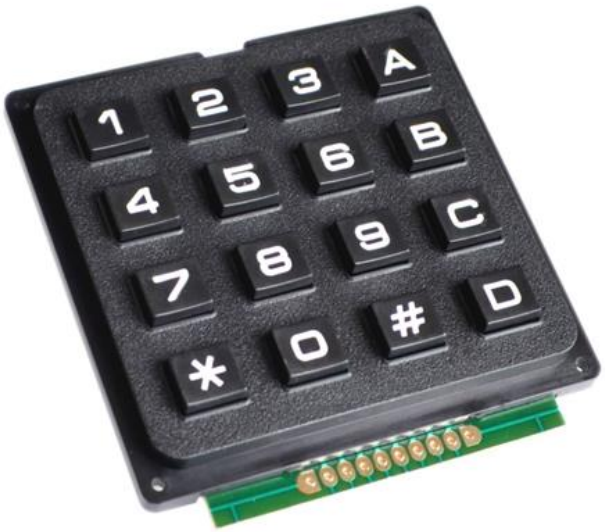
4x4 Matrix ABS Plastic Keypad Module

This 16-button kepyad provides a useful human interface component for micro-controller projects. This keypads are suited to all indoor applications including data-entry systems, remote controls, telephone, point of sales terminals or alarm systems.



Brief Data:

- Maximum Rating: 24VDC/30mA.
- Keypad Layout: 4x4 (Columns x Rows).
- Number of Keys: 16.
- Switch Type: Conductive Rubber.
- Non-Illuminated.
- Key Type: Polymer.
- Output Type: Matrix.
- Color: White.
- Key Color: Black.
- Mounting Type: Panel Mount, Rear.
- Termination Style: Card Edge/Solder Pad.
- Weight: 24g.

Construction:

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a micro-controller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.

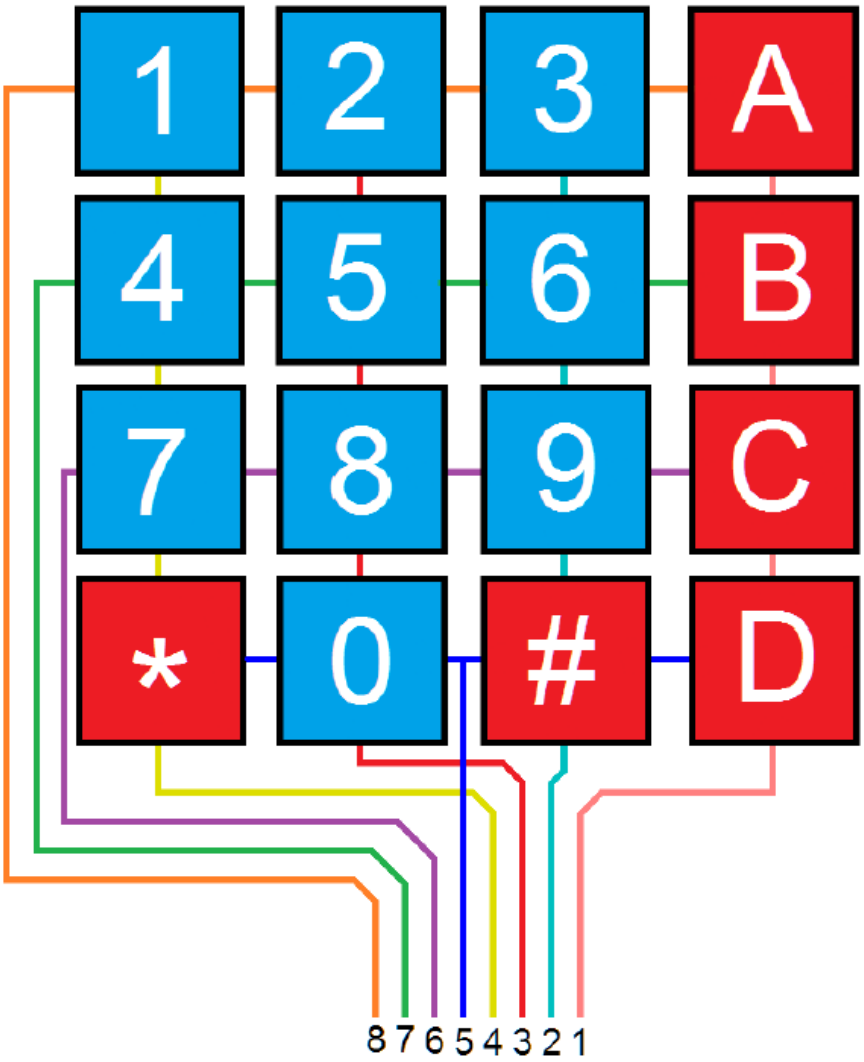


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed. For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1high. This means that a contact has been made between column 4 and row 1, so button 'A' has been pressed.

How to Connect and Read a Keypad with an Arduino

In this project, we will go over how to integrate a keyboard with an Arduino board so that the Arduino can read the keys being pressed by a user. Keypads are used in all types of devices, including cell phones, fax machines, microwaves, ovens, door locks, etc. They're practically everywhere. Tons of electronic devices use them for user input.

So knowing how to connect a keypad to a microcontroller such as an Arduino is very valuable for building many different types of commercial products. At the end when all is connected properly and programmed, when a key is pressed, it show up at the Serial Monitor on your computer. Whenever you press a key, it shows up on the Serial Monitor. Later, in another project, we will connect the keypad circuit, so that it will get displayed on an LCD. But for now, for simplicity purposes, we start at simply showing the key pressed on the computer.

For this project, the type of keypad we will use is a matrix keypad. This is a keypad that follows an encoding scheme that allows it to have much less output pins than there are keys. For example, the matrix keypad we are using has 16 keys (0-9, A-D, *, #), yet only 8 output pins. With a linear keypad, there would have to be 17 output pins (one for each key and a ground pin) in order to work. The matrix encoding scheme allows for less output pins and thus much less connections that have to make for the keypad to work. In this way, they are more efficient than linear keypads, being that they have less wiring.

Components Needed:

- Arduino Uno
- 4x4 Matrix Keypad
- 8 male to male pin header



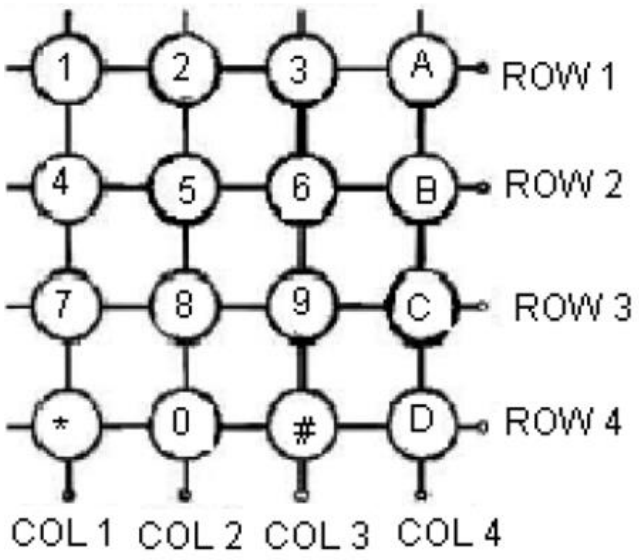
One of the most mysterious things about these keypads is that they usually come with no documentation, so a user is left to figure out the pin configuration. However, we at this site, have figured it out. With the keypad facing up so that the keys are up and facing you, from left to right, the 1st 4 pins are the row pins and the last 4 pins are the column pins.

When connecting the pins to the Arduino board, we connect them to the digital output pins, D9-D2. We connect the first pin of the keypad to D9, the second pin to D8, the third pin to D7, the fourth pin to D6, the fifth pin to D5, the sixth pin to D4, the seventh pin to D3, and the eighth pin to D2.

These are the connections in a table:

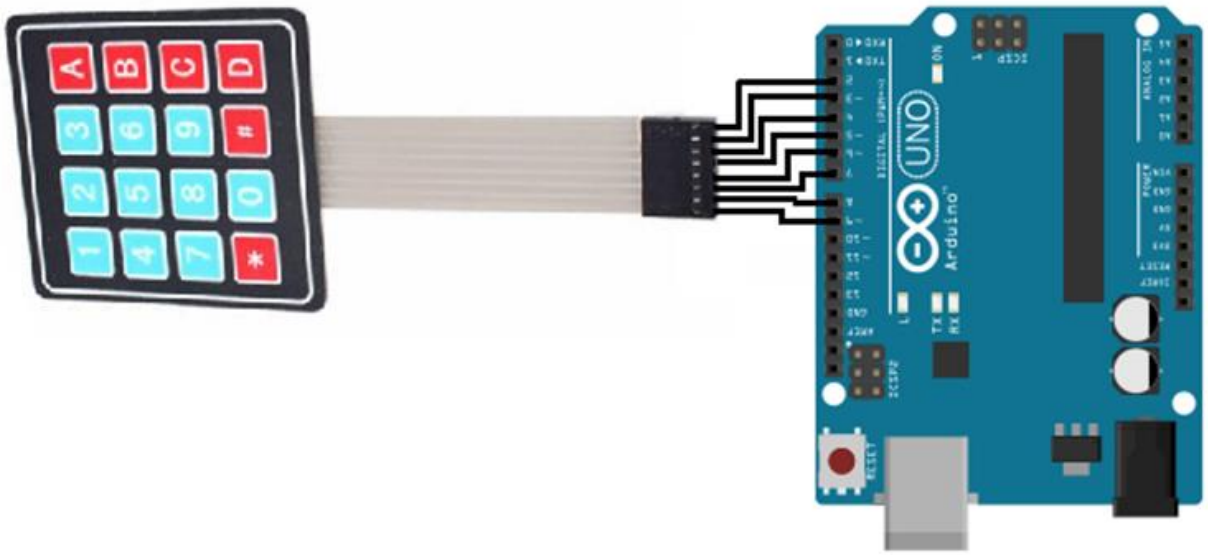
Keypad Pin	Connects to Arduino Pin
1	D9
2	D8
3	D7
4	D6
5	D5
6	D4
7	D3
8	D2

Circuit Diagram:



Output Arrangement	
Output Pin Number	**Symbol**
1	COL 1
2	COL 2
3	COL 3
4	COL 4
5	ROW 1
6	ROW 2
7	ROW 3
8	ROW 4

Keypad with Arduino Circuit Schematic:



Here you visually see all the connections that were written above.

Now that we have the physical setup, all we need now is the code.

Before you can run this, you have to import the Keypad library and then once you import it, then you can enter it into your program. Once it's entered into your program, you should see the line #include <Keypad.h>. If you do not see this, that means the Keypad library has not been successfully put into your code and it won't work.

You can download the Keypad library here:

Keypad Library.

http://playground.arduino.cc/code/keypad

Unzip the keypad.zip file. Put the Keypad folder in "*arduino*\libraries\".

When you download, change the name to folder to something other than Keypad. If the folder and the file you are importing have the same name, it won't work.

Arduino sketch listing:

```
==========================================================================

/*4x4 Matrix Keypad connected to Arduino
www.handsontec.com
This code prints the key pressed on the keypad to the serial port
*/

#include <Keypad.h>

const byte numRows= 4;    //number of rows on the keypad
const byte numCols= 4;    //number of columns on the keypad

//keymap defines the key pressed according to the row and columns just as appears on the keypad
char keymap[numRows][numCols]=
{
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

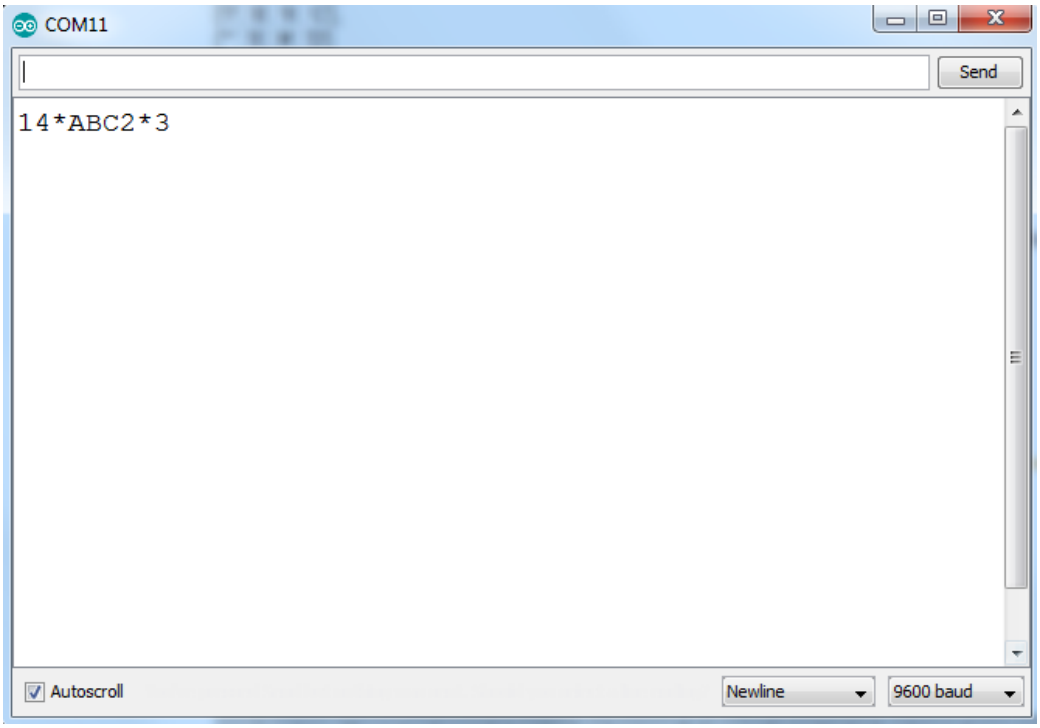
//Code that shows the keypad connections to the arduino terminals
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);

void setup()
{
  Serial.begin(9600);
}

//If key is pressed, this key is stored in 'keypressed' variable
//If key is not equal to 'NO_KEY', then this key is printed out
//if count=17, then count is reset back to 0 (this means no key is pressed during the whole keypad scan process
void loop()
{
  char keypressed = myKeypad.getKey();
  if (keypressed != NO_KEY)
  {
    Serial.print(keypressed);
  }
}
```

With this code, once we press a key on the keypad, it should show up on the serial monitor of the Arduino software once the code is compiled and uploaded to the Arduino board.


COM11
Send
14*ABC2*3
Autoscroll
Newline
9600 baud