



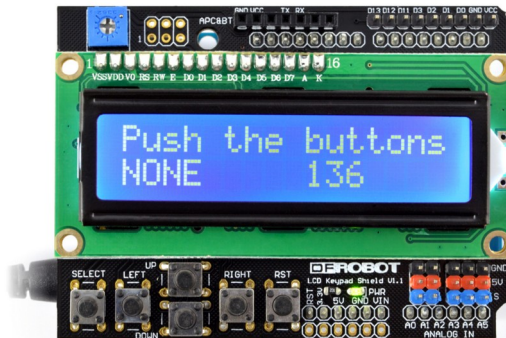
# LCD Shield – LCD SHIELD Menu



# LCD Shield – programmazione menu



- Idea: utilizzare una LCD shield 1602 per realizzare un progetto che richiedesse dei menu
- Scopo inserimento delle variabili (set-point etc)
- Soluzione : analizzando il video di Paolo Aliverti, comincio a studiare il funzionamento di questa scheda e la programmazione dei menu



Link per i video-tutorial di P. Aliverti

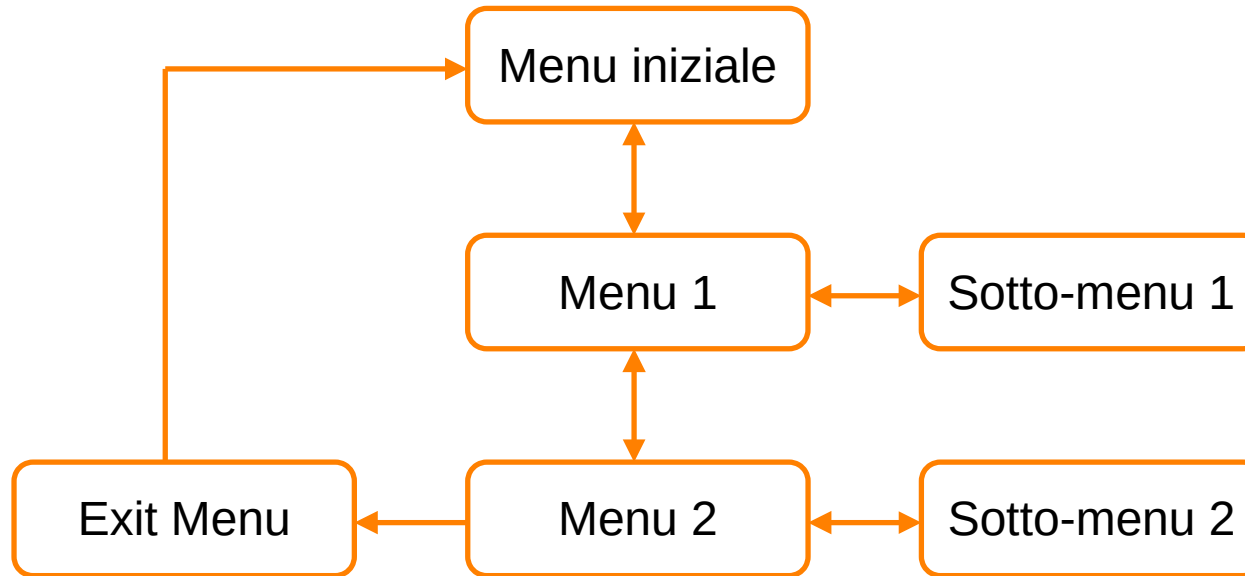
[Paolo Aliverti video 261](#)

[Paolo Aliverti video 262](#)

# LCD Shield – programmazione menu



- La prima cosa che dobbiamo stabilire è lo schema a blocchi dei menu che vogliamo realizzare

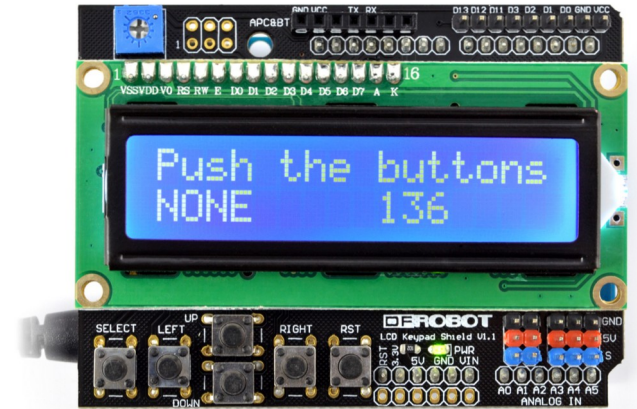


- Ovviamente lo schema può essere variato a nostro piacimento

# LCD Shield – programmazione menu



- Analizzando la shield, notiamo che ci sono 6 pulsanti
  - SEL
  - LEFT
  - RIGHT
  - UP
  - DOWM



più il pulsante RES (Reset)

- La scheda riesce a determinare il pulsante premuto tramite l'ingresso analogico A0 , in serie ai pulsanti sono inserite delle resistenze differenti che determinano una caduta di tensione
- Dobbiamo quindi preventivamente misurare la lettura A0 "AnalogRead(A0)" con un semplice sketch che legge, annotando i valori letti da Arduino

# LCD Shield – programmazione menu



## Sketch per lettura A0

```
File Modifica Sketch Strumenti Guida
Arduino Uno
A0-Lettura.ino
1 //per leggere il valore A0 ogni 500 ms
2
3 int lettura = 0;
4 int ingresso = A0;
5
6 void setup() {
7   Serial.begin(9600);
8   pinMode (ingresso,INPUT);
9 }
10
11 void loop() {
12   lettura = analogRead(ingresso);
13   delay(500);
14   Serial.println (lettura);
15 }
16
```

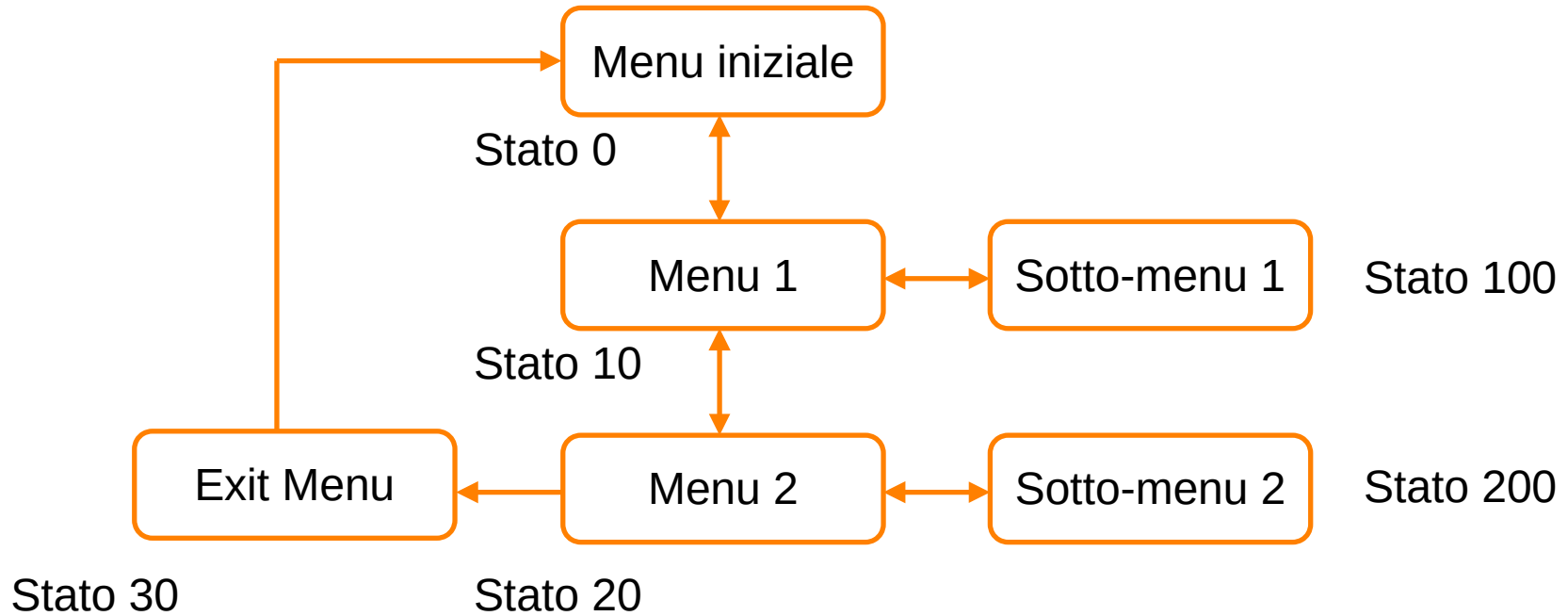
- Valori letti nel mio caso :

Botone	Lettura
Nessuno	1023
SEL	721
LEFT	480
RIGHT	0
UP	130
DOWN	307

# LCD Shield – programmazione menu



- Inseriamo il primo concetto (Stato del programma) che ci aiuterà alla navigazione all'interno dei vari menu:





# LCD Shield – programmazione menu



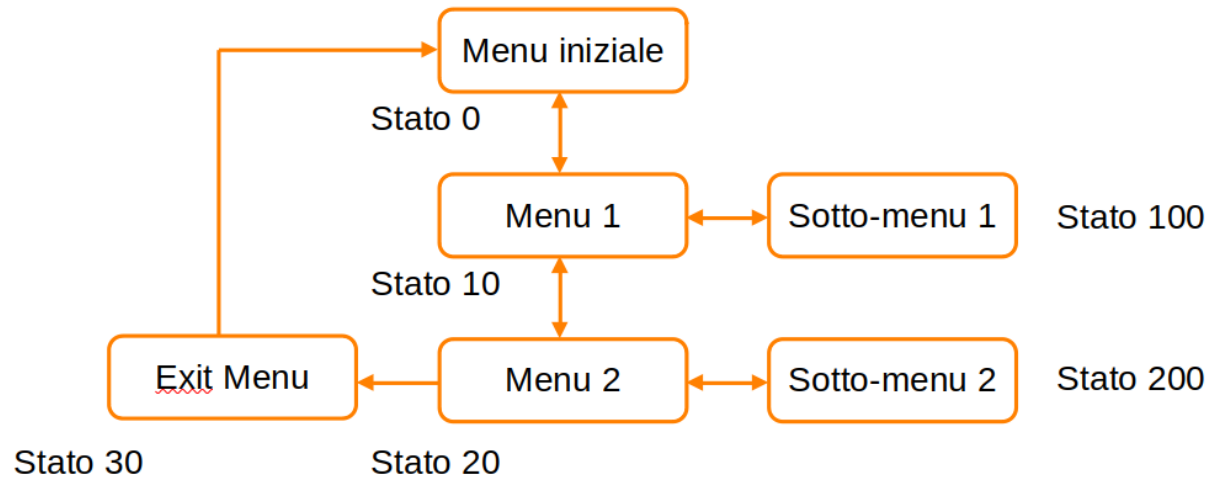
- Aliverti consiglia di creare la libreria “lcdshield.h”
- Questo per evitare tutte le volte di scrivere del codice per leggere e decodificare che tasto viene premuto.
- Per fare questo chiameremo solo la routine “readKeys()”
  - La quale leggendo il pin A0, ci ritornerà il tasto che è stato premuto

```
1  #ifndef lcdshield.h
2  #define lcdshield.h
3
4
5  #define KY_NONE 0
6  #define KY_LEFT 1
7  #define KY_RIGHT 2
8  #define KY_UP 3
9  #define KY_DOWN 4
10 #define KY_SEL 5
11 // th = soglia di lettura analogRead(a0)
12 #define TH_NONE 1023
13 #define TH_LEFT 480
14 #define TH_RIGHT 0
15 #define TH_UP 130
16 #define TH_DOWN 307
17 #define TH_SEL 721
18
19 int readKeys() {
20     int n = analogRead(A0);
21     if ((n >= TH_LEFT-5) && (n <= TH_LEFT+5)) {
22         return KY_LEFT;
23     } else if ((n >= TH_RIGHT-5) && (n <= TH_RIGHT+5)) {
24         return KY_RIGHT;
25     } else if ((n >= TH_UP-5) && (n <= TH_UP+5)) {
26         return KY_UP;
27     } else if ((n >= TH_DOWN-5) && (n <= TH_DOWN+5)) {
28         return KY_DOWN;
29     } else if ((n >= TH_SEL-5) && (n <= TH_SEL+5)) {
30         return KY_SEL;
31     } else return KY_NONE;
32 }
33
34
35 #endif
--
```

# LCD Shield – programmazione menu



- Iniziamo a creare la nostra struttura Menu, sottoforma di varie funzioni che inseriremo nel loop principale dello sketch:
  - mainloop = menu principale
  - mainmenu 10..30: menu a cascata
  - mainsubmenu 1 e 2 sottomenu



```
30 void loop() {
31   switch (stato) {
32     case 0:
33       mainloop();
34       break;
35     case 10:
36       mainmenu1();
37       break;
38     case 20:
39       mainmenu2();
40       break;
41     case 30:
42       mainmenuexit();
43       break;
44
45     case 100:
46       mainsubmenu1();
47       break;
48     case 200:
49       mainsubmenu2();
50       break;
51   }
52 }
```



# LCD Shield – programmazione menu

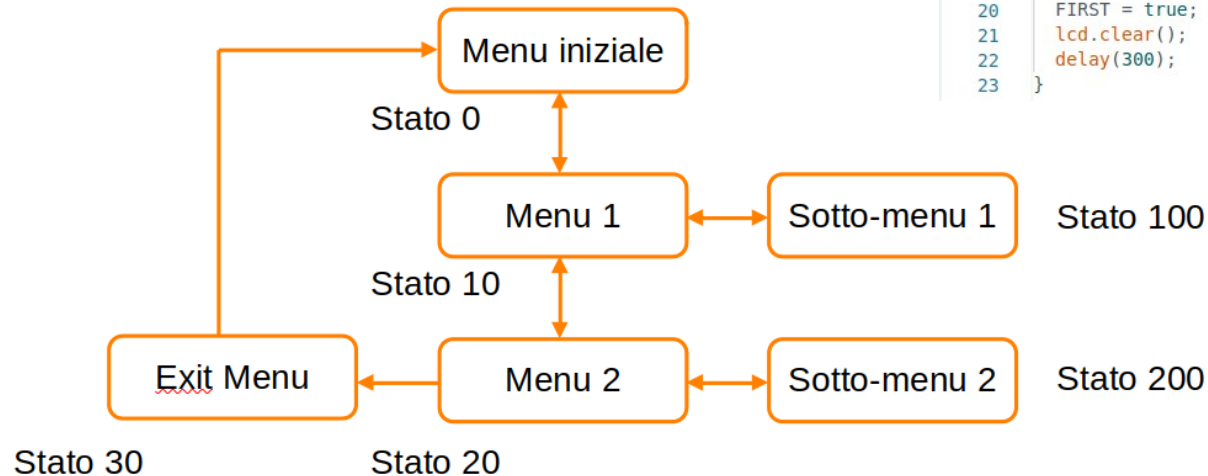


- Iniziamo a creare la nostra struttura Menu, sottoforma di varie funzioni che inseriremo nel loop principale dello sketch:
  - mainloop = menu principale
  - mainmenu 10..30: menu a cascata
  - mainsubmenu 1 e 2 sottomenu

Come passiamo da menù all'altro ?

Funzione `setStato()` è creata a questo scopo

```
15 |  
16 | int stato = 0; //variabile stato per passaggio tra i menu  
17 | //funzione di comodo per passare da uno stato all'altro navigando fra i vari menù  
18 | void setStato(int s) {  
19 |     stato = s;  
20 |     FIRST = true;  
21 |     lcd.clear();  
22 |     delay(300);  
23 | }
```





# LCD Shield – programmazione menu



- Una “finezza” che viene utilizzata nello sketch:
- Avete notato che viene utilizzata variabile FIRST di tipo booleano (Vero/Falso)
- Questo viene utilizzato per evitare che una volta entrato nel menu rispettivo, Arduino scriva di continuo delle stringhe nel sistema ma le scriva una sola volta al momento della prima entrata.

```
3 //utility per scrivere una sola volta le stringhe tra " "  
4 if (FIRST) //FIRST = true  
5 {  
6   lcd.print ("running");  
7   lcd.setCursor(0, 1);  
8   lcd.print ("S:menu");  
9   FIRST = false; //riportiamo true false  
10 }
```

- Nota importante: riportare sempre la variabile FIRST, nella condizione “false”

# LCD Shield – programmazione menu



*Domande ?*

