canale  $R_{DS(on)}$ , proporzionale alla tensione di gate di ingresso.

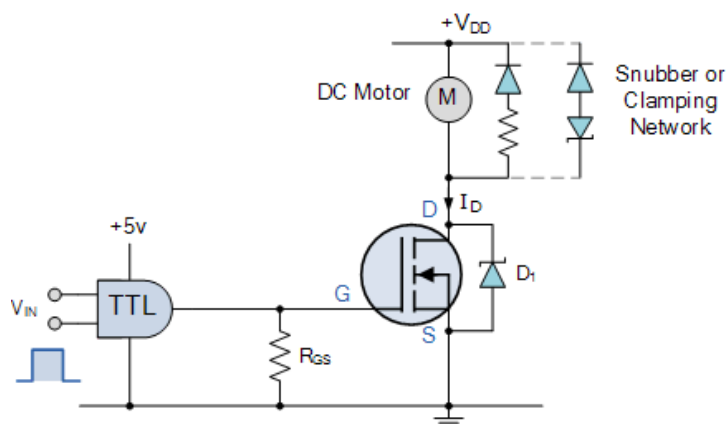
I MOSFET di potenza di tipo a *soglia bassa* potrebbero non commutare su "ON" fino a quando non sono stati applicati almeno 3 V o 4 V alla sua porta e se l'uscita dalla porta logica è solo +5 V logico potrebbe non essere sufficiente per portare completamente il MOSFET in saturazione.

Per micro controllori tipo Arduino, ESP32 ecc. sono disponibili MOSFET a *soglia bassa* progettati per l'interfacciamento con soglie comprese tra 1,5 V e 2,0 V.

I MOSFET di potenza possono essere utilizzati per controllare il movimento di motori CC o motori passo-passo brushless direttamente dalla logica del computer o utilizzando controller di tipo PWM (pulse-width modulation).

I MOSFET, controllati in PWM, possono essere utilizzati per controllare di velocità di funzionamento dei motori CC in modo fluido e silenzioso.

## CIRCUITO MOTORE CC MOSFET DI POTENZA SEMPLICE



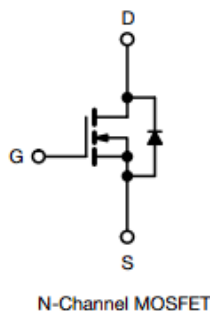
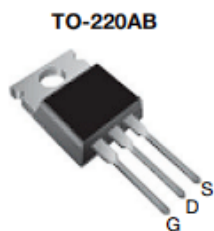
Poiché il carico del motore è induttivo, un semplice diodo di protezione è collegato in parallelo al motore per dissipare l'eventuale fem generata dal motore quando il MOSFET lo spegne.

È inoltre possibile utilizzare una rete di bloccaggio formata da un diodo zener in serie con il diodo per consentire una commutazione più rapida e un migliore controllo della tensione inversa di picco e del tempo di caduta.

Per una maggiore sicurezza, è anche possibile posizionare un diodo zener o al silicio  $D_1$  aggiuntivo attraverso il canale di un interruttore MOSFET quando si utilizzano carichi induttivi, come motori, relè, solenoidi, ecc., per sopprimere i transitori di commutazione di sovratensione e il rumore, fornendo una protezione aggiuntiva all'Interruttore MOSFET se necessario.

Il resistore  $R_{GS}$  viene utilizzato come resistore di pull-down per aiutare a ridurre la tensione di uscita TTL a 0 V quando il MOSFET è disattivato. Tipicamente si usa 1K.

## Power MOSFET



### FEATURES

- Dynamic dV/dt rating
- Repetitive avalanche rated
- 175 °C operating temperature
- Fast switching
- Ease of paralleling
- Simple drive requirements
- Material categorization: for definitions of compliance please see [www.vishay.com/doc?99912](http://www.vishay.com/doc?99912)



Available  
**RoHS\***  
 Available  
 HALOGEN  
**FREE**  
 Available

### Note

\* This datasheet provides information about parts that are RoHS-compliant and / or parts that are non RoHS-compliant. For example, parts with lead (Pb) terminations are not RoHS-compliant. Please see the information / tables in this datasheet for details

### DESCRIPTION

Third generation power MOSFETs from Vishay provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance and cost-effectiveness.

The TO-220AB package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 W. The low thermal resistance and low package cost of the TO-220AB contribute to its wide acceptance throughout the industry.

### PRODUCT SUMMARY

$V_{DS}$ (V)	100	
$R_{DS(on)}$ ( $\Omega$ )	$V_{GS} = 10\text{ V}$	0.27
$Q_g$ max. (nC)	16	
$Q_{gs}$ (nC)	4.4	
$Q_{gd}$ (nC)	7.7	
Configuration	Single	

### ORDERING INFORMATION

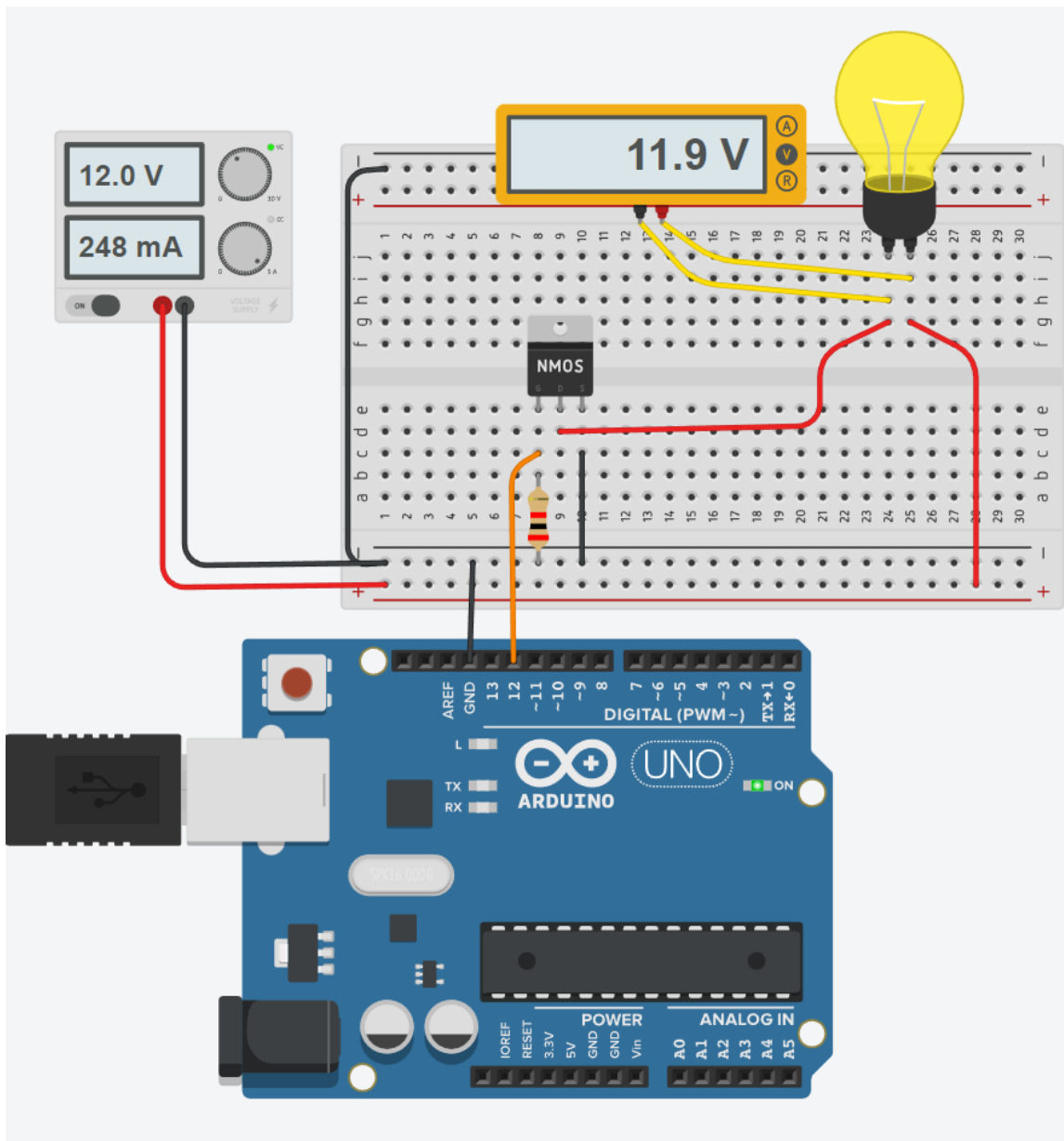
Package	TO-220AB
Lead (Pb)-free	IRF520PbF
Lead (Pb)-free and halogen-free	IRF520PbF-BE3

### ABSOLUTE MAXIMUM RATINGS ( $T_C = 25\text{ }^\circ\text{C}$ , unless otherwise noted)

PARAMETER	SYMBOL	LIMIT	UNIT
Drain-source voltage	$V_{DS}$	100	V
Gate-source voltage	$V_{GS}$	$\pm 20$	
Continuous drain current	$I_D$	$T_C = 25\text{ }^\circ\text{C}$	9.2
		$T_C = 100\text{ }^\circ\text{C}$	6.5
Pulsed drain current <sup>a</sup>	$I_{DM}$	37	A
Linear derating factor		0.40	W/ $^\circ\text{C}$
Single pulse avalanche energy <sup>b</sup>	$E_{AS}$	200	mJ
Repetitive avalanche current <sup>a</sup>	$I_{AR}$	9.2	A
Repetitive avalanche energy <sup>a</sup>	$E_{AR}$	6.0	mJ
Maximum power dissipation	$P_D$	60	W
Peak diode recovery dV/dt <sup>c</sup>	dV/dt	5.5	V/ns
Operating junction and storage temperature range	$T_J, T_{stg}$	-55 to +175	$^\circ\text{C}$
Soldering recommendations (peak temperature) <sup>d</sup>	For 10 s	300	
Mounting torque	6-32 or M3 screw	10	lbf · in
		1.1	N · m

## ESERCIZIO CON NMOS

Attivare una lampada a 12V e 250mA tramite un MOSFET.



### CODICE

```
void setup()
{
  pinMode(12, OUTPUT);
}

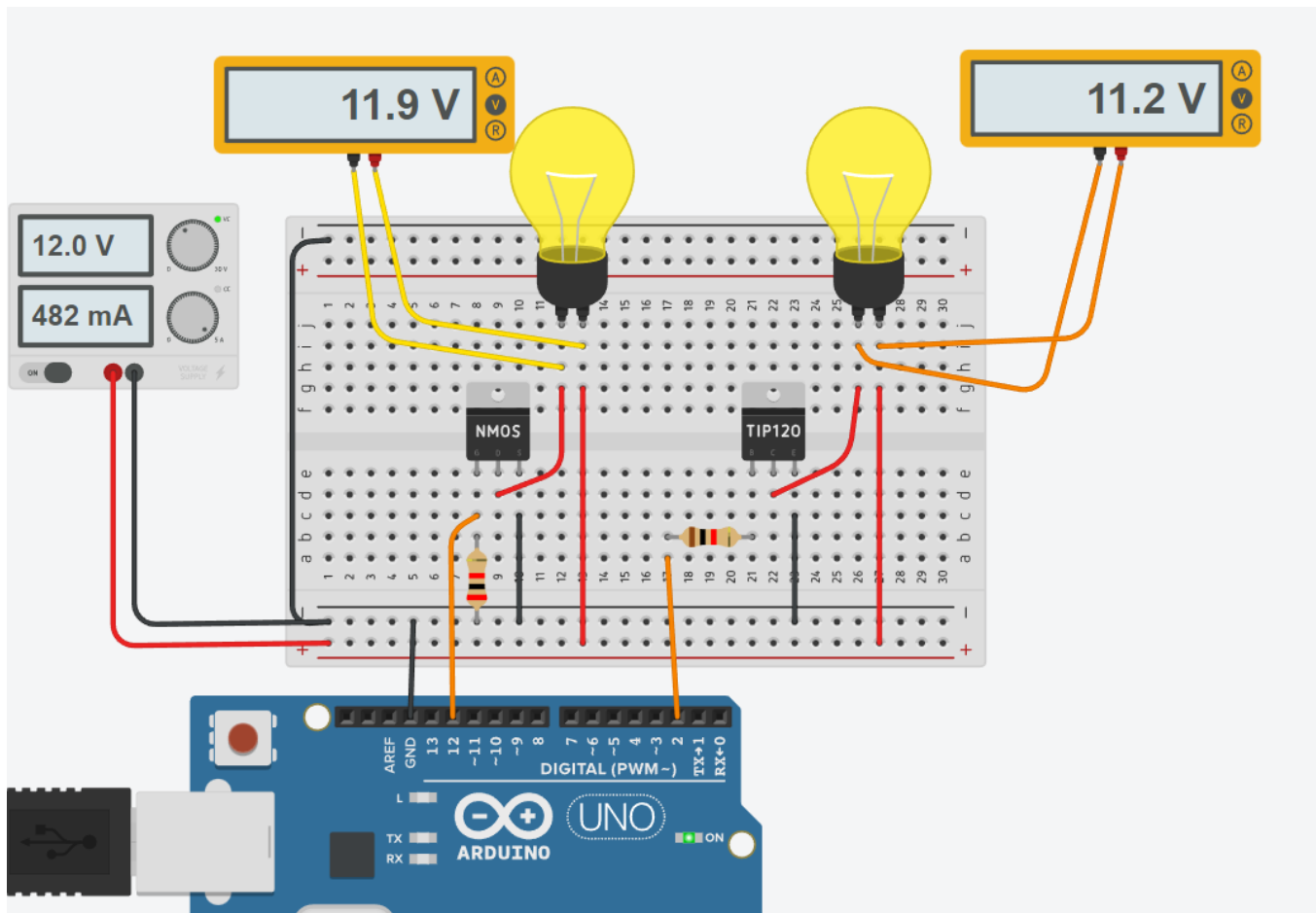
void loop()
{
  digitalWrite(12, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(12, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

## CONFRONTO FRA TRANSISTOR BJT E NMOS

Il circuito sottostante mostra lo stesso utilizzatore (lampadina) controllato tramite un TIP120 e NMOS.

La differenza sostanziale è che con l'NMOS si ha una caduta di tensione  $V_{DS}$  quasi trascurabile rispetto alla caduta di tensione  $V_{CE}$  del BJT.

In questo modo si ha un minore spreco di potenza elettrica ( $V \cdot I$ ) e quindi meno calore dissipato dal transistor.



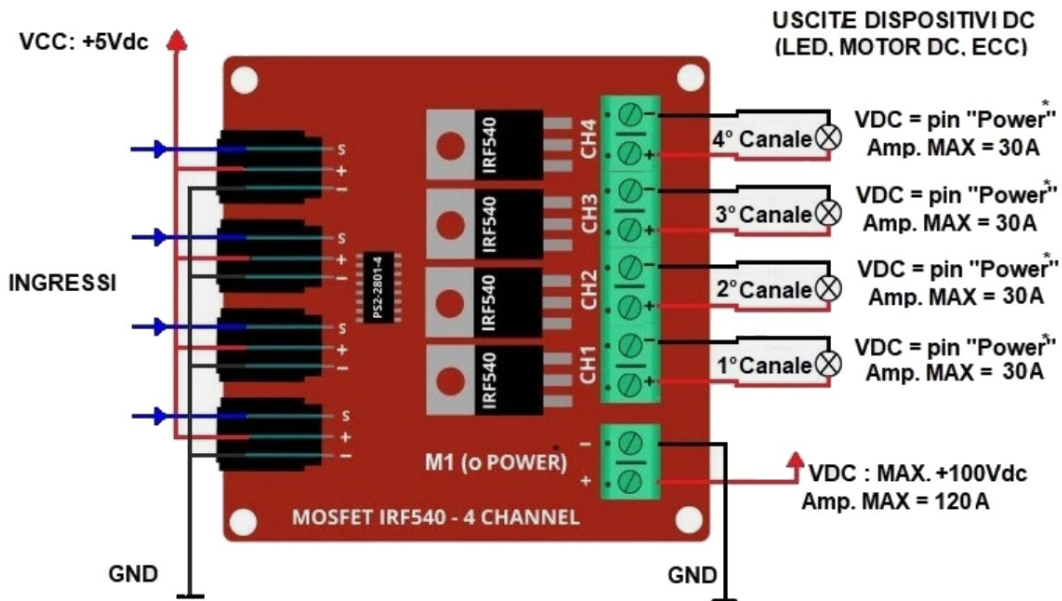
### CODICE

```
void setup()
{
  pinMode(12, OUTPUT);
  pinMode(2, OUTPUT);
}

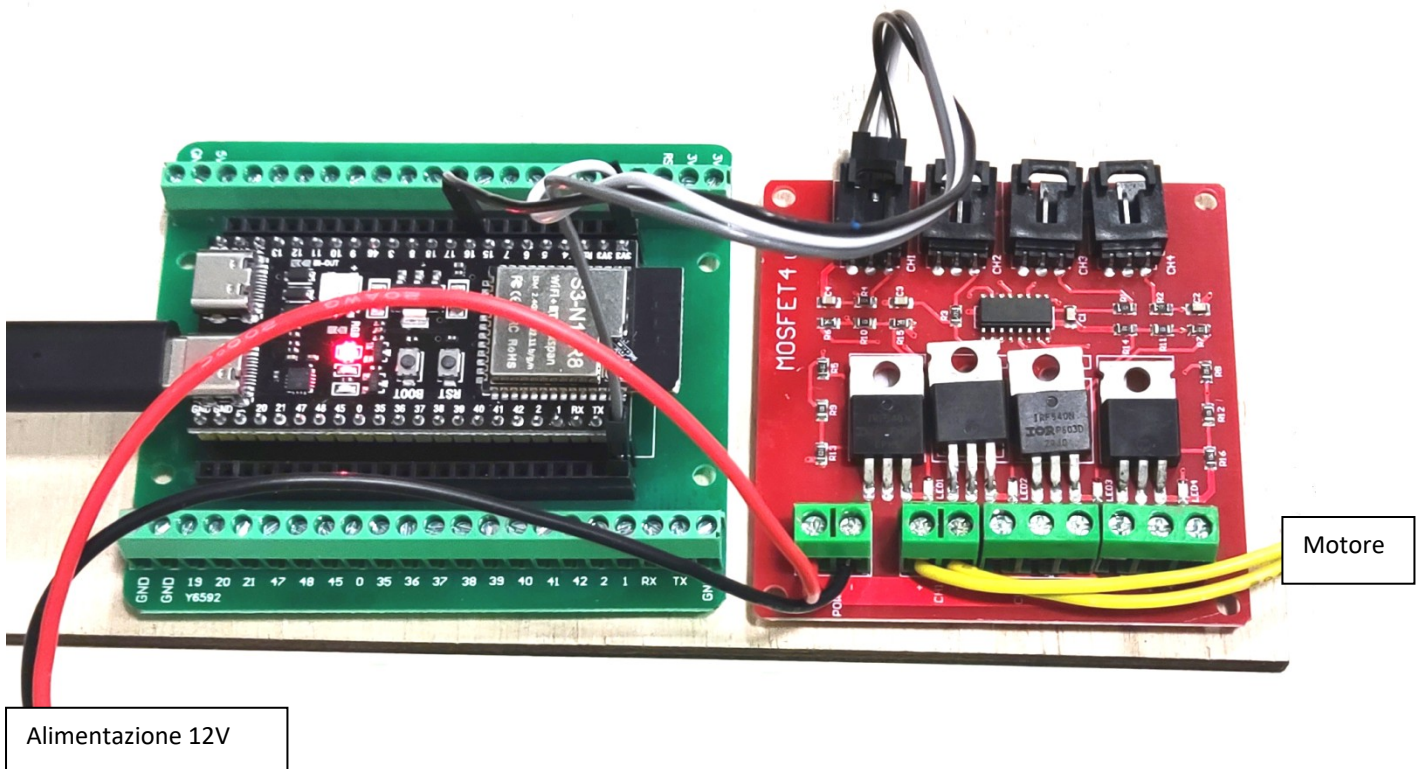
void loop()
{
  digitalWrite(12, HIGH);
  digitalWrite(2, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(12, LOW);
  digitalWrite(2, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

## SHIELD MOSFET 4 CANALI ROSSA

Questa scheda ospita 4 mosfet di potenza IRF540 (fino a 140W) e permette quindi comandare contemporaneamente fino a 4 carichi con una tensione massima di 100V. Questi mosfet possono essere utilizzati anche con una tensione di pilotaggio di 3.3V tipica dei micro ESP32.

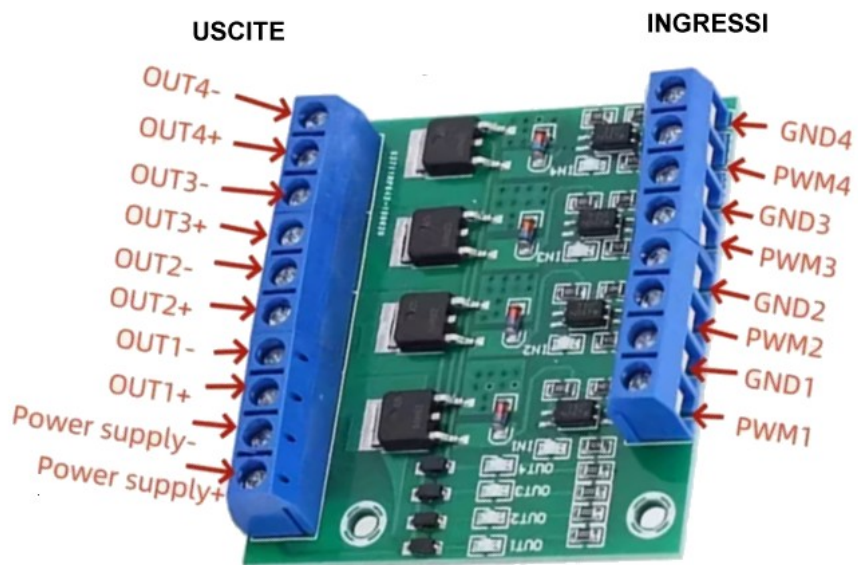


Ogni INGRESSO necessita di tre connessioni: massa, alimentazione e segnale (3.3-5V o PWM per regolazione corrente)

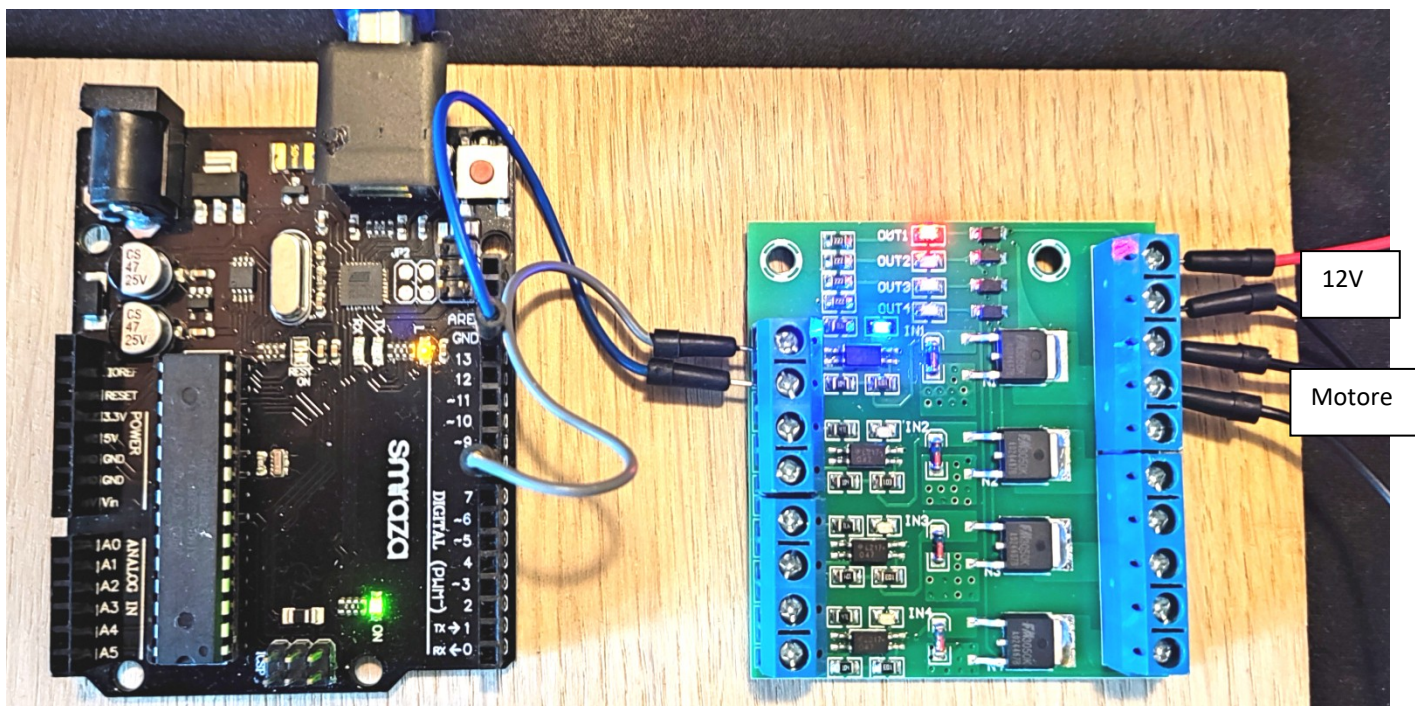


## SHIELD MOSFET 4 CANALI BLU

Questa scheda ospita 4 mosfet di potenza IRF540 (fino a 140W) con una tensione massima di 100V. Necessita di una tensione di comando di almeno 5V e quindi non può essere utilizzata gli EPS32.



Ogni INGRESSO necessita di 2 connessioni: massa, e segnale (5V o PWM per regolazione corrente)



```
int pinMotore=8;

void setup()
{
  pinMode(pinMotore,OUTPUT);
}

void loop()
{
  digitalWrite(8,HIGH);
  delay(2000);
  digitalWrite(8,LOW);
  delay(2000);
}
```



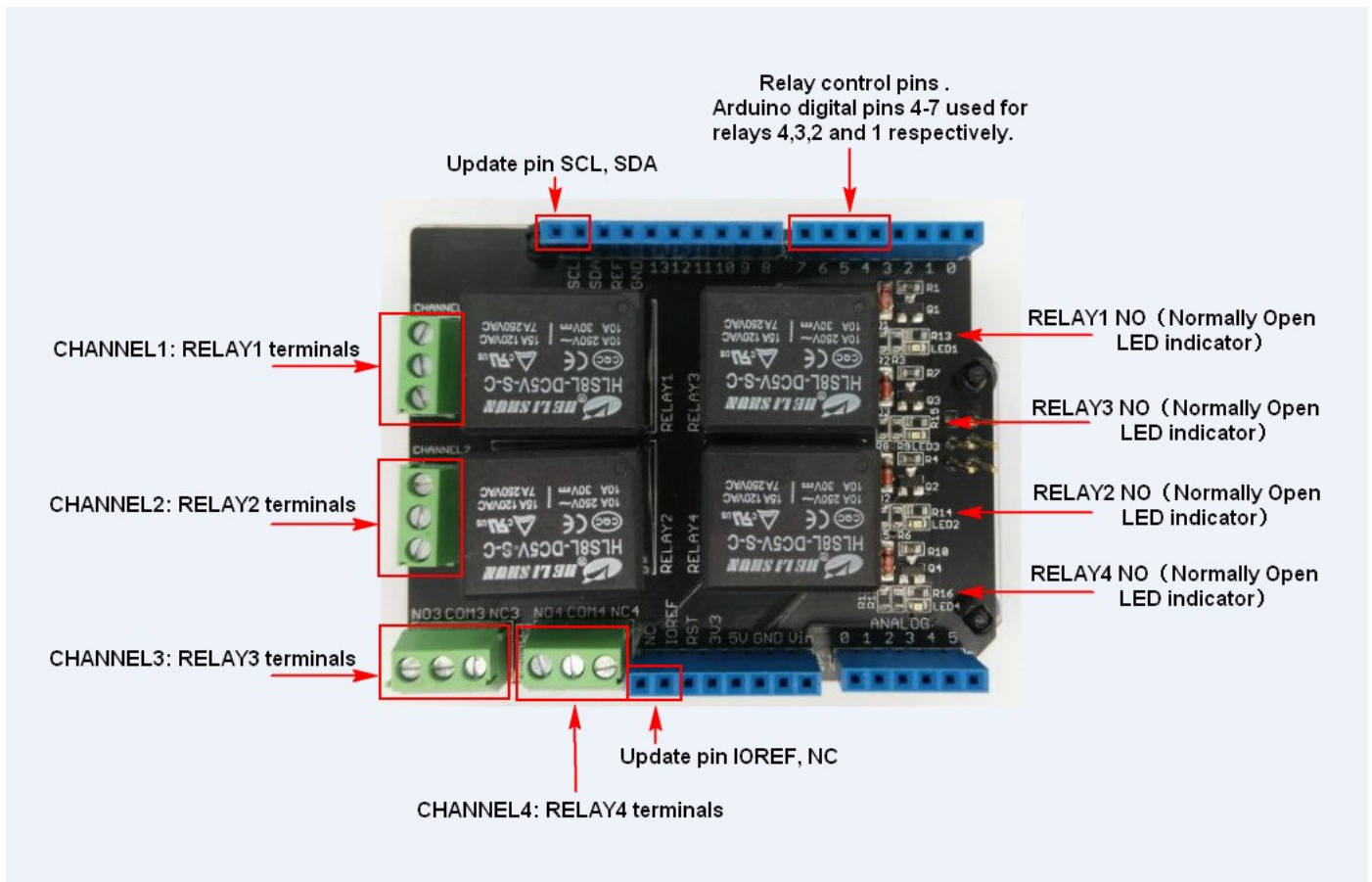
Relay Shield fornisce una soluzione per il controllo di dispositivi ad alta corrente che non possono essere controllati dai pin I/O digitali di Arduino a causa dei limiti di corrente e tensione.

Relay Shield è dotato di quattro relè di alta qualità e fornisce interfacce NO/NC, quattro indicatori LED dinamici per mostrare lo stato acceso/spento di ciascun relè e il fattore di forma dello shield standardizzato per garantire una connessione fluida alla scheda Arduino o ad altre schede compatibili con Arduino.

Articolo	Minimo	Tipico	Massimo	Unità
Tensione di alimentazione	4.75	5	5.25	VDC
Corrente di lavoro	8	-	250	mA
Tensione di commutazione	-	-	30	VDC
Corrente di commutazione	-	-	8	UN
Frequenza	-	1	-	Hz
Potenza di commutazione	-	-	70	O
Vita del relè	100000	-	-	Ciclo
scarica da contatto ESD	-	±4	-	KV
scarica dell'aria ESD	-	±8	-	KV
Dimensione	-	68,7X53,5X30,8	-	mm
Peso netto	-	55±2	-	G

**Attenzione**

- Applicare 2 strati di nastro isolante sulla parte superiore del connettore USB dell'Arduino in modo da impedire allo shield di entrare in contatto col connettore (cortocircuito ...).
- Non utilizzare una tensione superiore a 35 V CC lato relè.



- Digitale 4 – controlla il pin COM4 del RELAY4 (situato in J4)
- Digitale 5 – controlla il pin COM3 del RELAY3 (situato in J3)
- Digitale 6 – controlla il pin COM2 del RELAY2 (situato in J2)
- Digitale 7 – controlla il pin COM1 del RELAY1 (situato in J1)

Descrizione del pin dell'interfaccia/terminale J1:

COM1 (pin comune) : il pin del relè controllato dal pin digitale.

NC1 (normalmente chiuso) : questo terminale sarà collegato a COM1 quando il pin di controllo RELAY1 (pin I/O digitale 7) è impostato basso e scollegato quando il pin di controllo RELAY1 è impostato alto.

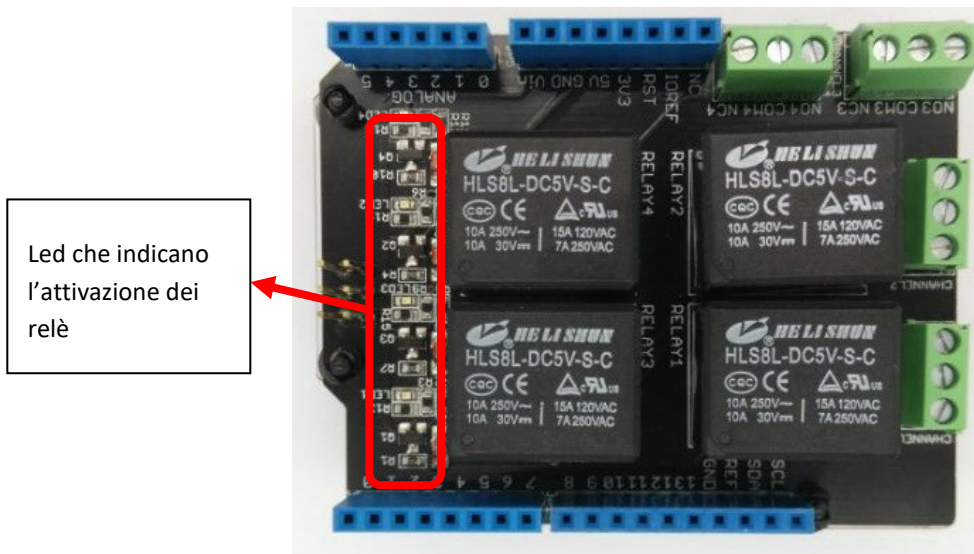
NO1 (normalmente aperto) : questo terminale sarà collegato a COM1 quando il pin di controllo RELAY1 (pin I/O digitale 7) è impostato su alto e scollegato quando il pin di controllo RELAY1 è impostato su basso.

I terminali J2-4 sono simili a J1, tranne per il fatto che controllano rispettivamente RELAY2-RELAY4.

**Nota:** per controllare i quattro diversi relè sono necessari solo quattro pin I/O digitali Arduino, dal 4 al 7. Inoltre, per alimentare il Relay Shield sono necessari anche i pin 5V e due GND Arduino.

## TEST DEI 4 RELE' DELLO SHIELD

Accendere e spegnere in sequenza tutti e quattro i rele' dello shield.



### CODICE

```
int pinRele1 = 4;
int pinRele2 = 5;
int pinRele3 = 6;
int pinRele4 = 7;

void setup() {
  pinMode(pinRele1, OUTPUT);
  pinMode(pinRele2, OUTPUT);
  pinMode(pinRele3, OUTPUT);
  pinMode(pinRele4, OUTPUT);
}

void loop() {
  digitalWrite(pinRele1,HIGH); delay(200);
  digitalWrite(pinRele2,HIGH); delay(200);
  digitalWrite(pinRele3,HIGH); delay(200);
  digitalWrite(pinRele4,HIGH);
  delay(2000);

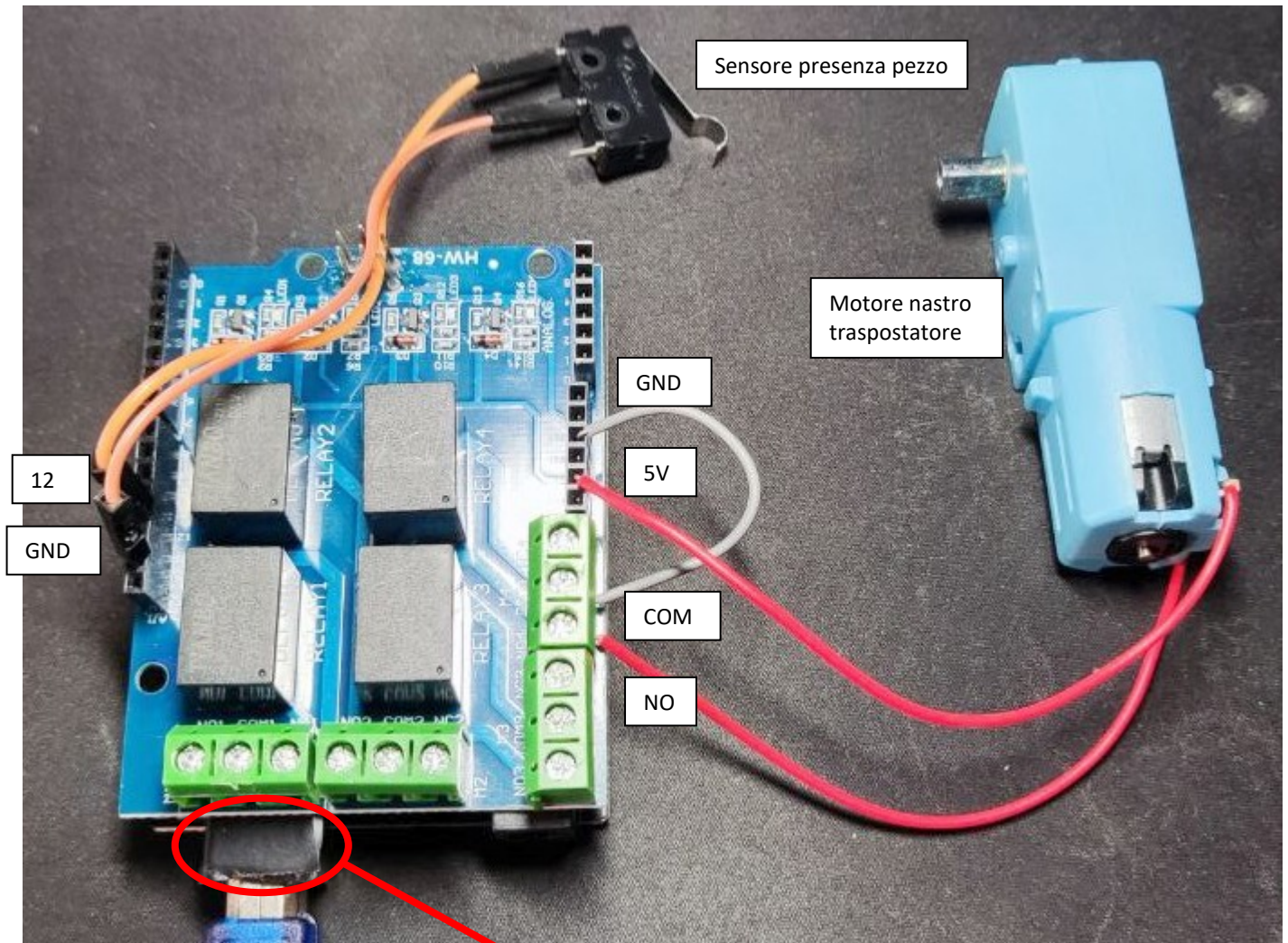
  digitalWrite(pinRele1,LOW); delay(200);
  digitalWrite(pinRele2, LOW); delay(200);
  digitalWrite(pinRele3, LOW); delay(200);
  digitalWrite(pinRele4, LOW);
  delay(2000);
}
```

## SIMULAZIONE NASTRO TRASPOSTATORE

Finchè il sensore di presenza (connesso in modalità pullup) non segnala la presenza del pezzo il motore resta attivo.

MODIFICHE DA FARE:

- 1- aggiungere uno slider (pulsante di attivazione) per avviare il motore solo se slider attivato
- 2- modificare il programma in modo da non usare il "delay"



### CODICE

```
int pinMotor = 4;
int pinFinecorsa = 12;
int statoFinecorsa=1;

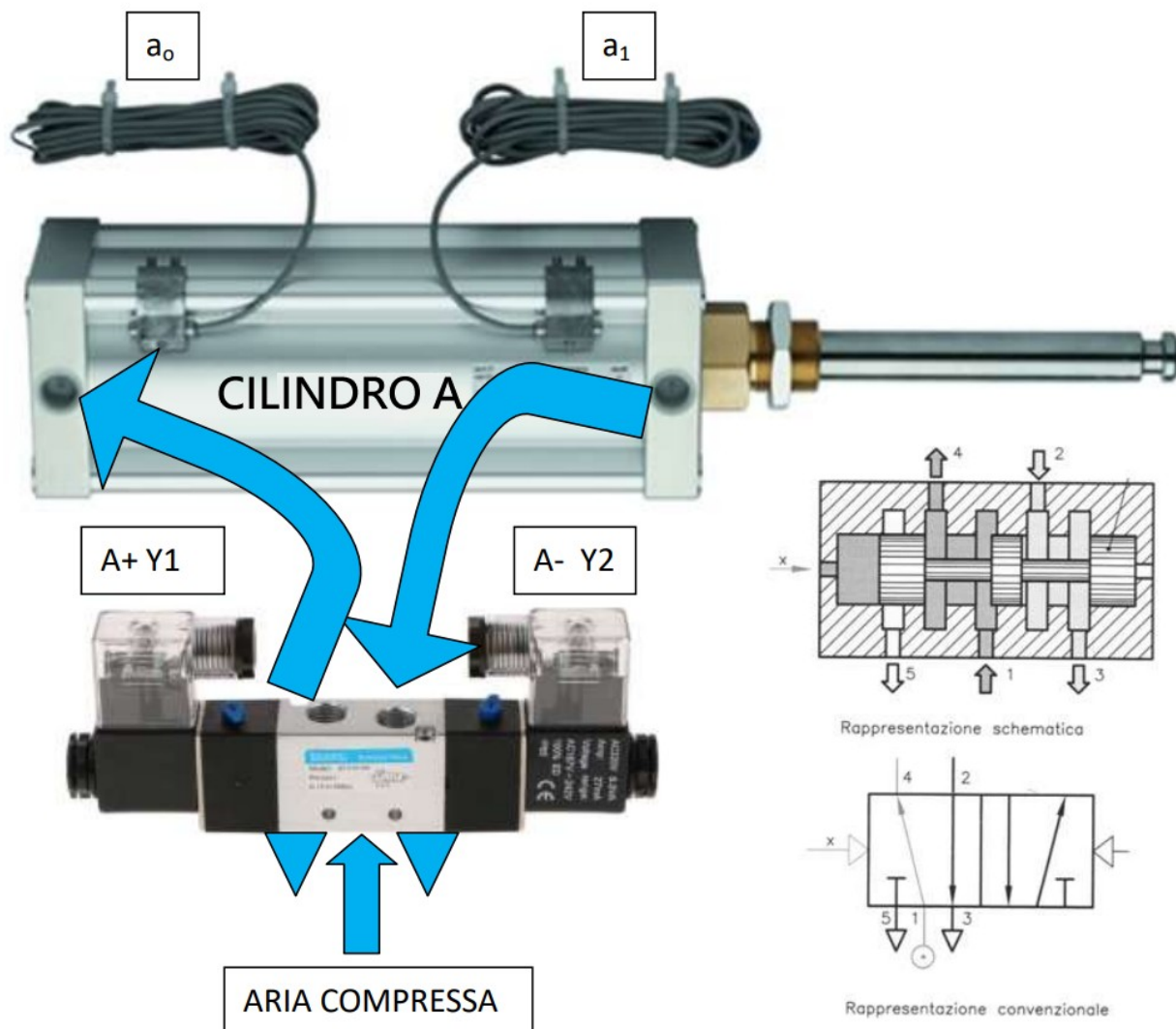
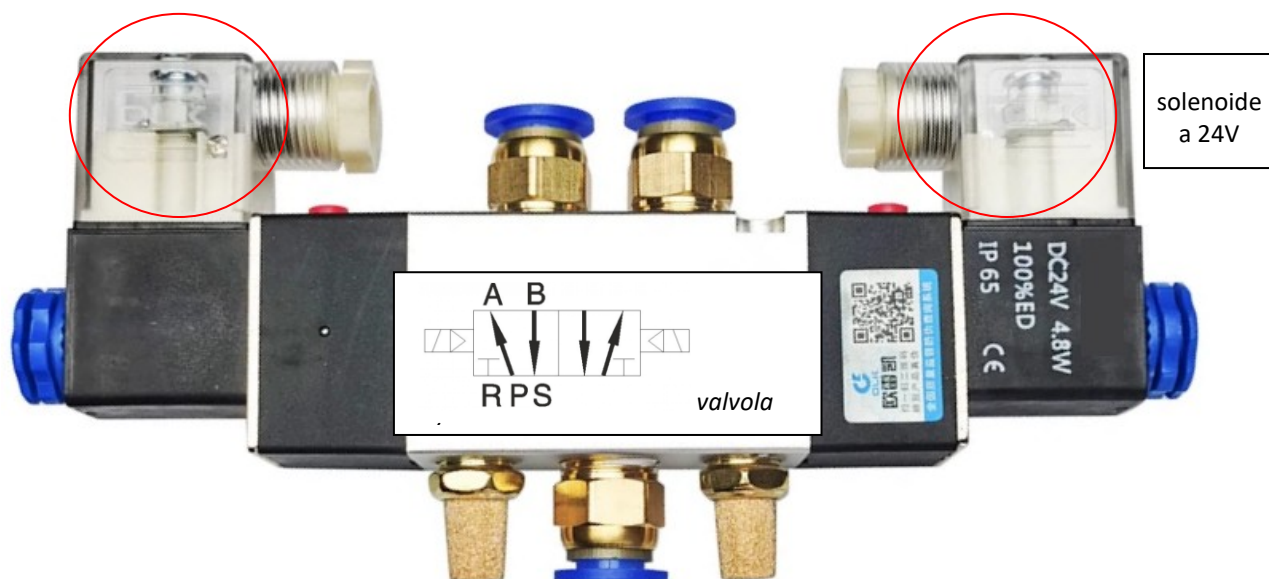
void setup() {
  pinMode(pinMotor, OUTPUT);
  pinMode(pinFinecorsa, INPUT_PULLUP);
}

void loop() {
  statoFinecorsa = digitalRead(pinFinecorsa);
  if (statoFinecorsa == 1) { digitalWrite(pinMotor,HIGH); }
  else { digitalWrite(pinMotor,LOW); }
  delay(20);
}
```

Mettere 2 strati di nastro isolante per evitare contatto diretto fra shield e connettore!

## ELETTROVALVOLE PNEUMATICHE

Una elettrovalvola (o valvola a solenoide) è una valvola che utilizza la forza elettromagnetica per funzionare. Quando una corrente elettrica viene fatta passare attraverso la bobina del solenoide (generalmente alimentata a 24V), viene generato un campo magnetico che provoca il movimento di un perno metallico che permette il passaggio dell'aria da una via ad un'altra.



## PANNELLO DI ELETTROPNEUMATICA

Il pannello per effettuare delle prove utilizza due cilindri connessi a due elettrovalvole 5/2.

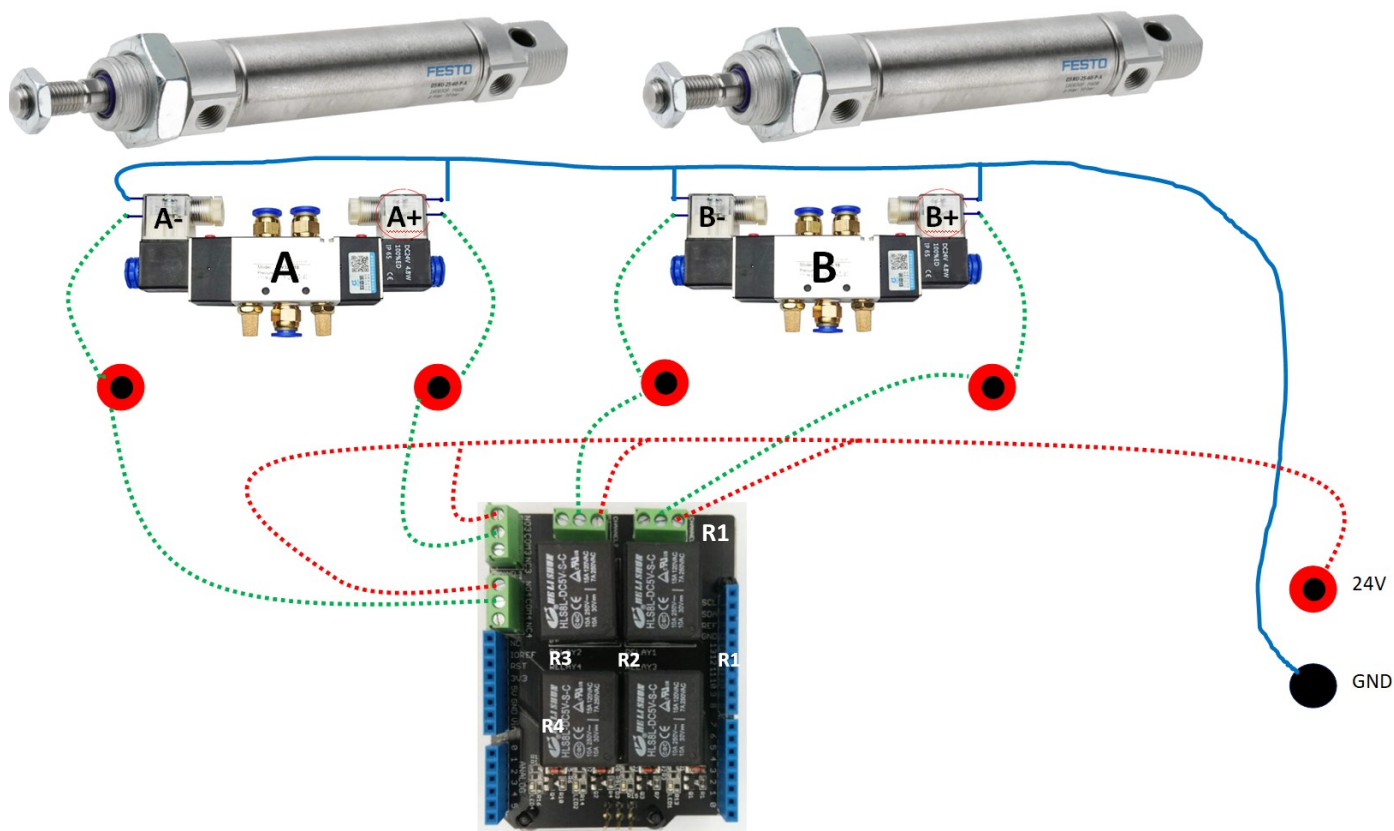
L'uscita dei 4 solenoidi delle elettrovalvole sono tutte connesse a massa.

I quattro ingressi invece sono disponibili per il collegamento esterno a 24V.

Per attivare una fase dei cilindri è necessario alimentare il solenoide corrispondente della elettrovalvola connessa tramite la chiusura del contatto NO dei relè comandati da Arduino in modo da far fluire la corrente dal generatore esterno (24V) al solenoide la cui uscita è già a massa (si chiude il circuito di alimentazione).

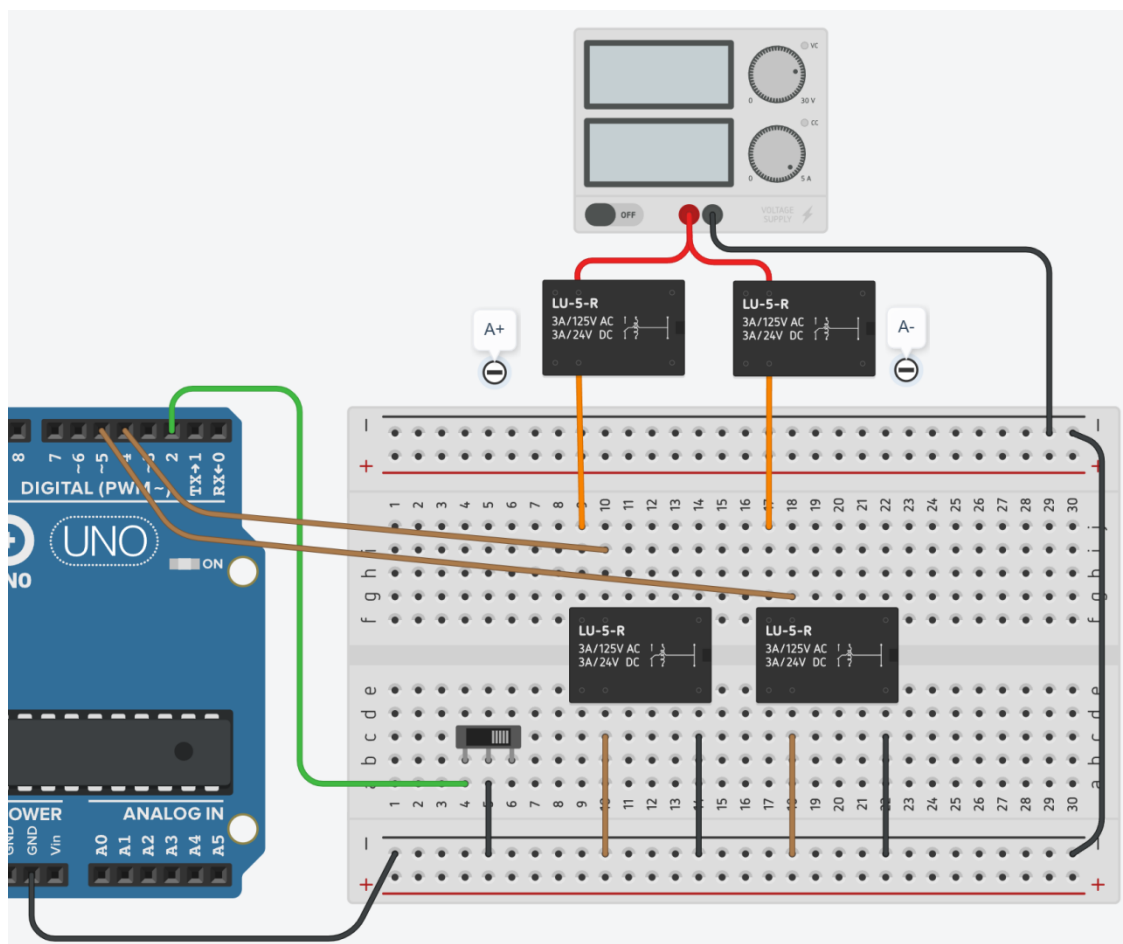
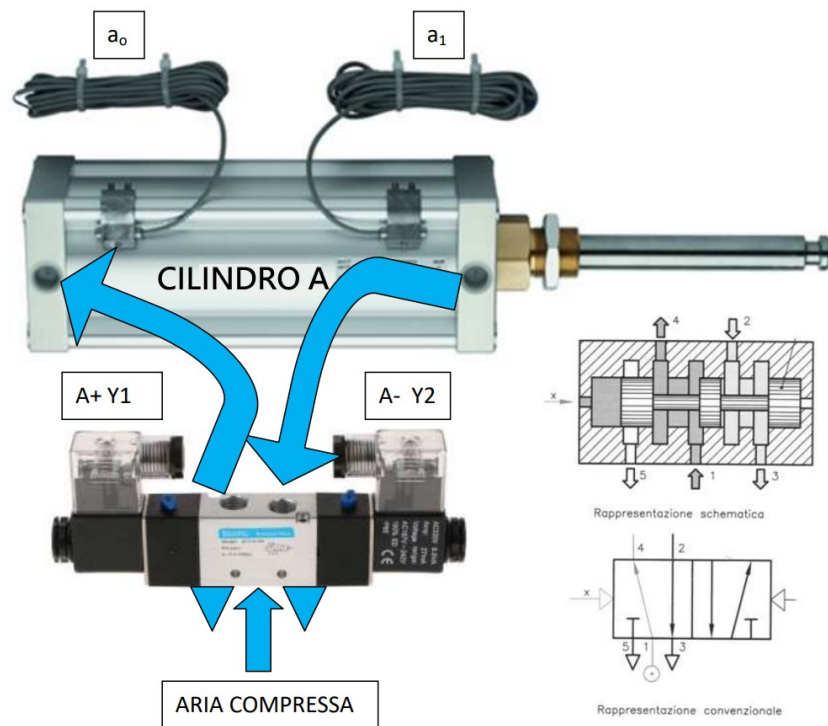
### ESEMPIO

Ad esempio per attivare la fase B+ del cilindro B (a destra) si deve attivare da Arduino il relè R1 in modo che la corrente dal generatore 24V possa arrivare al solenoide B+



## 1° ESERCIZIO

Simulare la sequenza pneumatica manuale A+ A-. Non utilizzare i finecorsa.  
 Mantenere lo stato dello stelo per 2 secondi. L'avvio avviene premendo Start.



## Codice

```
long t0=0;
int n=0;

void setup() {
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

  //disattivo bobine A
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
}

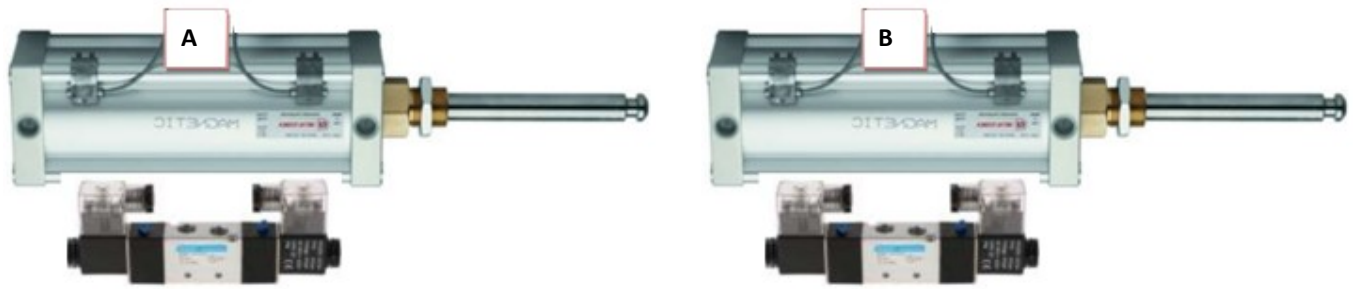
void loop() {
  int sensorVal = digitalRead(2);

  // Start
  if (sensorVal == LOW && n<1) {
    n= n+1;
    // A
    Serial.println("A+");
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(1000);
    // A-
    Serial.println("A-");
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    delay(1000);
  }

  if (sensorVal == HIGH) {
    if (millis() - t0 > 1000) {
      Serial.println("riposo");
      digitalWrite(4, LOW);
      digitalWrite(5, HIGH);
      n=0;
      t0= millis();
    }
  }
}
```

## 2° ESERCIZIO

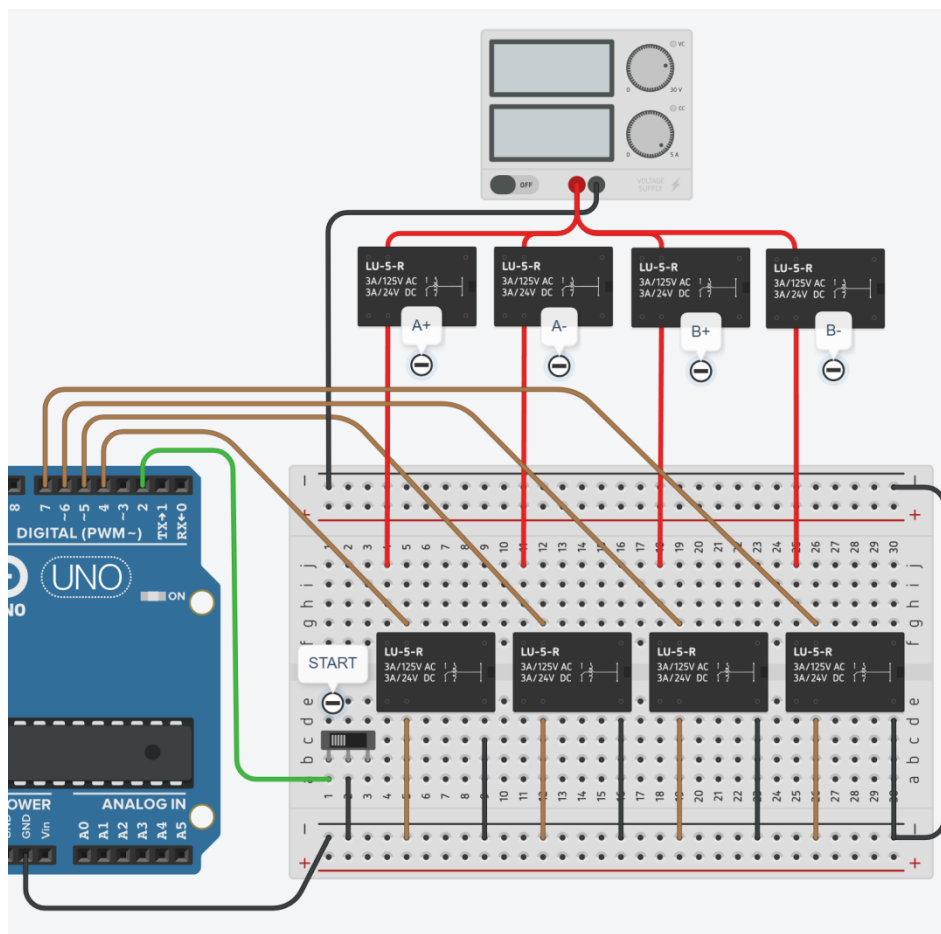
Realizzare la sequenza pneumatica manuale **A+A-B+B-** utilizzando lo shield relè per comandare le elettrovalvole a 24V. Non utilizzare i finecorsa. Mantenere lo stato dello stelo per 2 secondi. L'avvio avviene premendo Start.



24V



Simulazione Thinkercad



## Codice

```
long t0=0;
int n=0;

void setup() {
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

  //parto con A- attivo --> stelo dentro
  digitalWrite(4, LOW); digitalWrite(5, HIGH);
  //parto con B- attivo --> stelo dentro
  digitalWrite(6, LOW); digitalWrite(7, HIGH);
}

void loop() {
  int sensorVal = digitalRead(2);

  // Start
  if (sensorVal == LOW && n<1) {
    n= n+1;
    // A+
    Serial.println("A+");
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(2000);
    // A-
    Serial.println("A-");
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    delay(2000);
    // B+
    Serial.println("B+");
    digitalWrite(6, HIGH);
    digitalWrite(7, LOW);
    delay(2000);
    // B-
    Serial.println("B-");
    digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
    delay(2000);
  }

  if (sensorVal == HIGH) {
    if (millis() - t0 > 1000) {
      Serial.println("riposo");
      digitalWrite(4, LOW); digitalWrite(5, HIGH);
      digitalWrite(6, LOW); digitalWrite(7, HIGH);
      n=0;
      t0= millis();
    }
  }
}
```



## ELENCO DEI PIÙ COMUNI TRASDUTTORI.

Fenomeno da misurare	Trasduttori impiegati
Temperatura	Termocoppie Termistori (PTC, NTC) Rivelatori resistivi di temperatura Trasduttori basati su circuiti integrati
Luce	Fotocellule, fotodiodi e fototransistors CCD
Suono	Microfoni
Forza e pressione	Estensimetri Trasduttori piezoelettrici Trasduttori piezoresistivi
Posizione, spostamento e rotazione	Potenzimetri Trasduttori induttivi di spostamento Encoder ottici Giroscopi ottici
Presenza o prossimità di un oggetto	Trasduttori induttivi e capacitivi Trasduttori magnetici Trasduttori ad ultrasuoni (Sonar)
Flusso di fluidi	Misuratori diretti di portata Misuratori di velocità di un fluido
Livello pH	Sonde pH

## PROPRIETA' GENERALI DEI SENSORI

Le principali caratteristiche che definiscono la qualità del segnale di un sensore (statiche e dinamiche) sono:

### Range (Campo di Misura):

L'intervallo tra il valore minimo e massimo che il sensore può misurare (es. da -50°C a +150°C).

### Accuratezza (Accuracy):

La differenza massima tra il valore misurato e il valore reale. Indica quanto la misura è "vera".

### Precisione (Ripetibilità):

La capacità del sensore di fornire lo stesso risultato misurando più volte la stessa grandezza nelle stesse condizioni. *Un sensore può essere preciso ma non accurato (se non è calibrato).*

### Risoluzione:

La più piccola variazione della grandezza fisica che il sensore è in grado di rilevare.

### Linearità:

Quanto la curva di uscita del sensore si avvicina a una linea retta ideale.

Una bassa linearità richiede algoritmi di correzione complessi.

### Isteresi:

La differenza nel valore di uscita quando si misura la stessa grandezza in fase crescente o decrescente.

### Deriva (Drift):

La variazione del segnale nel tempo (a causa dell'invecchiamento) o a causa della temperatura, senza che cambi la grandezza misurata.

### Tempo di risposta (response time):

E' il tempo necessario affinché l'uscita del sensore raggiunga una certa percentuale (solitamente il **63.2%** o il **90%**) del suo valore finale stabile in seguito a una variazione "a gradino" dell'ingresso.

## TEMPO DI RISPOSTA DEI PRINCIPALI SENSORI

In ambito industriale, il tempo di risposta è il tempo necessario affinché l'uscita del sensore raggiunga una certa percentuale (solitamente il **63.2%** o il **90%**) del suo valore finale stabile in seguito a una variazione "a gradino" dell'ingresso. Questo concetto è spesso legato alla **Costante di Tempo**.

## SENSORI OTTICI E FOTOELETTRICI

Questi sensori lavorano basandosi sulla luce e sull'elettronica veloce. Non hanno parti meccaniche in movimento e non dipendono da trasferimenti termici.

- **Tecnologia:** Fotocellule, Barriere laser, Fibre ottiche, Sensori di visione.
- **Tempo di Risposta Tipico:** 10 us – 1 ms (microsecondi o millisecondi).
- **Applicazione:** Conteggio pezzi su nastri trasportatori veloci, rilevamento tacche di stampa, sicurezza macchine.

## SENSORI DI PRESSIONE

La velocità dipende dalla tecnologia interna.

- **Piezolettrici:** Velocissimi, reagiscono alla deformazione cristallina quasi istantanea.  
*Tempo: < 0.1 ms* (ideali per rilevare picchi di pressione o esplosioni).
- **Piezoresistivi / Estensimetrici:** Standard industriale.  
*Tempo: 1 ms – 5 ms.*
- **Capacitivi:** Generalmente usati per basse pressioni, leggermente più lenti.

## SENSORI DI PROSSIMITÀ (INDUTTIVI E CAPACITIVI)

Utilizzati per rilevare la presenza di oggetti (metalli o altri materiali) a breve distanza.

- **Induttivi (Metallo):** La velocità è legata alla frequenza di commutazione del campo magnetico.  
*Tempo: 0.2 ms – 5 ms* (Frequenze fino a 5 kHz).
- **Capacitivi:** Leggermente più lenti degli induttivi a causa della fisica del campo elettrico.  
*Tempo: 2 ms – 10 ms* (Frequenze 50-100 Hz).

## SENSORI DI POSIZIONE E DISTANZA

- **Encoder Ottici:** Risposta istantanea, limitata solo dalla frequenza di lettura del controllore.
- **Laser di distanza (Time of Flight):** Molto veloci (**1 ms – 10 ms**).
- **Ultrasuoni:** Più lenti perché il suono deve viaggiare attraverso l'aria, colpire l'oggetto e tornare indietro (la velocità del suono è "lenta" rispetto alla luce).  
*Tempo: 20 ms – 500 ms* (dipende dalla distanza massima impostata).

## SENSORI DI TEMPERATURA (I PIÙ LENTI)

La temperatura è una grandezza che possiede una "inerzia termica". Il calore deve fisicamente penetrare nel materiale del sensore.

- **Termocoppie (a giunto esposto):** Le più veloci a contatto.  
*Tempo: 0.1 s – 1 s.*
- **Termoresistenze (PT100) / Termocoppie (corazzate):** Devono scaldare la guaina protettiva in metallo.  
*Tempo: Da 2 secondi a decine di secondi.*
- **Pirometri Ottici (Infrarossi):** L'unica eccezione veloce per la temperatura. Non toccano l'oggetto, leggono la radiazione.  
*Tempo: 10 ms – 500 ms.*

# SENSORI DI PROSSIMITA'

Il sensore di prossimità si riferisce a un tipo di sensore senza contatto che emette un campo di energia per rilevare la presenza o l'assenza di un oggetto. Può trattarsi di un sensore di prossimità per dispositivi mobili, un sensore di prossimità per sistemi di sicurezza o i diversi tipi di sensori di prossimità utilizzati nell'automazione industriale.

A causa della loro natura senza contatto, i sensori di prossimità presentano molti vantaggi rispetto ai sensori di contatto. Sono affidabili, durevoli e richiedono poca manutenzione. Inoltre non producono alcun movimento fisico o trasferimento di calore all'oggetto target e possono essere utilizzati in ambienti difficili.

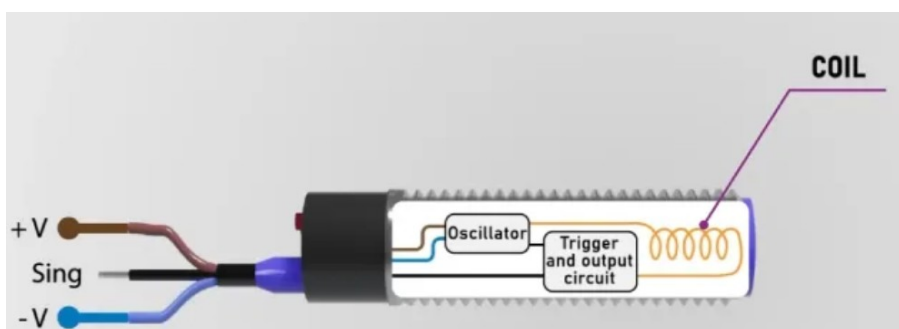
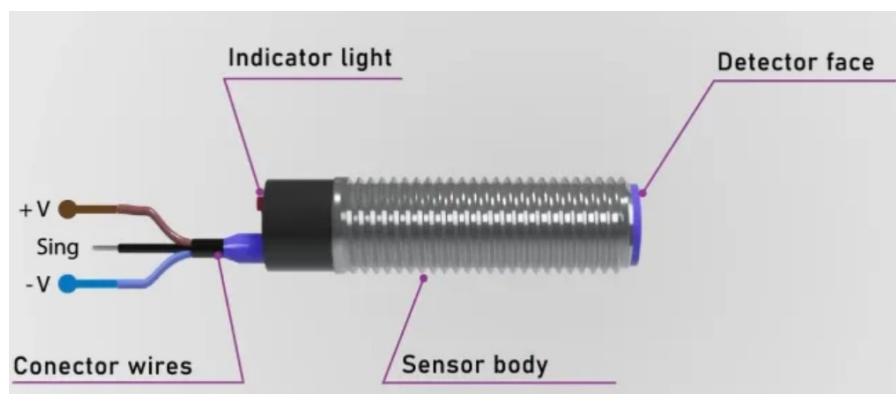
Un sensore di prossimità può utilizzare onde elettromagnetiche, luce o ultrasuoni per rilevare l'oggetto. Alcuni rilevano solo i metalli, mentre altri possono individuare sia bersagli metallici che non metallici.

## TIPI DI SENSORI DI PROSSIMITÀ

Sulla base delle diverse forme di tecnologie di rilevamento, i sensori di prossimità sono classificati in cinque categorie:

- Sensore di prossimità induttivo;
- Sensore di prossimità capacitivo;
- Sensore di prossimità a ultrasuoni;
- Sensore di prossimità magnetico;
- Sensore di prossimità ottico.

## 1. SENSORE DI PROSSIMITÀ INDUTTIVO



Il sensore di prossimità induttivo è chiamato così per utilizzare i principi dell'induttanza per rilevare la presenza di un bersaglio metallico, senza effettuare alcun contatto fisico. Uno dei tipi più comuni di questo sensore è il sensore di prossimità a correnti parassite.

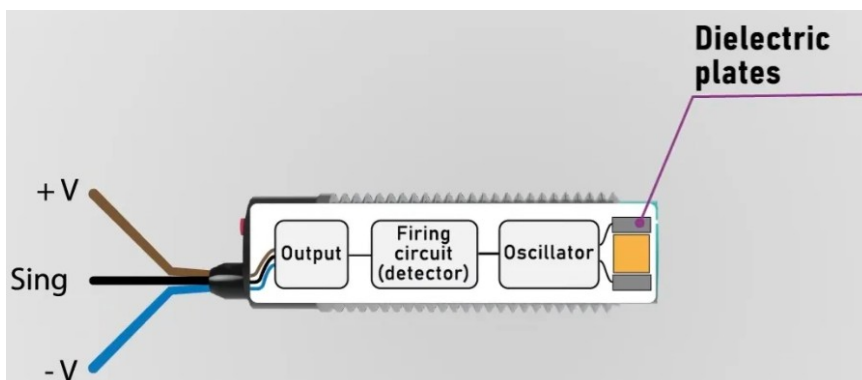
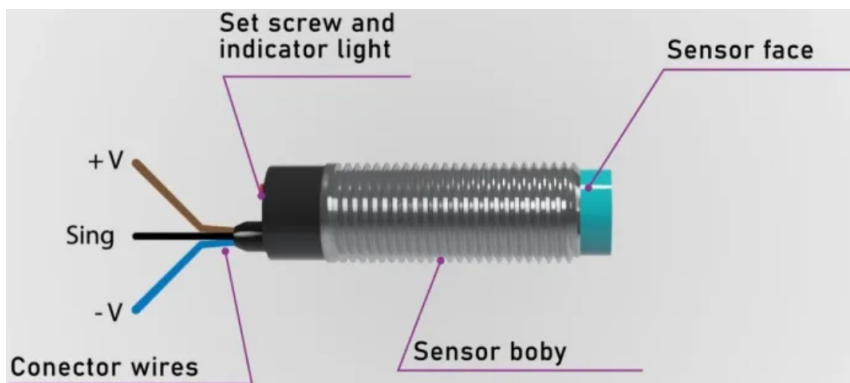
Questi sensori sono costituiti principalmente da un oscillatore, una bobina attorno a un nucleo di ferrite e un trigger di Schmitt. Ecco come funziona un sensore di prossimità induttivo:

- Durante il funzionamento, l'oscillatore genera una corrente alternata che produce un campo elettromagnetico alternato attorno alla bobina.
- Questo campo si irradia dalla bobina per formare la zona di rilevamento.
- Se un oggetto metallico entra in questa zona di rilevamento, il campo magnetico oscillante induce correnti elettriche nel suo corpo. Queste sono chiamate correnti parassite.
- Le correnti parassite iniziano quindi a produrre un campo magnetico alternato, interferendo con il campo oscillante originale del sensore e modificandone le proprietà.
- Questa modifica attiva il trigger di Schmitt e il sensore è in grado di rilevare.

Si noti che questi tipi di sensori di prossimità non possono rilevare oggetti non metallici in quanto tali materiali non producono correnti parassite.

**Applicazioni:** Uno dei vantaggi dei sensori induttivi è la loro capacità di operare in ambienti contaminati: sono resistenti alla presenza di olio, sporcizia e persino umidità. I sensori di prossimità induttivi sono quindi ampiamente utilizzati nelle applicazioni industriali, automobilistiche e di macchine utensili.

## 2. SENSORE DI PROSSIMITÀ CAPACITIVO



Il sensore di prossimità capacitivo utilizza un campo elettrico per rilevare la presenza di un oggetto target.

Si tratta fondamentalmente di un condensatore aperto la cui altra piastra è sostituita dal bersaglio, mentre l'aria tra la piastra del sensore e il bersaglio forma il dielettrico.

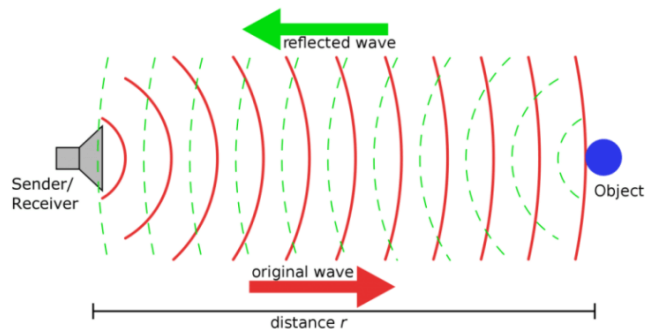
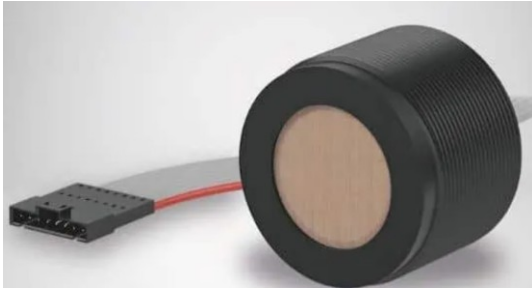
Ecco come funziona un sensore di prossimità capacitivo:

- Quando il bersaglio entra nel raggio del sensore, forma una capacità con la sua piastra di rilevamento, che aumenta man mano che l'oggetto si avvicina.
- Questa azione modifica il valore di capacità del circuito, che a sua volta produce un segnale elettrico utilizzato per rilevare la presenza.

- Il sensore capacitivo può rilevare sia metalli che non metalli. Questi possono essere polveri, granuli e liquidi o anche oggetti solidi.
- Poiché il principio di funzionamento del sensore di prossimità capacitivo si basa sull'aumento graduale della capacità, la sua velocità di rilevamento è generalmente inferiore a quella dei sensori induttivi.

**Applicazioni:** I sensori di prossimità capacitivi sono utilizzati in un'ampia gamma di applicazioni, compresi i processi di produzione di alimenti e bevande, il rilevamento del livello, la movimentazione dei materiali, i sistemi di controllo dell'automazione e altri ambienti industriali. Nel mondo dell'elettronica, questo è il tipo di sensore di prossimità per applicazioni di rilevamento di telefoni cellulari o tablet.

### 3. SENSORE DI PROSSIMITÀ A ULTRASUONI



Il sensore a ultrasuoni è leggermente diverso dai sensori induttivi e capacitivi. Questi tipi di sensori di prossimità funzionano emettendo onde ultrasoniche o onde sonore con una frequenza superiore al limite superiore dell'udito umano, che è di circa 20 kHz. Il sensore a ultrasuoni è costituito da queste parti:

- trasmettitore,
- ricevitore,
- processore di segnale,
- amplificatore
- modulo di alimentazione

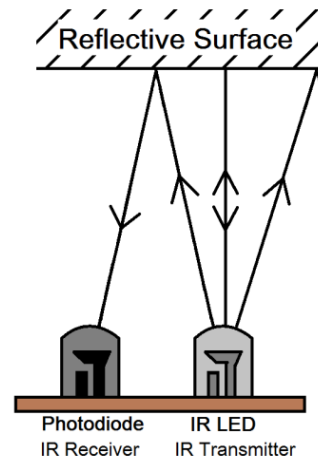
Il sensore funziona funzionando inviando impulsi sonori ad alta frequenza:

- Quando le onde sonore incontrano un ostacolo, rimbalzeranno al ricevitore.
- Il ricevitore utilizza quindi queste informazioni per determinare la presenza e la distanza tra l'oggetto e il sensore.

I sensori di prossimità a ultrasuoni offrono un'elevata velocità di rilevamento, anche per piccoli oggetti, e hanno un ampio raggio di rilevamento. Possono anche rilevare bersagli solidi e liquidi nella loro zona di rilevamento.

**Applicazioni:** I sensori di prossimità a ultrasuoni sono utilizzati principalmente nella robotica, nei sistemi di rilevamento ed evitamento degli ostacoli, nell'automazione industriale, nei sensori di parcheggio, ecc. Inoltre, questi tipi di sensori possono anche rilevare le vibrazioni, rendendoli adatti per le applicazioni di monitoraggio delle vibrazioni.

#### 4. SENSORE DI PROSSIMITÀ OTTICO



Il sensore ottico di prossimità funziona secondo il principio della riflessione della luce (In genere si utilizza infrarosso). Il sensore emette luce verso un oggetto target e misura la quantità di luce riflessa su di esso.

Generalmente i sensori ottici di prossimità vengono utilizzati in combinazione con un LED a infrarossi o un diodo laser. Ecco come funziona un sensore di prossimità ottico:

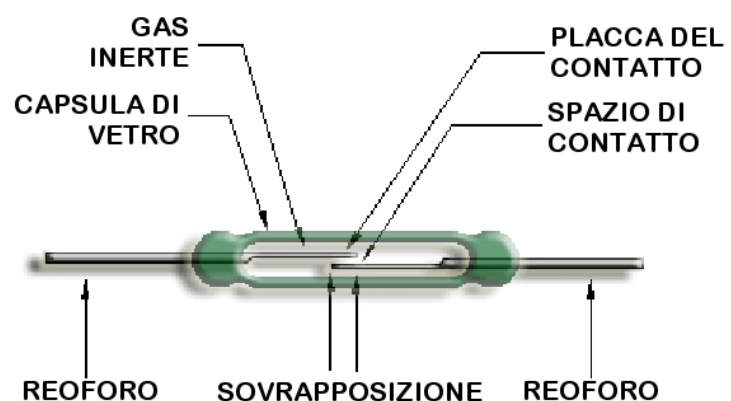
- Quando un oggetto target è abbastanza vicino al sensore, riflette parte dell'energia luminosa verso il rilevatore.
- Questo verrà poi amplificato e utilizzato come segnale elettrico per rilevare la presenza dell'oggetto.

I sensori di prossimità ottici di solito non sono influenzati da polvere, sporco o umidità.

Hanno anche un'alta risoluzione e possono rilevare con facilità anche oggetti molto piccoli nel loro raggio di rilevamento.

**Applicazioni:** I sensori di prossimità ottici sono ampiamente utilizzati per il rilevamento del livello nei liquidi, il rilevamento della posizione nelle macchine e nei processi di automazione. Sono anche utilizzati come metal detector nei sistemi di sicurezza e nei dispositivi di controllo degli accessi. Questi tipi di sensori di prossimità hanno trovato applicazione anche nei sistemi di navigazione per auto o droni.

#### 5. SENSORE DI PROSSIMITÀ MAGNETICO



Il sensore di prossimità magnetico funziona utilizzando l'attrazione tra il magnete e l'oggetto target per rilevare la presenza di un oggetto. Uno dei vantaggi di questi sensori è che possono rilevare bersagli magnetici attraverso materiali non metallici, come plastica e legno. Hanno anche un raggio di rilevamento abbastanza ampio.

Questo sensore può essere di diversi tipi e il suo funzionamento dipende dal tipo di tecnologia utilizzata.

I tipi di sensori di prossimità magnetici includono:

- tipo reed,
- switch,
- a riluttanza,
- magnetoresistivo,
- ad effetto Hall
- ad effetto GMR (magnetoresistivo gigante)

**Sensore reed switch:** il sensore di prossimità reed magnetico è costituito da due contatti ferromagnetici alloggiati in un involucro di vetro sigillato. Quando un magnete viene avvicinato al sensore, provoca la chiusura delle lamelle e completa un circuito.

**Sensore a riluttanza variabile:** questo sensore è costituito da un magnete permanente e una bobina captatrice attorno a un'espansione polare magnetica e funziona misurando le variazioni di riluttanza.

**Sensore ad effetto Hall:** questo tipo di sensore funziona misurando la resistenza alle variazioni di un materiale ferromagnetico quando ad esso viene applicato un campo magnetico.

**Sensore magnetoresistivo:** questi sensori funzionano rilevando le variazioni di resistenza elettrica causate dalla presenza di un magnete nelle vicinanze.

**Sensore GMR (gigante magneto resistivo):** questo tipo di sensore è composto principalmente da piastre ferromagnetiche separate da un distanziatore non magnetico. Quando un magnete viene avvicinato al sensore, provoca una variazione di resistenza che attiva un circuito.

**Applicazione:** i sensori di prossimità magnetici sono comunemente utilizzati come dispositivi di rilevamento della posizione in macchinari industriali, componenti automobilistici come l'albero motore e altri macchinari. Altre applicazioni includono la robotica e i sistemi di sicurezza. Questi tipi di sensori di prossimità offrono un funzionamento semplice e possono essere utilizzati in ambienti difficili come situazioni contaminate o con vibrazioni elevate.

**Tabella che riassume i vantaggi e gli svantaggi delle diverse tipologie di prossimità sensori:**

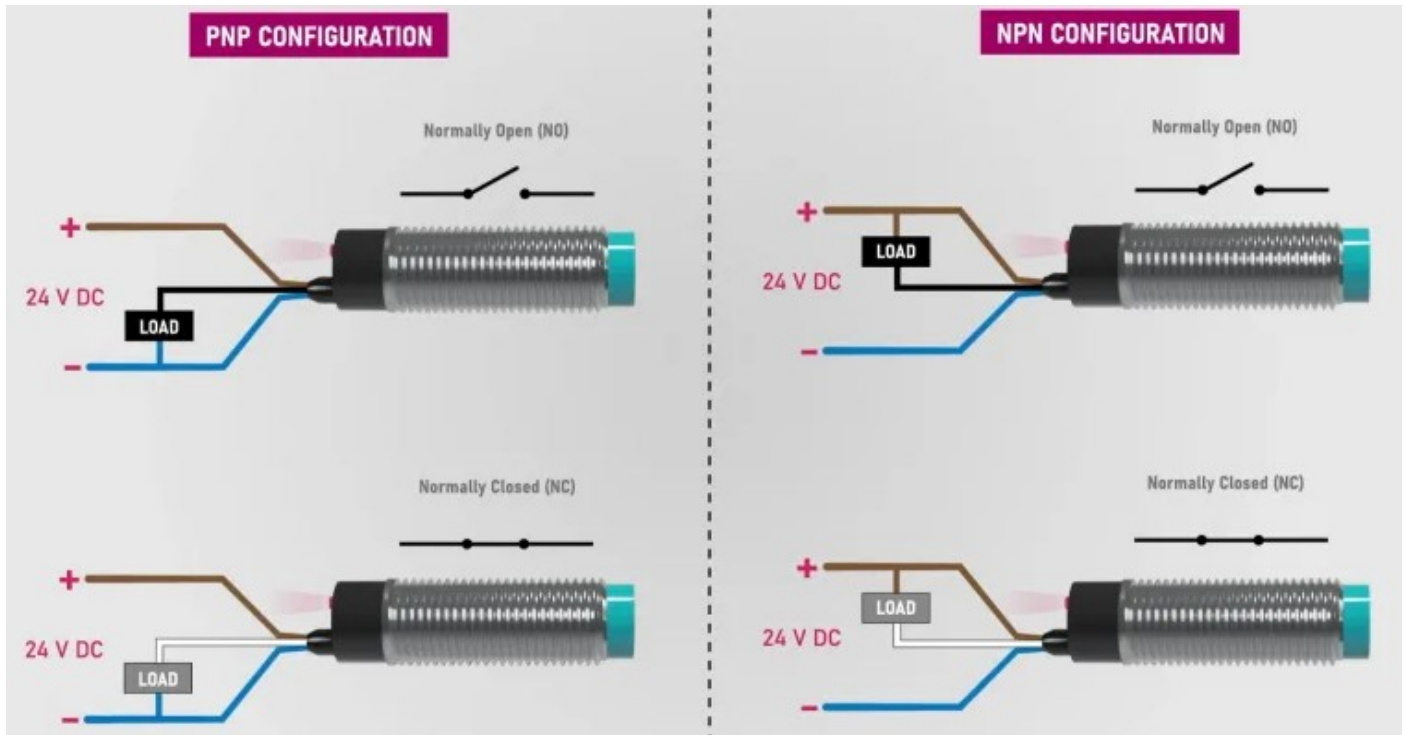
Digitare	vantaggio	difetto	Applicazioni tipiche
<b>Tipo induttivo</b>	Basso costo, velocità di risposta rapida, forte capacità anti-interferenza	Può rilevare solo oggetti metallici	Conteggio oggetti, controllo della posizione, rilevamento della presenza
<b>Tipo capacitivo</b>	Può rilevare oggetti metallici e non metallici, insensibile ai cambiamenti ambientali; velocità di risposta media	La distanza di rilevamento è fortemente influenzata dal materiale target	Rilevamento del livello del liquido, rilevamento del materiale, interruttore di prossimità
<b>Ultrasonico</b>	Può rilevare vari materiali, indipendentemente dal colore o dalla trasparenza, distanza misurabile; velocità di risposta bassa	Velocità di risposta lenta e suscettibilità al rumore ambientale	Misurazione della distanza, evitamento degli ostacoli, controllo del livello del liquido
<b>Tipo fotoelettrico</b>	Lunga distanza di rilevamento, elevata precisione e velocità di risposta rapida	Vulnerabile alle interferenze della luce ambientale e richiede una linea di vista libera	Conteggio oggetti, rilevamento della posizione, scansione di codici a barre

## CONFIGURAZIONI E APPLICAZIONI DEI SENSORI

Sia i sensori induttivi che quelli capacitivi offrono varie configurazioni per adattarsi alle diverse applicazioni industriali.

Questi sensori possono essere schermati o non schermati: quelli schermati consentono il montaggio a filo e quelli non schermati offrono un'area di rilevamento più ampia.

Sono disponibili in configurazioni normalmente aperte o normalmente chiuse, nonché in tipi di uscita NPN o PNP per la compatibilità con diversi sistemi di controllo.



## CONTATTO DRY (ASCIUTTO) E WET (BAGNATO)

Nel mondo dei sistemi elettrici ed elettronici, i termini "contatto a secco" e "contatto bagnato" sono frequentemente usati.

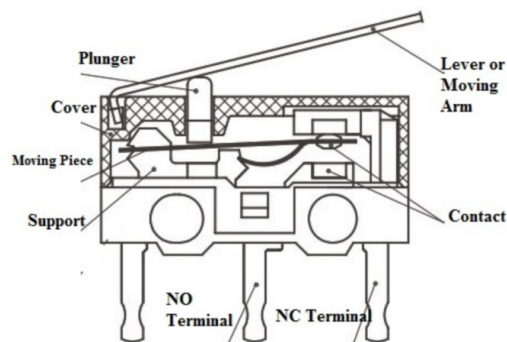
Il **contatto asciutto**, noto anche come "contatto senza tensione", è un tipo di contatto di relè o interruttore che non trasporta alcuna tensione o corrente dalla sua fonte di controllo. Richiede una fonte di alimentazione esterna per operare.

Essenzialmente, funziona come un semplice interruttore on/off, controllando il flusso di elettricità senza essere direttamente collegato alla fonte di alimentazione stessa. I contatti asciutti sono tipicamente utilizzati in applicazioni in cui un dispositivo o sistema deve essere controllato senza trasferire energia elettrica attraverso il componente di controllo.

Ad esempio, possono segnalare a un altro circuito di attivarsi o disattivarsi senza influenzarne le caratteristiche di tensione o corrente.

Il **contatto bagnato** invece trasporta la tensione internamente (ha una sua alimentazione dedicata) e può quindi a sua volta alimentare il dispositivo o il circuito collegato (es. una luce o un cicalino).

### Esempio di contatto asciutto e bagnato.



**METALWORK**  
PNEUMATIC

**W0950044180 reed**  
Batch: CO190238

- sensori magnetici di prossimità
- magnetic proximity switches
- magnetische nahungssensoren
- capteurs magnetiques de proximite

**3± max 30 V ac/dc**  
**200 mA 6 W**

**CE**

UL-CABLE  
2 FILI  
2 WIRES  
2 DRÄTHER  
2 FILS

marrone  
blu

BL- BW+

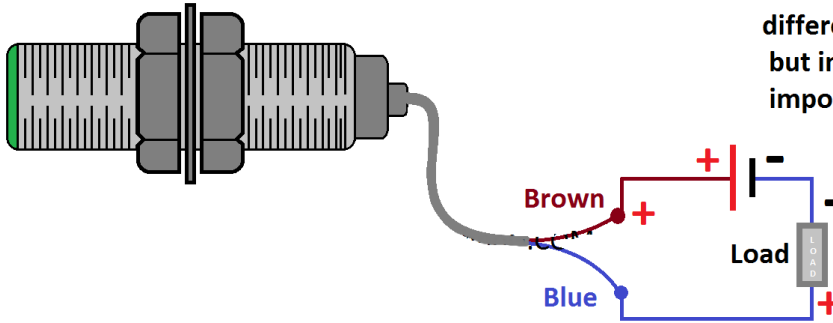
www.metalwork.it metalwork@metalwork.it

**NOTA:** per applicazioni in ambienti industriali e/o rumorosi è consigliabile l'uso di sensori a 3 fili dove oltre ai due fili di alimentazione (Vcc e massa) è presente un terzo filo che trasporta il segnale che risulterà quindi meno disturbabile.

SENSORE A 2 FILI

2 Wire Type

Sensitive Face

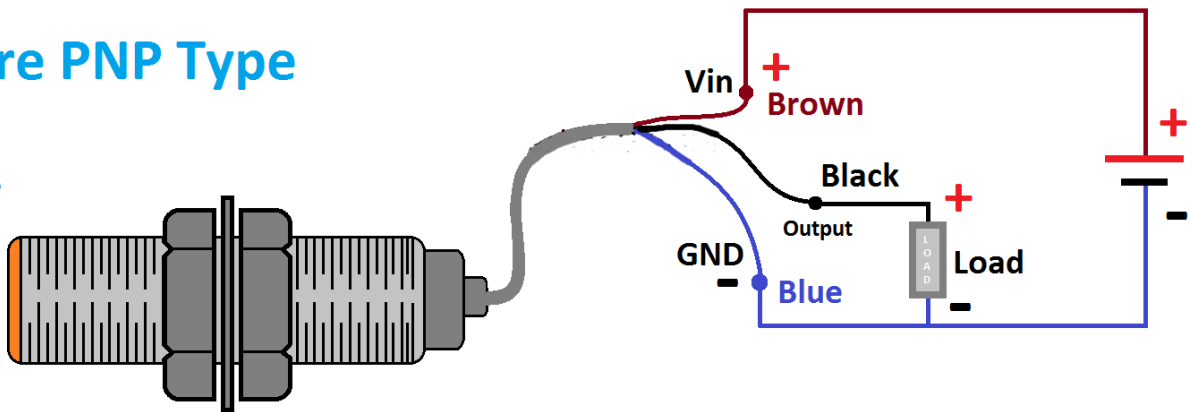


Here, a DC 2 Wire Type Sensor is shown, For AC 2-wire proximity Sensor, the connection also same just difference is here we care about polarity but in case of AC, Polarity not so much important Just connect the sensor in series with load and power supply

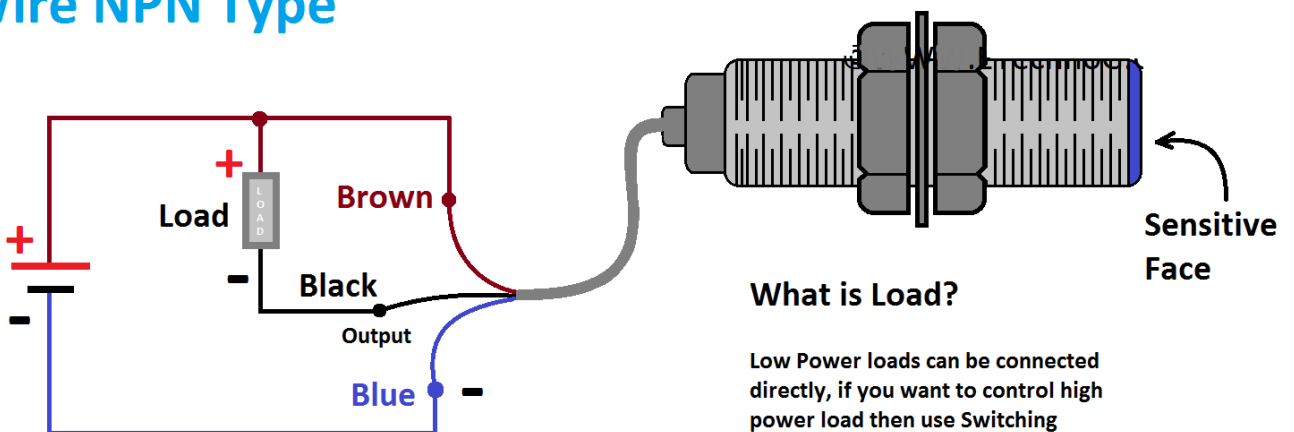
SENSORE A 3 FILI

3 Wire PNP Type

Sensitive Face



3 Wire NPN Type

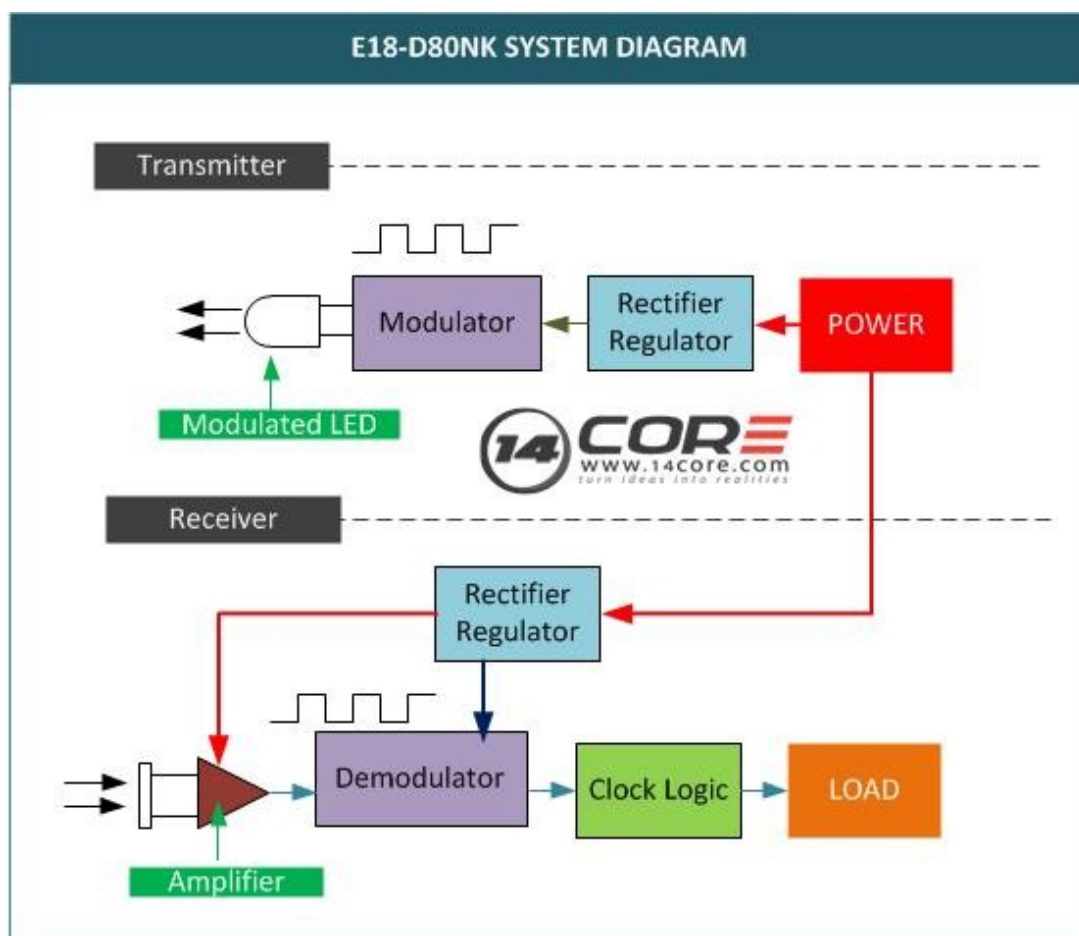


What is Load?

Low Power loads can be connected directly, if you want to control high power load then use Switching devices such as Relay, Contactors with the sensor output

## SENSORE A INFRAROSSI A TRE FILI E18-D80NK (NPN)

Si tratta di un fotoriflettore ad alta sensibilità per rilevare la distanza, da 3 cm a 80 cm. Quando l'infrarosso emesso dall'emettitore viene riflesso su una superficie bloccata, il fototransistor capta il segnale per il calcolo della distanza. Questo dispositivo è dotato di un potenziometro integrato per regolare la portata, rendendolo semplice e intuitivo da usare. L'utilizzo ideale di questo dispositivo è in ambito robotico, media interattivi, industriale e automobilistico, ecc.



Il sensore generalmente presenta tre fili di colore marrone (Vcc), blu (massa) e nero (segnale).

In alcuni casi si trova la combinazione di colore rosso (Vcc), verde (massa) e giallo (segnale).

Il sensore a infrarossi viene alimentato direttamente da Arduino (filo rosso).

Il cavo verde va a massa e il cavo giallo del segnale viene connesso al 10 di arduino (modalità INPUT).

Quando non è presente il pezzo viene letto 1 (5V) mentre quando è presente viene letto 0 (led posteriore acceso).

Il sensore di prossimità a infrarossi E18D80NK permette di rilevare degli ostacoli con una portata regolabile da 3 cm a 80 centimetri. Include un trasmettitore e un ricevitore a infrarossi, tutto in un'unica unità.

Il trasmettitore a infrarossi emette un segnale a infrarossi modulato che viene riflesso dagli oggetti lungo il percorso di riflessione e quindi interpretato dal ricevitore. Il sensore è meno influenzato dalla luce solare grazie alla sua luce a infrarossi.

Il sensore IR E18-D80 è ampiamente utilizzato nei robot per evitare ostacoli e nelle linee di assemblaggio industriali, nel parcheggio in retromarcia e in numerose altre applicazioni che richiedono automazione. Il raggio di rilevamento può essere modificato a seconda dello scopo di utilizzo tramite la vite multigiro situata sul retro del dispositivo.

L'uscita del segnale di commutazione cambia a seconda del rilevamento di un ostacolo.

Rimane alta quando non vengono rilevati ostacoli e diminuisce in caso di ostacoli.

La sonda ha una luce rossa posizionata dietro la sonda stessa che si accende quando rileva un ostacolo.

Il sensore E18 funziona a 5 V e può consumare da 5 mA a 30 mA di corrente, senza carico.

## Specifiche e caratteristiche

- Tensione di ingresso: 5 V CC
- Consumo di corrente: >25mA (min) > 100mA (max)
- Dimensioni: 1,7 cm (diametro) x 4,5 cm (lunghezza)
- Lunghezza del cavo: 45 cm
- Rilevamento di oggetti: trasparenti o opachi
- Tipo riflettente diffuso
- Campo di rilevamento: 3 cm-80 cm
- Uscita NPN (normalmente alta)
- Temperatura ambiente: -25 °C ~ 55 °C



La tabella seguente mostra la configurazione dei pin del sensore di prossimità IR. Ha 3 fili di uscita, generalmente codificati a colori: rosso per VCC, verde per la massa e giallo per l'uscita digitale.

Pin Type/Wire color	Pin Description
VCC (Marrone o Rosso)	Voltage input(+5V)
GND (Blu o Verde)	Ground terminal
Digital pin (Nero o giallo)	Digital signal output

## Cablaggio sensore di tipo NPN



Mettendo una resistenza da 300 ohm in serie ad un led come carico si noterà l'accensione del led esterno quando il pezzo è presente

## CABLAGGIO SENSORE DI DISTANZA A INFRAROSSI E18-D80NK (NPN)

Il sensore fornisce una uscita ALTA (5V) quando non c'è il pezzo e BASSA (massa) quando viene rilevato.

Essendo il sensore a 5V può essere connesso direttamente ad Arduino tramite i pin 5V e GND.

Il cavo nero (segnale) può essere collegato ad un pin digitale (INPUT) dove potrà essere rilevata la tensione di 5V o 0V.

Per sicurezza sarebbe meglio collegare una resistenza di 10K per limitare la corrente in ingresso al pin in ogni situazione.

### CODICE Arduino

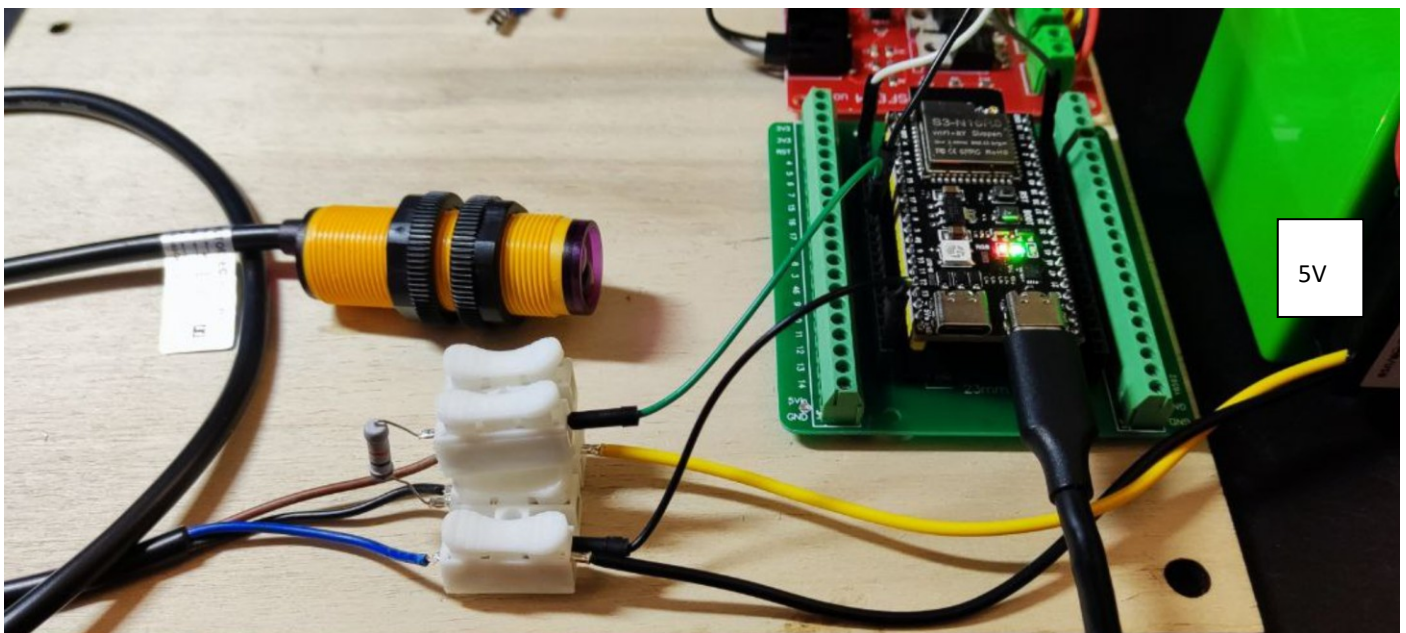
```
void setup() {
  Serial.begin(9600); //Start serial communication boud r
  pinMode(2,INPUT); //Pin 2 as signal input
}

void loop() {
  while(1) {
    delay(500);
    if(digitalRead(5)==LOW) {
      // If no signal print collision detected
      Serial.println("Collision Detected.");
    }
    else {
      // If signal detected print collision detected
      Serial.println("No Collision Detected.");
    }
  }
}
```



Se si utilizza un ESP32 (logica a 3.3V) è obbligatoria una alimentazione esterna di 5V (cavi giallo-nero dalla batteria).

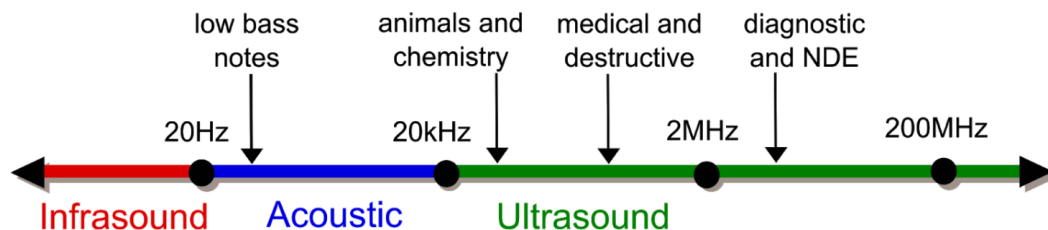
L'uscita del sensore a 5V (cavo nero) non sarebbe adatta agli ingressi GPIO del micro (max 3.3) e sarebbe meglio ridurla a 3.3V (ad es. con un partitore di tensione). Comunque mettendo una resistenza di 10K in ingresso al pin digitale in modo da limitare la corrente non dovrebbero esserci particolari problemi per un uso sporadico e di prototipazione (i pin di un ESP32 sono maggiormente protetti rispetto a quelli Arduino).



*NB: collegare la massa del micro a quella della batteria*

## SENSORE A ULTRASUONI

Gli ultrasuoni sono onde sonore con frequenze superiori a quelle udibili dall'orecchio umano: stiamo quindi parlando di frequenze che superano i 20 kHz e che trovano impiego per lo più in campo medico ed industriale.



utilizzare un sensore ad ultrasuoni come misuratore di distanze, in attività didattiche dove non sia richiesta un'elevata "qualità della misura".

Lasciamo quindi a successivi sviluppi la ricerca di misure precise ed accurate, per le quali dovremmo considerare la velocità istantanea del suono che è influenzata, principalmente, dalla temperatura e dall'umidità relativa del mezzo.

Per quanto premesso possiamo assumere come costante il valore della velocità del suono in un determinato mezzo, in particolare l'aria, dove le onde sonore viaggiano a 343,8 m/s a 20°C.

Anche gli ultrasuoni, come tutte le onde, sono soggetti a fenomeni di riflessione. Questa caratteristica ci permette di utilizzare il sensore per rilevare misure di distanza tra la sorgente emettitrice del segnale sonoro e l'oggetto colpito.

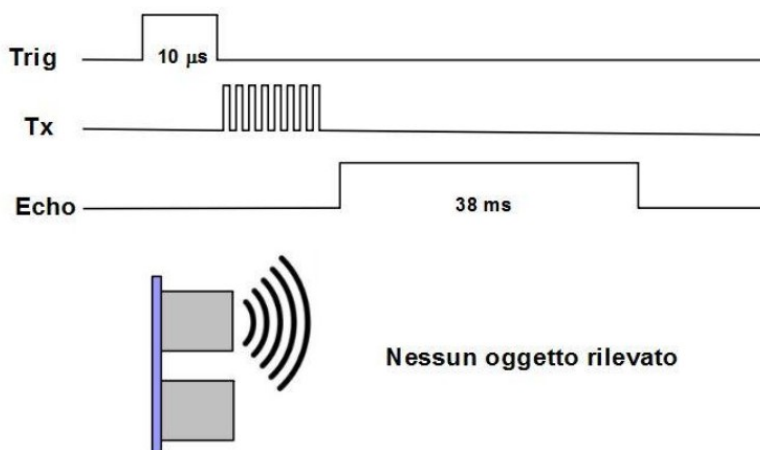
## FUNZIONAMENTO DEL SENSORE PER ARDUINO

Un impulso di tensione (3.3 o 5V) di almeno 10  $\mu$ s (microsecondi) di durata viene applicato al pin Trigger. Si genera così un treno di 8 impulsi ultrasonici a 40 KHz che si allontanano dal sensore viaggiando nell'aria circostante.

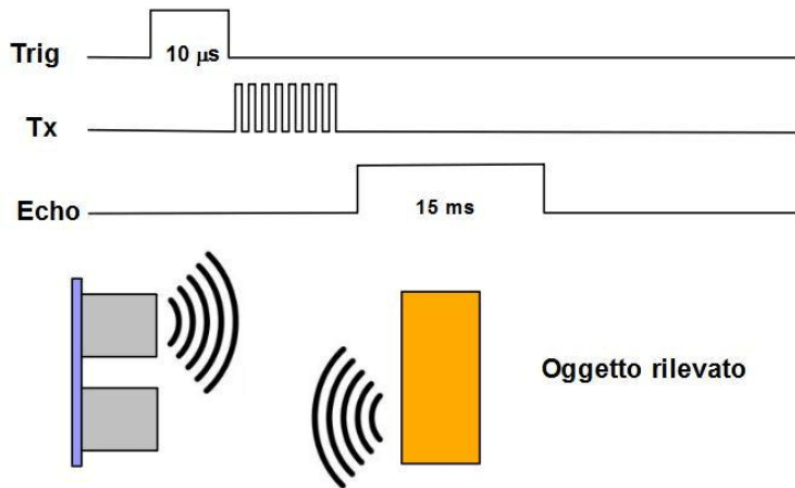
Si ottengono misure più accurate se l'ostacolo si trova di fronte al sensore o in un ipotetico settore circolare di 30° d'ampiezza (15° da ambo i lati rispetto alla direzione frontale).

Il segnale sull'Echo intanto diventa alto ed inizia la registrazione del tempo di ritorno in attesa dell'onda riflessa.

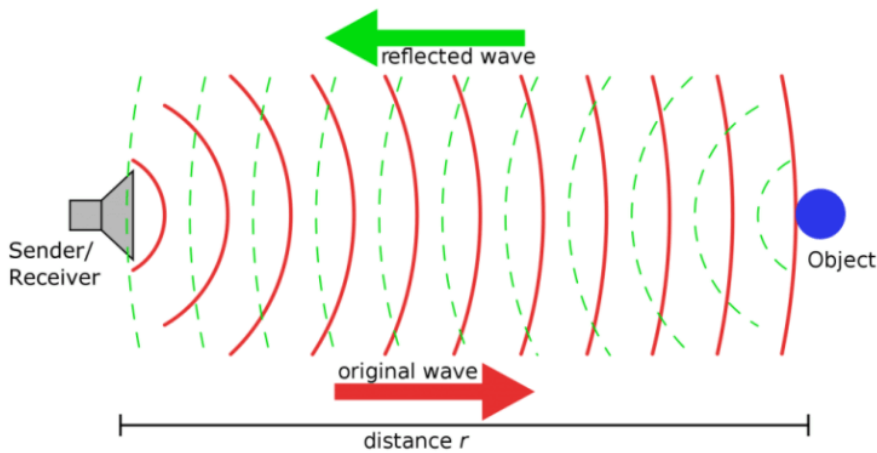
Se l'impulso non viene riflesso il segnale su Echo torna basso dopo 38 ms (millisecondi) e va interpretato come assenza di ostacolo. Ricordiamo l'HC-SR04 è in grado di misurare distanze comprese tra i 2 e i 400 cm corrispondenti, per il limite massimo, a circa 23 ms di durata del segnale su Echo.



Se invece il treno di onde ultrasoniche viene riflesso all'indietro da un oggetto, il segnale sul pin Echo diventa basso e contestualmente termina il rilievo della sua durata.



Il tempo ottenuto servirà per calcolare la distanza dell'oggetto: bisogna però tenere presente che l'onda ha percorso per due volte quella distanza, quando emessa verso l'oggetto e dopo la riflessione verso il sensore. Bisognerà quindi dividere per due la distanza calcolata con questo tempo.

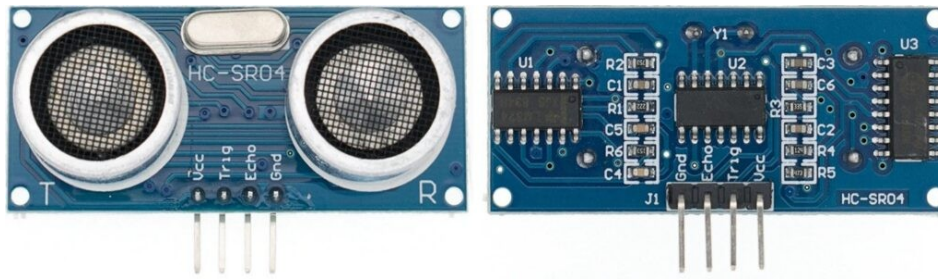


La velocità del suono nell'aria è di circa 343 m/s.

La funzione **pulseIn()** introdotta nello sketch ci permette di ottenere la durata dell'impulso ALTO sul pin Echo in microsecondi.

$$343 \frac{m}{s} = \frac{34300}{1000000} \frac{cm}{\mu s} = 0,0343 \frac{cm}{\mu s}$$

$$distanza [cm] = \left( 0,0343 \left[ \frac{cm}{\mu s} \right] \cdot durata ALTO su Echo [\mu s] \right) \div 2$$



Le principali caratteristiche sono:

- Tensione di lavoro: 3 – 5.5 Vdc.
- Corrente assorbita: 3 mA circa.
- Frequenza di lavoro: 40 KHz.
- Distanza min: 2 cm.
- Distanza max: 450 cm.
- Risoluzione: 3 mm.
- Angolo di misura: 15 – 20°.
- Ingresso: Trigger 10us Impulso TTL.
- Uscita: Echo segnale PWM TTL

Ha 4 pin:

- Vcc – viene collegato alla tensione di alimentazione da 3 a 5.5V.
- Trig – è il pin “Trigger” che deve essere portato alto per inviare il segnale ad ultrasuoni.
- Echo – è il pin che produce un impulso che si interrompe quando viene ricevuto il segnale riflesso dall’ostacolo.
- GND – viene collegato GND.

### PRINCIPIO DI FUNZIONAMENTO:

Il sensore HC-SR04 emette un treno di impulsi ad ultrasuoni (ne vengono emessi 8 quando verrà portato per 10 microsecondi a stato alto il suo pin Trigger). Gli impulsi si propagano nell’ambiente circostante e, se incontrano uno ostacolo, tornano indietro verso il sensore che li ha emessi. Quando il sensore “rileva” il ritorno dell’impulso sonoro porterà a stato basso il suo pin Echo (che nel frattempo era stato portato automaticamente alto).

Misurando il tempo che intercorre tra l’emissione del segnale sonoro ed il suo ritorno si può calcolare la distanza dell’ostacolo sul quale è rimbalzato.

La velocità del suono varia a seconda del mezzo (ad esempio, il suono si propaga più velocemente nell’acqua che non nell’aria), e anche al variare delle proprietà del mezzo, specialmente con la sua temperatura.

Nel caso dell’aria la velocità del suono è di 331,2 metri al secondo (1 192,32 km/h) a 0 °C e di 343,1 m/s (1 235,16 km/h) a 20 °C.

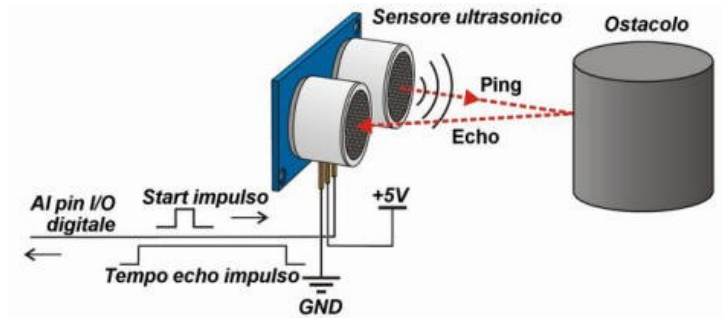
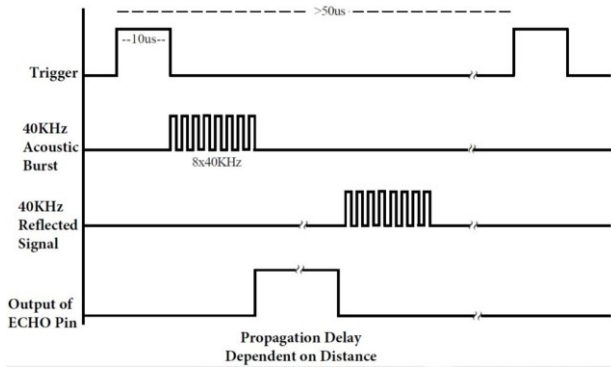
Convertiamo la velocità del suono da 343,1 m/s a 0,03431 cm/microsecondi.

Nel moto uniforme abbiamo:  $S = v \times t \rightarrow S(\text{cm}) = 0,03431 \times t$  (con t in microsecondi)

Poichè la distanza percorsa dal suono è doppia rispetto alla distanza dell’ostacolo (l’impulso sonoro deve andare verso l’ostacolo e tornare indietro) la nuova formula diventa:

$$S = 0.03431 \times t / 2 = 0,017155 \times t = t / 58.3 \text{ (cm)}$$

In sintesi, per utilizzare il sensore, dobbiamo inviare un impulso di 10 ms sul pin Trig e calcolare il tempo che impiega il sensore a rilevare il segnale di ritorno (quindi dopo quanto tempo si porterà basso il pin Echo del sensore).



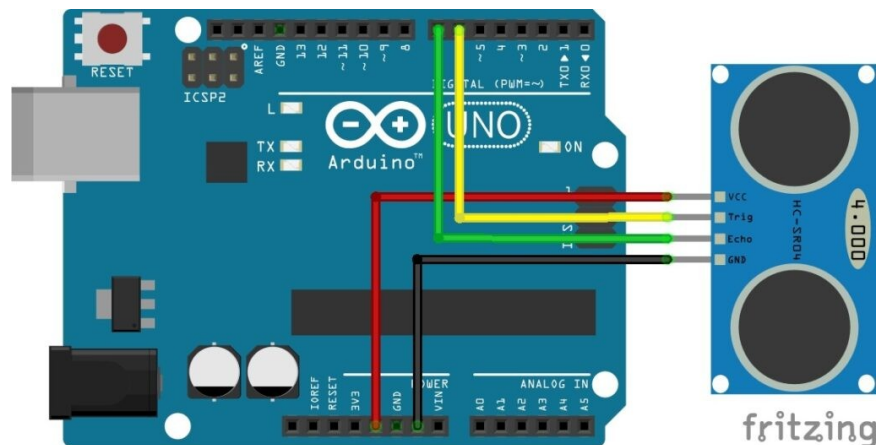
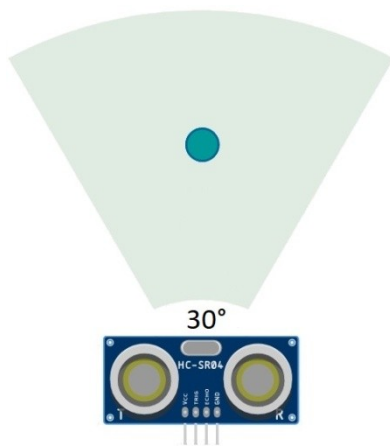
Il comando che si occuperà di contare il tempo che impiega l'impulso ad andare e ritornare è:

```
tempo = pulseIn(echoPin, HIGH)
```

La massima distanza di lavoro del sensore è 400 cm (poco meno di 24 ms); automaticamente il pin Echo dopo 38ms passa basso ed il segnale emesso deve essere considerato perso (bersaglio non presente).

Si devono aspettare almeno 50ms tra un invio di impulso ed un altro, per evitare che echi di vecchi impulsi siano erroneamente letti come validi e dare false letture.

Gli ostacoli da rilevare devono inoltre stare all'interno di un cono di 30° come da foto



Lo sketch da caricare nel nostro Arduino è:

```
#define trigPin 6
#define echoPin 7

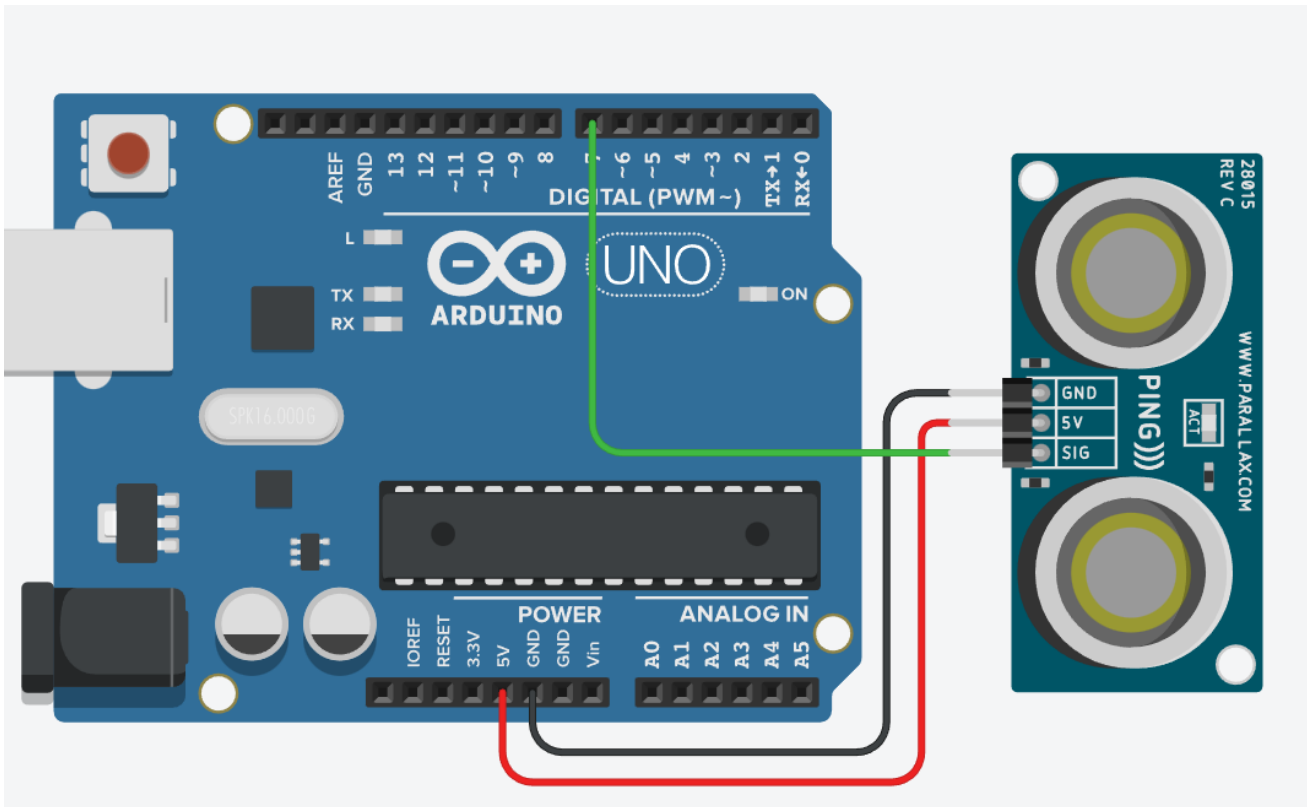
long durata, cm;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  durata = pulseIn(echoPin, HIGH);
  cm = durata / 58; // per i pollici la formula è durata / 148;
  Serial.print("Cm = ");
  Serial.println(cm);
  Serial.println();
}
```

## ESERCIZIO THINKERCAD MISURA DISTANZA CON SENSORE ULTRASUONI

Thinkercad non simula il sensore HC-SR04 ma un modello diverso a 3 pin. Il pin SIG viene utilizzato sia come TRIG che ECHO.



### **CODICE**

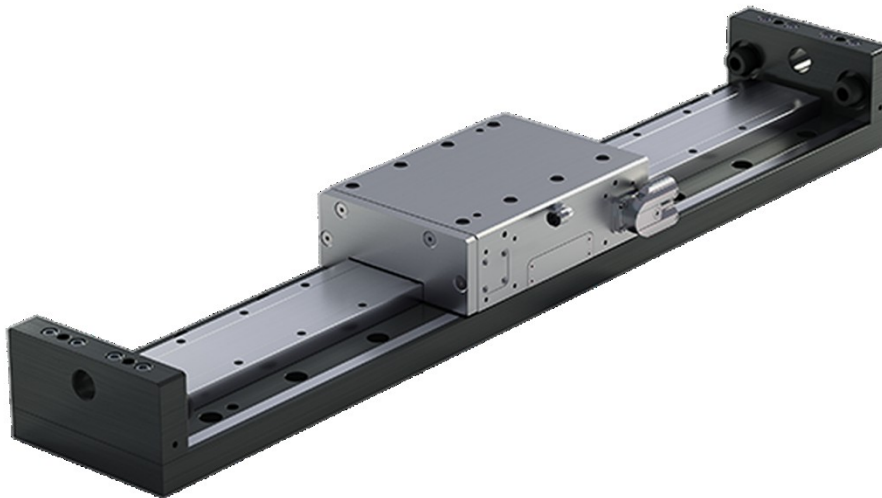
```
int cm = 0;

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

void loop()
{
  // measure the ping time in cm
  cm = readUltrasonicDistance(7, 7);
  Serial.print(cm);
  Serial.println("cm");
  delay(500); // Wait for 100 millisecond(s)
}

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microsecondo
  // measure the ping time in cm → d= velocità suono * tempo
  return (0.01723 * pulseIn(echoPin, HIGH));
}
```

I sistemi di movimentazione e posizionamento nell'automazione industriale sono tecnologie che automatizzano lo spostamento e il posizionamento preciso di materiali e prodotti, riducendo l'intervento umano e aumentando efficienza, velocità e sicurezza tramite componenti meccatronici.



## IL NASTRO TRASPORTATORE

Un nastro trasportatore è un dispositivo meccanico utilizzato per il trasporto continuo di oggetti o materiali da un punto all'altro, tipicamente in contesti di produzione industriale, logistica o movimentazione di grandi quantità di materiale.

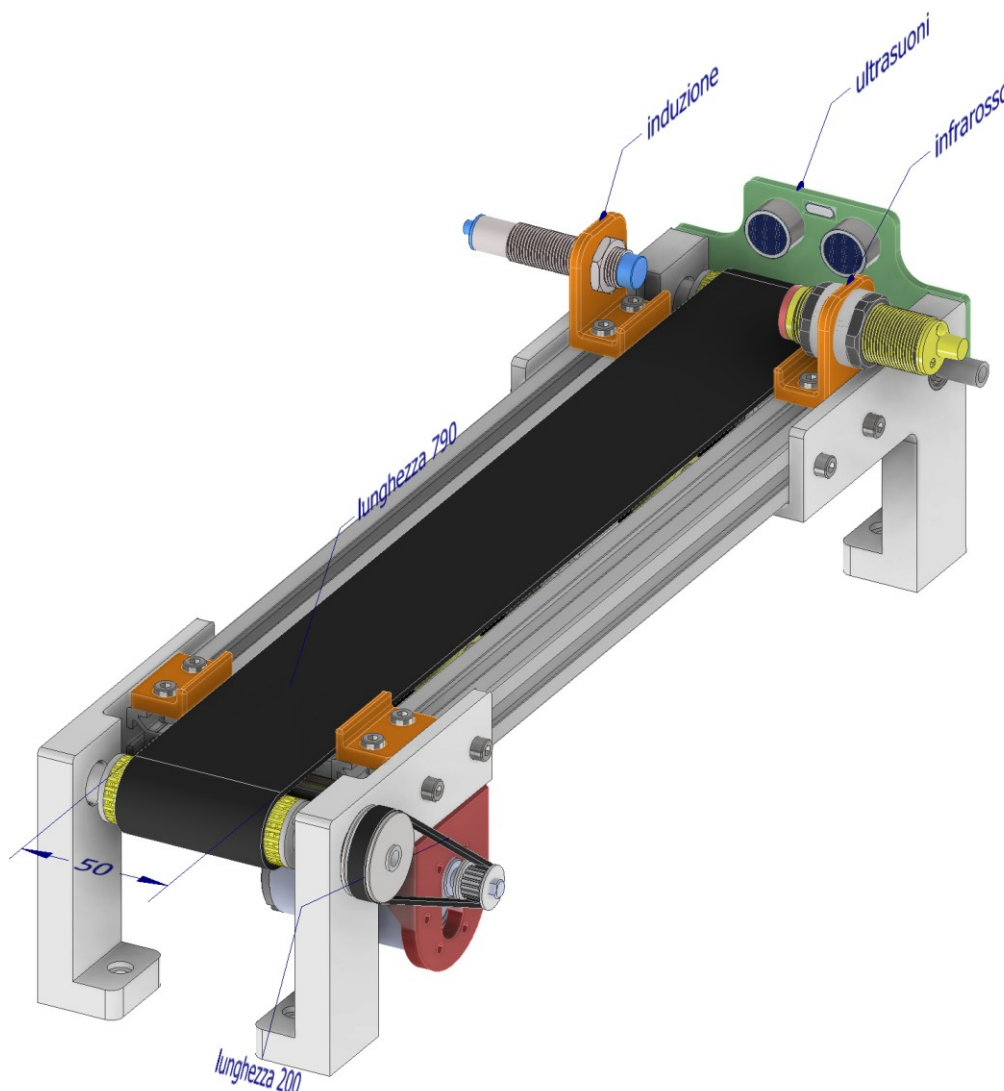
### Componenti e Funzionamento

Il nastro trasportatore è un sistema relativamente semplice ma molto efficiente, composto essenzialmente dai seguenti elementi:

- Nastro (o Tappeto): È la superficie flessibile (spesso in gomma, PVC, poliuretano, o metallo) che trasporta fisicamente il materiale. Forma un anello continuo.
- Tamburo Motore (o Puleggia Motrice): Un cilindro collegato a un motore elettrico che, ruotando, imprime la trazione e il movimento al nastro.
- Tamburo di Rinvio (o Puleggia di Rinvio): Un cilindro folle posto all'estremità opposta che supporta il nastro e aiuta a mantenerlo in tensione.
- Rulli (o Rulliere): Cilindri di supporto posti lungo il percorso per sostenere il nastro (sia sul lato di trasporto che su quello di ritorno) e prevenire l'afflosciamento, specialmente con carichi pesanti.
- Telaio: La struttura metallica che sostiene tutti i componenti.

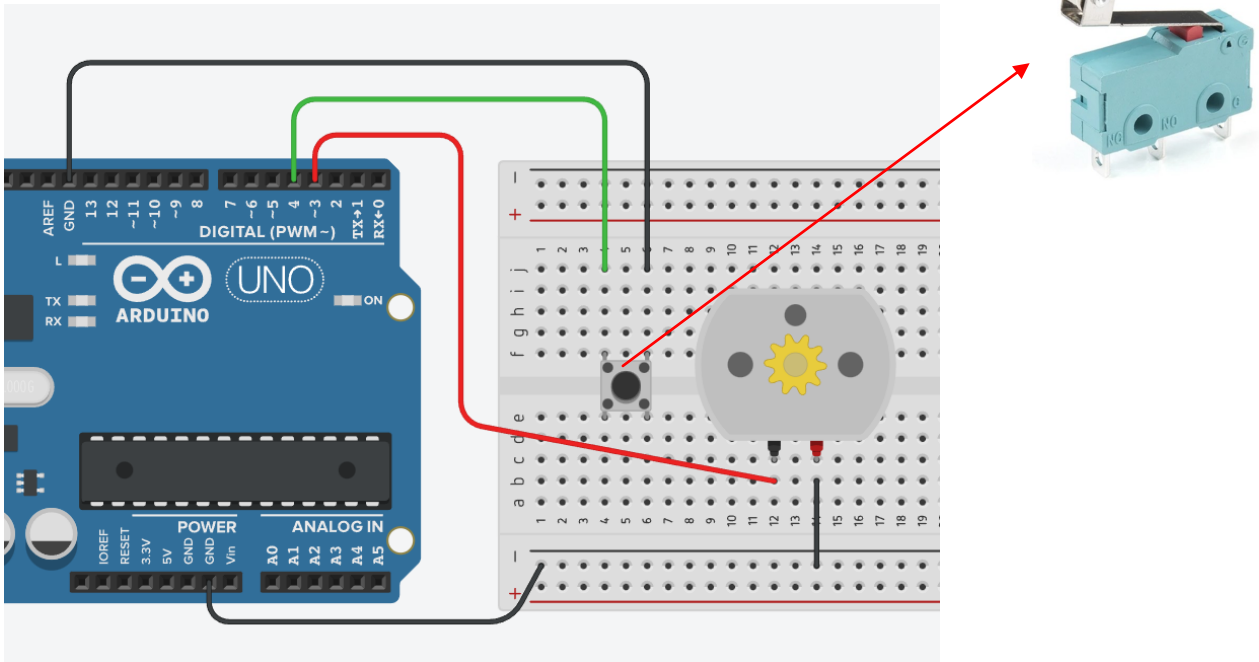
### PROGETTO

- regolare la velocità del nastro e il verso di rotazione
- individuare la presenza di un pezzo
- rilevare la distanza del pezzo



## MONITORARE STATO SENSORI SENZA BLOCCARE IL CODICE

L'obiettivo è spegnere il motore quando viene premuto il finecorsa meccanico e vogliamo stampare lo stato del sensore ogni 2 secondi senza bloccare il programma.



```
int pinFC1 = 4; // finecorsa meccanico
int pinMotore = 3; // alimentazione motore
int statoFC1=1;
```

```
long intervalloStampa = 2000; // 2 secondi
long tempoPrecedenteStampa = 0;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(pinFC1, INPUT_PULLUP);
  pinMode(pinMotore, OUTPUT);
}
```

```
void loop() {
  //stato finecorsa
  statoFC1 = digitalRead(pinFC1);
  // se finecorsa premuto (logica invertita dovuta all'uso del PULL_UP → LOW) spengo motore
  if (statoFC1== LOW) {
    digitalWrite(pinMotore, LOW); // spengo
  }
  else {
    analogWrite(pinMotore, 100); // accendo al minimo
  }
}
```

```
// tempo attuale
long tempoCorrente = millis();
```

```
//Stampa dello stato ogni 2 secondi
if (tempoCorrente - tempoPrecedenteStampa >= intervalloStampa) {
  tempoPrecedenteStampa = tempoCorrente; // aggiorno tempo stampa
  Serial.println(statoFC1 == HIGH ? "NON ATTIVO" : "ATTIVO");
}
}
```

La lettura dello stato del finecorsa viene fatta in tempo reale (< 1ms) mentre la stampa solo ogni 2 secondi.

Generalmente un sensore di qualsiasi tipo non ha un tempo di risposta nullo ma variabile da meno di 1ms fino a centinaia di ms e oltre. Di conseguenza non ha senso leggere lo stato del sensore in tempo reale ma a cadenza fissata.

Il codice seguente legge lo stato del finecorsa solo ogni 100ms.

```
int pinFC1 = 4;    // finecorsa meccanico
int pinMotore = 3; // alimentazione motore
long tempoPrecedenteLettura = 0;
long tempoPrecedenteStampa = 0;
int statoFC1=1;

// Intervalli in millisecondi
int intervalloLettura = 100; // 0.1 secondi
int intervalloStampa = 2000; // 2 secondi

void setup() {
  Serial.begin(9600);
  pinMode(pinFC1, INPUT_PULLUP);
  pinMode(pinMotore, OUTPUT);
}

void loop() {
  long tempoCorrente = millis();

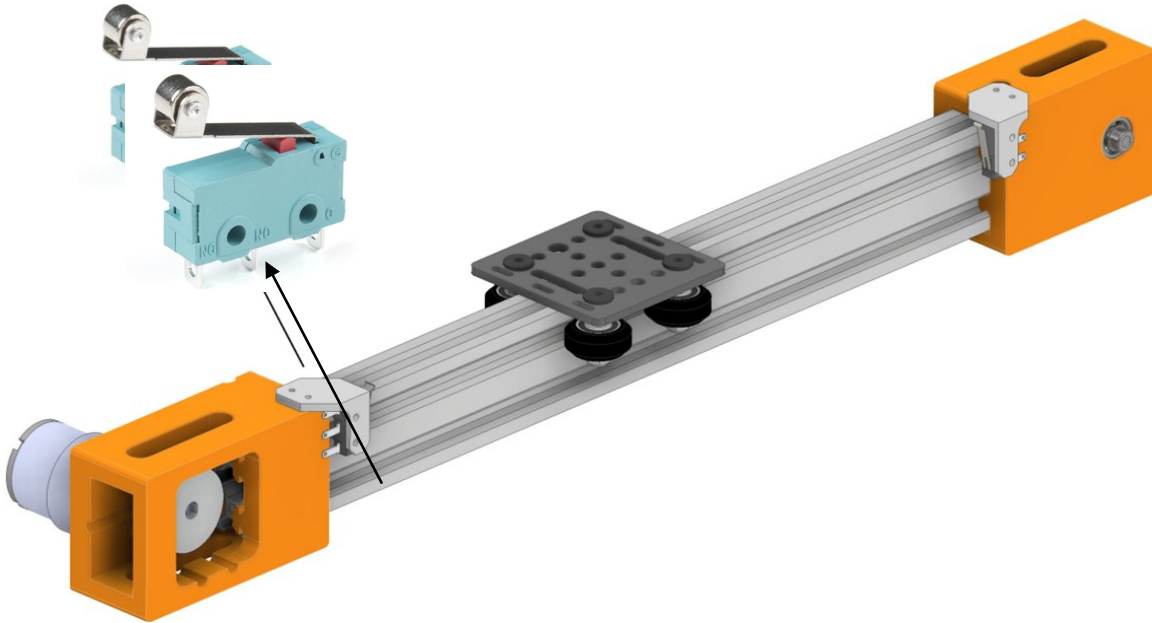
  // Lettura sensori ogni 0.1 secondi
  if (tempoCorrente - tempoPrecedenteLettura >= intervalloLettura) {
    tempoPrecedenteLettura = tempoCorrente;
    //stato finecorsa
    statoFC1 = digitalRead(pinFC1);
    if (statoFC1== LOW) {
      digitalWrite(pinMotore, LOW); // spengo
    }
    else {
      analogWrite(pinMotore, 100); // accendo al minimo
    }
  }

  //Stampa dello stato ogni 2 secondi
  if (tempoCorrente - tempoPrecedenteStampa >= intervalloStampa) {
    tempoPrecedenteStampa = tempoCorrente; // aggiorno tempo stampa
    Serial.println(statoFC1 == HIGH ? "NON ATTIVO" : "ATTIVO");
  }
}
```

Una guida lineare è un dispositivo che permette di movimentare un piano di appoggio lungo un percorso rettilineo in entrambe le direzioni. La traslazione può essere realizzata in vari modi, ad esempio attraverso una cinghia dentata (alta velocità e bassa precisione) o tramite una barra filettata (maggiore precisione e minore velocità).

La figura mostra con trasmissione a cinghia dentata GT2 e pulegge a 40 denti alle estremità del profilato T SLOT 2040.

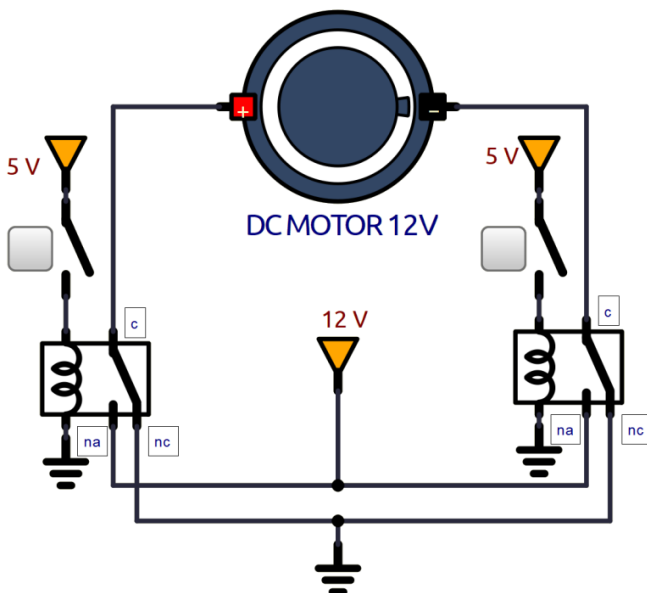
Viene impiegato un motoriduttore CC JGB37-52 e dei finecorsa meccanici su entrambi i lati. I finecorsa hanno lo scopo di bloccare il motore quando la piastra mobile raggiunge le estremità del profilato.



### CONTROLLO VERSO ROTAZIONE MOTORE

Attraverso due relè a doppio contatto è possibile comandare il verso di rotazione di un motore in tensione continua.

Col circuito sottostante se gli interruttori sono entrambi aperti o entrambi chiusi il motore è *fermo*.



#### RELE A DOPPIO CONTATTO

- c → contatto comune
- nc → contatto normalmente chiuso
- na → contatto normalmente aperto

Analizziamo cosa succede se chiudiamo l'interruttore di sinistra.

La bobina del relè di sinistra viene attivata e viene chiuso il contatto na del relè. La corrente fluisce attraverso il contatto appena chiuso e entra nel polo + motore del motore. Esce dal motore e va a massa tramite il contatto nc del relè di destra.

Il motore ruota in senso ORARIO. Se chiudiamo solo l'interruttore di destra la corrente entrerà nel motore dal polo -. Il motore ruota in senso ANTIORARIO.



## CURVA CARATTERISTICA DEL SENSORE TMP36

La curva caratteristica del sensore è la funzione matematica che permette di calcolare la grandezza fisica ( $T^{\circ}\text{C}$ ) in funzione della grandezza elettrica misurata (Volt).

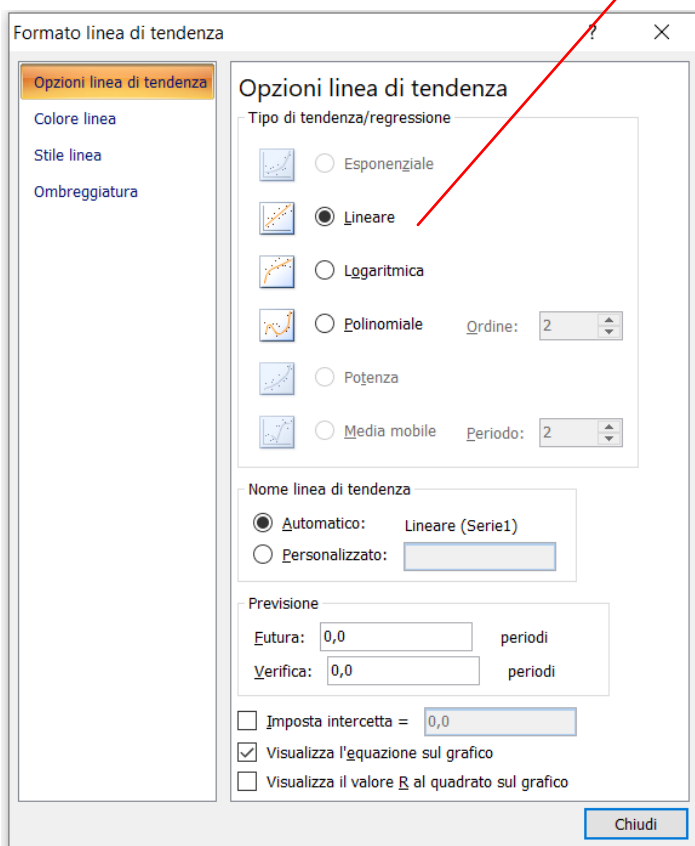
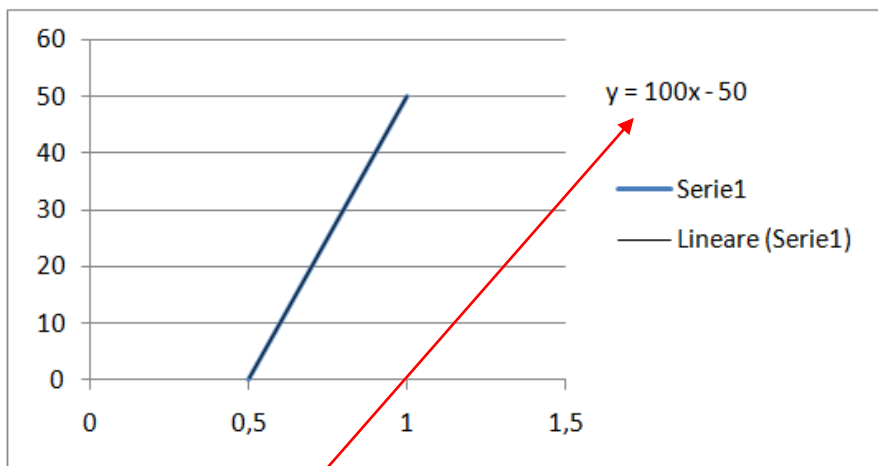
Se la funzione NON è lineare torna utile un "foglio di calcolo" per determinare la curva di tendenza che approssima al meglio la curva caratteristica del sensore.

Nel caso del TMP36 la curva è una semplice retta:  $T(^{\circ}\text{C}) = 100 \text{ Volt} - 50$ .

Se la tensione fornita dal sensore viene rilevata tramite Arduino su un PIN analogico (10 bit) si dovrà convertire il risultato della lettura nel seguente modo:

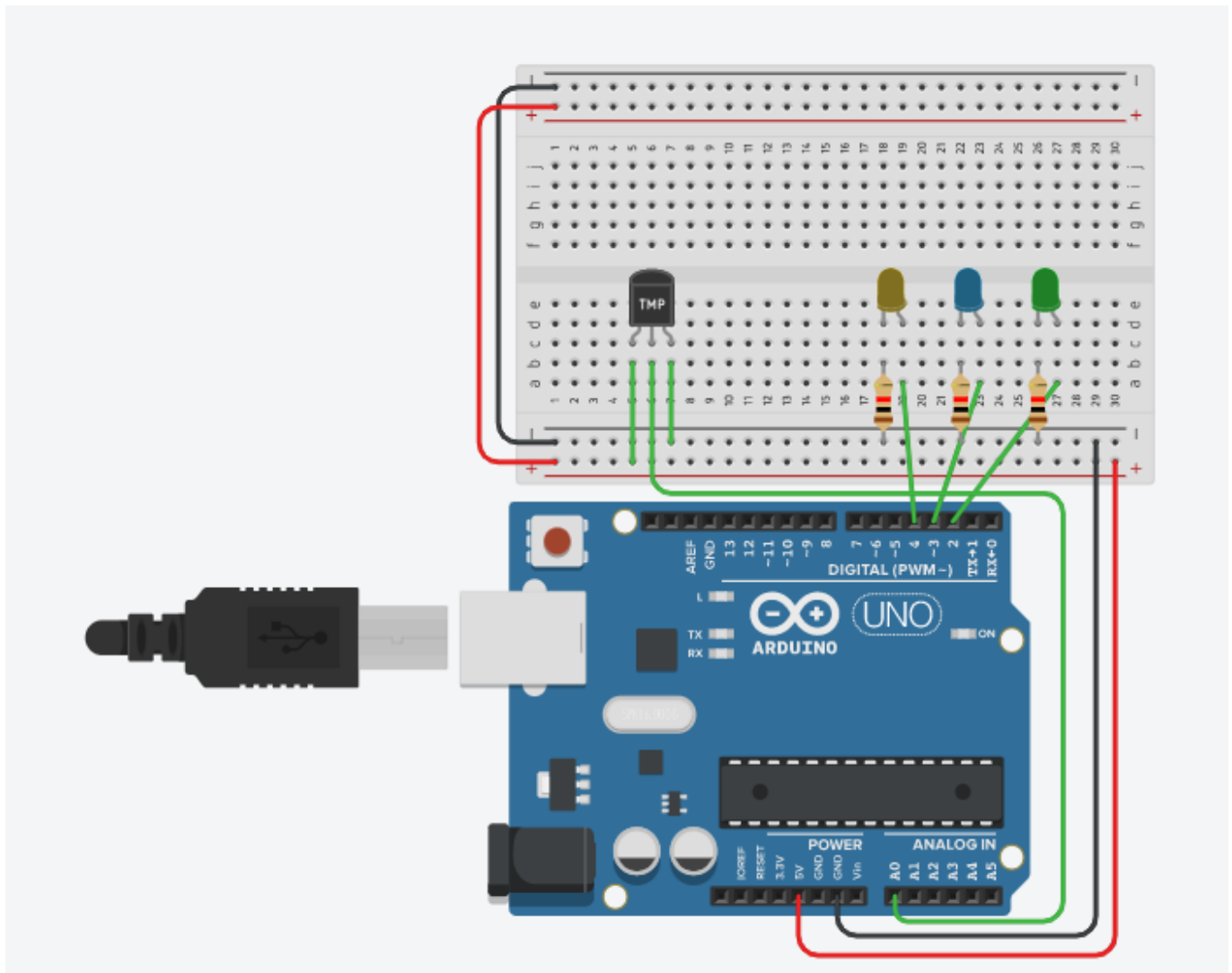
```
float volt = analogRead(PIN) * 5.0/1024.0; // usare i decimali per le divisioni!  
float temperatura = 100 * volt - 50;
```

Volt	T °C
0,5	0
1	50



## ESERCIZIO CON SENSORE TMP36

Accendere una striscia di 3 led in modo proporzionale alla temperatura rilevata dal sensore.



## CODICE

```
float volt;
float temperatura= 0;

void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);

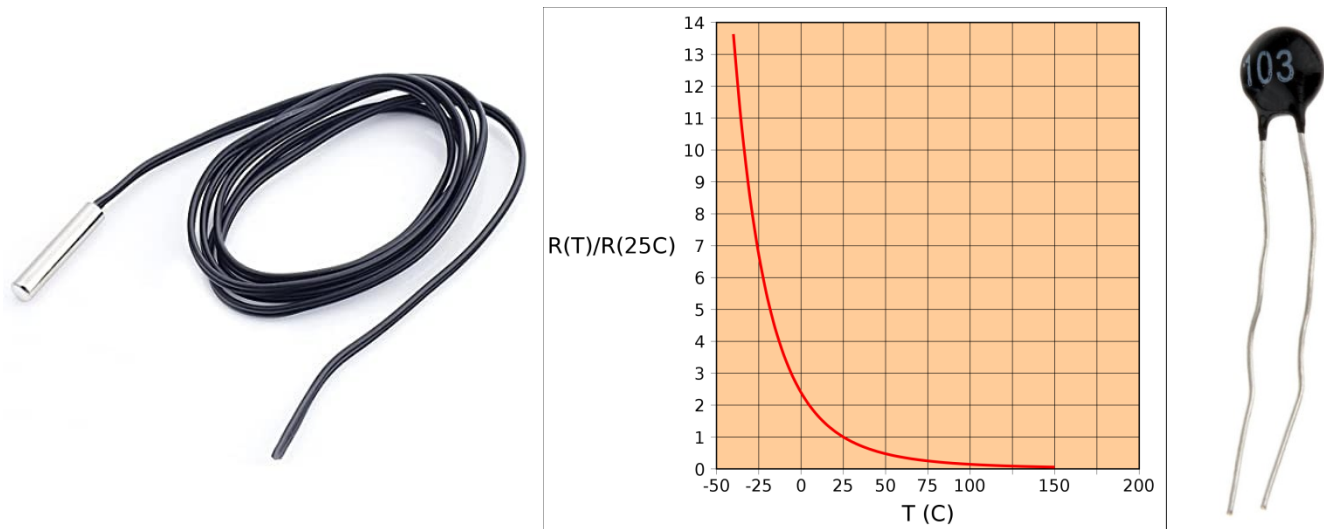
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  volt = analogRead(A0) * 5.0/1024.0; // usare i decimali nella divisione!
  temperatura = 100 * volt - 50;
  Serial.print(temperatura);
  Serial.println(" C");

  if (temperatura < 0) {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    Serial.println("SOTTO ZERO");
  }
  if (temperatura >= 0 && temperatura < 10) {
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    Serial.println("BASSA");
  }
  if (temperatura >= 10 && temperatura < 20) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    Serial.println("MEDIA");
  }
  if (temperatura >= 20) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
    Serial.println("ALTA");
  }
  delay(1000);
}
```

## TERMISTORE NTC (NEGATIVE TEMPERATURE COEFFICIENT)

Un termistore è un resistore il cui valore di resistenza varia con la temperatura.

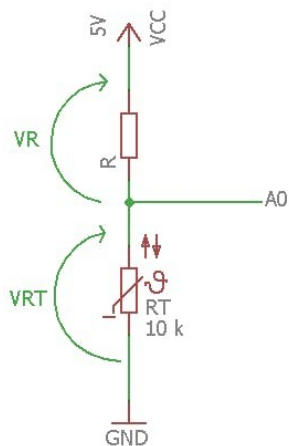


I termistori NTC possono essere caratterizzati con un'equazione detta equazione con parametro B o beta value:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln\left(\frac{R}{R_0}\right)$$

dove le temperature sono in kelvin (K) e  $R_0$  è la resistenza alla temperatura  $T_0$  (di solito  $25\text{ °C}=298,15\text{ K}$ ).

B è costante solo in prima approssimazione e di solito ne viene indicato l'intervallo di temperature in cui è valida e la sua tolleranza in % (ad esempio  $B_{25/85} \pm 2\%$  indica che B tra  $25\text{ °C}$  e  $85\text{ °C}$  ha un errore massimo di  $\pm 2\%$ ).

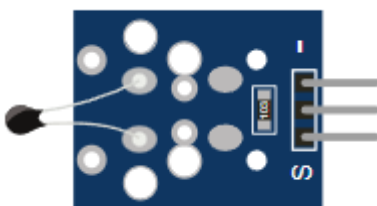


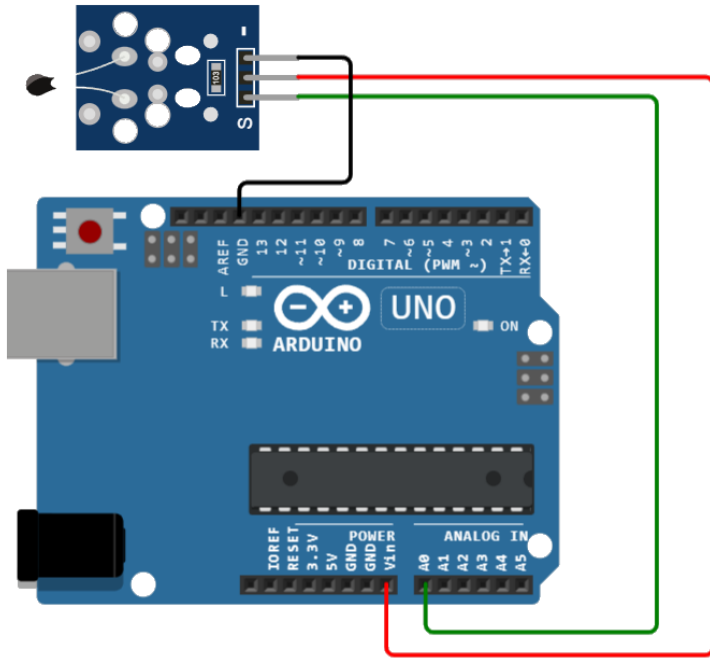
$$RT = R_0 e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

$$RT = V_{RT} / (V_R/R)$$

$$T = \frac{1}{\frac{\ln\left(\frac{RT}{R_0}\right)}{B} + \frac{1}{T_0}}$$

Il modulo sensore di temperatura per Arduino include un termistore NTC da 10K in serie con un resistore da 10K.





simulabile su "wokwi.com"

## CODICE

```
//Thermistor parameters: RT0: 10KΩ B: 3977 K +- 0.75% T0: 25 C +- 5%
//From datasheet
#define RT0 10000 // Ω
#define B 3977 // K
//-----

#define VCC 5 //Supply voltage
#define R 10000 //R=10KΩ

//Variables
float RT, VR, In, TX, T0, VRT;

void setup() {
  Serial.begin(9600);
  T0 = 25 + 273.15;
}

void loop() {
  VRT = analogRead(A0); // 0-1023 → tensione sul termistore
  VRT = (5.00 / 1023.00) * VRT; // converto in V
  VR = VCC - VRT; // tensione sulla resistenza R da 10K
  RT = VRT / (VR / R); // Resistenza di RT (V/I)
  In = log(RT / RT0);
  TX = 1 / (In / B + 1 / T0); //Temperature from thermistor in K
  TX = TX - 273.15; //Conversion to °C

  Serial.print("Temperatura: ");
  Serial.print(TX);
  Serial.println(" °C");
  delay(1000);
}
```

## SISTEMI DI CONTROLLO DELLA TEMPERATURA CON MICROCONTROLLORE

Lo schema a blocchi sottostante mostra un tipico schema di controllo della temperatura di un sistema.

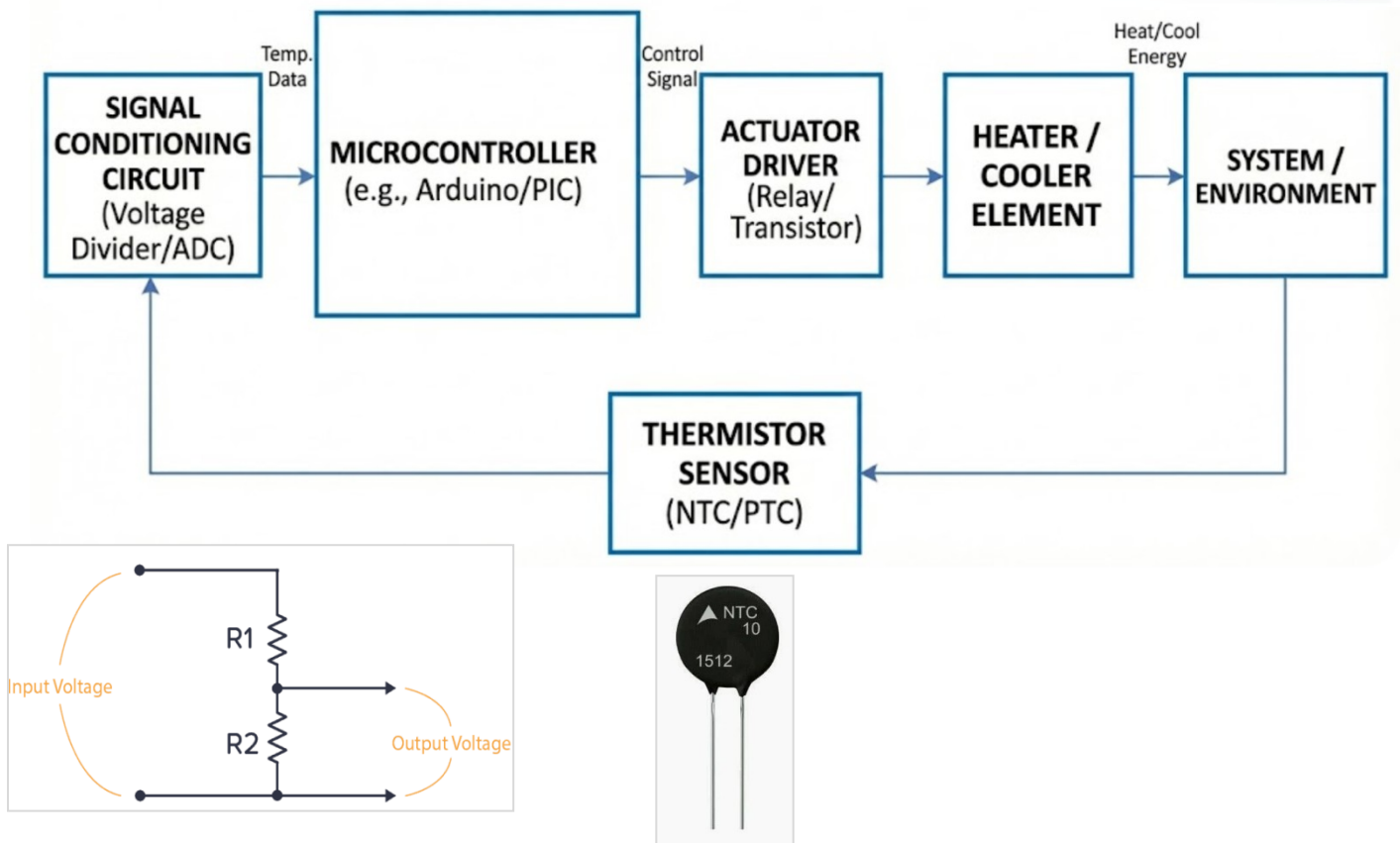
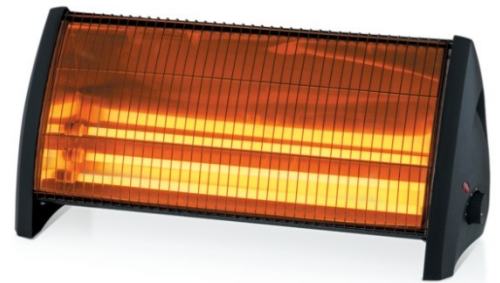
E' evidente la presenza del sensore (retroazione) che fornisce al microcontrollore, in tempo reale, la temperatura del sistema controllato.

Senza la retroazione è possibile solo una regolazione del sistema basata sulle leggi fisiche che lo governano.

Un qualsiasi disturbo esterno, non venendo rilevato, porterebbe il sistema ad una condizione di regime diversa da quella programmata

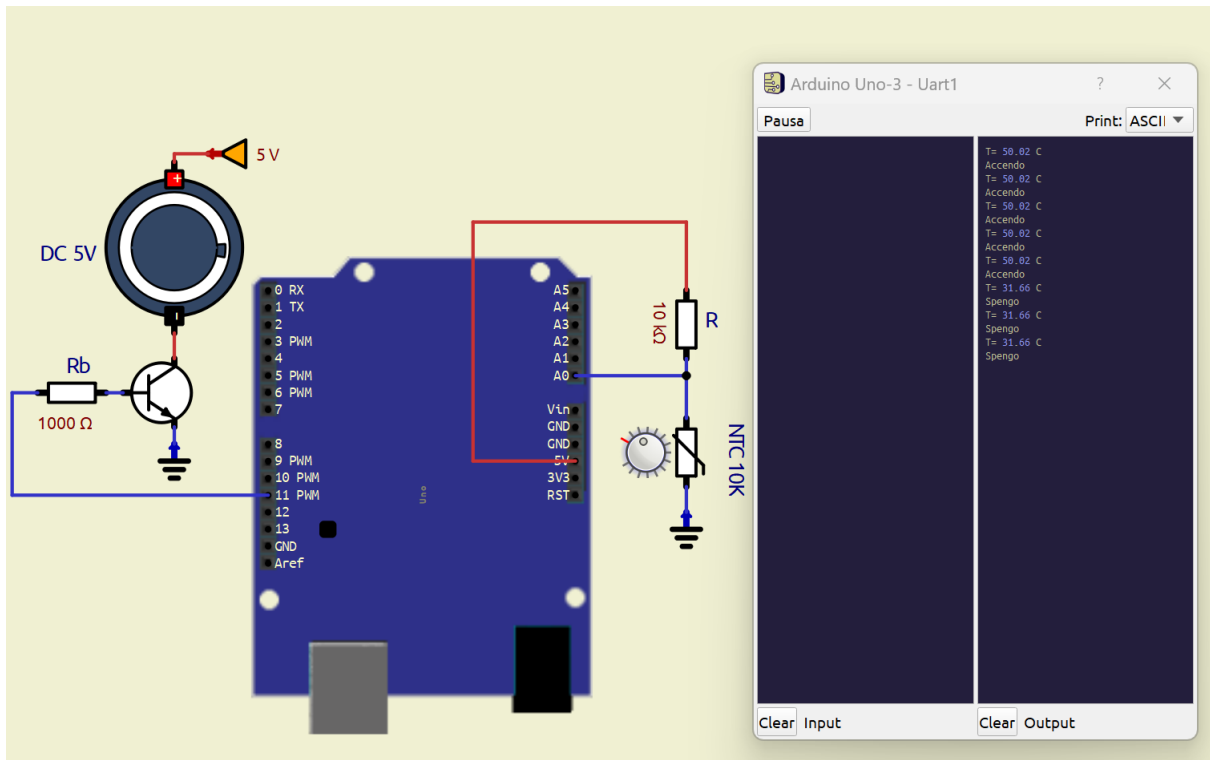
In un sistema retroazionato il segnale elettrico in uscita al sensore viene generalmente condizionato in modo da essere compatibile con gli ingressi analogici del microcontrollore (Arduino ad esempio è in grado di leggere tensioni da 0 a 5V).

Il microcontrollore implementa la logica di controllo (dalla più semplice ON-OFF alla più complessa PID). Genera quindi un segnale di controllo per il driver (transistor, relè ecc.) che comanda l'attuatore cioè l'elemento riscaldante (es. resistenza elettrica → calore per effetto Joule) che genera la potenza termica necessaria a controllare la temperatura del sistema.



## SISTEMA CONTROLLO TEMPERATURA ON-OFF CON TRANSISTOR

Vogliamo attivare una ventola di raffreddamento quando la temperatura del un competente elettronico supera quella critica utilizzando un termistore NTC 10K.



```
int pinVentola=11;
const float B = 3950;
const float R = 10000.0;
const float RT0 = 10000.0;
const float T0 = 298.15;
const float Vcc = 5.0;
//Variables
float Rs, VR, I, Ts, Vs;

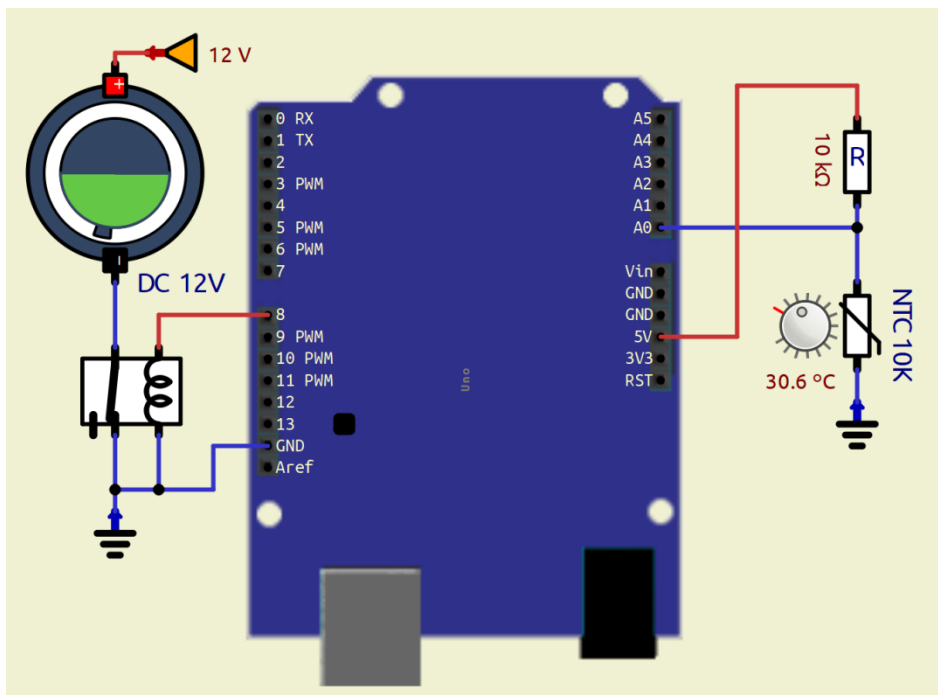
void setup()
{
  pinMode(pinVentola, OUTPUT);
  Serial.begin ( 9600);
}

void loop()
{
  Vs = analogRead(A0); // 0-1023 tensione sul termistore
  Vs = (5.00 / 1023.00) * Vs; // converto in Volt
  VR = Vcc - Vs; // tensione sulla resistenza R da 10K
  I = VR/R;
  Rs = Vs/I; // Resistenza di Rs (V/I)
  Ts =1/ (log(Rs/ RT0) / B + 1 / T0); //Ts in K
  Ts = Ts - 273.15; //Conversion to °C
  Serial.print("T= ");
  Serial.print(Ts);
  Serial.println(" C");

  if (Ts>=30) {
    Serial.println("Accendo");
    digitalWrite(pinVentola,HIGH);
  }
  else {
    Serial.println("Spengo");
    digitalWrite(pinVentola,LOW);
  }
  delay( 5000 );
}
```

## SISTEMA CONTROLLO TEMPERATURA ON-OFF CON RELÈ

L'utilizzo del relè ci permette di comandare motori di qualsiasi potenza, anche in alternata, in sicurezza.



```
int pinVentola=8;
const float B = 3950;
const float R = 10000.0;
const float RT0 = 10000.0;
const float T0 = 298.15;
const float Vcc = 5.0;
```

```
//Variables
float Rs, VR, I, Ts, Vs;
```

```
void setup()
{
  pinMode(pinVentola, OUTPUT);
  Serial.begin ( 9600);
}
```

```
void loop()
{
  int btnAvvio= digitalRead(12);
```

```
Vs = analogRead(A0); // 0-1023 tensione sul termistore
Vs = (5.00 / 1023.00) * Vs; // converto in Volt
VR = Vcc - Vs; // tensione sulla resistenza R da 10K
I = VR/R; // corrente che circola nel sensore Rs
Rs = Vs/I; // resistenza di Rs (V/I)
Ts =1/ (log(Rs/ RT0) / B + 1 / T0); //Ts in K
Ts = Ts - 273.15; //Conversion to °C
Serial.print("T= ");
Serial.print(Ts);
Serial.println(" C");
```

```
if (Ts>=30) {
  Serial.println("Ventola ON");
  digitalWrite(pinVentola,HIGH);
}
else {
  Serial.println("Ventola OFF");
  digitalWrite(pinVentola,LOW);
}
```

```
delay( 2000 );
}
```

## CONTROLLO ON-OFF CON ISTERESI

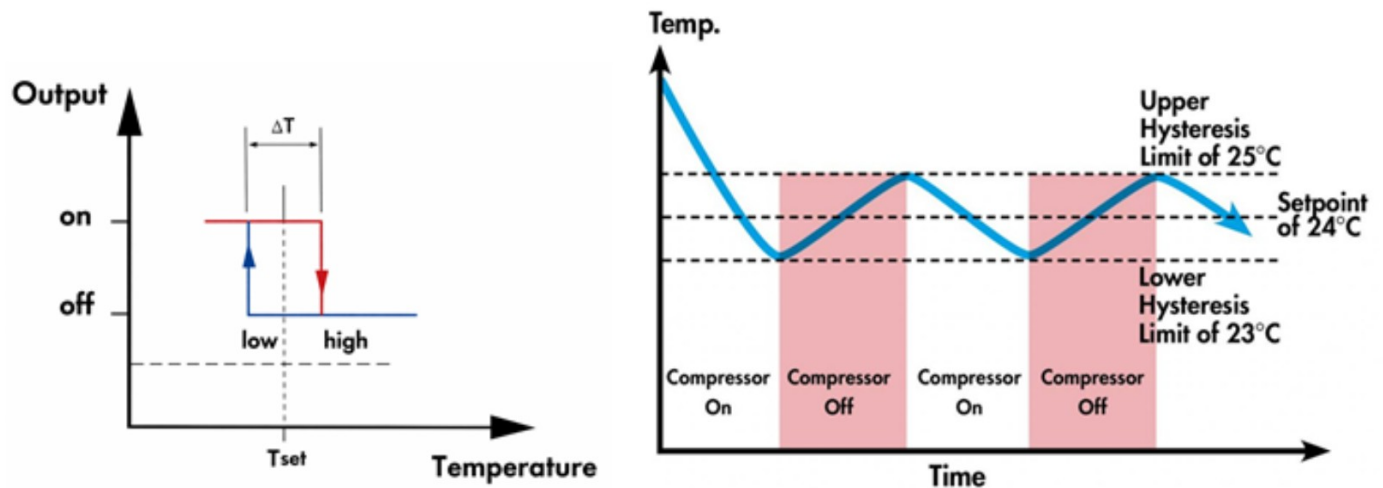
Il controllo On/Off con isteresi evita che l'attuatore venga acceso e spento ad elevata frequenza.

In molti casi ciò potrebbe portare usura e/o guasti soprattutto in presenza di attuatori meccanici o elettromeccanici.

Ad esempio, se vogliamo attivare il riscaldamento senza isteresi con un relè che accende e spegne la macchina che riscalda, con il setpoint a 25°C il sistema si accenderebbe a 24.99°C e si spegnerebbe a 25.01°C. In poco tempo il relè si guasterebbe!

Tramite l'isteresi introduciamo una zona di tolleranza della variabile da controllare in modo da limitare la frequenza di accensione e spegnimento dell'attuatore.

L'isteresi si può rappresentare in forma digitale (ON-OFF) a sinistra e in funzione del tempo a destra.



### Come funziona la logica con isteresi

Dobbiamo riscaldare il sistema quando la temperatura rilevata è inferiore a quella di setpoint.

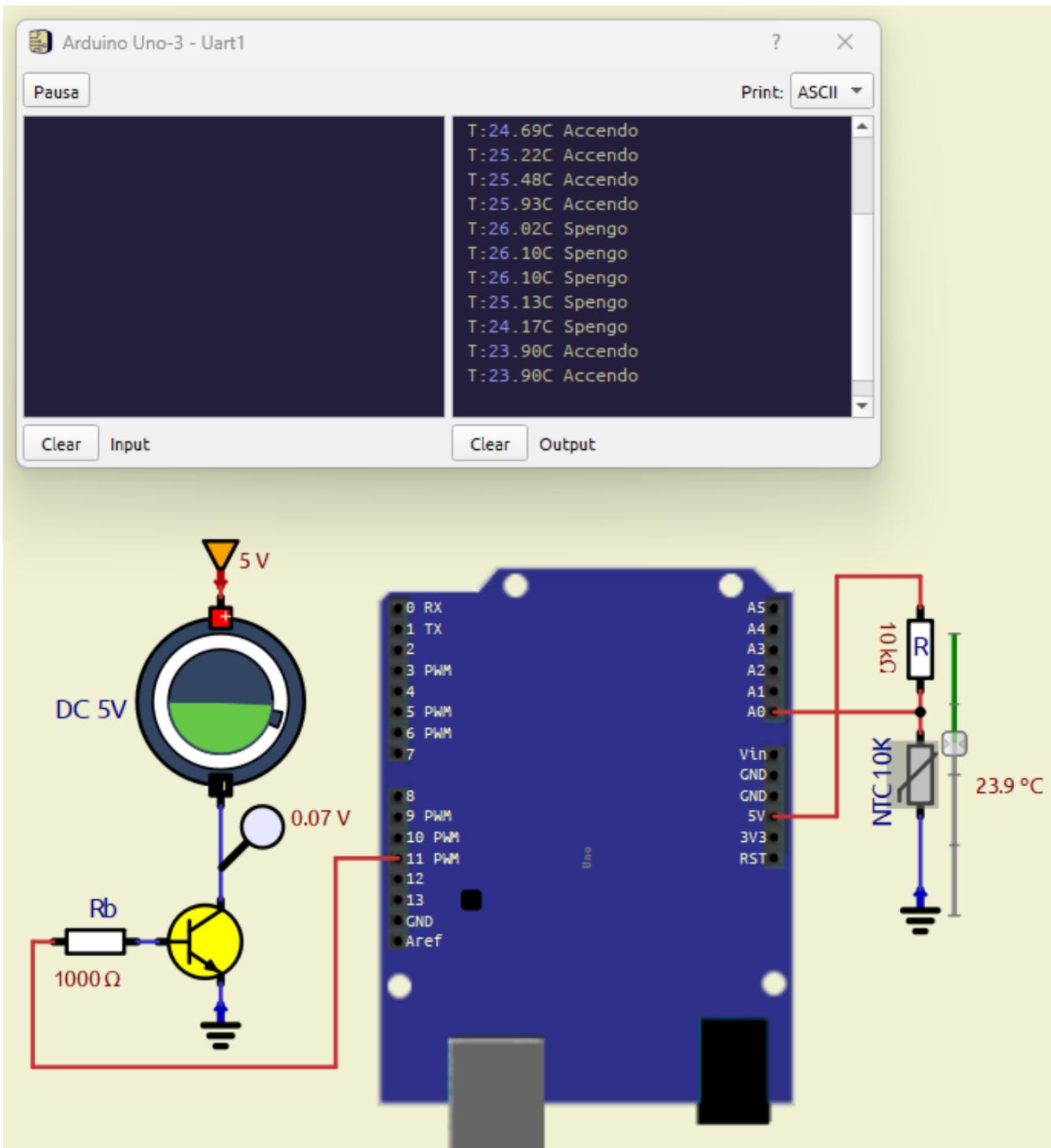
Fissiamo una zona di tolleranza (isteresi) che individua una temperatura massima e minima fra cui la grandezza può variare:

- Setpoint: 25°C
- Isteresi (H): 1°C
- Soglia di accensione: Setpoint - H = 24°C
- Soglia di spegnimento: Setpoint + H = 26°C

Tradotto in istruzioni di programmazione diventa:

```
if (Ts < (setPoint - isteresi)) {  
    attivaRiscaldamento = true;  
}  
else if (Ts > (setPoint + isteresi)) {  
    attivaRiscaldamento = false;  
}
```

**NOTA:** Quando la temperatura è compresa nella zona di isteresi (24-26) l'attuatore resta nell'ultimo stato impostato.



## Codice Arduino

```
const float B = 3950;
const float R = 10000.0;
const float RT0 = 10000.0;
const float T0 = 298.15;
const float Vcc = 5.0;
const float setPoint = 25.0;
const float isteresi = 1.0;
//Variabili
int pinVentola=11;
float Rs, VR, I, Ts, Vs;
bool attivaRiscaldamento= false;

void setup()
{
    pinMode(pinVentola, OUTPUT);
    Serial.begin ( 9600);
}

void loop()
{
    Vs = analogRead(A0); // 0-1023 tensione sul termistore
    Vs = (5.00 / 1023.00) * Vs; // converto in Volt
    VR = Vcc - Vs; // tensione sulla resistenza R da 10K
    I = VR/R;
    Rs = Vs/I; // Resistenza di Rs (V/I)
    Ts = 1/ (log(Rs/ RT0) / B + 1 / T0); //Ts in K
    Ts = Ts - 273.15; //Conversion to °C
    Serial.print("T:"); Serial.print(Ts); Serial.print("C ");

    // LOGICA CON ISTERESI
    if (Ts < (setPoint - isteresi)) {
        attivaRiscaldamento = true;
    }
    else if (Ts > (setPoint + isteresi)) {
        attivaRiscaldamento = false;
    }

    // ATTUAZIONE LOGICA
    if (attivaRiscaldamento) {
        digitalWrite(pinVentola, HIGH); // ON
        Serial.println("Accendo");
    } else {
        Serial.println("Spengo");
        digitalWrite(pinVentola, LOW); // OFF
    }
    delay( 1000 );
}
```

## **CODICE NON BLOCCANTE CON MILLIS**

```
const float B = 3950;
const float R = 10000.0;
const float RT0 = 10000.0;
const float T0 = 298.15;
const float Vcc = 5.0;
const float setPoint = 25.0;
const float isteresi = 1.0;
//Variabili
int pinVentola=11;
float Rs, VR, I, Ts, Vs;
bool attivaRiscaldamento= false;
long t_attivazione=0;

void setup()
{
    pinMode(pinVentola, OUTPUT);
    Serial.begin ( 9600);
}

void loop()
{
    Vs = analogRead(A0); // 0-1023 tensione sul termistore
    Vs = (5.00 / 1023.00) * Vs; // converto in Volt
    VR = Vcc - Vs; // tensione sulla resistenza R da 10K
    I = VR/R;
    Rs = Vs/I; // Resistenza di Rs (V/I)
    Ts =1/ (log(Rs/ RT0) / B + 1 / T0); //Ts in K
    Ts = Ts - 273.15; //Conversion to °C

    // LOGICA CON ISTERESI
    if (Ts < (setPoint - isteresi)) {
        attivaRiscaldamento = true;
    }
    else if (Ts > (setPoint + isteresi)) {
        attivaRiscaldamento = false;
    }

    // ATTUAZIONE LOGICA
    If (millis() - t_attivazione >1000) {
        t_attivazione = millis();
        Serial.print("T:"); Serial.print(Ts); Serial.print("C ");
        if (attivaRiscaldamento) {
            digitalWrite(pinVentola, HIGH); // ON
            Serial.println("Accendo");
        } else {
            Serial.println("Spengo");
            digitalWrite(pinVentola, LOW); // OFF
        }
    }
}
```

## CONTROLLO DI TEMPERATURA PROPORZIONALE

Prendiamo in considerazione un piccolo boiler rapido da 8L.

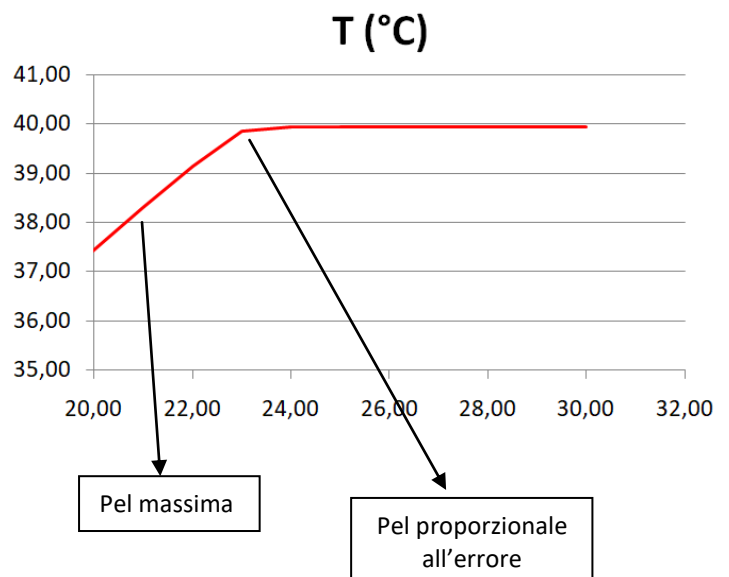
Il sistema in modalità ECO scalda l'acqua tramite un resistenza elettrica con una potenza di 500W.

Vogliamo studiare il sistema per capire quanto tempo ci vuole per portare l'acqua alla  $T=40^{\circ}\text{C}$  e determinare che potenza è necessaria per mantenere poi la temperatura di set point senza spegnere il generatore di potenza elettrica.

Si tratta quindi un sistema di controllo PROPORZIONALE in cui la potenza fornita all'acqua è proporzionale alla differenza di temperatura tra SET POINT e quella raggiunta dall'acqua:

$$P_{el} = k_p (T_{setpoint} - T(t)) \text{ watt}$$

**Ovviamente per implementare un controllo di tipo PROPORZIONALE serve un attuatore di tipo proporzionale (cioè la cui azione possa essere regolata e non solo attivata o spenta) come ad esempio una resistenza elettrica in cui si varia la corrente elettrica e quindi il calore generato (effetto Joule  $\rightarrow P_{el}=R I^2$ ).**



### Dati del problema:

Boiler  $\rightarrow 20 \times 20 \times 24 \text{ cm}$  con isolamento da 5cm in polistirolo  $k=0,05 \text{ w/mk}$

$T_{setpoint}= 40^{\circ}\text{C}$

Tambiente=  $20^{\circ}\text{C}$

Coefficienti convettivi:  $h_{aria}=20$ ;  $h_{acqua}=400$

$P_{elettrica}= 500\text{W}$

Calore disperso:  $Q_d= (T-T_a)/R_t$  con  $R_t= (1/A)(1/h_{aria}+1/h_{acqua}+s/k)$

Bilancio di energia del sistema:  $(P_{el} - Q_d) dt = m C_t dT$

Si tratta di una equazione differenziale con soluzione analitica:  $T(t) = T_a + R_t * P_{el} * (1 - e^{-(t/m C_t R_t)})$

Se consideriamo degli intervalli di tempo discreti  $\Delta t$  abbiamo:  $T' = \Delta T / \Delta t = (P_{el} - (T - T_a) / R_t) / m C_t$

Quindi la variazione di temperatura in un intervallo di tempo  $\Delta t$  vale:  $\Delta T = T' * \Delta t$

NOTA: l'analisi fisica del sistema ci permette di calcolare indicativamente quanto tempo ci vuole a raggiungere il SET-POINT con la potenza massima disponibile; arrivati vicino al SET-POINT possiamo attivare il controllo di tipo proporzionale modulando la potenza elettrica:  $P_{el}(T) = P_{el\_max} * (T_{sp}-T)$  dove  $(T_{sp}-T)$  rappresenta l'errore  $<1$

### Mini boiler e controllo PROPORZIONALE della T

ct H2O 4186 J/kg k  
 Pel 500 w  
 Tamb 293 k 20 °C  
 Tiniz 288 k 15 °C  
 Tsp 313 k 40 °C  
 mCt 33488  
 lato 0,2 m  
 altezza 0,2 m  
 A 0,24 m2  
 V 0,008 m3  
 m 8 Kg  
 s\_iso 0,005 m

#### BILANCIO DI ENERGIA

$$(Pe - Qd) dt = m Ct dT$$

$$Qd = (T-Ta)/Rt$$

$$(Pe - (T-Ta)/Rt) dt = m Ct dT$$

Soluzione analitica

$$T(t) = Ta + Rt * Pel * (1 - e^{-(t/mCtRt)})$$

Soluzione numerica

$$T' = \Delta T / \Delta t = (Pe - (T-Ta)/Rt) / m Ct$$

$$\Delta T = T' \Delta t$$

k iso 0,05 w/m k  
 Rt 0,63542 k/w

Resistenza

termica

$$Rt = (1/A) * (s/k + 1/hi + 1/he)$$

NB: finchè non ci avviciniamo alla Tsp forniamo la massima potenza a disposizione; poi iniziamo a calarla in modo proporzionale!

Δt 60 s

$$Pel = SE((F22 - F22) > 1; F22; F22 * (ASS(F22 - F22)))$$

ANALITICO						NUMERICO						
t (s)	min	ore	Pel	T (k)	T (°C)	Tnum	Tn (°C)	e	Pel	T'	ΔT	
0	0,00	0	500	293,00	20,00	293,00	20,00	20,00	500,0	0,0149	0,90	
60	1,00	0,02	500	293,89	20,89	293,90	20,90	19,10	500,0	0,0149	0,89	
120	2,00	0,03	500	294,79	21,79	294,79	21,79	18,21	500,0	0,0148	0,89	
180	3,00	0,05	500	295,68	22,68	295,68	22,68	17,32	500,0	0,0148	0,89	
240	4,00	0,07	500	296,56	23,56	296,57	23,57	16,43	500,0	0,0148	0,89	
300	5,00	0,08	500	297,45	24,45	297,45	24,45	15,55	500,0	0,0147	0,88	
360	6,00	0,1	500	298,33	25,33	298,34	25,34	14,66	500,0	0,0147	0,88	
420	7,00	0,12	500	299,21	26,21	299,22	26,22	13,78	500,0	0,0146	0,88	
480	8,00	0,13	500	300,09	27,09	300,10	27,10	12,90	500,0	0,0146	0,88	
540	9,00	0,15	500	300,96	27,96	300,97	27,97	12,03	500,0	0,0146	0,87	
600	10,00	0,17	500	301,83	28,83	301,85	28,85	11,15	500,0	0,0145	0,87	
660	11,00	0,18	500	302,70	29,70	302,72	29,72	10,28	500,0	0,0145	0,87	
720	12,00	0,2	500	303,57	30,57	303,58	30,58	9,42	500,0	0,0144	0,87	
780	13,00	0,22	500	304,44	31,44	304,45	31,45	8,55	500,0	0,0144	0,86	
840	14,00	0,23	500	305,30	32,30	305,31	32,31	7,69	500,0	0,0144	0,86	
900	15,00	0,25	500	306,16	33,16	306,18	33,18	6,82	500,0	0,0143	0,86	
960	16,00	0,27	500	307,01	34,01	307,03	34,03	5,97	500,0	0,0143	0,86	
1020	17,00	0,28	500	307,87	34,87	307,89	34,89	5,11	500,0	0,0142	0,85	
1080	18,00	0,3	500	308,72	35,72	308,74	35,74	4,26	500,0	0,0142	0,85	
1140	19,00	0,32	500	309,57	36,57	309,60	36,60	3,40	500,0	0,0142	0,85	
1200	20,00	0,33	500	310,42	37,42	310,44	37,44	2,56	500,0	0,0141	0,85	
1260	21,00	0,35	500	311,27	38,27	311,29	38,29	1,71	500,0	0,0141	0,84	
1320	22,00	0,37	500	312,11	39,11	312,14	39,14	0,86	432,1	0,0120	0,72	
1380	23,00	0,38	500	312,95	39,95	312,86	39,86	0,14	72,0	0,0012	0,07	
1440	24,00	0,4	500	313,79	40,79	312,93	39,93	0,07	35,5	0,0001	0,01	
1500	25,00	0,42	500	314,62	41,62	312,94	39,94	0,06	31,8	0,0000	0,00	
1560	26,00	0,43	500	315,46	42,46	312,94	39,94	0,06	31,4	0,0000	0,00	
1620	27,00	0,45	500	316,29	43,29	312,94	39,94	0,06	31,4	0,0000	0,00	
1680	28,00	0,47	500	317,12	44,12	312,94	39,94	0,06	31,4	0,0000	0,00	
1740	29,00	0,48	500	317,95	44,95	312,94	39,94	0,06	31,4	0,0000	0,00	
1800	30,00	0,5	500	318,77	45,77	312,94	39,94	0,06	31,4	0,0000	0,00	

**NB: con il controllo PROPORZIONALE resta sempre, a regime, un piccolo errore rispetto al SET-POINT.**

## CONTROLLO ON-OFF VS PROPORZIONALE

In questo esempio consideriamo il controllo di temperatura di un locale con un sistema di riscaldamento che immette una certa potenza termica  $P_{el}$ .

Il bilancio di energia dell'ambiente è rappresentato in maniera approssimativa da:

$$(P_{el} - Q_d) * dt = m C_t (T(t) - T_{esterna}) \quad [J]$$

con  $T(t)$ = temperatura del locale ( $^{\circ}C$ );  $m$ =massa di aria nel locale (kg)

Il calore disperso dal locale verso l'esterno è dato da:

$$Q_d = U A (T(t) - T_{esterno}) \quad [W]$$

con  $U = 1/R_t$   $[W/m^2K]$  l'inverso della resistenza termica delle pareti del locale

Il locale può essere simulato in Arduino con una funzione che riceve in ingresso la potenza di riscaldamento.

La funzione calcola la temperatura del locale ogni volta che viene chiamata (ogni 50ms). Poiché la funzione viene sempre chiamata ininterrottamente dall'avvio del programma permette di simulare l'andamento della temperatura nel tempo (NB: il codice nel loop non deve usare delay!). Per garantire il tutto utilizziamo un timer (interrupt) che non blocca il codice e in automatico chiama una funzione a cadenza prefissata di 50ms.

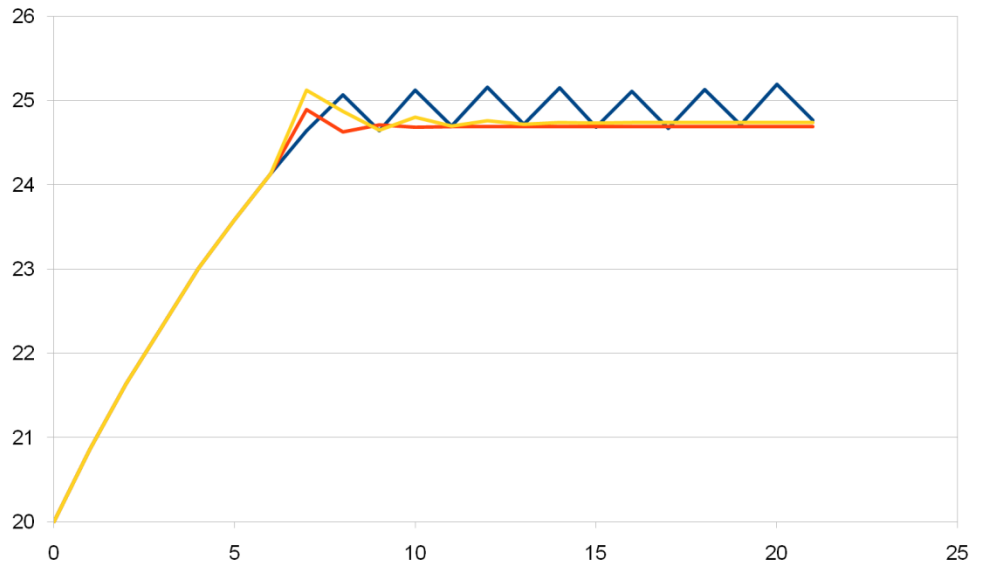
Esportando i dati tempo-temperatura è possibile costruire il seguente diagramma in cui si ha:

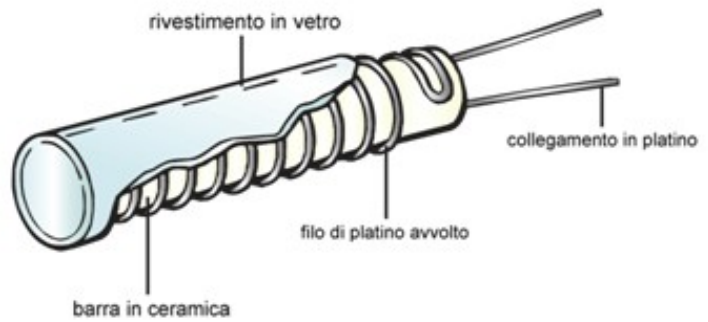
- linea BLU: controllo on-off
- linea ROSSA: controllo proporzionale  $K_p=150$
- linea GIALLA: controllo proporzionale  $K_p=180$

Si può osservare che il controllo proporzionale annulla l'oscillazione tipica del controllo ON-OFF. Non è in grado di ridurre a 0 l'errore ( $T_{sp}-T$ ). Oltre un certo valore di  $K_p$  il sistema diventa *instabile!*

Controllo di Temperatura

t	ON-OFF	PROP. $K_p=150$	PROP. $K_p=180$
s	$T(^{\circ}C)$	$T(^{\circ}C)$	$T(^{\circ}C)$
0	20	20	20
1	20,86	20,86	20,86
2	21,64	21,64	21,64
3	22,32	22,32	22,32
4	23,01	23,01	23,01
5	23,58	23,58	23,58
6	24,13	24,13	24,13
7	24,64	24,89	25,12
8	25,07	24,63	24,87
9	24,64	24,71	24,65
10	25,12	24,68	24,8
11	24,7	24,69	24,7
12	25,16	24,69	24,76
13	24,72	24,69	24,72
14	25,15	24,69	24,74
15	24,68	24,69	24,73
16	25,11	24,69	24,74
17	24,67	24,69	24,74
18	25,13	24,69	24,74
19	24,71	24,69	24,74
20	25,19	24,69	24,74
21	24,77	24,69	24,74





Molte industrie utilizzano le termoresistenze per misurare la temperatura e, la maggior parte di questi dispositivi, utilizza un sensore Pt100 o Pt1000. Questi due sensori di temperatura hanno caratteristiche simili, ma la loro differenza nella resistenza nominale determina quale sia la scelta ideale in base alla propria applicazione.

I rilevatori a resistenza di temperatura (RTD – Resistance temperature detectors), detti anche termoresistenze, sono noti dispositivi di misura della temperatura grazie alla loro affidabilità, accuratezza, versatilità, ripetibilità e facilità di installazione.

Il principio di base di una termoresistenza è che il suo sensore a filo, realizzato in un metallo con una resistenza elettrica nota, cambia il suo valore di resistenza quando la temperatura sale o scende. Sebbene le termoresistenze abbiano alcune limitazioni, tra cui una temperatura massima di misura di circa 600 ° C, nel complesso rappresentano la soluzione di misura della temperatura ideale per una moltitudine di processi.

### PERCHÉ UTILIZZARE UN SENSORE AL PLATINO

I fili dell'elemento di misura di una termoresistenza possono essere realizzati in nichel, rame o tungsteno, ma il platino (Pt) è oggi di gran lunga il metallo più popolare utilizzato. È più costoso di altri materiali, ma il platino ha diverse caratteristiche che lo rendono particolarmente adatto per le misure di temperatura, tra cui:

- Relazione quasi lineare tra resistenza e temperatura
- Alta resistività (59  $\Omega$  / cmf rispetto a 36  $\Omega$  / cmf per il nichel)
- Resistenza elettrica non degradabile nel tempo
- Eccellente stabilità
- Ottima passività chimica
- Elevata resistenza alla contaminazione

### DIFFERENZA TRA PT100 E PT1000

Tra le termoresistenze in platino, le Pt100 e Pt1000 sono le più comuni. Le Pt100 hanno una resistenza nominale di 100  $\Omega$  al punto di fusione del ghiaccio (0 ° C). La resistenza nominale delle Pt1000 a 0 ° C è invece di 1.000  $\Omega$ . La linearità della curva caratteristica, il campo di temperatura operativo e il tempo di risposta sono gli stessi per entrambi. Anche il coefficiente di temperatura della resistenza è lo stesso.

Tuttavia, a causa della diversa resistenza nominale, le letture delle sonde Pt1000 sono maggiori di un fattore 10 rispetto alle Pt100. Questa differenza diventa evidente quando si confrontano configurazioni a 2 fili, in cui si verifica l'errore di misura. Ad esempio, l'errore di misura in una Pt100 potrebbe essere di + 1,0 ° C, e quello di una Pt1000 con la stessa esecuzione potrebbe essere di + 0,1 ° C.

## COME SCEGLIERE IL GIUSTO SENSORE AL PLATINO

Entrambi i tipi di sensori funzionano bene nelle configurazioni a 3 e 4 fili, dove i cavi e i connettori aggiuntivi compensano gli effetti della resistenza dei fili conduttori sulla misura della temperatura. Le due tipologie di configurazione hanno un prezzo simile. Le sonde Pt100, tuttavia, sono più popolari delle Pt1000 per un paio di motivi:

Una sonda Pt100 è disponibile sia in esecuzione a filo avvolto che a film sottile, offrendo agli utenti la possibilità di scelta e flessibilità. Le sonde Pt1000 sono quasi sempre solo a film sottile

Poiché il loro uso è così diffuso in tutti i settori, le sonde Pt100 sono compatibili con una vasta gamma di strumenti e processi.

Quindi, perché si dovrebbe optare per la sonda Pt1000? Le situazioni in cui la maggiore resistenza nominale ha un vantaggio evidente sono le seguenti:

Una sonda Pt1000 è migliore nella configurazione a 2 fili e quando viene utilizzata con lunghezze di cavo più lunghe. Minore è il numero di fili e più lunghi essi sono, maggiore è la resistenza che si aggiunge alle letture, causando in tal modo imprecisioni. La maggiore resistenza nominale della sonda Pt1000 compensa questi errori aggiunti

Una sonda Pt1000 è migliore per le applicazioni alimentate a batteria. Un sensore con una resistenza nominale più elevata utilizza meno corrente elettrica e, pertanto, richiede meno energia per funzionare. Il consumo energetico ridotto prolunga la durata della batteria e l'intervallo tra la manutenzione, riducendo i tempi di fermo impianto e i costi

Poiché una Pt1000 consuma meno energia, l'autoriscaldamento è inferiore. Ciò significa meno errori di lettura a causa di temperature superiori a quelle ambientali

In generale, le sonde temperatura Pt100 sono più comunemente utilizzate nelle applicazioni di processo, mentre le Pt1000 sono utilizzate nei settori della refrigerazione, riscaldamento, ventilazione, automotive e dei costruttori di macchine.

## SOSTITUZIONE DELLE TERMORESISTENZE: NOTA SULLE NORME INDUSTRIALI

Le termoresistenze sono facili da sostituire, ma non si tratta semplicemente di sostituirle l'una con l'altra. Il problema a cui gli utenti devono prestare attenzione quando sostituiscono le sonde Pt100 e Pt1000 esistenti è la norma nazionale o internazionale.

La norma U.S.A. più vecchia, ad esempio, indica il coefficiente di temperatura del platino come  $0,00392 \Omega / \Omega / ^\circ \text{C}$  (ohm per ohm per grado centigrado). Nella nuova norma europea DIN / IEC 60751, che viene utilizzata anche in Nord America, è  $0,00385 \Omega / \Omega / ^\circ \text{C}$ . La differenza è trascurabile a temperature più basse, ma diventa evidente al punto di ebollizione dell'acqua ( $100^\circ \text{C}$ ), quando la norma più vecchia leggerà  $139,2 \Omega$  mentre quella più recente leggerà  $138,5 \Omega$ .

## CONVERTIRE LA RESISTENZA PT100/PT1000 IN TEMPERATURA

La variazione di una resistenza elettrica dipende dalla differenza di temperatura e dai coefficienti termici del materiale utilizzato. La resistenza nominale a 0 °C è pari a 100 Ω per Pt100 e 1 kΩ per Pt1000.

I coefficienti termici del platino sono pari a

$$A = 3,91 \cdot 10^{-3} \text{ [K-1]}$$

$$B = -0,588 \cdot 10^{-6} \text{ [K-2]}$$

La formula generale per calcolare una resistenza in funzione della temperatura è la seguente:

$$R(\vartheta) = R_{\vartheta 0} \cdot (1 + A \cdot (\vartheta - \vartheta_0) + B \cdot (\vartheta - \vartheta_0)^2)$$

R(ϑ):	Resistenza in funzione della temperatura [Ω]
R0:	Resistenza nominale elettrica a 0 °C [Ω]
ϑ:	Temperatura [°C]
ϑ0:	Temperatura di riferimento [°C]
A:	Coefficiente termico lineare [K-1]
B:	Coefficiente termico quadrato [K-2]

Il range di temperatura da 0 °C a 100 °C può essere descritto con un'equazione approssimata lineare. A tale scopo si sceglie la temperatura di riferimento ϑ0 = 0 °C.

I coefficienti A e B vengono sostituiti dal coefficiente medio  $\alpha = 3,91 \cdot 10^{-3} \text{ K-1}$ .

$$R(\vartheta) = R_0 \cdot (1 + \alpha \cdot \vartheta)$$

R(ϑ):	Resistenza in funzione della temperatura [Ω]
R0:	Resistenza nominale elettrica a 0 °C [Ω]
ϑ:	Temperatura [°C]
α:	Coefficiente termico medio [K-1]

Modificando la formula è possibile convertire in temperatura la resistenza misurata:

$$\vartheta(R) = \frac{R - R_0}{R_0 \cdot \alpha}$$

$$\vartheta(R) = \frac{\Delta R}{R_0 \cdot \alpha} \quad \text{con} \quad \Delta R = R - R_0$$

ϑ(R):	Temperatura in funzione della resistenza [°C]
α:	Coefficiente termico medio [K-1]
R:	Resistenza misurata della sonda Pt [Ω]
R0:	Resistenza nominale elettrica a 0 °C [Ω]
ΔR:	Variazione misurata della resistenza [Ω]

Quindi nota la variazione di resistenza si può risalire alla temperatura:

$$\Delta R = 2 \cdot R \cdot (V_{ab}/E) / (0.5 - V_{ab}/E); \rightarrow R_{pt1000} = R + \Delta R \rightarrow T = (R_{pt1000}/1000 - 1) / 0.00385$$

## CURVA CARATTERISTICA DELLE TERMORESISTENZE

La curva caratteristica rappresenta il rapporto lineare tra la resistenza elettrica e la temperatura.

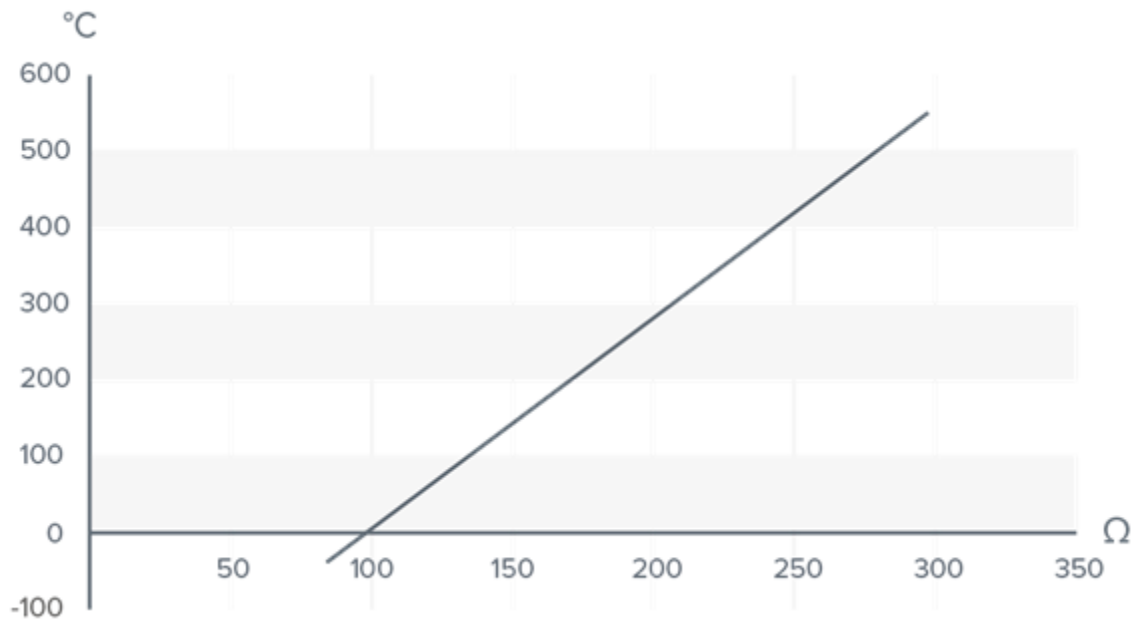
I valori concreti di Pt100 e Pt1000 possono essere dedotti graficamente dalle curve caratteristiche Pt100 / Pt1000 o letti direttamente dalle tabelle Pt100 / Pt1000.

Il platino si presta particolarmente bene come materiale, grazie alla sua elevata stabilità a lungo termine e alle caratteristiche elettriche piuttosto costanti alle alte temperature.

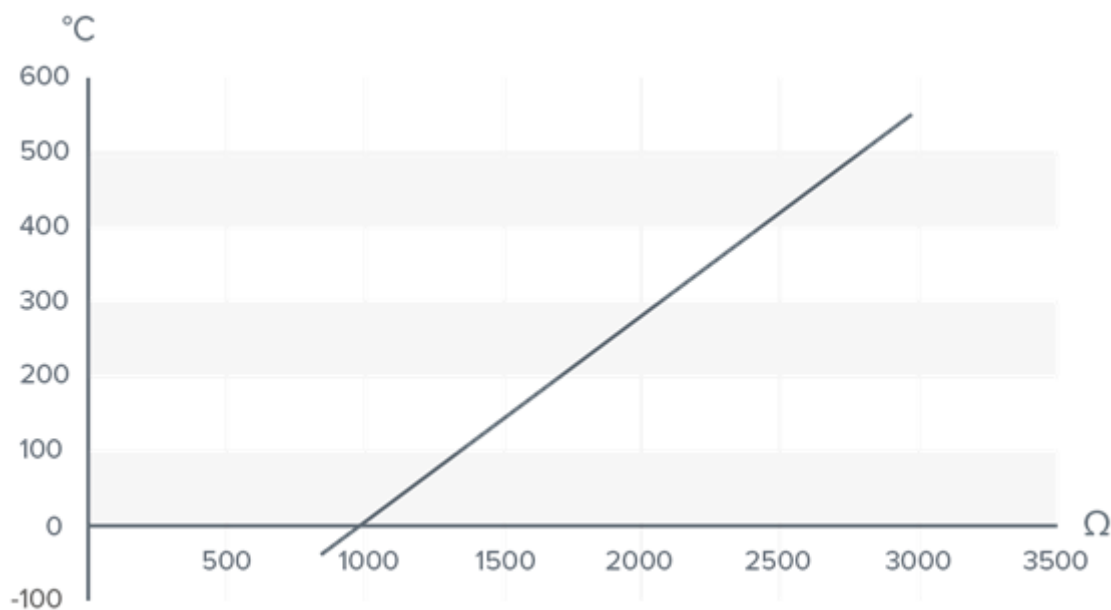
Per questo la curva caratteristica delle resistenze al platino è estremamente lineare anche a fronte di temperature elevate.

Aggiungendo al platino altre sostanze si ottengono risultati ancora migliori.

### CURVA CARATTERISTICA PT100



### CURVA CARATTERISTICA PT1000



## TERMOCOPPIE

Una termocoppia è un tipo di sensore di temperatura che sfrutta l'effetto termoelettrico per misurare temperature comprese tra -170 °C e +1200 °C. Una termocoppia è composta da due fili metallici diversi.



Thomas Johann Seebeck è stato un fisico estone, scopritore dell'effetto termoelettrico

I fili metallici sono collegati tra loro in un unico punto, solitamente la punta della termocoppia, nota come giunzione calda, giunzione di misura, punto di rilevamento o giunzione di rilevamento. Come suggerisce il nome, questa giunzione è esposta alla fonte di calore di interesse.

L'estremità opposta dei fili metallici è nota come giunzione fredda ed è collegata al dispositivo di misura. In genere, la giunzione fredda non è esposta allo stesso livello di energia termica della giunzione calda.

### Effetto termoelettrico

Tutte le termocoppie funzionano allo stesso modo: generano una piccola tensione quando sono esposte al calore.

Quando si riscalda un pezzo di metallo, il calore eccita gli elettroni presenti nel metallo, facendoli oscillare. Man mano che il metallo si riscalda, più elettroni tendono a "diffondersi" e a spostarsi verso l'estremità più fredda del metallo.

Ciò fa sì che l'estremità più calda abbia una carica leggermente positiva e quella più fredda una carica leggermente negativa, creando una differenza di tensione. Questo è noto come effetto termoelettrico o **effetto Seebeck**, dal nome dello scienziato tedesco Thomas Seebeck, che scoprì questo fenomeno nel 1821.

Thomas Seebeck dimostrò che il potenziale elettrico  $V$ , che nasce in un giunto è funzione della temperatura del giunto e tale potenziale dipende dalle caratteristiche fisiche della giunzione, secondo la relazione:

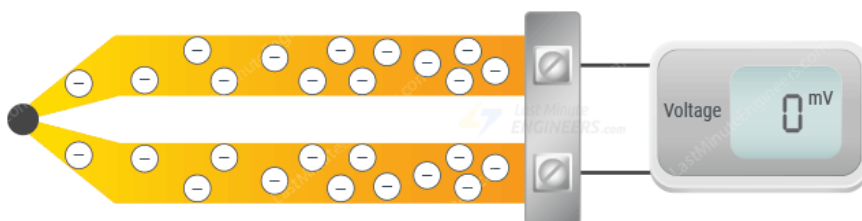
$$V_j = \alpha * T_j$$

dove  $\alpha$  è il coefficiente di Seebeck, che dipende dalla temperatura e dalle caratteristiche fisiche della giunzione.

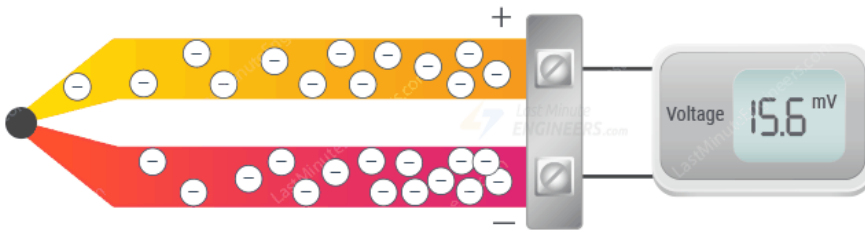
### Funzionamento della termocoppia

Una termocoppia funziona basandosi sul movimento degli elettroni nei suoi fili metallici, causato dalla differenza di calore tra le giunzioni calda e fredda.

Se i due fili della termocoppia fossero costituiti dallo stesso tipo di metallo, ad esempio rame, gli elettroni in entrambi i fili si allontanerebbero dal calore e si accumulerebbero nelle estremità fredde in quantità uguali, senza che si verifici alcuna differenza di tensione misurabile.



Ma se ricordate, le termocoppie sono costituite da due diversi tipi di filo metallico. Quindi, se due fili della termocoppia fossero costituiti da materiali diversi, ad esempio uno di rame e uno di ferro, i metalli condurrebbero il calore in modo diverso, determinando un gradiente di temperatura distinto. Ciò causa un accumulo variabile di elettroni alle estremità fredde, con conseguente differenza di tensione misurabile.



Questa differenza di tensione è molto piccola. La variazione effettiva di tensione per grado Celsius è minuscola. Ad esempio, per una termocoppia di tipo K, la variazione è di circa  $41 \mu\text{V}/^\circ\text{C}$ .

### Cavi per termocoppie

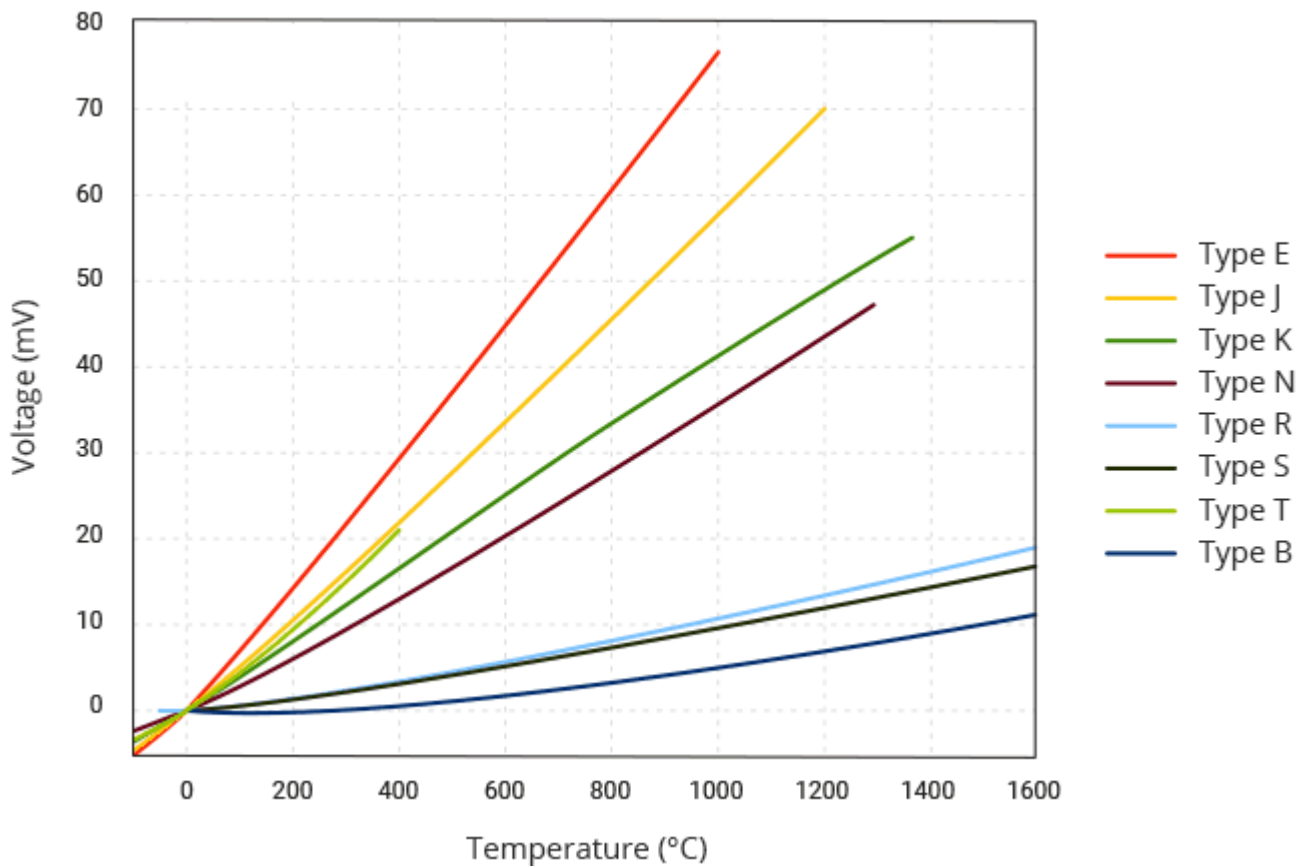
Quando esposti al calore, gli elettroni in ciascun filo della termocoppia reagiscono in modo diverso e si muovono a velocità diverse.

Il filo in cui si accumulano più elettroni nella giunzione fredda è chiamato filo conduttore negativo, mentre il filo in cui si accumulano meno elettroni nella giunzione fredda è chiamato filo conduttore positivo.

Questa differenza di carica tra i conduttori positivo e negativo può essere misurata e utilizzata per determinare la temperatura nella giunzione calda.

### Curve caratteristiche termocoppia

Esistono diversi tipi di termocoppie, come Tipo J, Tipo K, Tipo E, Tipo T, ecc., in base alla combinazione di metalli o leghe utilizzate per i due fili. Ogni tipo di termocoppia ha le proprie caratteristiche funzionali, di intervallo di temperatura, di precisione e di applicazione.



Esistono diversi tipi di termocoppie, che si differenziano per i metalli dei fili conduttori. Infatti, la combinazione di vari metalli determina le caratteristiche e le applicazioni più adatte: così si possono distinguere termocoppie a metallo base e termocoppie a metallo nobile.

Le termocoppie a metallo base, di cui i tipi K, J, T, E ed N, sono composte da metalli comuni come nichel, ferro e rame, e sono le più diffuse per la loro economicità e versatilità in molteplici applicazioni a bassa e media temperatura. Per la precisione:

- la termocoppia tipo K è composta da nichel-cromo e nichel-alluminio (-200 a +1200°C);
- la termocoppia tipo J è composta da ferro e costantana (lega rame-nichel), con un tipico range di temperatura da -210 a +1200°C;
- la termocoppia tipo T è composta da rame e costantana (da -270 a +400°C);
- la termocoppia tipo E è composta da nichel-cromo e costantana (da -270 a +980°C);
- la termocoppia tipo N è composta da nicrosil (nichel-cromo-silicio) e nisil (nichel-silicio), con un tipico range di temperatura da -270 a +1300°C.

Le termocoppie a metallo nobile sono realizzate con metalli pregiati, adatti a misurare temperature elevate (sopra i 1000°C). Offrono maggiore stabilità ma a costi notevolmente più alti, motivo per cui sono meno diffuse rispetto alle termocoppie a metallo base.

Tuttavia, la termocoppia più utilizzata nelle applicazioni industriali è quella di tipo K, perché risponde in modo prevedibile in un ampio intervallo di temperatura (da -175 °C a +1100 °C circa) e ha una sensibilità di circa 41 µV/°C. È costituita da un filo positivo in Chromel (lega di nichel-cromo) e da un filo negativo in Alumel (lega di nichel-alluminio).

Le principali caratteristica negative delle termocoppie sono:

- la bassa sensibilità per cui la tensione di uscita deve essere amplificata.
- non linearità su un ampio campo di misura

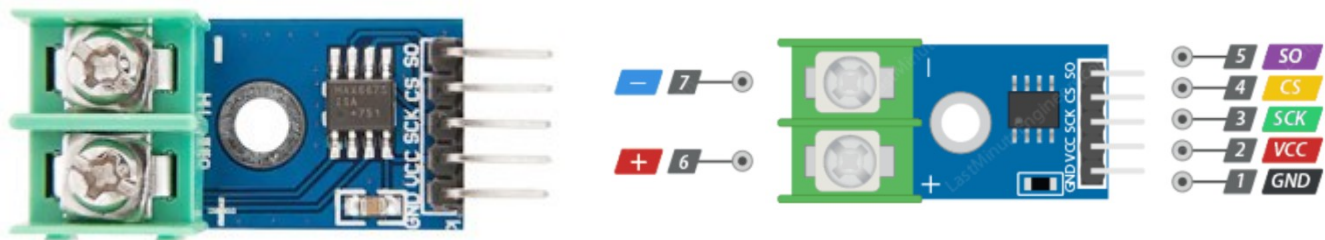
## DIGITALIZZATORE DI TERMOCOPPIA

Per rendere utile la termocoppia, è necessario calibrarla testandola a temperature note e registrando le tensioni generate. È quindi possibile utilizzare una formula per calcolare la temperatura in base alla tensione misurata.

È qui che entrano in gioco i circuiti integrati digitalizzatori per termocoppie come il MAX6675. Questi circuiti integrati (IC) sono progettati per eseguire la compensazione della giunzione fredda e digitalizzare il segnale ricevuto da una termocoppia. Ogni termocoppia ha un suo specifico digitalizzatore (essendo le curve diverse).

### Modulo MAX6675 per termocoppia K

Il cuore della scheda è un circuito integrato digitalizzatore per termocoppia di tipo K con compensazione della giunzione fredda di Microchip, il MAX6675.



Il breakout accetta una termocoppia standard di tipo K a un'estremità, digitalizza la temperatura misurata e invia i dati all'altra estremità tramite un'interfaccia SPI (seriale), interpretandoli e traducendoli per consentirne una semplice lettura.

Il circuito integrato MAX6675 include un convertitore analogico-digitale (ADC) a 12 bit, il che significa che il circuito integrato può rilevare le temperature fino a 0,25 °C (risoluzione a **12 bit → 4096**).

Il MAX6675 può misurare temperature comprese tra 0 °C e +1024 °C con una precisione di ±3 °C.

Tuttavia, è importante tenere presente che l'intervallo dipende dal tipo di sonda utilizzata.

Oltre al basso costo, alle dimensioni ridotte e all'ampio intervallo, il MAX6675 funziona da 3V a 5,5 V e assorbe circa 700 µA. La corrente massima che può assorbire è di circa 1,5 mA.

## TERMOCOPPIA TIPO K

La termocoppia di tipo K è la più diffusa nel settore industriale.

È costituita da una combinazione di fili a base di nichel (solitamente cromel/alumel).

Economica ma al tempo stesso affidabile garantendo una misurazione accurata.

Gli intervalli di misurazione sono generalmente compresi tra -200 e +1260 gradi centigradi con una deviazione standard di +0,75%. Il nichel impiegato (resistente alla corrosione e ossidazione) permette l'impiego in una vasta gamma di applicazioni.

Nello specifico, il filo della termocoppia di tipo K include un polo positivo composto per circa il 90% di nichel e per il 10% da cromo, ed un altro negativo composto per il 95% da nichel, 2% da alluminio, 2% da manganese e un restante 1% da silicio.

Dall'analisi della dipendenza del coefficiente di Seebeck in funzione della temperatura si vede che nel range da 0 a 1000°C la termocoppia K ha un comportamento pressoché lineare in quanto il coefficiente alfa è approssimativamente costante pari a 0.041.

Di conseguenza queste termocoppie sono molto usate perché non necessitano del circuito di linearizzazione.

### L'approssimazione lineare

Sebbene la risposta di una termocoppia non sia perfettamente lineare per calcoli rapidi si utilizza la seguente equazione:

$$V \approx \alpha \cdot (T - T_{rif})$$

Dove:

- V= Tensione in uscita (millivolt, mV).
- $\alpha$  = coefficiente di Seebeck medio per il Tipo K (0,041 mV/°C).
- T: Temperatura del giunto caldo (°C).
- Trif: Temperatura del giunto di riferimento (solitamente 0°C).

Quindi, assumendo il riferimento a 0°C, l'equazione semplificata è:

$$V = 0.041 T \text{ (volt)}$$

*NOTA: A 500°C l'errore è di circa 3.5°C mentre a 1000°C è di circa 6.7°C.*

## SONDA TERMOCOPPIA TIPO K COMMERCIALE

La sonda termocoppia in dotazione con il modulo è lunga circa 18 pollici e ha un intervallo di misurazione da 0 °C a 80 °C.



Il terminale rosso della sonda è il polo positivo realizzato in Chromel (lega di nichel-cromo), mentre il terminale blu è il polo negativo realizzato in Alumel (lega di nichel-alluminio).

La sonda è dotata di isolamento in fibra di vetro, un materiale noto per la sua capacità di resistere ad alte temperature e condizioni difficili. Questo la rende una scelta adatta per un'ampia gamma di progetti.

La sonda termina con un attacco filettato M6. Questo tipo di attacco consente di fissare la termocoppia a un oggetto, ad esempio un dissipatore di calore, dove può essere avvitata o fissata con un dado.

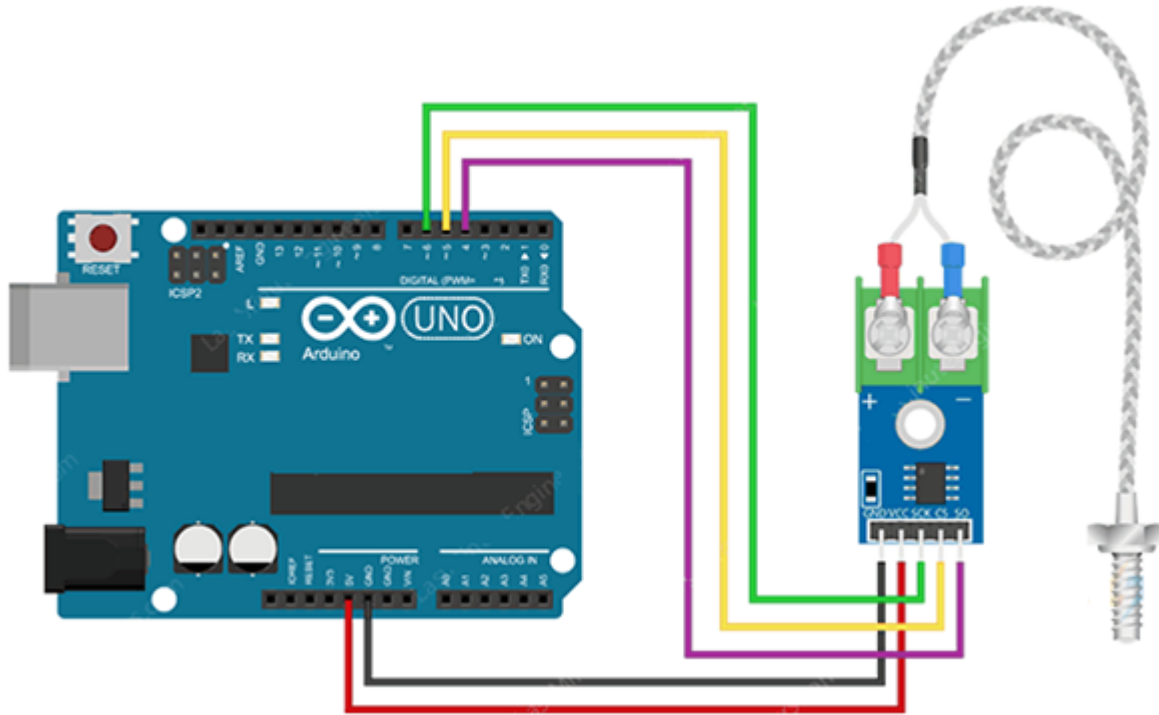
<b>Tensione di esercizio</b>	da 3,0 a 5,5 V
<b>Interfaccia</b>	SPI ad alta velocità
<b>Consumo attuale</b>	700µA (tipico), 1,5mA (max)
<b>Intervallo di temperatura</b>	0 – 1024 °C (di MAX6675) 0 – 80 °C (della sonda in dotazione)
<b>Precisione</b>	±3 °C
<b>Risoluzione</b>	12 bit (0,25 °C)
<b>Tempo di conversione</b>	~170 ms

## COLLEGAMENTO DEL MODULO MAX6675 A UN ARDUINO

Collegiamo il modulo MAX6675 all'Arduino. I collegamenti sono semplici.

Per prima cosa, collega il pin VCC del modulo ai 5 V dell'Arduino e il pin GND a terra. Ora colleghiamo i tre pin digitali da utilizzare come interfaccia SPI. Nell'esempio utilizziamo i pin 4, 5 e 6. Infine, colleghiamo la termocoppia al modulo.

Il terminale rosso (cavo Chromel) della termocoppia al terminale '+' del modulo e il terminale blu (cavo Alumel) al terminale '-'.



Poiché il modulo consuma pochissima energia (meno di 1,5 mA), è possibile alimentarlo tramite un pin di uscita digitale del microcontrollore. Se si decide di utilizzare questo metodo e di spegnere il MAX6675 tra una lettura e l'altra, è consigliabile attendere alcuni secondi dopo la riaccensione prima di tentare una lettura.

Codice

```
#include "max6675.h"

// Define the Arduino pins, the MAX6675 module is connected to
int SO_PIN = 4; // Serial Out (SO) pin
int CS_PIN = 5; // Chip Select (CS) pin
int SCK_PIN = 6; // Clock (SCK) pin

// Create an instance of the MAX6675 class with the specified pins
MAX6675 thermocouple(SCK_PIN, CS_PIN, SO_PIN);

void setup() {
  Serial.begin(9600);
  delay(500);
}

void loop() {
  // Read the temperature in Celsius
  Serial.print("Temperature: ");
  Serial.print(thermocouple.readCelsius());
  Serial.print("\xC2\xB0"); // shows degree symbol °
  Serial.print("C");

  delay(1000);
}
```

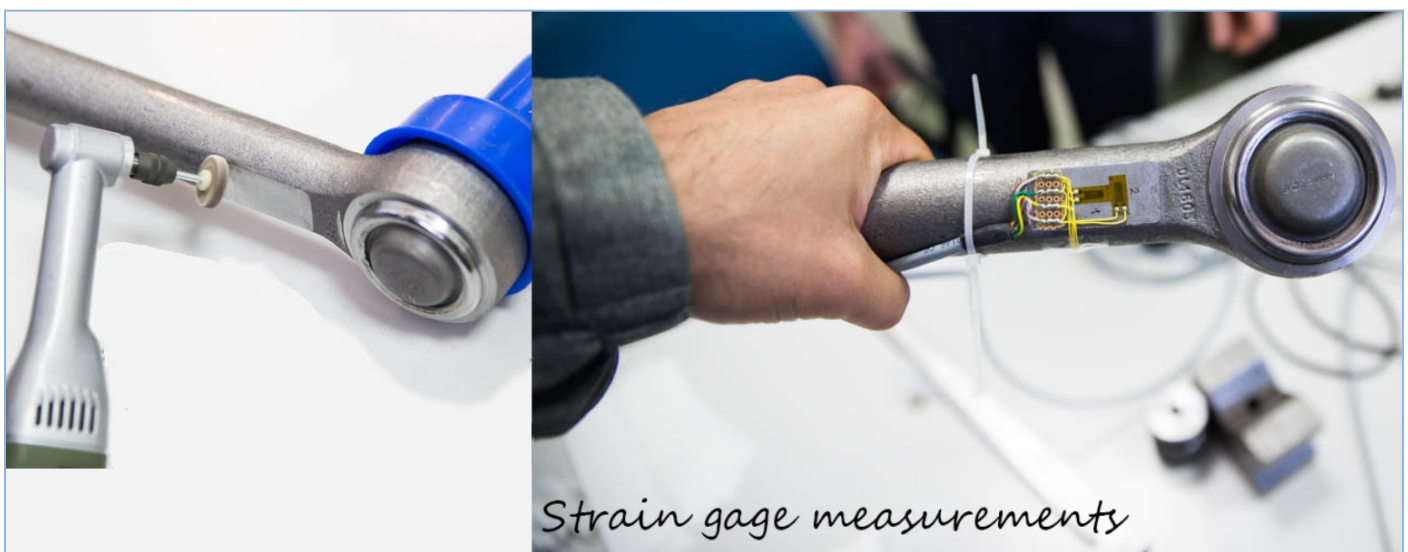
L'estensimetro è uno strumento di misura utilizzato per rilevare piccole deformazioni dimensionali di un corpo sottoposto a sollecitazioni meccaniche o termiche (es. applicazione di carichi o variazioni di temperatura).

Conoscendo a priori le caratteristiche meccanico/fisiche del materiale, misurando le deformazioni si possono facilmente ricavare i carichi a cui il materiale è sottoposto.

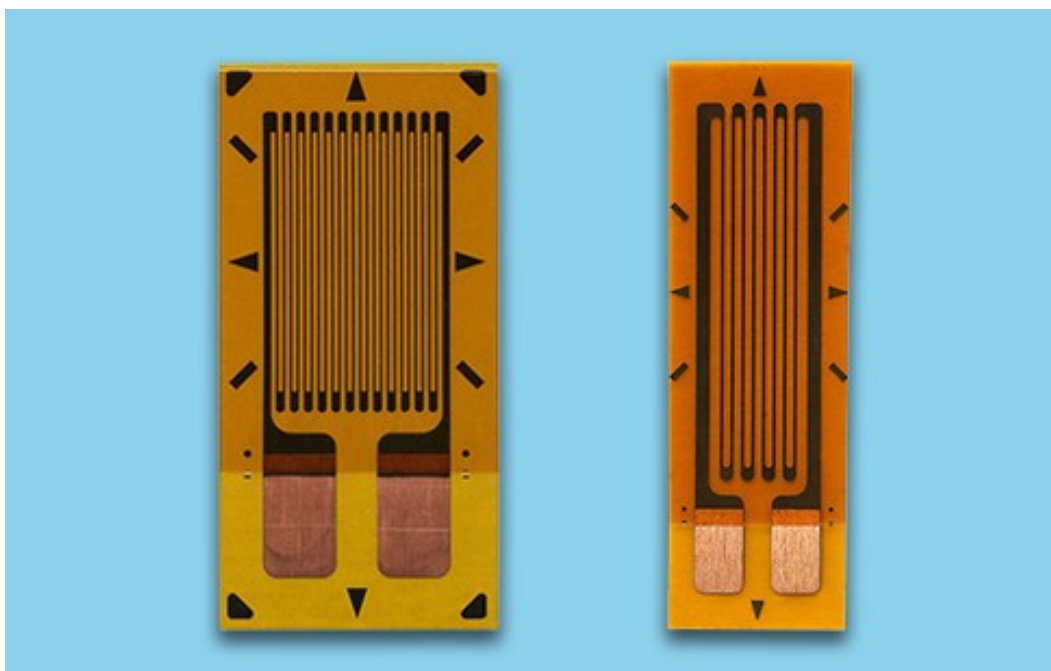
Inoltre, utilizzando estensimetri di giusta tipologia e applicandoli in modo oculato, si possono rilevare la direzione e il verso di queste deformazioni, e di conseguenza il vettore delle forze applicato al materiale sotto esame.

I campi d'applicazione sono molteplici:

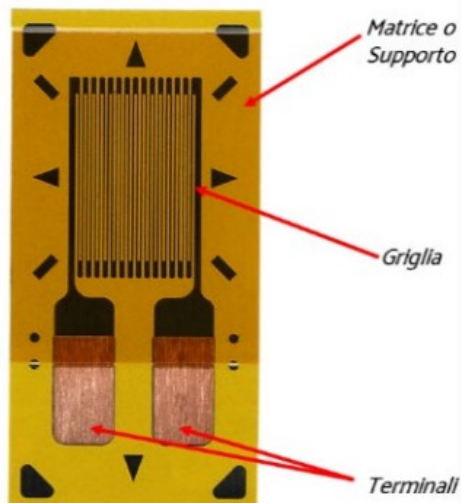
- prove in laboratorio su componenti meccanici o materiali;
- analisi statiche e dinamiche di componenti o sistemi meccanici già montati in situ;
- monitoraggio di componenti o sistemi strutturali;
- elemento sensore per trasduttori di grandezze meccaniche;



misure del carico assiale su braccetti di sterzo con estensimetri elettrici a resistenza



Gli elementi principali di un estensimetro sono la matrice e la griglia.



I materiali più comuni con cui si realizzano le griglie sono:

Tipo di lega	Composizione %
Costantana	45Ni, 55Cu
Karma	74 Ni, 20 Cr, 3 Fe, 3 Al
Isoelastica	36 Ni, 55,5 Fe, 8 Cr, 0,5 Mo
Platino Tungsteno	92 Pt, 8 W
Nicromo	80 Ni, 20 Cr
Kanthal	30 Cr, 30 Fe, 7,5 Al, 32,5 Co

Le griglie sono disponibili in lunghezze che variano da 0,2 mm a 120 mm.

Le matrici (dette anche supporto) vengono realizzate con delle resine anche rinforzate con fibra di vetro per migliorarne le prestazioni alle alte temperature.

Tipi di Resine	Temperatura di Utilizzo
Resina Epossidica	da -50 a 100 °C
Resina Fenolica	da -200 a 250 °C
Resina Poliammidica	da -200 a 200 °C
Resina Epossidica + Fibra di vetro	da -269 a 230 °C
Resina Fenolica + Fibra di vetro	da -200 a 300 °C
Resina Epossidica Fenolica + Fibra di vetro	da -269 a 300 °C
Piastrina metallica	da -269 a 300 °C

La piastrina metallica è il supporto tipico degli estensimetri saldabili applicati alle superfici con una saldatura per punti.

## RESISTENZA DEGLI ESTENSIMETRI

Gli estensimetri sono disponibili con resistenze da 120, 350 e 1000 Ohm.

Per le classiche misure di stress analysis si possono usare sia i 120 che i 350 Ohm (questi ultimi soprattutto su materiali cattivi conduttori di calore).

Per realizzare i trasduttori estensimetrici si usano sia i 350 che i 1000 Ohm.

La variazione di resistenza elettrica di un estensimetro deformato è dell'ordine delle frazioni di  $\Omega$ .

Queste variazioni di resistenza vanno misurate su circuiti che hanno resistenze assolute di centinaia di  $\Omega$ .

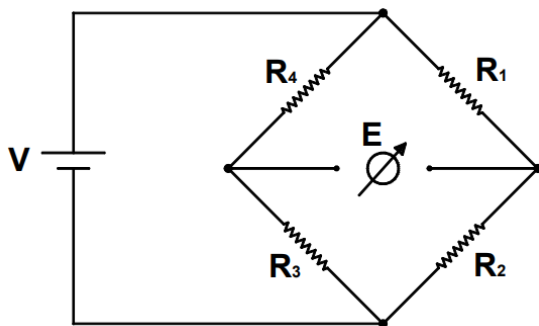
Ciò richiede l'utilizzo di un particolare circuito di misura, detto circuito a ponte o ponte di Wheatstone.

## IL PONTE DI WHEATSTONE

È costituito da 4 resistenze elettriche che occupano 4 lati di un rombo.

Il ponte viene alimentato da una tensione  $V$  ai capi della «diagonale di alimentazione».

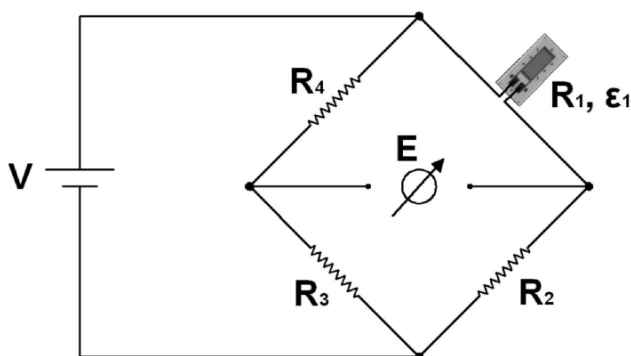
Ai capi dell'altra diagonale, detta «diagonale di misura», si misura la tensione di sbilanciamento  $E_{\text{ponte}}$ .



$$E = V \left( \frac{R_1}{R_1 + R_2} - \frac{R_4}{R_3 + R_4} \right) = V \frac{R_1 R_3 - R_2 R_4}{(R_1 + R_2)(R_3 + R_4)}$$

## COLLEGAMENTO A QUARTO DI PONTE

Si utilizza UN solo estensimetro che occupa uno dei lati del ponte.



$$E_{\text{ponte}} = \frac{V_{cc}}{4} * \frac{\Delta R}{R}$$

Scegliendo le altre 3 resistenze  $R_2, R_3$  e  $R_4$  (di precisione) di valore uguale a quella  $R_g$  dell'estensimetro a riposo l'equazione del ponte si semplifica nella seguente formula (valida per variazioni piccole rispetto alle  $R$ ):

$$E_{\text{ponte}} = \frac{V_{cc}}{4} * \frac{\Delta R}{R_g}$$

con  $R=R_1=R_2=R_3=R_4$  e  $\Delta R= R_1-R_{g_{\text{nominale}}}$  e  $R_g$  = resistenza iniziale dell'estensimetro [ $\Omega$ ]

## LEGAME DEFORMAZIONE ELASTICA E VARIAZIONE DI RESISTENZA ELETTRICA

Il legame tra deformazione e variazione di resistenza elettrica si esprime con la seguente relazione:

$$\varepsilon = \frac{1}{k} \cdot \frac{\Delta R}{R_g}$$

$\varepsilon$  = deformazione [ $\mu\text{m}$ ]

k = gage factor

$R_g$  = Resistenza iniziale dell'estensimetro [ $\Omega$ ]

$\Delta R$  = Variazione di Resistenza dell'estensimetro [ $\Omega$ ] =  $R_f - R_g$

$R_f$  = Resistenza finale dell'estensimetro [ $\Omega$ ]

Il gage factor K (anche detto fattore di taratura o sensibilità alla deformazione) è una quantità adimensionale che viene ottenuta sperimentalmente. I valori tipici del gage factor in funzione del tipo di griglia sono rappresentati nella Tabella:

Tipo di Lega	Gage Factor Nominale a +24°C
Costantana	2.0
Karma	2.1
Isoelastica	3.2
Nicromo	2.0
Platino Tungsteno	4.0
Kanthal	2.4

## MISURA DELLA DEFORMAZIONE E DELLA FORZA ASSIALE

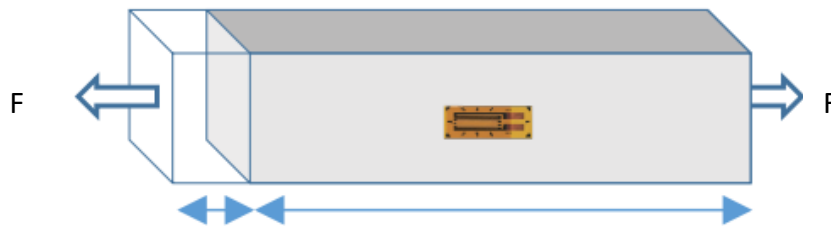
Su un provino di lunghezza  $L_0$  applichiamo una forza di trazione F.

Per effetto della forza il provino si allunga.  $\Delta L$  rappresenta la variazione di lunghezza.

Se sulla superficie del provino è incollato l'estensimetro come disposto in figura allora anch'esso subirà la stessa deformazione.

Il rapporto tra la variazione di lunghezza  $\Delta L$  e la lunghezza iniziale L è nota come deformazione meccanica e viene indicata con la lettera greca  $\varepsilon$ .

$$\frac{\Delta L}{L} = \varepsilon \quad \left[ \frac{m}{m} \right]$$



Essendo il rapporto tra due lunghezze la deformazione  $\varepsilon$  è una quantità adimensionale.

Essendo la  $\varepsilon$  una quantità piccola si preferisce usare un sottomultiplo del metro e cioè il  $\mu\text{m} = 10^{-6}$  m.

Per cui la deformazione viene espressa in  $\mu\text{m}/\text{m}$ . È ormai diventato di uso comune esprimere questa quantità in  $\mu\varepsilon$ .

Spesso la deformazione si trova anche espressa in %.

Così è del tutto equivalente scrivere:

$$200 \cdot 10^{-6} \frac{m}{m} = 200 \frac{\mu m}{m} = 200 \mu\varepsilon = 0,02\%$$

Se l'estensimetro è collegato a quarto di ponte si ha:

$$E_{\text{ponte}} = \frac{V_{cc}}{4} * \frac{\Delta R}{R} \quad \text{da cui si ricava la} \quad \Delta R = \frac{4 * E * R}{V}$$

Nota la  $\Delta R$  si ricava la  $\varepsilon$  dalla relazione  $\varepsilon = \frac{1}{k} \cdot \frac{\Delta R}{R_g}$  ed quindi la deformazione  $\Delta L = \varepsilon * L$

Secondo la legge di Hooke, carico specifico e allungamento unitario per piccoli valori del carico sono proporzionali ed il loro rapporto è definito come il modulo di Young o modulo di elasticità E lineare:

$$E = \frac{\sigma}{\varepsilon} \rightarrow \varepsilon = \frac{1}{E} \sigma \rightarrow \frac{\Delta L}{L} = \frac{F}{ES} \rightarrow \boxed{F = E * S * \frac{\Delta L}{L}}$$

## ESERCIZIO

### PROVINO in ACCIAIO sottoposto a trazione

E	206000 N/mm <sup>2</sup>
Sez. 20x5mm	100 mm <sup>2</sup>
L	100 mm

### ESTENSIMETRO in Platino collegato a quarto di ponte

V <sub>cc</sub>	5 V
R <sub>g</sub>	120 ohm
k	4,000
E ponte	0,5 mV

Variazione di resistenza elettrica

$$\Delta R = 0,048 \text{ ohm}$$

Deformazione relativa

$$\varepsilon = 0,0001 \text{ m} \quad \mu\text{m/m} \quad 100 = \mu\varepsilon$$

Allungamento

$$\Delta L = 0,00001 \text{ m} \quad 0,01 \text{ mm}$$

Forza di trazione

$$F = 2060 \text{ N}$$

### AMPLIFICATORE CON CAMPO VARIAZIONE FORZA

F	2000 N	1000 N
$\Delta L$	0,00971 mm	0,004854 mm
$\varepsilon$	0,000097	0,0000485
$\Delta R$	0,0466019 ohm	0,02330097 ohm
E ponte	0,0004854 V	0,000243 V
	0,485 mV	0,243 mV

### Guadagno amplificatore per arrivare a 5V

A	10300
V <sub>amplif. Max.</sub>	5000 mV
V <sub>amplif. Min</sub>	2500 mV

## ESERCIZIO

Valutare l'entità della forza nel caso in cui sia applicata perpendicolarmente all'estremità libera del pezzo (mensola).

**PROVINO ACCIAIO sottoposto a trazione**

E	206000	N/mm <sup>2</sup>
Sez. 20x5mm	100	mm <sup>2</sup>
L	100	mm

**ESTENSIMETRO in Platino collegato a quarto di ponte**

Vcc	12	V
Rg	1000	ohm
k	4	
E ponte	4,5	mV

Variazione di resistenza elettrica

$\Delta R$	1,500	ohm
------------	-------	-----

Deformazione relativa

$\varepsilon$	0,000375	m	$375 \mu\text{m/m}$
Allungamento			
$\Delta L$	0,000038	m	0,0375 mm
Forza di trazione rilevata			
F	7725	N	

**TENSIONE AMPLIFICATA CON FORZA NOTA**

F	7000	N
$\Delta L$	0,03398	mm
$\varepsilon$	0,000340	
$\Delta R$	1,3592233	ohm
E ponte	0,0040777	V
	4,078	mV

**Tensione amplificata di 200**

A	200	
Vamplif. Max.	816	mV

## CELLE DI CARICO

Una cella di carico è un componente elettronico (trasduttore) impiegato per misurare una forza applicata su un oggetto (in genere un componente meccanico) tramite la misura di un segnale elettrico che varia a causa della deformazione che tale forza produce sul componente.

L'applicazione più comune è nei sistemi di pesatura elettronici e nella misura di sforzi meccanici di compressione e trazione.

Questo componente è generalmente costituito da un corpo metallico (Acciaio inox martensitico o Alluminio).

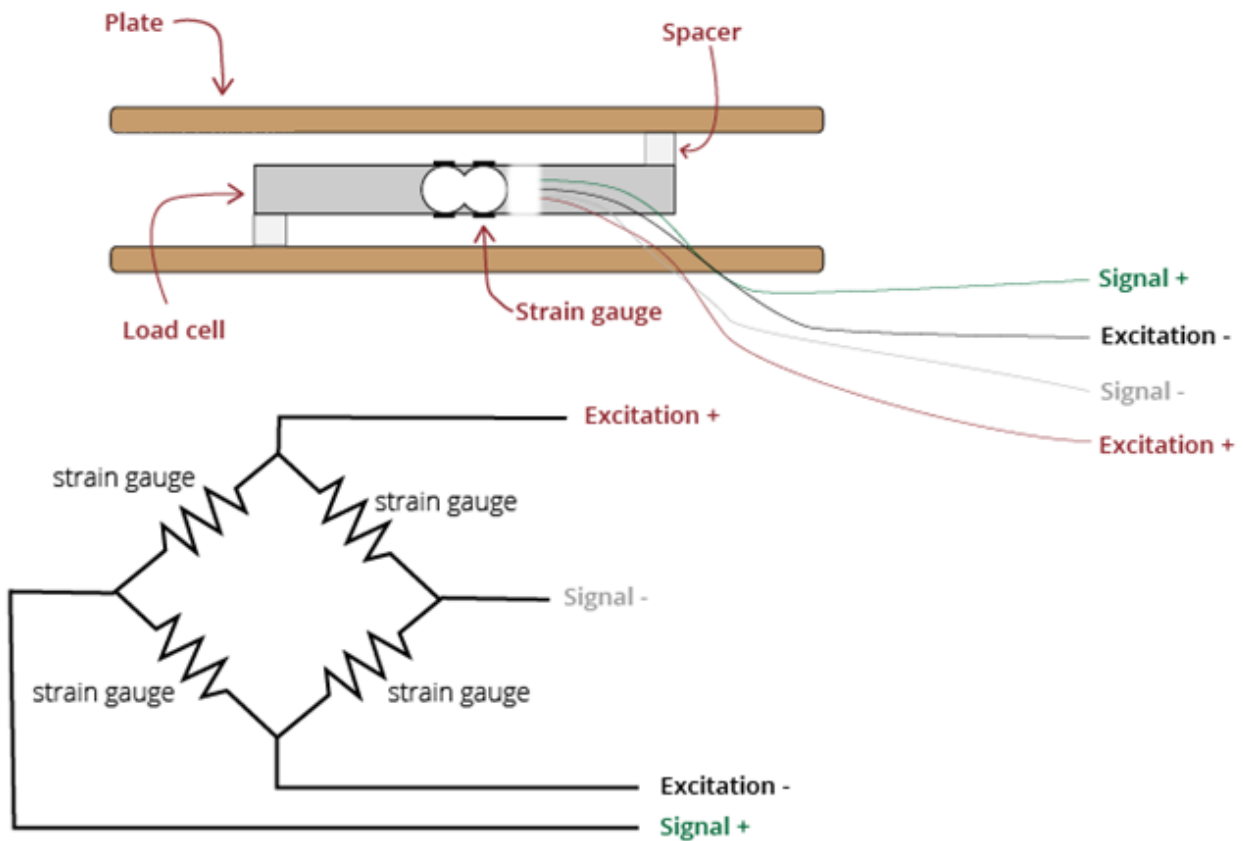
Al corpo della cella di carico vengono applicati uno o più estensimetri che rilevano la deformazione meccanica di compressione o trazione subita dal materiale tramite la variazione di resistenza elettrica causata dalla deformazione stessa.

Per aumentare la sensibilità dello strumento e migliorare così la qualità della misura la scelta più comune è quella di usare quattro estensimetri collegati tra di loro in una configurazione a ponte di Wheatstone, con i due estensimetri adiacenti posti a 90° l'uno rispetto all'altro; questa configurazione permette di aumentare la tensione in uscita dal ponte di circa 2,6 volte (per celle di carico in acciaio) rispetto alla tensione che restituirebbe una configurazione a quarto di ponte, inoltre permette di compensare l'effetto della temperatura che eventualmente comporterebbe errori.

Esistono comunque configurazioni più semplici che prevedono l'impiego di uno o due estensimetri. Il segnale elettrico ottenuto (differenziale) è normalmente dell'ordine di pochi millivolt e richiede un'ulteriore amplificazione ottenibile con un amplificatore da strumentazione prima di essere utilizzato.

Il segnale viene poi eventualmente elaborato mediante un algoritmo per calcolare la forza applicata al trasduttore.





I fili provenienti dalla cella di carico hanno solitamente i seguenti colori:

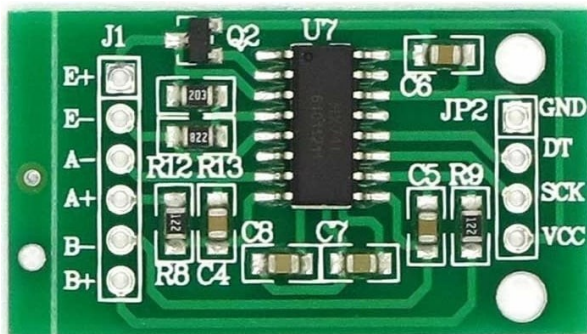
- **Rosso**: VCC (E+)
- **Nero**: GND (E-)
- **Bianco**: Uscita - (LA-)
- **Verde**: Uscita + (LA+)

## SCHEDA ELETTRONICA PER CELLA DI CARICO - HX711

Le celle di carico devono essere accoppiate a degli opportuni amplificatori

Il chip HX711 è composto da un amplificatore a guadagno variabile e da un convertitore analogico-digitale di precisione a 24 bit.

Presenta un'alta velocità di risposta e un'alta immunità ai disturbi.

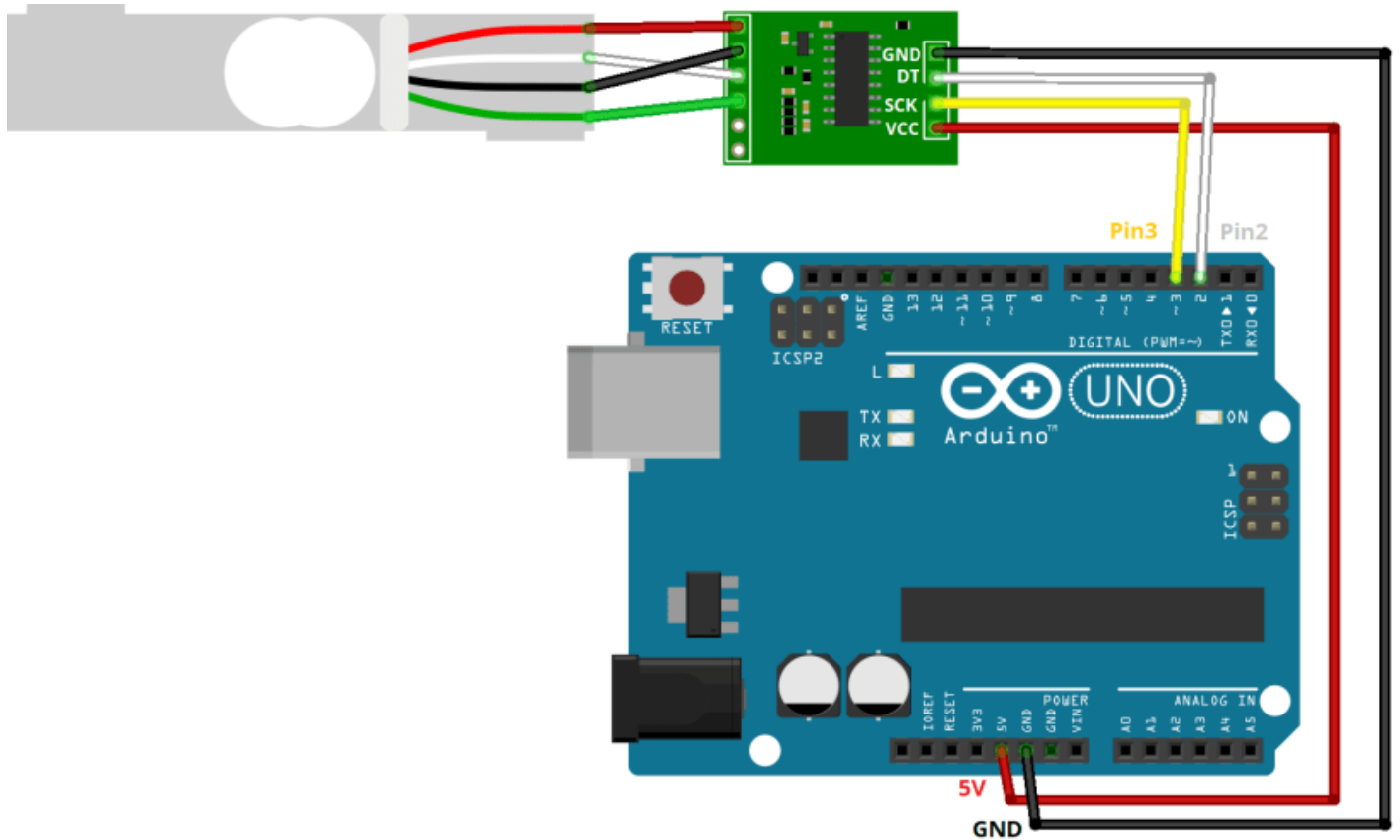


## SCHEMA DI COLLEGAMENTO AD ARDUINO

L'amplificatore HX711 comunica tramite un'interfaccia a due fili.

Puo essere collegato a un qualsiasi pin digitale della scheda Arduino.

Cella di carico	HX711	HX711	Arduino
Rosso(Mi+)	E+	GND	GND
Nero(E-)	E-	DT	pin 2
Bianco(UN-)	UN-	SCK	pin 3
Verde(LA+)	A+	VCC	5V



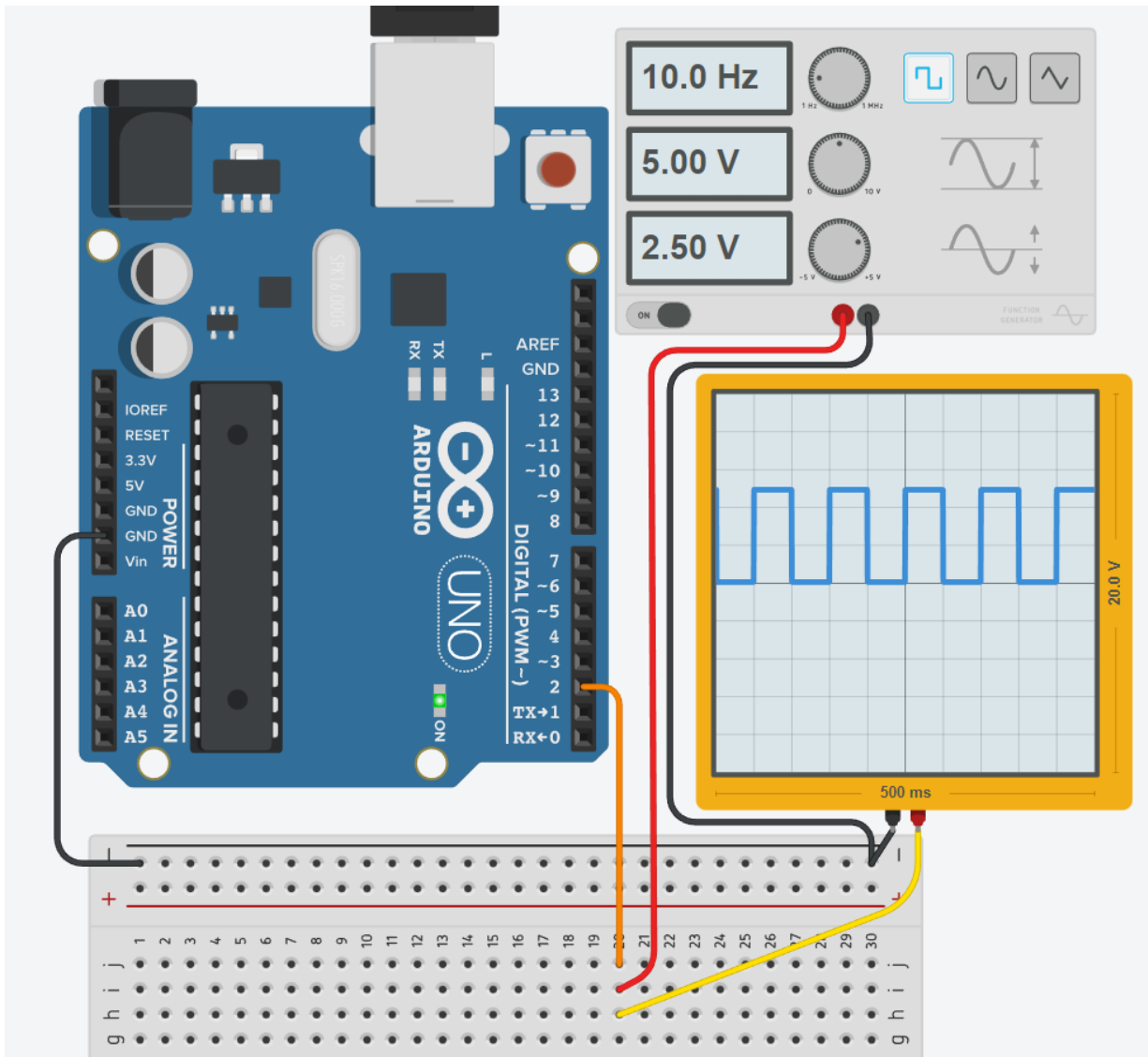
Per poter essere utilizzate correttamente le celle di carico necessitano di una accurata taratura iniziale effettuata con una sollecitazione di intensità nota.

## INTERRUPT E CONTEGGIO IMPULSI DA UN TRASDUTTORE

Un interrupt (*interruzione*) è un evento che viene generato in presenza di una variazione di livello (da 0→5V o 5→0V) su un particolare pin di Arduino (pin 2 e 3 per la scheda Arduino UNO).

Questo evento viene gestito direttamente dal microcontrollore ed è controllabile via software tramite delle apposite istruzioni. Quando viene generata una interruzione è possibile eseguire del codice in modo automatico che interrompe momentaneamente il normale flusso di codice all'interno del blocco loop().

Nello schema di figura il generatore di funzioni d'onda viene utilizzato per simulare un trasduttore (es. encoder ottico) che genera un treno di impulsi con frequenza proporzionale al valore della grandezza fisica misurata.

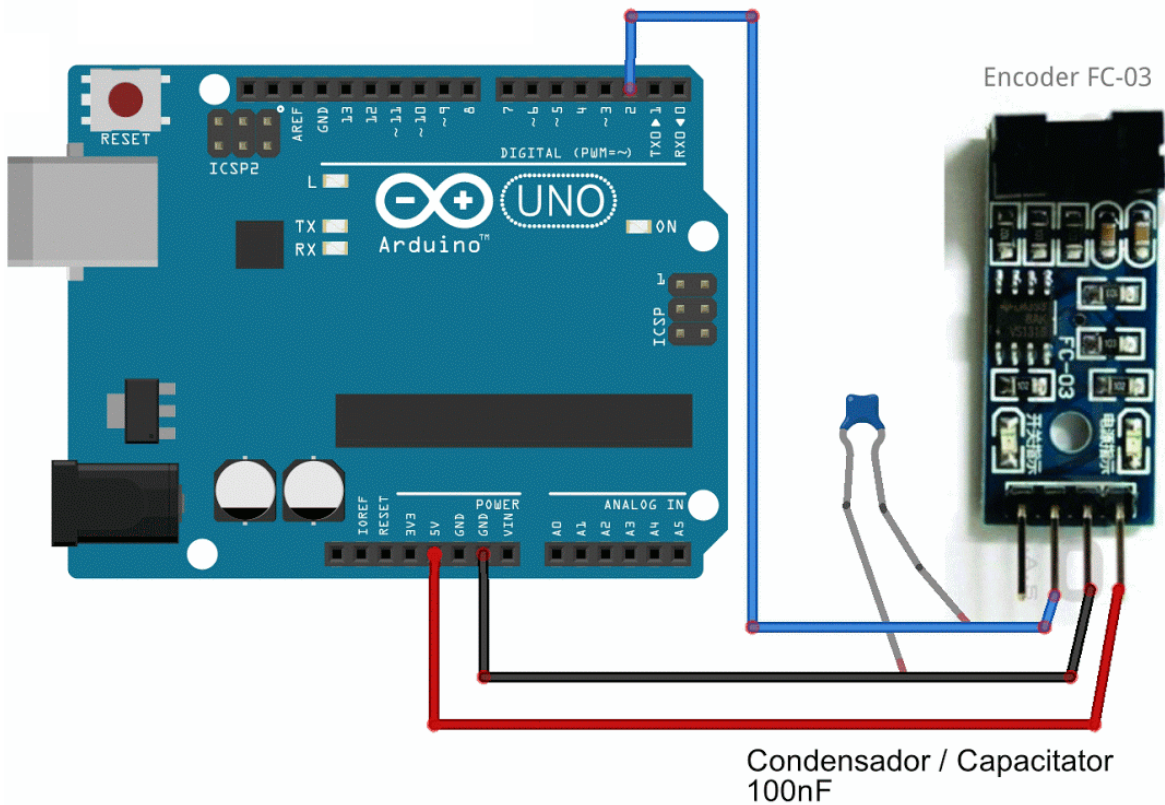


### Variabili di tipo "volatili" (*salvata nella memoria RAM di Arduino*)

Una variabile deve essere dichiarata volatile ogni volta che il suo valore può essere modificato da qualcosa al di fuori del controllo della sezione di codice in cui appare come ad esempio in un thread (processo parallelo) in esecuzione contemporaneamente al codice principale.

In Arduino, l'unico posto in cui è probabile che ciò avvenga è nelle sezioni di codice associate agli interrupt, chiamate routine di servizio di interrupt.

```
volatile int contatore = 0;
```



*Utilizzo shield Arduino FC-03 per contare gli impulsi rilevati da un encoder*

#### **CODICE**

```
volatile int contatore = 0;

void setup() {
  Serial.begin(9600);
  // uso il pin2 per l'interrupt (solo il 2 o il 3 di Arduino sono abilitati agli interrupt)
  attachInterrupt(digitalPinToInterrupt(2), interrupt0, RISING);
}

void loop() {
  delay(1000);
  Serial.print(contatore);
  Serial.println(" impulsi");
  contatore = 0;
}

void interrupt0()
{
  contatore++;
}
```

## ENCODER MAGNETICI

L'encoder è un dispositivo elettromeccanico che converte la posizione angolare meccanica del suo asse rotante in posizione angolare elettrica sotto forma di segnale elettrico numerico digitale e/o analogico.

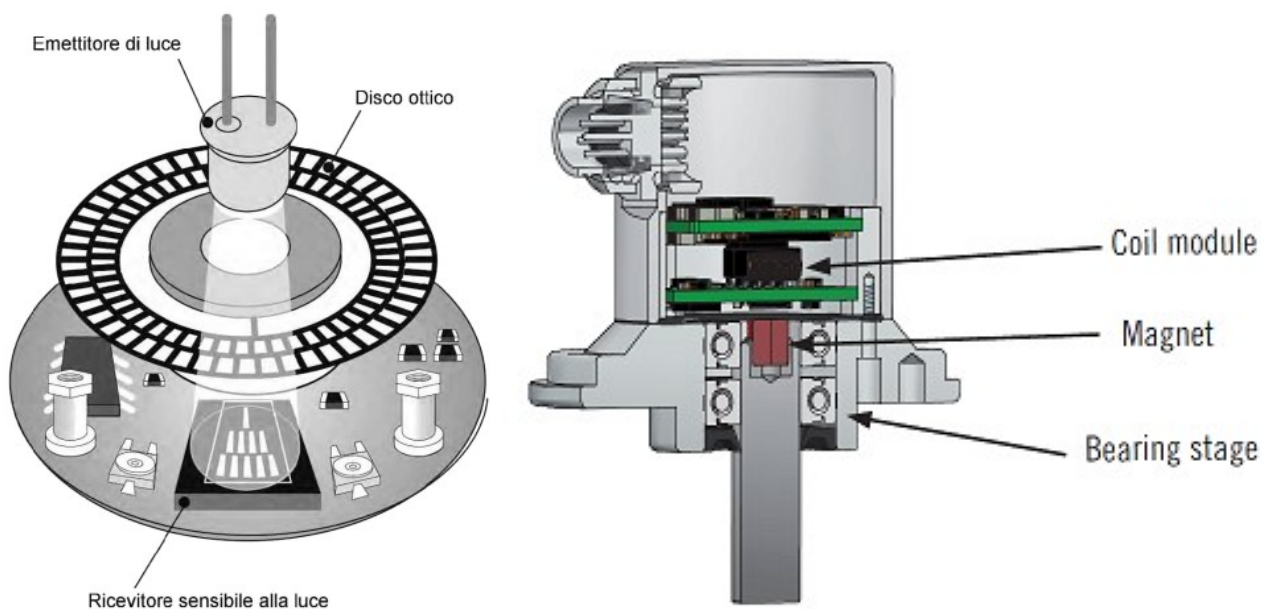
Collegato ad opportuni circuiti elettronici e con appropriate connessioni meccaniche, l'encoder è in grado di misurare spostamenti angolari, movimenti rettilinei e circolari nonché velocità di rotazione e accelerazioni.

Esistono varie tecniche per il rilevamento del movimento angolare: capacitiva, magnetica, induttiva, potenziometrica e fotoelettrica.

Gli encoder si possono classificare nelle seguenti categorie:

- ottici
- a variazione di campo magnetico/elettrico.

Gli encoder vengono principalmente impiegati nei seguenti settori applicativi: controllo dei processi industriali, robot industriali, macchine utensili, strumenti di misura, confezionamento, plotter, laminatoi e macchine per il taglio delle lamiere, bilance e bilici, antenne, telescopi, impianti ecologici, macchine da stampa e da imballaggio, macchine tessili e conciari, gru, carri ponte, presse, macchine per la lavorazione del legno, della carta, del marmo, del cemento, del vetro ecc.

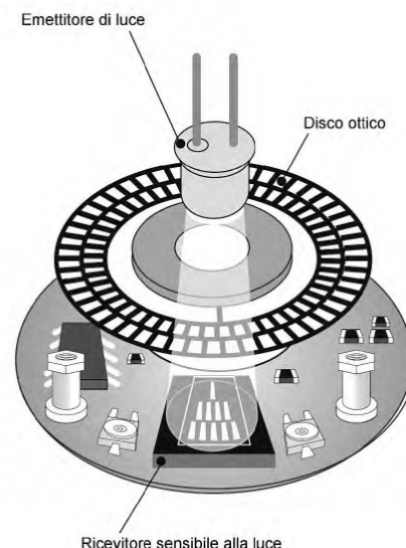


## ENCODER OTTICI

L'encoder è uno strumento per la misura di una posizione angolare basato sul **principio fotoelettrico**

Il principio base di funzionamento è il seguente: davanti a una sorgente luminosa è posto un disco che presenta alternativamente aree opache e trasparenti. Un fotorilevatore rileva l'intensità luminosa che attraversa il disco e fornisce quindi un'informazione indiretta sulla posizione angolare

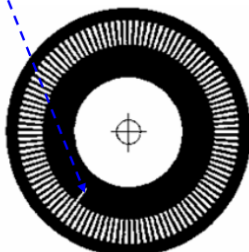
L'encoder può essere **incrementale** o **assoluto**



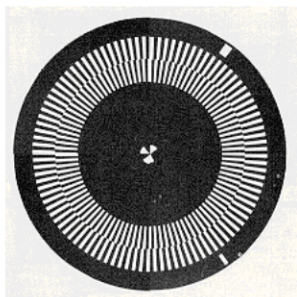
## ENCODER INCREMENTALE

Nella sua versione più semplice l'encoder incrementale è costituito da una sola traccia con zone alternativamente trasparenti o opache

Tacca per la definizione di uno zero meccanico assoluto.



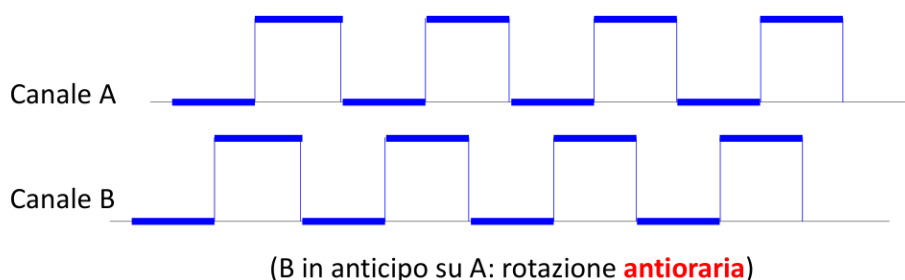
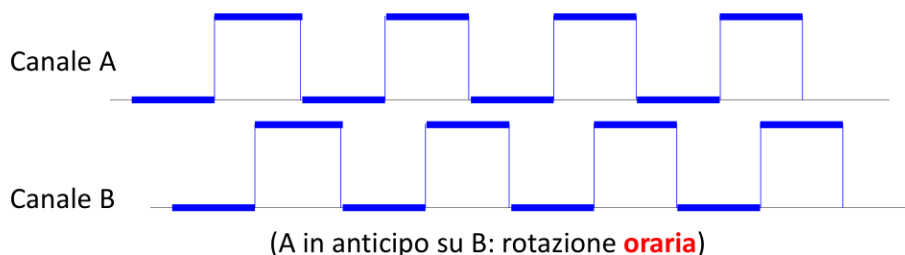
Per stimare anche il **verso di rotazione**, si aggiunge una **seconda traccia**, sfasata di 1/4 di passo.



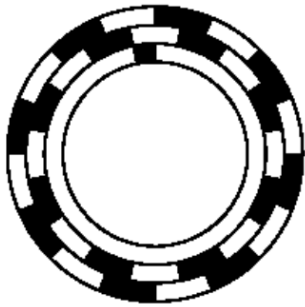
L'uscita del fotorilevatore viene squadrata da un circuito elettronico e dà luogo a un treno di impulsi.



- Non si ha la posizione angolare assoluta ma si possono solo contare gli impulsi dal momento dell'accensione
- Non si può rilevare il verso di rotazione



## ENCODER INCREMENTALE: RISOLUZIONE

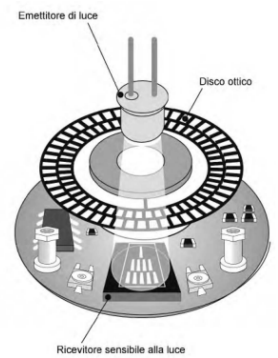


$N$ : numero passi (numero di zone chiare/scure per giro).

Poiché i due segnali sono sfasati di  $1/4$  di passo:

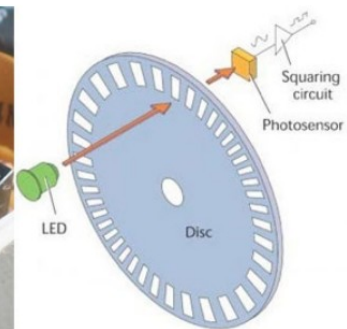
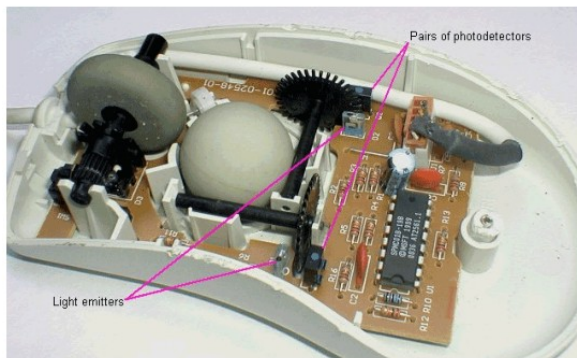
$$\text{Risoluzione: } 360^\circ / (4N)$$

Con  $N = 1000$  la risoluzione è di  $0.090^\circ$



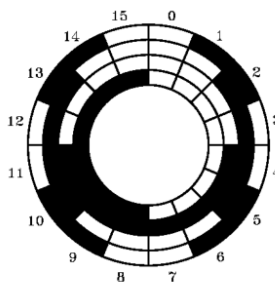
## ENCODER INCREMENTALE: ESEMPIO D'USO

Encoder incrementali su un mouse (a pallina)



## ENCODER ASSOLUTO

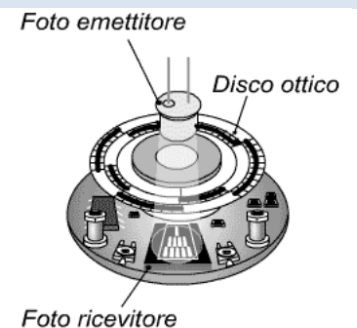
- L'**encoder assoluto** è un disco con aree trasparenti e opache, disposte su corone circolari concentriche
- La traccia più interna suddivide l'angolo giro in due, la seconda traccia suddivide ciascun angolo piatto in due angoli di  $90^\circ$  e così via



Con  $N$  corone circolari si ha quindi:

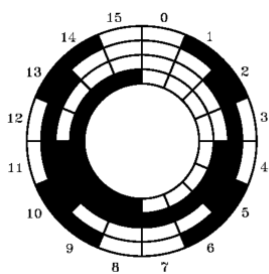
$$\text{Risoluzione: } 360^\circ / 2^N$$

Per le applicazioni robotiche sono richieste almeno 12 tracce (risoluzione di  $360^\circ / 4096 = 0.088^\circ$ ).



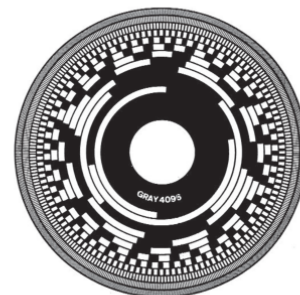
Viene codificata la **posizione angolare assoluta**: all'accensione, lo strumento fornisce l'angolo assoluto

Per evitare ambiguità di lettura si utilizzano codici binari a variazione singola (**codice Gray**): da un settore a quello successivo cambia una sola cifra.

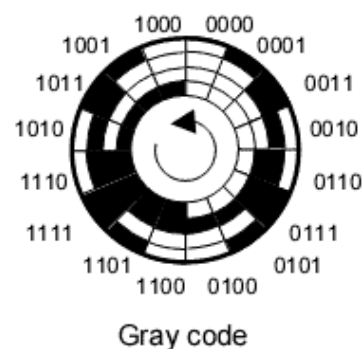


#	Codice	#	Codice
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

Encoder a 12 bit con codifica Gray



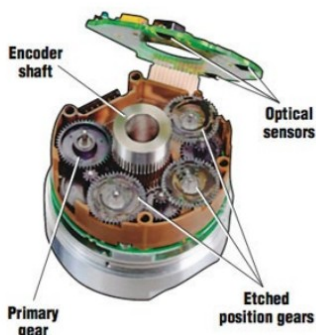
Numero decimale	Numero in binario puro	Numero in Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000



### ENCODER ASSOLUTO: SINGLE-TURN O MULTI-TURN

Gli encoder **single-turn** misurano solo la posizione sull'angolo giro: un encoder single-turn a 13 bit ha una risoluzione sul giro di  $360^\circ/2^{13} = 0.044^\circ$

Gli encoder **multi-turn** misurano anche il numero delle rotazioni, mediante opportuni ingranaggi: un encoder multi-turn a 29 bit può avere 13 bit dedicati alla misura sul giro e gli altri 16 a misurare fino a  $2^{16} = 65.536$  giri



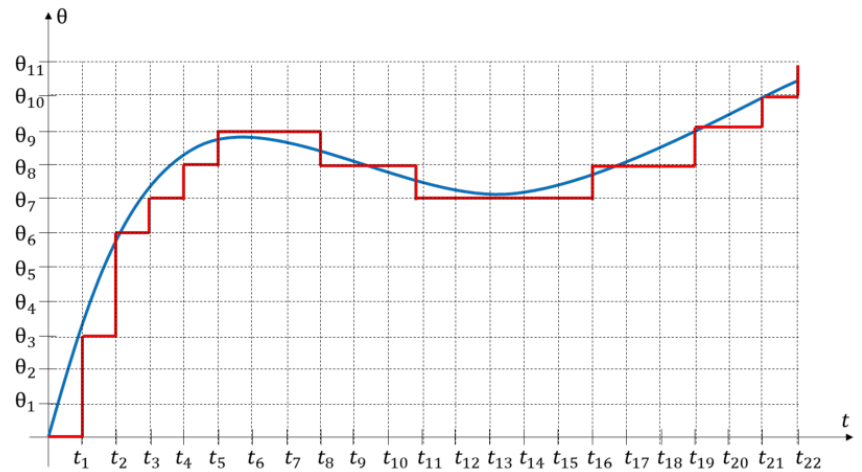
Encoder assoluto Sick

## MISURA DI VELOCITÀ DAL SEGNALE ENCODER

Dagli impulsi prodotti da un encoder (incrementale o assoluto) si può ricavare una **misura di velocità**. Esistono due metodi:

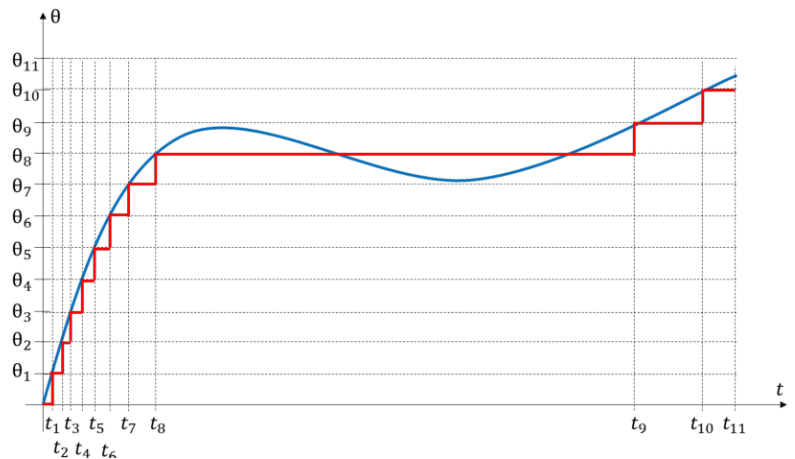
**A tempo fisso:** si campiona l'uscita dell'encoder a passo temporale fissato  $T_s$

$$\omega(k) = \frac{\theta(k) - \theta(k-1)}{T_s}$$



**A spazio fisso:** si misura il tempo necessario per cambiare l'uscita dell'encoder di un bit (variazione  $\Delta\theta$ )

$$\omega(k) = \frac{\Delta\theta}{t_k - t_{k-1}}$$

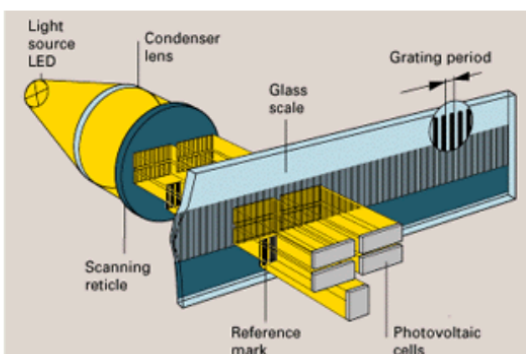


In genere si ottengono risultati migliori con i metodi a spazio fisso, in particolare associando metodi di filtraggio

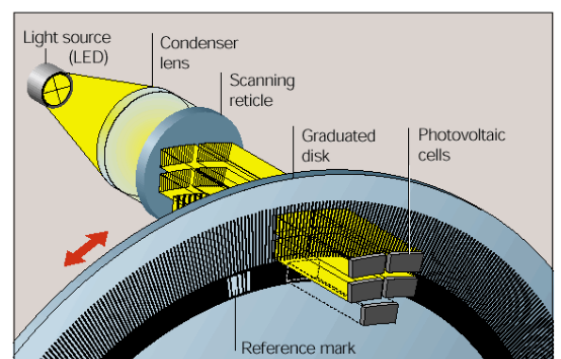
## ENCODER AVANZATI

Per grandi precisioni, si usano sistemi basati sull'**effetto di Moiré**: se si fanno scorrere parallelamente e uno sull'altro due reticoli di divisione, si rilevano oscillazioni periodiche di luminosità, che si possono convertire in segnali elettrici.

### Riga ottica



### Encoder rotativo







# ATTUATORI

In ingegneria, gli attuatori sono capaci di trasformare un segnale di input (normalmente elettrico) in movimento, come esempi di attuatori sono i motori elettrici, pistoni idraulici, relè, polimeri elettroattivi, attuatori piezoelettrici, ecc.

I motori sono usati soprattutto quando si richiedono movimenti circolari, ma possono essere impiegati per applicazioni lineari trasformando un movimento da circolare a lineare utilizzando un trasduttore a vite senza fine. D'altra parte, alcuni attuatori, come quelli piezoelettrici, sono intrinsecamente lineari.



Attuatori lineari ibridi

## MOTORE IN CORRENTE CONTINUA (C.C.)

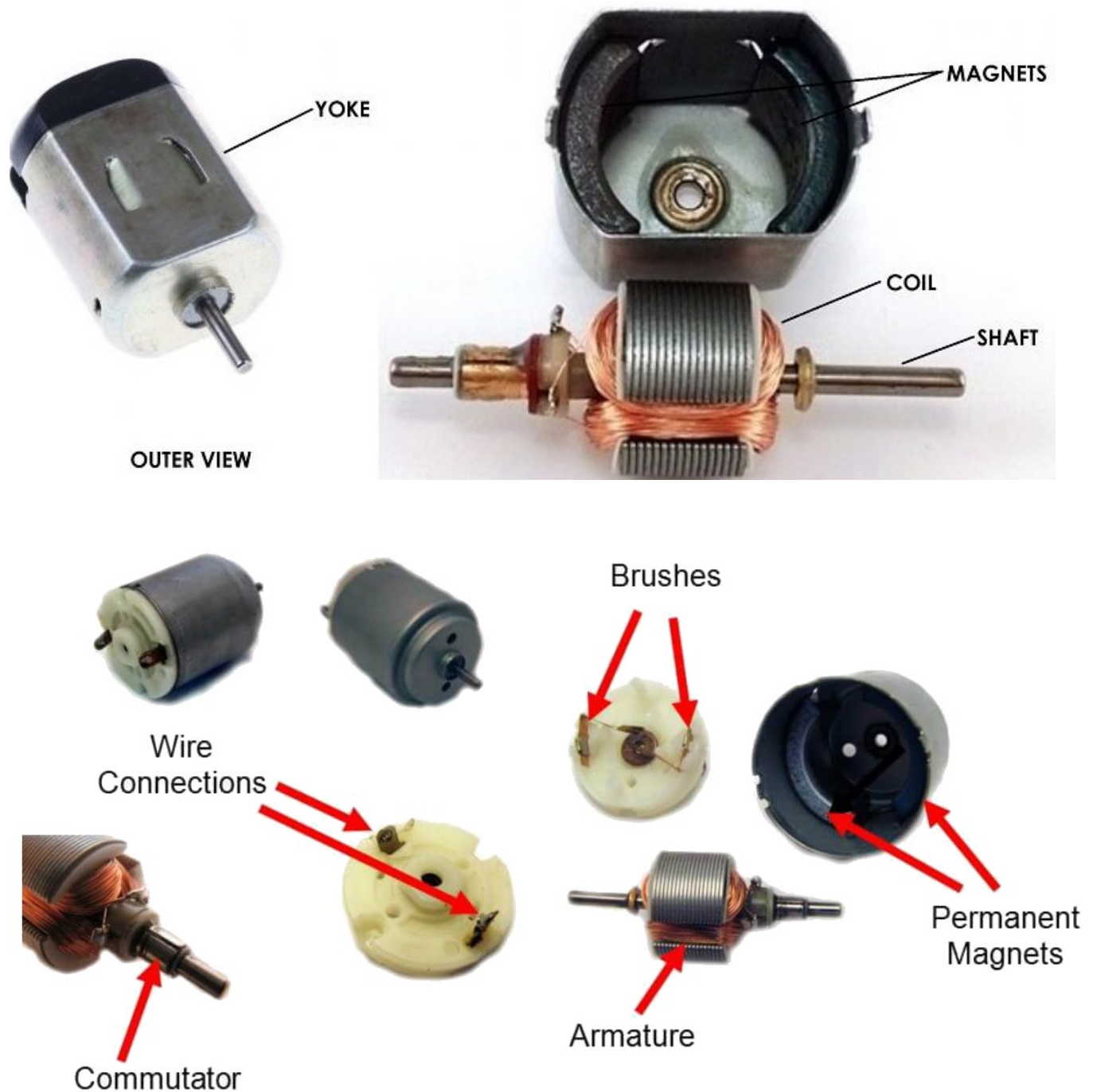
Un motore in corrente continua (CC) è una macchina elettrica che converte l'energia elettrica ( $V, I$ ) in energia meccanica (coppia motrice,  $n^\circ$  giri).

Il motore DC funziona grazie all'interazione tra un campo magnetico generato dallo statore e una corrente che attraversa gli avvolgimenti del rotore. Questa interazione produce una coppia che mette in rotazione l'albero del motore.

Il collettore (commutatore) e le spazzole permettono di mantenere la rotazione continua nel medesimo verso.

Applicando la massima tensione per cui il motore è stato progettato si otterrà la massima velocità.

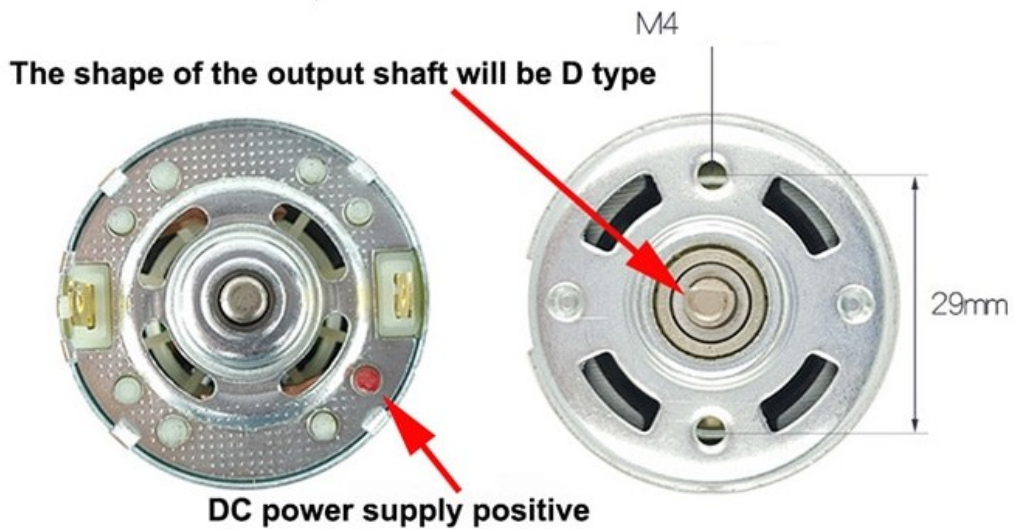
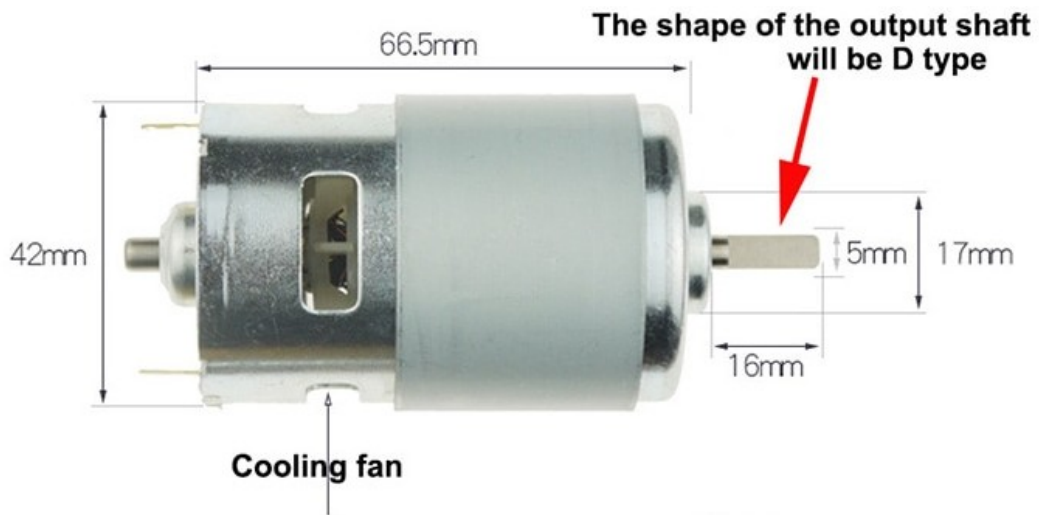
Diminuendo la tensione applicata il numero di giri calerà di conseguenza (in generale insieme alla coppia motrice).





<https://www.youtube.com/watch?v=peGZkxushel>

775 D SHAFT



**Statore (Magneti):** Parte fissa del motore che genera il campo magnetico necessario.

**Rotore (Armatura):** Parte rotante costituita da avvolgimenti di rame su un nucleo ferromagnetico.

**Albero:** Asse collegato al rotore che trasmette il movimento meccanico all'esterno.

**Cuscinetti:** Supporti che permettono la rotazione fluida dell'albero riducendo l'attrito.

**Carcassa:** Struttura esterna che protegge il motore e contribuisce alla dissipazione del calore.

**Collettore (Commutatore):** Cilindro a lamelle di rame collegato agli avvolgimenti, permette di invertire la corrente nei giusti momenti.

**Spazzole:** Contatti striscianti in grafite che trasferiscono corrente al rotore tramite il commutatore.

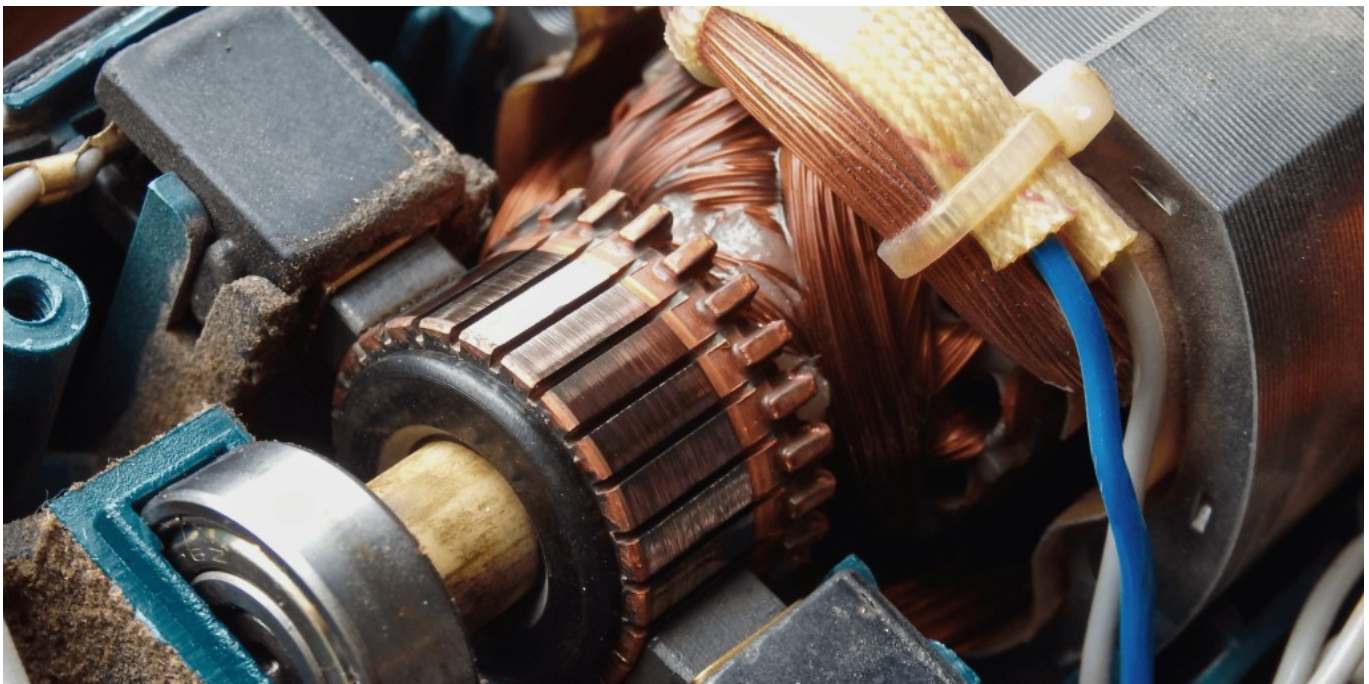
### L'importanza delle spazzole

Il commutatore (o collettore) è un cilindro formato da lamelle di rame montato sull'albero del rotore.

Le spazzole sono blocchetti di grafite premuti contro la superficie del commutatore.

Quando il rotore gira:

- le spazzole restano ferme (e nel tempo si usurano e andranno sostituite)
- il commutatore solidale all'asse gira sotto di esse
- il contatto tra i due porta la corrente a vari avvolgimenti del rotore in modo da farlo ruotare .

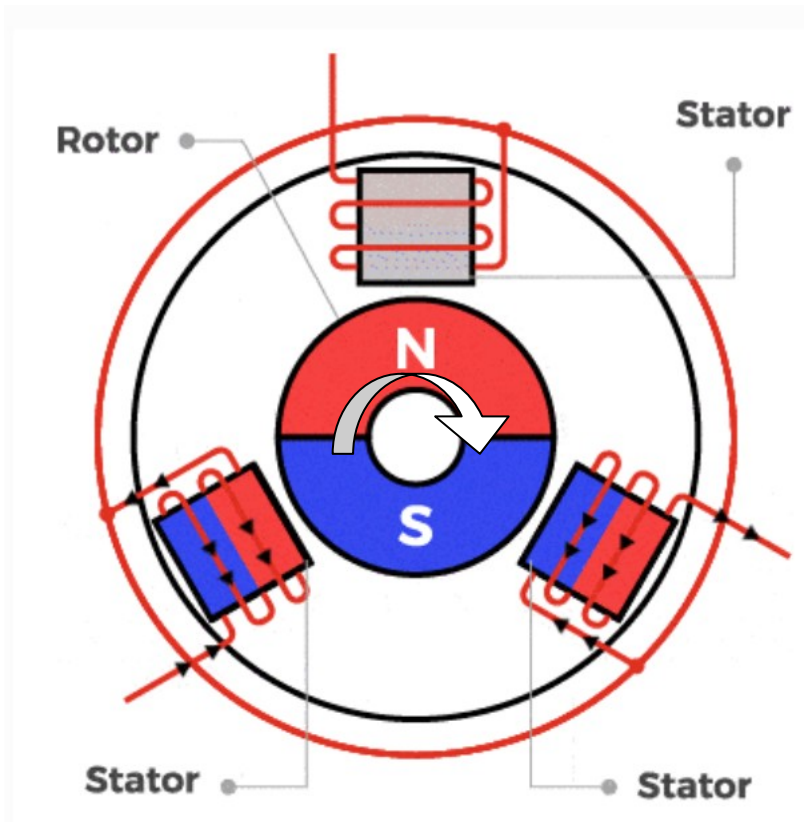


### Spazzole in carbone

Le spazzole in carbone sono le più utilizzate per le loro proprietà autolubrificanti, conduttività elettrica e resistenza alle alte temperature. Queste spazzole sono in grado di resistere all'elevato calore prodotto nei motori, il che le rende utili per applicazioni ad alta velocità o con carico elevato. Il loro principale svantaggio è che possono produrre più rumore e scintille rispetto alle altre tipologie.

## MOTORI CC SENZA SPAZZOLE BLDC (BRUSHLESS )

Nei motori brushless al posto delle spazzole si impiega una diversa forma di commutazione basata sull'elettronica (controller elettronico ESC) che permette di commutare il campo magnetico che interagisce con quello del rotore costituito da magneti permanenti. Gli avvolgimenti vengono posizionati sullo statore e non sullo stator come nel motore a spazzole.



### PARTI FONDAMENTALI DEL MOTORE BLDC

**Statore:** è la parte fissa con avvolgimenti a bobina; la figura presenta tre avvolgimenti statorici collegati a stella, distribuiti lungo la circonferenza dello statore per formare poli magnetici distribuiti uniformemente.

**Rotore:** è costituito da magneti permanenti, in genere si va da 2 a 8 poli magnetici, alternati tra poli nord e sud.

### FUNZIONAMENTO DEL MOTORE BRUSHLESS

Se al motore viene applicata una corrente continua fissa, questo genererà solo un campo magnetico costante e non potrà ruotare.

Sequenziando opportunamente l'alimentazione delle fasi dello statore (bobine), si genera un campo magnetico rotante.

I poli magnetici del rotore seguono il campo magnetico rotante dello statore in modo ordinato, ottenendo così la rotazione dell'albero motore.

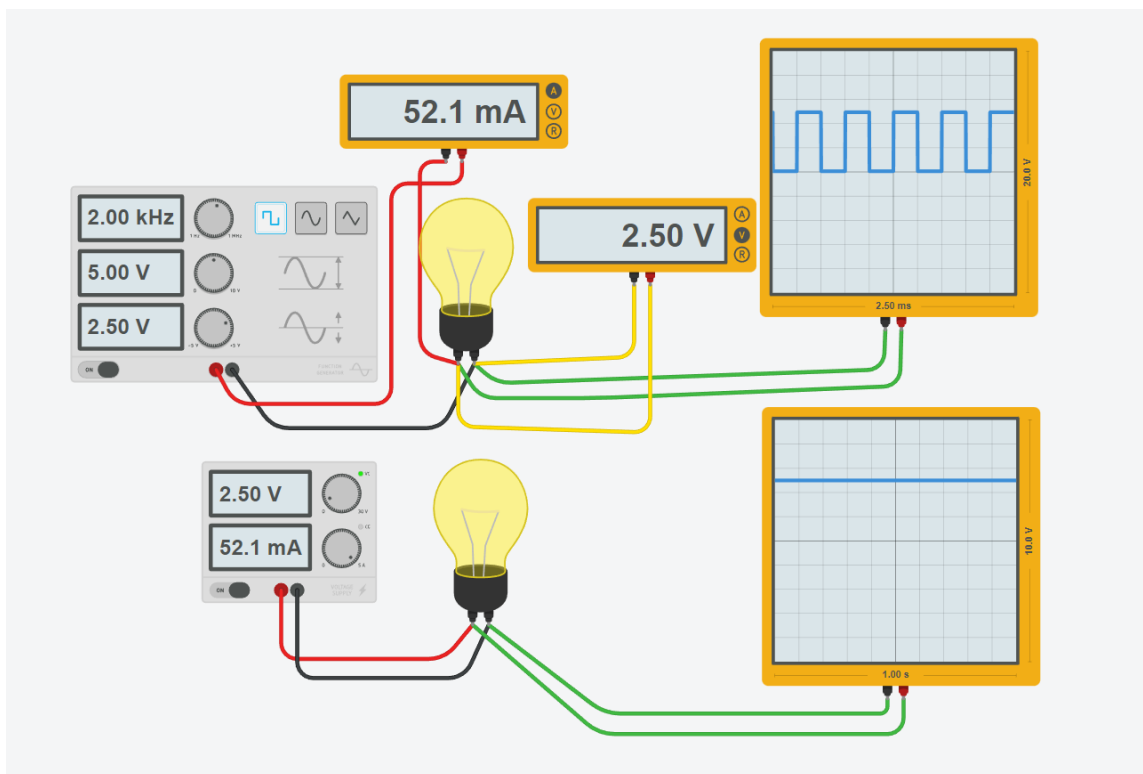
## PWM (PULSE WIDE MODULATION): MODULAZIONE DI LARGHEZZA D'IMPULSO

Un microcontrollore come Arduino non è in grado di generare un segnale analogico di tensione.

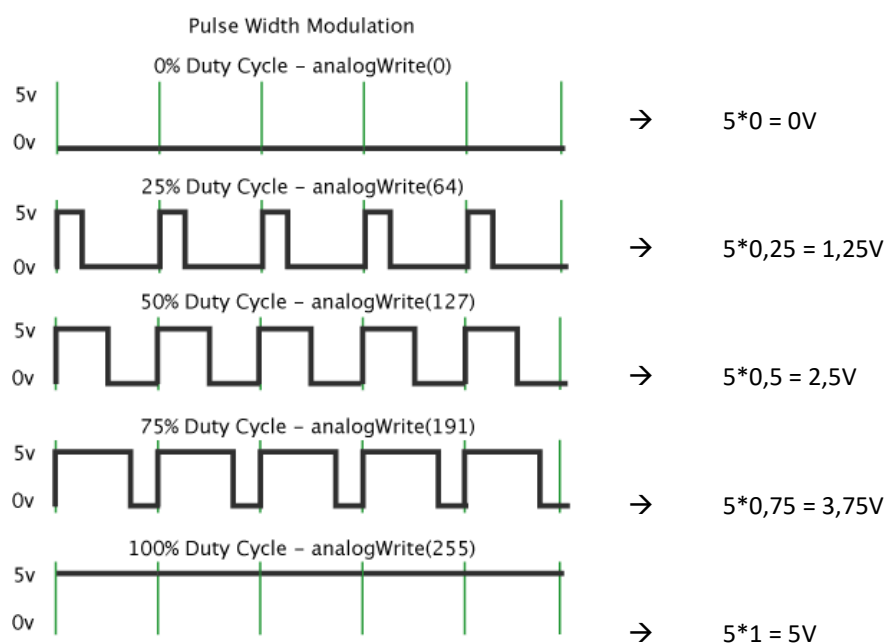
Tuttavia utilizzando la PWM è in grado di generare un'onda quadra ad **alta frequenza** modulata in ampiezza che viene percepita dalla maggior parte degli utilizzatori (resistenze, lampadine, motori CC) come una tensione continua.

Il circuito sottostante mostra l'effetto di una tensione periodica a 2kHz a onda quadra di ampiezza 5V e duty cycle del 50% (frazione di tempo in cui l'onda è allo stato attivo in proporzione al periodo totale).

Si nota, dalla tensione media e dalla corrente assorbita, che l'effetto sulla lampada è lo stesso generato dall'alimentazione a CC a 2.5V.



Con Arduino si può generare un segnale PWM a 8 bit (255 combinazioni) da 0 a 5V (risoluzione =  $5/255=0.196V$ ).



La regolazione della velocità del motore DC può ottenuta variando linearmente la tensione di alimentazione del motore. Si fornisce così una corrente continua e costante. La coppia motrice del motore (caso ideale) è data dalla relazione:

$$C_m = K_t * I \text{ (Nm)}$$

dove  $k_t$ =costante di coppia del motore e  $I$  è la corrente che attraversa gli avvolgimenti del motore.

Si evince che riducendo la tensione, cala proporzionalmente la corrente ( $I=V/R$ ) e quindi cala anche la coppia motrice  $C_m$ .

Con la tecnica PWM invece vengono forniti al motore degli impulsi di corrente il cui valore medio (se la frequenza degli impulsi è sufficientemente alta rispetto all'induttanza del motore) è circa pari a quella massima.

Ciò consente di mantenere alta la  $C_m$  del motore anche ad un regime di rotazione inferiore rispetto a quello nominale.

## PROBLEMI NELLA REGOLAZIONE PWM DI UN MOTORE DC

Se utilizzando la PWM con un motore DC si nota una riduzione della  $C_m$  allora le ragioni possono essere le seguenti:

### 1-Assenza del Diodo di Ricircolo (Flyback Diode)

Questa è la causa più comune. Un motore CC è un carico induttivo. Quando il MOSFET "stacca" (fase OFF del PWM), l'energia immagazzinata nel campo magnetico del motore deve andare da qualche parte.

**Senza diodo:** La corrente si interrompe bruscamente. L'energia viene dissipata sotto forma di picchi di alta tensione (che possono danneggiare il MOSFET) e la corrente media crolla.

**Con diodo:** Durante la fase OFF, il diodo permette alla corrente di continuare a circolare nel motore (fase di "freewheeling"). Questo mantiene la corrente più costante e, dato che la coppia è proporzionale alla corrente, la forza del motore rimane alta.

### 2- MOSFET non completamente "aperto" (Saturazione)

Arduino lavora a 5V (o 3.3V). Molti MOSFET di potenza comuni (come l'IRFZ44N) non sono "Logic Level": hanno bisogno di 10V sulla Gate per aprirsi completamente.

Se si usa un MOSFET standard con i 5V di Arduino, il componente oppone una resistenza elevata  $R_{DS(on)}$ . Invece di comportarsi come un interruttore perfetto, si comporta come una resistenza, scaldando molto e facendo cadere la tensione destinata al motore.

Soluzione: Usare un MOSFET Logic Level (es. serie IRL, come l'IRLZ44N).

### 3-Caduta di tensione dell'alimentatore

I motori della serie 775 hanno correnti di spunto (stallo) molto elevate, spesso superiori ai 10-20A.

Quando il PWM è al 50%, chiedi all'alimentatore di fornire la massima corrente in impulsi brevissimi.

Se l'alimentatore non ha condensatori di filtro adeguati o non è abbastanza potente, la tensione "siede" (crolla) durante l'impulso ON, riducendo la forza effettiva rispetto a una tensione continua stabilizzata.

#### 4-Frequenza del PWM troppo bassa

Quando la frequenza è molto bassa (ad esempio sotto i 50-100 Hz), i cicli di ON e OFF sono così lunghi che il motore ha il tempo di accorgersene.

Durante la fase di ON, il motore accelera bruscamente; durante la fase di OFF, inizia a rallentare per l'attrito e il carico.

Risultato sulla coppia: non abbiamo una spinta fluida, ma una serie di impulsi.

Questo genera il cosiddetto Torque Ripple (ondulazione della coppia).

Si sente il motore vibrare o "grattare" e si perde efficienza perché molta energia viene sprecata in vibrazioni meccaniche e calore anziché in rotazione.

#### 5- Frequenza del PWM troppo alta

Qui entra in gioco la fisica del motore. Un motore 775 ad esempio è essenzialmente una grossa bobina di filo (un induttore).

Gli induttori si oppongono alle variazioni rapide di corrente. La corrente in un induttore non arriva istantaneamente al massimo, ma segue una curva di crescita determinata dalla formula della costante di tempo:

$\tau = L/R$  dove L è l'induttanza e R è la resistenza del motore.

Se la frequenza PWM è troppo alta, il tempo in cui il MOSFET resta "acceso" è più breve del tempo necessario alla corrente per salire a un livello utile.

Conseguenza sulla coppia:

La corrente viene tagliata quando è ancora molto bassa. Il motore non riceve mai abbastanza corrente per generare la coppia nominale.

Per un motore DC 775, la frequenza ideale deve essere un compromesso:

- Sopra i 20 Hz per evitare vibrazioni meccaniche macroscopiche.
- Sopra i 18-20 kHz se vuoi eliminare il fischio udibile dall'orecchio umano (ma qui serve un driver MOSFET serio, non un optoisolatore economico).
- Tra 500 Hz e 4 kHz è solitamente il "punto dolce" per i moduli tipo il tuo: abbastanza veloce da rendere il movimento fluido, abbastanza lenta da permettere all'optoisolatore di commutare bene e alla corrente di salire negli avvolgimenti.

#### 6- Moduli mosfet optoisolati o troppo lenti

Se si utilizzano dei moduli mosfet è necessario verificare il tempo di risposta del modulo. Con un modulo YNMOS a 4 mosfet ad esempio non si riesce a regolare correttamente la velocità di un motore poiché il modulo è dotato di optoisolatori (sistemi di protezione per separare elettricamente Arduino dal circuito di potenza) che riducono il tempo di risposta dei mosfet.

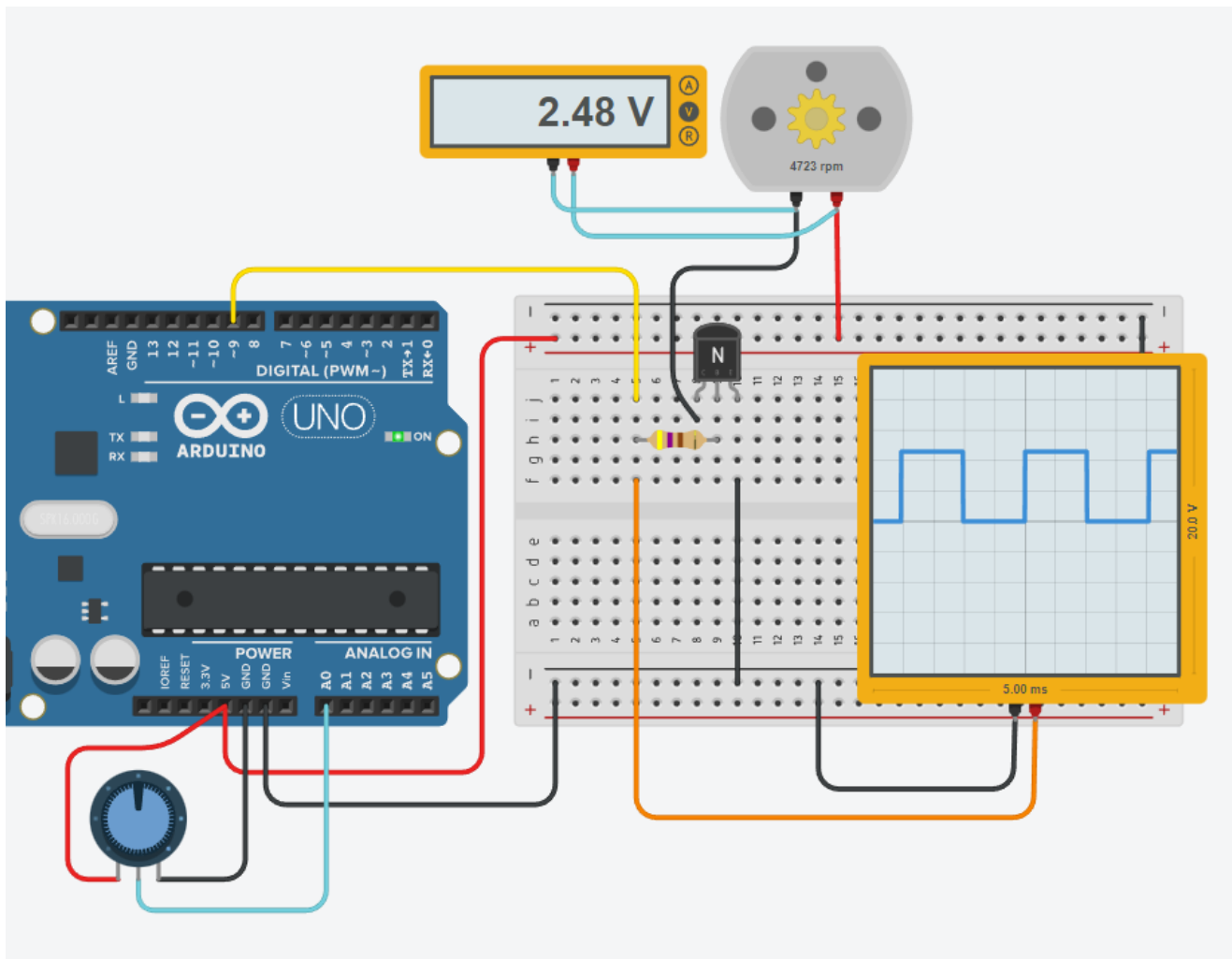
Il segnale che arriva ai transistor al posto di un'onda quadra perfetta diventa "smussata" o trapezoidale e il MOSFET rimane bloccato in una "via di mezzo" (regione lineare).

Il MOSFET passa molto più tempo nella **zona lineare** (dove si comporta come una resistenza) invece di essere o tutto aperto o tutto chiuso. Questo sottrae energia al motore e la trasforma in calore sul modulo.

## ESERCIZIO PWM MOTORE CC

Regolare la velocità di rotazione del motore mediante un potenziometro e la tecnica PWM.

*NB: ai capi di un motore va sempre messo un diodo di protezione del transistor di comando (anodo sul +) che è stato omesso per semplificare lo schema.*



Il motore DC non può essere alimentato direttamente da un PIN di Arduino poiché la corrente richiesta dal motore è superiore a quella fornita da un PIN. Se la corrente richiesta dal motore è di poche centinaia di mA si può usare l'uscita 5V di Arduino. Se si usa un alimentatore dedicato è necessario mettere la massa in comune con quella di Arduino per garantire il corretto funzionamento (serve lo stesso riferimento per la massa).

La regolazione della velocità del motore DC si effettua in 2 modi:

- regolando la tensione a capi del motore (ad esempio con un potenziometro)
- regolando il tempo (PWM) in cui la tensione massima di alimentazione del motore viene applicata ai suoi capi

**Il 2° metodo permette di regolare la velocità mantenendo anche la coppia motrice sempre elevata mentre nel 1° modo la coppia cala proporzionalmente alla tensione applicata.**

Per fornire una corrente sufficiente al motore è necessario utilizzare un amplificatore (transistor) comandato in PWM da un PIN di Arduino.

L'oscilloscopio permette di visualizzare il segnale di regolazione PWM (0-5V) generato da un PIN di Arduino.

## **CODICE**

```
int motorPin = 9; // PWM
int potPin = A0; // POTENTIOMETER
int potValue = 0;

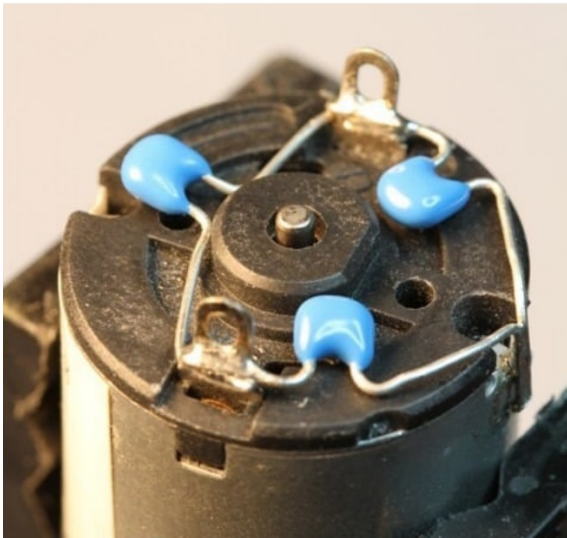
void setup()
{
  pinMode(potPin, INPUT);
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  // read the value from the sensor
  if (analogRead(potPin) != potValue)
  {
    potValue = analogRead(potPin);
    analogWrite(motorPin, potValue/4);
    Serial.println(potValue);
  }
  delay(20); // Wait for 20ms
}
```

## **DISTURBI ELETTROMAGNETICI NEI MOTORI CC A SPAZZOLE**

Molti piccoli motori a corrente continua presentano un forte "rumore di spazzola". Questo si ripercuote sui circuiti di Arduino e ne causa un funzionamento instabile. Questo problema può essere risolto saldando al motore alcuni condensatori ceramici antirumore da 0,1  $\mu$ F.

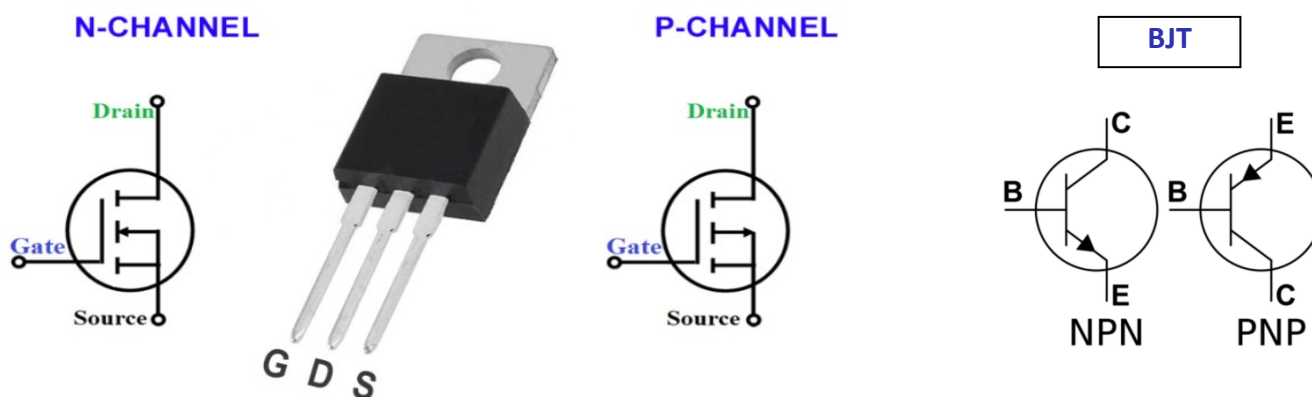
Ne serviranno 3 in totale: 1 tra i terminali del motore e 1 da ciascun terminale alla carcassa del motore.



## CONTROLLO VELOCITÀ MOTORE CC IN PWM CON MOSFET

Il MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) è un tipo di transistor a effetto di campo ampiamente utilizzato nei circuiti elettronici. Funziona come un elemento controllato in tensione e viene utilizzato per scopi di commutazione o amplificazione del segnale.

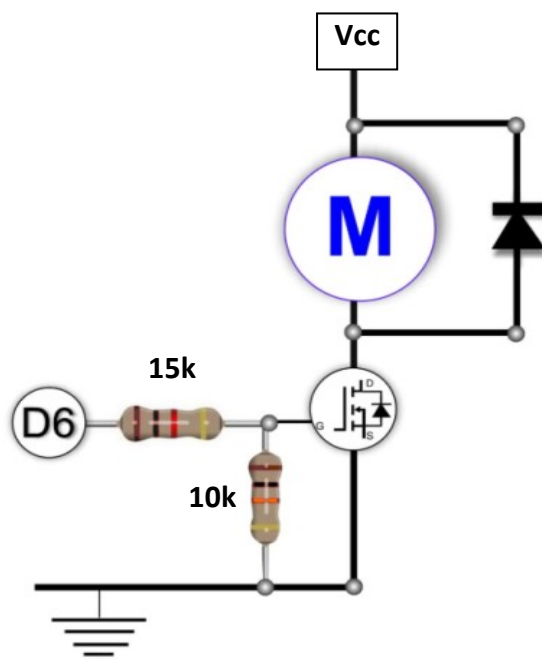
Il MOSFET è uno dei componenti fondamentali dell'elettronica moderna. In generale è più efficiente e veloce rispetto ai BJT.



Caratteristiche principali:

- È controllato dalla tensione applicata al GATE (i BJT sono controllati dalla corrente di base  $I_b$ ).
- Ha un'elevata impedenza di ingresso (dissipano pochissima potenza).
- Hanno velocità di commutazione elevata (spesso utilizzata nei circuiti digitali).
- Quando una tensione viene applicata al GATE, la corrente può scorrere tra il DRAIN e il SOURCE.

La figura sottostante mostra il tipico collegamento equivalente alla configurazione ad emettitore comune del BJT NPN.



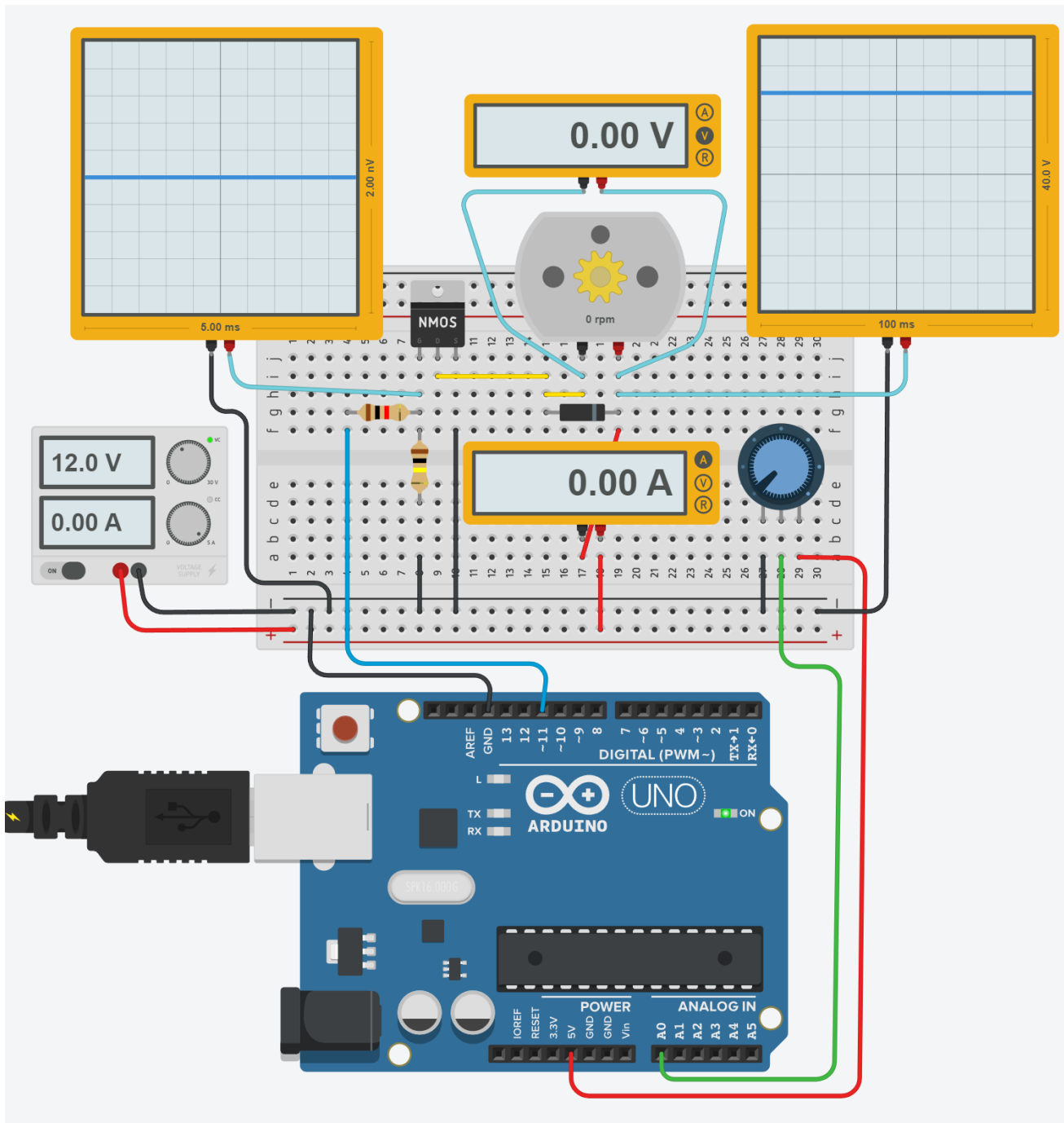
Il Mosfet **IR520** è un tipico Mosfet a canale N che può pilotare un carico da 10 A alla tensione di 100 Volt.

Il GATE è disaccoppiato con un condensatore all'interno del Mosfet e può essere collegato direttamente all'uscita di un micro.

Per sicurezza comunque conviene interporre una resistenza da 15 k per proteggere l'uscita logica di Arduino.

La resistenza da 10 k il GATE del transistor e la MASSA, stabilizza il controllo della base del transistor contro i rumori elettromagnetici.

Il circuito di figura mostra come regolare la velocità di un motore CC in PWM tramite un transistor mosfet. In ingresso al motore è collegato un AMPEROMETRO per visualizzare la corrente che scorre nel DRAIN mentre ai capi del motore è collegato un voltmetro per visualizzare la tensione ai capi del motore. Quando al GATE vengono applicati 0V (massa) il motore è fermo poiché il mosfet si comporta come un interruttore aperto.

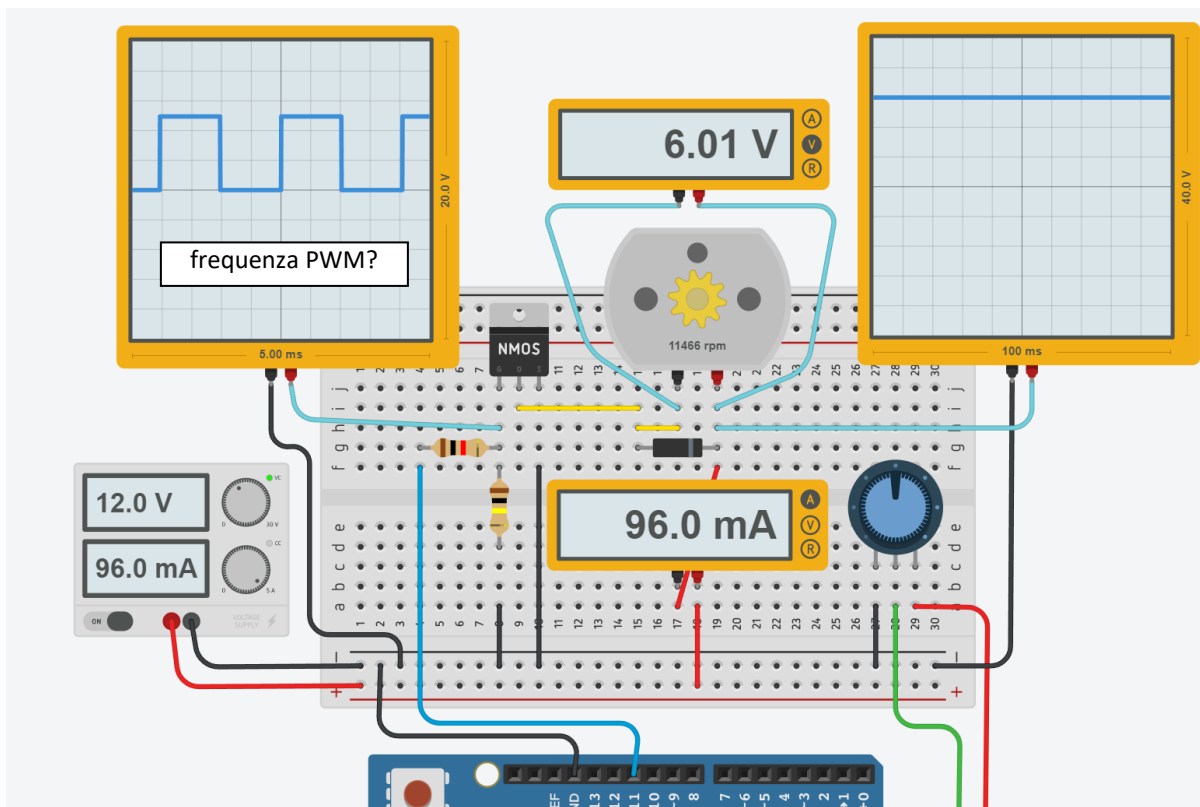


```
const int MOTOR_PIN = 11;
const int POT_PIN = A0;

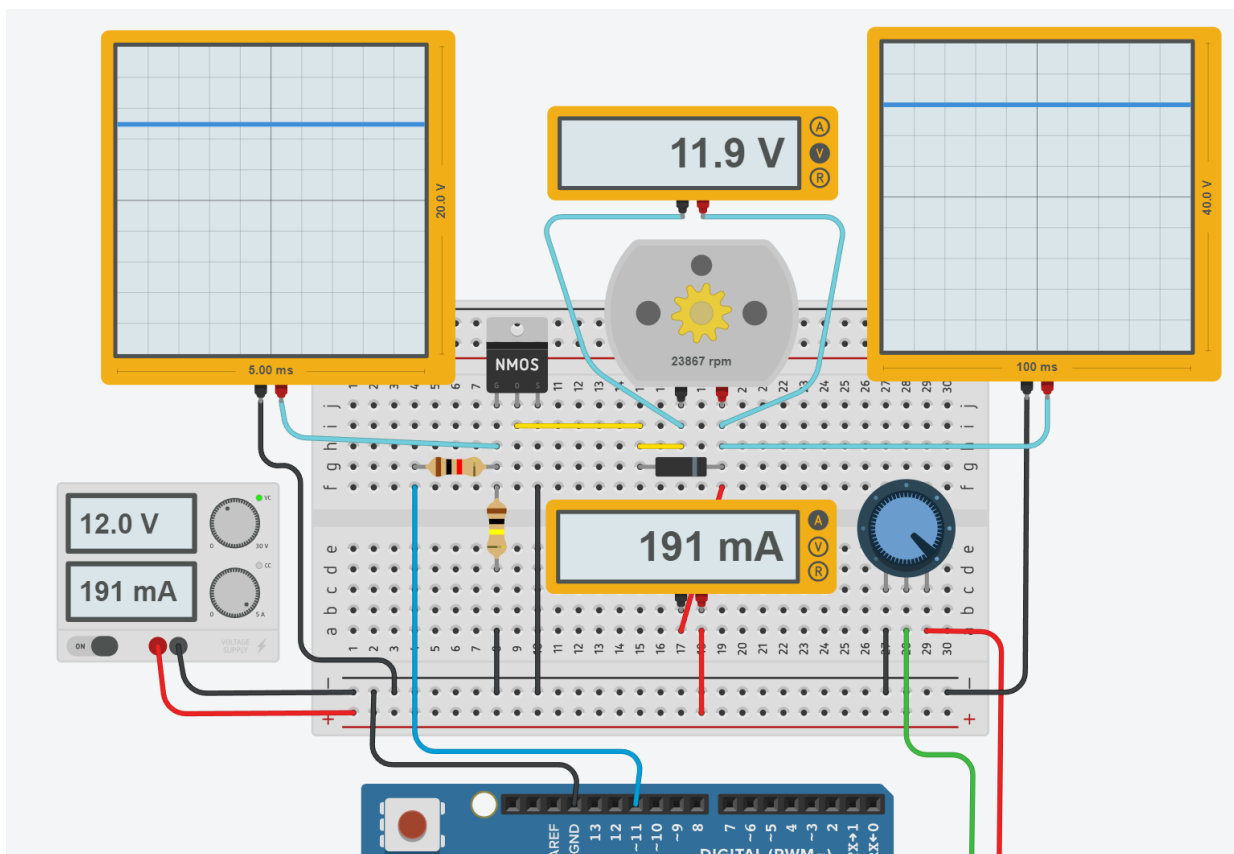
void setup()
{
  pinMode(MOTOR_PIN, OUTPUT);
}

void loop()
{
  int potVal = analogRead(POT_PIN);
  int speedVal = map(potVal, 0, 1023, 0, 255);
  analogWrite(MOTOR_PIN, speedVal);
}
```

Applicando un segnale PWM al 50% al motore arrivano circa 6V con un corrente assorbita di circa 95mA.

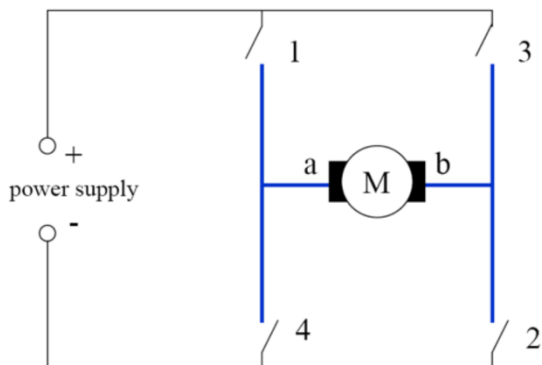


Applicando un segnale PWM al 100% al motore arrivano circa 11.9V con un corrente assorbita di circa 190mA. La prima cosa da osservare è che quasi tutta la tensione di alimentazione  $V_{cc}=12$  arriva la motore! Il mosfet è molto più efficiente di un transistor BJT che presenta sempre una  $V_{cs}$  non trascurabile.



## INVERSIONE VERSO DI ROTAZIONE MOTORE C.C. CON 2 RELE'

Per regolare il verso di rotazione di un motore CC sono necessari due relè opportunamente collegati al motore. Questo sistema NON consente anche la regolazione del numero di giri del motore. Per ottenere il duplice effetto (verso e regolazione velocità) è necessario un **ponte ad H** costituito da 4 transistor.

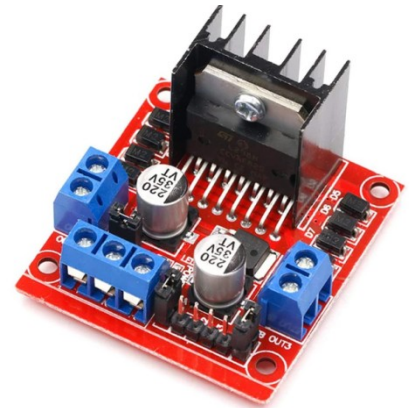


Drive forward:

- Close 1 and 2

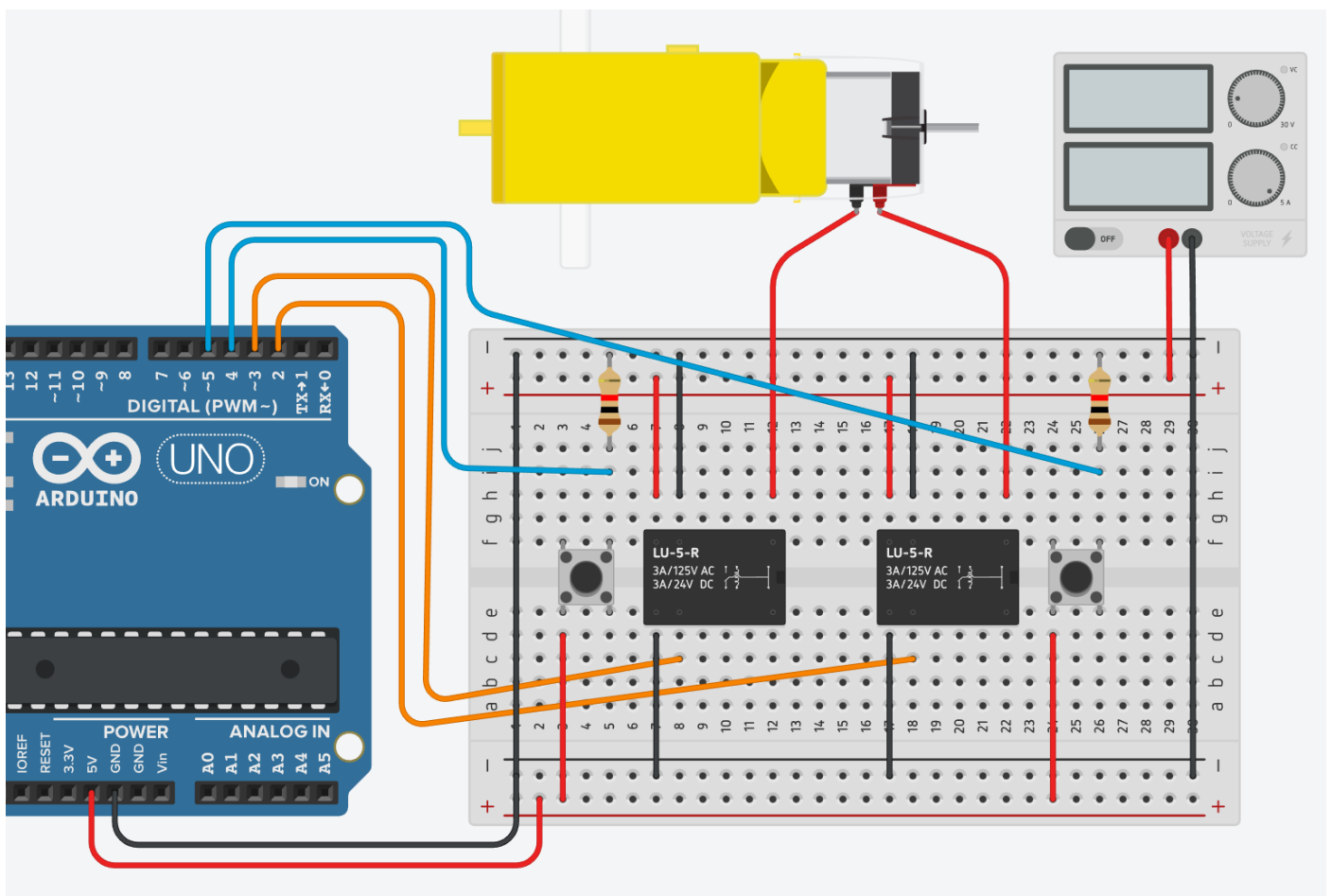
Drive backward:

- Close 3 and 4



## ESERCIZIO VERSO ROTAZIONE MOTORE CON RELE'

Controllare il verso di rotazione del motore C.C. con due pulsanti utilizzando 2 relè.

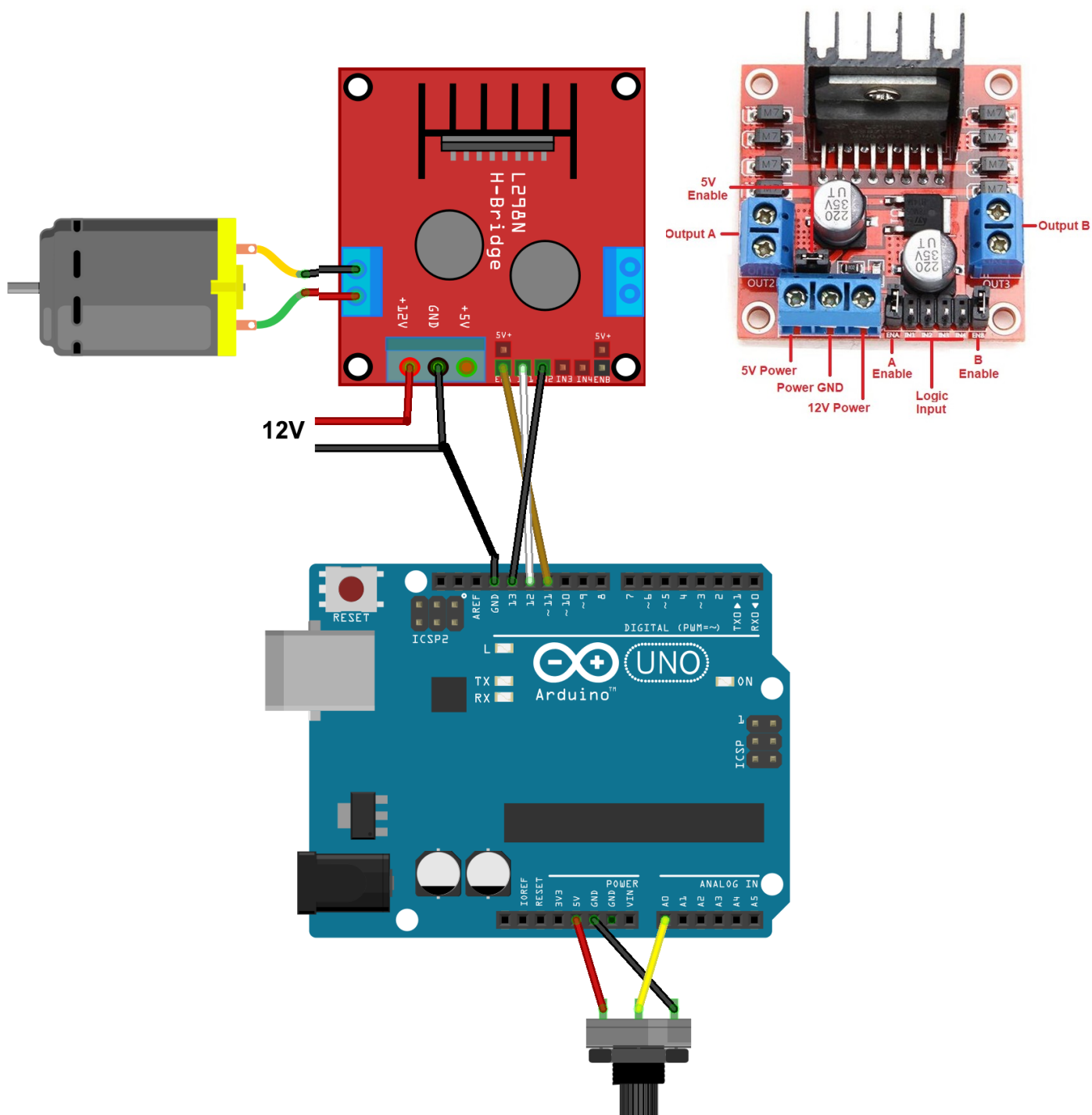


## *CODICE*

```
int incomingByte = 0; // for incoming serial data
```

```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  
  pinMode(4, INPUT);  
  pinMode(5, INPUT);  
  
  Serial.begin(9600);  
}  
  
void loop() {  
  
  int statoP1= digitalRead(4);  
  if (statoP1== HIGH) {  
    Serial.println("M1 ORARIO");  
    digitalWrite(2, HIGH);  
    digitalWrite(3, LOW);  
  }  
  else  
  {  
    Serial.println("M1 STOP");  
    digitalWrite(2, LOW);  
  }  
  
  int statoP2= digitalRead(5);  
  if (statoP2== HIGH) {  
    Serial.println("M1 ANTIORARIO");  
    digitalWrite(2, LOW);  
    digitalWrite(3, HIGH);  
  }  
  else  
  {  
    Serial.println("M1 STOP");  
    digitalWrite(3, LOW);  
  }  
  
  delay(100);  
}
```

Questa scheda di controllo per motori è basata sul driver Dual H-Bridge L298N e permette di pilotare due motori C.C. oppure un motore passo-passo bipolare con tensione operativa compresa nel range tra 5V e 35V e una corrente massima di 2A, controllandone la velocità e la direzione.



### NOTA BENE:

Per motori a bassa resistenza interna (tipica degli stepper) è necessario un driver di corrente e non un driver di tensione come l'L298N. I motori a bassa impedenza in generale vanno controllati in corrente e non in tensione.

Per valori di resistenza degli avvolgimenti del motore oltre 30-60 ohm un L298N funziona senza bruciarsi, ma la velocità massima è inferiore rispetto a quella ottenibile con un driver di corrente.

## CODICE

```
//L298N pilotare un motore DC con Arduino

//definizione dei pin
static int pinPotenziometro = A0; //pin analogico per valori del potenziometro
static int mA = 12; //pin digitale per gli stati logici da inviare al modulo
static int mB = 13; //pin digitale per gli stati logici da inviare al modulo
static int pinMotore = 11; //pin PWM per variare velocità motore

//variabili
int potenziometro; //valore letto dal potenziometro sul pin A0
int velocita; //valore PWM in uscita dal pin 11

void setup() {
  Serial.begin(9600);

  //inizializzo variabili
  potenziometro = 0;
  velocita = 0;

  //definisco tipologia pin
  pinMode(pinPotenziometro, INPUT); //input da potenziometro per la velocità
  pinMode(mA, OUTPUT); //output per lo stato logico del pin IN1 del modulo L298N
  pinMode(mB, OUTPUT); //output per lo stato logico del pin IN2 del modulo L298N
  pinMode(pinMotore, OUTPUT); //output PWM per il pin EN1 del modulo L298N

  //Imposto verso di rotazione del motore
  /*
    mA | mB | Evento
    ----|-----|-----
    LOW | LOW | fermo
    LOW | HIGH | rotazione oraria
    HIGH | LOW | rotazione antioraria
    HIGH | HIGH | Fermo
  */

  digitalWrite(mA, LOW);
  digitalWrite(mB, HIGH);
}

void loop() {

  //leggo il valore analogico del potenziometro sul pin A0 (0-1023.
  potenziometro = analogRead(pinPotenziometro);

  // Il range dei valori PWM e' da 0 a 255
  velocita = map(potenziometro, 0, 1023, 0, 255);

  Serial.print("velocita = ");
  Serial.print(velocita);

  analogWrite(pinMotore, velocita);
}
```

## SERVOMOTORI

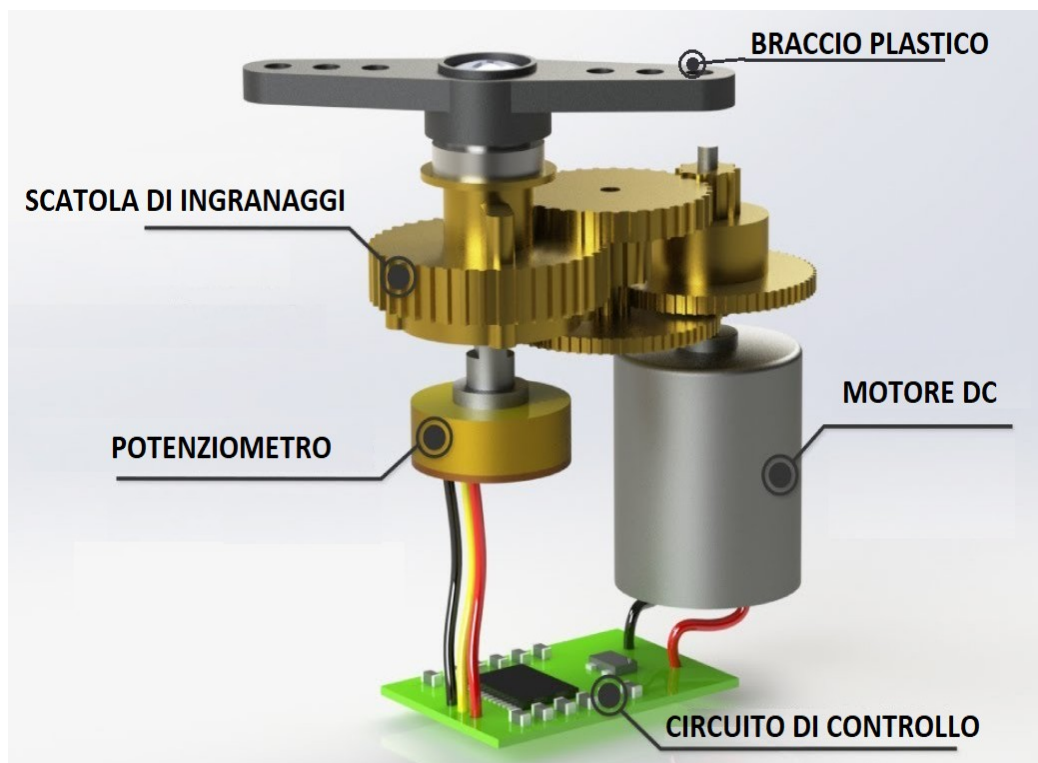
Un servomotore elettrico rotativo è un motore che permette il controllo di precisione della posizione angolare. Il servomotore classico è composto da due elementi principali: il sensore di posizione o feedback e il motore a cui si può aggiungere un riduttore e un freno in caso di necessità. Richiede inoltre un azionamento e/o un controllore più o meno sofisticato a seconda del livello di controllo che si vuole raggiungere.



Hobbistico



Industriale



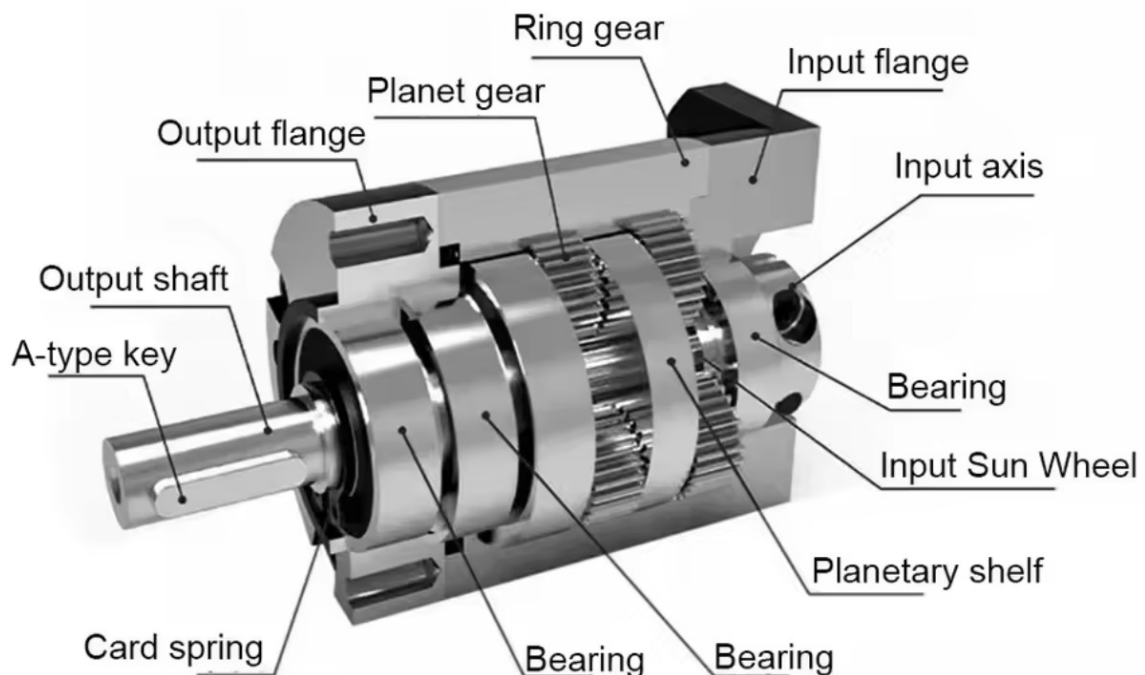
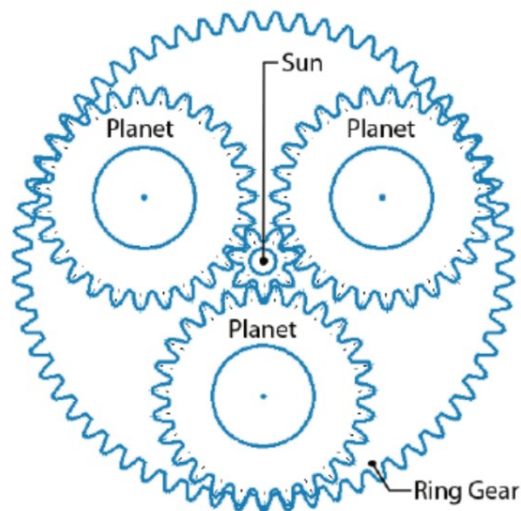
### PLANETARIO (EPICICLOIDALE)

Un riduttore epicicloidale è composto da tre elementi caratteristici:

- un ingranaggio Solare (Sun)
- più satelliti o ingranaggi Planetari (Planet)
- una corona dentata internamente (Ring Gear).

L'albero di ingresso si collega all'ingranaggio solare che trasmette il moto rotatorio agli ingranaggi planetari, che a loro volta ruotano sulla corona interna.

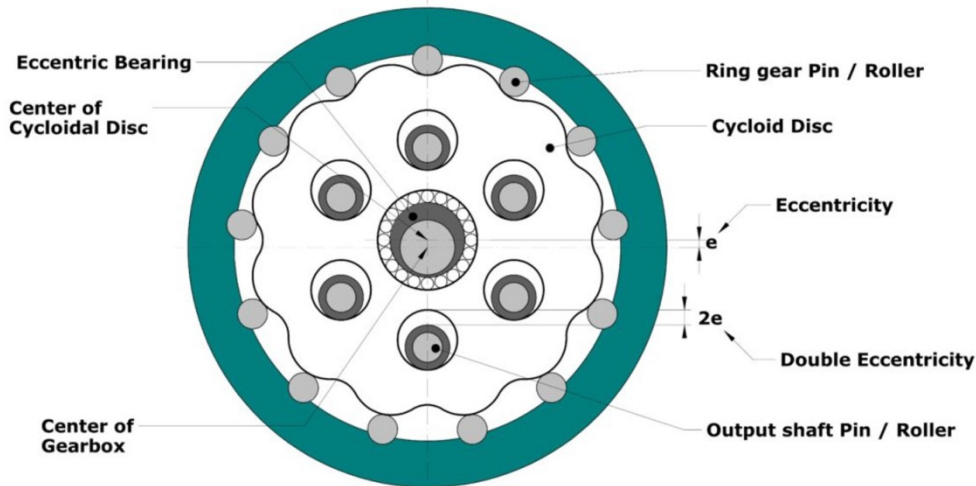
Gli ingranaggi planetari ruotano su perni rigidi solidali ad una piastra chiamata porta-planetario; questa rotazione del porta planetari è ciò che fa ruotare l'albero di uscita.



## CICLOIDALE

Essi sono costituiti da quattro componenti principali:

- Un albero eccentrico che trasmette il moto di ingresso;
- Un disco cicloide che viene montato sull'albero eccentrico;
- Una corona cicloide con cui ingrana il disco cicloide;
- Un albero di uscita.



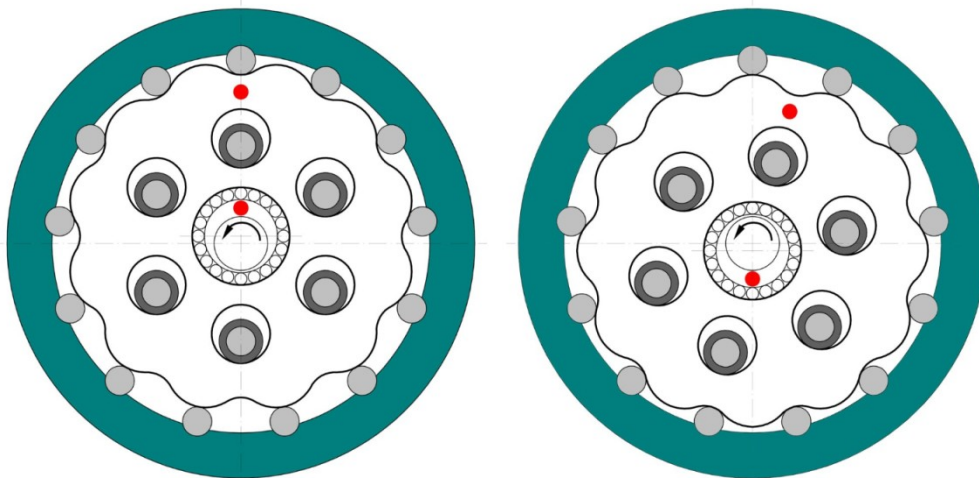
L'albero è mosso da perni che agiscono sui fori del disco cicloide. La rotazione dell'albero eccentrico provoca la rotazione del disco cicloide, che ingrana con la corona cicloide. La rotazione del disco avvia il movimento dei perni nei fori del disco cicloide.

Questo accoppiamento foro-perno consente la rotazione dell'albero di uscita. La velocità di rotazione di quest'ultimo è definita dal rapporto di trasmissione  $i = z_1 / (z_1 + z_2)$ , dove  $z_1$  è il numero di rulli del disco cicloide e  $z_2$  è il numero di lobi presenti sulla corona.

La peculiarità dei sistemi cicloidali è che è possibile ottenere sistemi funzionanti anche con differenza di un solo dente. Di conseguenza, è possibile ottenere rapporti di riduzione molto elevati.

Nella riduzione epicicloidale gli ingranaggi hanno ingranamenti multipli contrariamente a quanto avviene in una riduzione ad ingranaggi paralleli dove l'ingranamento è singolo (fattore di ricoprimento circa 1,15). Pertanto a parità di coppia da trasmettere questo si traduce in forze sui denti inferiori e quindi a parità di forze unitarie, la riduzione epicicloidale necessita di dimensioni contenute rispetto ad una trasmissione ordinaria.

Quando il cuscinetto eccentrico ruota di una rivoluzione, il disco cicloide avanza nella direzione opposta di  $360^\circ$ /passi di perni/rulli. Il verso di rotazione tra entrata e uscita in un riduttore cicloidale è discorde.



Alcuni dati da schede tecniche di riduttori cicloidali per motori Nema 17.



Shaft Diam: D-cut  $\Phi 8\text{mm}$   
 Shaft Length : 21.0 mm  
 Reducer Length: 27 mm  
 Backlash : 3 arcmin  
 Noise :  $\leq 40\text{ dB}$

Gearbox Ratio dependent data		Character	Unit	
Ratio		i		15
Normal Output torque No alternating torque		$T_{2N}$	Nm	6
Normal Output torque Alternating torque permitted for 10.000.000 load changes		$T_{2N 10M10}$	Nm	6
Normal Output torque Alternating torque permitted for 100.000.000 load changes		$T_{2N 100M10}$	Nm	6
Max Output torque for 30.000 output shaft rotations		$T_{2max}$	Nm	10
Emergency stop torque permitted 1000 times		$T_{2stop}$	Nm	12
Idle torque for $n1=3.000\text{ rpm}$ and $20^\circ\text{C}$ gearbox temperature		$T_0$	Nm	0.1
Max. mechanical input speed Operating temperature may not be exceeded		$n_1\text{ limit}$	rpm	3000
Torsional backlash based on output shaft		$j_1$	arcmin	$\leq 3$
Efficiency at $T_{2N}$ , gearbox temperature $70^\circ\text{C}$ and $n1=1.000\text{ rpm}$		$\eta$	%	96



TS 42 15 M 2 N  
 ① ② ③ ④ ⑤ ⑥

- ① : series number
- ② : size (42\*42mm)
- ③ : Ratio 1:15
- ④ : Output M-Shaft
- ⑤ : Motor Thickness  
2 -- 22.2mm  
3 -- 34mm  
4 -- 40mm  
5 -- 49.5mm
- ⑥ : N -- without rearshaft  
D -- with rearshaft

P/N	Ratio	Backlash	Efficiency	Reducer Length	Step Angle	Current	Holding Torque	Total length	Weight	Remark
TS4215M2N	15:1	3 arcmin	Ca. 96%	27 mm	1.8°	1.0 A	1.7 Nm	49 mm	290 g	
TS4215M3N						1.7 A	3.9 Nm	61 mm	390 g	
TS4215M4N						1.7 A	5.6 Nm	67 mm	450 g	
TS4215M4D						1.7 A	5.6 Nm	67 mm	450 g	Rear Shaft
TS4215M5N						2.0 A	7.9 Nm	76.5 mm	520 g	
TS4215M5D						2.0 A	7.9 Nm	76.5 mm	520 g	Rear Shaft

Il funzionamento di un riduttore armonico si basa sull'elasticità meccanica dei metalli e consente di ottenere alti rapporti di trasmissione ed elevata precisione e una costruzione estremamente compatta e leggera.



I componenti principali sono tre: uno speciale cuscinetto a sfere sottile e deformabile montato su un mozzo ellittico, detto "Wave Generator", una coppa cilindrica sottile e dentata all'esterno della sua estremità aperta, detta "Flexspline", deformabile radialmente ma torsionalmente rigida e un anello rigido dentato internamente, detto "Circular Spline".

Nel funzionamento più comune, il Wave Generator è utilizzato come ingresso veloce, collegando il mozzo a un motore o, comunque, alla sorgente di moto; il Circular Spline è fisso mentre il Flexspline è al carico.

Poiché durante il funzionamento la trasmissione di potenza avviene con un numero molteplice di denti in presa, anche con moduli molto piccoli dell'ordine di grandezza dei decimi di millimetro, è possibile trasmettere coppie relativamente elevate.

I riduttori armonici, grazie all'effetto di precarico generato dalla deformazione elastica, hanno gioco zero in uscita e garantiscono grandi precisioni e ripetibilità di posizionamento. Alti rapporti di riduzione, elevata capacità di coppia e una lunga durata sono altre peculiarità. Inoltre, grazie anche al modo in cui i denti ingranano tra loro, i rendimenti superano anche l'85% e rumore e vibrazioni sono inferiori rispetto ad altri tipi di riduttori.

I rapporti di riduzione, in un singolo stadio, sono tipicamente compresi tra 30:1 e 160:1 ma possono arrivare fino a 320:1.

Con ingombri minimi le coppie sono elevate e possono variare da pochi Nm a centinaia di Nm.

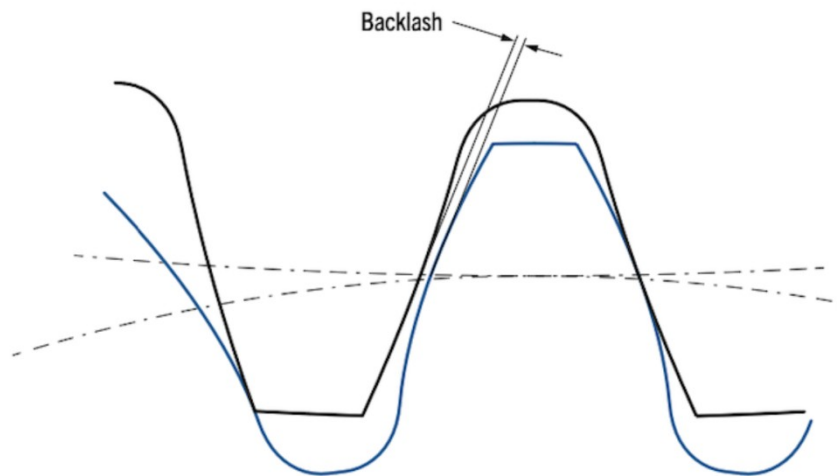
Le velocità massime in ingresso tipicamente sono comprese tra i 2500 e i 6.000 rpm ma, in alcuni casi, nei riduttori più piccoli, possono arrivare e superare i 10.000 rpm.



Technical Specifications		
ITEM	Unit	Info
Step angle	°	0.06
Phases		2
Ratio		1:30
Rated current	A	1.4
Resistance	Ω	2.3 ± 10%
Inductance	mH	3.8 ± 20%
Backlash	arcmin	≤ 2
Noise	dB	≤ 40
Sealing		ball bearing
Protective system		IP 40
Lubrication		life grease
Insulation class		B
Operating temperature	°C	-25 ~ +50
Storage temperature	°C	-25 ~ +80
Weight	g	ca. 383

## IL GIOCO NEI RIDUTTORI (BACKLASH)

Il backlash (gioco d'ingranaggi) è lo spazio tra i denti di due ruote dentate accoppiate che permette il funzionamento senza surriscaldamento. È un parametro critico che causa un ritardo nella trasmissione del movimento quando si inverte la rotazione (gioco di inversione). Riduttori di precisione, come quelli planetari, mirano a minimizzarlo.

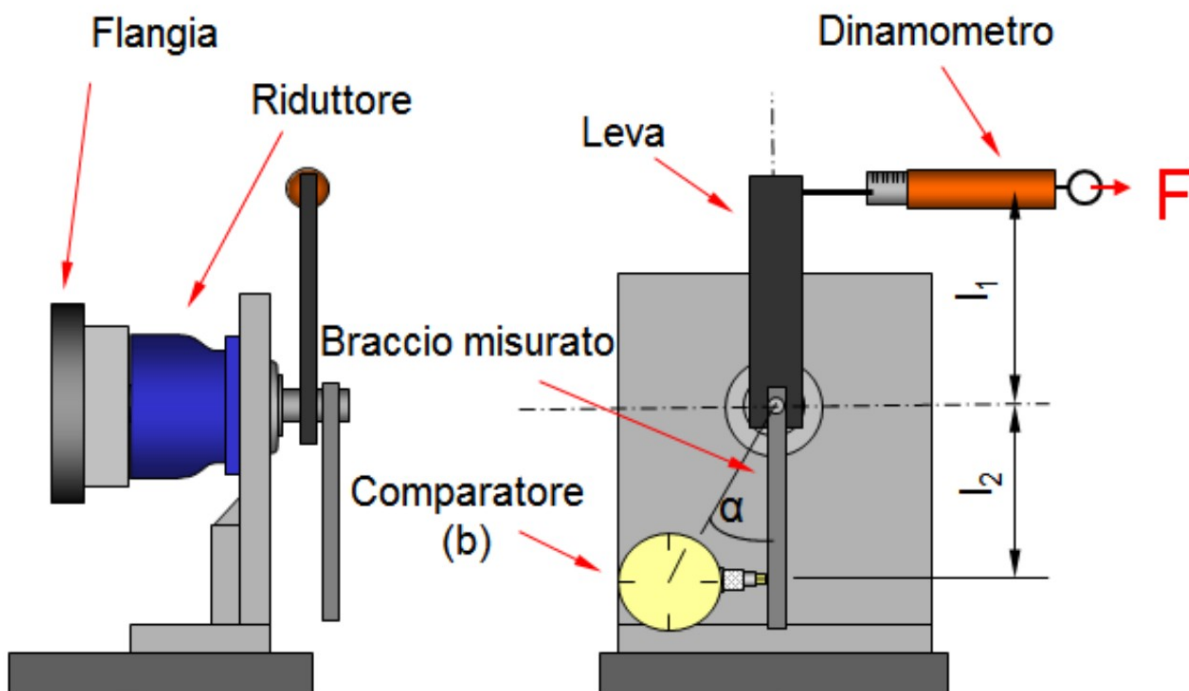


Il backlash è una necessità funzionale. Serve ad evitare interferenze meccaniche, surriscaldamenti, rumorosità e rotture precoci tra i denti.

Il gioco angolare è la caratteristica tecnica che ha maggiormente influenzato il mercato dei riduttori. Influisce non solo sulla precisione e sul comportamento dinamico dell'asse ma anche sulla rumorosità, sulla durata e soprattutto sul costo del riduttore.

### Misurazione

Il backlash si definisce come il movimento a vuoto misurato sull'albero di uscita quando quello di ingresso è bloccato. Ecco come viene misurato, tenendo conto che non c'è univocità di metodo tra i diversi costruttori:



## COMPARAZIONE E SCELTA

Caratteristica	Riduttore Planetario	Riduttore Cicloidale	Riduttore Armonico
<b>Principio Meccanico</b>	Ingranaggi a denti dritti o elicoidali (Sole, Satelliti, Corona).	Disco con profilo cicloidale che ruota attorno a perni fissi.	Deformazione elastica di una tazza metallica (Flexspline).
<b>Gioco (Backlash)</b>	1 – 15 arcmin (standard); <1-3 (alta precisione).	Molto basso (<1 arcmin).	Zero-backlash (intrinseco).
<b>Rapporto di Riduzione</b>	Basso-Medio (3:1 a 10:1 per stadio).	Alto (30:1 a 170:1 in un solo stadio).	Molto Alto (50:1 a 320:1 in un solo stadio).
<b>Efficienza</b>	Alta (90-97%).	Media (80-90%).	Media-Bassa (75-85%).
<b>Densità di Coppia</b>	Buona.	Eccellente (sopporta carichi d'urto elevati).	Buona (ma limitata dalla fatica del metallo).
<b>Ingombro e Peso</b>	Moderato (allungato se multistadio).	Compatto e robusto.	Estremamente compatto e leggero.
<b>Rigidità Torsionale</b>	Alta.	Altissima.	Moderata (limitata dalla flessibilità del componente).
<b>Costo</b>	Generalmente più economico.	Elevato (costruzione complessa).	Elevato (materiali speciali).

## ANALISI DEI PUNTI CHIAVE

### 1. Riduttore Planetario

È il "tuttofare". Offre un'ottima efficienza e una grande versatilità. È la scelta ideale quando è necessaria un'alta velocità di rotazione e una trasmissione di potenza fluida, ma può soffrire di un leggero gioco meccanico che lo rende meno adatto a posizionamenti millimetrici estremi rispetto agli altri due.

### 2. Riduttore Cicloidale

È il "carro armato" dei riduttori. Grazie al fatto che i componenti lavorano in compressione anziché in taglio (come i denti degli ingranaggi), può sopportare picchi di carico improvvisi senza rompersi. È perfetto per la base dei robot industriali o per macchine pesanti dove la precisione deve sposarsi con la robustezza.

### 3. Riduttore Armonico (Harmonic Drive)

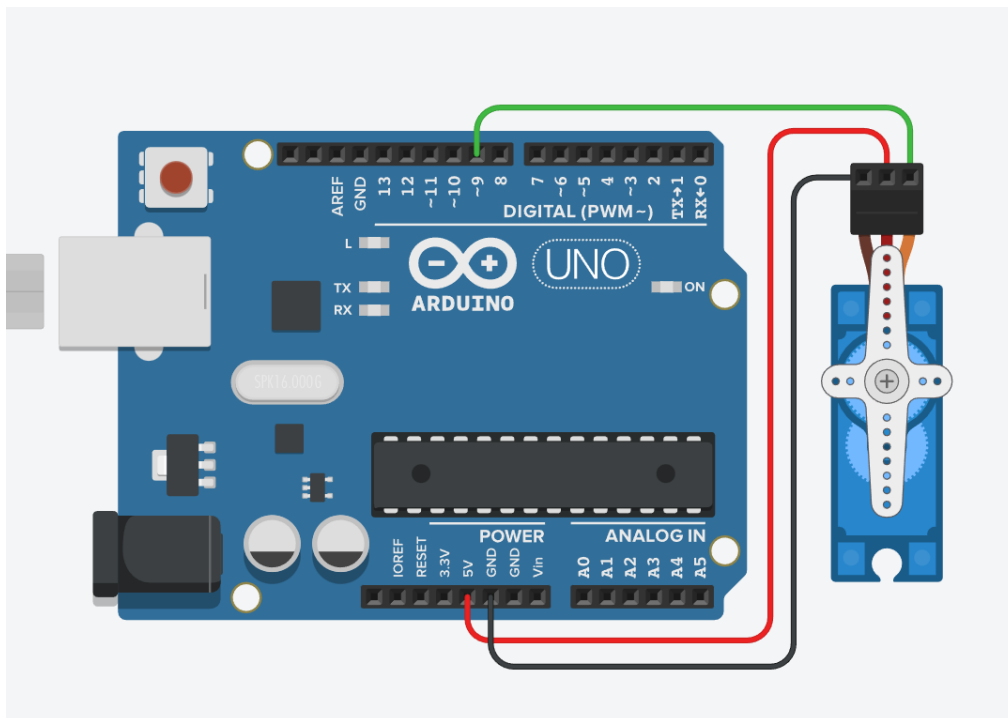
È il re della robotica collaborativa e dell'aerospazio. Sfrutta la deformazione di un anello metallico sottile. Il vantaggio principale è l'assenza totale di gioco e la capacità di ottenere rapporti di riduzione enormi in uno spazio minuscolo. Tuttavia, è più delicato e meno rigido rispetto a un cicloidale.

### Quale scegliere?

- Planetario per automazione generale, nastri trasportatori o dove l'efficienza energetica è prioritaria.
- Cicloidale per giunti di robot industriali pesanti o tavole rotanti soggette a forti urti.
- Armonico per bracci robotici leggeri (COBOT), strumenti chirurgici o sistemi satellitari dove ogni grammo conta.

## GESTIONE SERVOMOTORE DIRETTA CON ARDUINO

Muovere l'albero del servo a destra e a sinistra ( $180^{\circ} \rightarrow 0^{\circ}$  e viceversa) con passo di  $1^{\circ}$ .



### CODICE

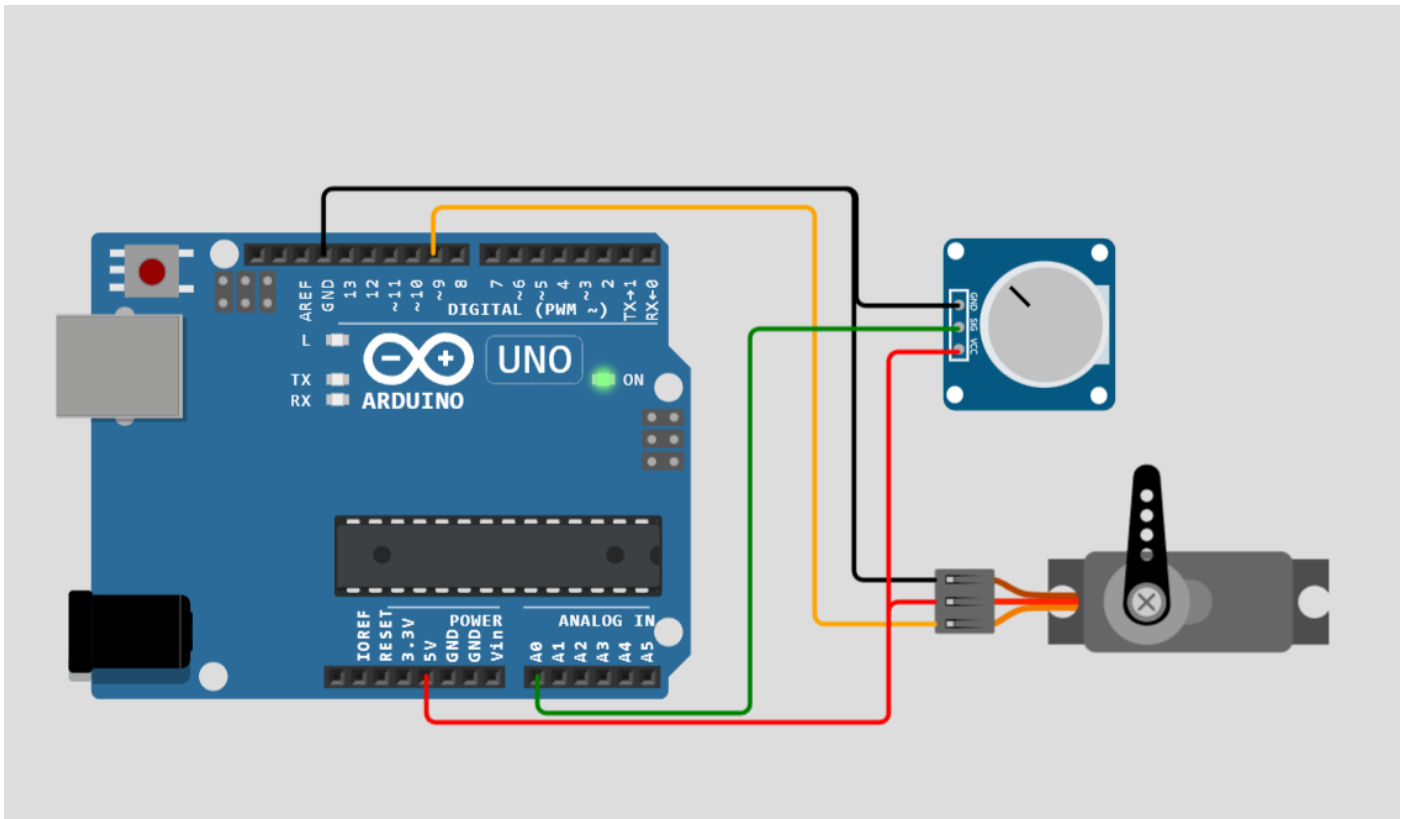
```
#include <Servo.h>
Servo servo_9;
int pos = 0;

void setup()
{
  servo_9.attach(9, 500, 2500);
}

void loop()
{
  // sweep the servo from 0 to 180 degrees in steps of 1 degrees
  for (pos = 0; pos <= 180; pos += 1) {
    servo_9.write(pos);
    delay(15); // Wait for 15 millisecond(s)
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    servo_9.write(pos);
    // wait 15 ms for servo to reach the position
    delay(15); // Wait for 15 millisecond(s)
  }
}
```

### COMPITO

1. Modificare il circuito affinché il servomotore abbia una alimentazione dedicata con transistor NPN.
2. Modificare il codice per gestire due servomotori tramite comandi inviati dal monitor seriale (1 → motore DX, 2 → motore SX, 3 → entrambi i motori e 4 → stop).



simulabile su "wokwi.com"

## CODICE

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

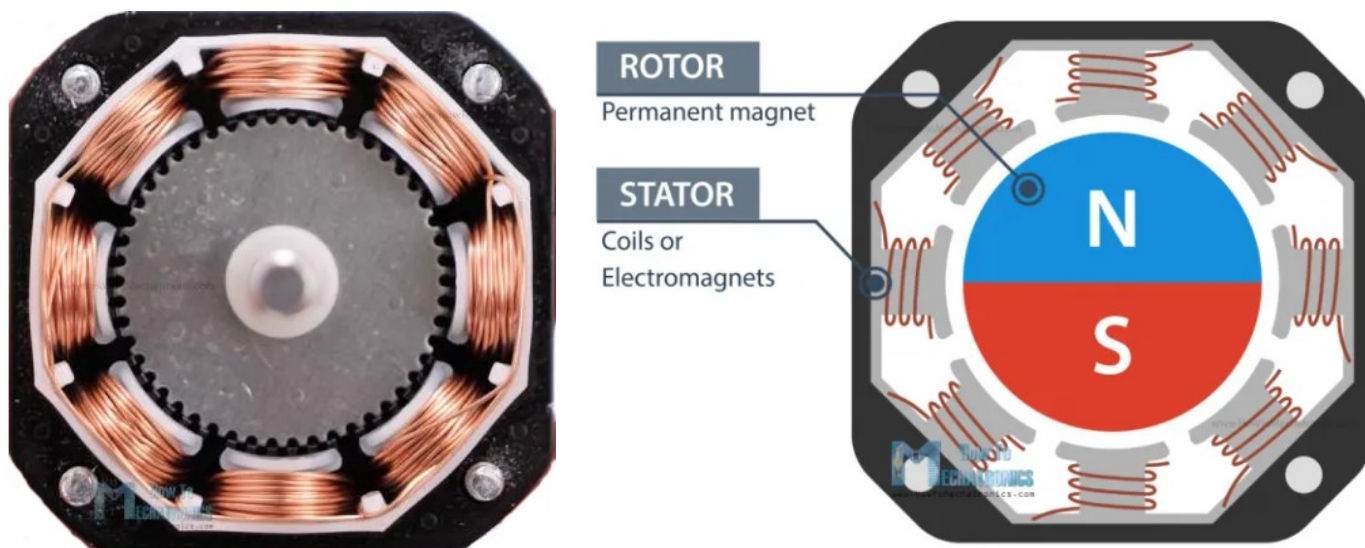
void loop() {
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```

## MOTORE STEPPER (PASSO-PASSO)

Il motore passo passo è un tipo di motore a corrente continua sincrono senza spazzole che, diversamente dagli altri tipi standard di motori elettrici, non ruota in continuazione per un numero arbitrario di giri fino a che non viene interrotto il flusso di corrente continua. Al contrario, i motori passo passo sono dispositivi con controllo digitale delle sorgenti in input e in output, per l'avvio e l'arresto di precisione.

Sono costruiti in modo tale che la corrente che li attraversa passi in una serie di bobine disposte in fase, che possono essere attivate o disattivate in rapida sequenza. Questo permette al motore di girare una frazione di rotazione alla volta, ed è a ciascuno di questi predeterminati passi (step in inglese) che il motore deve il suo nome (motore stepper).

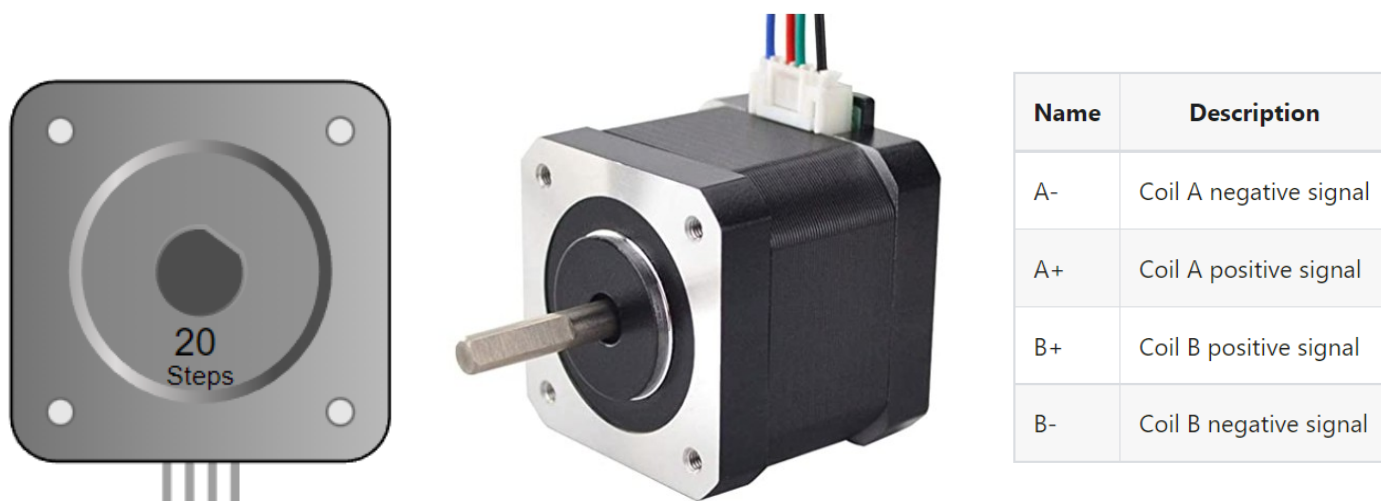
Un motore passo passo è costruito in modo da suddividere una singola rotazione completa in un numero di gran lunga minore di rotazioni parziali uguali.



Il risultato finale è che un motore passo passo può trasferire movimenti minuziosamente accurati a parti meccaniche che richiedono elevati gradi di precisione. Di default il motore passo-passo sposta l'albero motore di 1,8 gradi per passo (200 passi per giro). In generale la velocità massima di un motore stepper è di circa 1000 rpm.

All'aumentare della velocità diminuisce la coppia motrice disponibile (che può essere aumentata montando un riduttore).

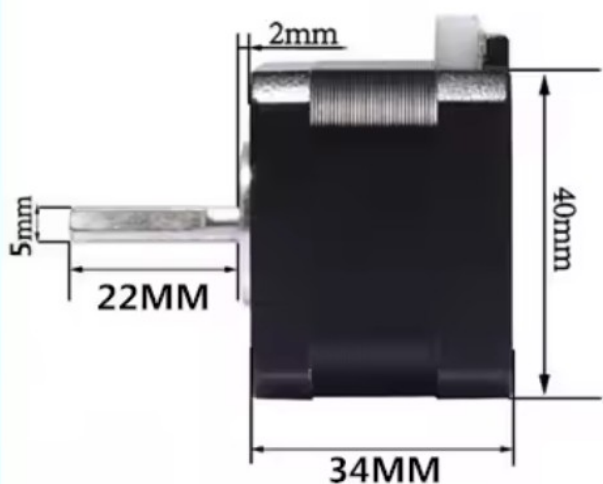
Usando opportuni driver il motore supporta anche mezzo passo (0,9 gradi per passo / 400 passi per giro) ed anche micropassi più piccoli (ad es.  $\frac{1}{2}$ ,  $\frac{1}{4}$  o  $\frac{1}{8}$  di passo).



Tipicamente le due coppie polari sono identificate dai seguenti colori. In ogni caso conviene fare riferimento alla scheda tecnica del produttore.



-  **BLACK A+**
-  **GREEN A-**
-  **RED B+**
-  **BLUE B-**



## DRIVER A4988

Quando si utilizza un motore passo-passo è necessario un chip driver (es. driver A4988) in grado di fornire l'elevata quantità di corrente richiesta dalle bobine del motore.

I motori passo-passo standard hanno 200 passi per giro (i passi sono distanziati di 1,8 gradi).

Il driver stepper supporta il microstepping: consente di muovere il motore a meno di un passo per ogni impulso.

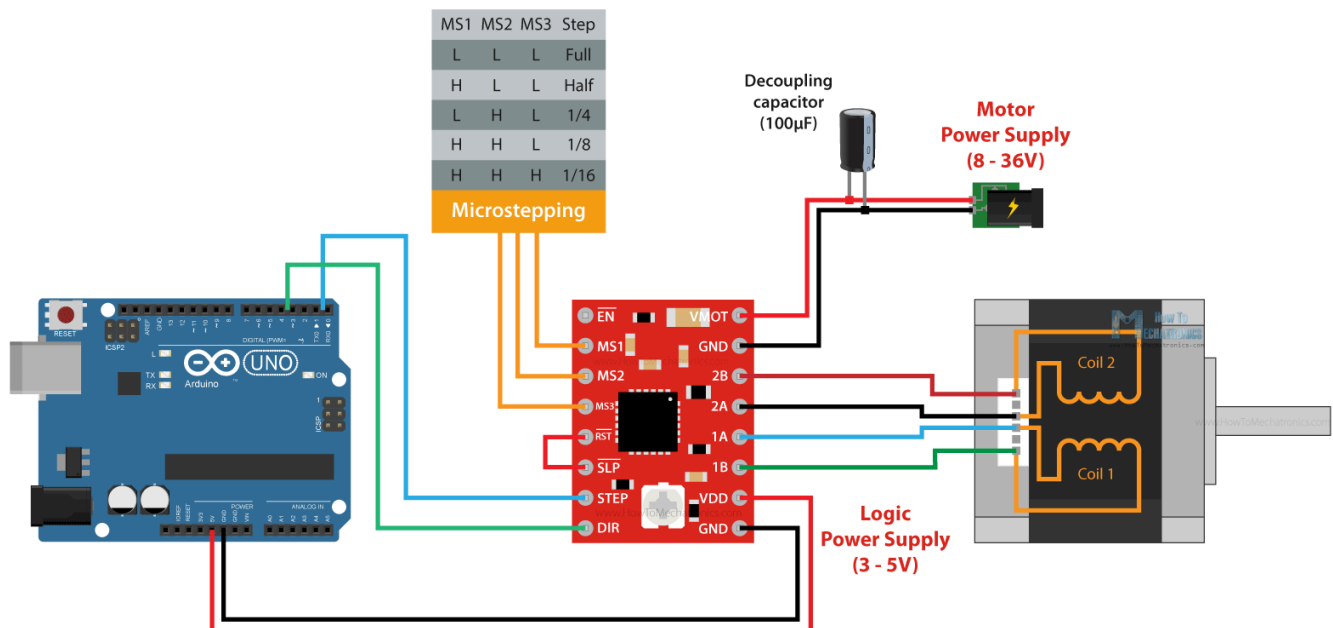
Il microstepping consente un controllo più preciso del movimento del motore (con una riduzione della coppia motrice).

Utilizzare i pin MS1/MS2/MS3 per selezionare la configurazione microstepping per il driver stepper:

MS1	MS2	MS3	Operation mode	Degrees	Microsteps/revolution
0	0	0	Full step (default)	1.8	200
1	0	0	Half step	0.9	400
0	1	0	1/4 step*	0.45	800
1	1	0	1/8 step*	0.225	1600
1	1	1	1/16 step*	0.1125	3200



Come collegare il driver A4988 con il motore passo-passo e il controller Arduino.



### NOTA BENE:

Per motori a bassa resistenza interna (tipica degli stepper) è necessario un driver di corrente e non un driver di tensione come l'L298N. I motori a bassa impedenza sono controllati in corrente, non in tensione.

Per valori di resistenza dell'avvolgimento oltre 30-60 ohm un L298N funziona senza bruciarsi, ma la velocità massima è molto inferiore rispetto a quella ottenibile con un driver di corrente.

## UTILIZZO DEL DRIVER PASSO-PASSO A4988

Collegare i pin del motore passo-passo ai pin 1B/1A/2A/2B del driver.

Il pin RESET deve essere HIGH e quindi si può collegare al pin SLEEP adiacente che è impostato HIGH di default.

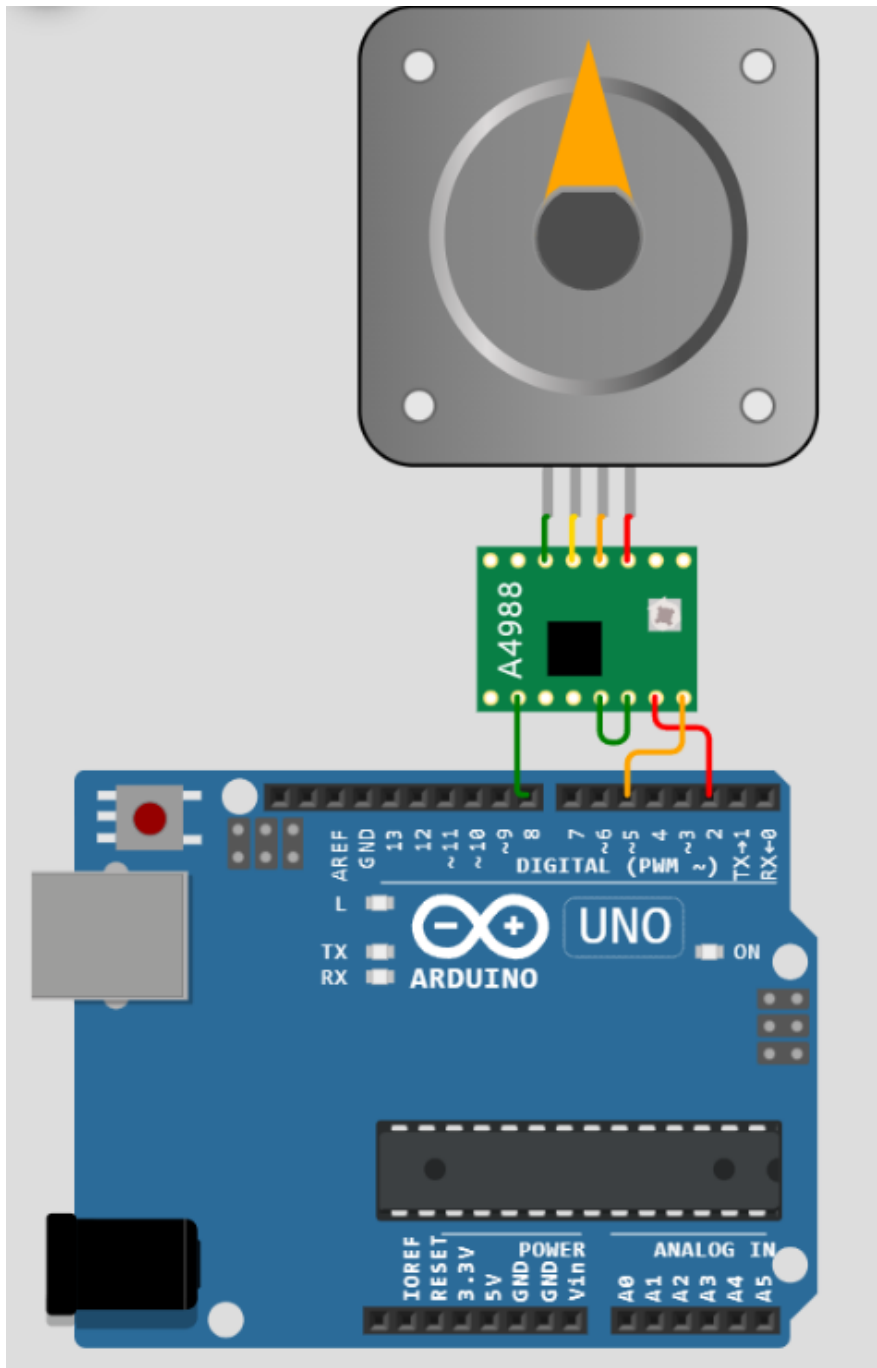
Utilizzare il pin STEP per spostare il motore passo-passo.

Ogni impulso ALTO su questo pin sposterà il motore di un passo (o micropasso, a seconda dei pin MS1/MS2/MS3).

Quando il pin DIR è ALTO, il motore passo-passo si sposterà in senso orario.

Quando il pin DIR è BASSO, il motore si muoverà in senso antiorario.

Ad esempio, se DIR, MS1 e MS3 sono LOW e MS2 è HIGH (modalità 1/4 step), l'impulso del pin STEP sposterà il motore di 1/4 step (0,45 gradi) in senso antiorario.



simulabile su "wokwi.com"

## CODICE

```
#define DIR_PIN 5 // X
#define STEP_PIN 2 // X
#define EN_PIN 8

#define DELAY_ST 5000

int stps=400; // 2 giri completi

void setup() {
  pinMode(DIR_PIN, OUTPUT);
  pinMode(STEP_PIN, OUTPUT);
  pinMode(EN_PIN, OUTPUT);
  digitalWrite(EN_PIN, LOW);
  delay(1000);
}

void loop() {
  // rotazione ORARIA
  step(true, DIR_PIN, STEP_PIN, stps);
  delay(1000);
  // rotazione ANTIORARIA
  step(false, DIR_PIN, STEP_PIN, stps);
  delay(1000);
}

// dir = true= oraria
void step(boolean dir, byte dirPin, byte stepperPin, int steps)
{
  digitalWrite(dirPin, dir);
  delay(100);
  for (int i = 0; i < steps; i++) {
    digitalWrite(stepperPin, HIGH);
    delayMicroseconds(DELAY_ST);
    digitalWrite(stepperPin, LOW);
    delayMicroseconds(DELAY_ST);
  }
}
```

## DRIVER DRV8825 CONTRO A4988

Possiamo controllare i motori passo-passo con altri driver come il DRV8825.

Il principio di funzionamento, le connessioni e la codifica sono quasi le stesse per entrambi questi driver.

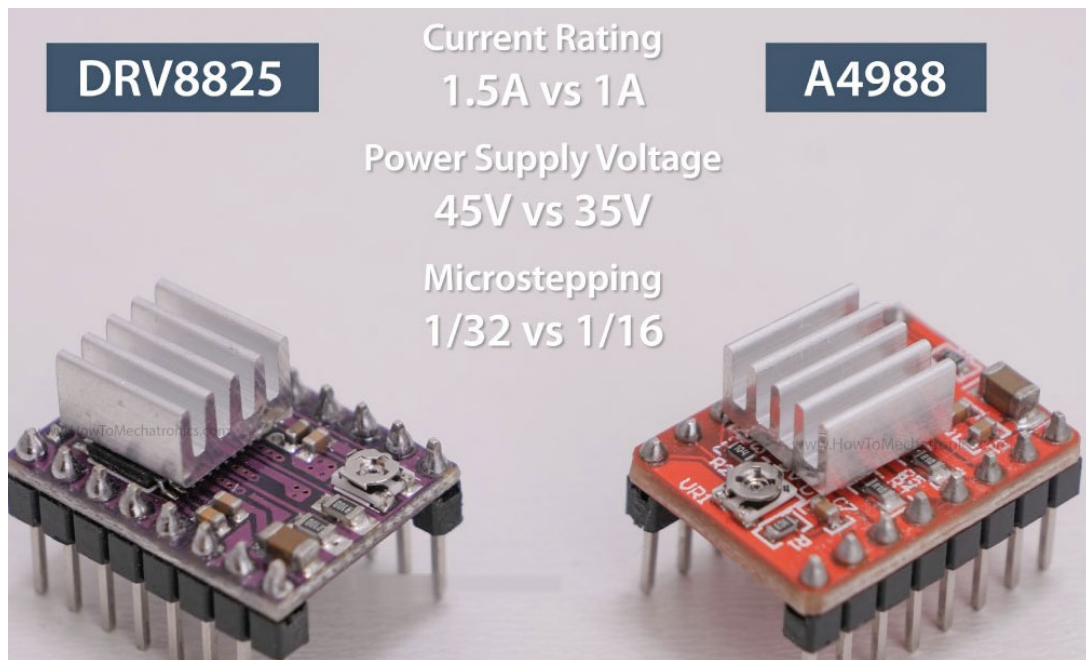
La differenza tra loro sta nelle loro caratteristiche tecniche.

Il DRV8825 è un driver passo-passo di Texas Instruments che può essere utilizzato come sostituto diretto del driver Allegro A4988 poiché le loro connessioni sono le stesse.

Le tre differenze principali tra loro sono che il DRV8825

- può fornire più corrente rispetto all'A4988 senza raffreddamento aggiuntivo (1,5 A vs 1 A)
- ha una tensione di alimentazione massima più alta (45 V vs 35 V)
- offre una risoluzione microstepping più elevata (32 vs 16 microstep )

Altri driver più recenti come il TMC2208 presentano caratteristiche ancora migliori e soprattutto una silenziosità in funzionamento decisamente migliore dei precedenti.





# SISTEMI DI REGOLAZIONE

Un sistema di **REGOLAZIONE** non prevede l'utilizzo di sensori ma si basa sulle leggi fisiche che governano il sistema.

Ad esempio per mantenere una certa temperatura dell'acqua in un recipiente è possibile far ricorso alle leggi della termotecnica per calcolare la dispersioni termiche dell'involucro e quindi la potenza termica che deve essere fornita tramite un elemento riscaldante.

Applicando correttamente le leggi della fisica si può ottenere il risultato richiesto senza un controllo continuo del sistema.

Nel caso di presenza di disturbi esterni (non adiabaticità del recipiente, prelievo di acqua, ecc.) il risultato non può essere raggiunto senza la presenza di opportune **sensori** che misurano in tempo reale la grandezza da controllare.

## SISTEMA DI RISCALDAMENTO RESISTIVO

Progettare un sistema di **REGOLAZIONE** della temperatura dell'acqua in un recipiente adiabatico da 10 litri.



La temperatura dell'acqua deve essere portata da 20°C a 50°C con una resistenza termica da 115 watt quando si preme un pulsante di accensione. Il sistema **NON** prevede l'utilizzo di sensori di temperatura e per questo motivo si parla di **REGOLAZIONE** e non di **CONTROLLO**.

Se si applicano correttamente le leggi della fisica alla base del processo di riscaldamento dell'acqua si può ottenere il risultato richiesto. Nel caso di presenza di disturbi esterni (non adiabaticità, prelievo acqua prima del termine fase di riscaldamento, ecc.) il risultato non può essere raggiunto senza la presenza di sensori.

Leggi fisiche coinvolte:

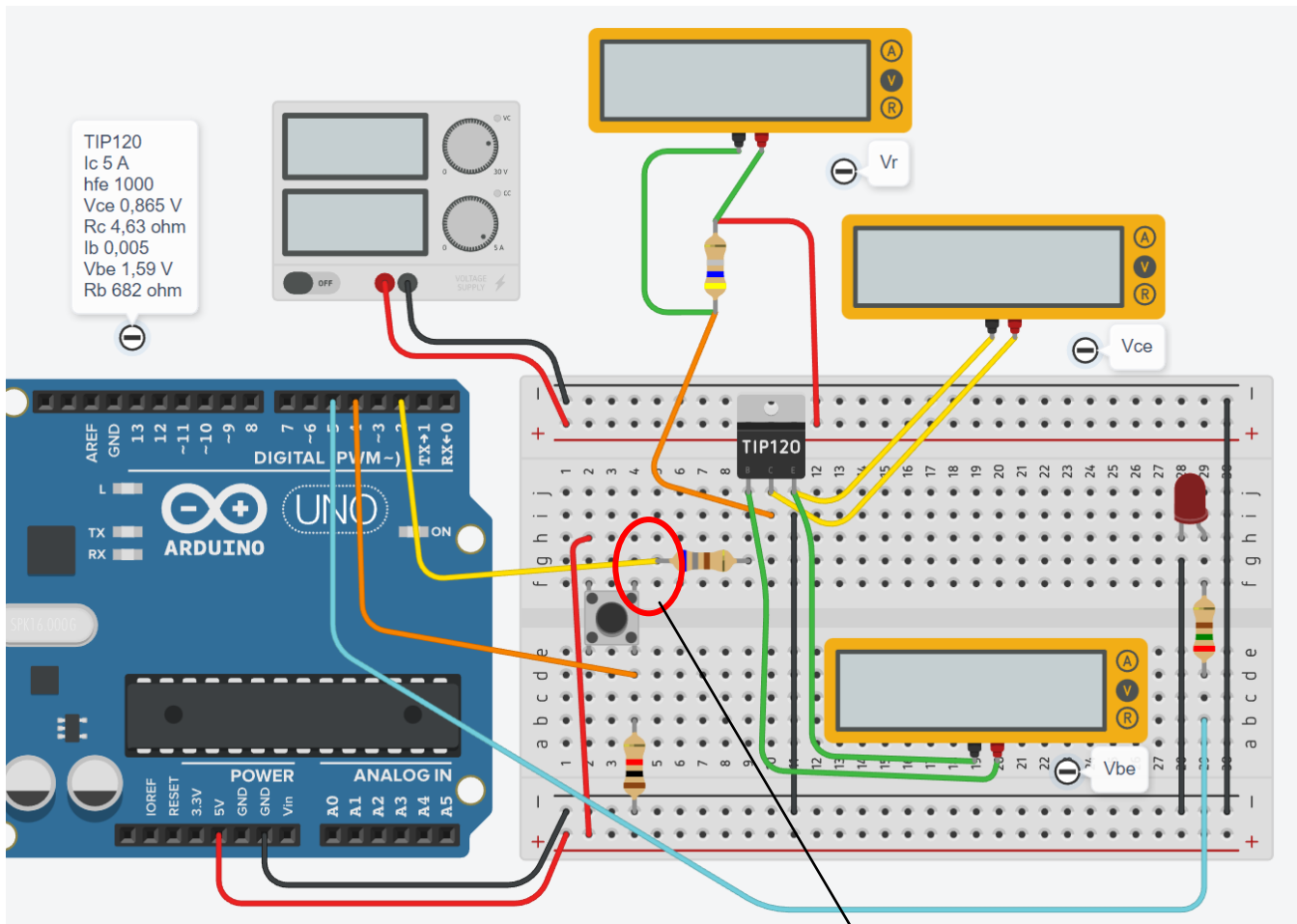
- 1- Energia termica fornita all'acqua:  $Q = m \cdot Ct \cdot (T_f - T_i)$  [Joule]
- 2- Potenza termica fornita dall'elemento riscaldante:  $Pot. = Q / t$  [w=J/s]

Ricavando il calore fornito dalla 2° equazione e sostituendolo nella prima si ottiene il tempo "t" necessario ad ottenere il risultato richiesto.

Il microcontrollore dovrà quindi attivare l'elemento riscaldante per il tempo calcolato.

$$t = m \cdot Ct \cdot (T_f - T_i) / Pot. = 10 \cdot 4186 \cdot (50 - 20) / 115 = 10920 \text{ s} = 95 \text{ minuti}$$

Per valutare il tempo trascorso in Arduino si deve impiegare la funzione "millis()" che ritorna il numero di millisecondi trascorsi dall'accensione.



TIP120  
 Ic 5 A  
 hfe 1000  
 Vce 0,865 V  
 Rc 4,63 ohm  
 Ib 0,005  
 Vbe 1,59 V  
 Rb 682 ohm

Attenzione a non mettere i pin del pulsante allineati in verticale alla resistenza!

```

long t0;
int tempoAttivazione= 10*1000; // secondi
bool flagAttivo = false; // flag per sapere se è attivo riscaldamento

void setup()
{
  pinMode(2, OUTPUT); // TIP120
  pinMode(5, OUTPUT); // LED
  pinMode(4, INPUT); // START
  Serial.begin(9600);
}

void loop()
{
  int statoStart= digitalRead(4); // STATO BOTTONE START

  if (statoStart==HIGH) {
    Serial.println("Attivazione");
    flagAttivo= true;
    t0= millis(); // ISTANTE ATTIVAZIONE
    digitalWrite(2, HIGH);
    digitalWrite(5, HIGH);
  }
  // CONTROLLO SE È PASSATO IL TEMPO DI ACCENSIONE PREVISTO
  if ( ((millis() - t0) > tempoAttivazione) && (flagAttivo== true) ) {
    Serial.println("Fine riscaldamento");
    digitalWrite(2, LOW);
    digitalWrite(5, LOW);
    flagAttivo= false;
  }

  delay(200);
}

```



# SISTEMI DI CONTROLLO

Un sistema di **REGOLAZIONE** non prevede l'utilizzo di sensori ma si basa sulle leggi fisiche che governano il sistema.

Ad esempio per mantenere una certa temperatura dell'acqua in un recipiente è possibile far ricorso alle leggi della termotecnica per calcolare le dispersioni termiche dell'involucro e quindi la potenza termica che deve essere fornita tramite un elemento riscaldante.

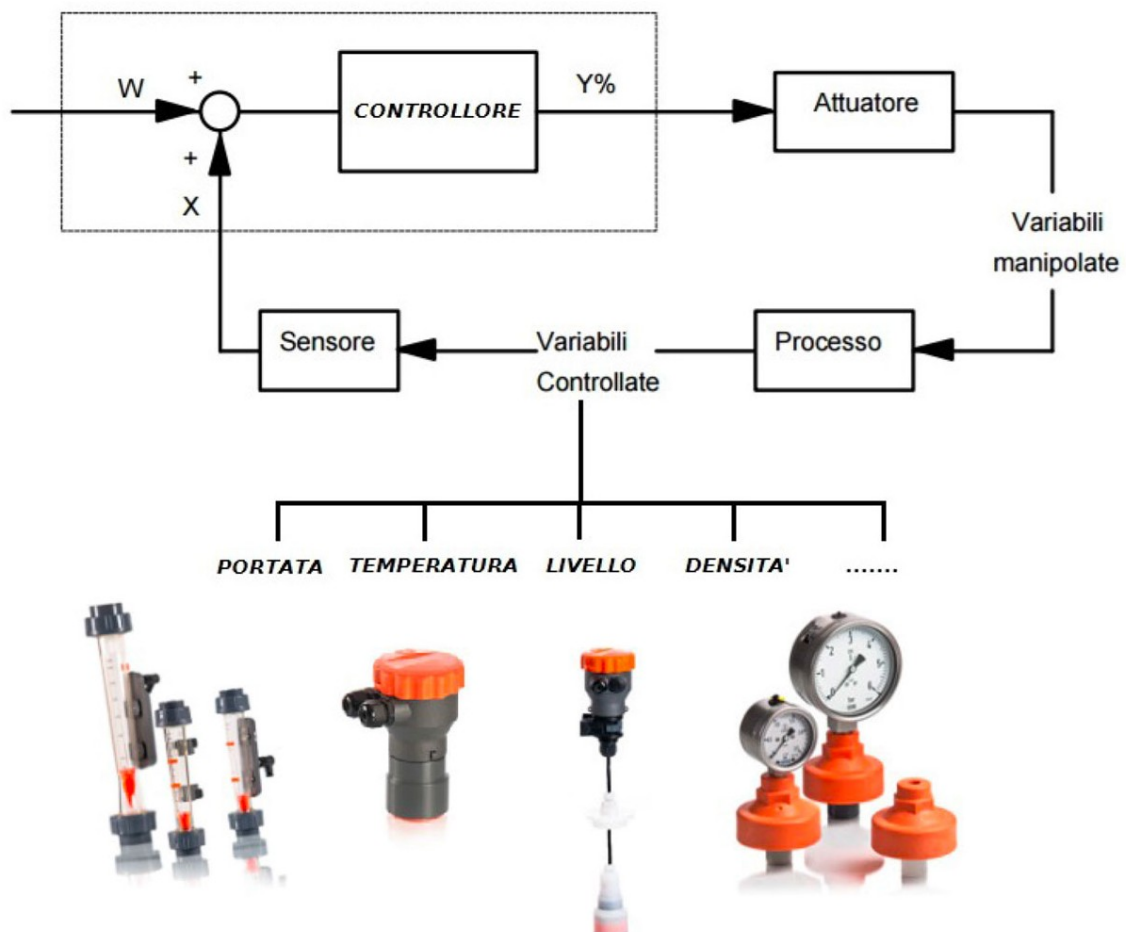
Applicando correttamente le leggi della fisica si può ottenere il risultato richiesto senza un controllo continuo del sistema.

Nel caso di presenza di disturbi esterni (non adiabaticità del recipiente, prelievo di acqua, ecc.) il risultato non può essere raggiunto senza la presenza di opportune **sensori** che misurano in tempo reale la grandezza da controllare.

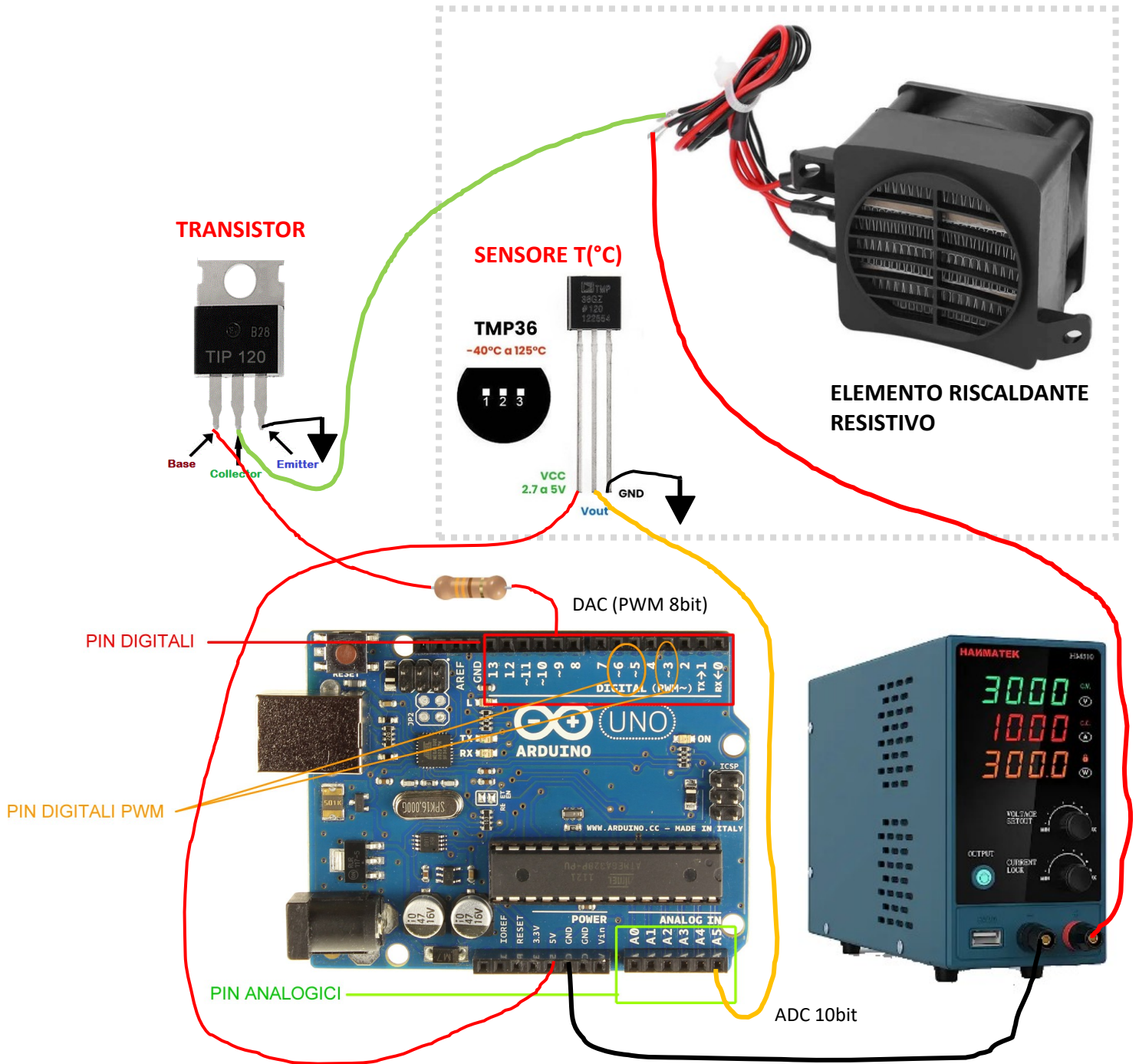
Sulla base della misura si interviene poi degli **attuatori** per portare la grandezza controllata al livello desiderato (retroazione).

In questa situazione si parla quindi di sistema di **CONTROLLO AD ANELLO CHIUSO**.

L'immagine seguente mostra lo schema di massima di un generico sistema di controllo.







Il microcontrollore e l'alimentatore di potenza devono sempre avere la massa in comune.

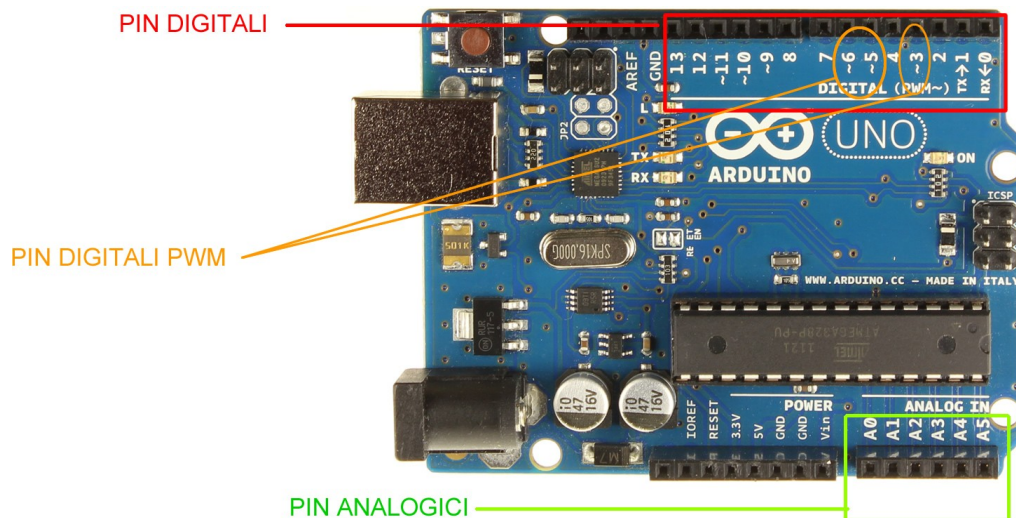
Il sensore e la base del transistor vengono alimentati dal microcontrollore.

L'attuatore è alimentato dal generatore di potenza e collegato al collettore del transistor.

## GENERARE SEGNALI ANALOGICI (DAC) CON ARDUINO

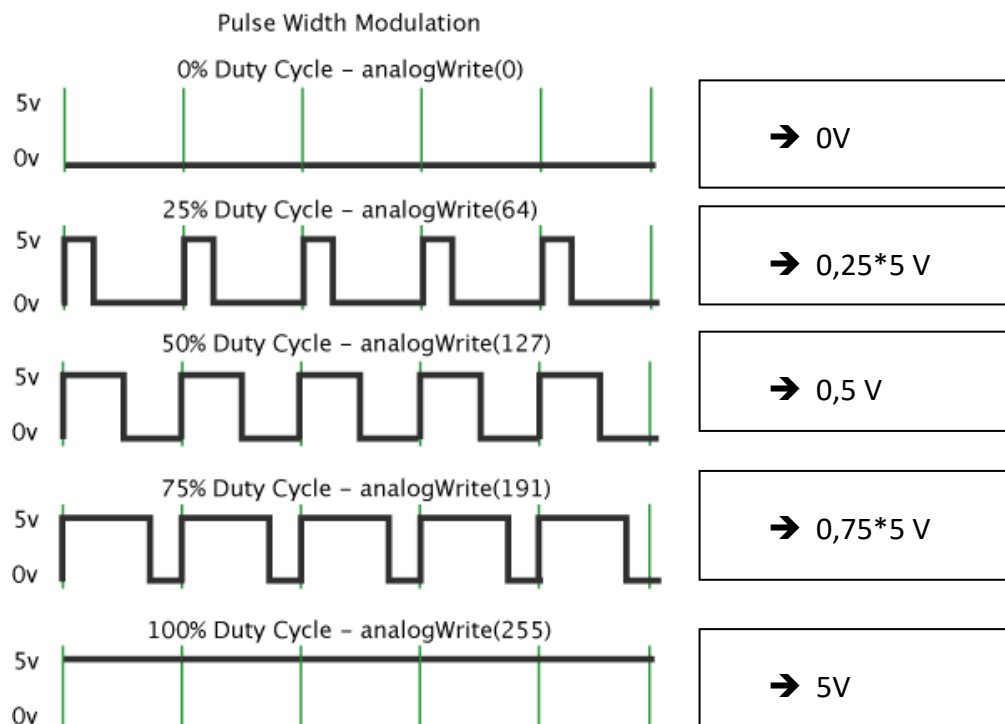
I pin digitali si dividono in base al supporto o meno della funzione PWM (pulse wide modulation).

I pin che hanno la PWM sono: 3,5,6,9,10,11.



Con un pin PWM è possibile generare in uscita un segnale analogico da 0-5V con una risoluzione di 8 bit ( $5/255 \text{ volt} \approx 0,02V$ ). Un segnale PWM è in termini molto semplicistici, un onda quadra 0-5V (ad alta frequenza) con delle durate prestabilite per la parta alta (5V).

Ciò permette di simulare un valore analogico di tensione compreso tra 0-5V con uno digitale con la maggior parte degli attuatori (transistor, relè, motori CC ...).

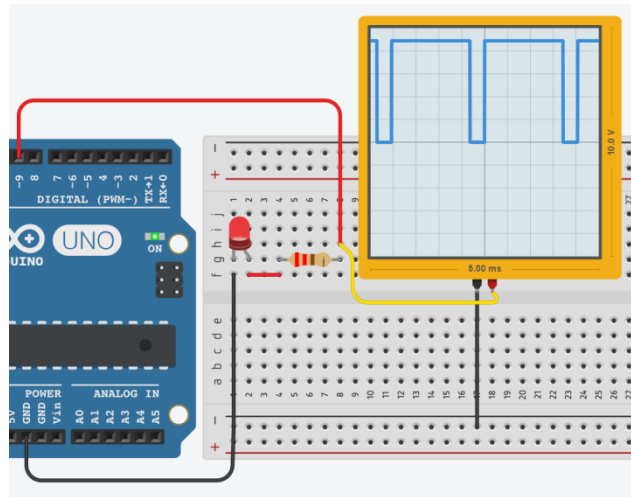
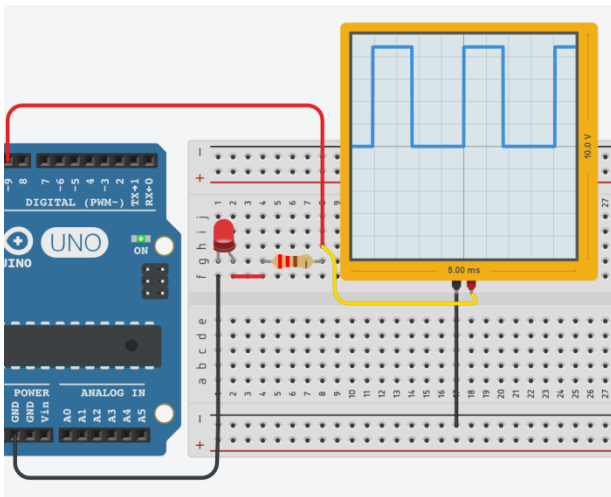
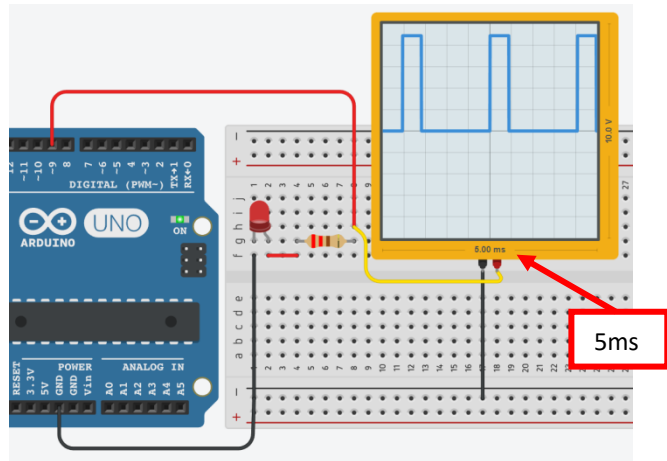
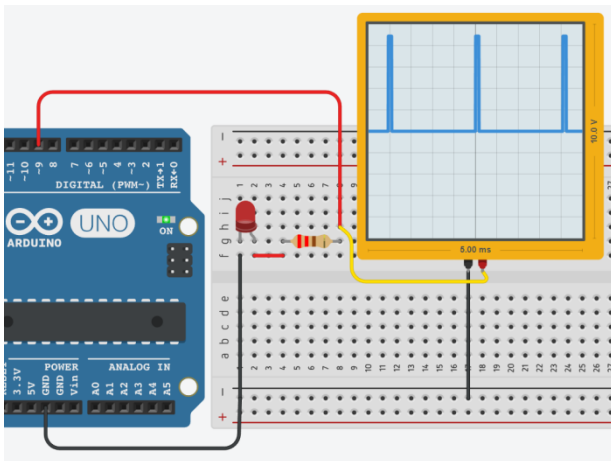


I segnali analogici in uscita sono di fondamentale importanza per poter effettuare sistemi di controllo evoluti come ad esempio il PID.

Tramite un segnale analogico che varia da 0 a 5V si può regolare la potenza assorbita da un attuttore (motori, elementi riscaldanti, generatori di forza ecc.) e di conseguenza l'effetto sul sistema controllato.

## ESERCIZIO: VARIARE LA LUMINOSITA' DI UN DIODO LED

Tramite la PWM si può variare la corrente che scorre in un diodo LED (variando la tensione sulla resistenza) e di conseguenza la sua luminosità. Questa tecnica viene usata nelle lampadine in CC trimmerabili.



### CODICE:

```
int ledPin = 9; // LED su Pin 9
```

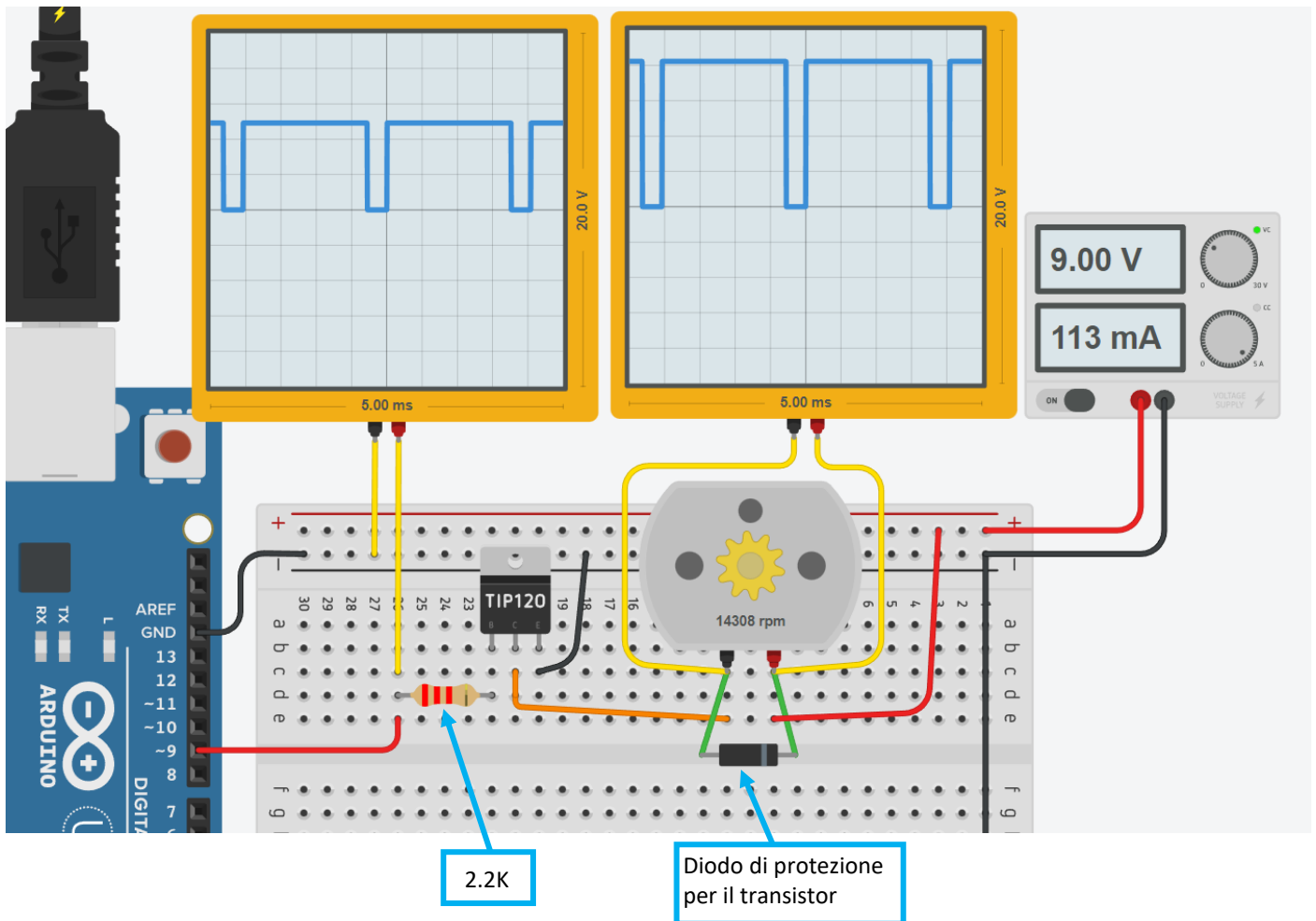
```
void setup(){  
  pinMode(ledPin, OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop(){  
  // aumento luminosità da 0 al massimo  
  for(int dimValue = 0; dimValue <= 255; dimValue = dimValue + 5){  
    analogWrite(ledPin, dimValue);  
    Serial.println(dimValue);  
    delay(30);  
  }  
  // diminuisco luminosità dal massimo a 0  
  for(int dimValue = 255; dimValue >= 0; dimValue = dimValue - 5){  
    analogWrite(ledPin, dimValue);  
    Serial.println(dimValue);  
    delay(30);  
  }  
}
```

## COME VARIARE LA VELOCITA' DI UN MOTORE C.C. MANTENENDO ALTA LA COPPIA MOTRICE

Tramite la PWM si può variare la corrente che scorre nel motore e di conseguenza la sua velocità.

Poiché la corrente assorbita dal motore è superiore ai 30-40 mA fornibili da Arduino è necessario utilizzare un transistor che viene comandato da Arduino tramite un segnale PWM.



### CODICE

```
#define DC_MOTOR_PIN 9
```

```
void setup() {  
  pinMode( DC_MOTOR_PIN, OUTPUT );  
}
```

```
void loop() {  
  for( int i = 0; i < 255; i=i+5){  
    analogWrite(DC_MOTOR_PIN, i);  
    delay(50);  
  }
```

```
  for( int i = 255; i > 0; i=i-5 ){  
    analogWrite(DC_MOTOR_PIN, i);  
    delay(50);  
  }
```

```
}
```

## SISTEMA DI CONTROLLO ON-OFF

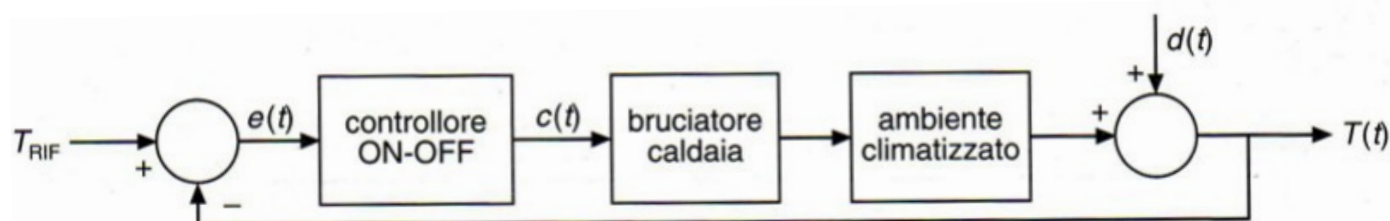
Il controllo ON-OFF è un controllo ad anello chiuso nel quale l'azione del controllore è discontinua.

Il controllore decide quando intervenire in base alla misura dello scostamento tra valore atteso e valore reale dell'uscita come nel controllo continuo, con la differenza che l'aggiustamento non viene applicato con continuità bensì quando lo scostamento oltrepassa una soglia predeterminata.

Prendiamo come esempio il controllo ON-OFF di temperatura per la climatizzazione di un ambiente, dove lo scopo del controllo è quello di mantenere il livello di temperatura di una stanza entro margini prestabiliti rappresentati da una soglia inferiore  $T_{inf}$  e una superiore  $T_{sup}$ .

Il valore della variabile in uscita  $T(t)$  viene confrontato con il valore desiderato  $T_{rif}$ .

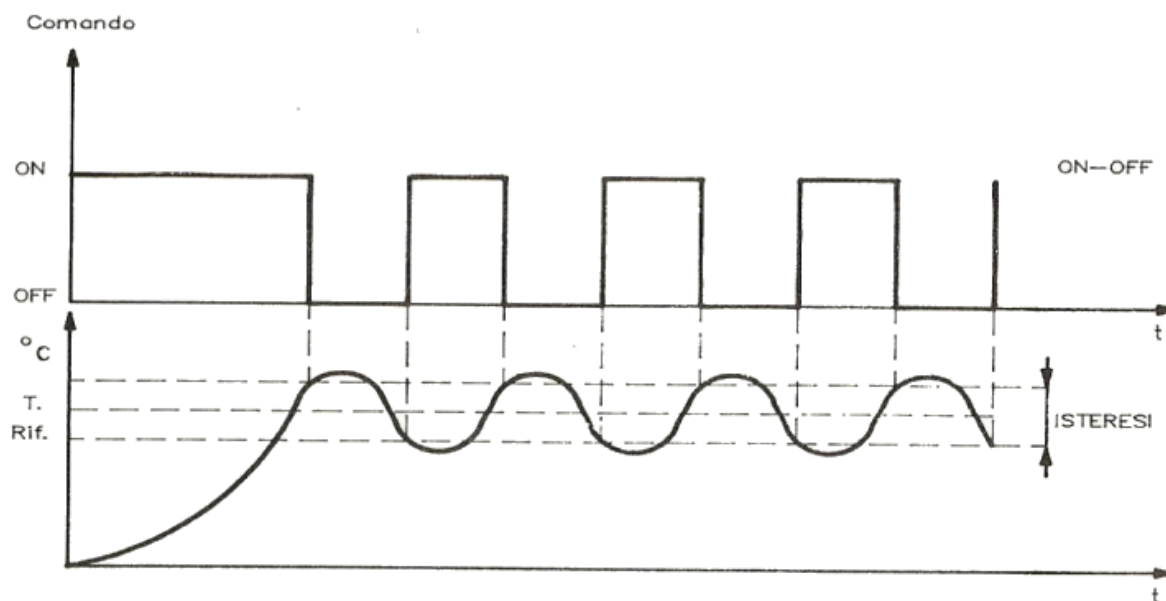
La differenza tra i due valori (errore "e") viene utilizzata per pilotare il controllore che interviene in modalità "tutto o niente" comandando con la sua uscita l'attivazione (ON) o la disattivazione (OFF) del bruciatore della caldaia.



Altro esempio è la cisterna d'acqua, dove lo scopo è mantenere il livello del liquido entro margini prefissati.

Quando il livello risulta inferiore alla soglia minima, il sistema interviene comandando l'apertura di un'elettrovalvola (rubinetto controllabile elettricamente) che rimane aperta (stato "ON") fin quando il livello sale e supera la soglia massima.

L'attuatore viene acceso e spento (mediante relè o transistor) sulla base del valore misurato della grandezza controllata con un'oscillazione definita "isteresi" (es.  $\pm 1^\circ\text{C}$ ).



## SISTEMA DI CONTROLLO PID (PROPORZIONALE – INTEGRALE – DERIVATIVO)

È un sistema in retroazione negativa ampiamente impiegato nei sistemi di controllo automatico.

È molto comune nell'industria, in particolare nella versione PI (senza azione derivativa).

Grazie a un input che determina il valore attuale, è in grado di reagire a un eventuale errore positivo o negativo tendendo verso il valore 0.

Il controllore acquisisce in ingresso un valore da un processo e lo confronta con un valore di riferimento.

La differenza, il cosiddetto segnale di errore, viene quindi usata per determinare il valore della variabile di uscita del controllore, che è la variabile manipolabile del processo.

Il PID regola l'uscita in base a:

- il valore del segnale di errore (azione proporzionale → coefficiente  $K_p$ );
- i valori passati del segnale di errore (azione integrale → coefficiente  $K_i$ );
- quanto velocemente il segnale di errore varia nel tempo (azione derivativa → coefficiente  $K_d$ ).

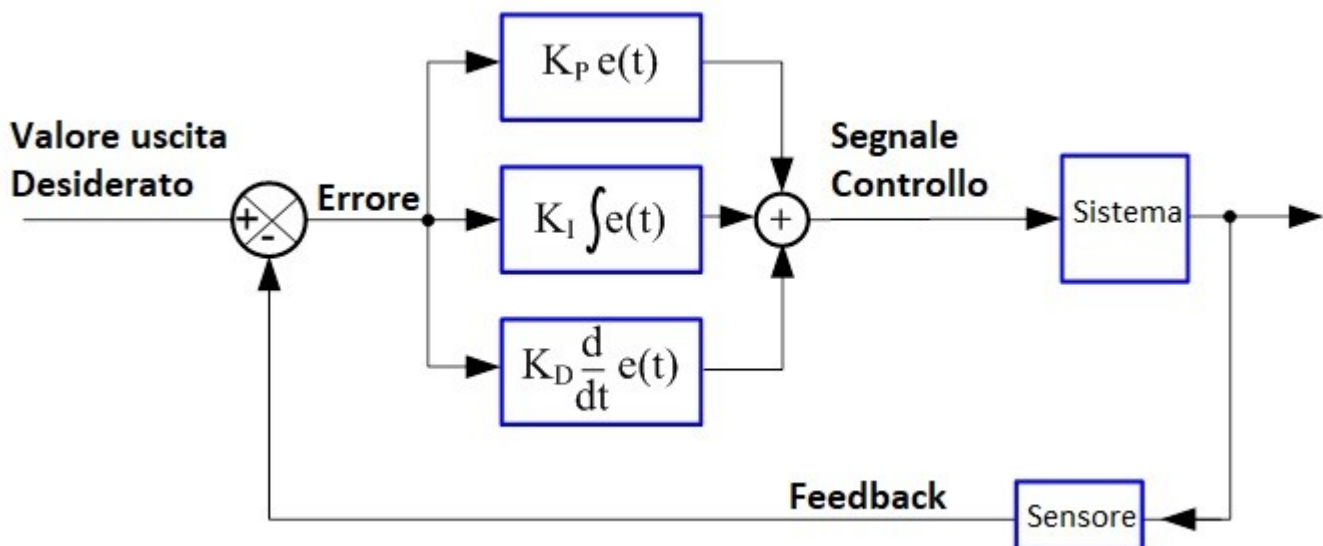
I controllori PID sono relativamente semplici da comprendere, installare e tarare, al confronto con più complessi algoritmi di controllo basati sulla teoria del controllo ottimo e del controllo robusto.

La taratura dei parametri avviene di solito attraverso semplici regole empiriche, come i metodi di Ziegler-Nichols, che risultano in controllori stabilizzanti di buone prestazioni per la maggior parte dei processi.

Molto spesso l'azione derivativa viene rimossa, risultando nel comunissimo controllore PI.

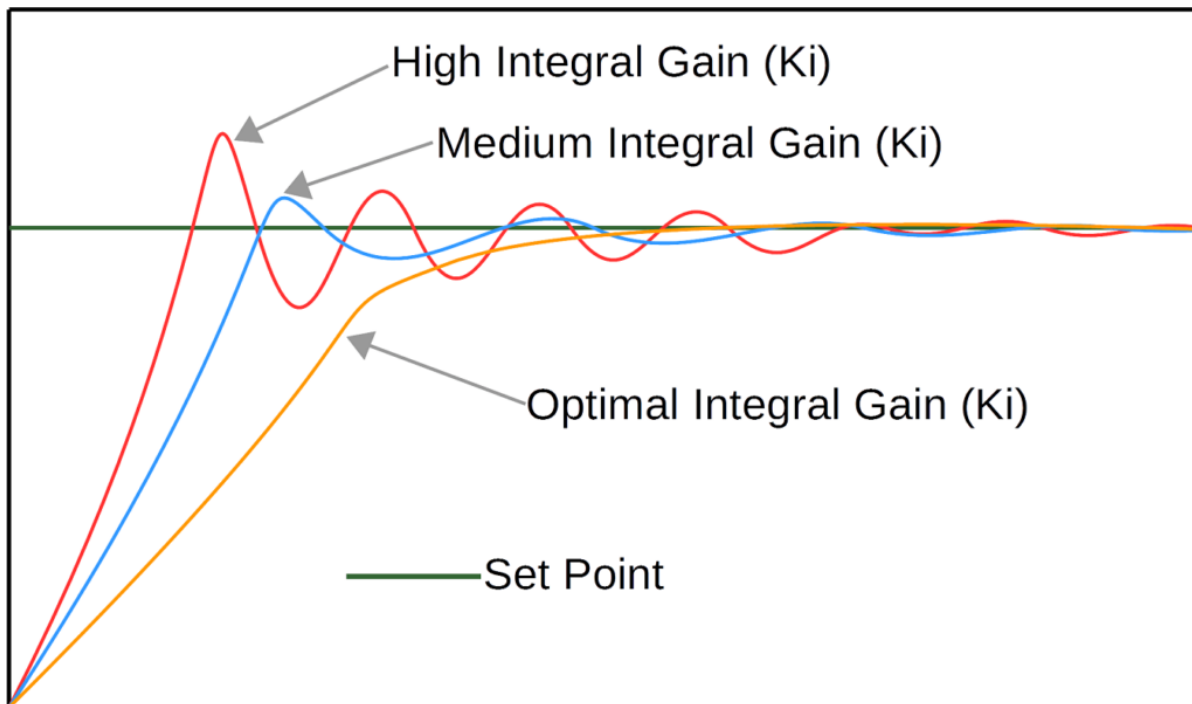
I controllori PID sono spesso sufficienti a controllare processi industriali anche complessi, ma la loro semplicità risulta in una serie di limiti che è bene tener presente:

- Non sono in grado di adattarsi a cambiamenti nei parametri del processo;
- Non sono stabili, a causa della presenza dell'azione integrale;
- Alcune regole di taratura, come quelle di Ziegler-Nichols, reagiscono male in alcune condizioni;
- Sono intrinsecamente monovariabili, non possono quindi essere usati in sistemi inerentemente multi variabili



Segnale controllo =  $K_p * e(t) + K_i * \int e(t) dt + K_d * de(t)/dt$  (in genere una tensione ...)

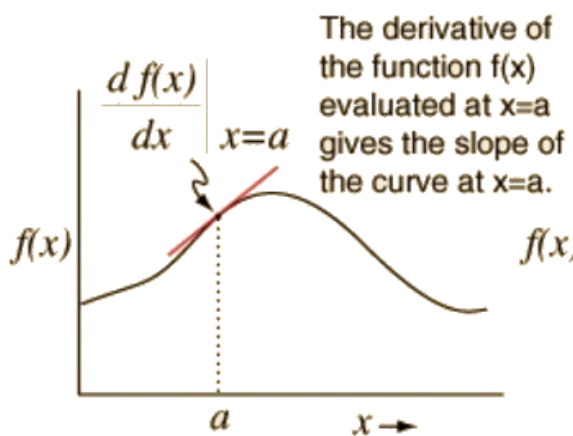
La scelta dei parametri  $K_p$ ,  $K_i$  e  $K_d$  è fondamentale per ottenere il risultato desiderato. Valore non corretti possono rendere il sistema instabile.



**IMPLEMENTAZIONE NUMERICA PID**

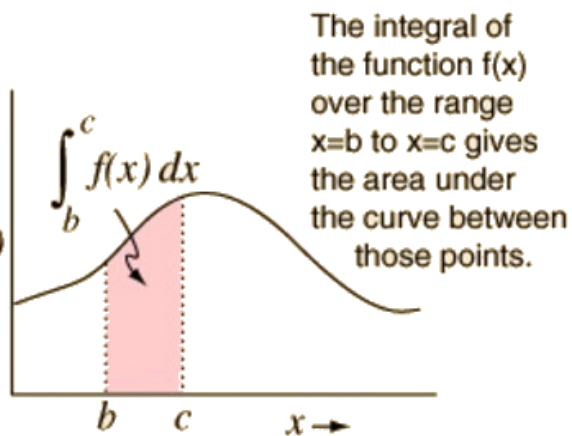
*Derivative*

$$\frac{df(x)}{dx}$$



*Integral*

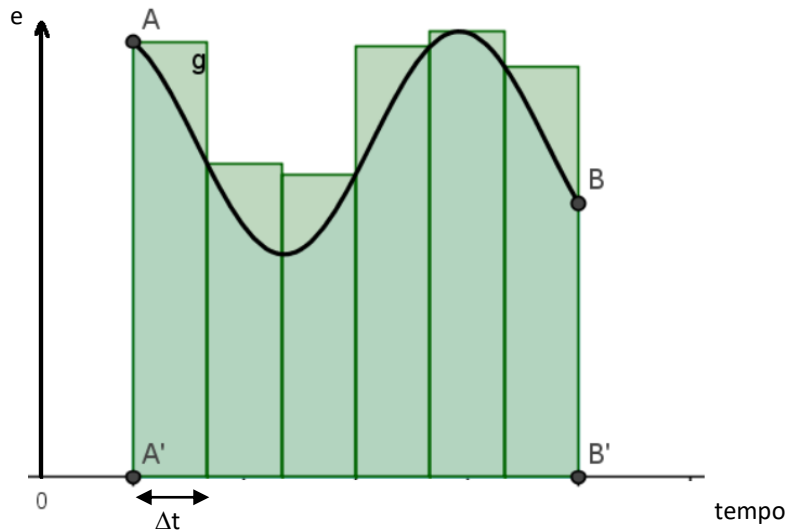
$$\int f(x) dx$$



Derivata dell'errore → variazione dell'errore nell'intervallo di tempo →  $D = \Delta e / \Delta t$

Integrale dell'errore → somma aree →  $I = \sum \Delta e * \Delta t$

## INTEGRAZIONE NUMERICA DELL'ERRORE (METODO DEI RETTANGOLI)



Sommando le aree dei rettangoli ottenuto col prodotto  $e^* \Delta t$  andiamo ad approssimare l'integrale della curva  $e(t)$ .

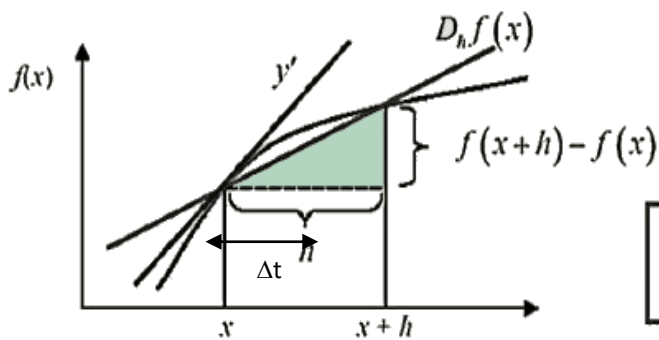
$$\int e(t) dt = \sum e^* \Delta t$$

più il  $\Delta t$  è piccolo e più è preciso il calcolo dell'integrale

## DERIVAZIONE NUMERICA DELL'ERRORE

La derivata puntuale di una funzione si può approssimare con il suo rapporto incrementale fissando un opportuno  $\Delta t$

$$y' = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad \rightarrow \quad y' \approx \frac{f(x+h) - f(x)}{h} = D_h f(x)$$



$D_h f(x)$   
derivata numerica di  $f(x)$

$$e'(t) = \Delta e / \Delta t$$

più il  $\Delta t$  è piccolo e più è preciso il calcolo della derivata

## TEMPO DI CAMPIONAMENTO DI UN SISTEMA AUTOMATICO

In un sistema di controllo automatico, l'**intervallo di campionamento** (spesso indicato con  $T_s$ , dall'inglese *Sampling Time*) è il tempo che intercorre tra due acquisizioni consecutive dei dati da parte del controllore digitale.

Esiste una relazione inversa tra l'intervallo di campionamento e la frequenza di campionamento  $f_s$ :

$$f_s = 1/T_s \text{ (Hz)}$$

- Se  $T_s = 0.1 \text{ s}$ , allora  $f_s = 10 \text{ Hz}$  (il sistema campiona 10 volte al secondo).
- Se  $T_s = 1 \text{ ms}$ , allora  $f_s = 1000 \text{ Hz}$  (il sistema campiona 1000 volte al secondo).

Scegliere l'intervallo di campionamento  $T_s$  è un atto di bilanciamento tra precisione del controllo e risorse computazionali. Se si campiona troppo lentamente, si perdono i dettagli (e il controllo diventa instabile); se si campiona troppo velocemente, si rischia di saturare la CPU o di avere problemi di precisione numerica.

Ecco una guida pratica in campo industriale.

Settore / Tipo di Processo	Tempo di Campionamento $T_s$	Note
Processi Termici / Chimici	1 s – 30 s (o più)	Sistemi molto lenti. Scaldare una vasca da 1000 litri richiede minuti; campionare ogni millisecondo è inutile.
Controllo di Livello / Pressione	100 ms – 1 s	Dinamiche medie. Le valvole regolatrici hanno tempi di attuazione meccanica limitati.
Automazione di Fabbrica (PLC)	10 ms – 50 ms	Logica sequenziale, nastri trasportatori, sensori ottici standard.
Controllo di Movimento (Assi/Robot)	1 ms – 5 ms	Controllo di posizione di motori brushless. Qui la velocità è critica per la precisione millimetrica.
Azionamenti Elettrici (Inverter)	100 $\mu\text{s}$ – 500 $\mu\text{s}$	Controllo di corrente e velocità del motore. Le costanti di tempo elettriche sono molto piccole.
Elettronica di Potenza	< 50 $\mu\text{s}$	Gestione della commutazione dei transistor (PWM). Richiede hardware dedicato come FPGA o DSP.

## COSA SUCCEDA CON UN TEMPO DI CAMPIONAMENTO ERRATO?

### **Campionamento troppo lento (Aliasing e Instabilità)**

Se  $T_s$  è troppo grande, il controllore "vede" il sistema in ritardo.

Questo introduce uno sfasamento portando a oscillazioni o, nel peggiore dei casi, all'instabilità del sistema.

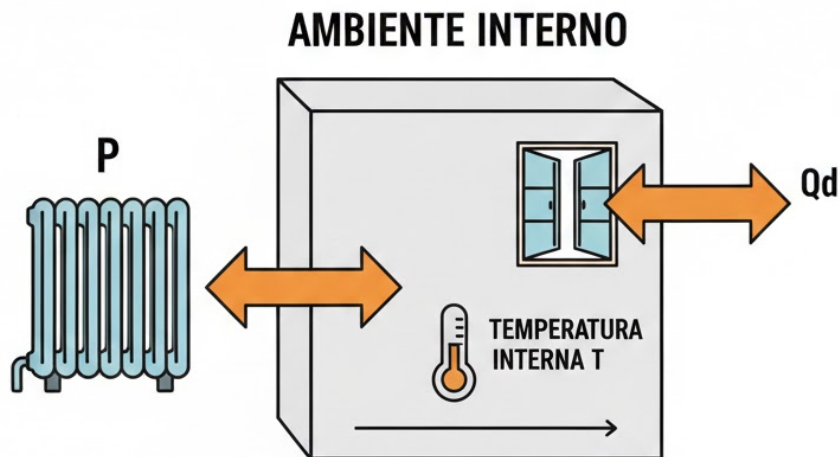
### **Campionamento troppo veloce (Numerical Issues)**

Differenziazione rumorosa: calcolare la derivata (azione D del PID) su segnali campionati troppo velocemente amplifica il rumore di misura.

Errori di arrotondamento: se  $T_s$  è piccolissimo, i coefficienti dei filtri digitali diventano vicini all'unità, e la precisione finita della CPU (floating point) può causare errori numerici significativi.

Tramite una funzione assegnata è possibile valutare la temperatura interna di un locale riscaldato tramite una potenza P.

## FLUSSO TERMICO NETTO = $P - Q_d$



La funzione viene chiamata ad intervalli regolari di 50ms per simulare il comportamento reale della temperatura nel tempo.

Utilizzando piccoli (infinitesimi) intervalli di tempo si approssima in modo efficace la realtà.

Il calcolo della temperatura è basato sul bilancio di energia del locale:

$$(P - Q_d) dt = m c_t dT$$

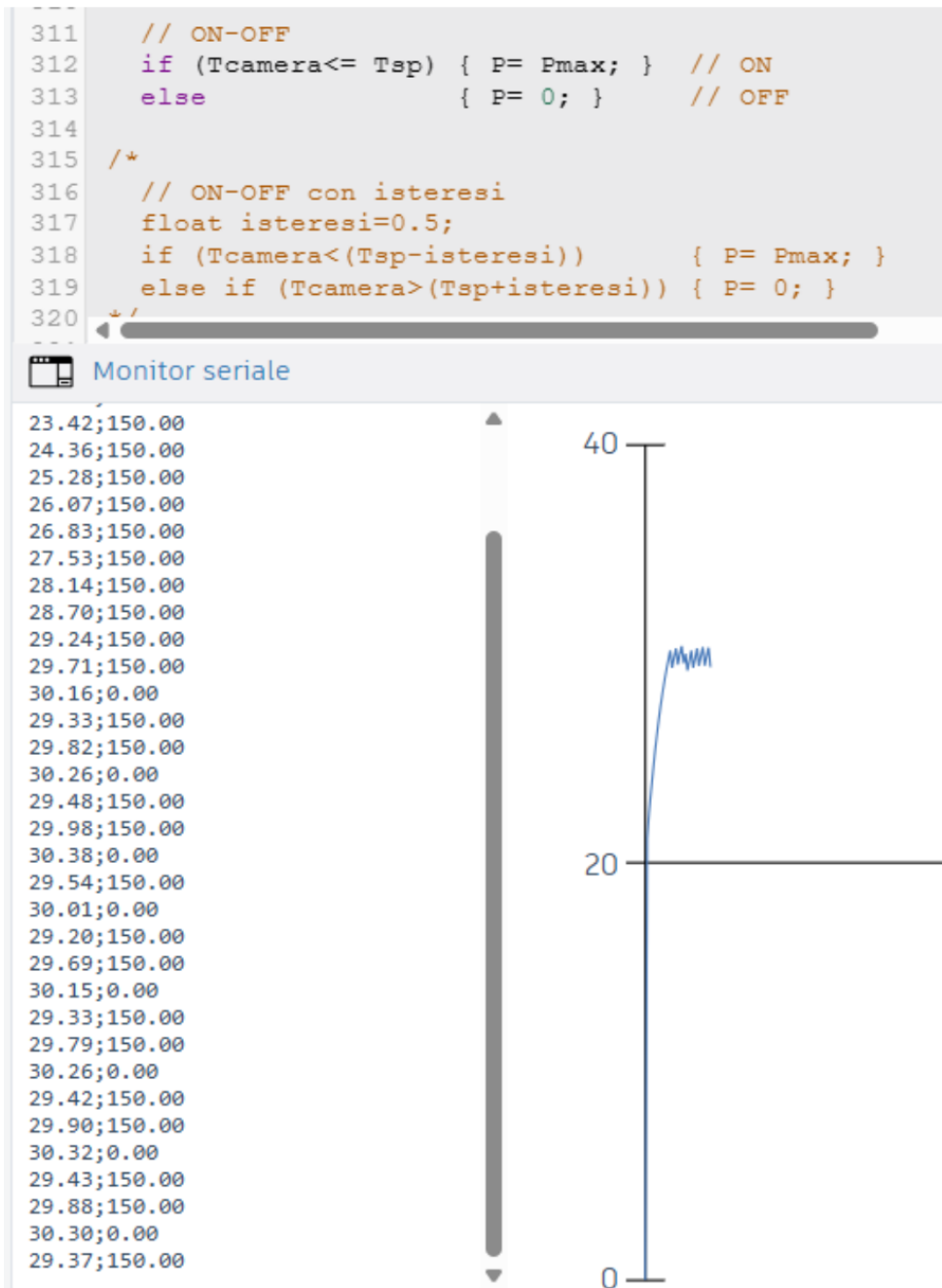
con

$$Q_d = \text{dispersioni termiche} = U A dT$$

```
// Simula la T(°C) in un locale riscaldato con una P(watt)
// che disperde calore verso l'ambiente tramite le pareti
float Camera(float Pel) {
  float U = 5;
  float Ct = 1006; // J/Kg K
  float A = 2; // m2 area for convection
  float massa = 0.1; // Kg
  float Tamb = 20; // °C
  static float T = Tamb; // °C
  static long last = 0; // static --> mantiene valore fra le varie chiamate
  int dt = 50; // ms
  // ogni 50ms calcola la temperatura del locale
  if (millis() - last >= dt) {
    last = millis();
    float Qd = U * A * (T - Tamb); // dispersioni
    T = T + (Pel - Qd) * (dt / 1000.0) / (massa * Ct);
  }
  return T;
}
```

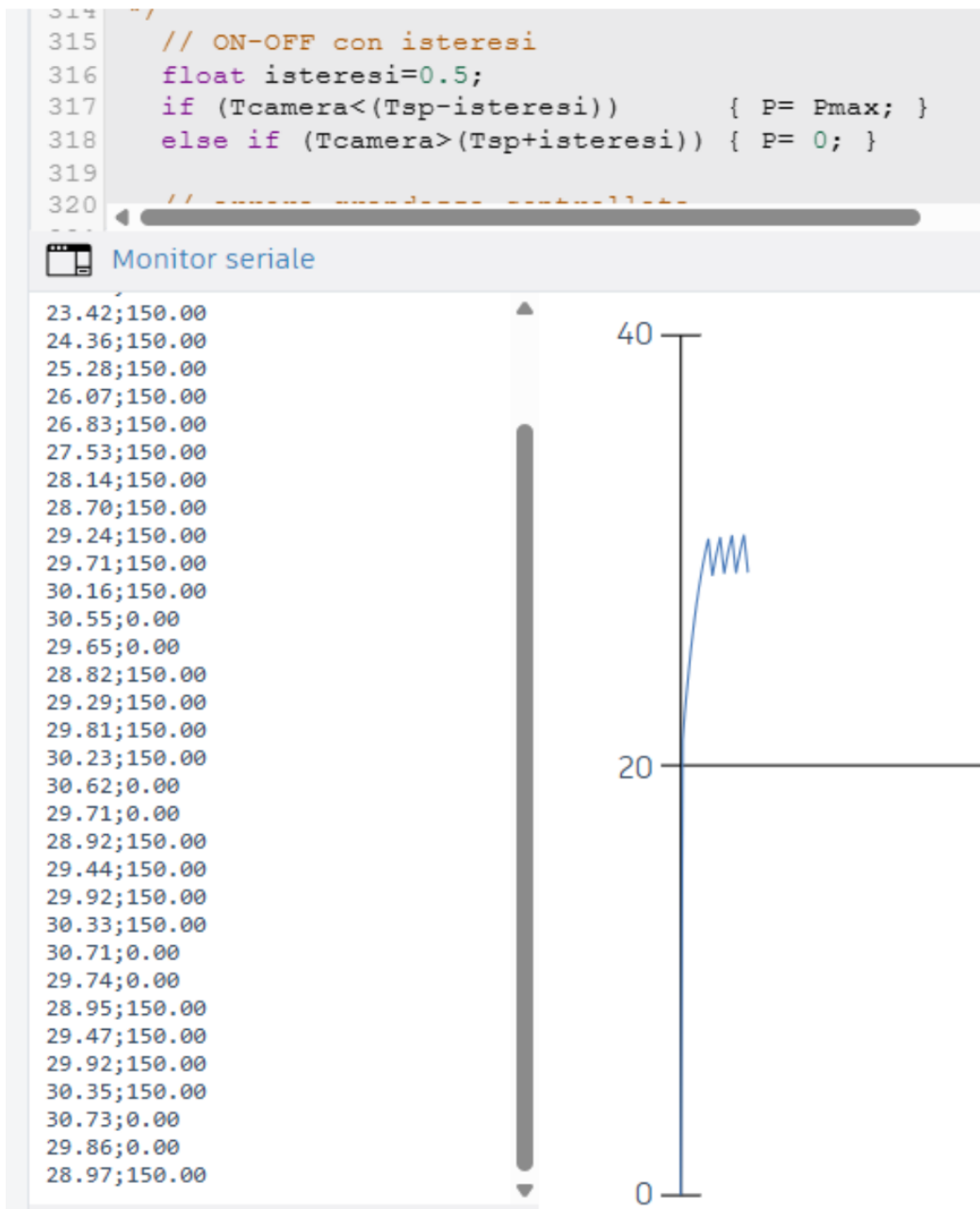
## CONTROLLO ON-OFF

Questo controllo accende/spegne intorno al SETPOINT con alta frequenza. L'errore a regime è di circa 0.5°C



## CONTROLLO ON-OFF CON ISTERESI

Con l'isteresi la frequenza di accensione e spegnimento si riduce con un errore di circa 1°C





## CONTROLLO PROPORZIONALE-INTEGRALE PI

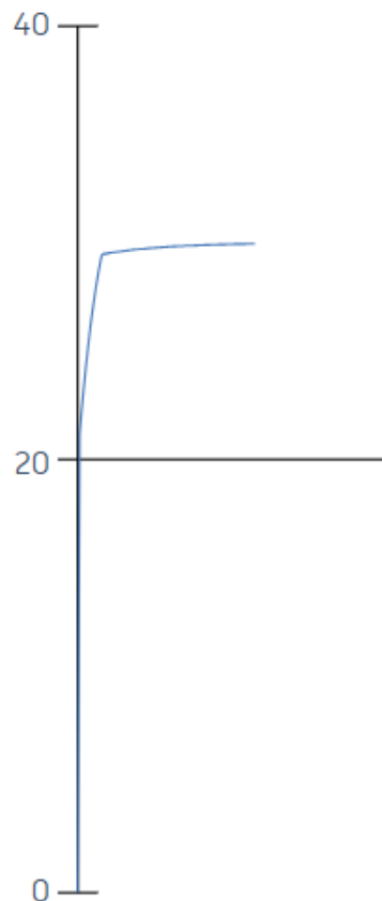
Con questo controllo a regime l'errore si riduce a 0.

E' necessario trovare i coefficienti  $K_p$  e  $K_i$  specifici per il sistema controllato prestando attenzione a non rendere instabile il controllo. A regime la potenza applicata è una frazione di quella massima (attuatore regolabile).

```
334 float errore= abs(Tsp - Tcamera);
335
336 // PROPORZIONALE + INTEGRALE
337 if (errore>2) {
338     P= Pmax; // max. fino a -2°C dal SP
339 }
340 else {
341     integral= integral + errore * dti;
342     P= Kp*errore+ Ki* integral; // controllo PI.
343     if (P>Pmax) P= Pmax;
344 }
345
346
```

### Monitor seriale

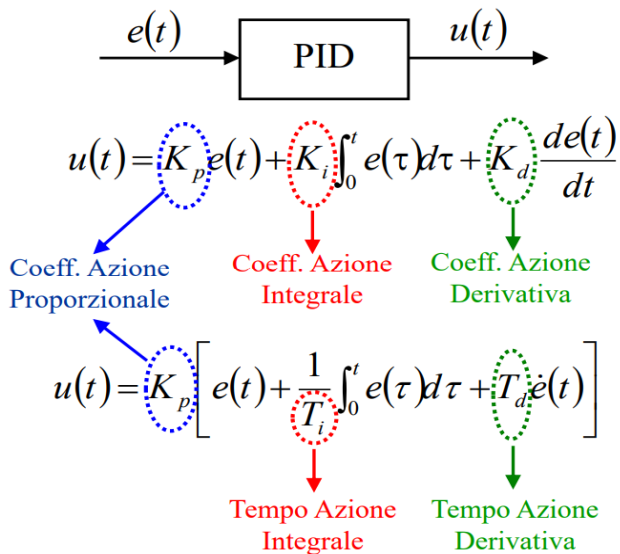
```
29.87;99.21
29.88;99.25
29.88;99.30
29.89;99.28
29.89;99.35
29.89;99.32
29.90;99.39
29.90;99.37
29.90;99.43
29.91;99.41
29.91;99.44
29.91;99.47
29.91;99.46
29.92;99.51
29.92;99.49
29.92;99.54
29.92;99.52
29.93;99.55
29.93;99.58
29.93;99.57
29.93;99.59
29.94;99.60
29.94;99.63
29.94;99.62
29.94;99.65
29.94;99.64
29.94;99.67
29.95;99.66
29.95;99.68
29.95;99.70
29.95;99.69
29.95;99.72
```



## REGOLE DI ZIEGLER-NICHOLS

Il metodo di Ziegler-Nichols, risalente al 1942, è tra i più usati ed è apprezzato per la sua semplicità, per il fatto di non richiedere un modello matematico del processo e per le prestazioni che riesce a produrre. Serve a trovare il cosiddetto "guadagno critico  $K_u$ ", dal quale si deriveranno gli altri parametri del PID:

- Il processo viene fatto controllare da un controllore esclusivamente proporzionale ( $K_i$  e  $K_d$  vengono impostati a zero);
- Il guadagno  $K_p$  del controllore proporzionale viene gradualmente aumentato;
- Il guadagno critico  $K_u$  è il valore del guadagno per cui la variabile controllata presenta oscillazioni sostenute, cioè che non spariscono dopo un transitorio (questa è una misura dell'effetto dei ritardi e della dinamica del processo);
- Si registra il periodo critico  $P_u$  delle oscillazioni sostenute e con la tabella allegata si determinano le costanti per il controllore P, PI o PID;

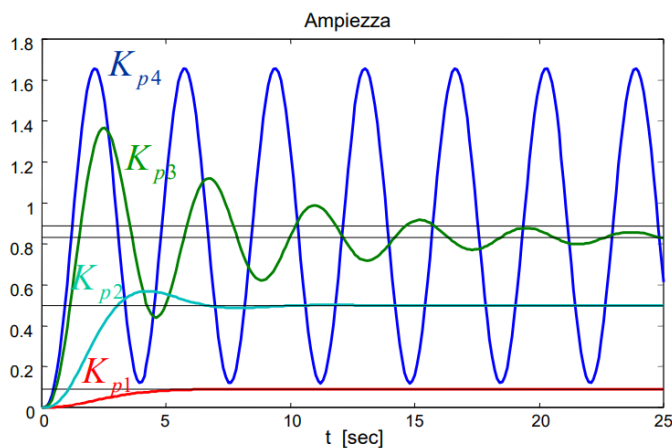


Tipo	$K_p$	$T_i$	$T_d$
<b>P</b>	$0,50K_u$	-	-
<b>PI</b>	$0,45K_u$	$P_u/1,2$	-
<b>PID</b>	$0,60K_u$	$P_u/2$	$P_u/8$

$$T_i = \frac{K_p}{K_i}$$

$$T_d = \frac{K_d}{K_p}$$

### ESEMPIO CALCOLO COSTANTI PID



$$K_{p1} = 0.1$$

$$K_{p2} = 1$$

$$K_{p3} = 5$$

$$K_{p4} = 8 \leftarrow \bar{K}_p$$

$$\bar{T} \cong 3.6$$

$$\text{PID "ideale"} \begin{cases} K_p = 0.6\bar{K}_p = 4.8 \\ T_i = 0.5\bar{T} = 1.8138 \\ T_d = 0.125\bar{T} = 0.4534 \end{cases}$$



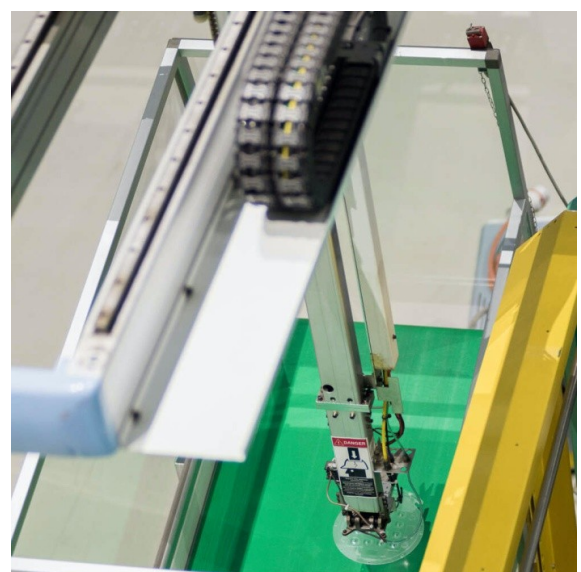
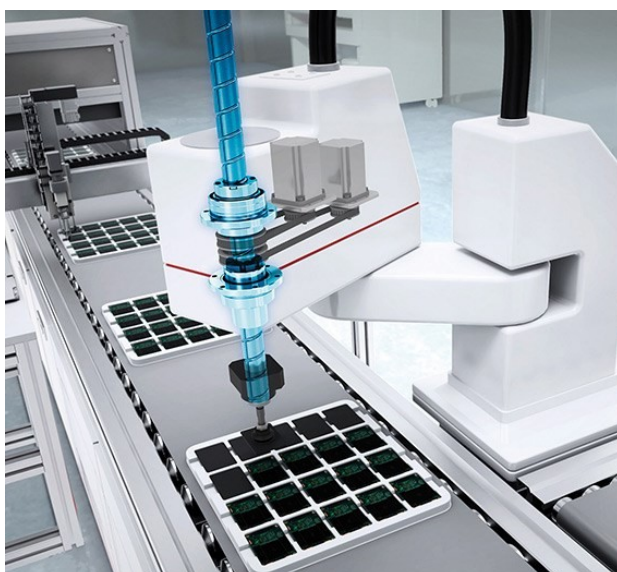
# ROBOTICA INDUSTRIALE

La robotica industriale è un settore sviluppato negli ultimi anni, che vede l'utilizzo di sistemi automatici come parte integrante del lavoro industriale. Nella robotica industriale, un sistema di automazione dunque, sostituisce un uomo nella catena di montaggio, compiendo sempre lo stesso lavoro ad un ritmo costante e frenetico.

Tutti gli strumenti meccanici progettati per compiere un determinato lavoro in autonomia rientrano a far parte della robotica industriale. Il sistema automatico nella fattispecie viene chiamato robot.

## ISO 8373

Esiste poi un'altra definizione che risale all'ISO 8373, che generalmente rappresenta il vocabolario per la materia di robotica industriale e dei robot implementati in questo settore. La definizione in questione afferma che un robot è un sistema con controllo automatico, riprogrammabile e multi-funzione, che può essere sia fisso a terra sia mobile, di tre o più assi ed il suo scopo è quello di essere utilizzato per operazioni di automazione industriale.

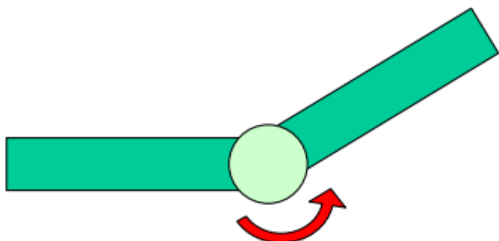


## SISTEMI ROBOTICI

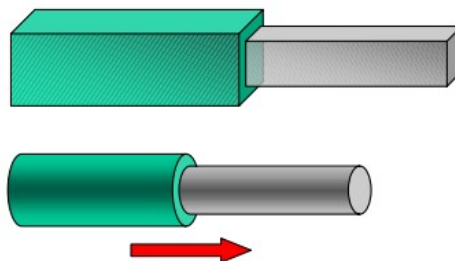
I sistemi robotici sono sistemi costituiti da un insieme di corpi rigidi (link) connessi mediante giunti rotazionali (R) o prismatici (P). Il numero di giunti indipendenti definisce i gradi di libertà del robot DOF (degrees of freedom).

## TIPI DI GIUNTO

Le tipologie principali di giunti sono due: rotazionali "R" e prismatici "P".



R = rotazionale

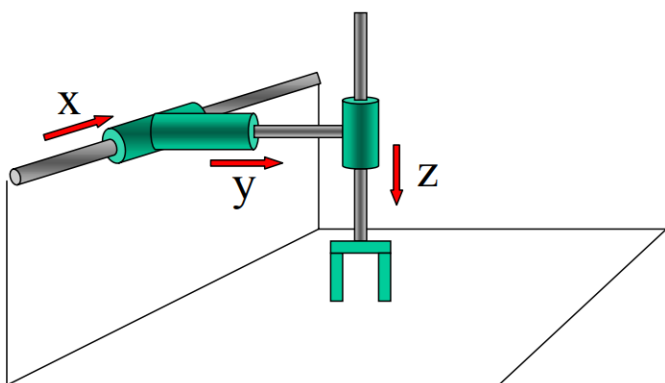


P = prismatico

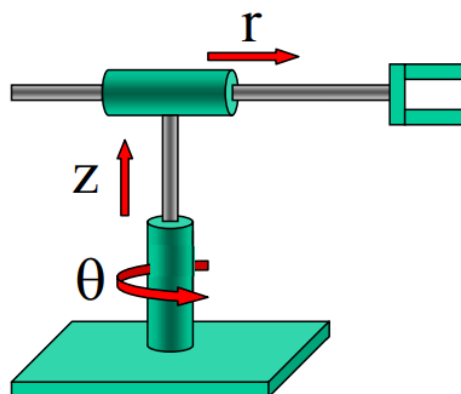
## TIPI DI ROBOT

Le tipologie principali di robot si differenziano in base al tipo di giunti adottato.

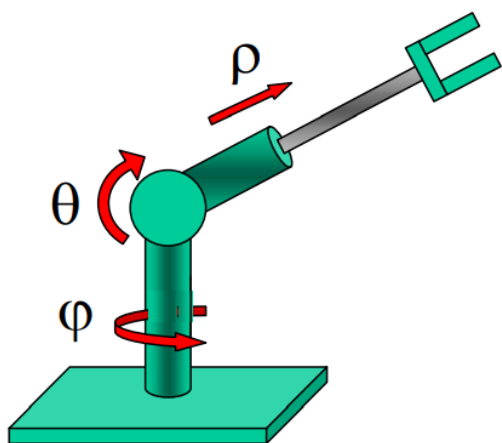
### CARTESIANO PPP



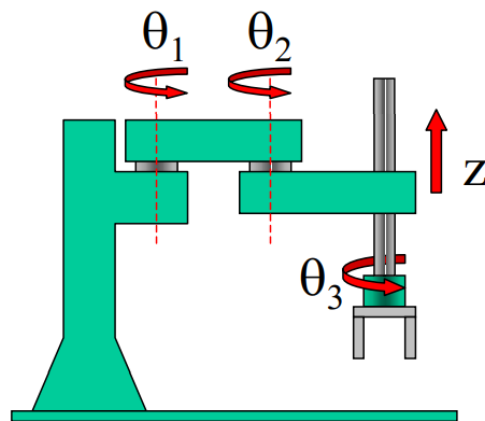
### CILINDRICO RPP



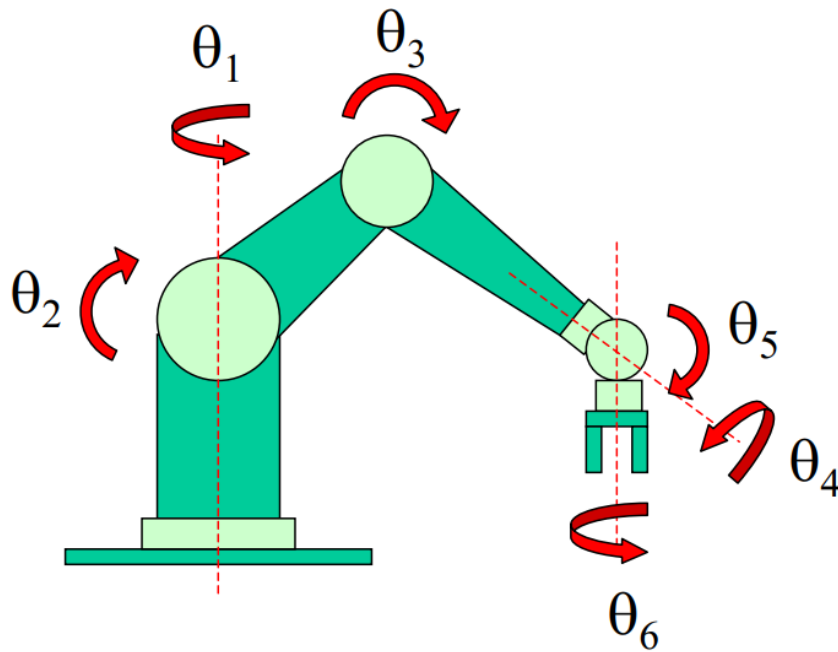
### POLARE RRP



### SCARA RRP



## ANTROPOMORFO RRR



## VIDEO

<https://www.youtube.com/@automazioneindustriale/videos>

[https://www.youtube.com/watch?v=PV8\\_7lgl7PE](https://www.youtube.com/watch?v=PV8_7lgl7PE)

<https://www.youtube.com/watch?v=u2BfVmmaWvk>

<https://www.youtube.com/watch?v=BLR5FoFBtal>

<https://www.youtube.com/shorts/x1fXdptiuSU>

<https://www.youtube.com/shorts/srx8h6kHS0Y>

<https://youtube.com/shorts/0Up7F0xpLy4?si=OEKDDoJUDGQxiDNJ>

<https://www.youtube.com/watch?v=UkZc3llwfSM&t=2s>


## ROBOT COLLABORATIVI (COBOT)

Robot collaborative (collaborative robot) o più semplicemente cobot.

I cobot aiutano ogni giorno gli operatori in tutte le attività pesanti o di precisione, portando nelle filiere qualità, velocità, sicurezza ed efficienza. Il tutto in un'unica soluzione.



### INDUSTRIAL ROBOTS

- Installazione difficile 
- Alte competenze a livello di programmazione 
- Installazioni fisse 
- Disponibilità di ampi spazi 
- Barriera di sicurezza necessaria 
- Elevati costi addizionali 

V S.

### COLLABORATIVE ROBOTS

-  Installazione facile
-  Possono essere programmati da chiunque
-  Distribuzione flessibile
-  Requisiti di spazio ridotti
-  Collaborano fianco a fianco con gli operatori umani
-  Economicamente convenienti e con un ROI molto rapido

## CINEMATICA DEL ROBOT

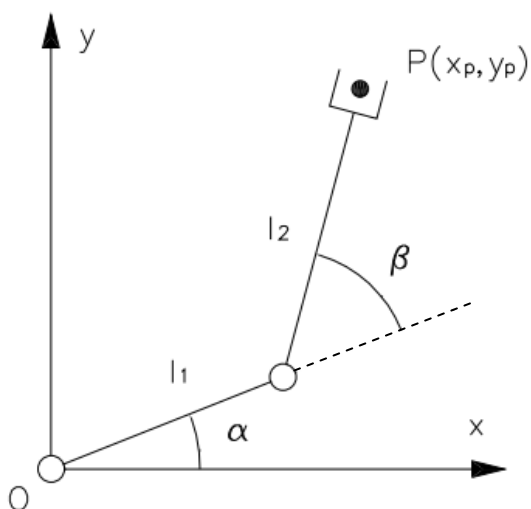
La cinematica del robot è lo studio del suo movimento prescindendo dalle cause che lo generano.

Il robot viene visto come una catena di corpi rigidi, dalla base all'end effector, connessi da giunti che consentono un singolo grado di libertà. La conoscenza del modello cinematico del robot è essenziale nei problemi di pianificazione del moto e controllo

## CINEMATICA DIRETTA DEL ROBOT PLANARE A 2 LINK

La cinematica diretta, noti gli angoli dei link 1 e link 2, permette di ricavare la posizione finale  $P(x_p, y_p)$  della pinza.

Gli angoli si misurano come indicato in figura e sono positive se in senso antiorario e neagativi in senso orario.



Da semplici considerazioni geometriche:

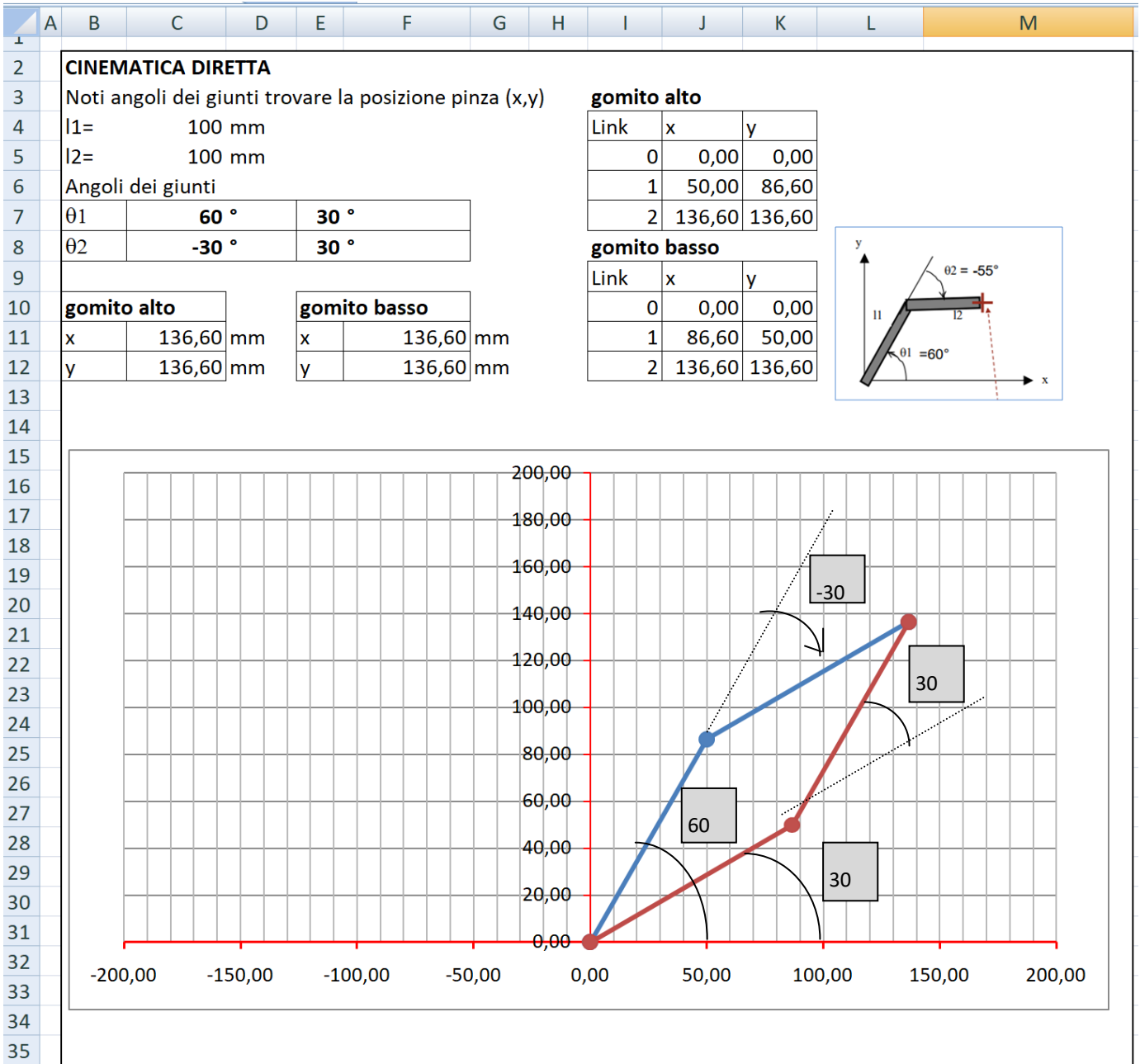
$$\begin{cases} x_p = l_1 \cos \alpha + l_2 \cos(\alpha + \beta) \\ y_p = l_1 \sin \alpha + l_2 \sin(\alpha + \beta) \end{cases}$$

## FOGLIO DI CALCOLO

Il problema presenta due possibili soluzioni dette a "gomito alto" e a "gomito basso".

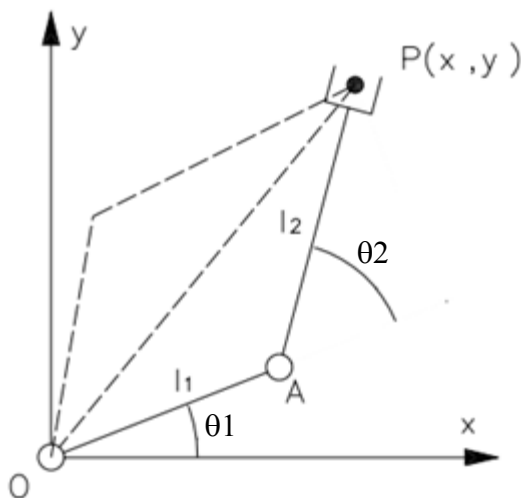
$$x = l_1 \cos(\alpha) + l_2 \cos(\alpha + \beta)$$

$$y = l_1 \sin(\alpha) + l_2 \sin(\alpha + \beta)$$



# CINEMATICA INVERSA DEL ROBOT DEL ROBOT PLANARE

Nota la posizione P(x,y) che si vuole raggiungere si devono ricavare gli angoli necessari.



$$\theta_2 = \arccos\left[\frac{(x^2 + y^2 - l_1^2 - l_2^2)}{2 \cdot l_1 \cdot l_2}\right]$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_2 \cdot S_2}{l_1 + l_2 \cdot C_2}\right)$$

S2 = sen θ2      C2 = cos θ2

## FOGLIO DI CALCOLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	<b>CINEMATICA DIRETTA</b>													
2	Noti gli angoli dei giunti trovare la posizione del polso (x,y)													
3	l1=	10 CM												
4	l2=	5 CM												
5				ALTO	x	y								
6	θ1	53,1 °	0	0,00	0,00									
7	θ2	-90 °	1	6,00	8,00									
8														
9	x	10,00												
10	y	5,0												
11														
12														
13														
14														
15														
16	<b>CINEMATICA INVERSA</b>													
17	Nota la posizione (x,y) del polso trovare gli angoli dei giunti													
18	l1	10 cm												
19	l2	5 cm												
20				ALTO	x	y								
21	x	10,00 cm	0	0,00	0,00									
22	y	5,00 cm	1	6,00	8,00									
23														
24	<b>ANGOLI GOMITO ALTO</b>													
25	θ2	-90 °												
26	θ1	53,1 °												
27														
28														
29	<b>Formule Excel</b>													
30	θ2	=GRADI(-ARCCOS((B21^2+B22^2-B18^2-B19^2)/(2*B19*B18)))												
31	θ1	=GRADI(ARCTAN(B22/B21))-GRADI(ARCTAN(B19*SEN(RADIANTI(B25)))/(B18+B19*COS(RADIANTI(B25))))												

### Cinematica Inversa Robot Planare

$l1 =$	100,0	mm
$l2 =$	90,0	mm
$x_A =$	190,0	
$y_A =$	0,0	

$l1 =$	100,0	mm
$l2 =$	90,0	mm
$x_B =$	120,0	
$y_B =$	30,0	

$l1 =$	100,0	mm
$l2 =$	90,0	mm
$x_C =$	100,0	
$y_C =$	100,0	

$\cos\theta_2 =$	1,0
$\theta_2 =$	0,0
$\sin\theta_2 =$	0,0
$\theta_1 =$	0,0
$x_1 =$	100,0
$y_1 =$	0,0

$\cos\theta_2 =$	-0,156
$\theta_2 =$	-98,9
$\sin\theta_2 =$	-0,988
$\theta_1 =$	60,0
$x_1 =$	50,0
$y_1 =$	86,6

$\cos\theta_2 =$	0,106
$\theta_2 =$	-83,9
$\sin\theta_2 =$	-0,994
$\theta_1 =$	84,3
$x_1 =$	10,0
$y_1 =$	99,5

A	x	y
0	0,0	0,0
1	100,0	0,0
2	190,0	0,0

B	x	y
0	0,0	0,0
1	50,0	86,6
2	120,0	30,0

C	x	y
0	0,0	0,0
1	10,0	99,5
2	100,0	100,0

A Gomito alto

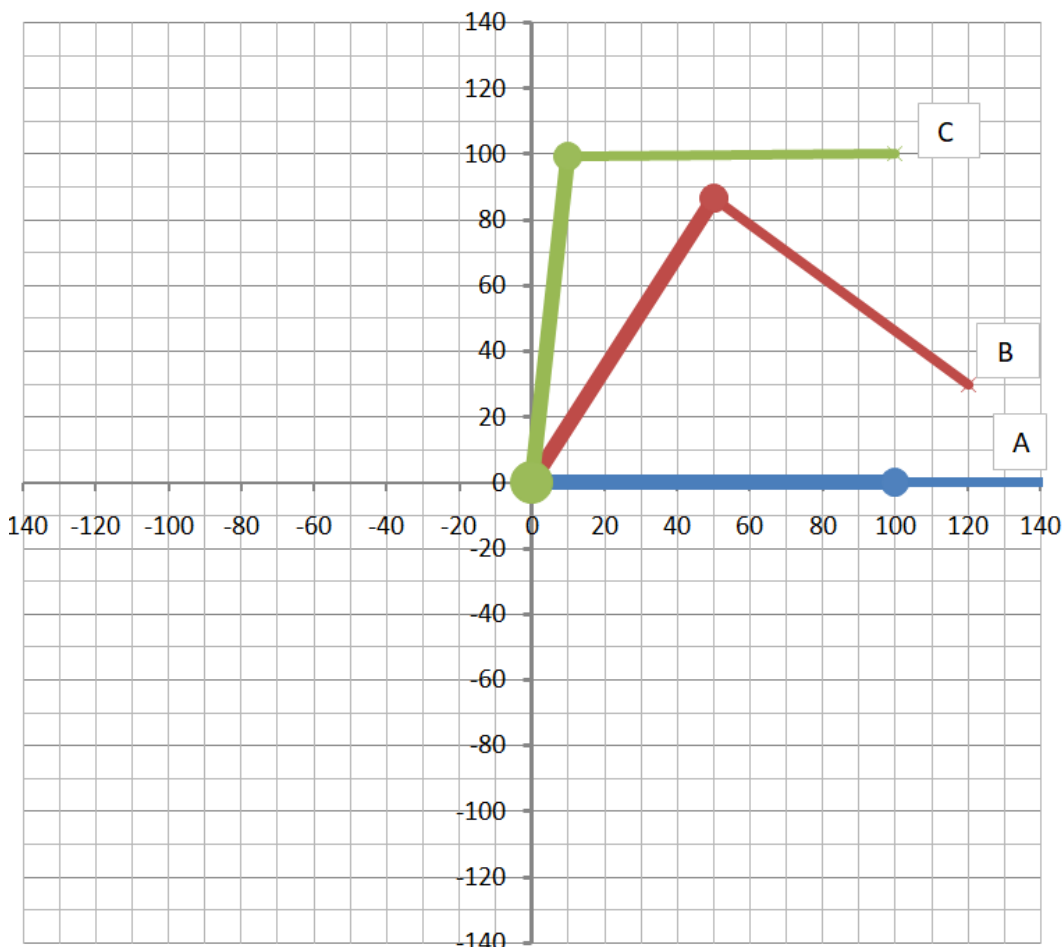
$\theta_1$	0,0
$\theta_2$	0,0

B Gomito alto

$\theta_1$	60,0
$\theta_2$	-98,9

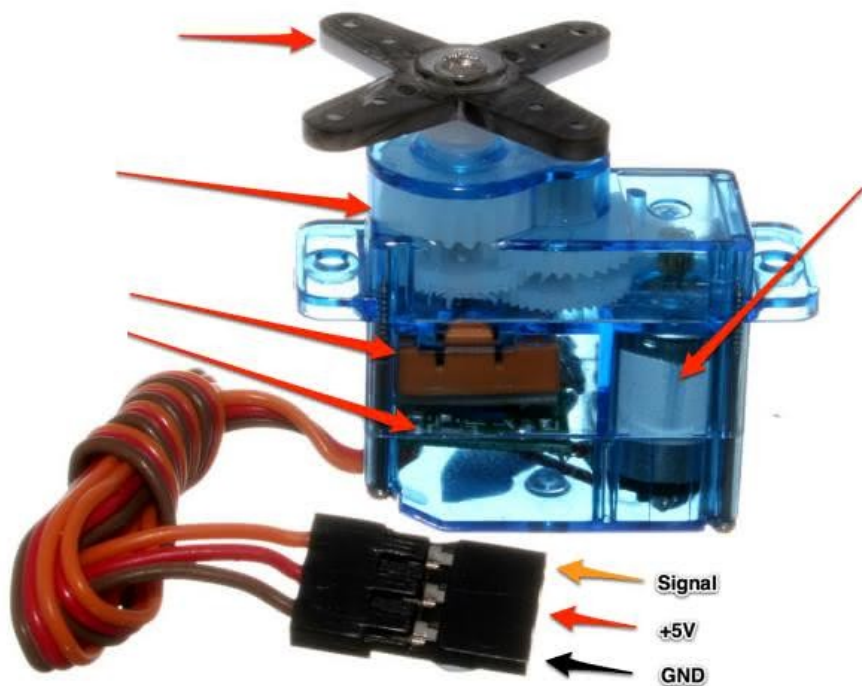
C Gomito alto

$\theta_1$	84,3
$\theta_2$	-83,9



## SERVO SG90

A differenza di un motore convenzionale, la maggior parte dei servomotori di base può ruotare solo tra 0 e +180 gradi. Sono costituiti da un motore a corrente continua, vari ingranaggi per aumentare la coppia (e ridurre la velocità) e un sistema servoassistito



Peso: 9G

Coppia a 4,8V: 1,8 kg/cm

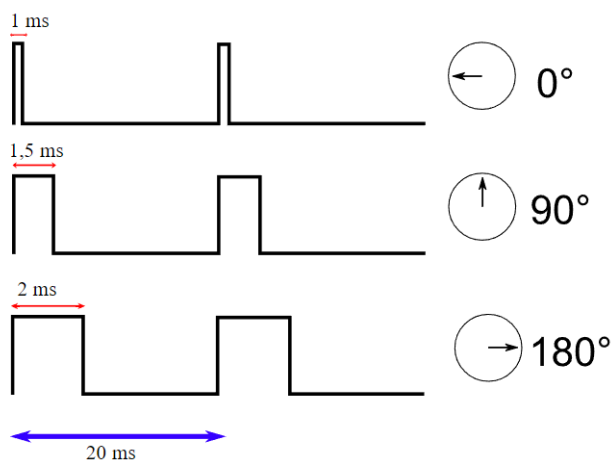
Velocità di rotazione (per 60° a 4,8V):  
0,12s

Coppia di bloccaggio (4,8 V): 1,6 kg/cm

Tensione di funzionamento: 3,5 ~ 6 V

Temperatura: -30 ~ 60 °C

Questo attuatore può essere controllato tramite un segnale a modulazione di larghezza di impulso (PWM) a 50 Hz, che genera un impulso ogni 20 ms. La posizione dell'attuatore può essere regolata variando la durata dell'impulso tra 1 ms e 2 ms.



Arduino mette a disposizione una libreria "Servo.h" che permette di comandare in modo semplice un servo motore.

La funzione principale è la seguente:

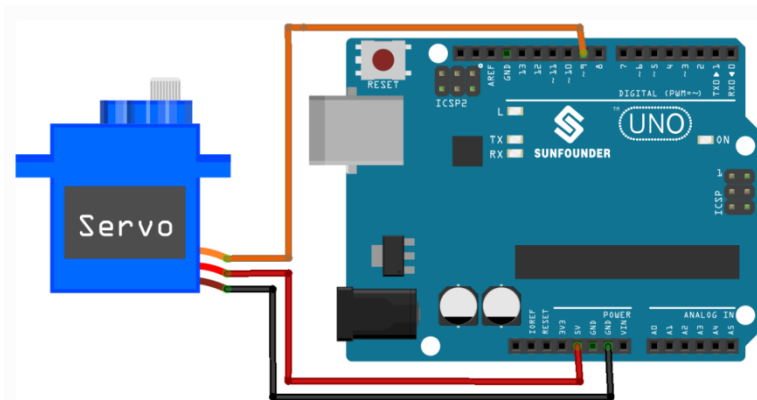
**servo.write(angolo)** → imposta la posizione del servo muovendo l'albero a un angolo specifico tra 0 e 180 gradi.

## INDIVIDUARE LA POSIZIONE 0

Per trovare lo "0" (o posizione neutra/iniziale) di un servomotore SG90, il metodo più affidabile consiste nell'utilizzare un microcontrollore come Arduino per inviare un segnale PWM specifico che comanda al servo di posizionarsi a 0 gradi.



### Metodo Software



L'SG90 richiede 5V.  
L'alimentazione deve essere stabile per evitare comportamenti anomali.  
La massima corrente totale gestibile da una scheda Arduino Uno (R3) è di circa 200-400 mA.  
Quindi un solo SG90 servo sotto sforzo supera il limite di Arduino e di conseguenza serve una alimentazione esterna.

Caricare un semplice codice su Arduino per portare il servo nella posizione 0 o 90 gradi (che solitamente rappresenta il centro meccanico) prima di montare le squadrette (le parti in plastica bianca).

Una volta caricato, il servo si sposterà nella posizione definita (0°). A questo punto, inserisci la squadretta di plastica sull'albero del servo in modo che sia allineata con la posizione desiderata (es. perpendicolare al corpo del servo per lo 0° o 90°).

### CODICE

```
#include <Servo.h>

Servo myservo;

void setup() {
  myservo.attach(9); // Il servo è collegato al pin 9
  myservo.write(0); // Comando per la posizione 0 gradi (o 90 per il centro)
}

void loop() {
  //
}
```

## REGOLAZIONE DELLA VELOCITÀ

Regolare la velocità di un servomotore SG90 (o MG90S) con Arduino non si fa tramite un comando diretto, poiché questi servo sono progettati per girare alla massima velocità verso la posizione specificata. Tuttavia, è possibile controllare la velocità rallentando il movimento, ovvero spostando il servo un grado alla volta con un piccolo ritardo (delay) tra ogni passo.

### Metodo principale: il ciclo for (Arduino)

Per rallentare il movimento, si scompone lo spostamento in piccoli angoli intermedi utilizzando un ciclo for e la funzione delay().

```
#include <Servo.h>

Servo myservo;
int pinServo = 9;
int delayVelocita = 20; // Più alto è il valore, più lento è il servo

void setup() {
  myservo.attach(pinServo);
}

void loop() {
  // Muove da 0 a 180 gradi
  for (int angolo = 0; angolo <= 180; angolo++) {
    myservo.write(angolo);
    delay(delayVelocita); // Regola la velocità qui
  }

  // Muove da 180 a 0 gradi
  for (int angolo = 180; angolo >= 0; angolo--) {
    myservo.write(angolo);
    delay(delayVelocita); // Regola la velocità qui
  }
}
```

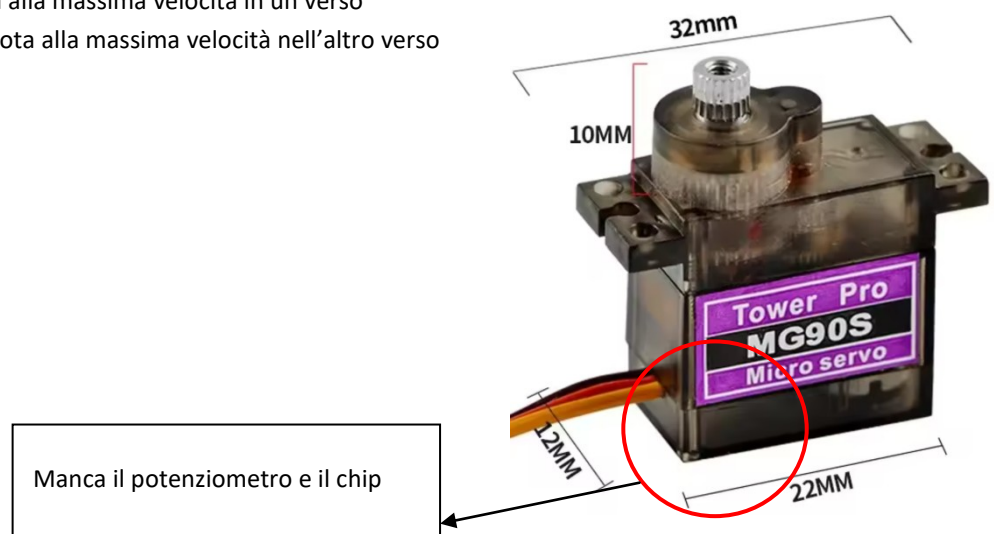
## SERVO A ROTAZIONE CONTINUA

Un MS90 360° è un servo a rotazione continua.

Si riconosce dal fatto che non è presente il potenziometro che permette di individuare la posizione angolare dell'albero.

Comandi:

- il comando write(90) lo ferma
- il comando write(0) ruota alla massima velocità in un verso
- il comando write(180) ruota alla massima velocità nell'altro verso



## SERVO MG90S

La differenza principale tra il micro servo SG90 e il MG90S risiede nel materiale degli ingranaggi e nella robustezza complessiva. Mentre condividono dimensioni quasi identiche, l'MG90S è considerato un aggiornamento "premium" dell'SG90 per progetti che richiedono maggiore forza e resistenza.



Red	Positive pole
Brown	Negative pole
Yellow	Signal

Product Model: MG90S	Temperature Range: 0-55 °C.
Weight: 15g	The dead-band setting: 5 microseconds.
Size: 22.8mm * 12.2mm * 28.5mm	Pulse width: 500~2500µsec
Maximum Angle: 180°±3°	Line Length: 250mm
Stall Torque: 1.8kg/cm(4.8V), 2.0kg/cm(6V)	Plug Type: JR, FUTABA General
Operating Speed: 0.12 sec/ 60°(4.8V), 0.09 sec/ 60°(6V)	

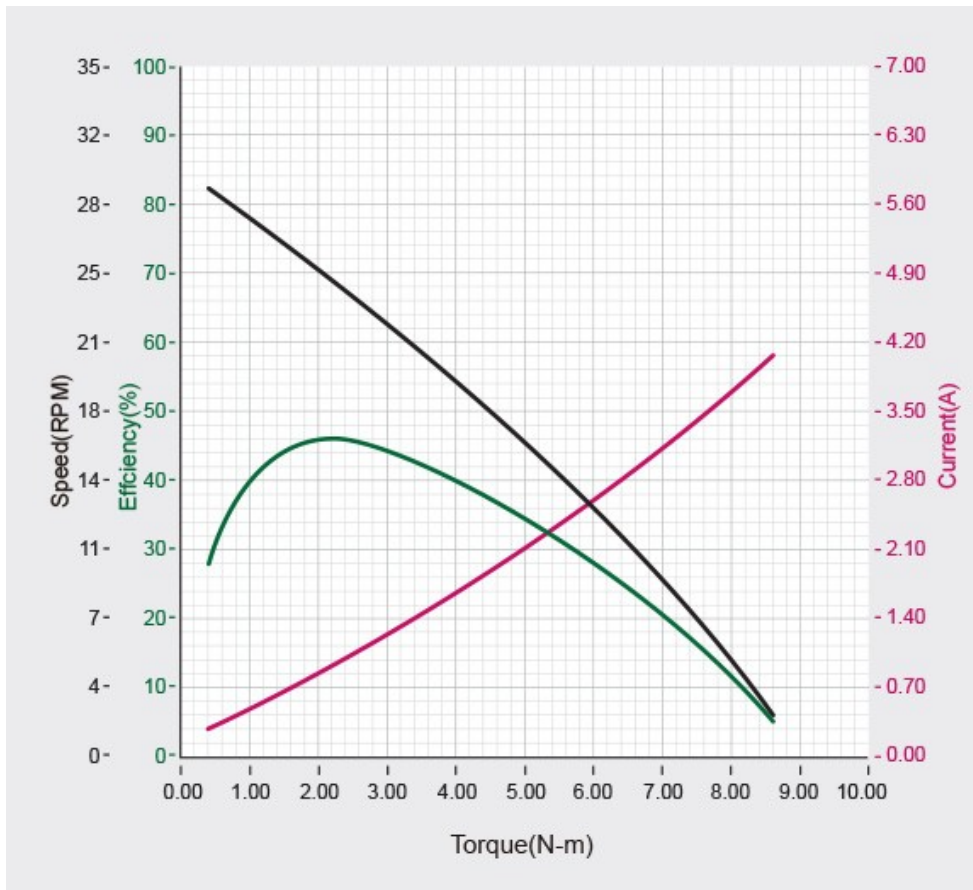


## DYNAMIXEL XM540-W270-T/R

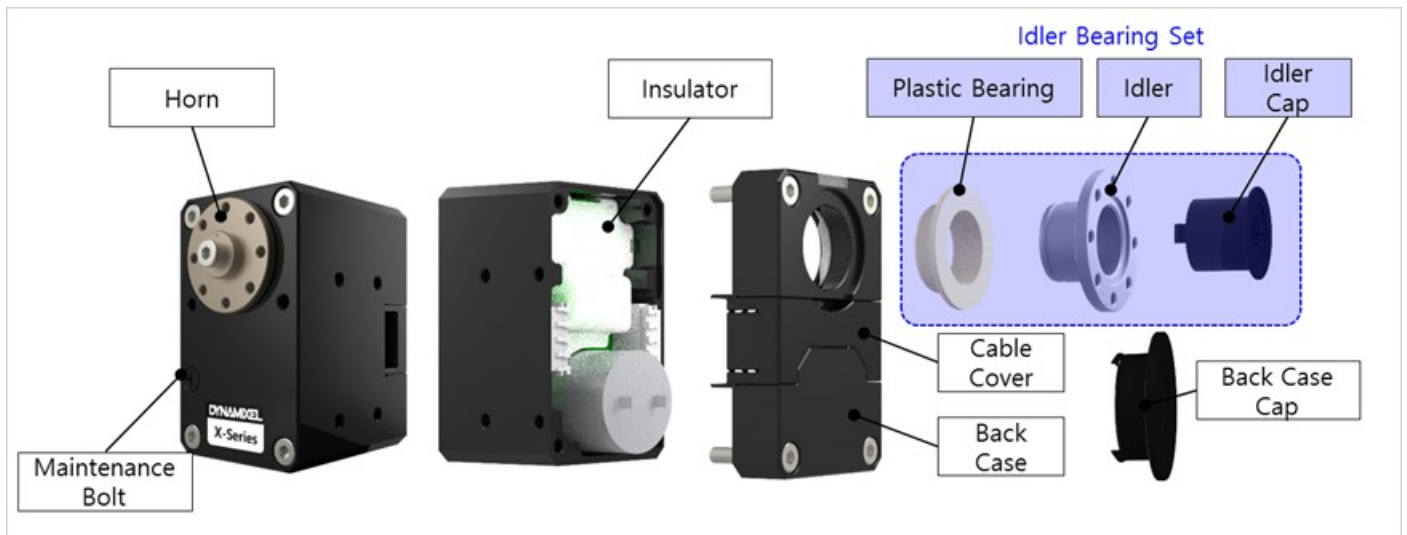
Si tratta di un servo di fascia ALTA.



Item	Specifications
MCU	ARM CORTEX-M3 (72 [MHz], 32Bit)
Position Sensor	Contactless absolute encoder (12Bit, 360 [°]) Maker : ams(www.ams.com), Part No : AS5045
Motor	Coreless
Baud Rate	9,600 [bps] ~ 4.5 [Mbps]
Control Algorithm	PID control
Resolution	4096 [pulse/rev]
Backlash	15 [arcmin] (0.25 [°])
Operating Modes	Current Control Mode Velocity Control Mode Position Control Mode (0 ~ 360 [°]) Extended Position Control Mode (Multi-turn) Current-based Position Control Mode PWM Control Mode (Voltage Control Mode)
Weight	165 [g]
Dimensions (W x H x D)	33.5 x 58.5 x 44 [mm]
Gear Ratio	272.5 : 1
Stall Torque	10.0 [N.m] (at 11.1 [V], 4.2 [A], 2.381 [Nm/A]) <b>10.6 [N.m] (at 12.0 [V], 4.4 [A], 2.409 [Nm/A])</b> 12.9 [N.m] (at 14.8 [V], 5.5 [A], 2.345 [Nm/A])

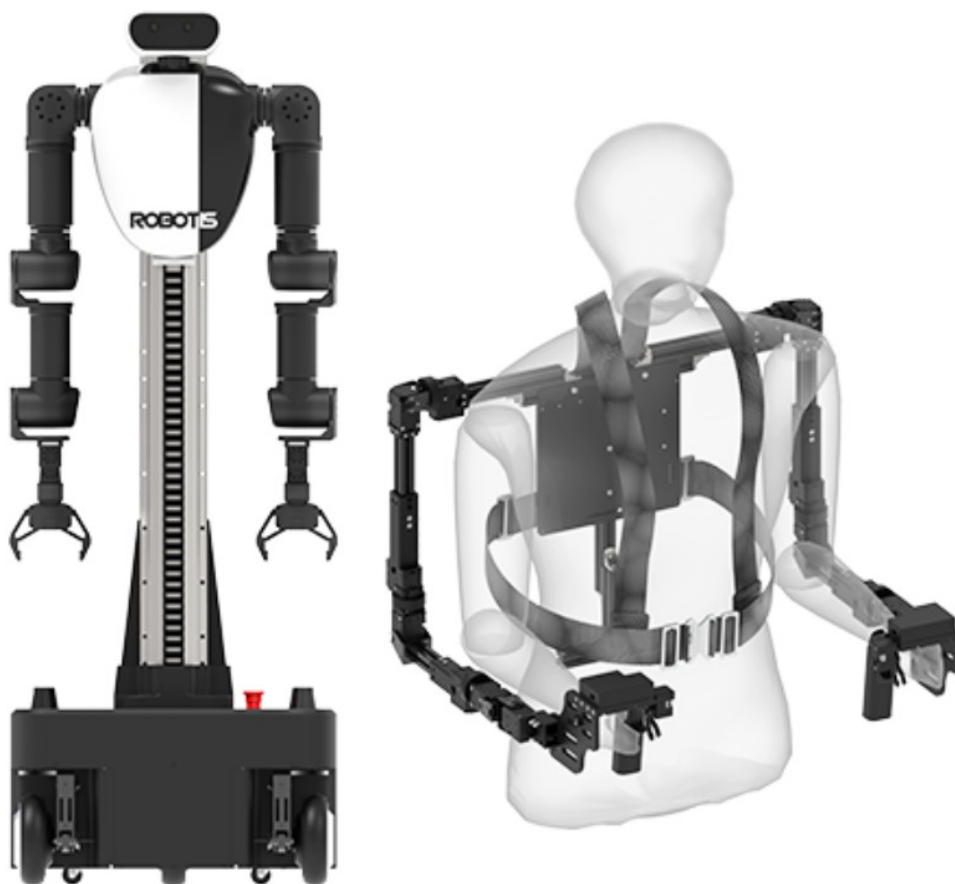


Il Dynamixel XM540-W270-T/R è attualmente uno dei servomotori professionali più equilibrati e performanti per la robotica di precisione. Rispetto alla serie MX, offre una gestione del calore e una durata meccanica nettamente superiori.



### Perché è considerato il "Top" per i Robot

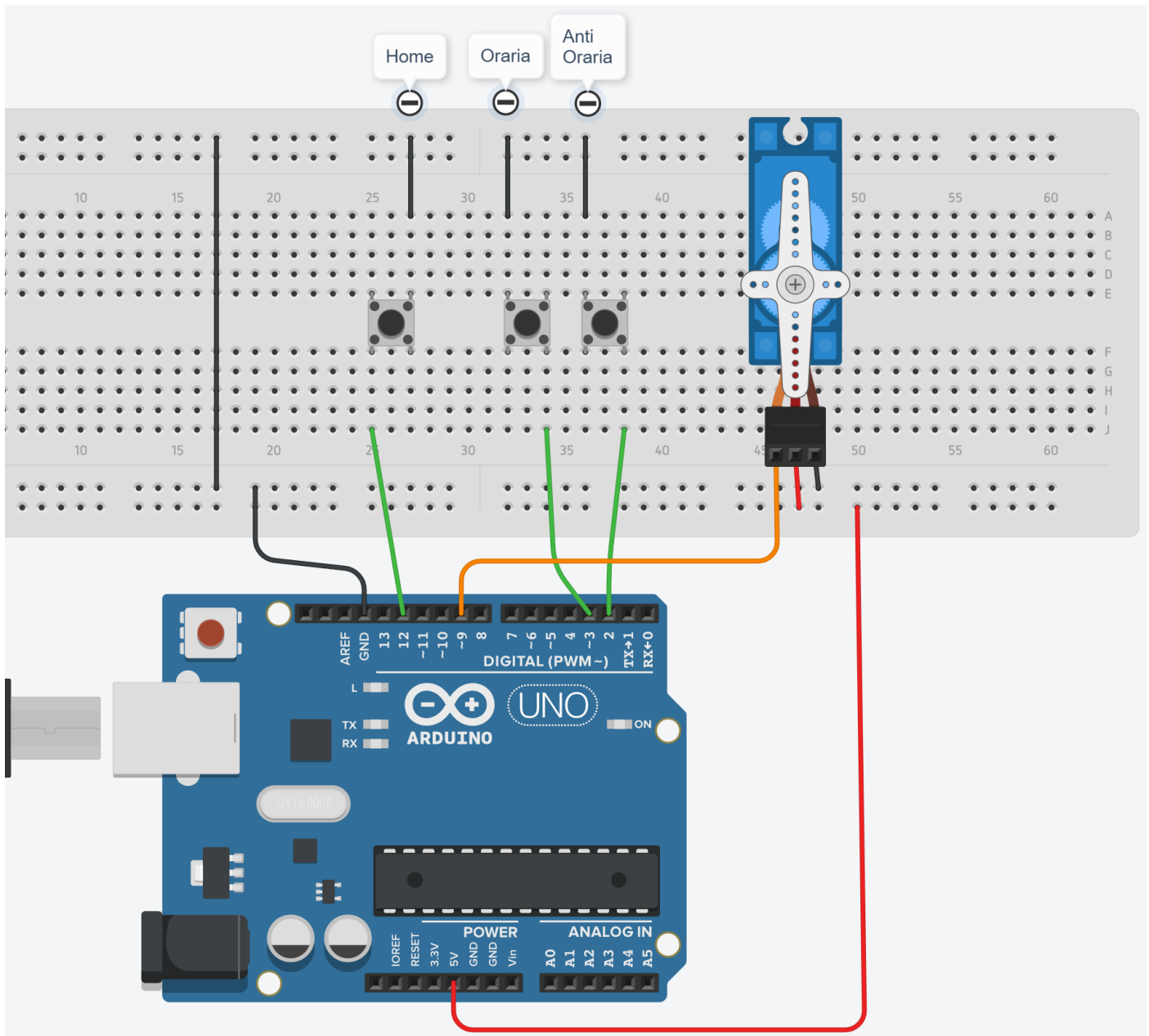
- **Controllo di Corrente (Torque Control):** È la funzione killer. Permette al robot di "sentire" gli ostacoli o di manipolare oggetti delicati regolando la forza, non solo la posizione.
- **Feedback Totale:** Monitora in tempo reale posizione (tramite encoder magnetico a 12-bit/4096 passi), velocità, carico, temperatura e tensione d'ingresso.
- **Cablaggio:** La comunicazione via protocollo RS-485 (versione R) o TTL (versione T) permette di collegare più servo in serie (daisy chain), semplificando drasticamente il design del robot.



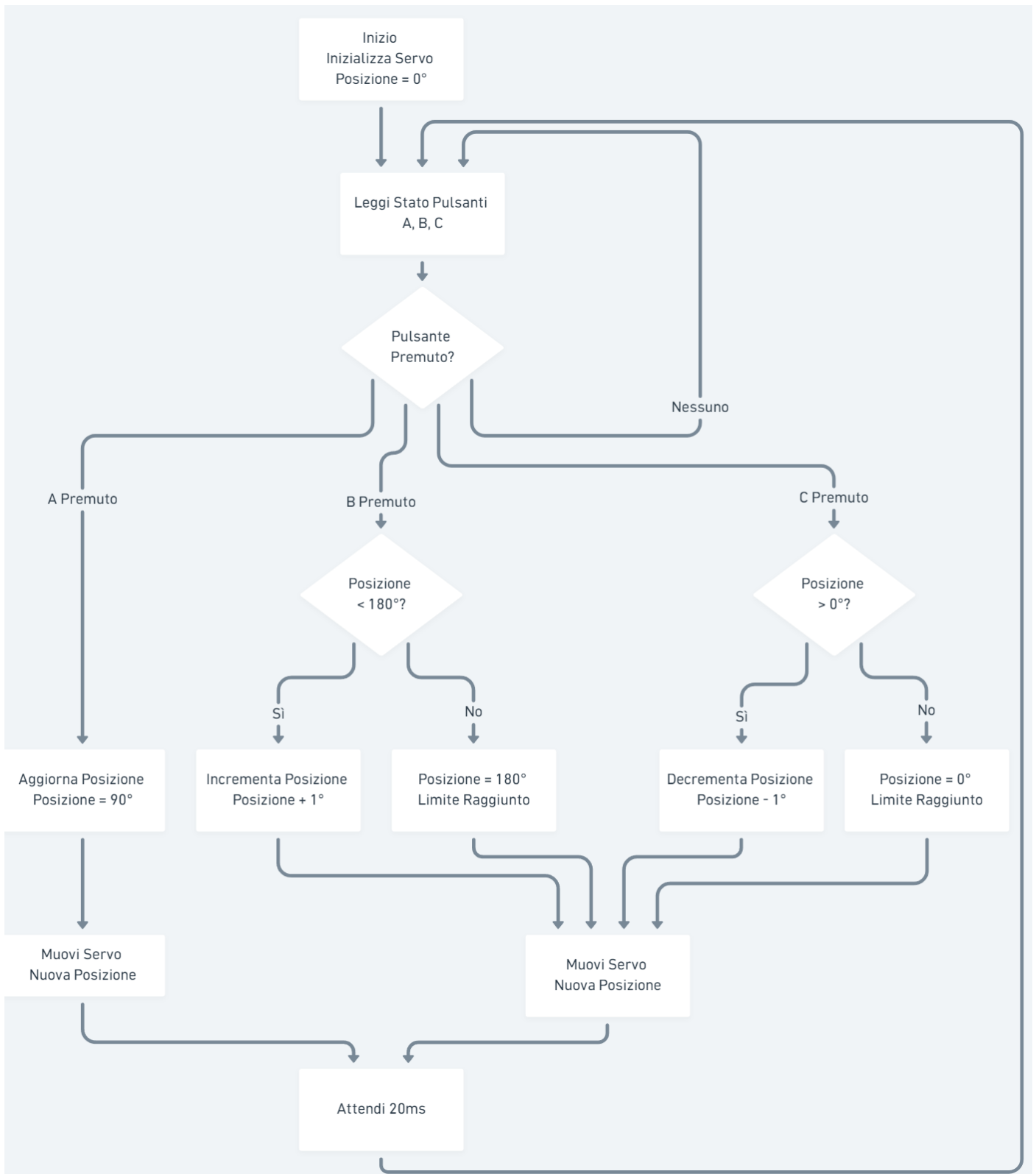
## ESERCIZIO

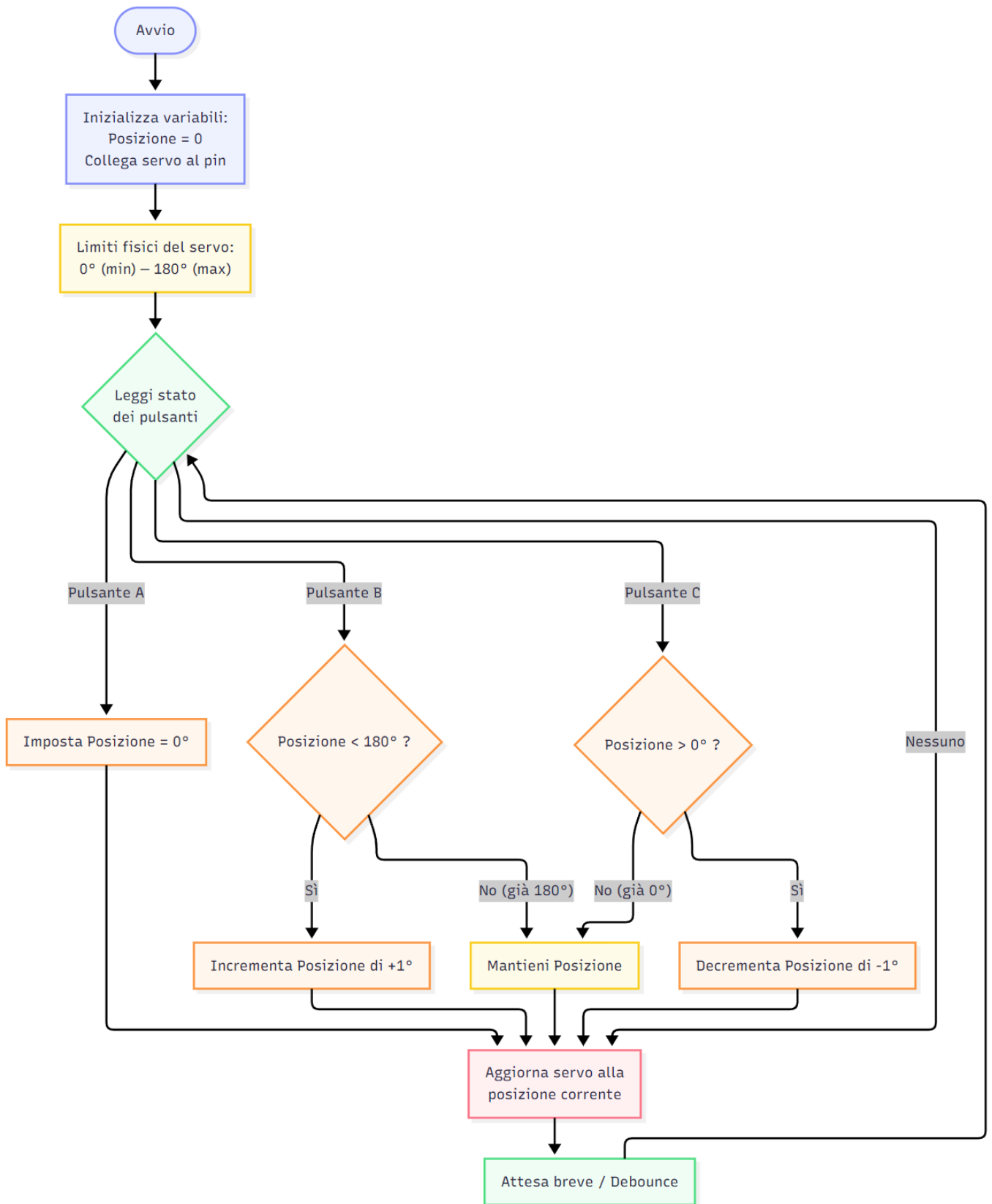
Vogliamo comandare la rotazione di un servomotore tramite tre pulsanti:

- pulsante A: sposta il motore nella posizione di riposo (0°)
- pulsante B: quando premuto sposta il motore di 1 grado alla volta in senso ORARIO
- pulsante C: quando premuto sposta il motore di 1 grado alla volta in senso ANTIORARIO



# DIAGRAMMA DI FLUSSO





## **Codice Arduino**

```
#include <Servo.h>

int btnA= 12; // home
int btnB= 3; // rotazione oraria
int btnC= 2; // rotazione antioraria
int pinServo1 = 9; // servo (PWM)

// Oggetti Servo
Servo Servo1;

// Variabili di posizione
int pos1 = 0;

void setup() {
  Serial.begin(9600);
  Servo1.attach(pinServo1);

  // Configurazione pulsanti in Pull-up
  pinMode(btnC, INPUT_PULLUP); // antioraria
  pinMode(btnB, INPUT_PULLUP); // oraria
  pinMode(btnA, INPUT_PULLUP); // home

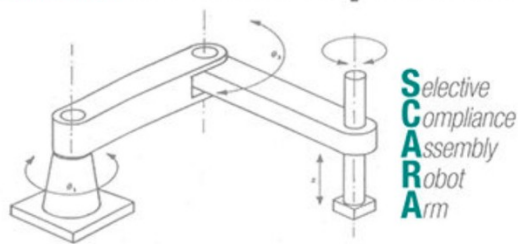
  // Posizionamento iniziale
  Servo1.write(pos1);
}

void loop() {
  // Gestione Servo A (0-180)
  if (digitalRead(btnC) == LOW && pos1 < 180) {pos1++; Serial.println(pos1);}
  if (digitalRead(btnB) == LOW && pos1 > 0) {pos1--; Serial.println(pos1);}
  if (digitalRead(btnA) == LOW ) {pos1=0; Serial.println(pos1);}

  // Muovi motore
  Servo1.write(pos1);
  delay(100);
}
```



## SCARA: velocità e precisione



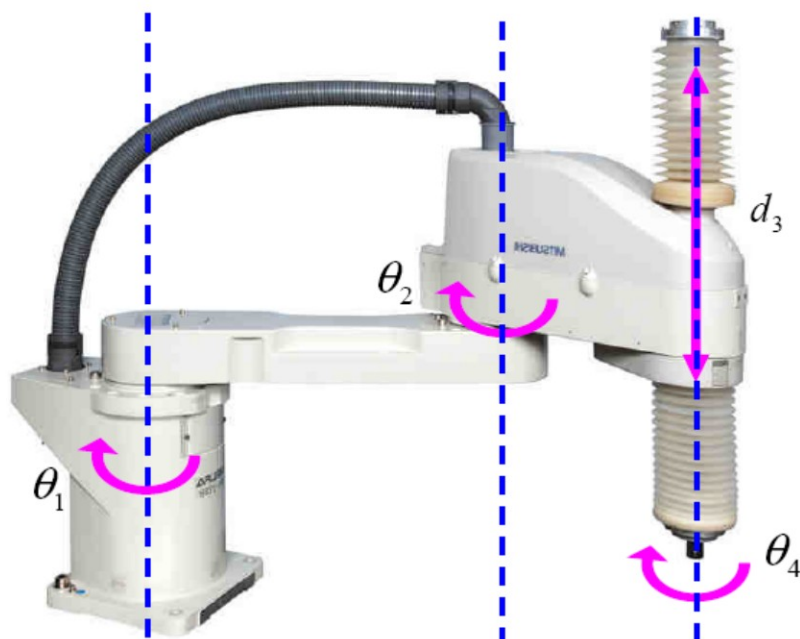
Il robot SCARA (Selective Compliance Assembly Robot Arm) è stato concepito per operazioni veloci e precise. La cinematica del robot SCARA è stata sviluppata all'inizio degli anni '70 in seguito all'osservazione secondo la quale i cicli di movimento più frequenti sono realizzabili con 4 assi.

Il vantaggio che presenta questo tipo di robot rispetto ad altri è dovuto al fatto che per sollevare un pezzo il movimento avviene su un solo asse. Il che ne semplifica la struttura rendendolo più affidabile.

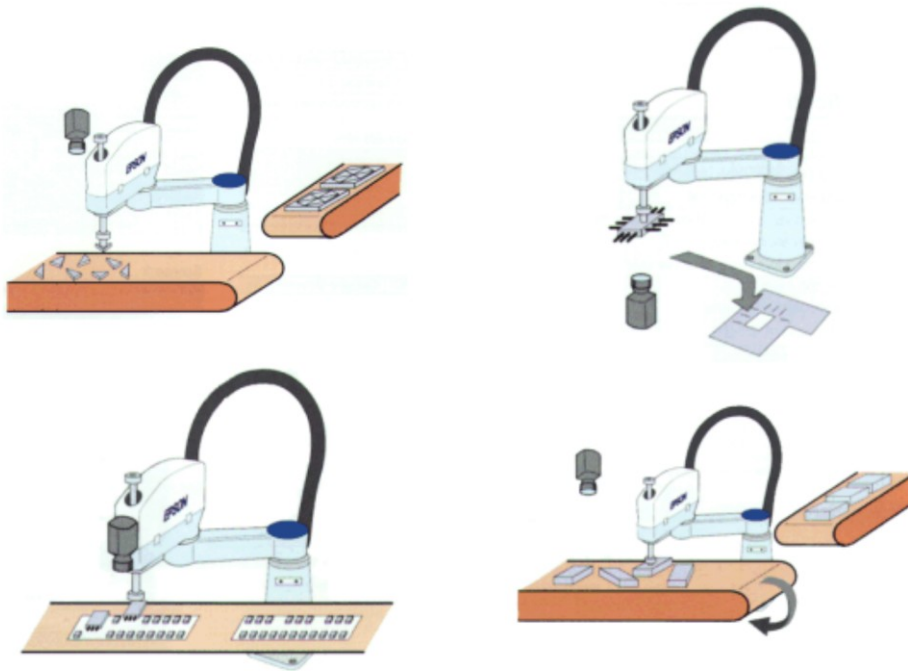
Perciò, laddove è possibile la movimentazione di parti su un livello, i vantaggi dello SCARA prevalgono sensibilmente rispetto a quelli delle altre cinematiche.

Il robot Scara presenta quindi 4 gradi di libertà. In un piano orizzontale si muovono 2 bracci articolati, incernierati ad una estremità con un asse verticale fisso, mentre all'altra estremità libera si trova 1 asse Z, il quale può muoversi sia verticalmente che ruotare intorno al proprio asse.

## MOVIMENTI E ANGOLI DEL ROBOT SCARA



## APPLICAZIONI TIPICHE DEL ROBOT SCARA



I robot SCARA offrono il massimo delle prestazioni di ripetibilità rispetto a tutti i tipi di robot.

Gli errori che si verificano nella posizione X-Y sono dovuti all'utilizzo di due motori in J1 e J2.

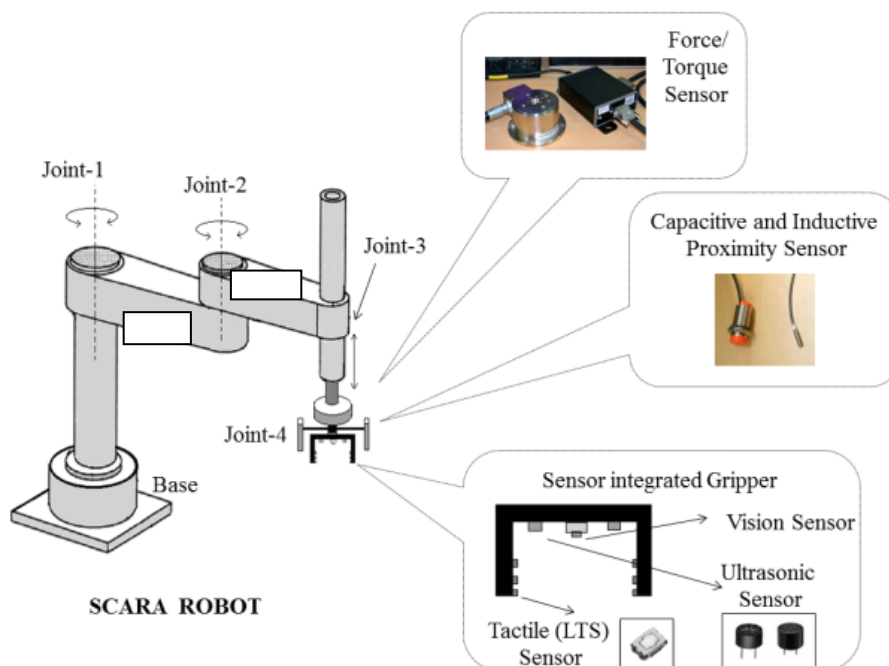
Gli altri tipi di robot utilizzano tre o più motori per contribuire alla posizione X-Y.

Il numero dei motori è direttamente proporzionale agli errori che potrebbero verificarsi.

L'eccellente ripetibilità è un elemento fondamentale per le piccole applicazioni di assemblaggio, in cui occorre rispettare tolleranze inferiori a diversi micron. Ad esempio, può trattarsi dell'inserimento dei connettori nelle schede elettroniche o dello spostamento di un ago in una piccola fessura per la distribuzione.

Uno SCARA può permettere raggi di azione da 100 mm a 1.200 mm, con capacità di carico da 1 kg a 200 kg.

## END EFFECTOR

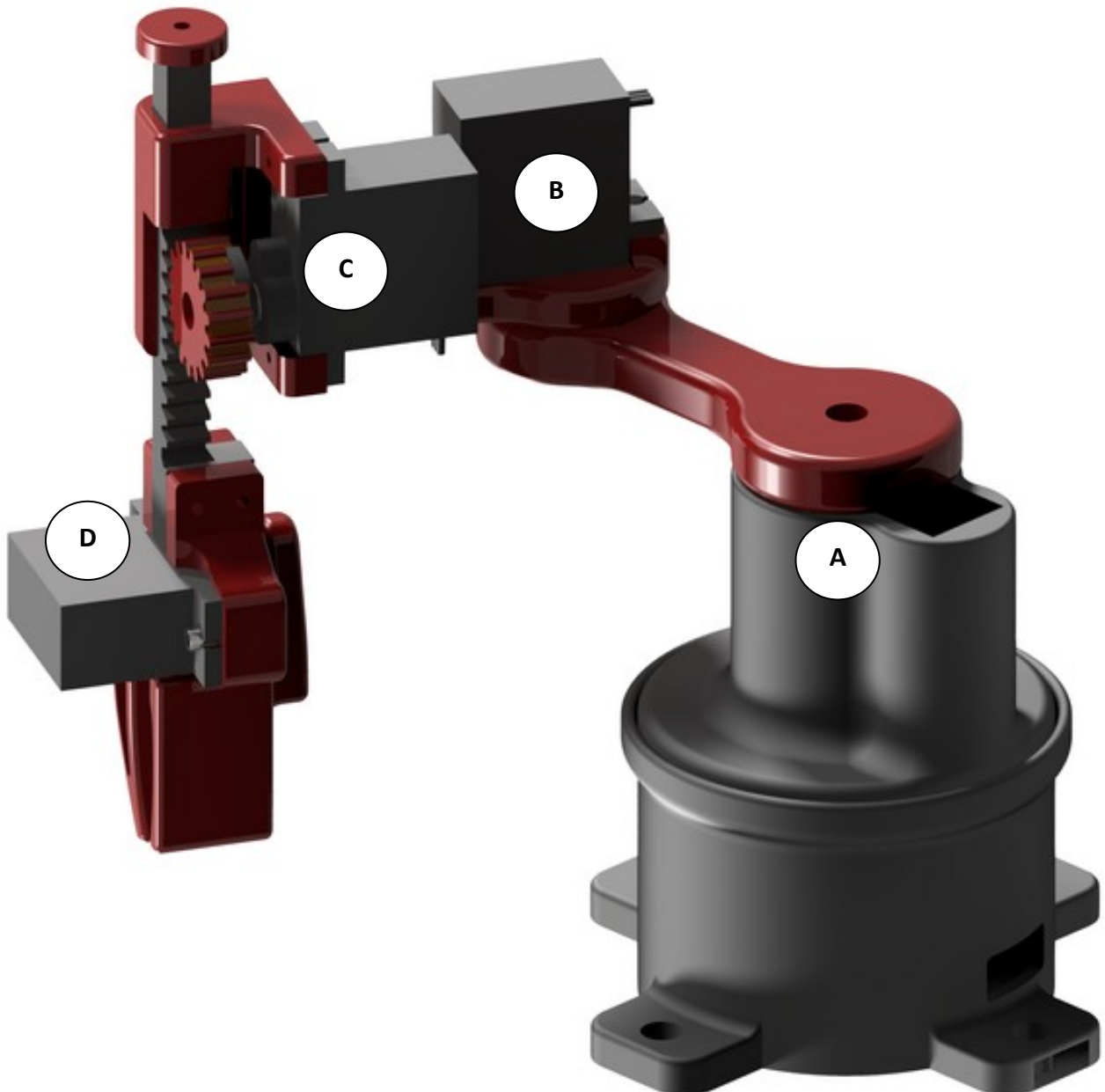


## COMANDO SCARA

Vogliamo comandare i 3 giunti (A,B,C) del robot Scara e l'end effector (D) tramite delle coppie di pulsanti.  
Un pulsante Home porta tutti i giunti nella posizione di riposo.

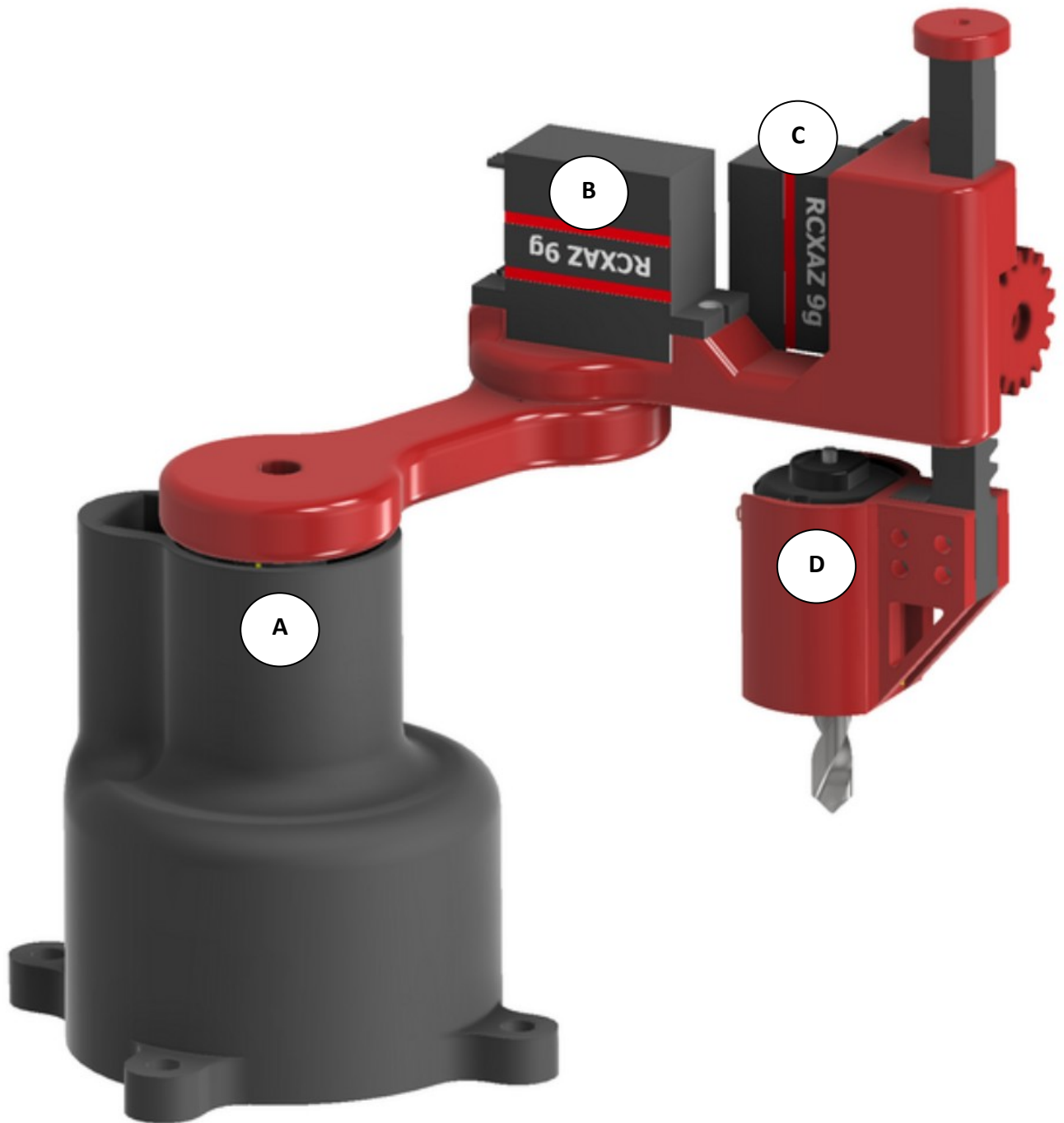
Robot Scara con pinza:

- A → giunto rotazionale link1 (angolo 0-180°)
- B → giunto rotazionale link 2 (angolo 0-180°)
- C → giunto prismatico link3 (angolo 0-180°)
- D → pinza (angolo 0-30°)

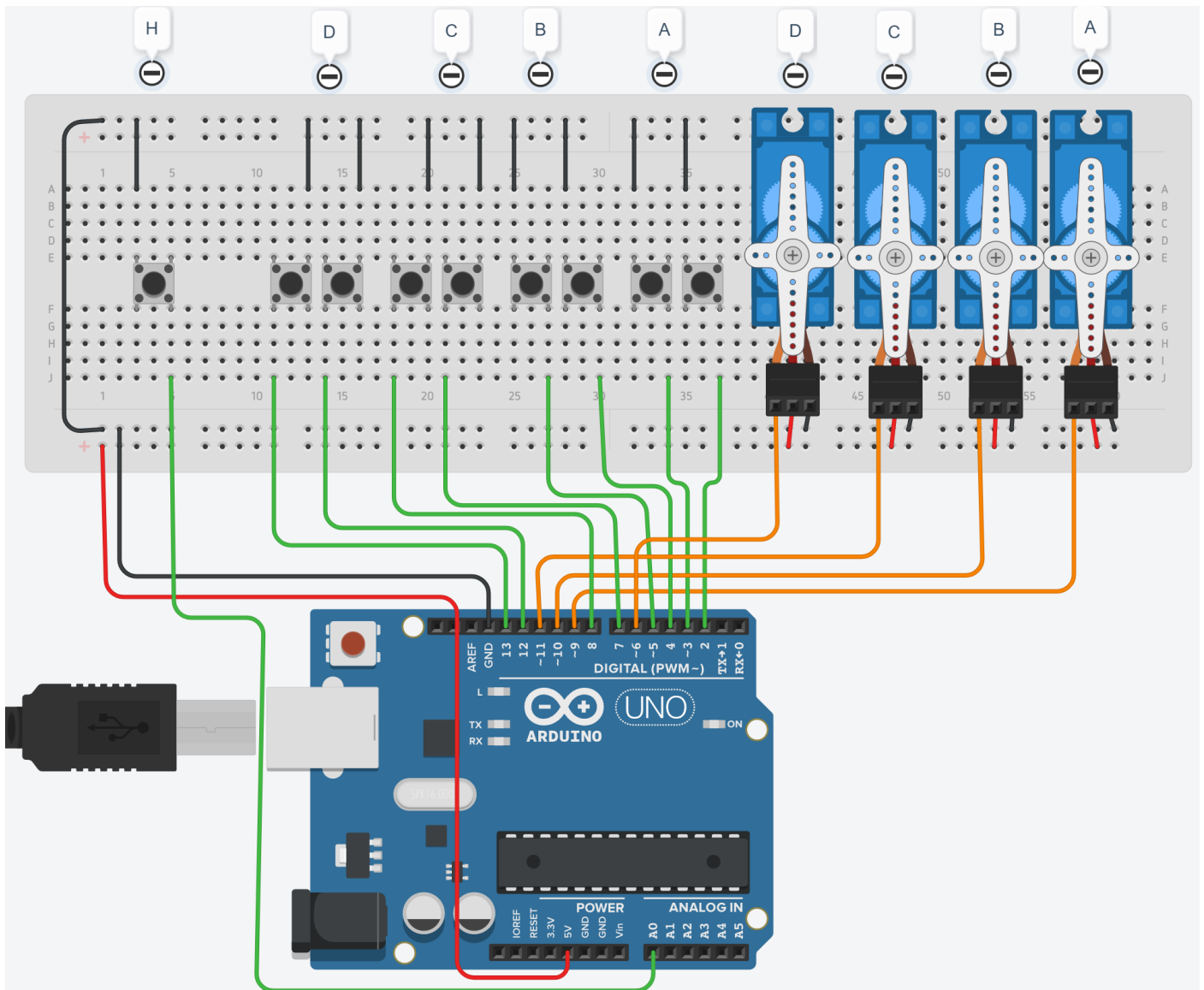


Robot Scara con trapano:

- A → giunto rotazionale link1 (angolo 0-180°)
- B → giunto rotazionale link 2 (angolo 0-180°)
- C → giunto prismatico link3 (angolo 0-180°)
- D → motore (ON-OFF)



## Schema circuitale



NB: è consigliato l'utilizzo di un alimentatore esterno per i Servo!

## Codice Arduino

```
#include <Servo.h>

const int btnAor = 2, btnAor = 3;
const int btnBaor = 4, btnBor = 5;
const int btnCaor = 7, btnCor = 8;
const int btnDaor = 12, btnDor = 13;

const int btnHome = A0;

// Oggetti Servo
Servo sA, sB, sC, sD;

// Variabili di posizione (inizializzate a 0 per Home)
int posA = 0, posB = 0, posC = 0, posD = 0;

void setup() {
  Serial.begin(9600);
  // PWM pin
  sA.attach(9); sB.attach(10); sC.attach(11); sD.attach(6);

  // Configurazione pulsanti in Pull-up
  pinMode(btnAor, INPUT_PULLUP); pinMode(btnAor, INPUT_PULLUP);
  pinMode(btnBaor, INPUT_PULLUP); pinMode(btnBor, INPUT_PULLUP);
  pinMode(btnCaor, INPUT_PULLUP); pinMode(btnCor, INPUT_PULLUP);
  pinMode(btnDaor, INPUT_PULLUP); pinMode(btnDor, INPUT_PULLUP);
  pinMode(btnHome, INPUT_PULLUP);

  // Posizionamento iniziale
  sA.write(posA); sB.write(posB); sC.write(posC); sD.write(posD);
}

void loop() {
  // Gestione Servo
  if (digitalRead(btnAor) == LOW && posA < 180) {posA++; Serial.println(posA);}
  if (digitalRead(btnAor) == LOW && posA > 0) {posA--; Serial.println(posA);}

  if (digitalRead(btnBaor) == LOW && posB < 180) {posB++; Serial.println(posB);}
  if (digitalRead(btnBor) == LOW && posB > 0) {posB--; Serial.println(posB);}

  if (digitalRead(btnCaor) == LOW && posC < 180) {posC++; Serial.println(posC);}
  if (digitalRead(btnCor) == LOW && posC > 0) {posC--; Serial.println(posC);}

  if (digitalRead(btnDaor) == LOW && posD < 180) {posD++; Serial.println(posD);}
  if (digitalRead(btnDor) == LOW && posD > 0) {posD--; Serial.println(posD);}

  // Gestione Pulsante Home
  if (digitalRead(btnHome) == LOW) {posA = 0; posB = 0; posC = 0; posD = 0;}

  // Muovo servo nelle posizioni indicate
  sA.write(posA); sB.write(posB); sC.write(posC); sD.write(posD);
  delay(100); // Regola la velocità di movimento dei pulsanti
}
```

## ROBOT ANTROPOMORFO

I robot antropomorfi sono robot con movimenti su 5 o più assi che ricordano nella forma e nelle possibilità di articolazione il braccio umano. Per questo motivo sono anche denominati bracci robotici antropomorfi.



## GRADI DI LIBERTÀ DI UN ROBOT

Per assi o gradi libertà nei robot industriali si intendono quei singoli componenti mobili che consentono al braccio robotico (antropomorfo o pallettizzatore) di muoversi in un determinato spazio nonché in un preciso campo di lavoro delimitato.

Questi giunti consentono al robot di posizionare ed orientare l'end effector (mano di presa o torcia) in diversi punti nello spazio conferendo al robot la capacità di eseguire la lavorazione richiesta.

Un esempio pratico sono operazioni di pallettizzazione o depallettizzazione in cui non si necessita, nella grande maggioranza dei casi, di un braccio antropomorfo ma si predilige un robot a 4 assi perché più veloce.

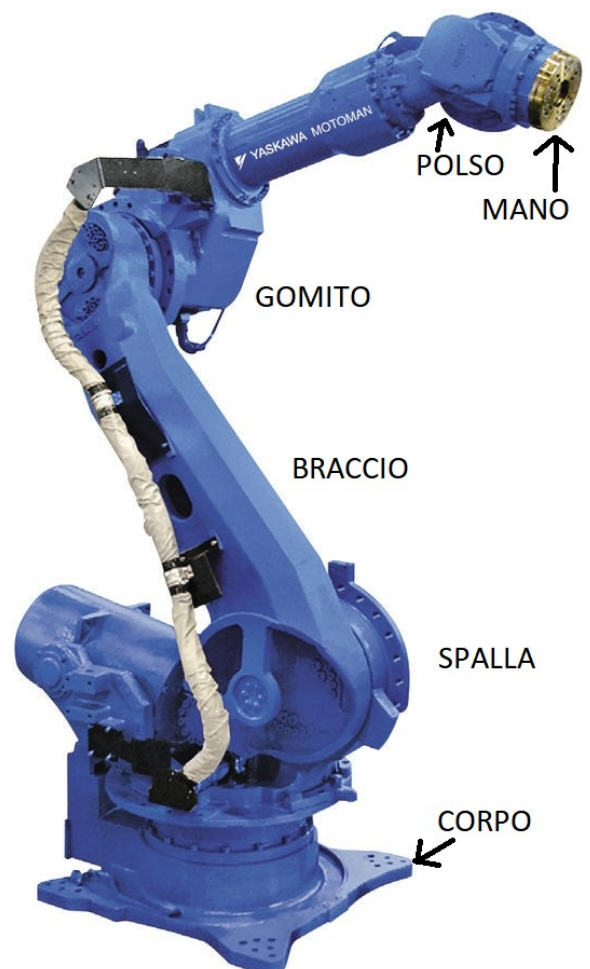
Nei processi di saldatura o di manipolazioni o di asservimento si scelgono robot antropomorfi a 6 assi che permettono più agilità e più precisione dei movimenti.

I robot a 6 assi possiedono sei articolazioni, consentendo loro di eseguire movimenti simili a quelli umani.

Queste articolazioni sono numerate e corrispondono alle giunzioni del braccio umano.

Analizzando un robot a 6 assi, vediamo una base con la prima articolazione, consentendo la rotazione del braccio meccanico.

Seguono la spalla, che permette di sollevare e abbassare il braccio, e il gomito, rappresentando lo snodo tra avambraccio e braccio. Proseguono poi il polso, che può muoversi su tre assi.



### Industria Manifatturiera e Automobilistica

- Saldatura: È una delle applicazioni più comuni. Il robot può eseguire saldature ad arco o a punti con una costanza millimetrica, muovendosi attorno al telaio di un'auto.
- Verniciatura: Grazie alla fluidità dei movimenti, garantiscono uno strato di vernice omogeneo, riducendo gli sprechi e proteggendo gli umani dai fumi tossici.
- Assemblaggio: Montaggio di componenti meccanici, inserimento di viti o incastro di parti plastiche.

### Logistica e Movimentazione (Pick & Place)

- Pallettizzazione: Impilare scatole su un pallet. Un robot antropomorfo può gestire carichi pesanti e orientare ogni scatola nel modo corretto per ottimizzare lo spazio.
- Asservimento macchine: Il robot preleva un pezzo grezzo e lo inserisce in una macchina a controllo numerico (CNC) o in una pressa, per poi estrarre il prodotto finito.

### Lavorazioni Meccaniche di Precisione

- A differenza dei robot rigidi, l'antropomorfo può seguire profili curvi complessi:
- Sbavatura e Lucidatura: Rifinitura di componenti in metallo o plastica dopo la fusione.
- Taglio Laser o Plasma: Seguendo percorsi 3D nello spazio per tagliare forme complesse su pezzi non piani.

### Settore Alimentare e Farmaceutico

- Confezionamento: Manipolazione di cibi (spesso con pinze speciali a ventosa) per inserirli nelle vaschette.
- Laboratorio: Gestione di provette e campioni biologici in ambienti sterili dove la presenza umana potrebbe contaminare il processo.

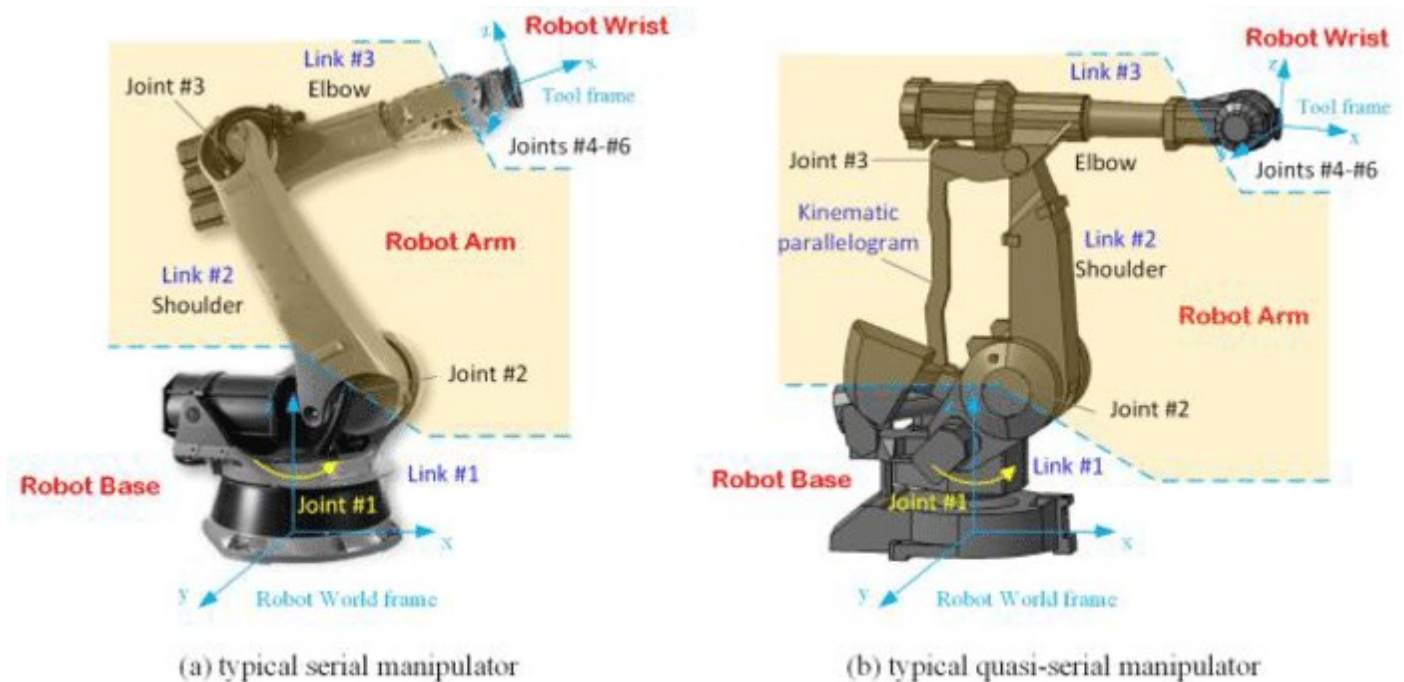
### Applicazioni "Collaborative" (Cobot)

- Una sottocategoria emergente è quella dei Cobot (robot collaborativi), progettati per lavorare fianco a fianco con l'uomo senza barriere di sicurezza:
- Controllo Qualità: Il robot solleva il pezzo e lo mostra a una telecamera o a un operatore umano per l'ispezione.
- Avvitatura assistita: L'operatore posiziona il pezzo e il robot inserisce le viti con la coppia di serraggio esatta.



## ROBOT SERIALI E QUASI SERIALI

I robot seriali sono manipolatori industriali composti da una catena cinematica aperta di bracci e giunti (rotanti o prismatici) collegati in serie, dalla base all'end-effector. I robot antropomorfi sono un tipico *robot seriale*.



I *robot quasi-seriali* hanno un'architettura pressoché identica (vedi Fig. 1b), ma a differenza delle loro controparti seriali, il braccio di un manipolatore quasi-seriale contiene un parallelogramma cinematico, che può essere trattato come un circuito chiuso interno.

Per questo motivo, tali robot vengono spesso definiti quasi-seriali.

In pratica, il parallelogramma cinematico consente ai progettisti di robot di migliorare le proprietà dinamiche del robot riducendo le masse mobili (posizionando il pesante motore di attuazione n. 3 sulla base del robot anziché sul gomito).

Solitamente, il parallelogramma non influisce in modo significativo sul controllo del manipolatore e non modifica le equazioni cinematiche dirette/inverse del manipolatore.

D'altra parte, il modello di rigidità del manipolatore quasi-seriale differisce sostanzialmente da quello del manipolatore seriale, poiché il riposizionamento dell'elemento flessibile del manipolatore (trasmissione dell'attuatore n. 3) influenza in modo significativo il comportamento di rigidità.

Caratteristica	Robot seriali	Robot quasi-seriali
Struttura	Catena cinematica aperta (articolazione 1 → articolazione 6)	Catena aperta con parallelogrammi cinematici aggiunti
Rigidità	Minore rigidità	Maggiore rigidità
Carico utile/Precisione	Soggetto ad accumulo di errori	Capacità di carico migliorata e deflessione ridotta
Complessità	Semplice, facile da programmare	Progettazione meccanica leggermente più complessa
Ideale per	Ampia portata, flessibilità	Compiti che richiedono maggiore precisione e rigidità



Follow @vanessaloiola

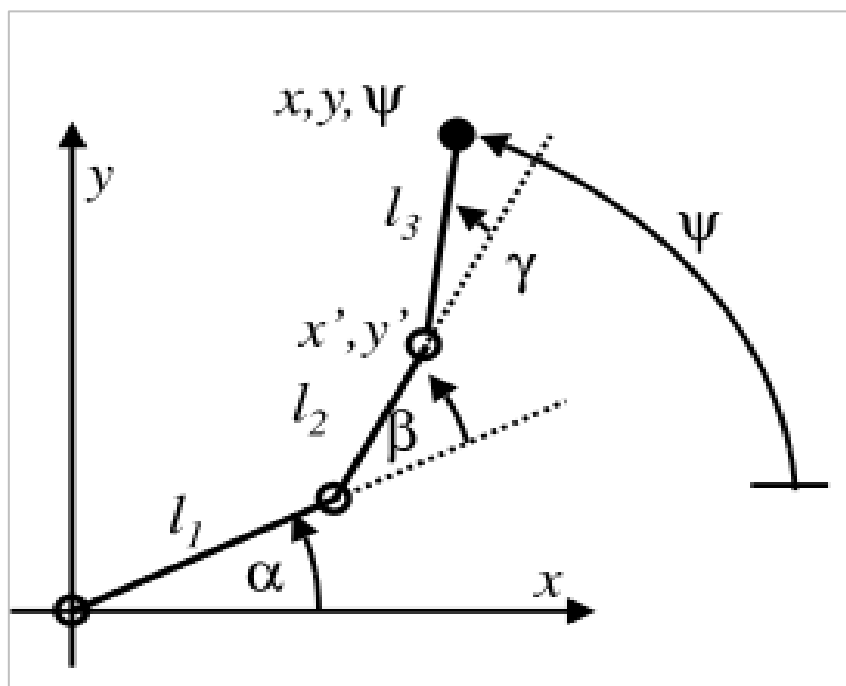
La cinematica diretta è risolta dalle seguenti equazioni:

$$\begin{cases} x = l_1 \cos(\alpha) + l_2 \cos(\alpha + \beta) + l_3 \cos(\alpha + \beta + \gamma) \\ y = l_1 \sin(\alpha) + l_2 \sin(\alpha + \beta) + l_3 \sin(\alpha + \beta + \gamma) \\ \psi = \alpha + \beta + \gamma \end{cases}$$

La cinematica inversa si può risolvere calcolando inizialmente le coordinate  $x'$  e  $y'$  del centro del terzo accoppiamento rotoidale e applicando poi ad esse la soluzione del robot SCARA classico:

$$\begin{cases} x' = x - l_3 \cos(\psi) \\ y' = y - l_3 \sin(\psi) \\ \beta = \pm \arccos\left(\frac{x'^2 + y'^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \\ \alpha = \text{atan2}(y', x') - \text{atan2}(l_2 \sin(\beta), l_1 + l_2 \cos(\beta)) \\ \gamma = \psi - \alpha - \beta \end{cases}$$

Il robot ha evidentemente due soluzioni e le configurazioni singolari si hanno per  $\beta = 0, \pi$ .

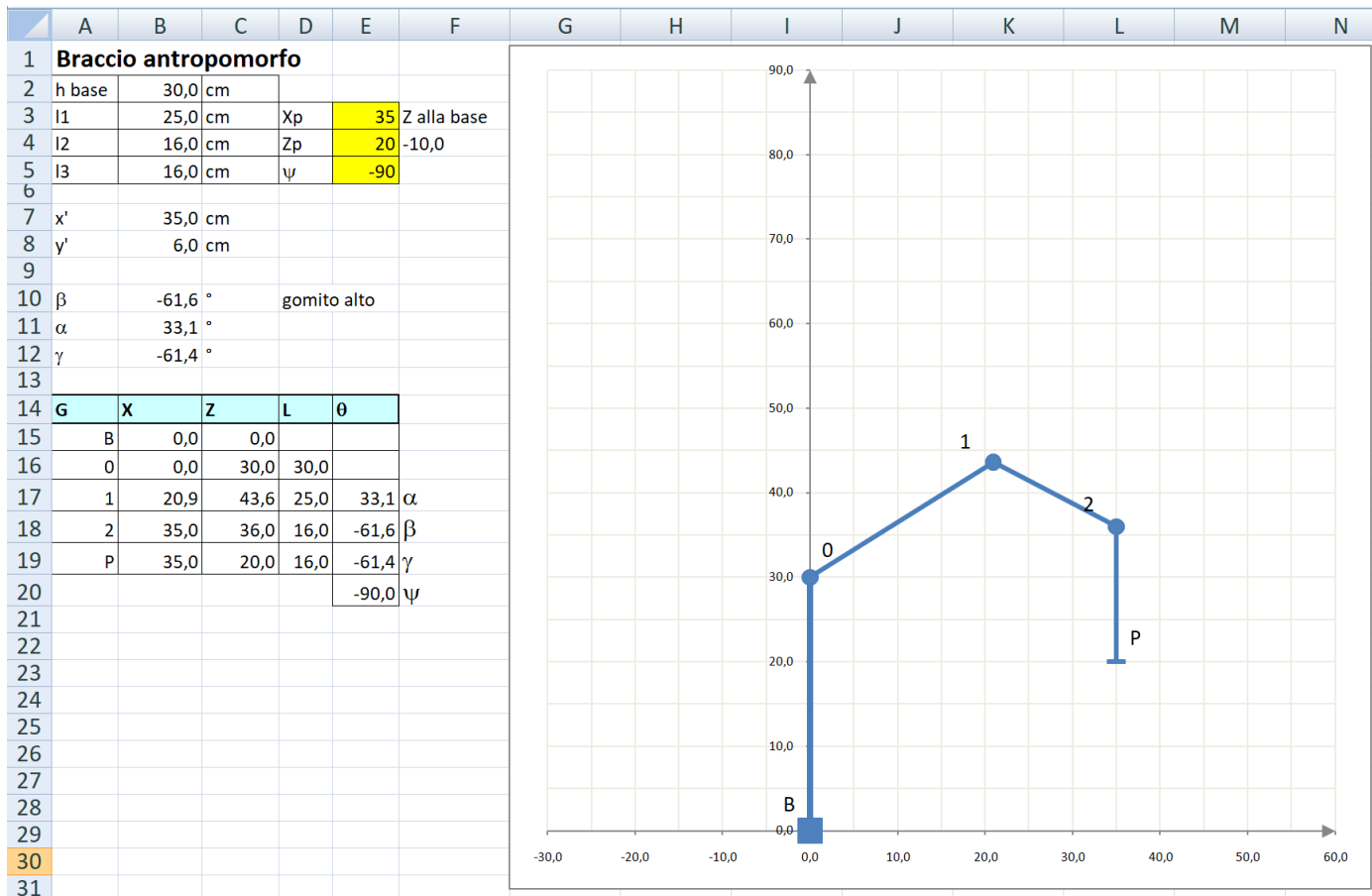


**NB:** l'angolo  $\Psi$  deve essere assegnato (dato di input noto).

# CINEMATICA ROBOT ANTROPOMORFO CON EXCEL

Creare un foglio di calcolo per la cinematica inversa del robot antropomorfo.

In giallo i dati in ingresso.



Calcolo angoli dei giunti

	A	B	C	D	E	F	G
1	<b>Braccio antropomorfo</b>						
2	h base	30,0	cm				
3	l1	25,0	cm	Xp	35	Z alla base	
4	l2	16,0	cm	Zp	20	-10,0	
5	l3	16,0	cm	ψ	-90		
6							
7	x'	35,0	cm				
8	y'	6,0	cm				
9							
10	β	=GRADI(ARCCOS((B7^2+B8^2-B3^2-B4^2)/(2*B3*B4)))					
11	α	GRADI(angolo)					
12	γ	-61,4 °					

	A	B	C	D	E	F	G	H	I
1	<b>Braccio antropomorfo</b>								
2	h base	30,0	cm						
3	l1	25,0	cm	Xp	35	Z alla base			
4	l2	16,0	cm	Zp	20	-10,0			
5	l3	16,0	cm	ψ	-90				
6									
7	x'	35,0	cm						
8	y'	6,0	cm						
9									
10	β	-61,6 °					gomito alto		
11	α	=GRADI(ARCTAN.2(B7;B8)-ARCTAN.2(B3+B4*COS(RADIANTI(B10));B4*SEN(RADIANTI(B10))))							
12	γ	GRADI(angolo)							

	A	B	C	D	E	F	
1	<b>Braccio antropomorfo</b>						
2	h base	30,0	cm				
3	l1	25,0	cm	Xp	35	Z alla base	
4	l2	16,0	cm	Zp	20	-10,0	
5	l3	16,0	cm	ψ	-90		
6							
7	x'	35,0	cm				
8	y'	6,0	cm				
9							
10	β	-61,6 °					gomito alto
11	α	33,1 °					
12	γ	=SESS-SB\$11-B10					

## SCARA VS ANTROPOMORFO

La scelta tra un robot SCARA e uno antropomorfo dipende strettamente dall'applicazione, dallo spazio di lavoro e dalla precisione richiesta.

### 1. Robot SCARA (Selective Compliance Articulated Robot Arm)

Questi robot sono progettati per movimenti orizzontali veloci e precisi, ideali per operazioni "sopra un piano".

- Assi: Solitamente 4 assi.
- Movimento: Braccio articolato mobile sul piano orizzontale, con un asse verticale per lo spostamento su/giù (Z).
- Punti di forza: Estrema velocità e ripetibilità. Sono perfetti per il pick-and-place, l'assemblaggio di piccoli componenti e il confezionamento.
- Limiti: Raggio d'azione limitato a una forma cilindrica; non possono inclinare l'utensile.
- Costo indicativo: Generalmente tra i \$15.000 e gli \$80.000.

### 2. Robot Antropomorfo (Braccio Articolato)

Questi robot imitano la struttura e la flessibilità del braccio umano, offrendo la massima versatilità.

- Assi: Solitamente 6 assi (o più), il che permette loro di raggiungere punti nello spazio da qualsiasi angolazione.
- Movimento: Rotatorio su tutti i giunti, inclusa una "spalla", un "gomito" e un "polso" articolato.
- Punti di forza: Versatilità totale. Possono eseguire compiti complessi come saldatura, verniciatura, pallettizzazione e lavorazioni su superfici curve.
- Limiti: Programmazione più complessa e ingombro potenzialmente maggiore rispetto a uno SCARA per compiti semplici.
- Costo: Variabile, ma tendenzialmente superiore a causa della maggiore complessità meccanica ed elettronica.

### Tabella Comparativa Rapida

Caratteristica	Robot SCARA	Robot Antropomorfo
<b>Gradi di libertà</b>	4 (limitati)	6+ (elevati)
<b>Velocità</b>	Molto elevata in orizzontale	Elevata, ma variabile
<b>Precisione</b>	Eccellente per assemblaggi	Ottima, ma più sensibile al carico
<b>Spazio di lavoro</b>	Cilindrico, "piatto"	Sferico, molto ampio
<b>Applicazione tipica</b>	Pick & Place, Assemblaggio	Saldatura, Smerigliatura, Verniciatura

## END EFFECTOR (END OF ARM TOOLING EOAT)

L'end effector è l'elemento di collegamento tra robot e pezzo in lavorazione(workpiece) in grado di apportare valore aggiunto al processo finale. Senza un end effector, un robot non può fare molto infatti non può considerarsi una macchina.

Ci sono diversi tipi di end effector come pinze, torce di saldatura, sensori forza e/o di coppia, strumenti di rimozione del materiale, sistemi di aspirazione, cambio utensili, ecc.

Gli end effector possono essere alimentati elettricamente, idraulicamente, pneumaticamente, ecc.

## PERCHÉ È NECESSARIO L'END OF ARM TOOLING (EOAT)

Lavori diversi richiedono strumenti diversi.

Anche processi che potrebbero sembrare identici a prima vista possono richiedere soluzioni molto diverse a causa delle differenze di ambiente o del modo in cui i pezzi sono presentati.

Sia i robot industriali che quelli collaborativi sono progettati per disporre di un'ampia gamma di EOAT.

Generalmente i robot non sono forniti con utensili specifici all'estremità del braccio, essendo dotati di una flangia di adattamento unificata in uscita dell'ultimo giunto, possono collegare qualsiasi sistema per adattarsi all'applicazione.

Spetta quindi a chi implementa il robot (integratore), valutare il lavoro da fare e selezionare gli strumenti più appropriati per i compiti da svolgere.

### **Prima di tutto occorre capire l'applicazione per la determinazione dell' EOAT:**

- Qual è il peso degli oggetti che il cobot dovrà spostare?
- Quali sono le forme e le dimensioni degli oggetti da spostare?
- L'oggetto (workpiece) che dobbiamo lavorare può avere delle variazioni (es. di dimensioni che cambiano nel tempo)?
- Che tipo di superfici hanno gli oggetti (es. dure, morbide, porose, asciutte, oliate)?
- Gli oggetti sono rigidi, instabili o devono essere maneggiati con cura?
- Il sistema di presa può avere un movimento solo aperto/chiuso o di corse intermedie e flessibili?
- Il Sistema di presa deve avere requisiti relativi all'igiene, (es. camera bianca; FDA)?
- Il Sistema di presa deve avere proprietà per resistere a temperature elevate?
- Come saranno presentati gli oggetti al cobot (es. singolarizzati, orientati, ecc.)
- Il modo di presentazione degli oggetti è ripetibile?
- Ci sono altre operazioni che il cobot deve eseguire durante il processo? (per esempio: prelievo e posizionamento e pressione di un pulsante, ecc.)
- Quanti processi occorre eseguire e quanti oggetti occorre manipolare?
- Vi è l'esigenza di esercitare una forza attraverso l'EOAT come parte del processo (per esempio: torcere, avvitare, premere un pulsante, lucidare)?
- Quale livello di precisione è richiesto nel posizionamento al deposito?
- Sono necessari sensori aggiuntivi per lo stato del sistema di presa (es. segnali di presa pezzo)?
- Ci sono persone accanto al Cobot durante i processi, per quanto tempo? (sempre, saltuariamente, mai)

Queste sono alcune domande la cui risposta può determinare l'impiego di un EOAT piuttosto che un'altro.

### Pneumatico

I dispositivi sono alimentati da aria compressa che determina attraverso la pressione la forza del dispositivo e attraverso la portata dell'alimentazione la sua velocità.

L'impiego della pneumatica è estremamente comune nella produzione e viene utilizzata per tutta una serie di applicazioni di tipo pick and place.

Questo tipo di azionamento è in genere economico nei componenti ma richiede una linea d'aria o un compressore d'aria per funzionare in maniera continuativa.

L'azionamento pneumatico tende anche ad essere meno preciso di altri azionamenti.

D'altra parte, fornisce una presa forte utile per la movimentazione di pezzi da lavorare in pinza.

### Elettrico

L'azionamento elettrico permette la costruzione ad esempio di servo-pinze.

I vantaggi sono la precisione di posizionamento, il controllo della velocità, il controllo della forza. Tendenzialmente l'azionamento elettrico rende semplice la configurazione e l'integrazione.

L'azionamento elettrico rende i sistemi tipicamente più costosi.

### Idraulico

Gli attuatori idraulici presentano alcuni vantaggi in certe situazioni:

alta densità di potenza, basso costo di acquisto dei componenti, alta rigidità, alta velocità;

rappresentano una tecnologia comunemente usata in robotica in applicazioni che prevedono attività pesanti.

### Vuoto

Il vuoto viene impiegato in strumenti che usano ventose, piuttosto che ganasce, per prendere e spostare gli oggetti.

Adatto a pezzi non porosi, puliti e con una superficie uniforme alla quale la ventosa può aderire. Il vuoto come azionamento rende gli EOAT più leggeri e convenienti.

L'alimentazione a pressione negativa di solito si ottiene mettendo un tubo di Venturi (che si presenta sotto forma di collegamento in linea) all'estremità di una linea di aria compressa.

### Elettromagnetico

Le pinze elettromagnetiche (e le pinze a magneti permanenti) offrono una soluzione alternativa per le parti metalliche ferrose.

Questi tipi di utensili sono particolarmente utili per gli articoli che non sono uniformi e non hanno superfici piane che potrebbero altrimenti essere impegnative per le pinze con ganasce o ventose.

Particolare attenzione deve essere data alle parti che potrebbero essere sensibili ai campi elettromagnetici, come i componenti elettronici.



## TIPOLOGIE DI EOAT

Una volta che sono stati individuati i requisiti specifici e le complicazioni associate alle parti da manipolare, si può passare a selezionare il giusto tipo di utensile per il lavoro che occorre fare. Le variabili per la scelta degli EOAT sono ampie. Le opzioni vanno dagli strumenti pronti all'uso a soluzioni personalizzate costruite da specialisti dell'integrazione.

### PINZA A 2 GRIFFE A MOVIMENTO PARALLELO (PNEUMATICHE, ELETTRICHE)

Questa è una pinza semplice utilizzata in molte applicazioni (esempio immagine [ONROBOT](#)).

L'alimentazione può essere elettrica, pneumatica e anche idraulica. Il mercato fornisce un'ampia varietà di dimensioni, forze, pesi e intervalli di corsa.

Vengono generalmente utilizzate quando si afferrano le parti da due superfici parallele e piate.

Possono afferrare una parte sia con il movimento di chiusura che con il movimento di apertura della pinza.

Ci sono anche pinze angolari a 2 ganasce con dita girevoli a 90 gradi che si possono completamente retrarre e fornire un gioco extra che è vantaggioso per applicazioni particolari.



### PINZA A 3 GRIFFE (PNEUMATICHE, ELETTRICHE)

Disponibile anche in varianti angolari, questo dispositivo EOAT (esempio immagine [SCHUNK](#)) è comunemente utilizzato per prelevare parti cilindriche.

Questo tipo di pinza ha tre ganasce distanziate di 120 gradi l'una dall'altra.

Grazie ad un movimento all'unisono, "centra" la parte oltre a fornire una presa sicura.

Può anche afferrare sia un esterno che un interno in quanto la forza di chiusura è la medesima sia in apertura che in chiusura.



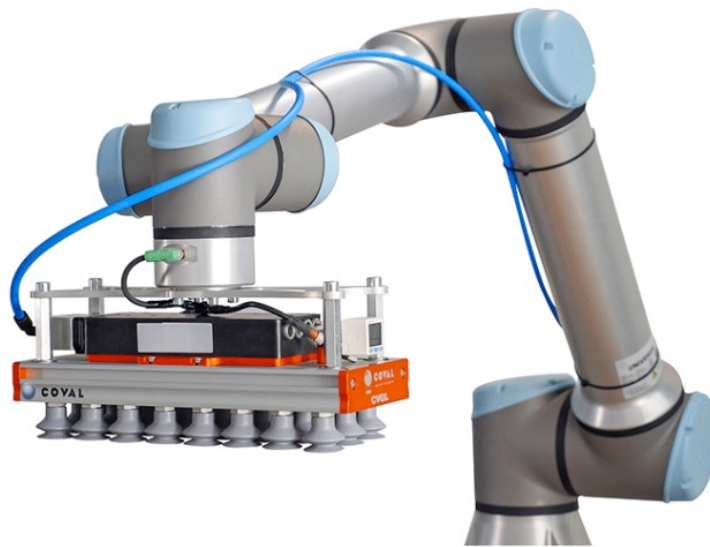
## VENTOSE (PNEUMATICHE)

La ventosa (esempio immagine [COVAL](#)) è uno strumento che aderisce a una superficie liscia sfruttando la pressione generata al suo interno.

Può essere di svariate forme, la più comune è quella circolare. Consiste essenzialmente in una camera in cui si viene a formare una depressione e il cui bordo è conformato in maniera da ottenere una buona tenuta al passaggio dell'aria.

Una volta appoggiata sulla superficie sulla quale si vuol fare aderire, la camera interna viene posta in depressione, o deformando il corpo della ventosa per far aumentare il volume della camera, o aspirando l'aria per mezzo di una pompa.

La forza con cui la ventosa aderisce alla superficie dipende dalla differenza fra la pressione interna e quella atmosferica e dalla superficie della ventosa. Impiego tipico per processi di imballaggio veloci che impongono requisiti particolare alle ventose in termini di tenuta e resistenza all'usura.



## PINZE AD ESPANSIONE (PNEUMATICHE)

Le pinze a espansione (esempio immagine [GIMATIC](#)) vengono utilizzate per afferrare superfici cilindriche dall'interno.

Una pinza a soffiutto utilizza una camera d'aria in elastomero per contattare la parte. Viene gonfiata con aria compressa provocando l'espansione e la presa della parte. Quando l'aria viene rilasciata, la camera d'aria si sgonfia e lascia cadere la parte.



## PINZA A ESPANSIONE (IDRAULICHE)

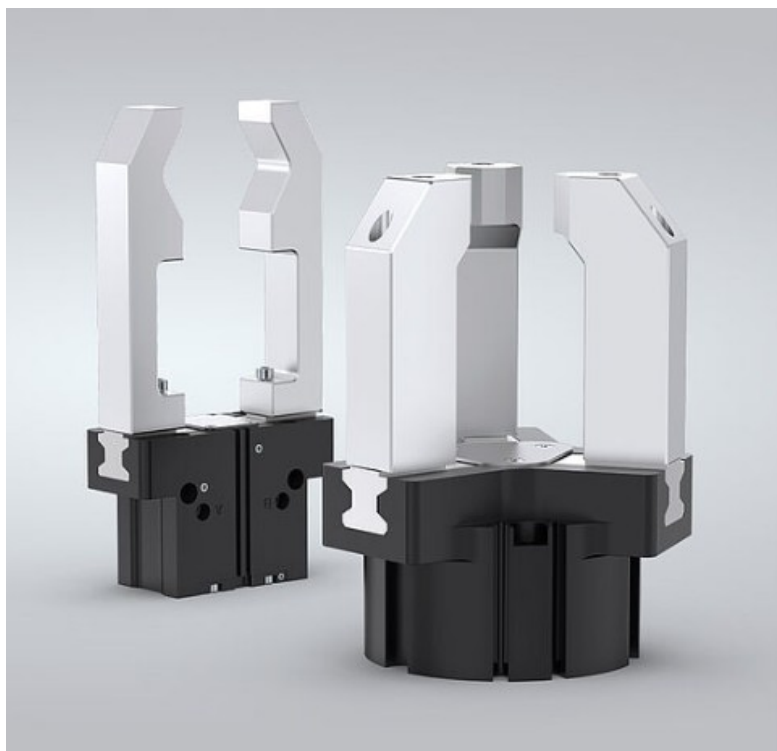
Questo dispositivo EOAT (esempio immagine [AMF](#)) è utilizzato per afferrare superfici cilindriche e parallele utilizzando un movimento lineare e una forma a cono per espandere quindi aprire o chiudere le «dita» metalliche per afferrare l'esterno o l'interno di una parte con la potenza dell'attuazione idraulica.



## PINZE O-RING (PNEUMATICHE, ELETTRICHE)

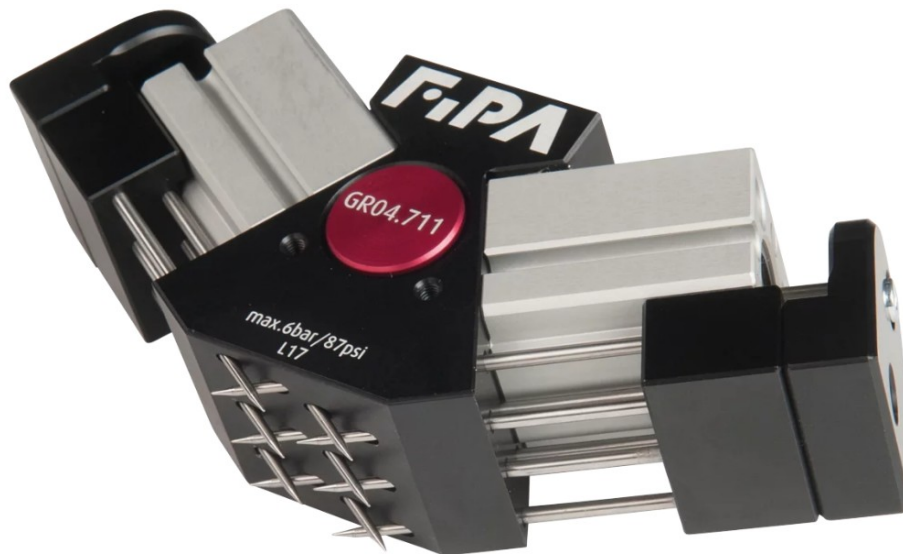
Questa attrezzatura (esempio immagine [ZIMMER Group](#)) per estremità del braccio è un dispositivo speciale progettato per gestire le guarnizioni O-ring. Simile a una pinza parallela a 3 ganasce, la pinza O-ring ha sei o talvolta otto dita che si espandono radialmente e afferrano il diametro interno dell'O-ring.

Una volta allungato, l'O-ring può essere posizionato dalla macchina in una scanalatura all'estremità di un albero cilindrico.



## PINZE AD AGHI (PNEUMATICHE, ELETTRICHE)

Questa è una variante di pinza (esempio immagine [FIPA](#)) è progettata per raccogliere materiali in fogli porosi o tessuti, come i tessuti in cotone, pelle, lana penetrando la superficie con più aghi affilati. Le pinze di presa ad aghi offrono la possibilità di afferrare pezzi difficili indipendentemente dalla loro forma. Aghi incrociati garantiscono una forza di presa elevata.



## VENTOSE CON GENERATORI DI VUOTO (ELETTRICHE)

Generatore di vuoto elettrico (esempio immagine [SCHMALZ](#)) per la movimentazione di pezzi ermetici o leggermente porosi. Tipicamente dotati di interfaccia integrata per il controllo e il monitoraggio del processo di gestione. Per l'uso nella robotica mobile, nella manipolazione di piccoli pezzi completamente automatizzata e nelle attività di movimentazione fissa. Esistono ventose per ogni applicazione, la selezione è fondamentale.



## VENTOSE FLOTTANTI (PNEUMATICHE)

Le ventose flottanti (esempio immagine [SCHMALZ](#)) funzionano in base al principio di Bernoulli e consentono una movimentazione senza contatto di pezzi sottili e delicati, quali ad es. pellicola, carta o celle solari.

La ventosa flottante genera un cuscinio d'aria sul quale è "sospeso" il pezzo.

Con l'ausilio di una portata elevata vengono compensate le perdite, in modo che sia possibile movimentare e separare pezzi anche porosi.



## SISTEMI DI PRESA MAGNETICI (AZIONAMENTO PNEUMATICO; ELETTRICO)

Questi tipi di pinze (esempio immagine [SMC](#)) hanno una presa sicura garantita da un magnete permanente.

Il design è compatto e leggero, ha requisiti minimi di spazio, basso peso totale a parità di carico ammesso.

È possibile ottenere forze di taglio elevate mediante inserti di vari materiali posti in prossimità della zona di contatto magnetico.

La presa di pezzi ferromagnetici viene effettuata, mediante il campo magnetico di un magnete permanente integrato.

Il magnete viene mosso mediante l'aria compressa, in maniera tale da attivare e disattivare la presa.

Le ventose magnetiche vengono azionate mediante valvole pneumatiche.



## PINZE ELETTROSTATICHE (ELETTRICHE)

Le pinze elettrostatiche (spesso chiamate electroadhesion grippers) sono dispositivi di presa che utilizzano l'attrazione elettrostatica per sollevare e manipolare oggetti senza bisogno di componenti meccanici mobili o sistemi a vuoto.

Le forze elettrostatiche sono la chiave per un tipo di pinza che può afferrare e manovrare oggetti di forma strana e fragili senza danneggiarli o segnarli.

Oggetti naturali come frutta e verdura hanno forme e dimensioni uniche e imprevedibili e spesso non devono essere danneggiati mentre vengono spostati. Le pinze di presa basate sull'elettrostatica offrono nuove soluzioni convincenti nella manipolazione robotica dei pezzi.

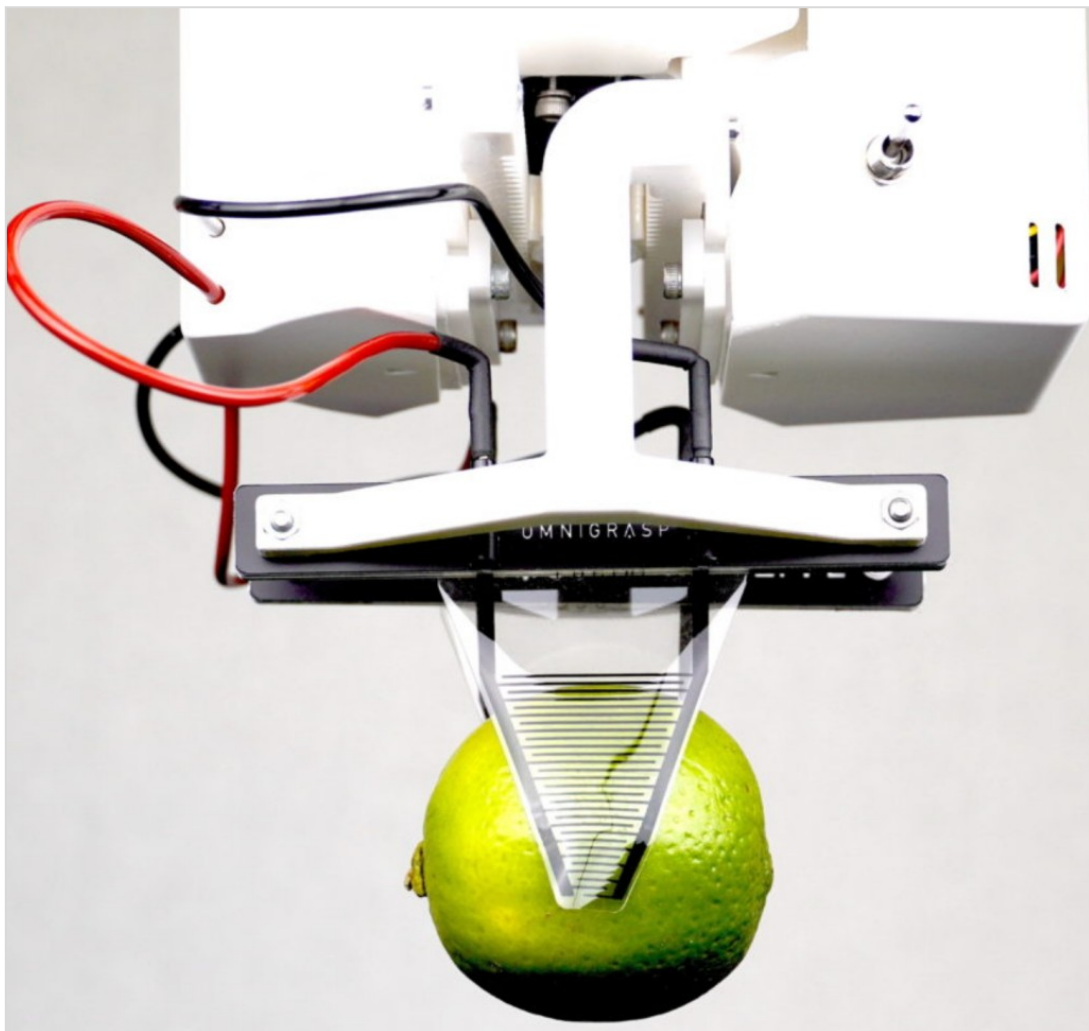
L'elettrostatica è una tecnologia altamente flessibile e a bassissima energia che elimina la necessità di una tecnologia del vuoto ingombrante e ad alto consumo energetico o di pinze meccaniche.

Per il costo energetico del funzionamento di una pinza a vuoto convenzionale per una settimana, una pinza a elettrodeposizione può funzionare per 10 anni.

Le pinze elettrostatiche offrono una soluzione flessibile per la manipolazione di smartphone e tablet, circuiti stampati (PCB), circuiti flessibili, tessuti e tessuti, pannelli solari, lastre di vetro, fogli di fibre composite pre-preg e lamiere.

Queste pinze eliminano la necessità di pinze specifiche per ogni pezzo, riducono i tempi morti necessari per il cambio delle pinze e soprattutto rendono possibili nuove applicazioni.

Questa tecnologia permette una manipolazione delicata senza graffi e sbavature grazie alla sua presa pulita elimina la necessità di rimuovere i residui lasciati dalle ventose.



Video: <https://omnigrasp.com/>

## PINZE ADATTIVE (ELETTRICHE)

Sono pinze tipicamente elettriche (esempio immagine [ROBOTIQ](#)) compatibili con tutti i principali robot industriali. La pinza adattiva a due/tre dita rende il robot ancora più preciso e sensibile, persino con le parti da afferrare più fragili e complesse.

Sono possibili tre modalità di presa:

- avvolgente,
- parallelo esterno
- parallelo interno.

Queste modalità permettono di afferrare parti dalle differenti geometrie, quali: piatto, quadrato, cilindrico, forato. Sfruttando poi il feedback di corrente possono anche fare una misura della forza impressa al serraggio.



Video: <https://robotiq.com/it/prodotti/pinze-adattive#Two-Finger-Gripper>

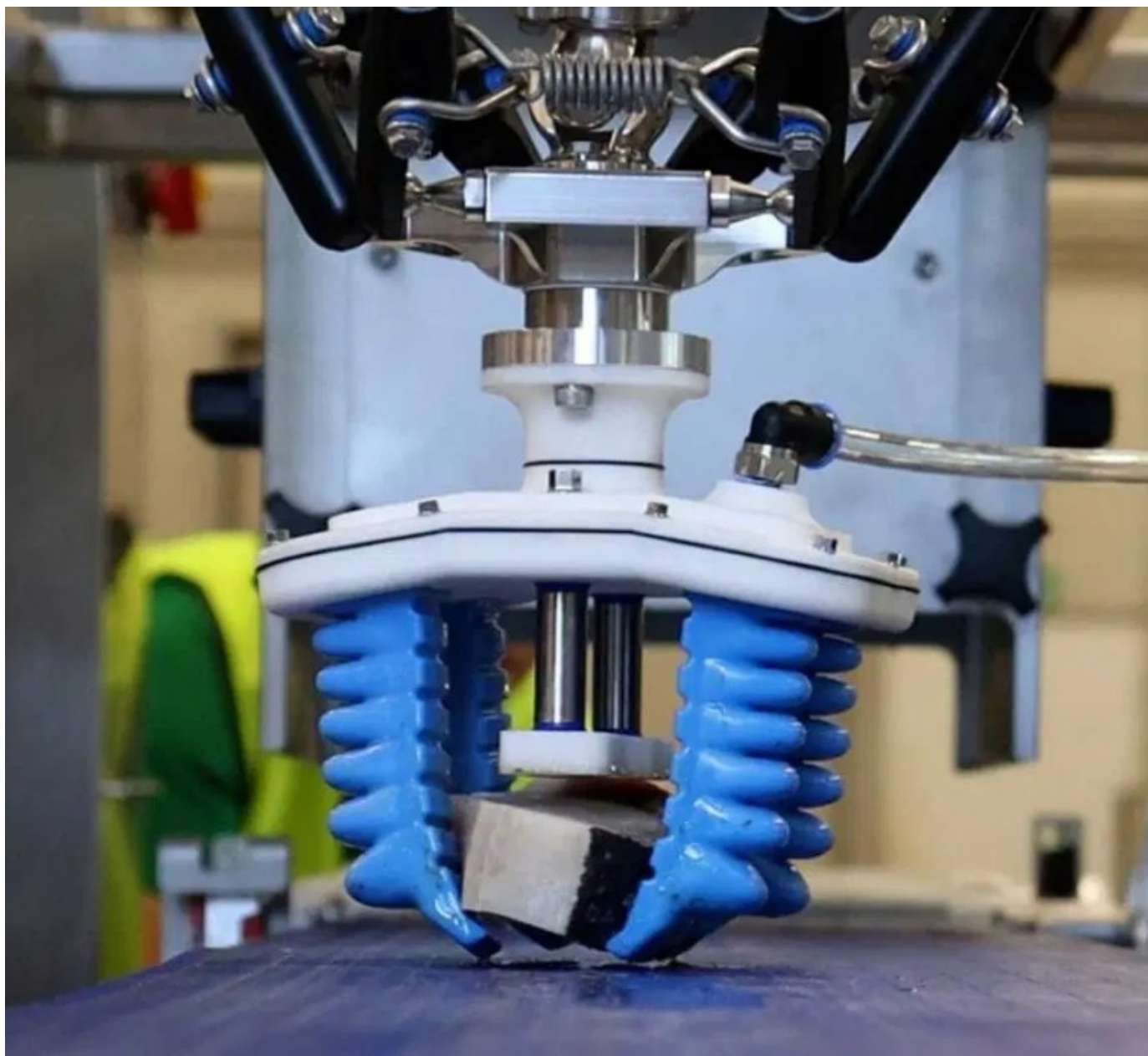
## SOFT GRIPPER (PNEUMATICHE)

Le pinze morbide (esempio [SOFT GRIPPING](#)) hanno guadagnato molta attenzione negli ultimi anni grazie ai loro vantaggi di facile fabbricazione, flessibilità e adattabilità.

I modelli teorici della pinza sono stabiliti per caratterizzare l'angolo di flessione, la forza di bloccaggio, la forza di attrito e la capacità di sopportare il peso.

Si tratta di attuatori in elastomero morbido azionati ad aria consentono applicazioni industriali che in precedenza erano off-limits per l'automazione. Si adattano alle forme applicando forze proporzionali alla pressione dell'aria o al vuoto.

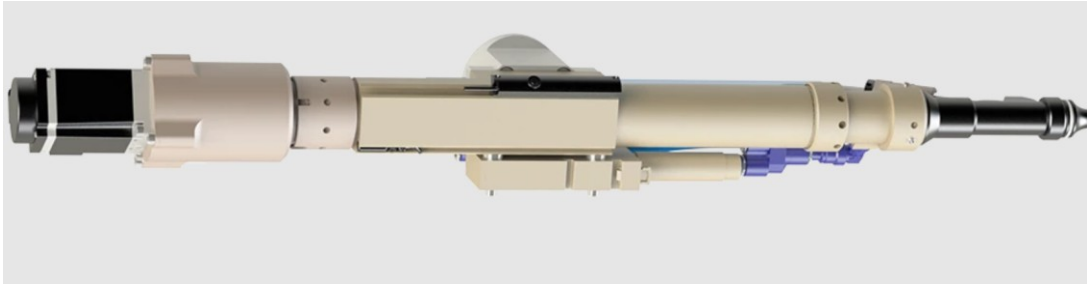
Adatti per manipolazione diretta del cibo ma non solo.



Video: <https://soft-gripping.com/it/>

## UTENSILI DI PROCESSO (ELETTRICI, PNEUMATICI)

In questo caso pensiamo al polso un utensile (esempio avvitatore della [WEBER](#) ) necessario per compiere una azione coadiuvato dal posizionamento del braccio robotico. Esistono infinite applicazioni, tra le più note oltre l'avvitatura, la rivettatura.



## UTENSILI PER LAVORAZIONI A CONTATTO (ELETTRICI, PNEUMATICI)

In questo caso pensiamo al polso un utensile (esempio [ONROBOT](#) ) necessario per compiere una azione coadiuvato dal movimento del braccio robotico che oltre a eseguire traiettorie deve esercitare una forza sul piano in lavorazione. Esistono infinite applicazioni, tra le più note la smerigliatura, la stinatura, la pulitura, la lucidatura.



## UTENSILI PER LAVORAZIONI SENZA CONTATTO (ELETTRICI, PNEUMATICI)

In questo caso pensiamo al polso un utensile (esempio [AIM ROBOTICS](#) ) necessario per compiere una azione coadiuvato dal movimento del braccio robotico che deve eseguire traiettorie definite nello spazio. Esistono infinite applicazioni, tra le più note la dosatura, incollaggio.

