



corkino

User Manual



arduino based USB controller



HARDWARE A

corkino

componenti

circuiti

SOFTWARE B

integrazione software

arduino to MaxMSP

interfaccia grafica

gen~ patches.comb filter

codice comb.gendsp

A

HARDWARE

info

Le seguenti istruzioni sono concepite per rendervi famigliari gli elementi di comando del device, in modo che possiate conoscere tutte le sue funzioni. Dopo aver letto attentamente le istruzioni, conservatele in modo da poterle rileggere ogni volta che ne abbiate bisogno.

Idealizzato e prodotto da Gabriela Krassimirova Velitchkova


27/06/2021


a Filomena e i suoi giri sul GRA

ATTENZIONE: per ridurre il rischio di incendio o di scariche elettriche, non esponete questo apparecchio a pioggia o umidità. Per ridurre il pericolo di scariche elettriche evitate di rimuoverne il coperchio o il pannello inferiore. Non esistono all'interno dell'apparecchio parti la cui regolazione è a cura dell'utente. Per eventuale assistenza, fate riferimento esclusivamente a personale qualificato. Per maggiori informazioni contattare info@velitch.com

A

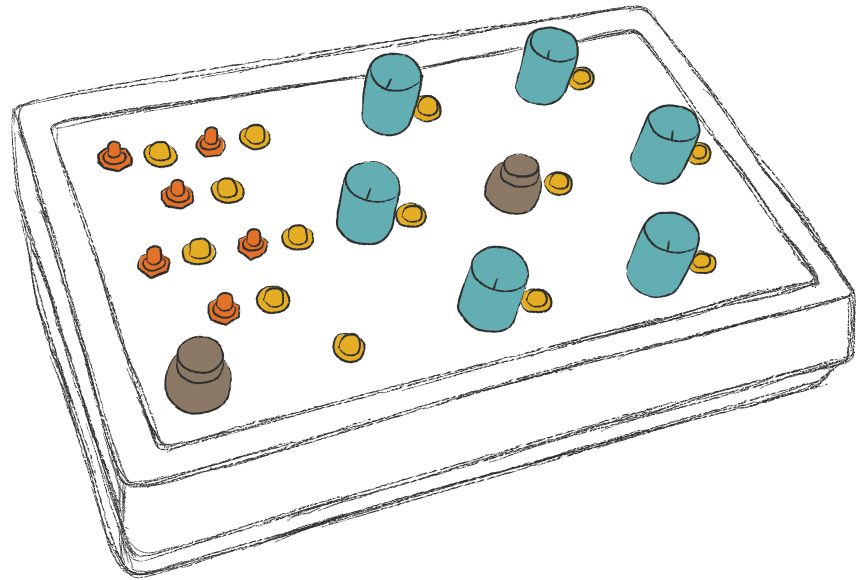
corkino

 push button

 potenziometri

 I/O switch

 led



6

componenti



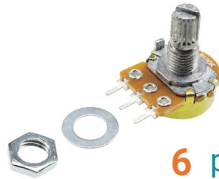
1 arduino uno



2 I/O switch



6 push button



6 potenziometri B100K



14 led



8 resistenze 1 MOhm

14 resistenze 220 Ohm

contenitore

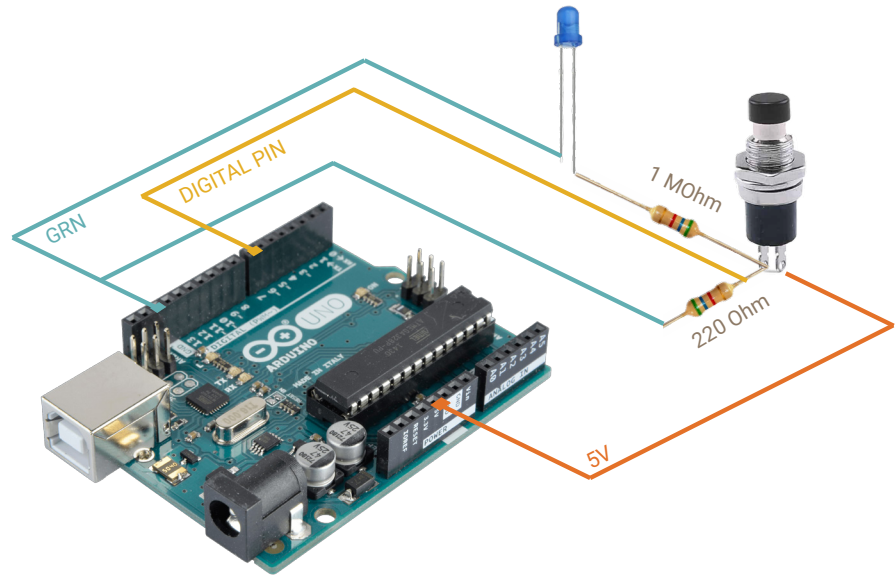
cavi "Filomena"

pellicola adesiva
"Cleopatra Negra"

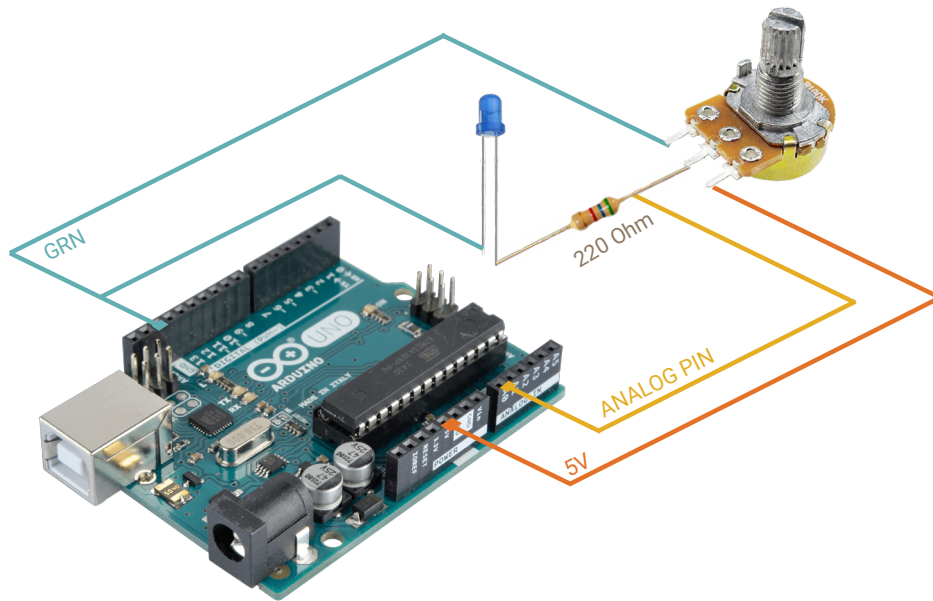
A

push
button
&
led

circuiti

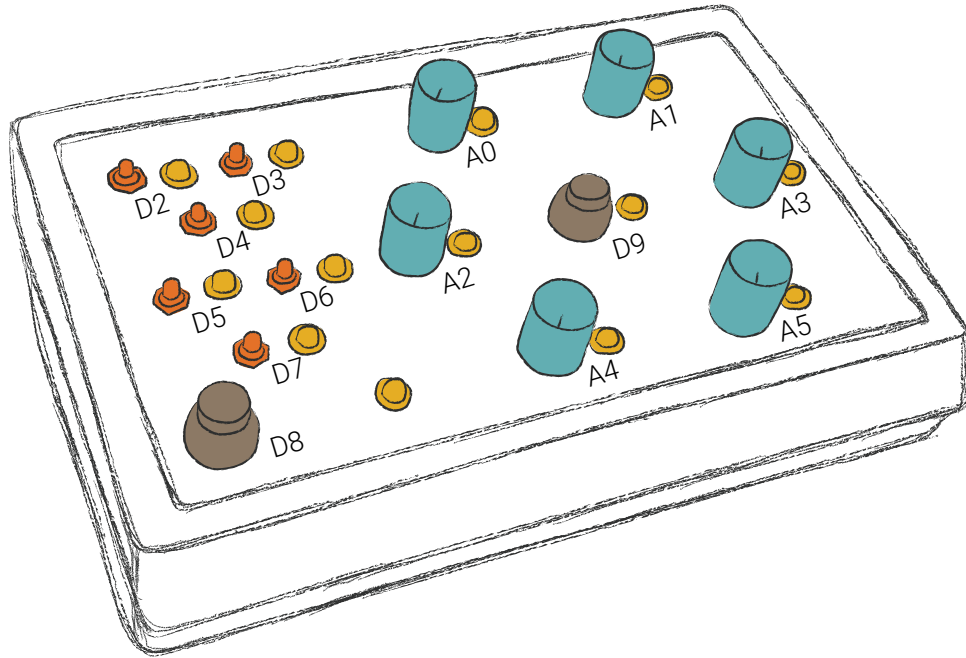


8



potentiometer
&
led

mapping



B

SOFTWARE

integrazione software

corkino è un controller USB con scheda arduino uno integrata.

Sul sito ufficiale di arduino è disponibile la sezione [download](#), dalla quale possiamo scaricare l'**IDE** per il sistema operativo che utilizziamo (sono supportati Windows, Linux e Mac OS X).

Grazie all'uso della **porta seriale**, **corkino** è in grado di inviare dati a qualsiasi software che utilizzi questo tipo d'interfaccia.

Una utilissima funzionalità dell'IDE è il monitor della porta seriale disponibile nella voce di menu Strumenti > Monitor seriale. Tale strumento si rivelerà molto importante per l'integrazione del software di arduino con software terzi.

arduino to MaxMSP

Per iniziare ad usare le gen~ patches di **corkino** è necessario aver installato sul PC il software MaxMSP disponibile a pagamento a questo [link](#).

Se MaxMsp è installato, basta collegare il dispositivo tramite cavo USB ed avviare la patch `corkino.maxpat`

Le **gen~ patches** di **corkino** sono fornite di un'interfaccia grafica interattiva (pag.13). Le gen~ patches sono modificabili in ogni loro funzione. In questo manuale vengono esplicate le impostazioni di default. Per ulteriori informazioni contattare info@velitch.com

Per interfacciare il vostro **corkino** con il software MaxMsp, tramite interfaccia grafica selezionare la porta seriale COM_ da voi utilizzata.

interfaccia grafica



gen~ patches: comb filter

corkino viene fornito con la gen~ patch `comb.gendsp` (codice a pag. 15-17).

Caratteristiche:

input: segnale, microfono.

sei pulsanti: controllo preset "ROOM FREQUENCY".

sei potenziometri: T60, high cut, random time, random amount, signal gain, effect gain.

2 pulsanti I/O: recall/save preset, random on/off.

gains: signal, effect, master.

serial port: COM_X

playbar

sonogram

codice comb.gendsp

```
fbcomb_lp_T60 (Xn, koeff, T60, freqcut, ms, rand)

{

Delay delay_1(samplerate);
Delay delay_2(samplerate);
Delay delay_3(samplerate);
History history_1(0);

//COMB

del_time = ((ms*koeff)+rand) * (samplerate/1000.);

potenza = ((-3. * (del_time / 1000.))) / T60;
g = pow(10., potenza);

delay1 = delay_1.read(del_time) * g;

feedback = Xn + delay1;
```



```

//LOWPASS

omega = ((freqcut * twopi) / samplerate);
cosw = cos(omega);
cosq = cosw * cosw;
alpha = (cosw - 1.) + (sqrt((cosq - (4. * cosw)) + 3.));
mul_1 = (1. - alpha) * history_1;

    LP = mul_1 + (feedback * alpha);

//ALLPASS1

menog = (-1.) * g;
allpass = (delay_2.read(del_time)) * menog;

    all1 = (delay_2.read(del_time)) + ((allpass + LP) * g);

//ALLPASS2

allpass2 = (delay_3.read(del_time)) * menog;

    Yn = (delay_3.read(del_time)) + ((allpass2 + all1) * g);

```

```
history_1_2 = fixdenorm(Yn);  
history_1 = history_1_2;  
delay_1.write(feedback);  
delay_2.write(allpass + LP);  
delay_2.write(allpass2 + all1);  
    return Yn;  
}
```




corkino

by Velitch



codebox

L'oggetto `codebox` permette la compilazione di algoritmi scritti in linguaggio di programmazione `GenExpr`. La sintassi del linguaggio `GenExpr` è simile a quella di C e JavaScript per istruzioni di espressioni semplici.



```
1 require "GKV_Filter_Library"; //Oxn, freq, T60, freqcut)
2
3 c1 = fbcomb_lp_T60 (in1, 1., in3, in4, in5, in6);
4
5 c2 = fbcomb_lp_T60 (in1, 2., in3, in4, in5, in6);
6
7 c3 = fbcomb_lp_T60 (in1, 3., in3, in4, in5, in6);
8
9 c4 = fbcomb_lp_T60 (in1, 4., in3, in4, in5, in6);
10
11 c5 = fbcomb_lp_T60 (in1, 5., in3, in4, in5, in6);
12
13 c6 = fbcomb_lp_T60 (in1, 6., in3, in4, in5, in6);
14
15 c7 = fbcomb_lp_T60 (in1, 7., in3, in4, in5, in6);
16
17 c8 = fbcomb_lp_T60 (in1, 8., in3, in4, in5, in6);
18
19
20
21
22 coutL = (c1*0.3)+(c2*0.75)+(c3*0.55)+(c4*0.4)+(c5*0.4)+(c6*0.5)+(c7*0.45)+(c8*0.52);
23 coutR = (c1*0.7)+(c2*0.25)+(c3*0.45)+(c4*0.6)+(c5*0.6)+(c6*0.5)+(c7*0.55)+(c8*0.48);
24
25 out1 = coutL;
26 out2 = coutR;
```

require

Quando si definiscono gli operatori in `GenExpr`, può essere utile conservarli in un file separato in modo che possano essere riutilizzati frequentemente senza dover riscrivere costantemente la definizione dell'operatore in un `codebox`.

Per includere gli operatori `GenExpr` definiti in un file separato, che d'ora in poi chiameremo libreria, si utilizza l'operatore `require`. L'operatore `require` prende il nome di un file `genexpr` e carica le sue definizioni nell'oggetto `codebox`.