

In questo tutorial per Arduino con shield Grove, proveremo a costruire una centralina di rilevamento dei sensori con display disegnati, usando il programma MakeBlock 5

## TUTORIAL IN-PROGRESS

Questo tutorial, i suoi file ed eventuali video li trovi qui: [coderdojotrento.it/grove2](https://coderdojotrento.it/grove2)

**IMPORTANTE:** prima di proseguire, leggi bene come usare mBlock qui: [coderdojotrento.it/grove1](https://coderdojotrento.it/grove1)

### Potenziometro - il sensore

#### Seleziona il dispositivo dell'Arduino

- attacca il potenziometro al canale A3

- crea una variabile, chiamala `angolo`

- metti questo codice

- clicca la bandierina verde, e prova a girare il potenziometro, il valore nella variabile `angolo` dovrebbe cambiare

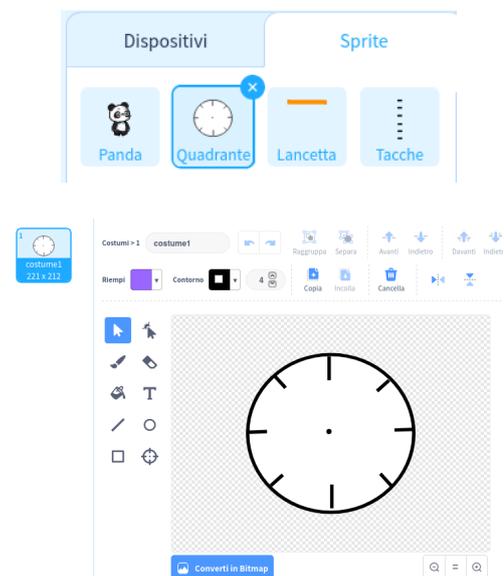


### Potenziometro - il quadrante

Crea un **NUOVO** sprite e chiamalo `Quadrante`

- Il quadrante non ha codice
- vai nei costumi e disegna il quadrante

**ATTENZIONE:** centrato dove c'è il mirino  !



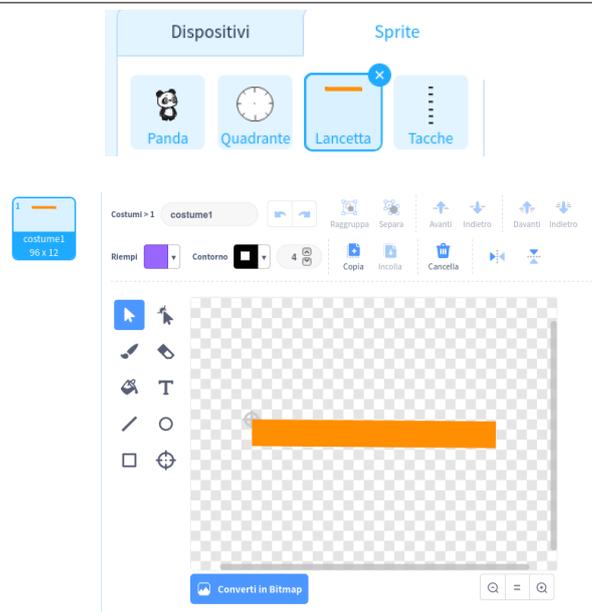
## Potenzionometro - disegniamo la lancetta

Crea un **NUOVO** sprite e chiamalo Lancetta

### ATTENZIONE:

- la lancetta va disegnata **IN ORIZZONTALE**

- il centro  deve essere **A SINISTRA**



## Potenzionometro - codice lancetta

Il potenziometro ci indicherà in che direzione puntare la lancetta

I valori letti del potenziometro dovrebbero andare da 0 a 1023, ma gli angoli vanno da 0 a 360, quindi dobbiamo fare una proporzione.

**ATTENZIONE:** il tuo potenziometro potrebbe avere un valore massimo diverso da 1023, potresti dover cambiare quel numero

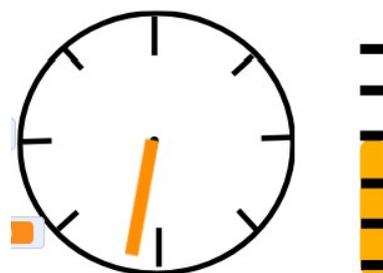


## Potenzionometro - provalo!

Clicca sulla bandierina verde, la lancetta dovrebbe girare in base a quanto ruoti il potenziometro !

## Display lineare

Proveremo a disegnare un altro display

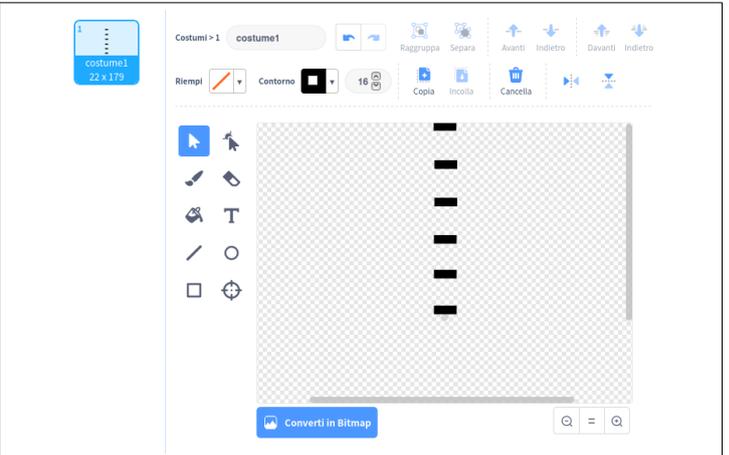


## Display lineare - sprite Tacche

- Crea uno sprite nuovo, chiamalo LineareTacche  
- Disegna delle tacche

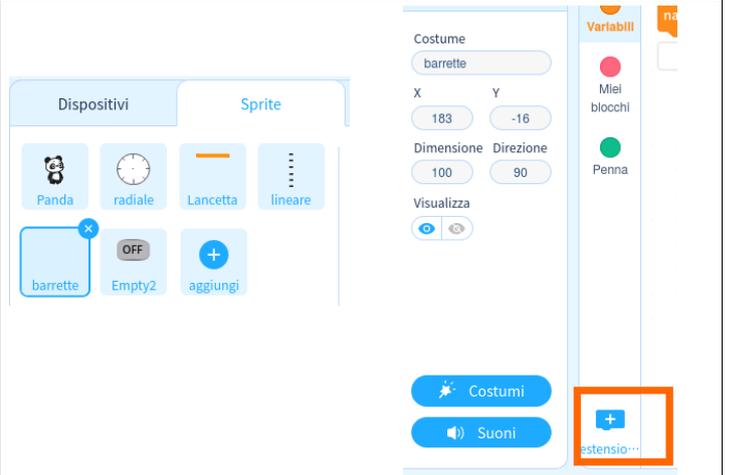


**ATTENZIONE:** il centro  deve essere in prossimità della prima tacca in basso



### Display lineare - Barretta

- Crea un **NUOVO** sprite, chiamalo Barretta
- lo sprite avrà un costume vuoto
- abilita l'estensione Penna, cliccando sul "+ estensioni" in basso in fondo



- nel codice dello sprite `barrette`, metti questo:

- crea una variabile `altezza_tacche` mettendo approssimativamente l'altezza in pixel dello sprite `Tacche`

- pulisci il tratto precedente sovrascrivendo con uno sfondo bianco

- disegna la barretta in arancione

**ATTENZIONE:** intanto usa `angolo`, la variabile cambiata dal potenziometro, per provare che funzioni  
In seguito cambieremo sensore e variabile.

```

quando cliccato
  imposta altezza_tacche a 175
  penna su
  elimina tutto
  per sempre
    vai al livello davanti
    penna su
    vai a Tacche
    usa penna di colore
    usa penna di dimensione 30
    penna giù
    cambia y di altezza_tacche
    penna su
    vai a Tacche
    usa penna di colore
    usa penna di dimensione 20
    penna giù
    cambia y di angolo * altezza_tacche / 1023
  
```

## Display lineare : Provalo!

Una volta scritto il codice sopra, clicca sulla bandierina verde e prova a ruotare il potenziometro, oltre allo sprite Lancetta dovrebbe aggiornarsi anche la Barretta

## Termometro - sensore

Adesso proviamo a far funzionare il display lineare con un altro sensore, il termometro

- collega il sensore di temperatura (*Temperature*) al canale A0

- aggiungi la variabile `temperatura` e aggiorna il codice del dispositivo Arduino così:



## Termometro – sensore – Provalo !

Clicca sulla bandierina verde e prova ad alitare sul termometro, guarda quanto varia la misurazione

## Termometro - Barretta

**Sfida:** Nello sprite *Barretta*, sostituisci questa linea di codice con un codice che usa la variabile `temperatura` appena creata. La scala usata per il potenziometro sicuramente non andrà bene e dovrai cambiarla, aggiungendo anche operazioni (il termometro potrebbe variare all'incirca da 550 a 600, ma i numeri per il tuo particolare termometro / stanza potrebbero variare sensibilmente)

- Una volta cambiata la linea, prova il tutto cliccando sulla bandierina verde e alitando sul termometro



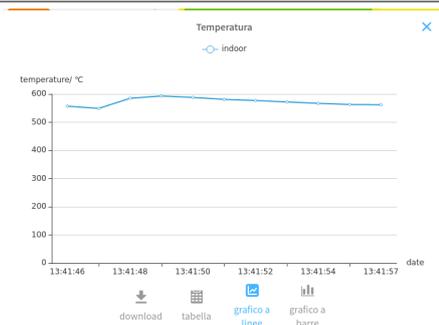
## Temperatura - grafico

- abilita l'estensione Grafico dati
- crea la variabile `stringa_ora`
- Nello sprite del panda, metti questo codice
- il grafico rileverà la temperatura ogni minuto



## Temperatura – Provalo!

Clicca sulla bandierina verde, dovrebbe apparire un grafico del genere che ogni secondo si aggiorna con un valore nuovo, prova ad alitare sul sensore per vedere se cambia



## Temperatura – Sfida Celsius

La temperatura viene mostrata in unità che variano circa da 500 a 600, ma in quale scala? E se volessimo mostrare il tutto nei familiari gradi centigradi, come potremo fare?

Per capirlo dobbiamo andare [nella wiki](#) del sensore di temperatura del Grove. In un esempio di codice C dell'Arduino troviamo delle formule, mostriamo qui le più rilevanti:

```
const int B = 4275; // B value of the thermistor
const int R0 = 100000; // R0 = 100k
```

- La variabile `a` corrisponde al valore che ottieni con `leggi pin analogico (A) 0` in mBlock
- La variabile finale `temperature` è la temperatura in Celsius che ci serve
- nelle formule è presente la funzione `log`, in mBlock lo trovi cambiando voce in questo blocco:

valore assoluto (modulo) ▼ di

**Sfida:** Prova a ricavare la temperatura in gradi Celsius e riportarla nel grafico in mBlock usando le formule dalla wiki.

```
int a = analogRead(pinTempSensor);

float R = 1023.0/a-1.0;
R = R0*R;

float temperature = 1.0/(log(R/R0)/B+1/298.15)-273.15;
```

## Luminosità

Carichiamo uno sfondo colorato, e facciamo variare la luminosità dello sfondo in base ai rilevamenti del sensore di luce

### Luminosità – il sensore

- collega il sensore di luce (*Light sensor*) all'Arduino sul canale A2



- crea la variabile `luce`

- modifica il codice nell'Arduino aggiungendo la riga per la variabile `luce`



### Luminosità – il sensore – Provalo !

Clicca sulla bandierina verde e prova a coprire il sensore con una mano, guarda quanto varia la misurazione

### Luminosità – lo sfondo

- Carica un NUOVO costume per lo sprite dello SFONDO



- nel codice dello SFONDO, metti questo codice

Dato un qualunque costume, è possibile cambiarne l'intensità luminosa cambiando l'effetto luminosità, i cui valori variano da -100 (completamente nero) a 100 (completamente bianco)

Per Arduino, nei nostri esperimenti, la variabile **luce** rilevata dai sensori è variata così (nel tuo ambiente potrebbero essere diverso):

- 0 sensore coperto con una mano
- 50 luce ambiente
- 800 quando esposto alla luce di una torcia

Nel codice riportiamo un valore basso di 50 supponendo che proverai a testare il sensore con una mano, se usi una torcia dovrai mettere un valore più alto intorno a 800



### Luminosità – Prova !

Clicca sulla bandierina verde: prova a coprire il sensore con una mano, lo sfondo dovrebbe diventare scuro

### Foglio Google

**NOTA: questa parte si può fare solo se hai un account Google**

Proviamo a scrivere tutti i valori letti su un foglio Google

### L'accendiamo?

Proviamo a creare un sistema mentre un bottone è premuto tiene accesa una luce LED collegata all'Arduino

### L'accendiamo? – bottone fisico

- attacca il pulsante del kit Grove (*Button*) al canale D8

- aggiungi questo codice



```

quando cliccato
per sempre
imposta angolo a leggi pin analogico (A) 3
imposta temperatura a leggi pin analogico (A) 0
imposta premuto a leggi pin digitale 8
attendi 0.01 secondi

```

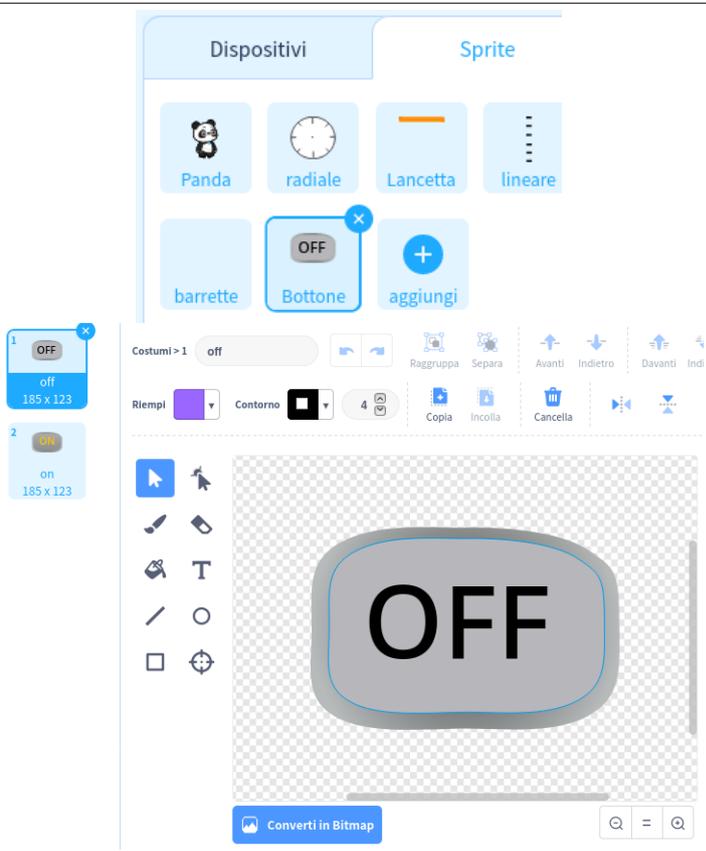
**L'accendiamo? - bottone fisico – Provalo!**

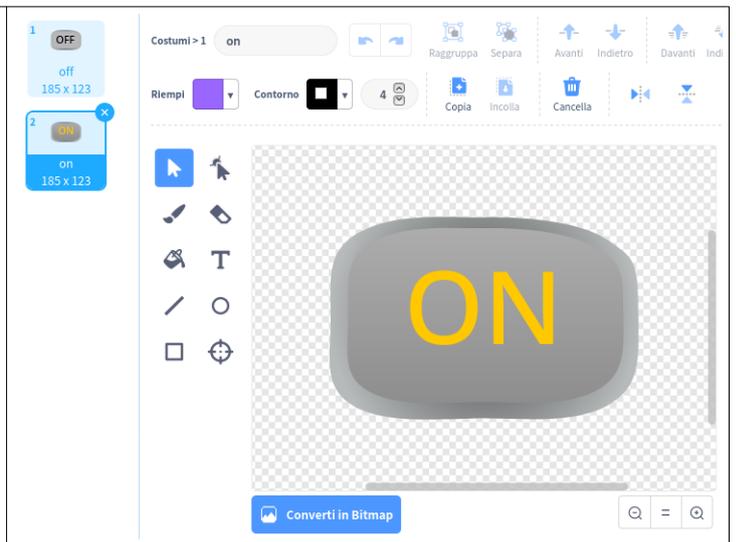
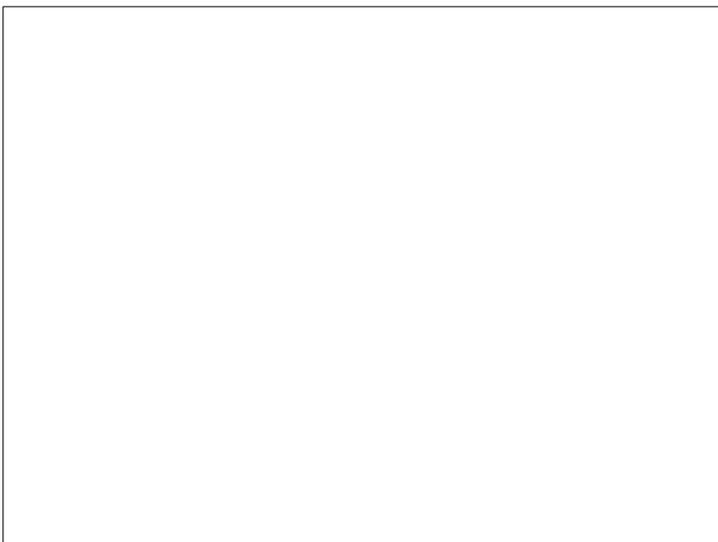
Clicca sulla bandierina verde e verifica che la variabile `premuto` si aggiorni quando premi il bottone

**L'accendiamo? - il bottone disegnato**

- Disegniamo un bottone creando un nuovo sprite
- nello sprite, disegna due costumi, uno chiamalo `off` e l'altro `on`

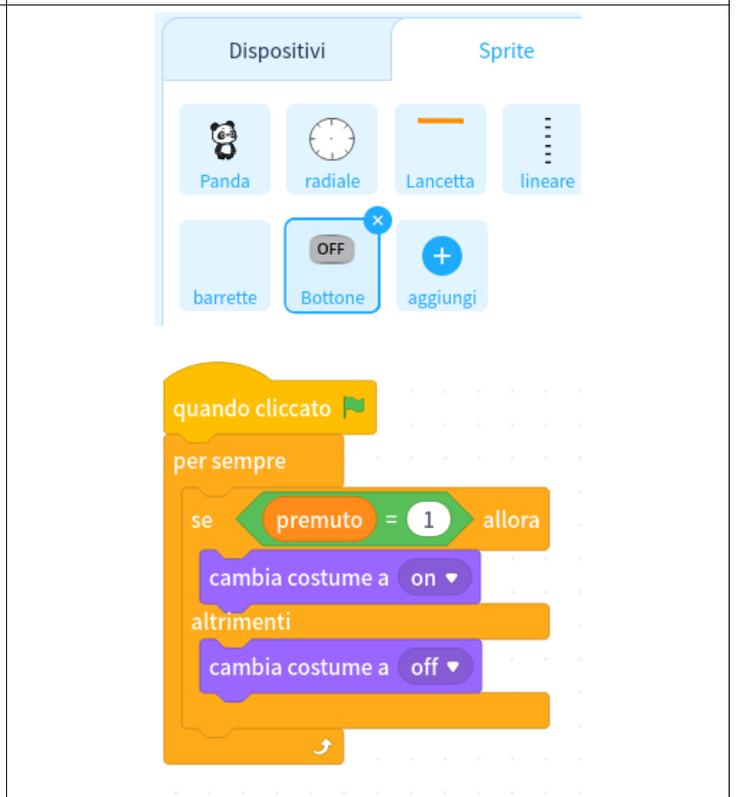
**NOTA: DEVI DISEGNARE UNO SPRITE AVENTE DUE COSTUMI**





**L'accendiamo? – bottone disegnato - codice**

Nello sprite del bottone, metti questo codice



**L'accendiamo? - LED**

Proviamo a collegare il LED fornito nel kit del Grove

- prendi l'attuatore LED (*Light Socket Kit*), all'inizio lo troverai senza lucetta sopra
- monta una luce LED nell'angolo in basso
- collega l'attuatore col cavetto sullo shield Grove in D2



**L'accendiamo? - codice Arduino**

Ora nel dispositivo dell'Arduino, modifica il codice aggiungendo istruzioni per accendere la luce quando il bottone è premuto.

### L'accendiamo lampeggiante

Come facciamo a far lampeggiare il LED SOLO QUANDO il bottone è premuto?

Il nostro programma dovrebbe fare due cose contemporaneamente:

- a) leggere come al solito i valori dai sensori
- b) accendere e spegnere il led a intervalli di un secondo, solo quando il bottone è premuto

**!!!! ATTENZIONE 1 !!!!** Per accendere il LED a intervalli, la soluzione naturale sembrerebbe usare il blocco `attendi 1 secondi`, ma questo

**BLOCCA** l'esecuzione del programma, e l'Arduino non potrà fare nient'altro, nemmeno cose semplici come leggere valori dai sensori !!!!

**!!!! ATTENZIONE 2 !!!!** Potremmo essere tentati di separare le due cose in due blocchi che inizino

con `quando cliccato`: purtroppo, in mBlock al

momento (v5.2) nel dispositivo Arduino **NON** possiamo usare blocchi di codice eseguiti contemporaneamente, quindi **MAI** metterci due `Quando cliccato`

(Nota: negli Sprite invece possiamo usare quanti  
Quando cliccato vogliamo)

### L'accendiamo lampeggiante – codice

Per ovviare ai problemi descritti sopra, possiamo:

- usare un solo Quando cliccato e un solo per sempre
- far lampeggiare il cronometro nei secondi pari

```
quando cliccato
per sempre
imposta angolo a leggi pin analogico (A) 3
imposta temperatura a leggi pin analogico (A) 0
imposta premuto a leggi pin digitale 8
attendi 0.01 secondi
se premuto = 1 allora
se arrotonda resto della divisione di cronometro diviso 2 = 0 allora
imposta il pin digitale 2 uscita come high
altrimenti
imposta il pin digitale 2 uscita come low
altrimenti
imposta il pin digitale 2 uscita come low
```

### L'accendiamo – Provalo!

Clicca sulla bandierina verde, e premi il bottone verde: la luce dovrebbe lampeggiare. Mentre tieni premuto, prova anche a girare il potenziometro, la lancetta dovrebbe girare (perché le variabili si stanno aggiornando)

### L'accendiamo – Sfida

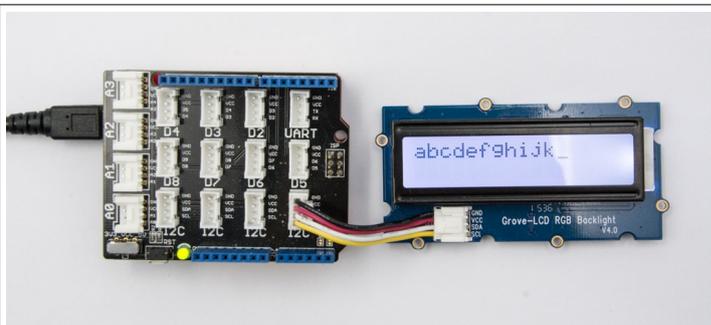
Modifica il codice precedente in modo che una volta premuto e rilasciato il bottone le luci lampeggino da sole per 6 secondi.

Come prima, le variabili devono potersi aggiornare (verifica girando il potenziometro dopo la pressione)

**SUGGERIMENTO:** per tener conto del tempo, usa la variabile `cronometro`. Poi potresti azzerare il cronometro, o meglio ancora salvare il momento in cui viene premuto il bottone in una variabile, che potresti chiamare `tempo_accensione`.

## Display LCD: collegamento

Attacca il display ad una delle porte denominate I2C in basso



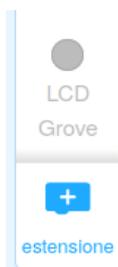
## Display LCD: l'estensione

Nella sezione Device clicca su Estensioni e cerca 'Grove', tra le varie estensioni troverai 'LCD Grove' di paulcoiffier, aggiungila



## Display LCD: i blocchi

Adesso troverai una nuova sezione con dei blocchi in grigio (e in francese :-)) per impostare il display



**NOTA: questi blocchi funzionano SOLO nella modalità 'Carica file'**

Potremmo:

- **Mettere Quando si avvia Arduino Uno**

**ATTENZIONE: NON 'Quando si clicca bandierina verde'**

- ripulire lo schermo del display
- definire il viola come colore di sfondo (componenti Rosso Verde Blu che vanno da 0 a 255)
- dentro un ciclo, proviamo a leggere un angolo dal potenziometro collegato in D3 e a riportare sul display l'angolo (*Afficher texte*) sulla linea in cima al display (*ligne 0*)

Una volta completato il programma, clicca Carica

File:

Cambia Modalità ?

Carica File Dal vivo

Carica

## Motori

Ci sono vari motori servo, tipicamente ne trovi che ruotano da 0 a 180 gradi.

Qui in particolare **consideriamo SOLO il caso di un motore 'piccolo e leggero'**.

**!!! ATTENZIONE !!!:** in genere, i motori non vanno attaccati direttamente alle uscite dell'Arduino perché lo possono danneggiare sottraendogli la corrente necessaria al suo funzionamento.

Per gestire correttamente i motori servirebbe una scheda separata chiamata *motor shield* e batterie, in assenza delle quali per il solo fine di questo mini-esperimento dobbiamo limitarci a un solo motore 'piccolo e leggero' che dovrai lasciare girare liberamente.

**NON sottoporre a sforzi il motore** (niente mani sull'elica!) - più si sforza e più sottrae corrente al nostro prezioso Arduino che potrebbe danneggiarsi.

Detto tutto ciò, attacca un motore servo in D6 e aggiungi il codice seguente per farlo girare quando giri il potenziometro.

- Quando hai finito, clicca Carica file:

Cambia Modalità ?

Carica File Dal vivo

Carica

e prova a girare il potenziometro



```
quando si avvia Arduino Uno
  Effacer texte
  Définir couleur fond : R 255 V 0 B 255
  per sempre
    imposta angolo a leggi pin analogico (A) 3
    Afficher texte angolo sur ligne 0
    imposta l'angolo di servo pin 6 come angolo / 1023 * 180
    attendi 0.01 secondi
```



Ti è piaciuto ? Trovi questo e altri tutorial gratuiti  
sul sito di *CoderDojo Trento* alla sezione *Risorse* !  
[coderdojotrento.it](http://coderdojotrento.it)

**Ringraziamenti:** Questo tutorial è stato realizzato grazie al sostegno  
di CoderDolomiti APS [coderdolomiti.it](http://coderdolomiti.it)

