16Key4x4MembraneSwitchKeyBoard v1.2 >> Hex KeyPad

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

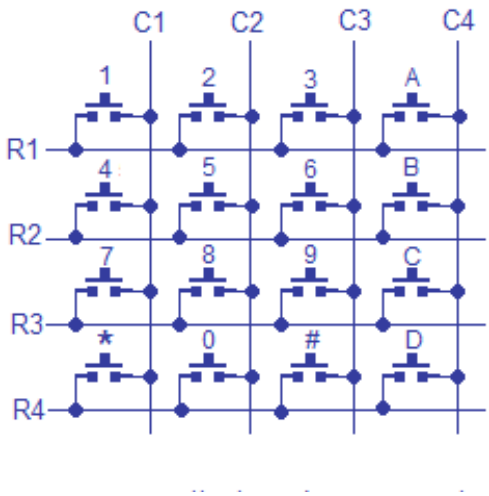
Circuit Rating: 35V (DC), 100mA, 1W
Contact resistance: 10 ~ 500
(Varies according to the lead lengths and different from those of the material used)
Insulation resistance: 100M 100V
Dielectric Strength: 250VRms (50 ~ 60Hz 1min)
Electric shock jitter: <5ms
Life span: tactile type: 1 million times

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)

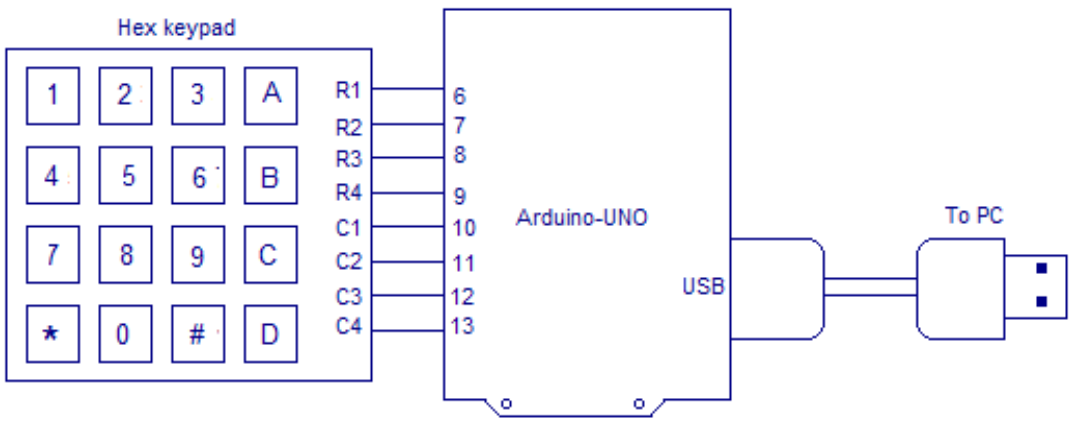
Hex keypad.

Hex key pad is simply an arrangement 0f 16 push button switches in a 4X4 matrix form. Typically a hex keypad will have keys for number 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and letters A, B, C, D, *, #. The hex keypad will have 8 connection wires namely R1, R2, R3, R4 and C1, C2, C3, C4 representing the rows and columns respectively. The schematic diagram and photo of a typical hex keypad is shown in the figure below.



The program identifies the pressed key by a method called column scanning. In this method a particular row is kept low and other rows are held high. The the logic status of each column line is scanned. If a particular column is found low, then that means the key that comes in between that column and row is short(pressed). Then the program registers that key being pressed. Then the same procedure is applied for the subsequent rows and the entire process is repeated. For example if row 1 is kept low and column 1 is found low during scanning, then that means key"1" is pressed. Full circuit diagram of interfacing hex keypad is shown below.

Circuit diagram.



Rows R1, R2, R3 and R4 are interfaced to digital pins 6, 7, 8 and 9 pins of the arduino respectively. Columns C1, C2, C3 and C4 are interfaced to the digital pins 10, 11, 12, 13 of the arduino. The arduino is connected to PC through the USB port. The circuit is powered from the USB itself and no external power supply is needed. The full program for interfacing hex keypad to arduino is given below.

Program (Version 1)

```
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);
  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");
   delay(200);}
  else
  {
   if(colm2==LOW)
   {Serial.println("2");
    delay(200);}
   else
   {
   if(colm3==LOW)
   {Serial.println("3");
     delay(200);}
   else
   {
   if(colm4==LOW)
   {Serial.println("A");
      delay(200);}
   }}}

  digitalWrite(r1,HIGH);
  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");
    delay(200);}
  else
  {
   if(colm2==LOW)
   {Serial.println("5");
    delay(200);}
```

```
  else
  {
  if(colm3==LOW)
  {Serial.println("6");
    delay(200);}
  else
  {
  if(colm4==LOW)
  {Serial.println("B");
     delay(200);}
  }}}

 digitalWrite(r1,HIGH);
 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");
   delay(200);}
 else
 {
  if(colm2==LOW)
  {Serial.println("8");
     delay(200);}
  else
  {
  if(colm3==LOW)
  {Serial.println("9");
      delay(200);}
  else
  {
  if(colm4==LOW)
  {Serial.println("C");
      delay(200);}
  }}}
 digitalWrite(r1,HIGH);
 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("*");
    delay(200);}
 else
 {
  if(colm2==LOW)
  {Serial.println("0");
      delay(200);}
  else
  {
  if(colm3==LOW)
  {Serial.println("#");
    delay(200);}
  else
  {
  if(colm4==LOW)
  {Serial.println("D");
     delay(200);}

  }}}

}
```

About the program.

Code Serial.begin(9600); sets the baud rate for serial communication at 9600. Baud rate is the number of signal changes happening in a second in a digitally modulated transmission line. Firstly row1 is held high and other rows are held low using digitalWrite command. Then the status of each column is read using digitalRead command. Then each column is checked for low using an if-else loop. If a particular column in found low, the key intersecting that column and row1 is assumed to be pressed and the name of that key is displayed on the serial monitor window using Serial.print command. A delay of 200ms is given between each condition checking loops in order to avoid multiple key registering during a single key press. If this delay is increased further more the response of the keypad will be reduced. After some trial and error, I found the optimum value for my scheme to be 200mS. Then row 2 is made low and other rows are kept high. The column scanning using if loop-else loop is done again. Then the same thing is done for row 3 and then for row 4. The the entire cycle is repeated over time. The result will be the name of the pressed key displayed on the serial monitor any time.

Simpler Version of Above Program (Version 2)

We can easily simplify the program written above with smart usage of Arrays and For Loops.
We are adding a simplified version of the above program (which is half the number of lines of code above). You may compare both codes for understanding the topic.

```
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);
pinMode(col[i],INPUT);
digitalWrite(col[i],HIGH);
} }
void loop()
{
for(i=0; i<=3; i++)
{
digitalWrite(row[0],HIGH);
digitalWrite(row[1],HIGH);
digitalWrite(row[2],HIGH);
digitalWrite(row[3],HIGH);
digitalWrite(row[i],LOW);
for(j=0; j<=3; j++)
{
col_scan=digitalRead(col[j]);
if(col_scan==LOW)
{
keypress(i,j);
delay(300);
}}
}}
void keypress(int i, int j)
{
if(i==0&&j==0)
Serial.println("1");
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");
}
```