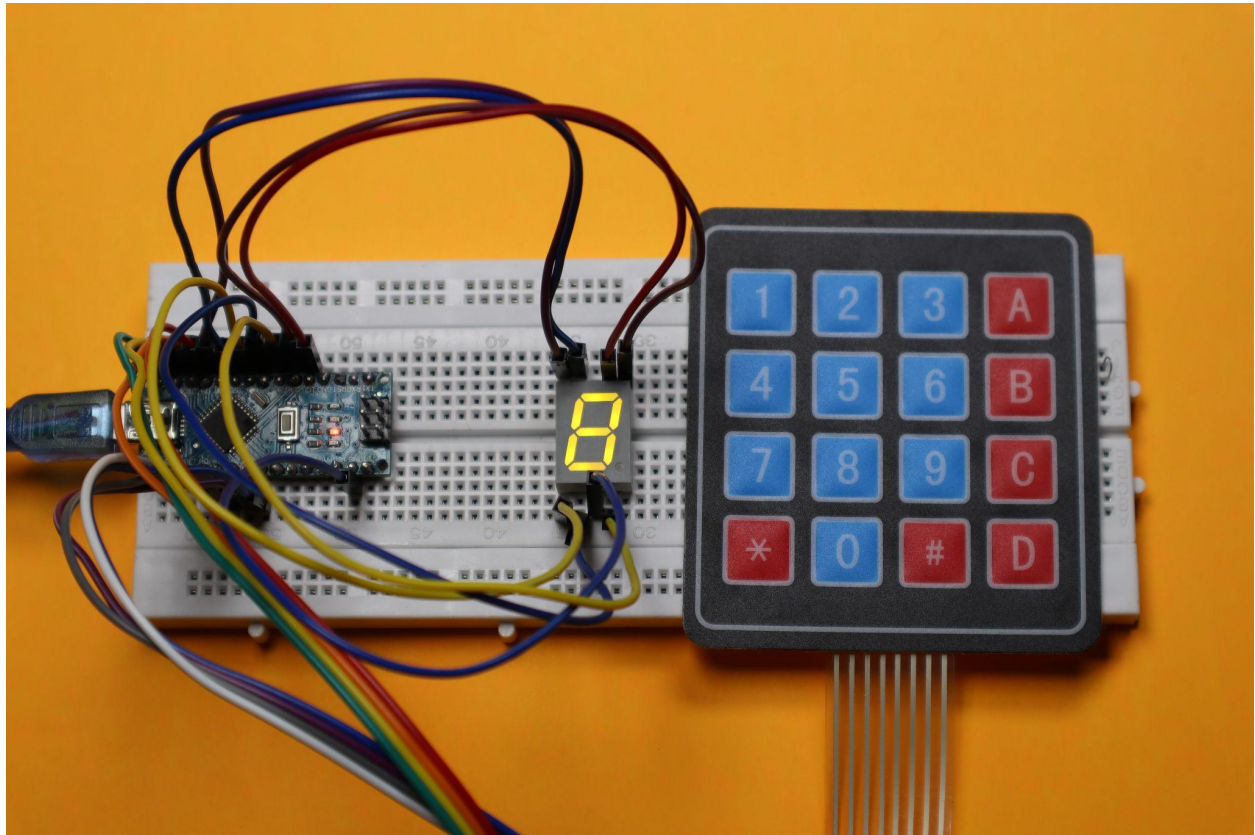


Control Seven segment Display using 4X4 Keypad



Material Required 🛒 :

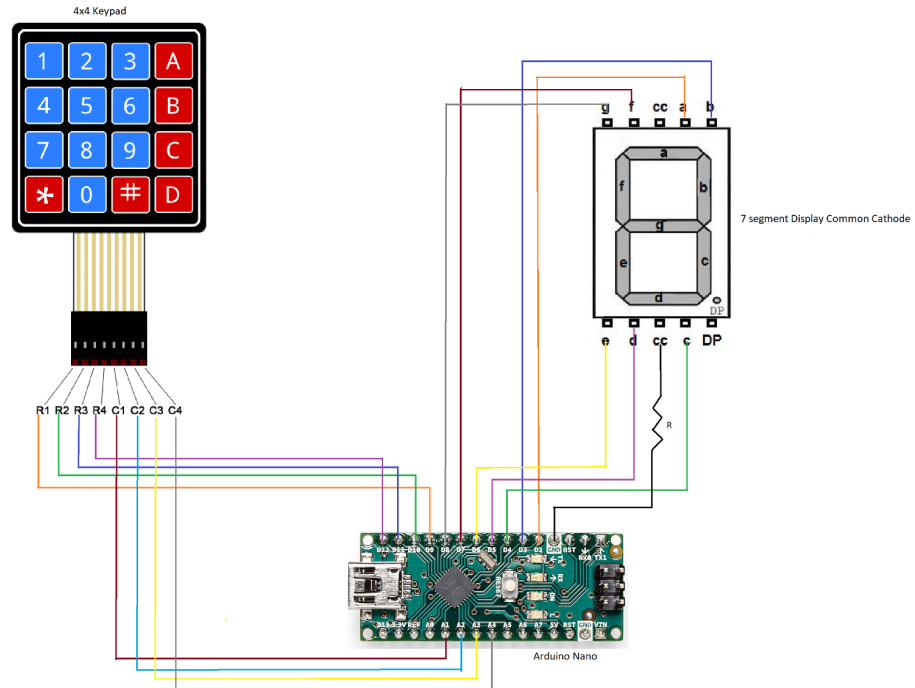
S No.	Components	Buy Components
1	4X4 Keypad	https://amzn.to/3B81YXN
2	Arduino Nano	https://amzn.to/3ef8pA1
3	Arduino Nano Cable	https://amzn.to/3ef8pA1
4	Connecting wires	https://amzn.to/3cl97EY
5	Breadboard Large	https://amzn.to/3QdsROy
6	7 segment Display (C-C)	https://amzn.to/3CSt1aZ

Download this library :

```
#include <Keypad.h>
```

Circuit Diagram ⚡:

Control 7 segment display using keypad



Code 🖥️:

```
#include <Keypad.h>

const int a = 2;
const int b = 3;
const int c = 4;
const int d = 5;
const int e = 6;
const int f = 7;
const int g = 8;

//Keypad connections
/* R0 - 9
 * R1 - 10
```

```
* R2 - 11
* R3 - 12
*
* C0 - A1
* C1 - A2
* C2 - A3
* C3 - A4
*/

const byte ROWS = 4; /* four rows */
const byte COLS = 4; /* four columns */
/* define the symbols on the buttons of the keypads */
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','c'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9, 10, 11, 12}; /* connect to the row pinouts of the
keypad */
byte colPins[COLS] = {A1,A2,A3,A4}; /* connect to the column pinouts of the
keypad */

/* initialize an instance of class NewKeypad */
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);

void setup(){
  Serial.begin(9600);
  for(int i =2;i<=8;i++)
  {
    pinMode(i,OUTPUT);
  }
}

void loop(){
  char customKey = customKeypad.getKey();

  if (customKey)
  {
    Serial.println(customKey);
  }
}
```

```
if(customKey == '1')
{
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(a,0);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,0);
    digitalWrite(g,0);
}

if(customKey == '2')
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,0);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,0);
    digitalWrite(g,1);
}

if(customKey == '3')
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,0);
    digitalWrite(f,0);
    digitalWrite(g,1);
}

if(customKey == '4')
{
    digitalWrite(a,0);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,1);
    digitalWrite(g,1);
}
```

```
}

    if(customKey == '5')
{
    digitalWrite(a,1);
    digitalWrite(b,0);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,0);
    digitalWrite(f,1);
    digitalWrite(g,1);
}

    if(customKey == '6')
{
    digitalWrite(a,1);
    digitalWrite(b,0);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,1);
    digitalWrite(g,1);
}

    if(customKey == '7')
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,0);
    digitalWrite(g,0);
}

    if(customKey == '8')
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,1);
```

```
    digitalWrite(g,1);  
}  
  
if(customKey == '9')  
{  
    digitalWrite(a,1);  
    digitalWrite(b,1);  
    digitalWrite(c,1);  
    digitalWrite(d,1);  
    digitalWrite(e,0);  
    digitalWrite(f,1);  
    digitalWrite(g,1);  
}  
  
if(customKey == '0')  
{  
    digitalWrite(a,1);  
    digitalWrite(b,1);  
    digitalWrite(c,1);  
    digitalWrite(d,1);  
    digitalWrite(e,1);  
    digitalWrite(f,1);  
    digitalWrite(g,0);  
}  
  
}
```