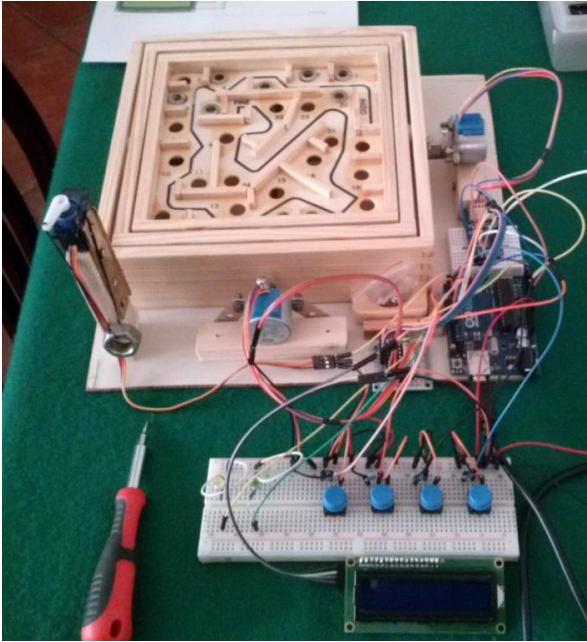


E – labirinto – cronaca di un fallimento



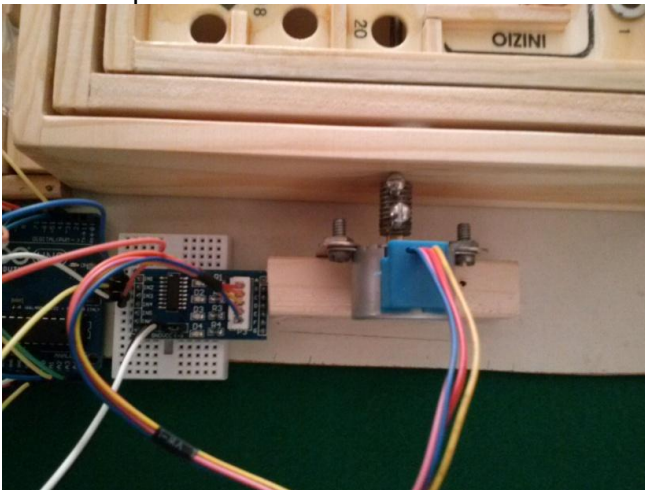
Non tutte le ciambelle riescono con il buco e questa e' la relazione di un fallimento, poiche' il risultato finale non e' stato raggiunto. Ma il prototipo funziona correttamente ed oltretutto e' piuttosto interessante, per cui e' stata comunque pubblicata questa scheda.

L'idea e' quella di utilizzare Arduino per pilotare una pallina di ferro in un labirinto di legno. Uno di quei giochi vintage che ormai e' anche difficile trovare.

Prima della fase di "pilotaggio" bisogna individuare le mosse da compiere per portare la pallina alla meta e, per ogni mossa, la corretta inclinazione dei due piani (alto / basso e destra / sinistra). E' stato quindi predisposto l'impianto rappresentato in foto, la cui costruzione e' stata resa laboriosa dalla preparazione del piano e dei supporti per i

componenti e dalla produzione di due manicotti in metallo (con relativi grani di fissaggio), in grado di collegare gli alberi di due motori (originariamente si era pensato di utilizzare dei servomotori) con gli alberi delle manopole del labirinto.

Per vincere l'inerzia della pallina ed evitare quindi che si muova solo quando i piani sono troppo inclinati e di fatto prenda poi una velocita' tale da renderla ingovernabile, si e' pensato di aggiungere un sistema di "shock meccanico", e cioe' una piccola massa, che battendo sulla cassa del labirinto, provoca un microterremoto e, almeno nelle intenzioni, vince l'inerzia iniziale mettendo in moto la pallina.



L'esperimento non e' riuscito poiche', nonostante l'estrema attenzione posta nell'inclinare i piani (un grado per volta, spesso seguiti dalla massa di shock), la pallina, una volta in moto, prende velocita' e, per via della sua massa, risulta pressoché impossibile arrestarla prima delle buche disseminate lungo il percorso.

Operativamente il sistema consente, in fase iniziale di manovrare i motori per portare, a poco a poco in bolla i piani del labirinto dopodiche' si ferma in attesa che venga premuto il pulsante di shock. Una volta premuto il pulsante, il sistema azzerà i

contatori di passi ed inizia ad eseguire e ad evidenziare sul display i movimenti dei motori richiesti tramite i quattro pulsanti che comandano:

- la scelta della manopola (alto/basso oppure sinistra/destra)
- aumento dei gradi di inclinazione
- diminuzione dei gradi di inclinazione
- effetto shock

Il pulsante di scelta della manopola funziona in flip/flop: ogni volta che viene premuto rende operativa la manopola al momento non attiva. I led segnalano la manopola attiva: il led verde segnala "alto/ basso" mentre quello giallo segnala "destra/sinistra"

Nota: Questo prototipo 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 \(PDF scaricato nell'area di download\)](#)

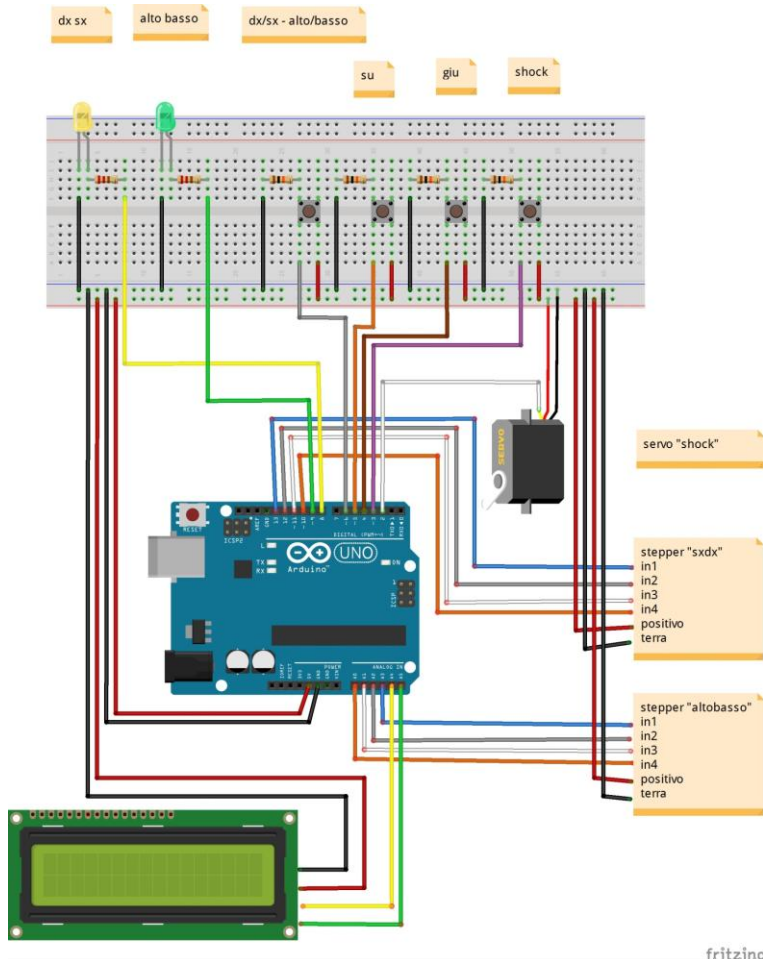
Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Materiali

Oltre ai manicotti, ai piani ed ai supporti per il fissaggio dei componenti sono necessari:

- due motori passo passo con relativi driver
- un servo sg90
- quattro pulsanti
- quattro resistenze da 10 k ohm
- un led giallo
- un led verde
- due resistenze da 220 ohm

Schema



Arduino: labirinto – cronaca di un fallimento

Programma

/ questo programma pilota le manopole di un labirinto di legno. Si tratta di un programma di tuning, utilizzato per costruire, passo per passo, la tabella che sara' poi utilizzata dal programma che pilotera' la pallina.*

Il circuito e' formato da due motori passo passo che manovrano le manopole, un servo che manovra la massa di shock (una massa che, su comando, batte sulla cassa del labirinto per provocare un leggero tilt per "innescare" il movimento della pallina), un display lcd, quattro pulsanti e due led.

Operativamente il sistema consente, in fase iniziale di manovrare i motori passo passo per portare, a poco a poco in bolla i piani del labirinto, dopodiche' si ferma in attesa che venga premuto il pulsante di shock.

Una volta premuto il pulsante di shock, il sistema azzerava i contatori di passi ed inizia ad eseguire e ad evidenziare sul display i movimenti dei motori richiesti tramite i quattro pulsanti che comandano:

- la scelta della manopola (alto/basso oppure sinistra/destra)
- aumento dei gradi di inclinazione
- diminuzione dei gradi di inclinazione
- effetto shock

Il pulsante di scelta della manopola funziona in flip/flop, ogni volta che viene premuto rende operativa la manopola al momento non attiva. I led segnalano la manopola attiva. Il led verde segnala "alto/ basso" mentre il led giallo "destra/sinistra"

*E' in questo modo possibile individuare per ogni tratto del labirinto, la giusta inclinazione dei piani e registrarla su di un foglio per poi riportarla nella tabella del programma che pilotera' la pallina dall'inizio alla fine */*

```
#include <Wire.h> // libreria wire presente, di fault, nell'IDE
#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd a due porte
// nota: i terminali SDA e SCL del display devono essere collegati, rispettivamente, alle porte
// analogiche 4 e 5
//-----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
#include <Servo.h> // richiama la libreria di gestione dei servomotori
Servo shock ; // crea il servo oggetto "shock" da utilizzare per manovrare la massa di shock.
// nota: in un programma possono essere create, al massimo otto servo oggetti
#include <Stepper.h> // richiama la libreria di gestione dei motori passo passo
Stepper sxdx(32, 13, 11, 12, 10); // numero di impulsi per un giro dell'albero motore (32)
// ed elenco delle porte di sxdx, in sequenza di attivazione
/* nota: Le porte 13, 12, 11 e 10 di Arduino sono collegate, rispettivamente,
ai pin In1, In2, In3 ed In4 del driver sxdx, ma la sequenza di attivazione deve essere
In1, In3, In2 ed In4 per cui la dichiarazione delle porte e' 13, 11, 12, e 10
*/
Stepper altobasso (32, 14, 16, 15, 17); // numero di impulsi per un giro dell'albero motore (32)
// ed elenco delle porte altobasso, in sequenza di attivazione
/* nota: le porte 14, 15, 16 e 17 di Arduino (le porte analogiche da A0 ad A3) sono collegate,
rispettivamente, ai pin In1, In2, In3 ed In4 del driver altobasso, ma la sequenza di attivazione
deve essere In1, In3, In2 ed In4 per cui la dichiarazione delle porte e' 14, 16, 15, e 17
*/
//***** definizione delle costanti *****
//
int batti = 3; // il pulsante di selezione dello shock e' collegato alla porta 3
int giu = 4; // il pulsante di diminuzione dei gradi e' collegato alla porta 4
int su = 5; // il pulsante di aumento dei gradi e' collegato alla porta 5
int manopola = 6; // il pulsante di selezione della manopola (dx sx oppure alto basso)
// e' collegato alla porta 6
int ledverde = 9; // il led verde e' collegato alla porta 9
int ledgiallo = 8; // il led giallo e' collegato alla porta 8

//*****definizione delle variabili *****
int statobatti = 0; // variabile in cui viene inserito lo stato del pulsante batti
int statogiu = 0; // variabile in cui viene inserito lo stato del pulsante giu
int statosu = 0; // variabile in cui viene inserito lo stato del pulsante su
int statomanopola = 0; // variabile in cui viene memorizzata la manopola su cui si sta
// operando: 1 = sxdx, 2 = altobasso
int acquisiscimanopola = 0; // variabile in cui viene inserito lo stato del pulsante manopola
int impulsisxdx = 0; // variabile in cui viene inserito il numero di impulsi per il motore
sxdx
int impulsialtobasso = 0; // variabile in cui viene inserito il numero di impulsi per il
// motore altobasso
int totimpulsisxdx = 0; // variabile in cui viene inserito il numero totale di impulsi per il
// motore sxdx
```

Arduino: labirinto – cronaca di un fallimento

```
int totimpulsialtobasso = 0; // variabile in cui viene inserito il numero totale di impulsi per il
// motore altobasso
int gradishock = 90; // variabile in cui viene inserita l'angolazione di shock
int switchprimavolta = 0; // semaforo per separare la fase di equilibratura dalla fase di
// gestione delle angolazioni
/**
***** routine di pulizia del display *****
***/
void puliscischermo (void)
{
  lcd.clear ();
}
/**
***** routine di azionamento dei motori *****
***/
void posizionaservo (void)
{
  altobasso.step(impulsialtobasso*4);
  delay (50);
  impulsialtobasso = 0;
  sdx.step (impulsisdx*4);
  delay (50);
  impulsisdx = 0;
  if (gradishock != 90)
  {
    shock.write (0);
    delay (500);
    shock.write (160);
    delay (500);
    shock.write (0);
    gradishock = 90;
  }
  delay (100);
}
/**
***** routine di gestione delle manopole *****
***/
void gestionemanopole (void)
{
  // **** individua la manopola sulla quale agire (usa un pulsante in flip/flop)****
  acquisiscimanopola = digitalRead (manopola); // aquisisce il segnale di settaggio della manopola
  if (acquisiscimanopola == 1)
  {
    if (statomanopola == 1) // se il precedente statomanopola era 1 (sdx)
    {
      statomanopola = 2; // posiziona manopola su altobasso
      digitalWrite (ledgiallo, HIGH); // accende il led giallo (opera su manopola altobasso)
      digitalWrite (ledverde, LOW);
      delay (500);
    }
    else // se invece il precedente statomanopola era 2 (altobasso)
    {
      statomanopola = 1; // posiziona manopola su dxsx
      digitalWrite (ledverde, HIGH); // accende il led verde (opera su manopola dxsx)
      digitalWrite (ledgiallo, LOW);
      delay (500);
    }
  }
  // ***** diminuzione dei gradi di inclinazione *****
  statogiu = digitalRead (giu); // acquisisce lo stato del pulsante giu
  if (statogiu == 1) // se e' stato premuto il pulsante giu
  {
    if (statomanopola == 1)
    {
      impulsisdx = -1;
      totimpulsisdx--;
    }
    if (statomanopola == 2)
    {
      impulsialtobasso = -1;
      totimpulsialtobasso--;
    }
  }
  // ***** aumento dei gradi di inclinazione *****
  statusu = digitalRead (su); // acquisisce lo stato del pulsante su
  if (statusu == 1) // se e' stato premuto il pulsante su
  {
    if (statomanopola == 1)
```

Arduino: labirinto – cronaca di un fallimento

```
{
  impulsisxdx = 1;
  totimpulsisxdx++;
}
if (statomanopola == 2)
{
  impulsialtobasso = 1;
  totimpulsialtobasso++;
}
}
// ***** posizionamento del sistema di shock *****
statobatti = digitalRead (batti); // acquisisce lo stato del pulsante batti
if (statobatti == 1) // se e' stato premuto il pulsante batti
  gradishock = 160; // predispose il movimento del meccanismo di shock
}

/**
//***** routine di visualizzazione impulsi *****
/**
void visualizzaimpulsi (void)
{
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga
  lcd.print ("sdx = ");
  lcd.print (totimpulsisxdx);
  lcd.print (" ");
  lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
  lcd.print ("altobasso = ");
  lcd.print (totimpulsialtobasso);
  lcd.print (" ");
}

void setup()
{
  lcd.begin(16,2); // inizializza il display (16 caratteri per due righe)
  sdx.setSpeed(100); // imposta una media velocita' di rotazione dell'albero
  altobasso.setSpeed(100); // dei due motori passo passo: 100 impulsi al secondo, pari a
  // 3,125 giri al secondo dell'albero motore e ad una rotazione di 17,5 gradi al secondo
  // del perno in uscita
  shock.attach(2); // assegna il servo oggetto "shock" alla porta 2
  pinMode (batti, INPUT); // definisce il pulsante batti come unita' di input
  pinMode (giu, INPUT); // definisce il pulsante giu come unita' di input
  pinMode (su, INPUT); // definisce il pulsante su come unita' di input
  pinMode (manopola, INPUT); // definisce il pulsante su come unita' di input
  pinMode (ledverde, OUTPUT); // definisce il led verde come unita' di output
  pinMode (ledgiallo, OUTPUT); // definisce il led giallo come unita' di output
  gradishock = 90; // pone il servo shock in posizione intermedia
  puliscischermo (); // lancia la routine di pulizia dello schermo
  lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga
  lcd.print ("metti in piano");
  lcd.setCursor(0,1); // posiziona il cursore all'inizio della seconda riga
  lcd.print ("e premi shock ");
  while (switchprimavolta == 0) // loop di messa in piano del labirinto
  {
    gestionemanopole(); //lancia la routine di gestione delle manopole
    if ((impulsisxdx != 0) || (impulsialtobasso != 0)) // verifica se c'e qualcosa da fare
      posizionaservo (); // se e' cambiata qualche impostazione aziona i motori
    if (statobatti == HIGH) // se si desidera uscire dal ciclo
    {
      switchprimavolta = 1; // setta lo switchprimavolta in modo da non ripetere piu' il ciclo
      statobatti = 0; // rimette a zero il valore del pulsante batti
      gradishock = 90; // ripristina il valore standard del braccio del servomotore
      totimpulsialtobasso = 0; // azzerà il registroaltobasso
      totimpulsisxdx = 0; // azzerà il registro sdx
      delay (500);
    }
  }
  puliscischermo ();
}

void loop()
{
  gestionemanopole(); //lancia la routine di gestione delle manopole
  if ((impulsisxdx != 0) || (impulsialtobasso != 0) || (gradishock != 90)) // verifica se e'
  // cambiato qualcosa
  posizionaservo (); // se e' cambiata qualche impostazione aziona i motori
  visualizzaimpulsi ();
}
```