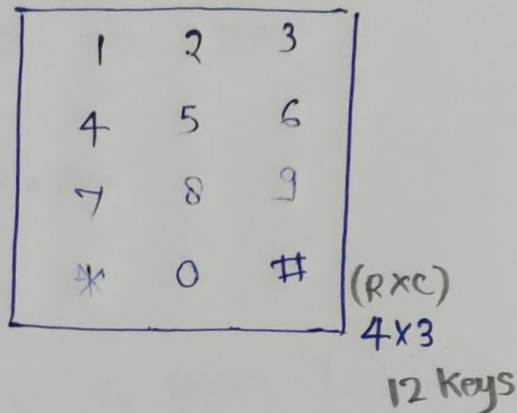
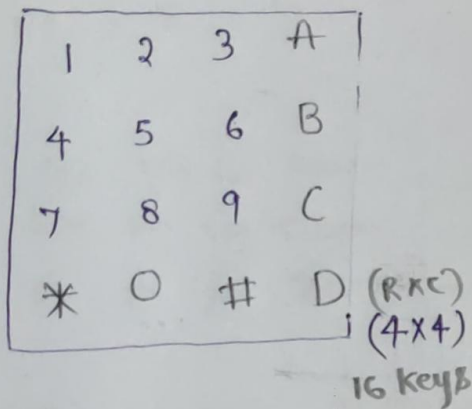


# INTERFACE KEYPAD (4x4 or 4x3 membrane)

with ARDUINO



→ Program that require user input.

- ✓ menu Selection
- ✓ Password Entry
- ✓ robot operation

Graphic Overlay (Polyster polycarbonate)

Metal Domes

Top Circuit

Spacer

Bottom Circuit

Adhesive Layer

(8) Connector

Multiplexing

reduce I/O Connection

16 individual pushbuttons

17 I/O pins one for each pushbutton + one for GND

For 4x4 - 8 pins Require

	C1	C2	C3	C4
R1 →	1	2	3	A
R2 →	4	5	6	B
R3 →	7	8	9	C
R4 →	*	0	H	D

(4x4) ⇒ 8 pins

One pin for Row } → One Pushbutton  
 One pin for Column }  
 One pin for GND } → GND.

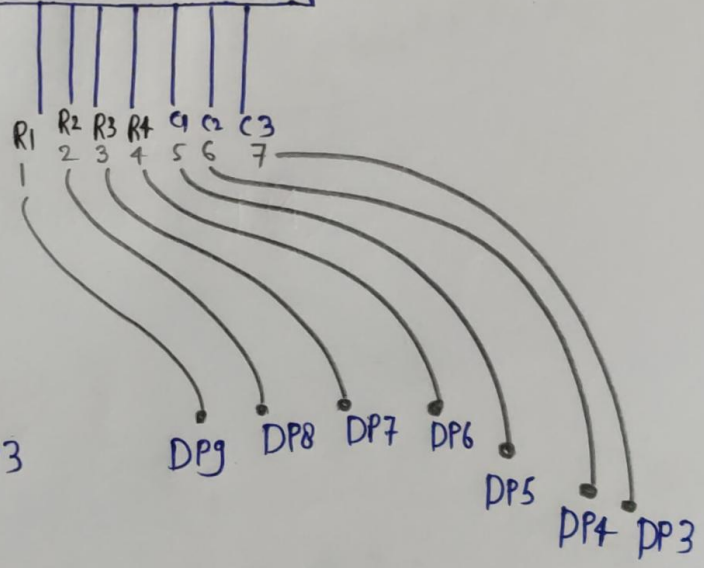
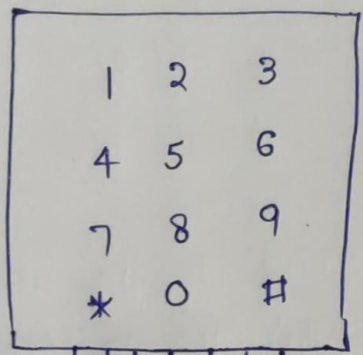
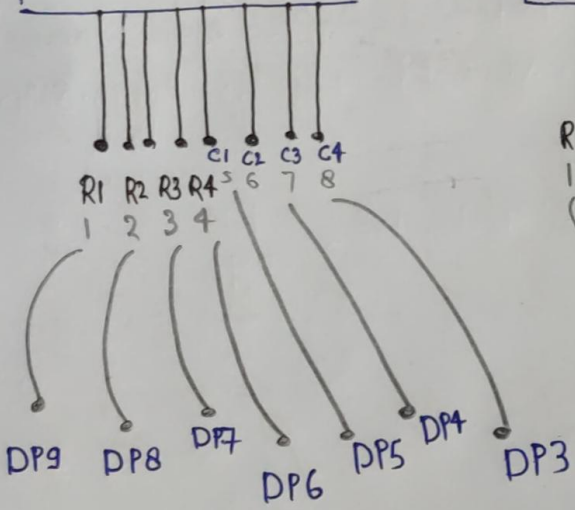
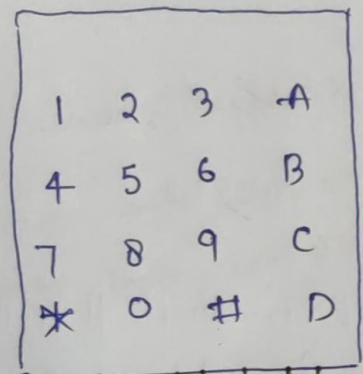
When the button is pressed { One of the ROWS }  
 { One of the Column } connected  
 allowing current flow b/w them.

Ex : C1 > activated when pressed (4)  
 R2

# 4x4 4x3 Keypad Pin Out

female Dupont Connector.

row pins are on the left (dark strip).



8 pin male to male Dupont ribbon Cable

# STALLING KEYPAD LIBRARY

pg 4

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

// mark stanley

Sketch > Include Library > Manage Libraries.

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

```
const byte ROWS = 4
```

```
const byte COLUMNS = 4
```

```
char Keys[ROWS][COLUMNS] = {
```

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

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

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

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

```
};
```

```
byte rowPins[ROWS] = { 9, 8, 7, 6 };
```

```
byte colPins[COLUMNS] = { 5, 4, 3, 2 };
```

Keypad Pin def.

Arduino Pin def.



Keypad Keypad = Keypad(makeKeymap(keys), PG-5  
rowPins, DP Row Array  
colPins, DP Column Array  
ROWS, keypad Row Size - 4  
COLS); keypad Column Size - 4

void setup()

{  
Serial.begin(9600);

}

void loop()

{  
char key = keypad.getKey();

Read the key

if (key)

{  
Serial.print("Key Pressed:");  
Serial.print(key);

}

}

Some Useful functions for <Keypad.h>

char waitForKey()

KeyState getState() — returns the current state of key.  
IDLE.  
PRESSED.  
RELEASED.  
— HOLD.

boolean KeyStateChanged()

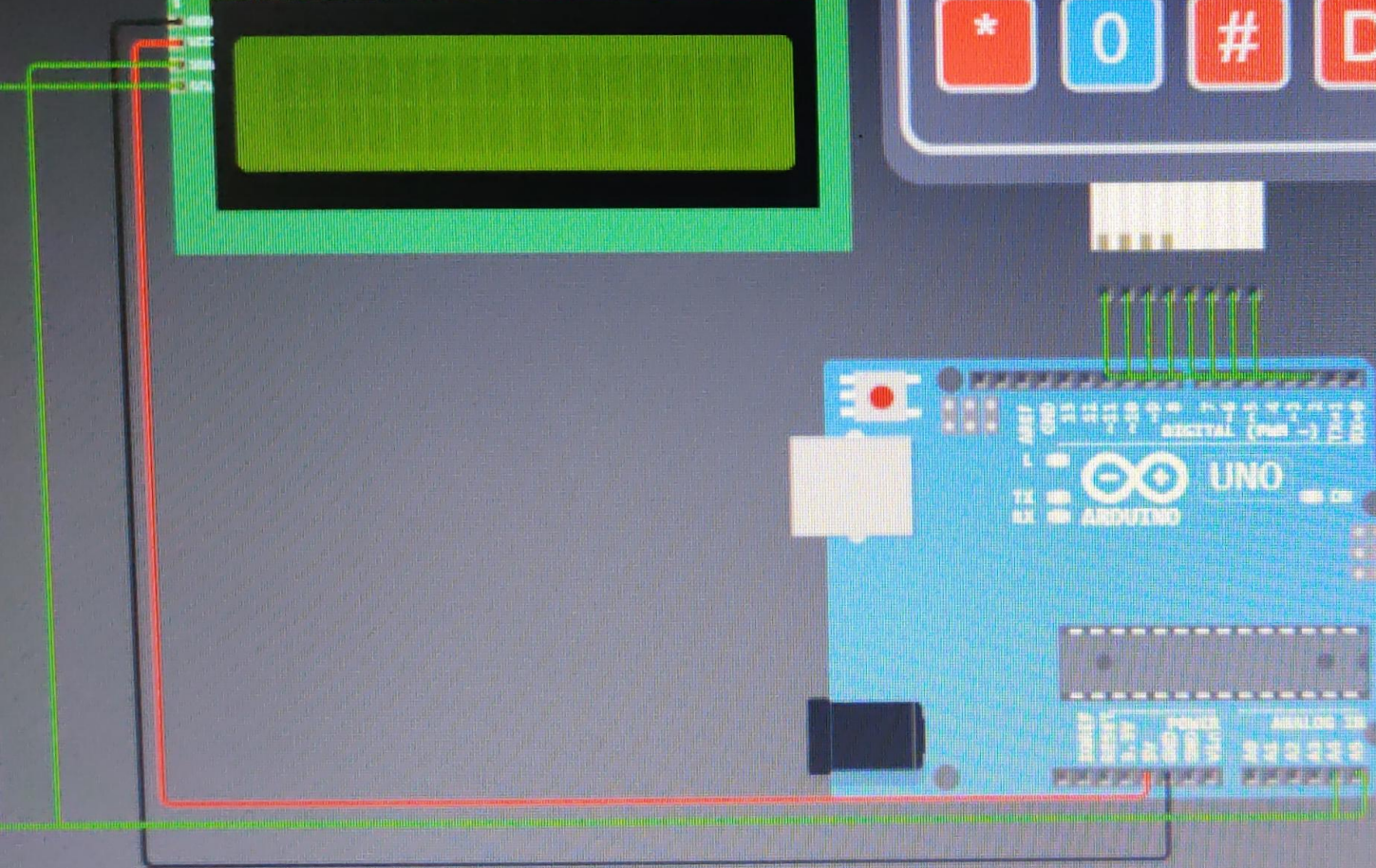
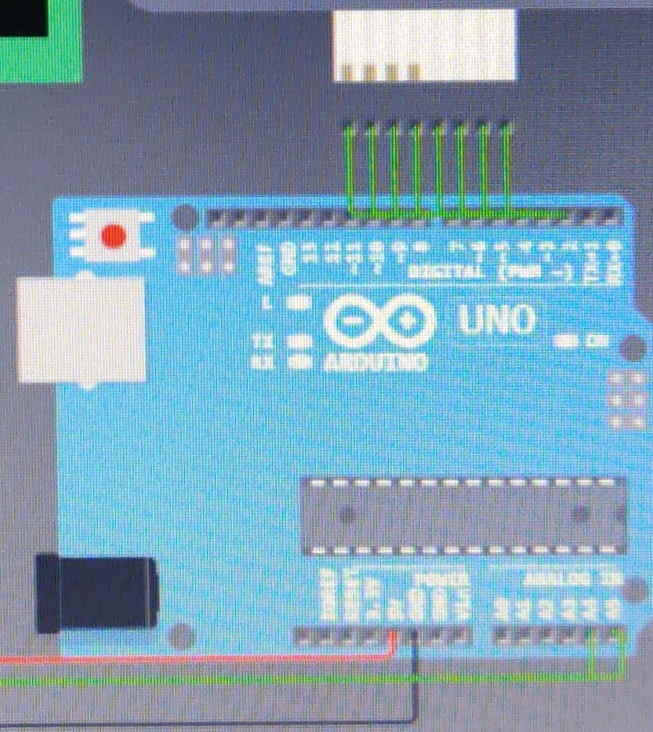
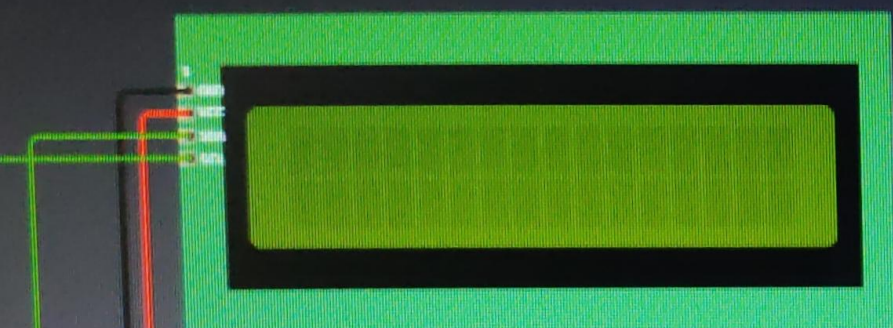
setHoldTime(unsigned int time)

setDebounceTime(unsigned int time)

addEventListener(KeypadEvent)

— triggers an event if the keypad is used.







sketch.ino

diagram.json

libraries.txt

Library Manager

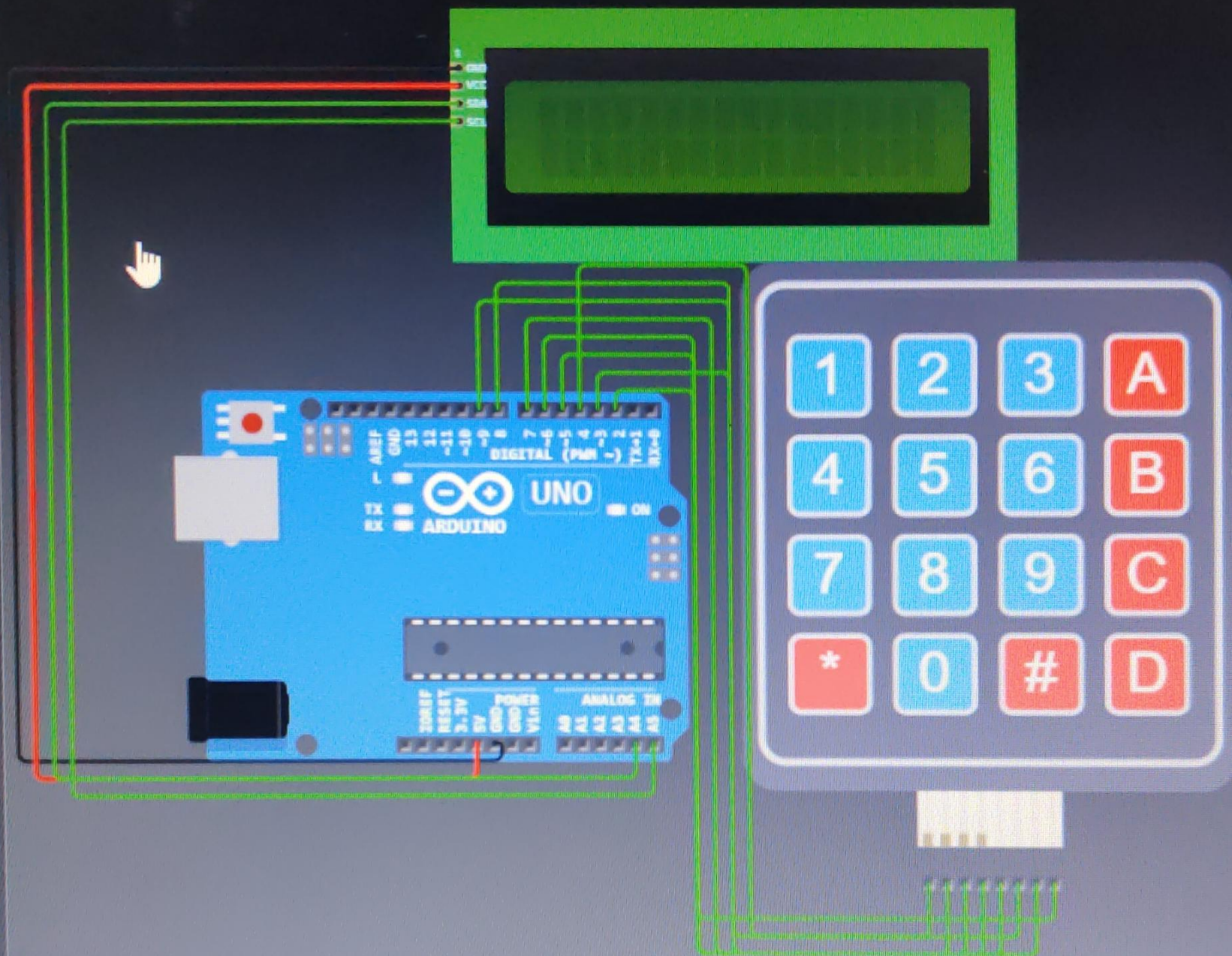
```
1  #include <Keypad.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <Wire.h>
4  const byte ROWS = 4; //four rows
5  const byte COLS = 4; //four columns
6
7  char keys[ROWS][COLS] = {
8      {'1','2','3','A'},
9      {'4','5','6','B'},
10     {'7','8','9','C'},
11     {'*','0','#','D'}
12 };
13
14 byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
15 byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
16
17 //Create an object of keypad
18 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
19 LiquidCrystal_I2C lcd(0x27,16,2);
20
21 void setup(){
22     lcd.init();
23     lcd.clear();
24     lcd.backlight();
25 }
26
27
28 void loop(){
29     lcd.setCursor(1,0);
30     char key = keypad.getKey(); // Read the key
```



```
18 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
19 LiquidCrystal_I2C lcd(0x27,16,2);
20
21 void setup(){
22     lcd.init();
23     lcd.clear();
24     lcd.backlight();
25
26 }
27
28 void loop(){
29     lcd.setCursor(1,0);
30     char key = keypad.getKey();// Read the key
31
32     // Print if key pressed
33     if (key){
34         lcd.print("Key Pressed : ");
35         lcd.println(key);
36     }
37 }
```









sketch.ino

diagram.json

libraries.txt

Library Manager

```
1 #include <LiquidCrystal_I2C.h>
2 #include <Wire.h>
3 #include <Keypad.h> //Header file for Keypad from https://github.com/Chris--A/
4
5 const byte ROWS = 4; // Four rows
6 const byte COLS = 4; // Three columns
7
8
9 char keys[ROWS][COLS] = {
10
11     {'1','2','3','A'},
12
13
14     {'4','5','6','B'},
15
16
17     {'7','8','9','C'},
18
19
20     {'*','0','#','D'}
21
22 };
23
24
25
26
27 byte rowPins[ROWS] = { 9,8,7,6};
28 // Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
29 byte colPins[COLS] = { 5,4,3,2 };
30 // Connect keypad COL0, COL1 and COL2 to these Arduino pins.
```