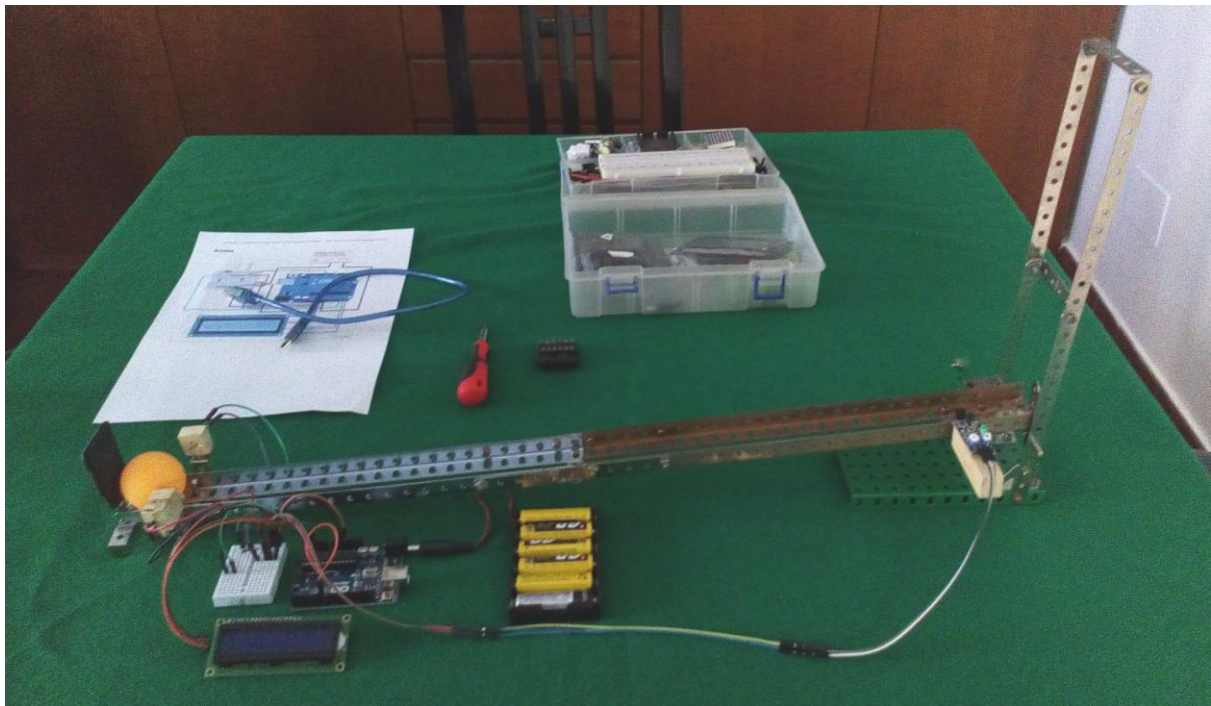


## 29bis- tempo, spazio ed una pallina che corre sui binari – time, space and a ball running on tracks (some notes at end of this section)



In questo progetto si utilizzano un modulo avoidance, un led ed una fotoresistenza per misurare il tempo che impiega una pallina per percorrere un tratto di piano inclinato. Maggiori dettagli sulle caratteristiche del modulo avoidance sono presenti nella scheda del progetto 29, reperibile [qui](#), mentre la fotoresistenza e' trattata nel progetto 09, reperibile [qui](#).

La parte "meccanica" e' composta da un binario, inclinabile a piacere, sul quale sono posizionate due stazioni di rilevamento. La prima stazione, ovvero al stazione di "start" e' un modulo avoidance piazzato in prossimita' del punto di partenza della pallina e quindi quando la pallina transita da detta stazione ha una velocita' vicina allo zero, mentre la seconda stazione, composta da un led e da una foto resistenza, e' posizionata in prossimita' della fine del binario, e quindi quando la pallina transita da questa seconda stazione e' nel punto di massima velocita'.

Arduino conosce la distanza intercorrente tra le due stazioni (0,53 metri) per cui, e' in grado di esporre non solo il tempo impiegato dalla pallina per compiere il tragitto, ma anche di calcolarne accelerazione e velocita' finale. Considerando, con un po' di elasticita', che il moto della pallina e' un moto rettilineo uniformemente accelerato, e' stata utilizzata la seguente formula per calcolare l'accelerazione:

$$\text{accelerazione} = \text{spazio}/(0,5*\text{tempo}^2)$$

mentre la velocita' finale e' stata calcolata con la classica formula:

$$\text{velocita'} = \text{accelerazione} * \text{tempo}$$

Variando l'inclinazione dei binari e' possibile verificare la variazione di tempo, velocita' ed accelerazione in funzione dell'inclinazione.

Prima di procedere alla compilazione del programma deve essere installata, se non gia' presente, la seguente libreria:

- LiquidCrystal\_I2C.h reperibile in <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>

## arduino – avoidance: tempo, spazio ed una pallina sui binari – time, space and a ball running on tracks

Per installare la libreria e' necessario seguire la procedura illustrata nei precedenti progetti, e sintetizzabile in:

- download della libreria in formato complesso
- installare la nuova libreria andando in IDE-> sketch-> includes Library-> add .zip library
- verificare l'avvenuta installazione (andando in IDE-> sketch-> includes Library-> Contributed library)

**Nota:** Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

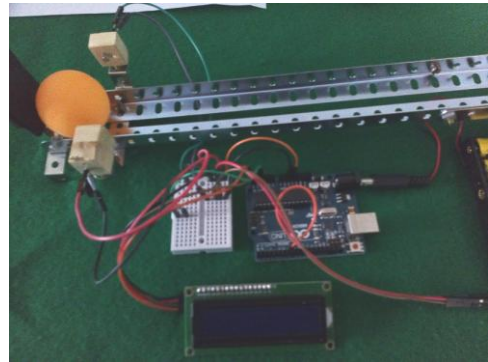
- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a [giocarduino@libero.it](mailto:giocarduino@libero.it)

**Here some notes about this project, translated by google translator**



In this project, an "avoidance" module, a LED, and a photoresistance are used to measure time used by a ball to traverse an inclined plane. More details about "avoidance" module are found in project 29, available [here](#), while photoresistance is treated in project 09, available [here](#). The plant consists by an inclined tracks, on which two detection stations are located. The first station, or the "start" station, is an "avoidance" module located near the starting point, so, when the ball passes, has a velocity close to zero, while the second station, consisting of a led and a photoresistor, is positioned at end of tracks and so, when the ball passes from this second station, is at its highest speed.



Arduino knows distance between stations (0.53 meters), so it's able to expose not only time spent by ball to make journey, but also to calculate its acceleration and final speed. Considering, with some elasticity, that motion is an uniformly accelerated straight motion, the following formula has been used to calculate acceleration:

$$\text{acceleration} = \text{space} / (0,5 * \text{time}^2)$$

while the final velocity is calculated using the classic formula:

$$\text{velocity} = \text{acceleration} * \text{time}$$

By varying tracks inclination, you can check variation about time, speed and acceleration.

Warning: before proceeding to program compilation must be installed, if not already done, the libraries:

- LiquidCrystal\_I2C.h found [here](#)

For library installation, see process shown in previous projects, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

**Note:** This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

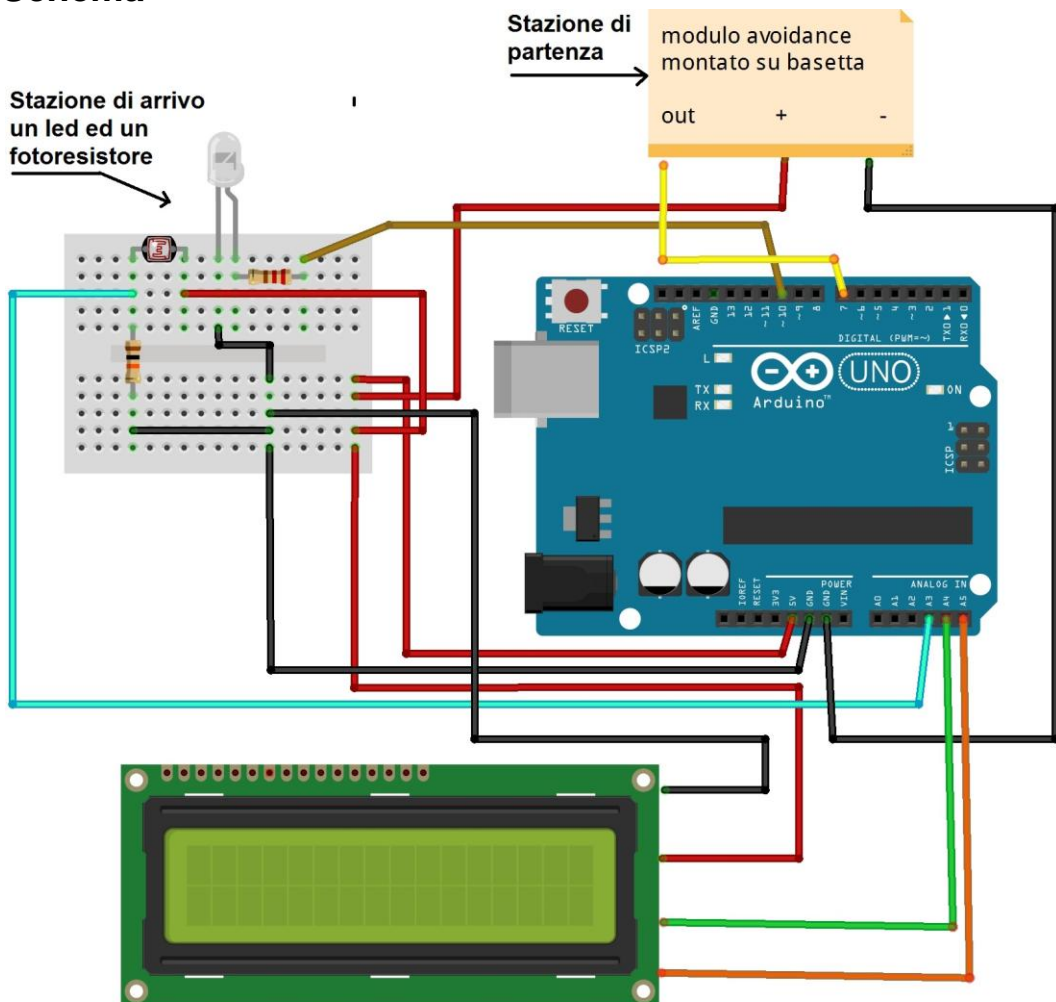
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to [giocarduino@libero.it](mailto:giocarduino@libero.it) (simple words and short sentences, please)

## Materiali

- Un modulo “avoidance” montato su basetta keys
- Una resistenza da 220 ohm
- Una resistenza da 10k ohm
- Un led a luce bianca
- Una fotoresistenza
- Un display lcd con driver I2C (facoltativo)
- una pallina ed un binario inclinabile
- Un po' di cavetteria

## Schema



## Programma

```

/*Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per
rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
trasferimento nell'IDE, premendo CTRL+T.
Questo programma utilizza un modulo avoidance e cioe' un modulo con un trasmettitore ed
un ricevitore di raggi infrarossi, ed un fotoresistore eccitato dalla luce di un led, per
misurare il tempo impiegato da una pallina per percorrere un piano inclinato
Lo schema vede il pin "out" del modulo avoidance collegato alla porta 7 di arduino, il
fotoresistore collegato alla porta analogica 3, un led collegato alla porta 10 ed un display lcd
con driver I2C i cui pin SDA e SCL sono rispettivamente collegati alla porta A4 ed A5
Questo progetto puo' essere replicato anche se non si dispone di un display lcd poiche' le
informazioni sono riportate anche sul monitor seriale. La stazione di partenza, infine, puo'
essere sostituita da un fotoresistore e da un led
-----
This program uses: an "avoidance" module, that is a module with an infrared transmitter/receiver,
and a photoresistor excited by a led, to measure time taken by a ball to move on an inclined
plane. The schematik sees the "out" pin of the "avoidance" module connected to Arduino pin 7, the
photoresistor connected to pin A3, a led connected to pin 10, and an LCD display with I2C drivers
whose SDA and SCL pins are respectively connected to A4 and A5 pins. This project can be
replicated even if you don't have an lcd display as information are also shown on serial monitor.
The starting station, finally, can be replaced by a photoresistor and a led.
Warning: cut&paste from PDF to IDE loses formatting. To restore it press CTRL + T.
*/

#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd
#define stazione1 7 // il pin "out" del modulo avoidance (la stazione di start) e'
// collegato alla porta 7
#define pinled 10 // il led di illuminazione della fotoresistenza e' collegato alla porta 10
// addr en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // definisce la tipologia del display
int pallinainmovimento = 0; // 0 = pallina ferma; 1 = pallina in movimento
int statostazione1 = 0; // variabile di memorizzazione dello stato (LOW oppur HIGH) del
// modulo avoidance
int statostazione2 = 0; // valore analogico generato dalla fotoresistenza
float momentoinizio = 0; // momento del passaggio dalla stazione di partenza
float momentofine = 0; // momento del passaggio dalla stazione di arrivo
float tempo = 0; // tempo (millisec) impiegato per percorrere il tragitto
float spazio = 0.530; // lunghezza del percorso, in metri
float accelerazione = 0; // accelerazione in metri al secondo quadrato
float velocita = 0; // velocita' finale, in metri al secondo
//
// ***** routine di calcolo ed esposizione dei risultati *****
// ***** computing and shows results routine *****
//
void esponidati ()
{
  tempo = (momentofine - momentoinizio) /1000; // tempo in secondi
  accelerazione = (spazio / (0.5 * (tempo * tempo)));
  velocita = (accelerazione * tempo);
  Serial.print ("tempo (in secondi): ");
  Serial.print (tempo);
  Serial.print (" velocita' finale (metri al secondo): ");
  Serial.print (velocita);
  Serial.print (" accelerazione (metri al secondo quadrato): ");
  Serial.println (accelerazione);
  lcd.clear ();
  lcd.print ("tempo: ");
  lcd.print (tempo);
  lcd.setCursor (0, 1);
  lcd.print ("v: ");
  lcd.print (velocita);
  lcd.print (" a: ");
  lcd.print (accelerazione);
  delay (6000); // attende sei secondi, per consentire la lettura dei risultati
  lcd.clear ();
  lcd.print ("unita di misura:");
  lcd.setCursor (0,1);
  lcd.print ("metri e secondi");
}
//
//
void setup()
{
  lcd.begin(16, 2);

```

## arduino – avoidance: tempo, spazio ed una pallina sui binari – time, space and a ball running on tracks

```
Serial.begin(9600);           // inizializza il monitor seriale
pinMode (stazione1, INPUT);   // il sensore viene definito come unita' di input
pinMode (pinled, OUTPUT);
digitalWrite (pinled, HIGH);
lcd.clear ();
lcd.print ("unita di misura:");
lcd.setCursor (0,1);
lcd.print ("metri e secondi");
}
//
//
void loop()
{
  statostazione1 = digitalRead (stazione1); // rileva lo stato della stazione di partenza
  statostazione2 = analogRead (A3);        // rileva lo stato della stazione di arrivo
  if (pallinainmovimento == 0)           // se non e' ancora iniziata la corsa della pallina
  {
    if (statostazione1 == LOW)            // se e' iniziata la corsa
    {
      pallinainmovimento = 1;             // setta il semaforo "pallina in movimento"
      momentoinizio = millis ();          // memorizza il momento di partenza della pallina
    }
  }
  else                                     // se la pallina era gia' in movimento
  {
    if (statostazione2 < 512)             // se la pallina e' passata dalla stazione di arrivo
    {
      momentofine = millis ();            // memorizza il momento di arrivo
      pallinainmovimento = 0;             // predisporre il semaforo per il prossimo giro
      esponidati ();                      // lancia la routine di calcolo ed esposizione dei dati
    }
  }
}
```