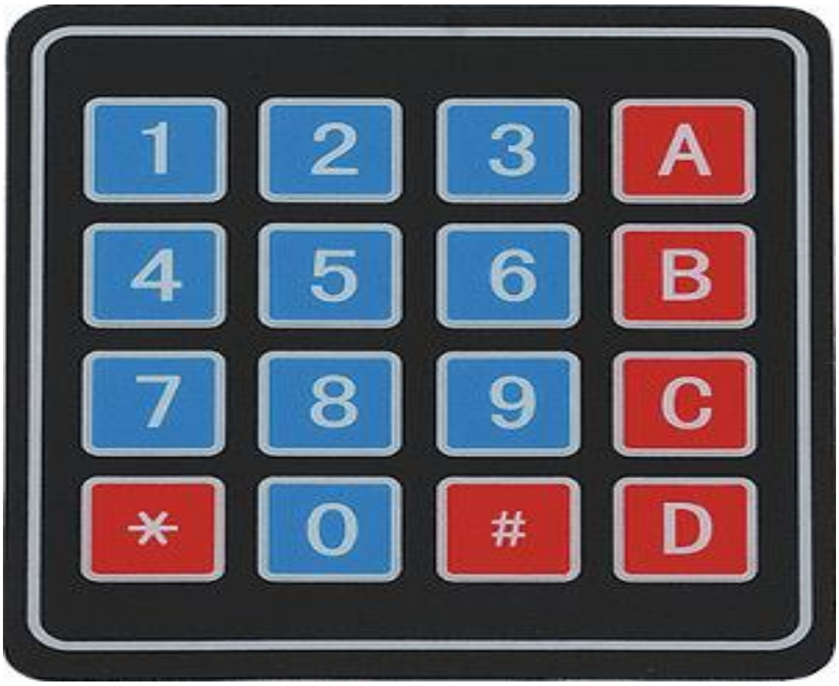
NIELIT, Gorakhpur

Course Name: A-level (1^{st} Sem.) **Subject:IoT**

Topic: Keypad interfacing with Arduino **Date: 09.04.2020**

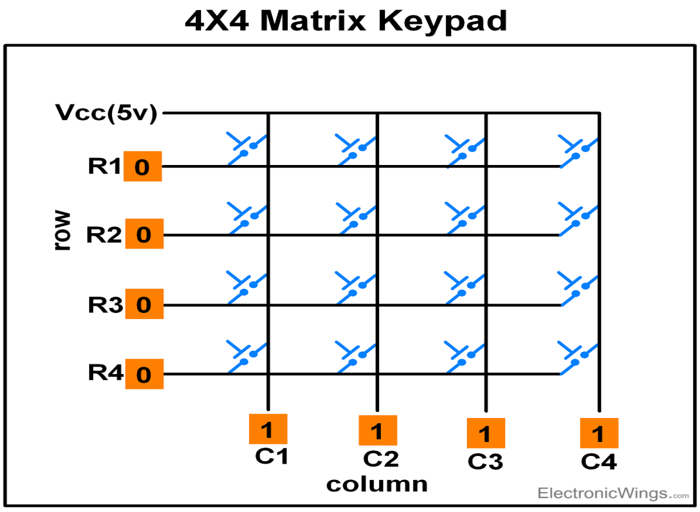
Introduction

- When we want to interface one key to the microcontroller then it needs one GPIO pin. But when we want to interface many keys like 9, 12 or 16 etc., then it may acquire all GPIO pins of microcontroller.
- To save some GPIO pins of microcontroller, we can use matrix keypad. Matrix keypad is nothing but keys arrange in row and column.
- E.g. if we want to interface 16 keys to the microcontroller then we require 16 GPIO pins but if we use matrix 4x4 keypad then we require only 8 GPIO pins of microcontroller.



4x4 Keypad

4x4 Keypad Matrix Structure

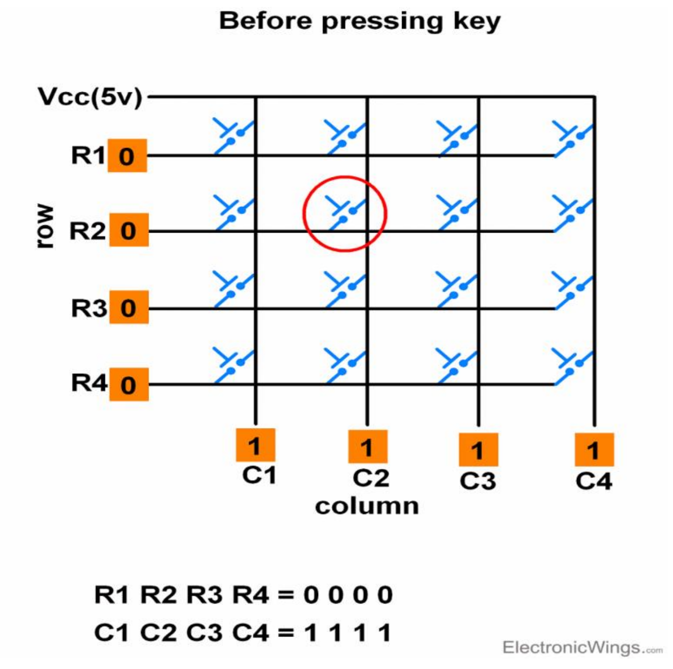


Keyboards are organized in a matrix of rows and columns. When a key is pressed, a row and a column make a contact Otherwise; there is no connection between rows and columns.

Keypad Matrix Working

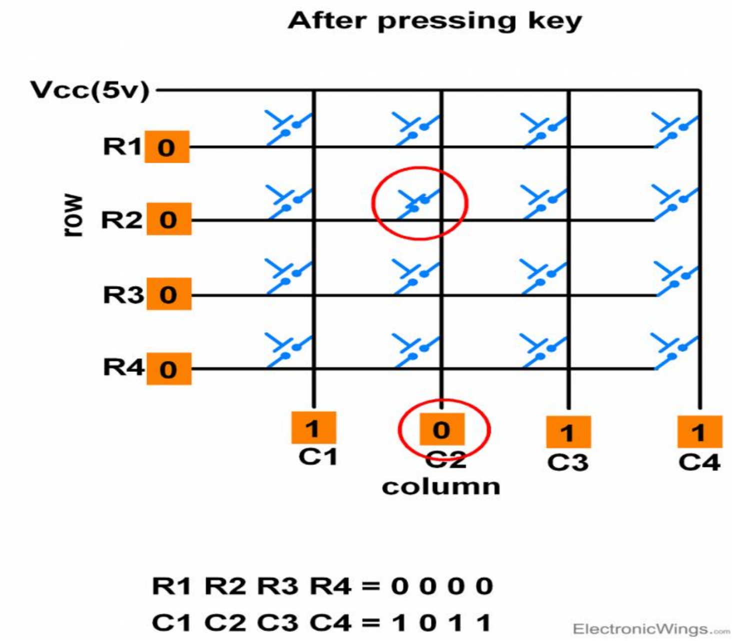
Scanning of Keys

To detect a pressed key, the microcontroller grounds all rows by providing 0 to the output latch, and then it reads the columns shown in above fig.



- If the data read from columns is = 1111, no key has been pressed shown in above fig. and the process continues till key press is detected.

- Now, consider highlighted key in above fig. is pressed. After pressing key, it makes contact of row with column shown below.



- If one of the column bits has a zero, this means that a key press has occurred.
- For example, if C1:C4 = 1011, this means that a key in the C2 column has been pressed.
- After detecting a key press, microcontroller will go through the process of identifying the key.

Process of Identifying the Key

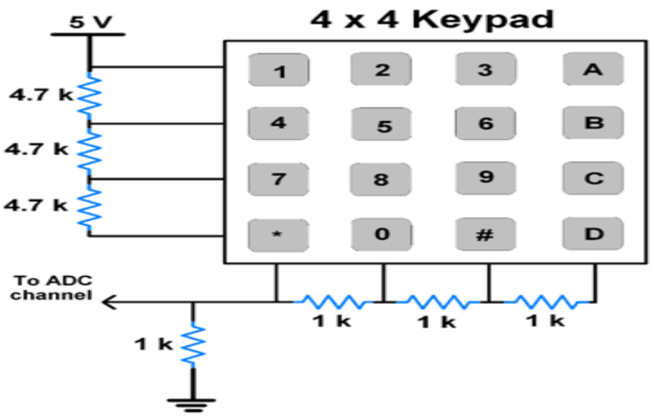
Starting from the top row, the microcontroller will ground it by providing a low to row R1 only.

- Now read the columns, if the data read is all 1s, no key in that row is pressed and the process continues for the next row.
- So, now ground the next row, R2. Read the columns, check for any zero and this process continues until the row is identified.
- E.g. In above case we will get row 2 in which column is not equal to 1111.
- So, after identification of the row in which the key has been pressed we can easily find out the key by row and column value.

Keypad Interfacing using One wire

It is possible to interface Keypad of any size with just one Analog pin. It is based on voltage divider network.

E.g. 4x4 keypad interfacing with one wire as shown below.



Note: Resistor value combinations can be different.

INSTALL THE LIBRARY

We'll use the Keypad library by Mark Stanley and Alexander Brevig. This library takes care of setting up the pins and polling the different columns and rows. To install the Keypad library, **go to Sketch > Include Library > Manage Libraries and search for "keypad"**. Click on the library, then click install.

Code

```
#include <Keypad.h>

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] = {
  {'0','1','2','3'},
  {'4','5','6','7'},
  {'8','9','A','B'},
  {'C','D','E','F'}
};
byte rowPins[ROWS] = {10, 11, 12, 13}; /* connect to the row pinouts of the keypad */
byte colPins[COLS] = {6, 7, 8, 9}; /* 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);
}

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

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

Functions Used

1. makeKeymap(keys)

    - This function is used to initialize the internal keymap to be equal to the user defined key map (in function syntax given above, keys).

2. Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols)

    - This defines an object customKeypad of the class Keypad and initializes it.
    - *rowPins* and *colPins* are the pins on Arduino to which the rows and columns of the keypad are connected to.
    - *rows* and *cols* are the number of rows and columns the keypad has.

3. customKeypad.getKey()

    - This function is used to identify which key is pressed on the keypad.