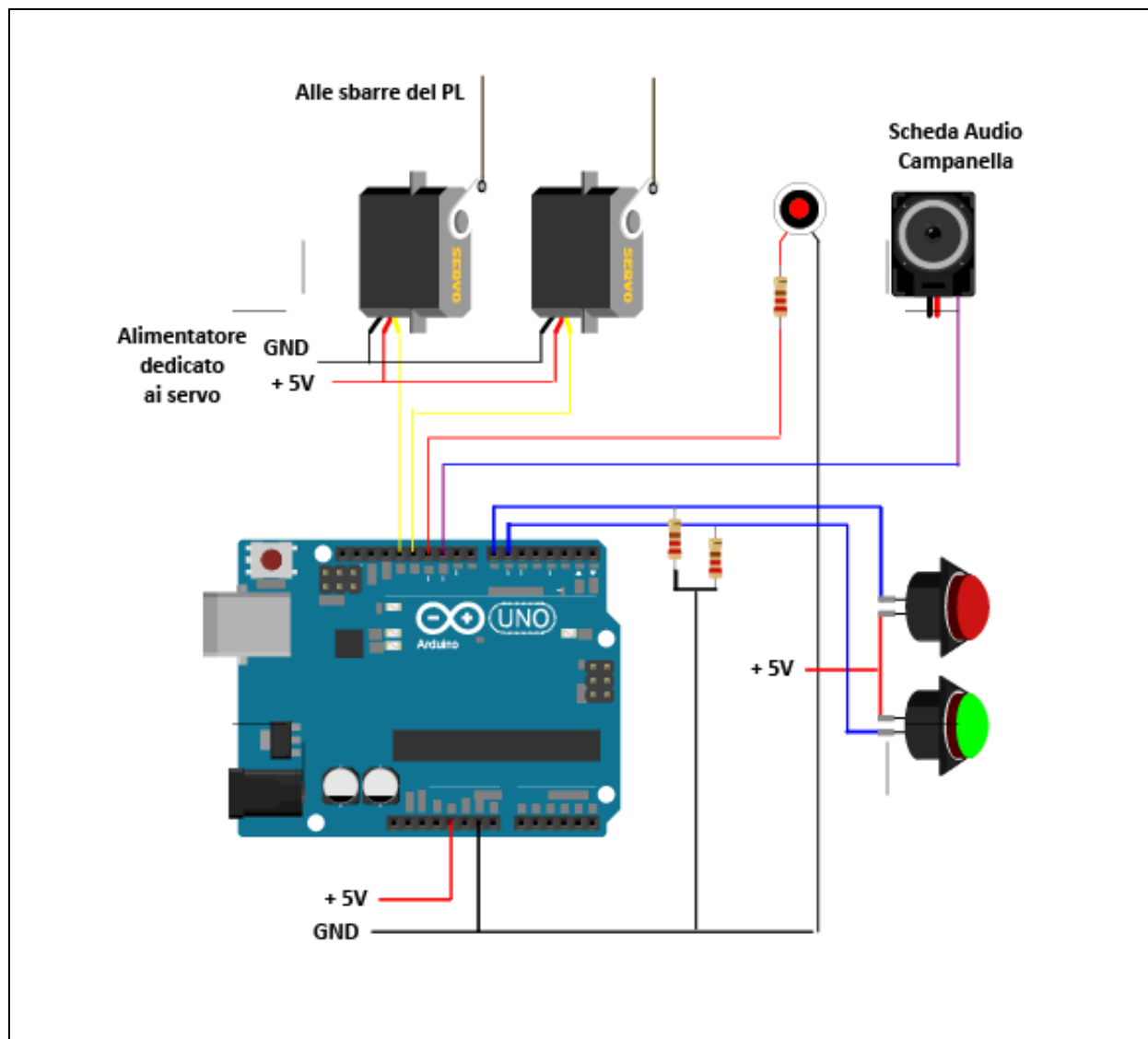


## Il Passaggio a Livello



Le sbarre del passaggio a livello della Stazione di Fossacesia sono azionate da due servomotori per aeromodellismo comandati da una scheda Arduino su cui gira un programma (*sketch*) che utilizza una particolare libreria che permette di avere una movimentazione delle sbarre fluida, senza scatti.

Premendo il tasto di chiusura si accende il segnale rosso, parte la campanella e poi incominciano a scendere le sbarre sfalsate come nella realtà; nella sequenza inversa invece prima incominciano ad alzarsi le sbarre e poi si spegne il segnale. Avendo deciso in seguito di far comandare il PL da Rocrail abbiamo semplicemente collegato in parallelo ai pulsanti i contatti di due relè connessi a due porte libere di un decoder Lenz LX100: si è trattato solo di pigrizia perché, il nostro Arduino potevamo anche fare da decoder come descritto nella sezione sul decoder DCC universale.

Per il suono della campanella abbiamo utilizzato una economica schedina con altoparlante utilizzata nelle bambole parlanti su cui abbiamo registrato il suono, ma qualsiasi altra soluzione funzionerebbe altrettanto bene.

Il SW da noi impiegato come base, lo *sketch*, è stato trovato in rete e testato sul plastico dove abbiamo constatato che le sbarre si muovevano con piccoli, antiestetici scatti. Questo avviene impiegando la libreria Servo() fornita con il SW arduino.

<http://www.mauroalfieri.it/elettronica/passaggio-a-livello-per-plastici.html>

Sostituendo la libreria con una che utilizza *microstep* abbiamo ottenuto una corsa piu' fluida e realistica

<https://github.com/netlabtoolkit/VarSpeedServo>

### Ecco lo sketch modificato

```
/**
 * Passaggio a livello per plastici con Arduino
 *
 * Autore: Mauro Alfieri
 * Web: www.mauroalfieri.it
 * Tw: @mauroalfieri
 *
 * utilizzati due servo per due sbarre
 */

#include <VarSpeedServo.h>

#define swClose 2
#define swOpen 3
// pin 2 chiude sbarre
// pin 3 apre sbarre

#define ledSemaf 10
//Pin che accende il semaforo
#define campanella 8
//pin che aziona la campanella

#define posMin 900
#define posMax 1500
#define timePL 10

// Questi sono i valori sperimentali utilizzati per posizionare le sbarre:1500 corrisponde a
// sbarre abbassate, 900 a sbarre alzate. LOW corrisponde a semaforo ROSSO

boolean statoPL=false;
```

```

int currentGrad= 750;

VarSpeedServo myServo;
VarSpeedServo myServo1;

const int servoPin = 12;

const int servoPin1 = 5;

void setup()
{
  pinMode( swClose,INPUT );
  pinMode( swOpen,INPUT );
  pinMode( ledSemaf,OUTPUT );
  pinMode( campanella,OUTPUT );

  //myservo.attach(servoPin);
  myServo.attach (servoPin, posMin, posMax);
  myServo1.attach (servoPin1, posMin, posMax);

  myServo.slowmove (1500, 4); // porta le sbarre in posizione intermedia all'accensione
  myServo1.slowmove (1500, 4); // " " " "

  digitalWrite( ledSemaf,LOW ); // accende il semaforo
  digitalWrite( campanella,LOW );
}

void loop()
{
  if ( digitalRead(swClose) == HIGH ) { statoPL = true; }
  if ( digitalRead(swOpen) == HIGH ) { statoPL = false; digitalWrite( campanella,HIGH );
  delay(1000); digitalWrite( campanella,LOW ); }

  if ( statoPL == true ) {

    myServo.slowmove (1900 , 3); // sbarre alzate
    myServo1.slowmove (2100 , 3);

    digitalWrite( ledSemaf,LOW );
    //digitalWrite( campanella,LOW );

  }
}

```

```
if ( statoPL == false ){  
  
    //digitalWrite( campanella,LOW );  
  
    myServo.slowmove (900, 3); // sbarre abbassate  
    myServo1.slowmove (900, 3);  
    digitalWrite( ledSemaf,HIGH);  
    //digitalWrite( campanella,HIGH );  
    }  
}
```