

## Hex KeyPad >> 16Key4x4MembraneSwitchKeyBoard v1.2

Contact resistance of 500 ohm  
Insulation resistance 100Mohm  
Key Operating Force 150-200N  
Rebound time 1 (ms)  
Life of 100 million (times)  
Operating Temperature 60grade

Circuit Rating: 35V (DC), 100mA, 1W  
Contact resistance: 10 ~ 500  
(Varies according to the lead lengths and different from those of the material used)  
Insulation resistance: 100M 100V  
Dielectric Strength: 250VRms (50 ~ 60Hz 1min)  
Electric shock jitter: <5ms  
Life span: tactile type: 1 million times

Operating pressure: Touch feeling: 170 ~ 397g (6 ~ 14oz)  
Switch travel: touch-type: 0.6 ~ 1.5mm

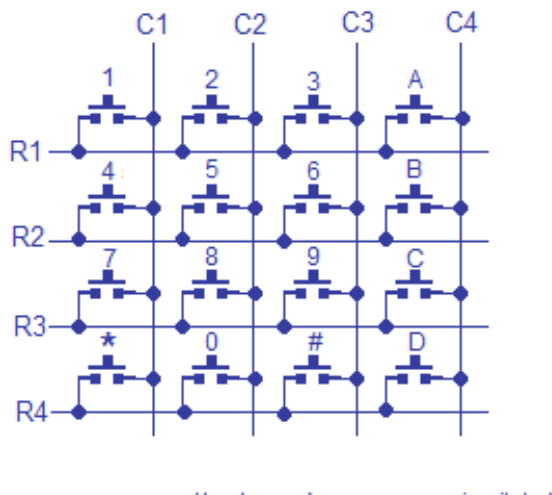
Operating temperature: -40 to +80  
Storage temperature: -40 to +80  
Temperature: from 40,90% to 95%, 240 hours  
Vibration: 20G, max. (10 ~ 200Hz, the Mil-SLD-202 M204.Condition B)

### Tastierino esadecimale.

Il tastierino esadecimale non è altro che una disposizione di 16 pulsanti in una matrice di forma 4X4. Normalmente un tastierino esadecimale avrà i pulsanti per i numeri 0,1,2,3,4,5,6,7,8,9 e le lettere A,B,C,D,\*,#.

Il tastierino esadecimale avrà una connessione ad 8 fili, chiamati R1, R2,R3,R4, e C1,C2,C3,C4 che stanno a rappresentare rispettivamente le righe e le colonne.

Il diagramma schematico e la foto di un tipico tastierino esadecimale sono mostrate nella figura sottostante.



Il programma identifica il pulsante premuto tramite un metodo chiamato column scanning, con questo metodo una riga in particolare viene tenuta a valore basso mentre le altre a valore alto.

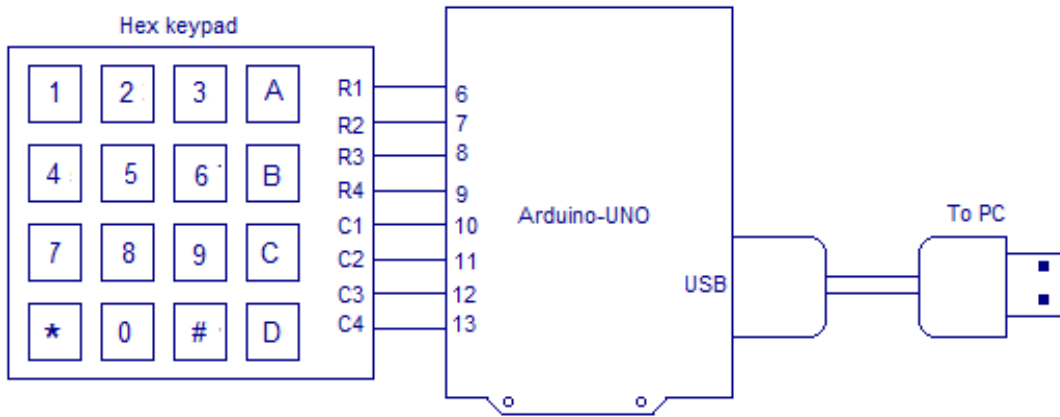
Lo stato logico di ogni colonna viene scannerizzato, e se una colonna in particolare viene rilevata bassa, allora significa che il pulsante che si trova nel mezzo di quella colonna e quella riga è short(premuto). Quindi il programma registra che quel pulsante viene premuto.

La stessa procedura viene eseguita per le susseguenti righe e l'intero processo viene ripetuto.

Per esempio se la riga numero 1 viene tenuta bassa e la colonna numero 1 viene rilevata bassa durante lo scanning, allora significa che il pulsante "1" è stato premuto.

L'intero diagramma circuitale per l'interfacciamento del tastierino esadecimale è mostrato nella figura sottostante.

### Diagramma del circuito.



Le righe R1,R2,R3 e R4 sono interfacciate rispettivamente ai pin digitali 6,7,8 e 9 di Arduino.

Le colonne C1,C2,C3 e C4 sono interfacciate ai pin digitali 10,11,12 e 13 di Arduino.

L' Arduino è connesso al PC tramite la porta USB.

Il circuito è alimentato dalla stessa USB e non sono necessarie alimentazioni esterne.

L'intero programma per l'interfacciamento del tastierino esadecimale ad arduino è dato qui sotto.

### Programma (1° versione)

```
int r1=6;
int r2=7;
int r3=8;
int r4=9;
int c1=10;
int c2=11;
int c3=12;
int c4=13;
int colm1;
int colm2;
int colm3;
int colm4;

void setup()
{
  pinMode(r1,OUTPUT);
  pinMode(r2,OUTPUT);
  pinMode(r3,OUTPUT);
  pinMode(r4,OUTPUT);
  pinMode(c1,INPUT);
  pinMode(c2,INPUT);
  pinMode(c3,INPUT);
  pinMode(c4,INPUT);
  Serial.begin(9600);
  digitalWrite(c1,HIGH);
  digitalWrite(c2,HIGH);
  digitalWrite(c3,HIGH);
  digitalWrite(c4,HIGH);
}

void loop()
{
  digitalWrite(r1,LOW); // la riga 1 viene impostata a valore basso, mentre le altre rimangono a valore alto
  digitalWrite(r2,HIGH);
  digitalWrite(r3,HIGH);
  digitalWrite(r4,HIGH);
  colm1=digitalRead(c1);
  colm2=digitalRead(c2);
  colm3=digitalRead(c3);
  colm4=digitalRead(c4);
  if(colm1==LOW)
  {Serial.println("1"); // Mostra il valore "1" sul display se la prima colonna e la prima riga sono a
    // valore basso
    delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
  else
  {
    if(colm2==LOW)
    {Serial.println("2"); // Mostra il valore "2" sul display se la seconda colonna e la prima riga sono a
      // valore basso
      delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
    else
    {
      if(colm3==LOW)
      {Serial.println("3"); // Mostra il valore "3" sul display se la terza colonna e la prima riga sono a
        // valore basso
        delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
      else
```

```

{
if(colm4==LOW)
{Serial.println("A"); // Mostra il valore "A" sul display se la prima colonna e la prima riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
}}

digitalWrite(r1,HIGH); //la riga 2 viene impostata a valore basso mentre le altre rimangono a valore alto
digitalWrite(r2,LOW);
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
colm1=digitalRead(c1);
colm2=digitalRead(c2);
colm3=digitalRead(c3);
colm4=digitalRead(c4);
if(colm1==LOW)
{Serial.println("4"); // Mostra il valore "4" sul display se la prima colonna e la seconda riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm2==LOW)
{Serial.println("5"); // Mostra il valore "5" sul display se la seconda colonna e la seconda riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm3==LOW)
{Serial.println("6"); // Mostra il valore "6" sul display se la terza colonna e la seconda riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm4==LOW)
{Serial.println("B"); // Mostra il valore "B" sul display se la quarta colonna e la seconda riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
}}
}}

digitalWrite(r1,HIGH); //la riga 3 viene impostata a valore basso mentre le altre rimangono a valore alto
digitalWrite(r2,HIGH);
digitalWrite(r3,LOW);
digitalWrite(r4,HIGH);
colm1=digitalRead(c1);
colm2=digitalRead(c2);
colm3=digitalRead(c3);
colm4=digitalRead(c4);
if(colm1==LOW)
{Serial.println("7"); // Mostra il valore "7" sul display se la prima colonna e la terza riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm2==LOW)
{Serial.println("8"); // Mostra il valore "8" sul display se la seconda colonna e la terza riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm3==LOW)
{Serial.println("9"); // Mostra il valore "9" sul display se la terza colonna e la terza riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm4==LOW)
{Serial.println("C"); // Mostra il valore "C" sul display se la quarta colonna e la terza riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
}}
}}

digitalWrite(r1,HIGH); //la riga 4 viene impostata a valore basso mentre le altre rimangono a valore alto
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,LOW);
colm1=digitalRead(c1);
colm2=digitalRead(c2);
colm3=digitalRead(c3);
colm4=digitalRead(c4);
if(colm1==LOW)
{Serial.println("*"); // Mostra il valore "*" sul display se la prima colonna e la quarta riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{

```

```

if(colm2==LOW)
{Serial.println("0"); // Mostra il valore "0" sul display se la seconda colonna e la quarta riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm3==LOW)
{Serial.println("#"); // Mostra il valore "#" sul display se la terza colonna e la quarta riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate
else
{
if(colm4==LOW)
{Serial.println("D"); // Mostra il valore "D" sul display se la quarta colonna e la quarta riga sono a
// valore basso
delay(200);} //imposta un ritardo per evitare pressioni multiple indesiderate

}}}
}

```

### Riguardo il programma.

Code Serial.begin(9600); imposta il baud rate per la comunicazione seriale a 9600; Il baud rate è il numero di cambiamenti che avvengono in un secondo in una linea di trasmissione modulata digitalmente.

Prima di tutto, la riga numero 1 viene tenuta a valore alto, e le altre righe vengono tenute a valore basso tramite il comando digitalWrite, quindi lo stato di ogni colonna viene letto tramite il comando digitalRead. Ogni colonna viene controllata per verificare che il suo stato sia basso usando un ciclo if-else.

Se una colonna in particolare viene trovata a valore basso, il pulsante che si trova nell'intersezione di quella colonna e della riga numero 1 viene considerato premuto, e il nome di quel pulsante viene mostrato nel monitor seriale usando il comando Serial.print .

Un ritardo di 200ms viene dato tra i loop che verificano le condizioni per evitare registrazioni multiple di pulsanti premuti una sola volta.

Se questo ritardo venisse aumentato ulteriormente la velocità di risposta del tastierino diminuirebbe.

Dopo vari tentativi, ho scoperto che il valore ottimale per il mio schema è di 200ms.

A questo punto la riga numero 2 viene impostata bassa, e le altre righe vengono impostate alte.

La scannerizzazione delle colonne viene ripetuto usando il ciclo if else, quindi la stessa cosa viene fatta per le righe numero 3 e 4, e l'intero ciclo si ripete nel tempo.

Il risultato avrà il nome del pulsante premuto e verrà mostrato nel monitor seriale in qualsiasi momento.

### Versione semplificata del programma visto sopra ( 2°Versione )

Possiamo facilmente semplificare il programma scritto sopra con un uso intelligente degli array e dei cicli for.

Aggiungiamo una versione semplificata del programma ( il quale ha la metà delle righe rispetto al programma visto sopra).Puoi comparare i due codici per cercare di capirne l'argomento.

```

int row[]={6,7,8,9}; // Defining row pins of keypad connected to Arduino pins
int col[]={10,11,12,13}; //Defining column pins of keypad connected to Arduino
int i,j; // Two counter variables to count inside for loop
int col_scan; // Variable to hold value of scanned columns
void setup()
{
Serial.begin(9600);
for(i=0;i<=3;i++)
{
pinMode(row[i],OUTPUT); // viene assegnato un valore alle righe e alle colonne tramite
pinMode(col[i],INPUT); // l'incremento della variabile "i" da 0 a 3
digitalWrite(col[i],HIGH);
} }
void loop()
{
for(i=0; i<=3; i++)
{
digitalWrite(row[0],HIGH); // tutte le righe vengono impostate alte, e solo quella
digitalWrite(row[1],HIGH); // con il valore "i" viene assegnata come bassa
digitalWrite(row[2],HIGH);
digitalWrite(row[3],HIGH);
digitalWrite(row[i],LOW);
for(j=0; j<=3; j++)
{
col_scan=digitalRead(col[j]); // le colonne vengono scannerizzate per verificare
// che siano a valore basso
if(col_scan==LOW)
{
keypress(i,j);
delay(300);
} }
} }

```

```
}}
void keypress(int i, int j)
{
if(i==0&&j==0)           // i valori delle righe e delle colonne vengono
Serial.println("1");     // verificati per determinare il valore da mostrare sul display.
if(i==0&&j==1)
Serial.println("2");
if(i==0&&j==2)
Serial.println("3");
if(i==0&&j==3)
Serial.println("A");
if(i==1&&j==0)
Serial.println("4");
if(i==1&&j==1)
Serial.println("5");
if(i==1&&j==2)
Serial.println("6");
if(i==1&&j==3)
Serial.println("B");
if(i==2&&j==0)
Serial.println("7");
if(i==2&&j==1)
Serial.println("8");
if(i==2&&j==2)
Serial.println("9");
if(i==2&&j==3)
Serial.println("C");
if(i==3&&j==0)
Serial.println("*");
if(i==3&&j==1)
Serial.println("0");
if(i==3&&j==2)
Serial.println("#");
if(i==3&&j==3)
Serial.println("D");
}
```