



**Attribuzione - Non commerciale -  
Condividi allo stesso modo 4.0  
Internazionale (CC BY-NC-SA 4.0)**

## **LA BRACHISTOCRONA**

### **OBIETTIVI**

- 1.** Dimostrare che la brachistocrona è il percorso di maggior velocità per arrivare da un punto A ad una certa altezza ad un punto B più in basso, ma non direttamente sotto.
- 2.** Dimostrare che la brachistocrona è anche tautocrona, ovvero da qualunque punto si parta sulla curva il tempo impiegato a raggiungere la fine è costante.

### **MATERIALI**

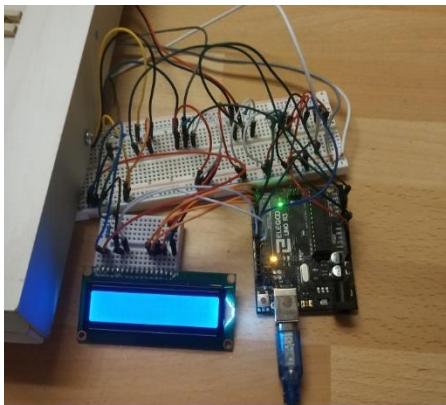
#### **- CIRCUITO**

- Scheda Arduino Uno
- 2 Pulsanti
- 3 Fotorisistori
- Servomotore
- Display LCD
- Cavi, resistenze

#### **- STRUTTURA**

- Pannelli in legno
- Lastra metallica
- Fili di ferro e spago
- Pezzetti di cartoncino
- Biglie

## IMMAGINI



Circuito



Meccanismo di partenza



Display lcd



Struttura completa

## FUNZIONAMENTO

Quando viene caricato il programma accende lo schermo lcd con tre caselle vuote per registrare i tempi delle biglie e il servomotore, posizionato in cima alle rampe si posiziona a un'angolazione tale da poter appoggiare le biglie su un filo di ferro (legato al motore) pronte a partire. Premuto il primo pulsante il motore gira di 90° sollevando l'asta di ferro e facendo partire le biglie contemporaneamente. Con l'azionamento del motore viene fatto partire un timer interno che viene interrotto con l'arrivo della biglia in fondo al percorso.

In fondo ad ogni rampa è posizionato un fotosensore (determina l'intensità di luce a cui è sottoposto), che col passaggio della biglia viene coperto da un'aletta di cartoncino che lo copre oscurandolo; questo, quando l'intensità luminosa diminuisce del 25% ferma il timer della determinata biglia e il tempo viene mostrato sullo schermo a lcd. Quando si preme nuovamente il pulsante, i timer vengono azzerati, lo schermo cancellato e il motore si riposiziona pronto a tenere ferme le biglie.

Il secondo esperimento presenta dinamiche diverse e necessita quindi un codice diverso, per passare al secondo esperimento basta premere il secondo pulsante. Nel secondo esperimento si usa una biglia per volta e la si posiziona inizialmente in cima alla brachistocrona e successivamente

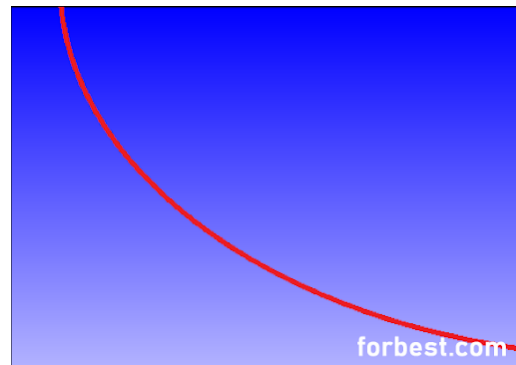
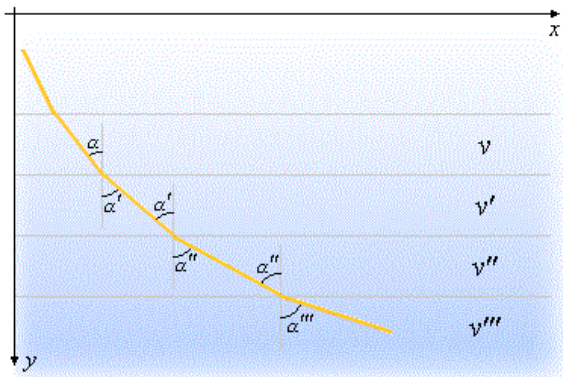
a metà rampa per osservare la costanza del tempo di discesa dalla curva. Ciò che cambia a livello di Arduino è che l'unico sensore che registra dati è quello della brachistocrona e i dati sullo schermo vengono eliminati dopo tre misurazioni (così da poter verificare che impiega lo stesso tempo da ogni posizione).

### SPIEGAZIONE FISICA DELLA BRACHISTOCRONA

La brachistocrona è stata individuata come traiettoria curva più veloce per andare da un punto A ad un punto B posto ad un'altezza inferiore ad A attraverso un parallelismo tra una biglia che accelera continuamente per azione della forza di gravità e la legge di Snell (la rifrazione della luce).

È stato determinato che la luce impiega sempre il minor tempo possibile per percorrere una certa distanza, questo comportamento si osserva soprattutto nella rifrazione. Quando la luce passa in un mezzo in cui viaggia più velocemente non viaggia in modo lineare, ma devia di un certo angolo  $\theta_2$ , per cui  $\sin(\theta_1) * n_1 = \sin(\theta_2) * n_2$ , proprio perché è la traiettoria più veloce per arrivare a quel punto.

Se consideriamo un pallina che aumenta velocità ogni frazione infinitesimale di secondo come della luce che attraversa un'infinita quantità di materiali nei quali viaggia sempre più veloce, allora per la legge di Snell otteniamo una curva come quella in figura, ovvero una brachistocrona.



### CODICE ARDUINO BRACHISTOCRONA

Thomas Fensterer

```
#include <Servo.h>
#include <LiquidCrystal.h>
```

```
Servo lever;
LiquidCrystal screen(12, 11, 5, 4, 3, 2);
```

## Thomas Fensterer SBS a.s. 2021-22

```
int cooldown = 0, timewriting = 0, cooldown2 = 0, startingpoint = 0;
int lightcal1, lightval1, lightcal2, lightval2, lightcal3, lightval3, resultnumber = -1, expcool = 0;
bool buttons핀, experimentnumber, rotate = false, timered1, timeredcheck1 = false, timered2;
bool timeredcheck2 = false, timered3, timeredcheck3 = false, experiment = true, rotateonce = false;
float timer = 0.0, finaltime1, finaltime2, finaltime3;
```

```
void setup() {
  Serial.begin(115200);
  screen.begin(16, 2);
  pinMode(9,INPUT);
  lever.attach(8);
  lever.write(0);
  screen.setCursor(0,0);
  screen.print(" t1  t2  t3 ");
}
```

```
void loop(){
  buttons핀 = digitalRead(9);
  if(buttons핀 == HIGH){
    if(rotate == false && cooldown == 0){
      rotate = true;
      rotateonce = true;
    }
    else if(rotate == true && cooldown == 500){
      rotate = false;
      rotateonce = true;
    }
  }
  experimentnumber = digitalRead(7);
  if(experimentnumber == HIGH){
    if(experiment == true && cooldown2 == 0){
      experiment = false;
      startingpoint = 1;
    }
    else if(experiment == false && cooldown2 == 2000){
      experiment = true;
      startingpoint = 2;
    }
  }
  //ESPERIMENTO BRACHISTOCRONA
  if(experiment == true){
    if(cooldown2 > 0 && startingpoint == 2){
```

## Thomas Fensterer 5BS a.s. 2021-22

```
    cooldown2--;
}
if(rotate == true){
    if(rotateonce == true){
        lightcal1 = analogRead(A0);
        lightcal2 = analogRead(A1);
        lightcal3 = analogRead(A2);
        lever.write(90);
        rotateonce = false;
    }
    timer++;
    if(timeredcheck1 == false){
        timered1 = false;
    }
    else{
        timered1 = true;
    }
    if(timeredcheck2 == false){
        timered2 = false;
    }
    else{
        timered2 = true;
    }
    if(timeredcheck3 == false){
        timered3 = false;
    }
    else{
        timered3 = true;
    }
    if(cooldown < 500){
        cooldown++;
        delay(1);
    }
    else{
        delay(1);
    }
}
else{
    screen.setCursor(0,1);
    screen.print("      ");
    if(rotateonce == true){
        lever.write(0);
        rotateonce = false;
    }
}
```

## Thomas Fensterer 5BS a.s. 2021-22

```
    }
    timer = 0;
    timered1 = true;
    timeredcheck1 = false;
    timered2 = true;
    timeredcheck2 = false;
    timered3 = true;
    timeredcheck3 = false;
    if(cooldown > 0){
        cooldown--;
        delay(1);
    }
}
lightval2 = analogRead(A1);
if(lightval2 < (0.75*lightcal2) && timered2 == false){
    finaltime2 = (timer*1.36)/1000;
    screen.setCursor(6,1);
    screen.print(finaltime2);
    timeredcheck2 = true;
}
lightval1 = analogRead(A0);
if(lightval1 < (0.75*lightcal1) && timered1 == false){
    finaltime1 = (timer*1.36)/1000;
    screen.setCursor(0,1);
    screen.print(finaltime1);
    timeredcheck1 = true;
}
lightval3 = analogRead(A2);
if(lightval3 < (0.75*lightcal3) && timered3 == false){
    finaltime3 = (timer*1.36)/1000;
    screen.setCursor(12,1);
    screen.print(finaltime3);
    timeredcheck3 = true;
}
}
// ESPERIMENTO TAUCRONA
else{
    if(cooldown2 < 2000 && startingpoint == 1){
        cooldown2++;
    }
    if(rotate == true){
        if(rotateonce == true){
            lightcal2 = analogRead(A1);
```

## Thomas Fensterer 5BS a.s. 2021-22

```
    lever.write(90);
    rotateonce = false;
  }
  timer++;
  if(timeredcheck2 == false){
    timered2 = false;
  }
  else{
    timered2 = true;
  }
  if(cooldown < 500){
    cooldown++;
    delay(1);
  }
  else{
    delay(1);
  }
  }
  else{
    if(resultnumber == 2){
      screen.setCursor(0,1);
      screen.print("      ");
      resultnumber = -1;
    }
    lever.write(0);
    timer = 0;
    timered2 = true;
    timeredcheck2 = false;
    if(cooldown > 0){
      cooldown--;
      delay(1);
    }
  }
  lightval2 = analogRead(A1);
  if(lightval2 < (0.75*lightcal2) && timered2 == false){
    finaltime2 = (timer*1.36)/1000;
    if(resultnumber < 3){
      resultnumber++;
    }
  }
  else{
    resultnumber = 0;
  }
  screen.setCursor((resultnumber*6),1);
```

