

Gli approfondimenti di

# L@BOROBOTICA

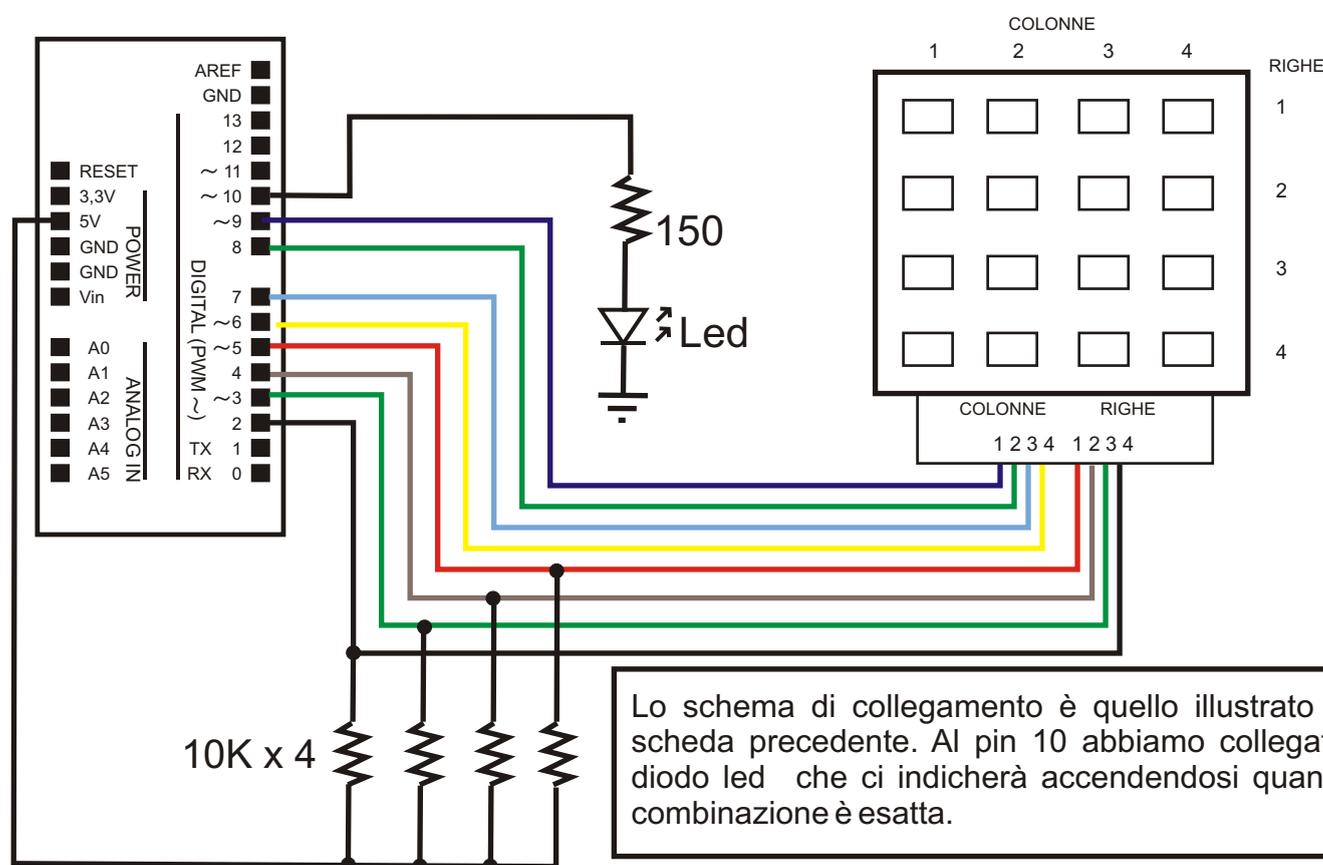
## CRIPTEX



Se avete letto “Il codice Da Vinci” di Dan Brown vi avrà sicuramente intrigato la descrizione del “Criptex”; una cassaforte a combinazione la cui progettazione Brown attribuisce a Leonardo.

Con l'aiuto del nostro fido Arduino e del tastierino 4x4, potremo dotarci di un criptex personale, non così fascinoso ma altrettanto efficace. Potremo usare il dispositivo per proteggere dagli intrusi l'ingresso a zone riservate, oppure per ricevere combinazioni che apriranno casseforti. Per fare tutto questo bisognerà che il numero inserito con la tastiera venga confrontato con quello segreto (che noi stabiliremo).

A questo punto nasce un problema: i numeri vengono decodificati da Arduino sequenzialmente, per esempio 2, 5, 6, mentre noi per il confronto con il codice segreto, avremo bisogno di avere un numero di almeno 5 cifre. In questa lezione vedremo come procedere.



```
//Sketch 29: CRIPTEX 24/10/2014
int C1=9; int C2=8; int C3=7; int C4=6;
int R1=5; int R2=10; int R3=3; int R4=2;
int led=10;
int numeri=0;
int codice_memorizzato=12369;
int confronto;
int codice=0;
char combi[5];
```

Il settaggio è identico alla scheda precedente; abbiamo aggiunto il codice memorizzato, che sarebbe il codice segreto da confrontare con quello immesso tramite tastiera, in più c'è anche un array di caratteri (char combi [5]) il quale ci permetterà di "allineare" in un unico numero tutte le singole cifre immesse da tastiera.

```
void setup()
{
  Serial.begin(9600);
  pinMode(R1, INPUT); digitalWrite(R1,HIGH);
  // R1 in ingresso con pull-up interno abilitato
  pinMode(R2, INPUT); digitalWrite(R2,HIGH);
  pinMode(R3, INPUT); digitalWrite(R3,HIGH);
  pinMode(R4, INPUT); digitalWrite(R4,HIGH);

  pinMode(C1, OUTPUT); pinMode(C2, OUTPUT);
  pinMode(C3, OUTPUT); pinMode(C4, OUTPUT);
  pinMode(led,OUTPUT);
  digitalWrite(C1, HIGH); digitalWrite(C2, HIGH);
  digitalWrite(C3, HIGH); digitalWrite(C4, HIGH);
  //Settaggio a stato alto delle colonne
}
```

```
void loop()
{
  selezione();
```

Per rendere più leggibile lo sketch tutta la routine di identificazione dei tasti è stata rinchiusa nella struttura selezione();

```
if (numeri>4)
```

Quando "numeri" diventa maggiore di 4 il programma entra nella struttura if.

```
{
  codice=((atoi (combi)) );
```

"Atoi" è un nuovo comando, converte un array di caratteri ascii in un integer (ASCII TO INTEGER)

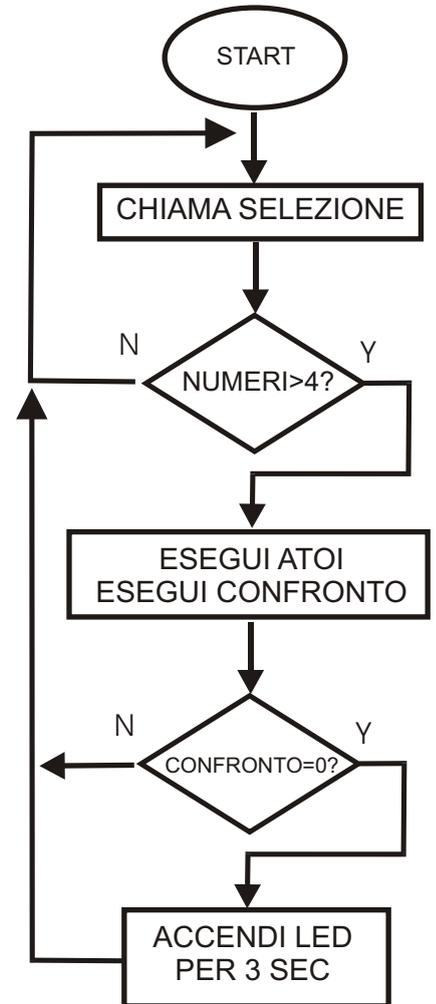
```
Serial.println(codice);
Serial.println(codice_memorizzato);
confronto=codice_memorizzato-codice;
Serial.println(confronto);
numeri=0;
if(confronto==0)
```

Questo è il punto di confronto tra il codice inserito e il codice segreto.

```
{
  digitalWrite(led,HIGH);
  delay(3000);
  digitalWrite(led,LOW);
  codice=0;
  numeri=0;
```

Se la sottrazione tra i due numeri dà zero accende il led.

Azzera numeri, pronto per ripartire



Commentiamo ora insieme la struttura “selezione()” con la gestione dell’array combi e il tasto cancelletto che viene usato per il reset.

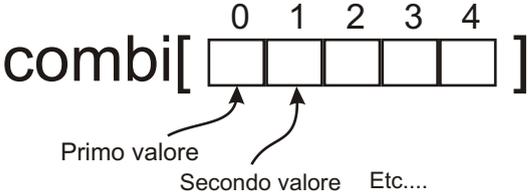
```
void selezione()
{
  delay(40);

  //*****
  //Seleziono la prima colonna
  digitalWrite(C1, LOW);
  //Controllo il primo pulsante della prima riga
  if(digitalRead(R1) == LOW)
  {
    Serial.println("1");
    combi[numeri]='1';

    numeri=numeri+1;

    //fintanto che il pulsante è premuto
    //(LOW) non uscire dal ciclo while
    while(digitalRead(R1) == LOW) ;
    delay(40);
  }
  //*****
  /*
```

In questo punto viene “caricato” l’array con la serie di numeri digitati. Si comincia con il primo, in questo caso 1. Il contatore dell’array viene così incrementato (numeri=numeri+1). Il successivo numero digitato viene immagazzinato nella seconda posizione dell’array e così via. Quando numero raggiunge il valore di 5, viene fatto l’atoi per convertire la serie di caratteri ascii in integer.



La sezione fra gli asterischi deve essere duplicata per ogni tasto (come nello sketch 28). L’unica sezione leggermente differente è quella relativa al tasto cancelletto, che ci servirà per resettare il contatore numeri (in caso di errori di digitazione). Ecco la sezione: (naturalmente con la terza colonna attivata) \*/

```
//*****
if(digitalRead(R4) == LOW)
{
  Serial.println("#");
  numeri=0; //azzera il contatore dell'array
  while(digitalRead(R4) == LOW);
  delay(40);
}
//*****
```

