

4×4 Keypad interfacing with Arduino UNO

Code:

```
#include <Keypad.h> // declare header file for keypad

const byte rows =4; // declare number of rows as data type byte
const byte columns =4; // declare number of columns as data type byte

char hexkeypad[rows][columns] = { {'1','2','3','A'},
                                     {'4','5','6','B'},
                                     {'7','8','9','C'},
                                     {'*','0','#','D'}
                                   }; // declare a matrix for initialize keys because keypad is 2D matrix pattern

// initialize pins for row and columns connection
byte rowpins[rows] = {2,3,4,5};
byte columnpins[columns] = {6,7,8,9};

//create object for keypad class to use its functionality for detection keys when pressed

Keypad kpd = Keypad(makeKeymap(hexkeypad), rowpins, columnpins, rows, columns );

// here kpd is an object of Keypad class, now we need to pass four parameters like
parameterizedconstructor

//makeKeymap() is a built in method to define 4*4 matrix key pattern

// rowpins is the variable used for ROW pin initialize

// columnpins is the variable used for COLOUMN pin initialize
```

```

//rows is the variable to define row numbers

//columns is the variable to define column numbers


//initialize all pins as output lines

// here we use byte data type because in built in function for keypad

//we use byte data type and this function return type is byte

byte i =0;


void setup()
{
    Serial.begin(9600);// it's a command and 9600 is baud rate
    for(i=0; i<8; i++) // this for loop is used to make all rows
    {
        // and columns lines as outputline
        pinMode(i, OUTPUT);
    }

}


void loop()
{
    char keypressed =kpd.getKey(); // getKey() is a built in function to assign a
    //pressed key to the variable keypressed

    // this built in function is called by object kpd

    if(keypressed)
    {

```

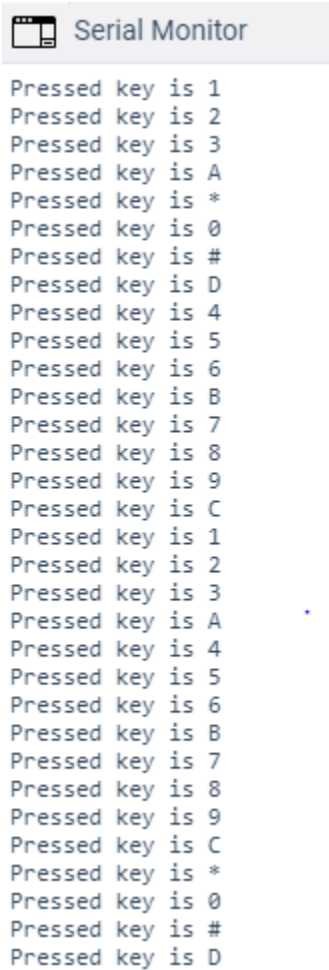
```
Serial.print("Pressed key is "); // print message

Serial.println(keypressed); // print pressed key as character

}

}
```

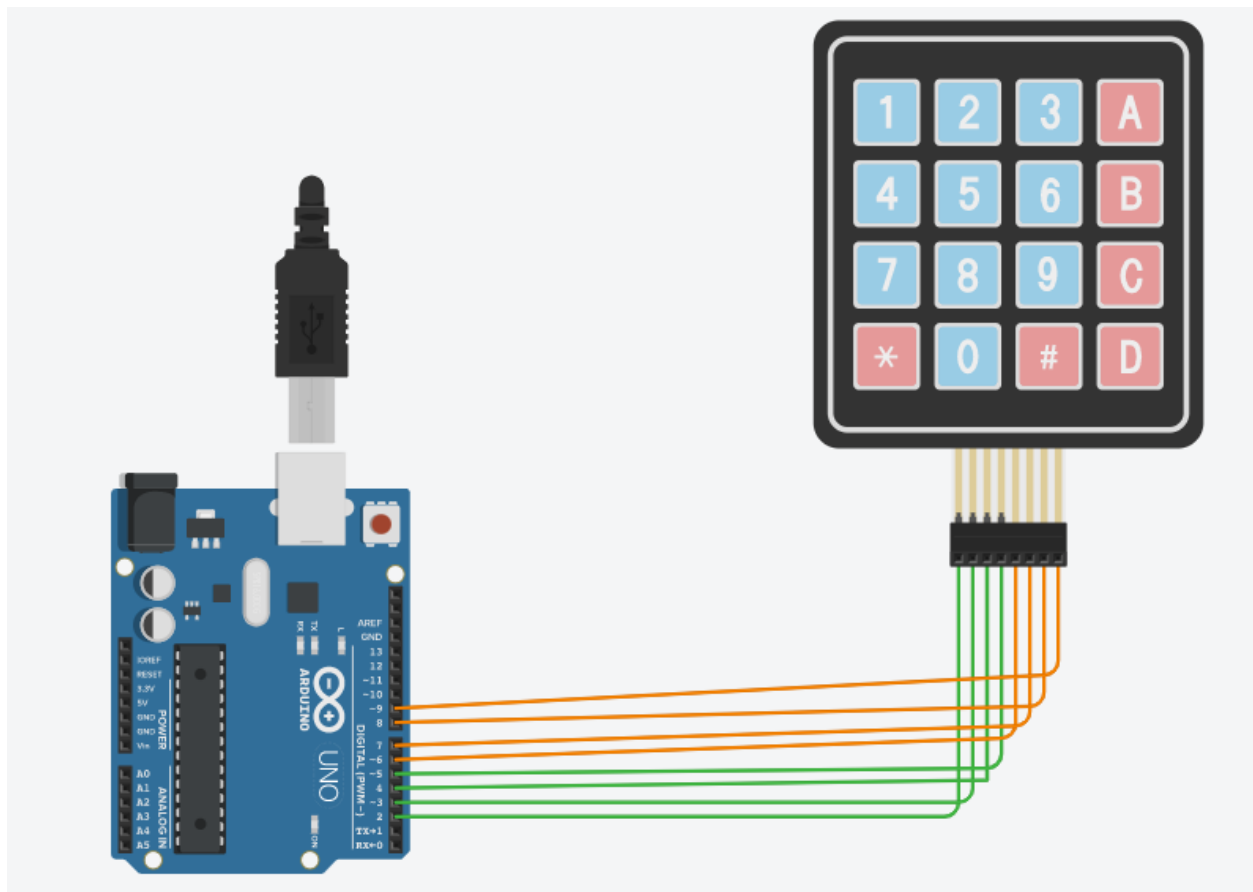
Output:



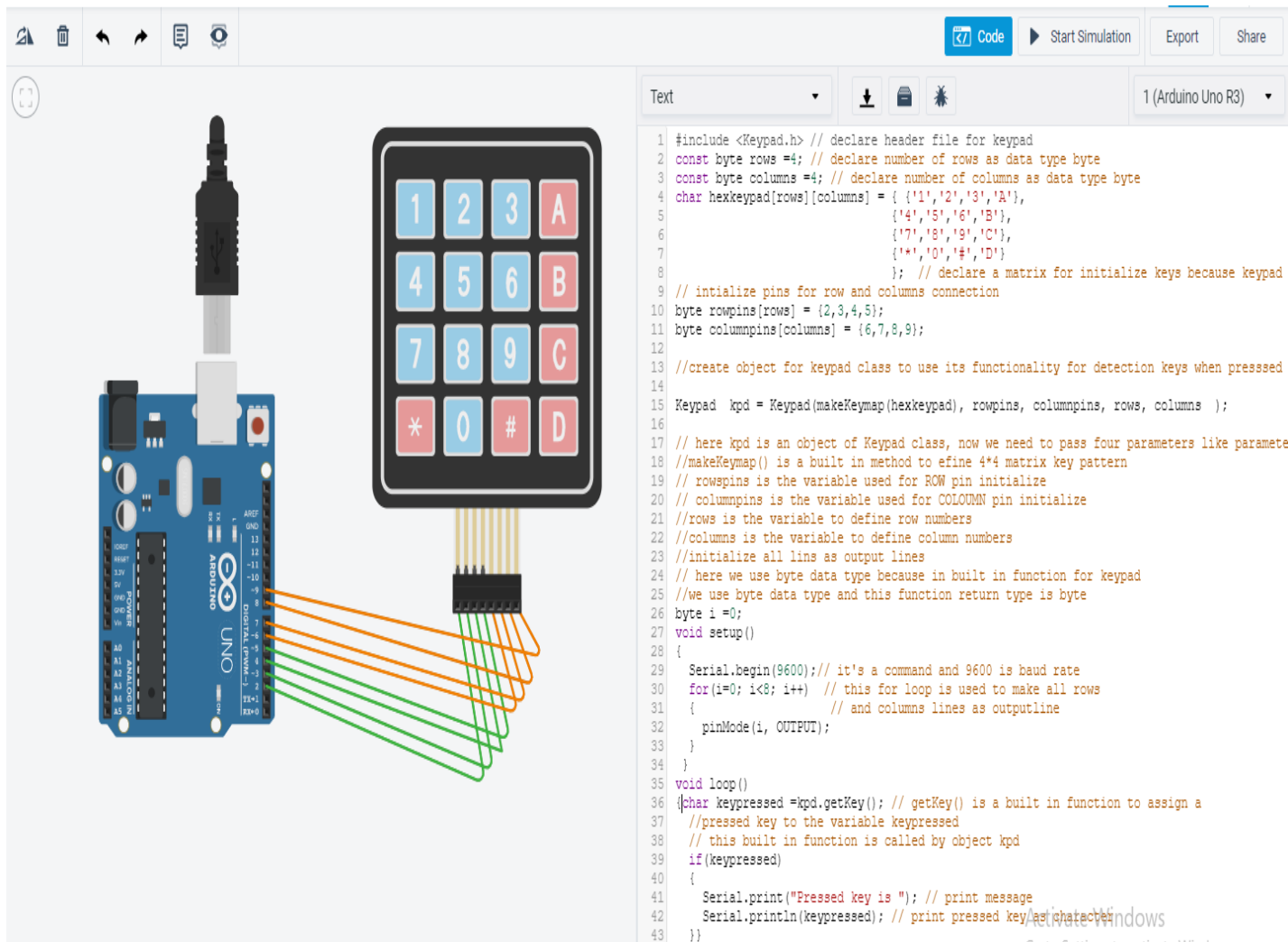
The screenshot shows the Serial Monitor window with the title "Serial Monitor". The output text is as follows:

```
Pressed key is 1
Pressed key is 2
Pressed key is 3
Pressed key is A
Pressed key is *
Pressed key is 0
Pressed key is #
Pressed key is D
Pressed key is 4
Pressed key is 5
Pressed key is 6
Pressed key is B
Pressed key is 7
Pressed key is 8
Pressed key is 9
Pressed key is C
Pressed key is 1
Pressed key is 2
Pressed key is 3
Pressed key is A
Pressed key is 4
Pressed key is 5
Pressed key is 6
Pressed key is B
Pressed key is 7
Pressed key is 8
Pressed key is 9
Pressed key is C
Pressed key is *
Pressed key is 0
Pressed key is #
Pressed key is D
```

Design



Design and Code



The image displays the Arduino IDE interface. On the left, a circuit diagram shows an Arduino Uno R3 board connected to a 4x4 keypad. The keypad has 16 buttons arranged in a 4x4 grid. The top row contains buttons labeled 1, 2, 3, and A. The second row contains 4, 5, 6, and B. The third row contains 7, 8, 9, and C. The bottom row contains *, 0, #, and D. The keypad is connected to the Arduino board via a 16-pin ribbon cable. The connections are as follows: the first four pins of the ribbon cable (pins 1-4) are connected to the Arduino's digital pins 2, 3, 4, and 5 (rows); the next four pins (pins 5-8) are connected to the Arduino's digital pins 6, 7, 8, and 9 (columns); and the remaining eight pins (pins 9-16) are connected to the Arduino's ground pins (pins 1, 2, 3, 4, 5, 6, 7, and 8).

On the right, the C++ code for the keypad is shown. The code includes the `<Keypad.h>` header file, declares the number of rows and columns, and defines a 4x4 matrix of key values. The code then initializes the pins for the rows and columns, creates a `Keypad` object, and sets up the serial communication. The `loop()` function reads the key pressed and prints the message "Pressed key is " followed by the key value.

```
1 #include <Keypad.h> // declare header file for keypad
2 const byte rows =4; // declare number of rows as data type byte
3 const byte columns =4; // declare number of columns as data type byte
4 char hexkeypad[rows][columns] = { {'1','2','3','A'},
5                                     {'4','5','6','B'},
6                                     {'7','8','9','C'},
7                                     {'*','0','#','D'}
8                                     }; // declare a matrix for initialize keys because keypad
9 // initialize pins for row and columns connection
10 byte rowpins[rows] = {2,3,4,5};
11 byte columnpins[columns] = {6,7,8,9};
12
13 //create object for keypad class to use its functionality for detection keys when pressed
14
15 Keypad kpd = Keypad(makeKeymap(hexkeypad), rowpins, columnpins, rows, columns );
16
17 // here kpd is an object of Keypad class, now we need to pass four parameters like parameter
18 //makeKeymap() is a built in method to define 4*4 matrix key pattern
19 // rowpins is the variable used for ROW pin initialize
20 // columnpins is the variable used for COLUMN pin initialize
21 //rows is the variable to define row numbers
22 //columns is the variable to define column numbers
23 //initialize all pins as output lines
24 // here we use byte data type because in built in function for keypad
25 //we use byte data type and this function return type is byte
26 byte i =0;
27 void setup()
28 {
29     Serial.begin(9600); // it's a command and 9600 is baud rate
30     for(i=0; i<8; i++) // this for loop is used to make all rows
31     { // and columns lines as outputline
32         pinMode(i, OUTPUT);
33     }
34 }
35 void loop()
36 {char keypressed =kpd.getKey(); // getKey() is a built in function to assign a
37 //pressed key to the variable keypressed
38 // this built in function is called by object kpd
39 if(keypressed)
40 {
41     Serial.print("Pressed key is "); // print message
42     Serial.println(keypressed); // print pressed key as character
43 }
```