Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 5: Matrix keypad interface with Arduino

A 4x4 matrix keypad with an Arduino. We'll use the keypad to detect which key is pressed.

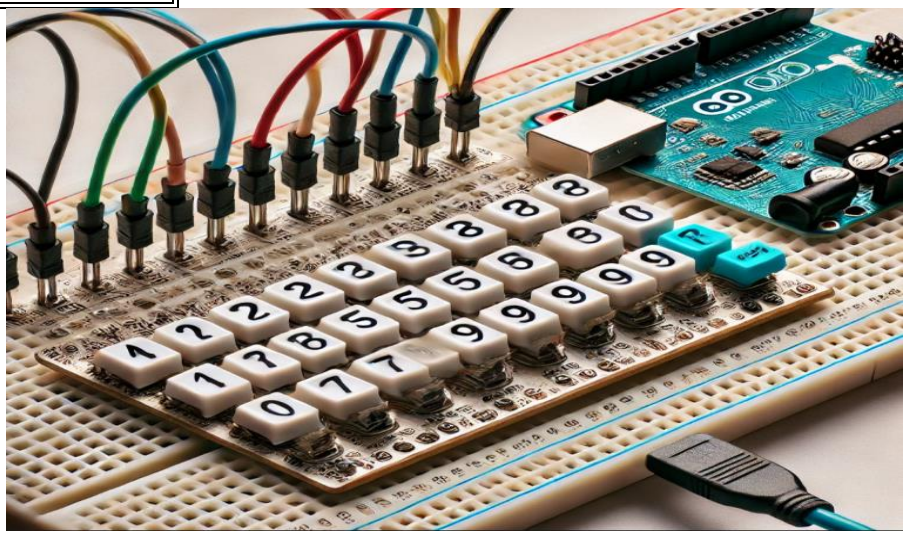
Components Required:

1. Arduino board (e.g., Uno)
2. 4x4 Matrix Keypad
3. Jumper wires

Keypad Layout:

A typical 4x4 matrix keypad has 4 rows and 4 columns, making 16 keys. Here's the general layout:

Key	Pin
1 2 3 A	R1, C1, C2, C3, C4
4 5 6 B	R2, C1, C2, C3, C4
7 8 9 C	R3, C1, C2, C3, C4
* 0 # D	R4, C1, C2, C3, C4



matrix keypad with Arduino

Email: zahraa.hashim@uomus.edu.iq

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 5: Matrix keypad interface with Arduino

Wiring:

- Connect the **4 row pins (R1-R4)** and **4 column pins (C1-C4)** of the keypad to **8 digital pins** on the Arduino.
- Example connections:
  - R1 → Pin 9
  - R2 → Pin 8
  - R3 → Pin 7
  - R4 → Pin 6
  - C1 → Pin 5
  - C2 → Pin 4
  - C3 → Pin 3
  - C4 → Pin 2

```
#include <Keypad.h>

// Define the keypad size

const byte ROWS = 4; // 4 rows

const byte COLS = 4; // 4 columns

// Define the key map

char keys[ROWS][COLS] = {

 {'1', '2', '3', 'A'},

 {'4', '5', '6', 'B'},

 {'7', '8', '9', 'C'},

 {'*', '0', '#', 'D'}

};

// Connect keypad rows and columns to Arduino pins

byte rowPins[ROWS] = {9, 8, 7, 6}; // Connect to R1, R2, R3, R4

byte colPins[COLS] = {5, 4, 3, 2}; // Connect to C1, C2, C3, C4
```

Email: zahraa.hashim@uomus.edu.iq

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 5: Matrix keypad interface with Arduino

```
// Create the Keypad object
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
void setup() {
  Serial.begin(9600);
}
void loop() {
  // Get the key pressed
  char key = keypad.getKey();
  // If a key is pressed, print it to the Serial Monitor
  if (key) {
    Serial.print("Key pressed: ");
    Serial.println(key);
  }
}
```

Explanation:

1. **Keypad Library**:
   - The Keypad.h library simplifies working with matrix keypads.
   - Install the library via the Arduino IDE's Library Manager.
2. **Keymap**:
   - Define the keypad layout in the keys array to match your physical keypad.
3. **Row and Column Pins**:

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 5: Matrix keypad interface with Arduino

- Match the pin numbers to how you've wired the keypad to the Arduino.

4. **Detect Key Press**:
   - The getKey() method checks if any key is pressed and returns its value.

Testing:

1. Upload the code to your Arduino.
2. Open the Serial Monitor.
3. Press a key on the keypad, and its value will be displayed on the Serial Monitor.

You can use this setup as the foundation for more advanced projects, like a password-protected lock system!

Summary Table of Operators

Operator	Name	Description	Example
`>`	Greater Than	Checks if left operand is greater than right	`if (temperature > 30)`
`<`	Less Than	Checks if left operand is less than right	`if (temperature < 10)`
`>=`	Greater Than or Equal To	Checks if left operand is greater than or equal to right	`if (voltage >= 2.5)`
`<=`	Less Than or Equal To	Checks if left operand is less than or equal to right	`if (lightIntensity <= 50)`
`==`	Equal To	Checks if two values are equal	`if (buttonState == HIGH)`
`!=`	Not Equal To	Checks if two values are not equal	`if (sensorValue != 0)`