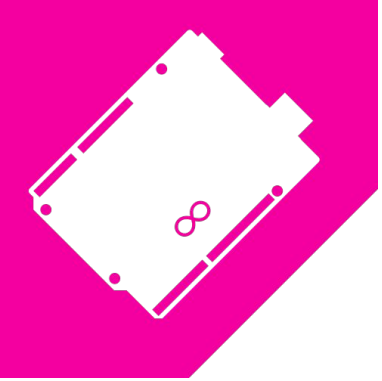
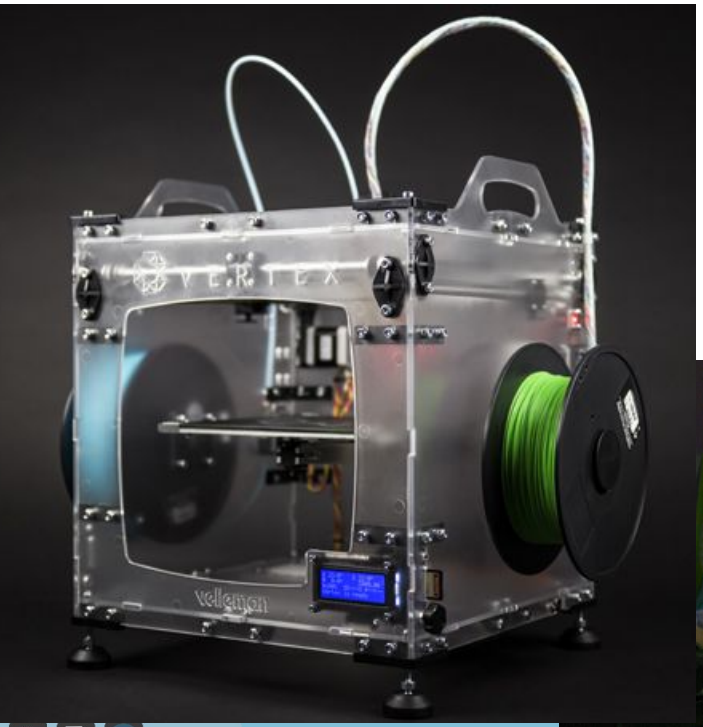


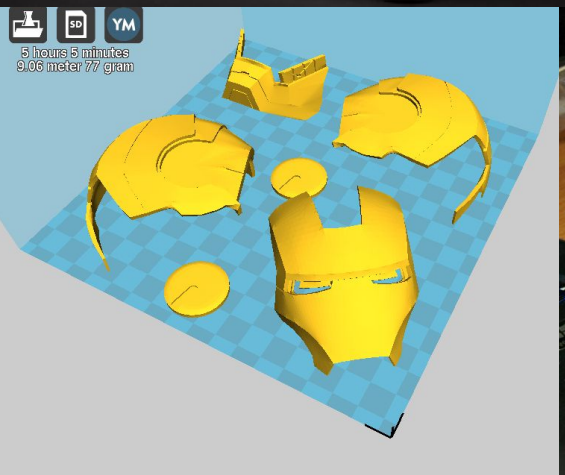
Lezione 5

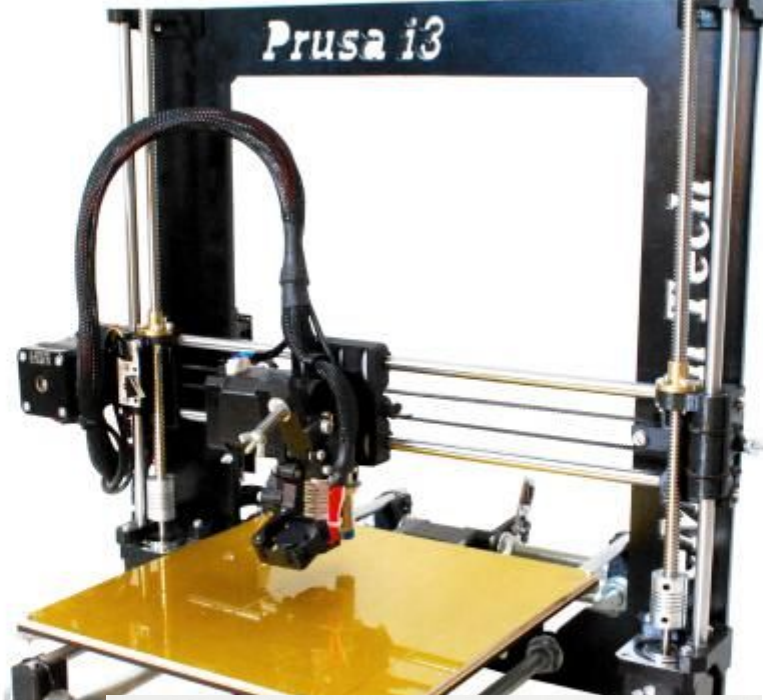
Un corso gentilmente offerto con il sudore e le lacrime di MugRomaTre e Roma Tre

e Magliana

Stampa3D


VERTEX
vellemen

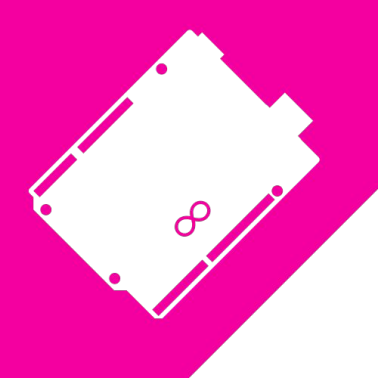

5 hours 5 minutes
9.06 meter 77 gram


Prusa i3


MUG
R3

Cosa serve la stampa 3d

- Ricreare Oggetti
- Realizzare Oggetti non possibili tecniche classiche di produzione
  - vedi motori superdraco
- Prototipare rapidamente
- Stupire gli amici (e non)



Stampanti 3d

- non esiste “La” stampante 3d
- Tecnologie differenti, capacità differenti, pubblico differente
- 100€ ~ 100000€
  - il prezzo non è sbagliato
- chiaramente non tutti hanno bisogno del macchinario di fascia alta
  - grazie ai brevetti scaduti abbiamo anche macchine a basso prezzo

Tecnologie di Stampa

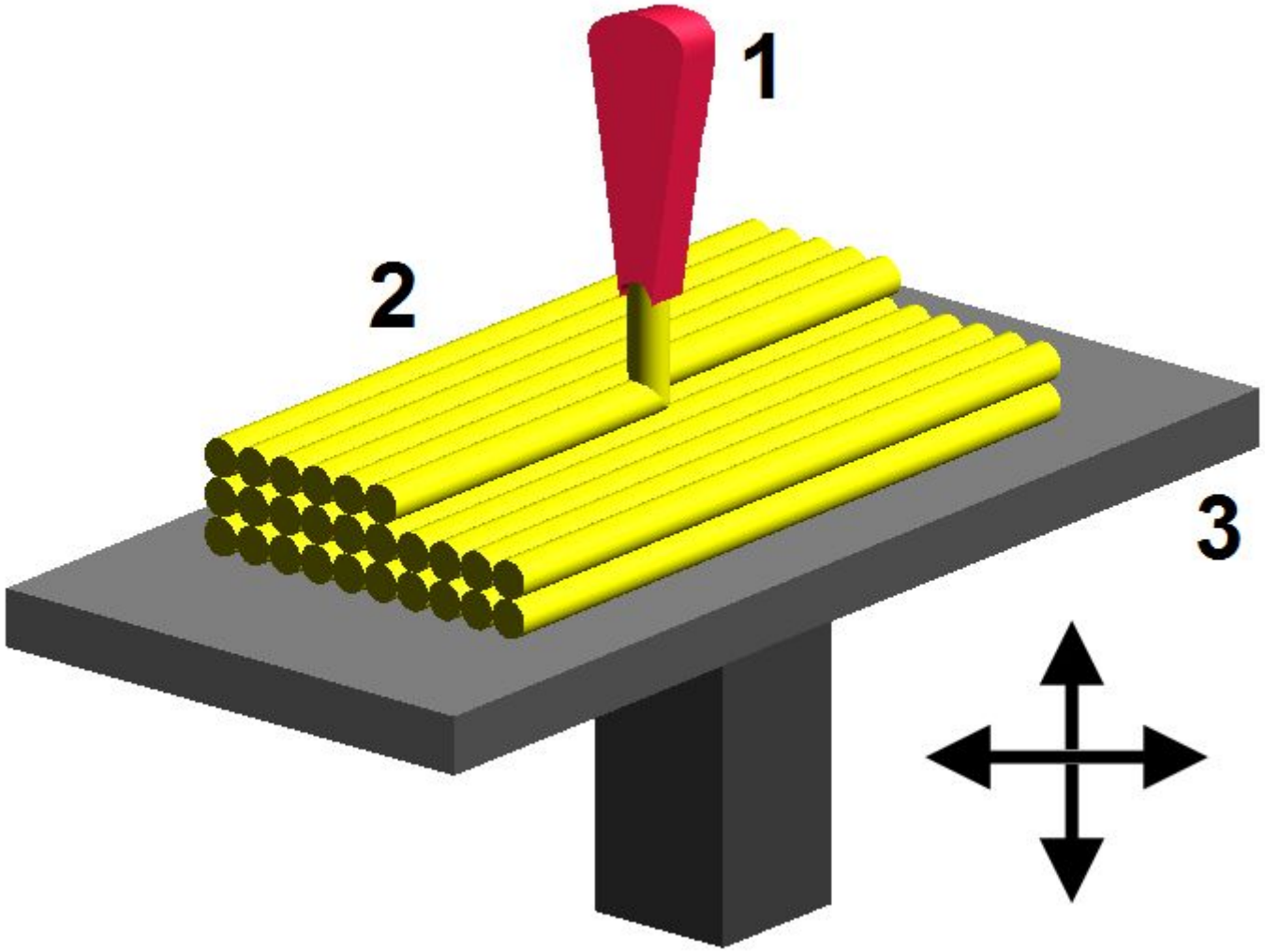
La tecnica fa la macchina

- filamento fuso (la più vecchia?)
- resina indurita (DLP-SLA)
- synthering (SLS)
  - metallo
  - plastiche
- ...

La stampa a filamento fuso

- un filamento di plastica è fuso e depositato un livello sopra l'altro
- quando sono scaduti i brevetti, sono nate delle versioni opensource sviluppate dal progetto RepRap di stampanti
  - la nostra è la Prusai3
- vari meccanismi per muovere la testina di stampa, doppia testina
- bassi costi (quanto? 300$ sembra essere il minimo per un kit, ma esistono progetti fatti di materiali di recupero)
  - vedi: junkyard 3d printer, dvd drive 3d printer
- il problema dell'overhang e del supporto
  - ogni layer deve poggiare su qualcosa, per oggetti più complicati bisogna stampare anche del materiale di supporto, che porta via tempo e qualità finale

La stampa a filamento fuso



La testina (1) deposita materiale fuso, che si solidifica e si lega al materiale (2) già depositato in precedenza sul (3) piano di stampa. O la testina, o Il piatto, o tutti e due si muovono per poter raggiungere 3 gradi di libertà

Resina Indurita
Digital Light Processing – StereoLitography (credo)

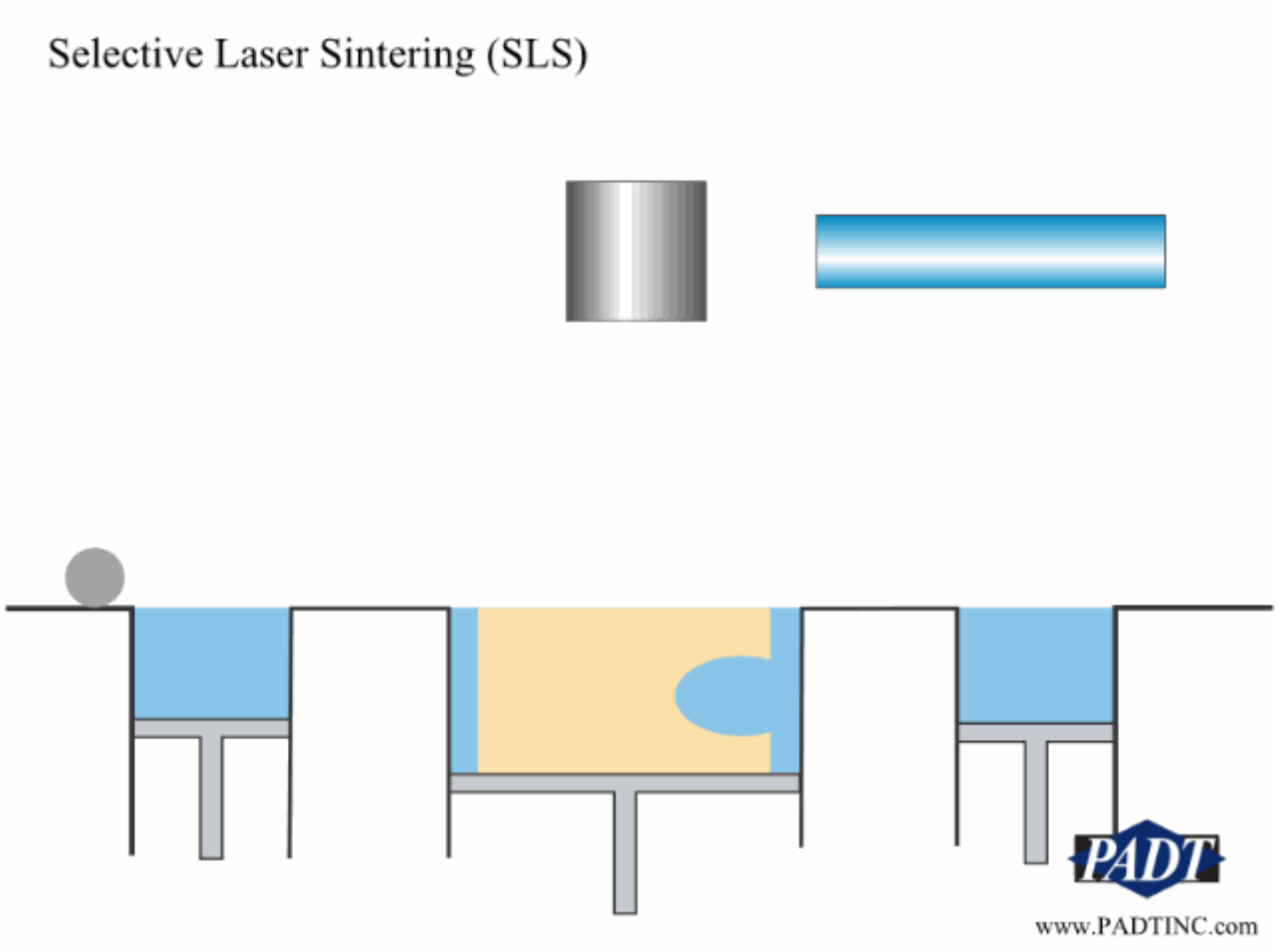
- della resina è indurita un livello dopo l'altro
  - si usa un proiettore o un laser per “disegnare” un livello dell'oggetto, la resina di indurisce quando viene esposta a questa immagine

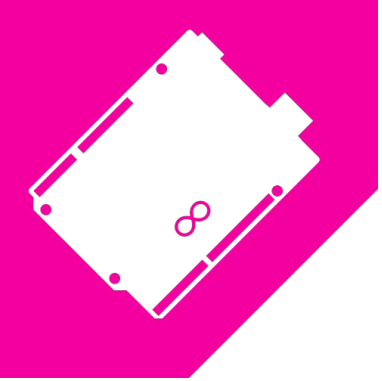
- stampe più accurate, più costose
  - più alta risoluzione e precisione usando un laser
- il problema inverso del supporto
  - la resina deve avere un qualcosa a cui aggrapparsi

Selective Laser Synthering

- Stampe a livello professionale
  - i film recenti di stopmotion hanno usato questo processo per realizzare i volti e le espressioni dei personaggi
- su uno strato di polvere di materiale, un laser crea un livello, e poi viene steso un nuovo strato
- precisione molto più alta
  - sempre grazie al laser
- forme più ardite: non ho bisogno di sostegni
  - il materiale non aggregato fornisce sostegno ai livelli creati
  - ma ho bisogno di vie di fuga per la polvere non utilizzata
- una variante è usare un agente aggregante al posto del laser
  - questo permette anche la stampa diretta di oggetti colorati, basta usare delle colle con inchiostro

Selective Laser Synthering

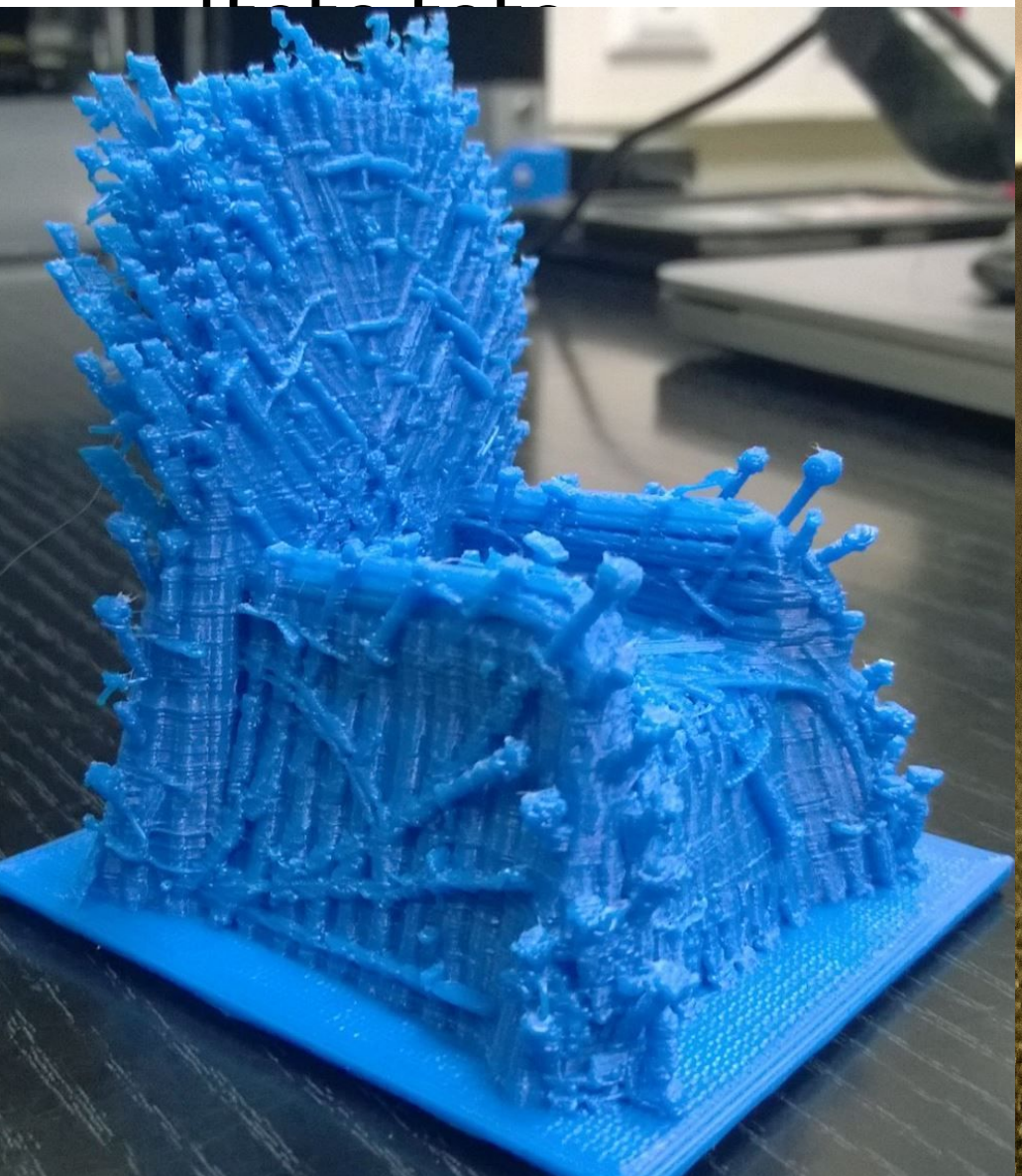


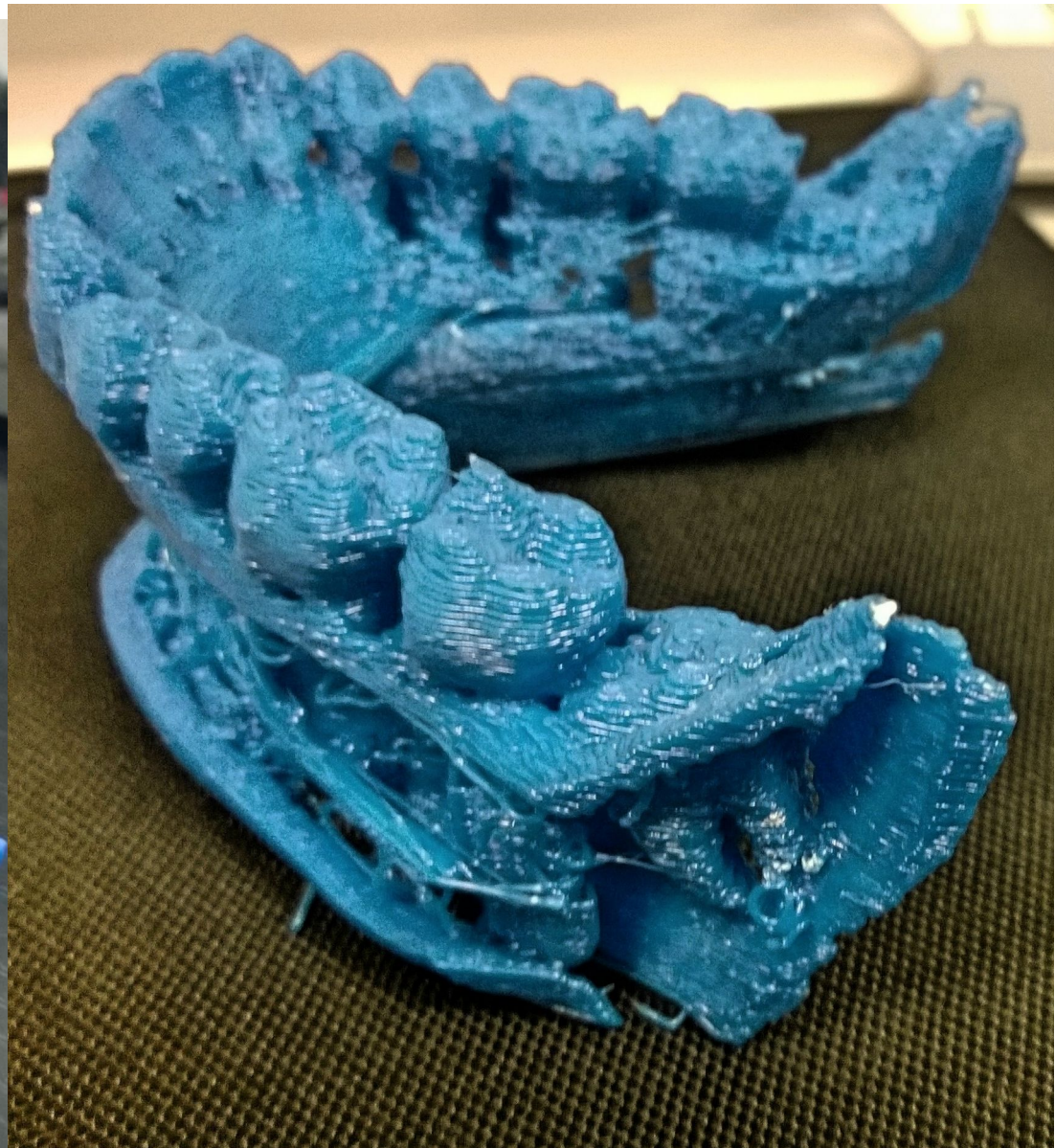


Alcune Nostre Stampanti

- Noi in quanto hobbysti poveri possediamo una stampante a filamento
  - prusa i3, kit, costato poco
    - tanto sudore, tante lacrime, problemi frequenti
- l'università ci ha prestato una nuova stampante 3d, sempre a filamento
  - vertex3d, kit, costato un poco di più
    - tanto sudore, tante lacrime, ma dovrebbe essere più reliable
- Per il resto della lezione farò riferimento solo a loro

Le nostre stampe

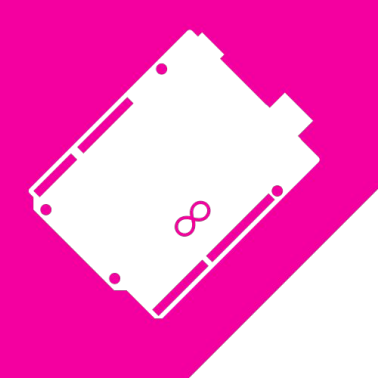




Materiali Per la stampa

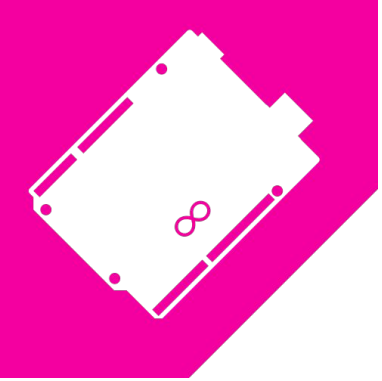
Ogni produttore ha le sue varianti, la qualità della miscela e la consistenza delle dimensioni del filo sono importanti

- Pla
  - plastica biodegradabile
    - presenta problemi di stoccaggio
  - facile da stampare
- Abs
  - meccanicamente più resistente, plastica dei lego
    - i fumi sono irritanti X_X
- Pet
  - la stessa delle bottiglie dell'acqua
  - interessanti proprietà ottiche e meccaniche
- NinjaFlex
  - filamento elastico e gommoso
  - costoso, per utilizzi specializzati come giunti e ruote



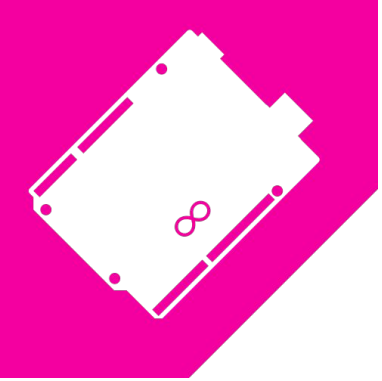
Ma cosa si manda in stampa?

- in input la stampante prende un file Gcode
- il file Gcode è un protocollo di comandi che dicono alla stampante dove muovere la testina e quanto materiale estrudere
- è a basso livello: il risultato va bene solo per una stampante. Su un'altra potrebbe produrre spazzatura



Gcode - 1

- Protocollo a basso livello per il controllo dei macchinari CNC
  - Computer Numerical Control
  - stampanti 3d, frese computerizzate, macchine da taglio
- https://en.wikipedia.org/wiki/G-code
- http://reprap.org/wiki/G-code
- G0 X10 Y2 Z200 ; vai a punto 10,2,200 muovendoti in linea
- G0 X23 Y32 Z200 E22 F1500 ; vai estrudendo 22 mm di materiale e con velocità 1500 mm/min
- G28 ; torna all'origine degli assi

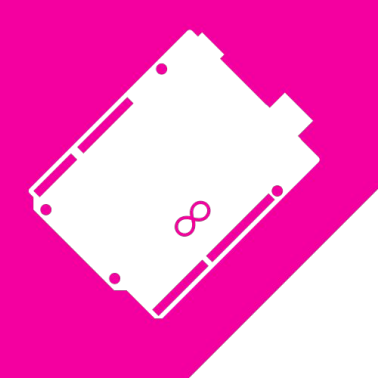


Gcode - 2

- una stampante esegue quello che gli dice il gcode
  - a parte errori meccanici
- non tutte le sequenze di gcode producono un oggetto che abbia un senso
  - alcune sequenze però producono della musica, facendo risuonare i motori https://www.youtube.com/watch?v=pKsvXfUvCkQ
- per produrre un oggetto, mandiamo alla stampante una sequenza di comandi che disegna un oggetto per fette

Slicing

- è l'azione di affettare un modello per produrre gcode
  - è necessario conoscere i parametri della stampa e del materiale
- Cura e Slic3r gratis
  - ma anche soluzioni a pagamento
- tanti parametri per una stampa
  - orientamento, riempimento, velocità, temperatura, riempimento, sostegno
- Demo



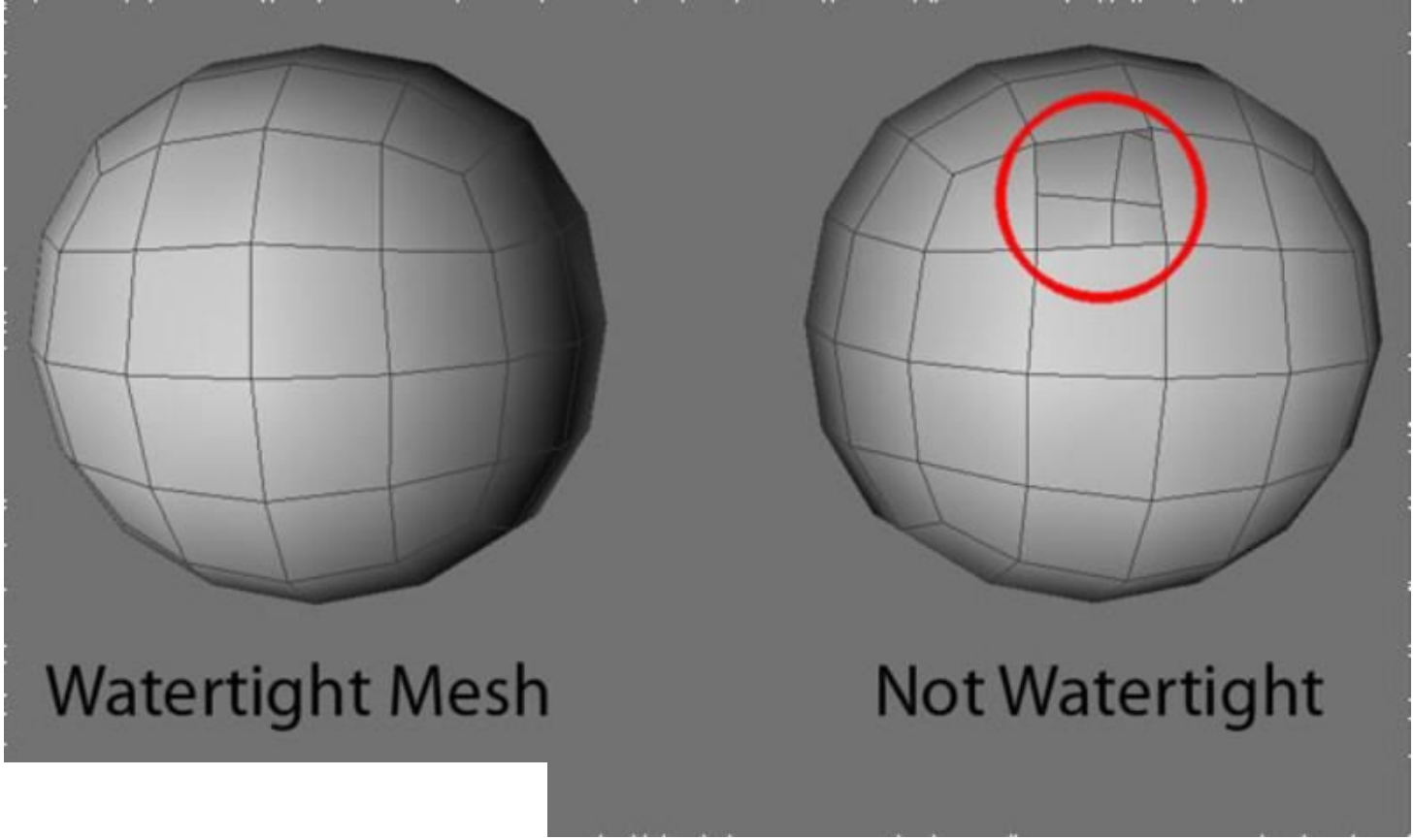
Stampabilità - 1

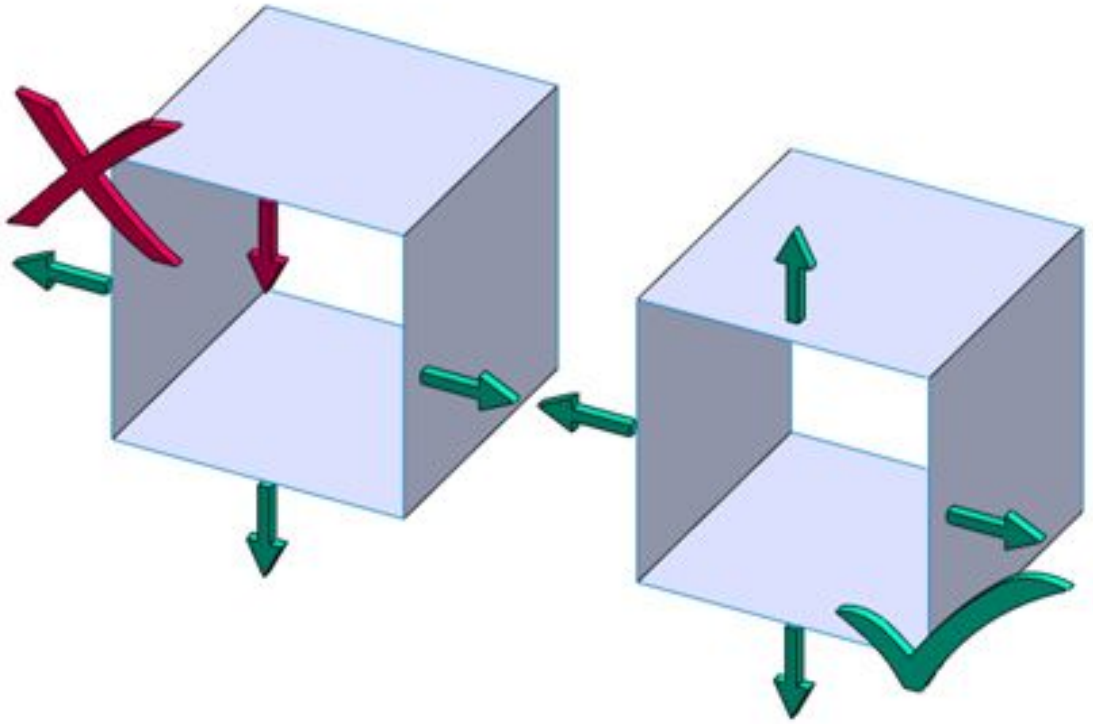
Per ottenere buoni risultati, due requisiti:

- Watertight
  - superficie chiusa
  - praticamente se il modello fosse riempito d'acqua, non dovrebbe perdere
  - è resposabilità del modellatore
  - molti modelli dei videogiochi non sono watertight

Stampabilità – 2

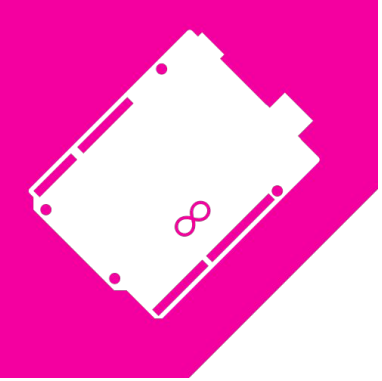
- Orientamento delle superfici
  - in grafica, ogni superficie ha una direzione che indica dove è il fuori e il dentro
  - i software di slicing usano questa informazione per capire dove è il dentro e il fuori
  - è responsabilità del software di modellazione calcolare l'orientamento
  - per favore non usate Sketchup, che tende a fare casini


Watertight Mesh
Not Watertight



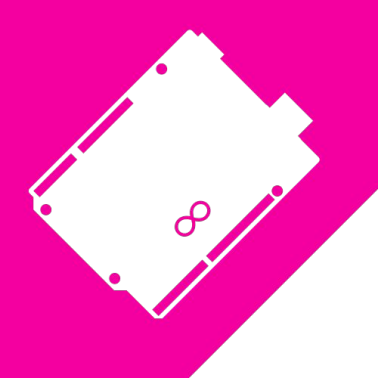
Modellazione

- lo slicer prende in input un modello 3d. Come produco questo modello?
  - Software di modellazione
    - Maya, Blender, Meshmixer, 123d design...
    - Openscad, Autocad, Librecad...
  - Scansione3D
    - fotogrammetria
    - scansioni laser
  - Molto spesso si lavora con più programmi per completare un modello



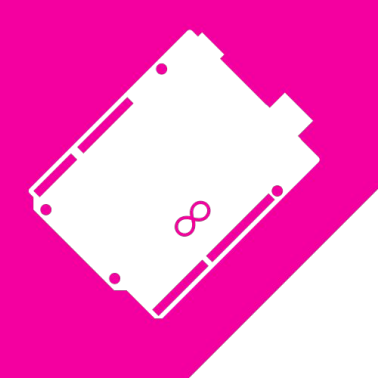
Openscad

- http://www.openscad.org/index.html
- Software di modellazione basato sulla geometria costruttiva
  - Unione, Differenza, Intersezione di forme
- Demo



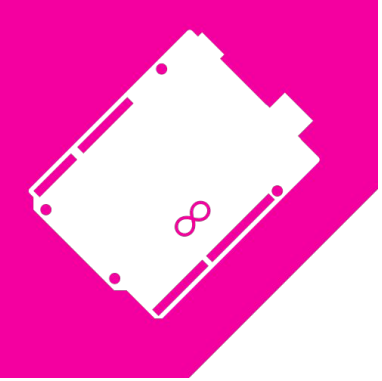
Openscad – il codice

```
intersection(){
    difference(){
        cube([10, 40, 40]);

        translate([0,10,0]) cube([10, 20, 24]);

        translate([0, 20, 24]) rotate(a=90, v=[0, 1, 0])
                                    cylinder(r=10, h=10);
    }

    translate([0, 40, 0]) rotate(a=90, v=[1,0,0])
                linear_extrude(height=40)
                        polygon(points=[[0, 0], [10, 0], [5, 40]]);
}
```



Costi

- Abbiamo creato il modello, lo abbiamo affettato con delle impostazioni appropriate, e lo abbiamo mandato in stampa. Quanto ci è costato?
- Sorprendentemente, il materiale è il minore dei costi
  - pochi grammi di materiare, su bobine da 1kg dal costo di 30€
- Il costo maggiore è l'operatore umano
  - e la stampante, se è costata molto



Tempi

- Anche per produrre qualcosa di semplice, I tempi che una persona spende a lavorare su un modello, a scegliere i parametri di stampa, e a seguire una stampa, sono alti
- Senza contare il tempo necessario per la manutenzione dei macchinari
  - e per chi è malato di questo hobby, anche i tempi dedicati al tuning e all'upgrade della tua macchina

Ricapitolando: Lavoro Necessario per una stampa

- costruzione della stampante e calibrazione
  - o solo accensione, se spendiamo di più per una stampante già assemblata e calibrata
- scelta del modello
  - o costruzione del modello
- slicing
  - non una scienza
- stampa

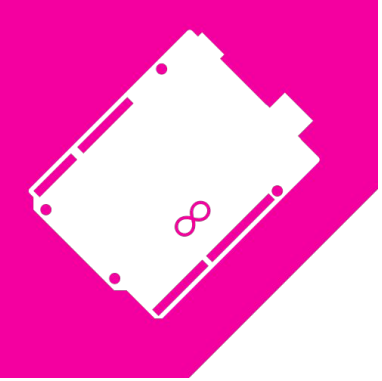
e non dimentichiamo la manutenzione

Stampa 3D: tanto lavoro, ma i risultati possono essere incredibili

Link Utili

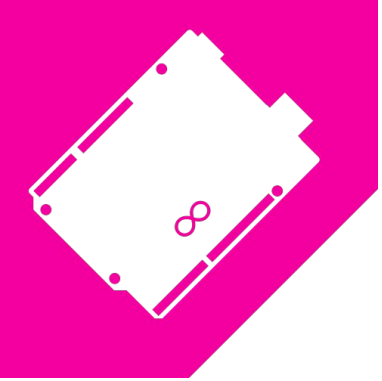
- https://en.wikibooks.org/wiki/OpenSCAD_User_Manual
- https://www.thingiverse.com/
- https://www.reddit.com/r/reprap
- https://www.reddit.com/r/3dprinting
  - hanno anche una lista di kit economici!
- http://reprap.org/
- http://www.123dapp.com/
  - software gratuiti di modellazione di autodesk

Grazie! una parola dal nostro Presidente



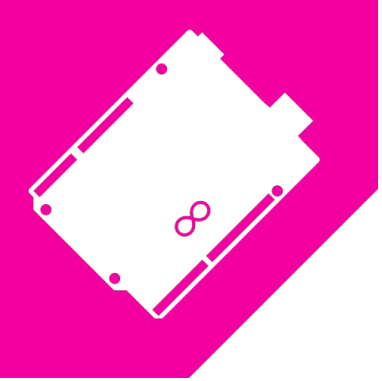
Cosa proviamo oggi? - 1

- Blink
- SerialAnalogRead
- Knob
- Wave
- PhotoServo
- PhotoServoClock



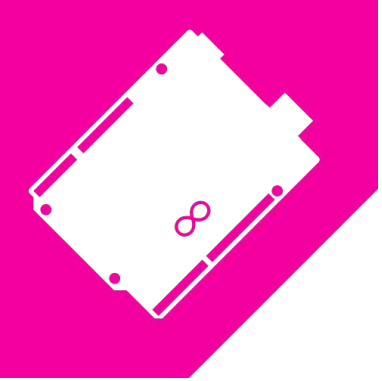
Blink

- Voglio accendere e spegnere il led sul pin 13, acceso per 700ms e spento per 350ms
- Hint: delay(ms) aspetta ms millisecondi
- Hint: non devo aggiungere nessun led, perché ce n'è uno già sulla scheda arduino



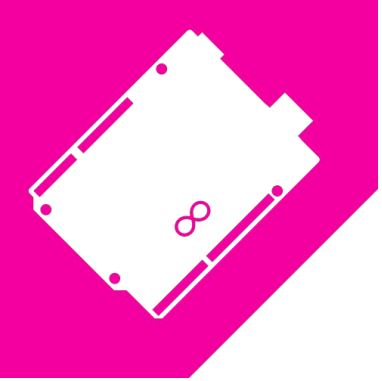
Blink Soluzione

```
/*
 * the setup function runs once when you press
 *  reset or power the board
 */

void setup() {
  pinMode(13, OUTPUT);
          // initialize digital pin 13 as an output.
}

          // the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);
          // turn the LED on (HIGH is the voltage level)
  delay(700);
          // wait for 700 ms
  digitalWrite(13, LOW);
          // turn the LED off by making the voltage LOW
  delay(350);
          // wait for 350 ms
}
```



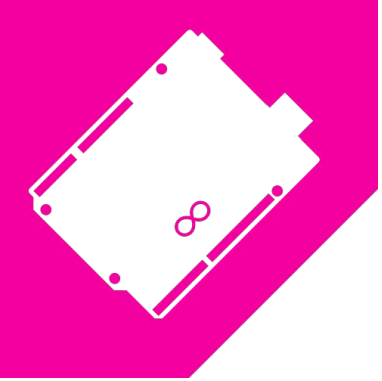
SerialAnalogRead

- Voglio stampare sul monitor seriale il voltaggio letto sul pin A0
- Ricordati di aprire il monitor seriale!
  - puoi anche aprire il plotter seriale per vedere
- Sul pin A0 potrei mettere il trimmer o il la fotoresistenza
  - come spiegato nelle slide della lezione precendente
- Hint: analogRead(pin) tutta la vita
- Hint: Serial.println(val)



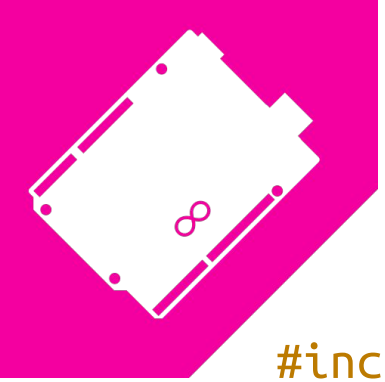
SerialAnalogRead Soluzione

```
void setup() {
  Serial.begin(9600);
            // initialize the serial communication:
            // remember to select this same velocity on the serial monitor
}

void loop() {
            // send the value of analog input 0:
  Serial.println(analogRead(A0));
            // wait a bit for the analog-to-digital converter
            // to stabilize after the last reading:
  delay(2);
}
```



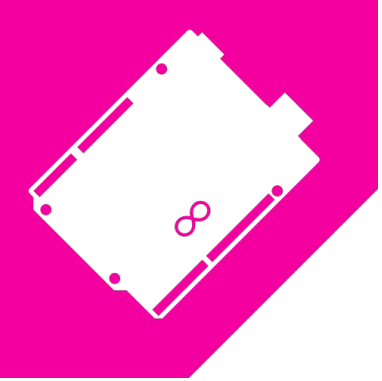
Knob

- Voglio controllare la posizione del servo con un potenziometro
- Servo: marrone → gnd, rosso→ 5V, arancione → pin 9
- Hint: int res= map(value, fromLow, fromHigh, toLow, toHigh) scala value da un range ad un altro.
  - y = map(x, 0, 1023, 0, 180); scala x da 1023 a 180



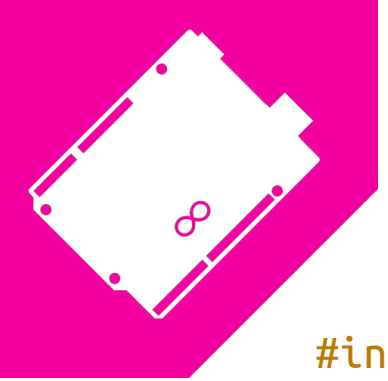
Knob Soluzione

```
#include <Servo.h>

Servo myservo;
    // create servo object to control a servo
    // hint: variables outside a function are accessible everywhere

void setup() {
    myservo.attach(9);
            // attaches the servo on pin 9 to the servo object
}

void loop() {
    int val = analogRead(A0);
            //reads the value of the potentiometer
            //(value between 0 and 1023)
    val = map(val, 0, 1023, 0, 180);
            //scale it to use with a servo (value between 0 and 180)
    myservo.write(val);
            //sets the servo position according to the scaled value
    delay(15);
            // waits for the servo to get there
}
```



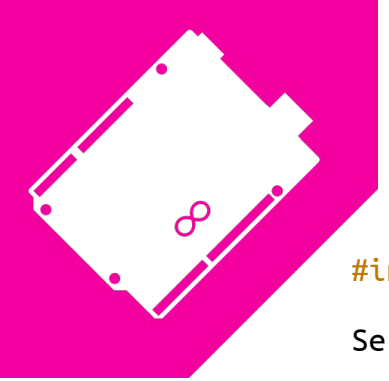
Wave

- Voglio essere salutato dal servo
- Il servo dovrebbe muoversi a destra e sinistra, e poi aspettare 10 secondi prima di salutare di nuovo
  - Hint: per chi non sa cosa è un for loop: http://www.arduino.cc/en/Reference/For
- Pro: posso utilizzare la fotoresistenza per farmi salutare solo quando sono davanti alla arduino?
  - Hint: si



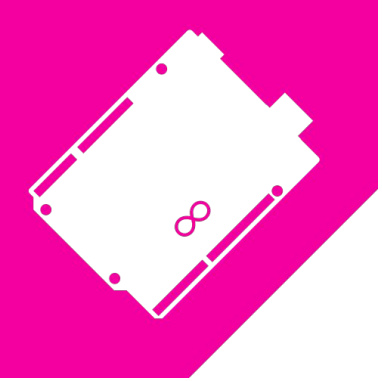
Wave Soluzione

```
#include <Servo.h>

Servo myservo;
    // create servo object to control a servo

void setup() {
    myservo.attach(9);
            // attaches the servo on pin 9 to the servo object
    myservo.write(90);
            //go to middle position
    delay(1000);
}

void loop() {
            //a for loop that repeats 3 times
    for(int i=0; i<3; i++){
            myservo.write(0);
            delay(1000);
            myservo.write(180);
            delay(1000);
    }
    myservo.write(90);
    delay(10000);
}
```



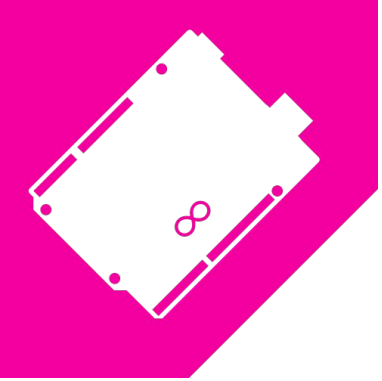
Wave Pro Soluzione

```
#include <Servo.h>

Servo myservo;
      // create servo object to control a servo

void setup() {
      myservo.attach(9);
                  // attaches the servo on pin 9 to the servo object
      myservo.write(90);
                  //go to middle position
      delay(1000);
}

void wave(){
            //using a separate function to keep everything clean
      for(int i=0; i<3; i++){
                  myservo.write(0);
                  delay(1000);
                  myservo.write(180);
                  delay(1000);
      }
      myservo.write(90);
      delay(1000);
}

int lastReading=0;
      //a good place to remember last value, across function calls
void loop() {
      int reading=analogRead(A0);
      if(abs(reading - lastReading) > 20){
            //we have a significant change in luminosity! somebody is here
            wave();
      }
      lastReading=analogRead(A0);
            //update value for next cycle
      delay(100);
}
```



PhotoServo

- Voglio controllare la posizione del servo con la fotoresistenza
- Praticamente il codice uguale all'esempio Knob
  - La fotoresistenza non ha una risposta lineare, ci si può sbizzarrire con map
- Hint: nelle slide della scorsa volta ho mostrato come collegare una fotoresistenza

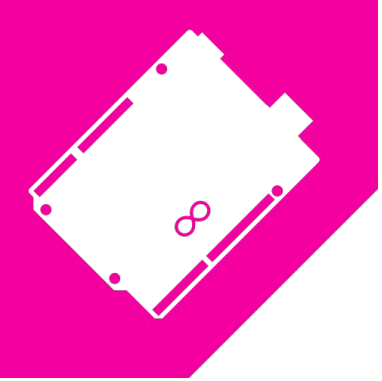


PhotoServoClock

- Come l'esempio di prima, ma voglio che ogni 3 secondi il braccetto vada a 90 gradi
- Hint: long val = millis() ritorna il numero di millisecondi passati dall'inizio dello sketch
  - Se chiamo millis() più volte, posso sottrarre i risultati per sapere quanti millisecondi sono passati fra due chiamate
  - Posso controllare che siamo passati 3000 millisecondi per fare una azione, e aggiornare un contatore

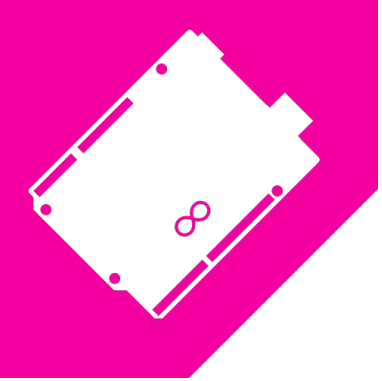
PhotoServoClock Soluzione

```
#include <Servo.h>

Servo myservo;
      //create servo object to control a servo

int potpin = 0;
      // analog pin used to connect the potentiometer
int val;
      // variable to read the value from the analog pin

long timePoint;
      //variable to save a time
void setup() {
      myservo.attach(9);
            // attaches the servo on pin 9 to the servo object
      timePoint = millis();
            //record current time
}

void loop() {
      if(millis() - timePoint > 3000){
            //have 3000 ms passed? if yes execute this action
            myservo.write(90);
            delay(500);
            timePoint = millis();
                  //update timePoint, ready for next tick
      }
      val = analogRead(potpin);          // reads the value of the potentiometer (value between 0 and 1023)
      val = map(val, 0, 1023, 0, 180);   // scale it to use it with the servo (value between 0 and 180)
      myservo.write(val);                // sets the servo position according to the scaled value
      delay(200);                         // waits for the servo to get there
}
```



Cosa proviamo oggi? - 2

- Tone
- LightTeremin
- Rgbled
- Buttons
- TeaTimer



Tone

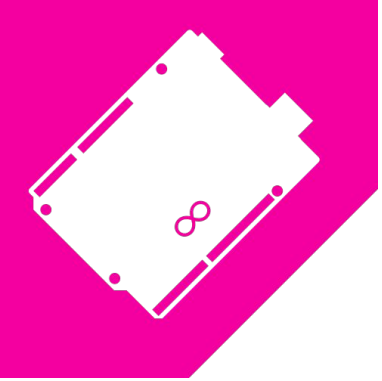
- Voglio usare la funzione tone() per far suonare un buzzer
  - http://arduino.cc/en/Reference/Tone
  - tone(pin, frequency, duration)
    - noTone(pin) per interrompere la nota
    - delay(duration*1.3) dopo tone, per separare più note
- Hint: Il La ha una frequenza di 440Hz
- il Buzzer lo spiego nella prox slide

Buzzer

- è un cosetto che fa rumore
- Quello nel nostro kit è un cristallo piezoelettrico che con una corrente alternata vibra
  - ha un senso: il – va collegato a GND, il + al pin arduino

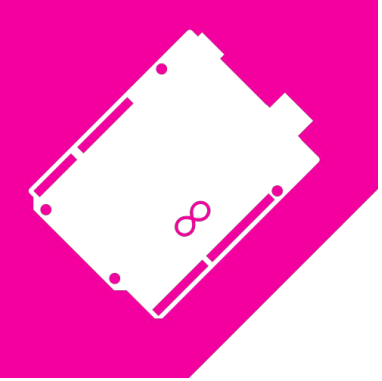
Tone Soluzione

```
void setup() {
    tone(8,220,500);
    delay(500*1.3);
    tone(8,220,500);
    delay(500*1.3);
    tone(8,220,500);
    delay(500*1.3);
    tone(8,174,375);
    delay(375*1.3);
    tone(8,262,125);
    delay(125*1.3);
    tone(8,220,500);
    delay(500*1.3);
    tone(8,174,250+125);
    delay(375*1.3);
    tone(8,262,125);
    delay(125*1.3);
    tone(8,220,1000);
    delay(1000);
    noTone(8);
}

void loop() {
  // empty
}
```



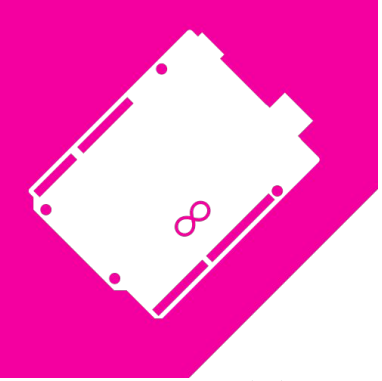
LightTeremin

- “The theremin (/ˈθɛrəmɪn/[1] THERR-ə-min; originally known as the ætherphone/etherphone, thereminophone[2] or termenvox/thereminvox) is an early electronic musical instrument controlled without physical contact by the thereminist (performer)” - Wikipedia
- Voglio controllare il suono emesso dal buzzer con una fotoresistenza
  - la fotoresistenza l'abbiamo vista alla lezione 2
  - bisogna convertire il range dell'analogRead nel range di tone
    - value=map(value, 0, 1023, 50, 10000); //per esempio



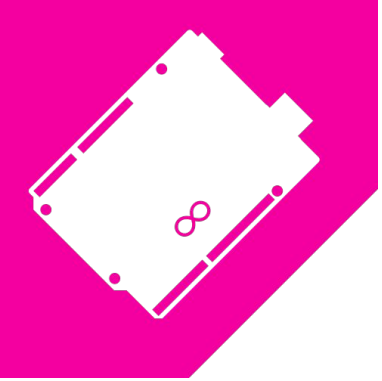
LightTeremin Soluzione

```
void setup() {
    //non serve niente
}

void loop() {
    int note = map(analogRead(A0), 0, 1023, 50, 10000);
    tone(8, note);
}
```



RGBled

- Nel kit di oggi è incluso un led rgb, già fornito di resistenze
  - possiamo attaccarlo direttamente ad arduino
  - un led rgb contiene all'interno 3 led separati
    - infatti si controlla con 4 piedini: - → GND, R → Rosso, G → Verde, B → Blu
- Voglio accendere e spegnere i led
- Hint: per accendere un led lo si attacca ad un pin, si imposta il pin come output, e si mette il pin nello stato high
  - per accenderne 3 basta collegare il led a tre pin differenti e ripetere la procedura
- Hint: è possibile anche usare analogWrite al posto di digitalWrite per ottenere meno luminosità e combinare i colori
  - random(max) o random(min, max) ritornano un valore a caso nell'intervallo

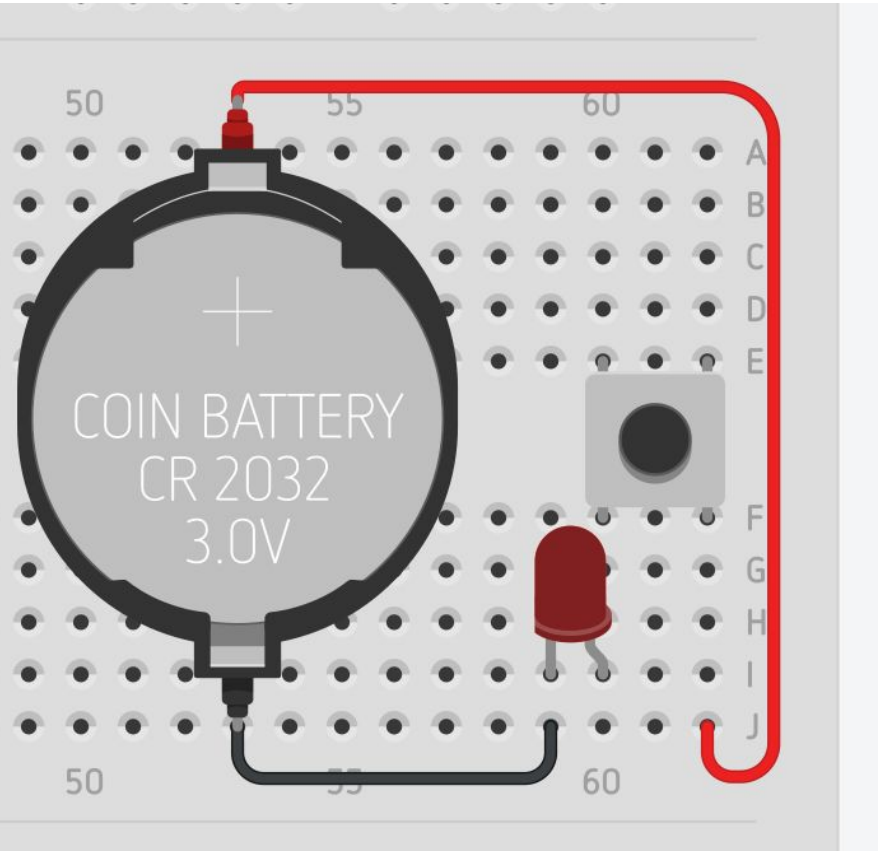


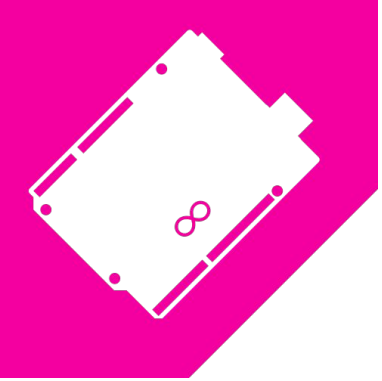
RGBled Soluzione

```
void setup() {
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop() {
  analogWrite(3, random(255));
  analogWrite(5, random(255));
  analogWrite(6, random(255));
}
```

Buttons

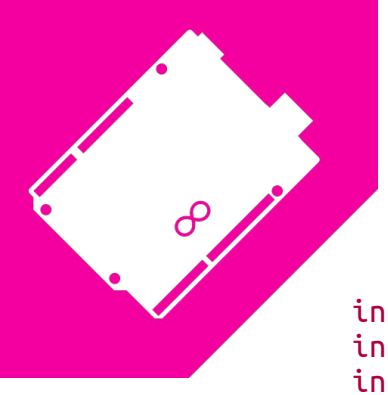
- Un bottone, quando è premuto, conduce elettricità fra i due piedini posti sullo stesso lato
  - attenzione quando lo mettete nella breadboard perché può essere un po' duro





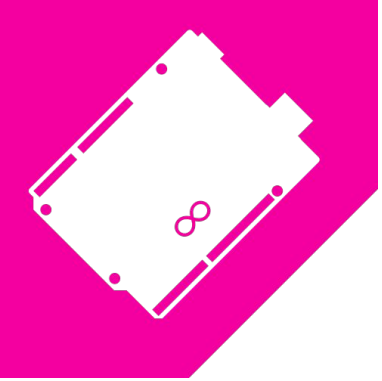
Buttons - 2

- Voglio leggere via seriale quando un bottone su un pin INPUT_PULL viene premuto o rilasciato
  - il bottone collega il pin a GND
- Voglio leggere solo una volta per pressione il messaggio in seriale
  - se stampo in seriale semplicemente il risultato di digitalRead, ad ogni esecuzione di loop il messaggio verrà rimandato
  - ho bisogno di ricordarmi lo stato precendente del bottone, e mandare un messaggio solo quando lo stato attuale cambia rispetto al precendete



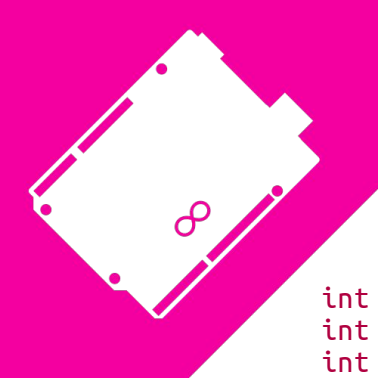
Buttons Soluzione

```
int button1 = 4; //pin for button 1
int button2 = 5; //pin for button 2
int but1_status = HIGH;
int but2_status = HIGH;
      //variables to save previous buttons state
void setup(){
      pinMode(button1, INPUT_PULLUP);
            //default state of button1 is HIGH
      pinMode(button2, INPUT_PULLUP);
            //default state of button2 is HIGH
      Serial.begin(9600);
}

void loop(){
      int but1_now=digitalRead(button1);
      if(but1_now != but1_status){
            if(but1_now==HIGH){
                  Serial.println(“bottone 1 rilasciato!”);
            }else{
                  Serial.println(“bottone 1 premuto!”);
            }
            but1_status=but1_now;
      }

      int but2_now=digitalRead(button2);
      if(but2_now != but2_status){
            if(but2_now==HIGH){
                  Serial.println(“bottone 2 rilasciato!”);
            }else{
                  Serial.println(“bottone 2 premuto!”);
            }
            but2_status=but2_now;
      }

      delay(300);
}
```



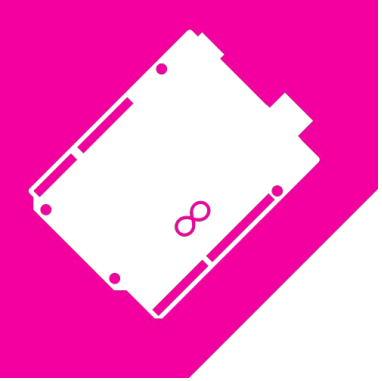
Tea Timer

- Qualcosa di più complicato
- Voglio programmare un timer per il tè, che mi avvisi con il buzzer quando è il momento di levare la bustina
- il programma ha un conto alla rovescia. ogni volta che premo il pulsante il conto alla rovescia dovrebbe essere aumentato di 10 secondi
- quando il conto alla rovescia è esaurito, il programma dovrebbe emettere 3 toni con il buzzer
- Hint: usare delay() renderebbe più complicato l'algoritmo, è meglio usare millis() per confrontarlo con l'istante in cui dovrebbe scadere il conteggio



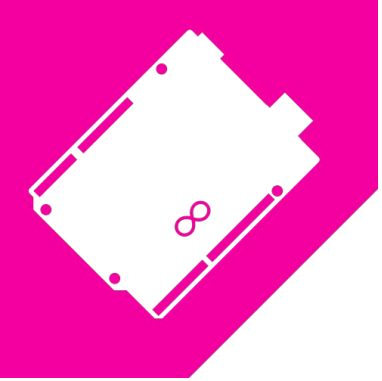
Tea Timer Soluzione

```
int but = 4;
int but_status = HIGH;
int bzz = 8;

long future;
      //here we will save when the timer will expire
void setup(){
      pinMode(but, INPUT_PULLUP);
      int first_press=digitalRead(but);
      while(first_press==HIGH){
            delay(50);
            first_press=digitalRead(but);
                  //wait a little and read again
      }
            //button pressed! we can start the timer
      future=millis() + 10000;
}

void loop(){
      int but_now=digitalRead(but);
      if(but_now == LOW && but_s == HIGH){
            //we have a press if the previous state was different from this state
            future = future + 10000;
                  //increment timer deadline
      }
      but_status=but_now;
            //remember button state

      if(millis() > future){
            //timer finished! it's time to buzz
            tone(8, 440, 500);
            delay(500 * 1.3);
            tone(8, 660, 500);
            delay(500 * 1.3);
            tone(8, 990, 1000);
            delay(1000 * 1.3);
            noTone(8);
      }
}
```



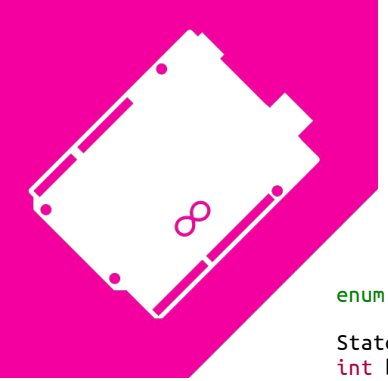
Cosa proviamo oggi? - 3

- sciacquone a stati
- simone dice



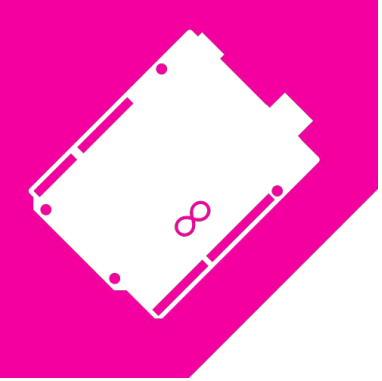
Sciacquone a stati

- usiamo una macchina a stati per implementare uno sciacquone
- praticamente completiamo il codice di prima
- usiamo un buzzer per fare il rumore dell'acqua che scarica, e dell'acqua che ricarica
  - vedi esercizi buzzer
  - hint: tone(8, random(50, 10000)); fa un bel rumore
- e un bottone per simulare il pulsante
  - vedi esercizi button



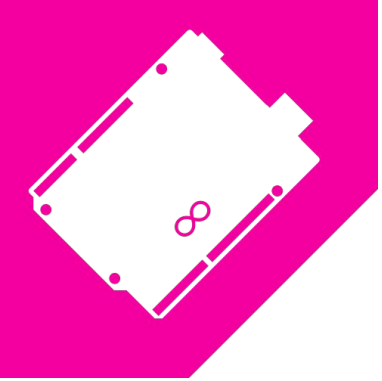
soluzione

```
enum StatoSciacquone {SCIACQUONABILE, SCARICO_ACQUA, RICARICO_ACQUA};

StatoSciacquone wc;
int buzz=8;
int button = 9;

void setup(){
        pinMode(button, INPUT_PULLUP);
        wc=SCIACQUONABILE;
}

void ricaricaAcqua(){
        for(int i=0; i<5; i++){
                tone(buzz, 1000 + i*100, 100);
                delay(1000);
        }
        noTone(buzz);
}

void scaricaAcqua(){
        long future= millis() + 4000;
        while(millis()<future){
                tone(buzz, random(50, 10000));
                delay(10);
        }
        noTone(buzz);
}

void aspettaCatenella(){
        while(digitalRead(button)==LOW);
                //wait for button to be in a rest state
                //here button is HIGH
        while(digitalRead(button)==HIGH);
                //wait for a press
}

void loop(){
        switch(wc){
        case RICARICO_ACQUA:
                ricaricaAcqua();
                wc=SCIACQUONABILE;
                break;
        case SCARICO_ACQUA:
                scaricaAcqua();
                wc=RICARICO_ACQUA;
                break;
        case SCIACQUONABILE:
                aspettaCatenella();
                wc=SCARICO_ACQUA;
                break;

        }
}
```



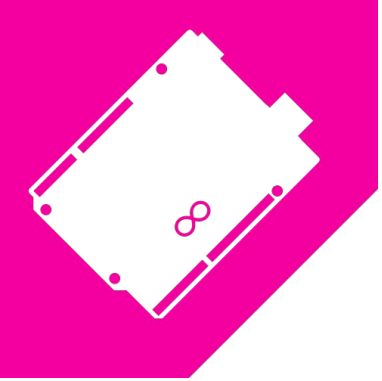
simone dice

- simon says, il gioco come è spiegato nelle slide
- un led rgb, 3 bottoni. niente più
- magari anche un buzzer, no?



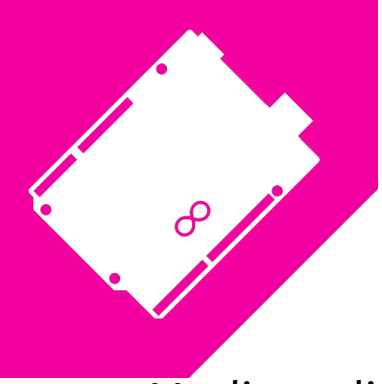
simone dice soluzione

```
int difficolt;
long istante_spegnimento;
int led_color;
//0 per rosso, 1 per verde, 2 per blu

//variabili di stato per i bottoni
int rb_status=HIGH;
int gb_status=HIGH;
int bb_status=HIGH;


//pin dei componenti
int red_pin=3;
int green_pin=4;
int blue_pin=5;
int red_button=6;
int green_button=7;
int blue_button=8;
int buzz = 9;

void setup(){
    pinMode(red_pin, OUTPUT);
    pinMode(green_pin, OUTPUT);
    pinMode(blue_pin, OUTPUT);
    pinMode(red_button, INPUT_PULLUP);
    pinMode(green_button, INPUT_PULLUP);
    pinMode(blue_button, INPUT_PULLUP);

    difficolt = 0;
    istante_spegnimento = 0;
//sporco trucco: cosi obbligo la prima esecuzione di loop a scegliere il colore del led
}

void loop() {
    //parte 1: scelta del nuovo colore
    if (millis() > istante_spegnimento) {
        digitalWrite(red_pin, LOW);
        digitalWrite(green_pin, LOW);
        digitalWrite(blue_pin, LOW);
        led_color = random(3);
            //scegli il nuovo colore
        digitalWrite(led_color + red_pin, HIGH);
        istante_spegnimento = millis() + 2000 - difficolt * 100;
            //spegni il led fra 2 secondi, meno se sei stato bravo
    }


    //parte 2: leggo l'input dei bottoni (ma solo uno per loop viene registrato)
    int button_pressed=-1;
        //-1 per nessun bottone premuto, 0 per rosso, 1 per verde, 2 per blu
    if(rb_status == HIGH && digitalRead(red_button) == LOW){
        button_pressed=0;
    }else if(gb_status == HIGH && digitalRead(green_button) == LOW){
        button_pressed=1;
    }else if(bb_status == HIGH && digitalRead(blue_button) == LOW){
        button_pressed=2;
    }
        //update buttons
    rb_status = digitalRead(red_button);
    gb_status = digitalRead(green_button);
    bb_status = digitalRead(blue_button);


    //parte 3: controllo se ho dato una risposta sbagliata o giusta
    if(button_pressed != -1){
        if(button_pressed==led_color){
            //win!
            difficolt++;
            istante_spegnimento=0;
            noTone(8);
            tone(buzz, 440, 500);
        }else{
            //lose
            istante_spegnimento=0;
            noTone(8);
            tone(buzz, 100, 1000);
        }
    }

}
```



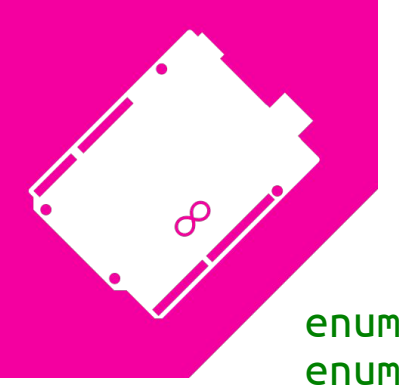
Cosa proviamo oggi? - 4

- Cassaforte



Cassaforte

- Voglio realizzare un sistema di cassaforte con arduino in cui la combinazione per aprire la inserisco tramite un potenziometro (un po' come le manopole delle cassaforti vere).
- -Per inserire i numeri tramite potenziometro posso dividere il range dei valori in segmenti (nel nostro caso 4 quindi da 0 a 255,256 a 511 ecc..) e premere un pulsante quando decido di inserire quel numero.
- -Per rendere più facile l'inserimento conviene usare un buzzer che notifica l'utente quando sta passando da un segmento ad un altro.
- -Quando l'arduino aspetta l'inserimento del primo numero il led si illumina di rosso, quando aspetta il secondo di verde ed infine per il terzo di blu.
- -Se l'utente inserisce il numero giusto la cassaforte aspetterà il prossimo numero giusto, altrimenti tornerà allo stato iniziale e cioè aspetterà il primo numero della combinazione.
- -Quando la combinazione inserita sarà giusta possiamo creare una breve animazione col buzzer e il led per notificarlo (ma se siamo creativi possiamo anche fare qualcosa di più figo con motori, relè, petardi o qualsiasi cosa ci venga in mente)
- https://www.youtube.com/watch?v=Iq0LVFTbjg4&feature=youtu.be per vedere il video di una soluzione

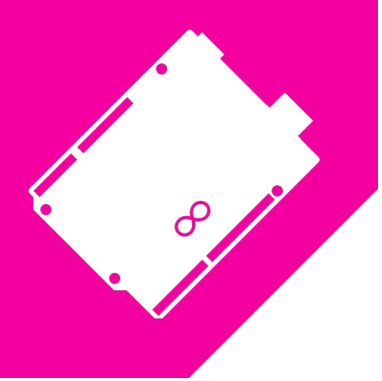


Cassaforte – parte 1

```
enum StatiCassaforte {ROSSO, VERDE, BLU};
enum StatiManopola {UNO, DUE, TRE, QUATTRO};
StatiCassaforte stato = ROSSO;
StatiManopola manopola = UNO;

int pinPotenziometro = A0;
int pinRosso = 7;
int pinVerde = 8;
int pinBlu = 9;

int pinBuzzer = 6;
int pinBottone = 10;
boolean premuto = false;

void setup() {
  pinMode(pinRosso, OUTPUT);
  pinMode(pinVerde, OUTPUT);
  pinMode(pinBlu, OUTPUT);
  pinMode(pinBuzzer, OUTPUT);
  pinMode(pinBottone, INPUT_PULLUP);
  Serial.begin(9600);

}
```

Cassaforte - parte 2

```
void loop() {


  StatiManopola newManopola = aggiornaManopola(analogRead(pinPotenziometro));

  if (newManopola != manopola)
    tone(pinBuzzer, 700, 50);
  manopola = newManopola;
  Serial.println(analogRead(pinPotenziometro));
  switch (stato)
  {
    case ROSSO:
      digitalWrite(pinRosso, HIGH);
      digitalWrite(pinVerde, LOW);
      digitalWrite(pinBlu, LOW);
      if (leggi_bottone())
      {
        if (manopola == TRE)
          stato = VERDE;
      }


      break;

    case VERDE:
      digitalWrite(pinRosso, LOW);
      digitalWrite(pinVerde, HIGH);
      digitalWrite(pinBlu, LOW);
      if (leggi_bottone())
      {
        if (manopola == UNO)
          stato = BLU;
        else
          stato = ROSSO;
      }
      break;

    case BLU:
      digitalWrite(pinRosso, LOW);
      digitalWrite(pinVerde, LOW);
      digitalWrite(pinBlu, HIGH);
      if (leggi_bottone())
      {
        if (manopola == QUATTRO)
        {
          stato = ROSSO;
          youWin();
        }
        else
          stato = ROSSO;
      }
      break;

    default:
      Serial.println("Se sei entrato qui è successo qualcosa di molto strano");
      break;
  }
  delay(50);
}
```



Cassaforte - parte 3

```
StatiManopola aggiornaManopola(int potenz)
{
  if (potenz < 256)
    return UNO;
  if (potenz < 512)
    return DUE;
  if (potenz < 768)
    return TRE;
  return QUATTRO;
}

boolean leggi_bottone()
{

  if (digitalRead(pinBottone) == LOW)
  {
    if (!premuto)
    {
      premuto = true;
      tone(pinBuzzer, 200, 50);
      return true;
    }
  }
  else
  {
    premuto = false;
  }

  return false;
}

void youWin()
{
  tone(pinBuzzer, 700);
  digitalWrite(pinBlu, LOW);
  digitalWrite(pinRosso, HIGH);
  delay(550);
  tone(pinBuzzer, 860);
  digitalWrite(pinVerde, HIGH);
  digitalWrite(pinRosso, LOW);
  delay(550);
  tone(pinBuzzer, 900);
  digitalWrite(pinVerde, LOW);
  digitalWrite(pinBlu, HIGH);
  delay(1580);
  noTone(pinBuzzer);
}
```