**Experiment-9**

**Aim:** Hands-on experimentation of 4x4 matrix keyboard interfacing with ATMega32 in C.

**Objectives:** After successfully completion of this experiment students will be able to,

* Use C language for ATMega32 microcontroller programming on AVRStudio.
* Experiment with 4 x 4 keypad matrix on ATMega32 AVR Development Board and learn hand on 4x4 keypad matrix in ATMega32.

# **Equipment required:**

* Windows7 or later based host computer
* ATMega32 Development board
* USBasp Programmer
* Jumper Wires
* LCD

# **Software required:**

* AVR Studio7 installation setup
* USBasp driver installation setup

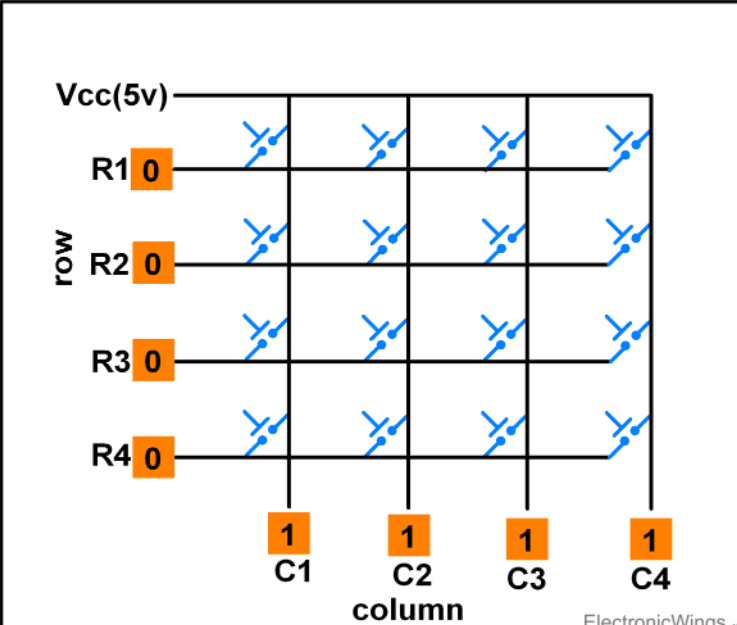
# **Theory:**

The keypad is used as an input device to read the key pressed by the user and to process it.

4x4 keypad consists of 4 rows and 4 columns. Switches are placed between the rows and columns. A keypress establishes a connection between the corresponding row and column between which the switch is placed.

In order to read the keypress, we need to configure the rows as outputs and columns as inputs. Columns are read after applying signals to the rows in order to determine whether or not a key is pressed and if pressed, which key is pressed.

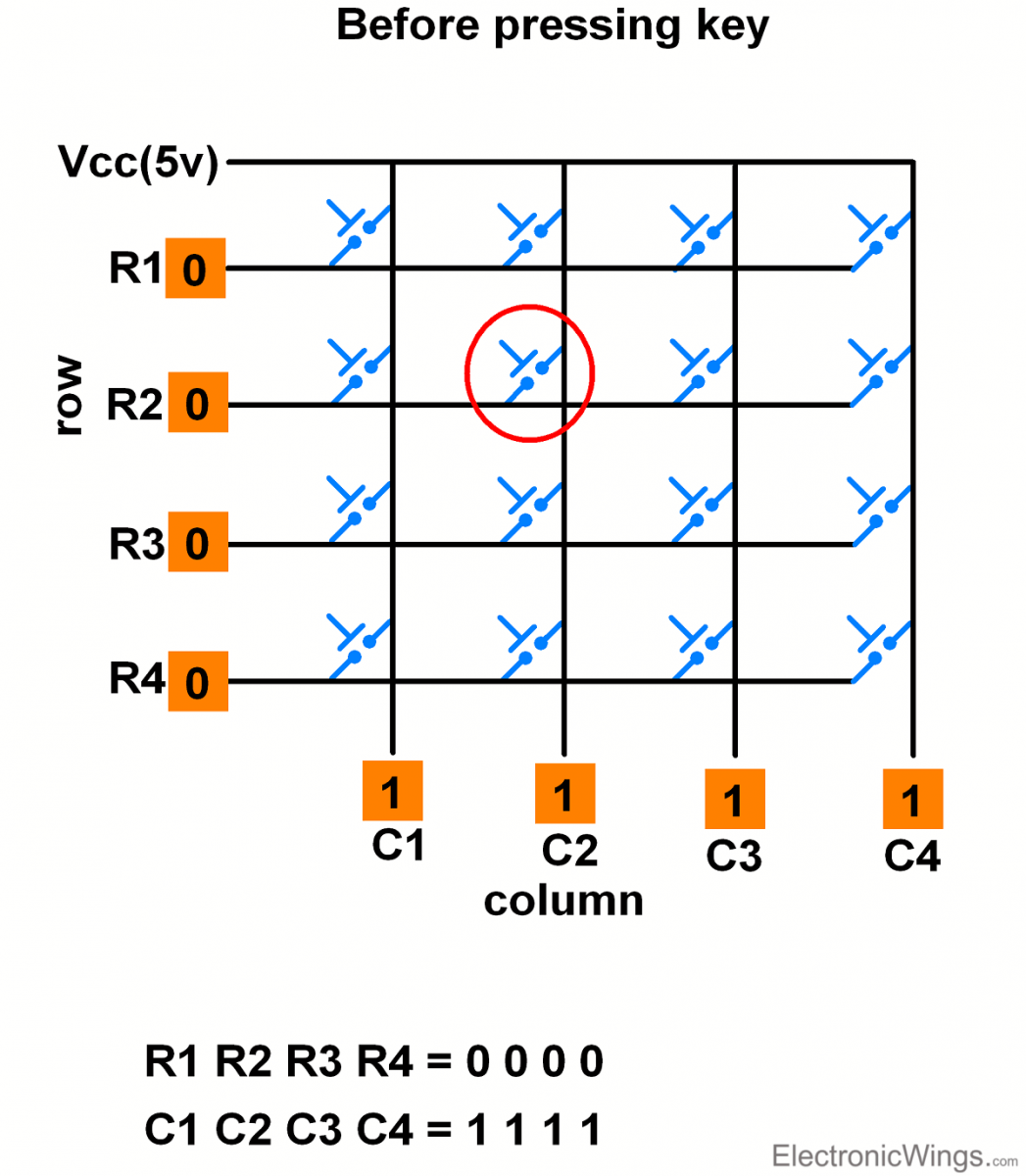
**4x4 Keypad Matrix**



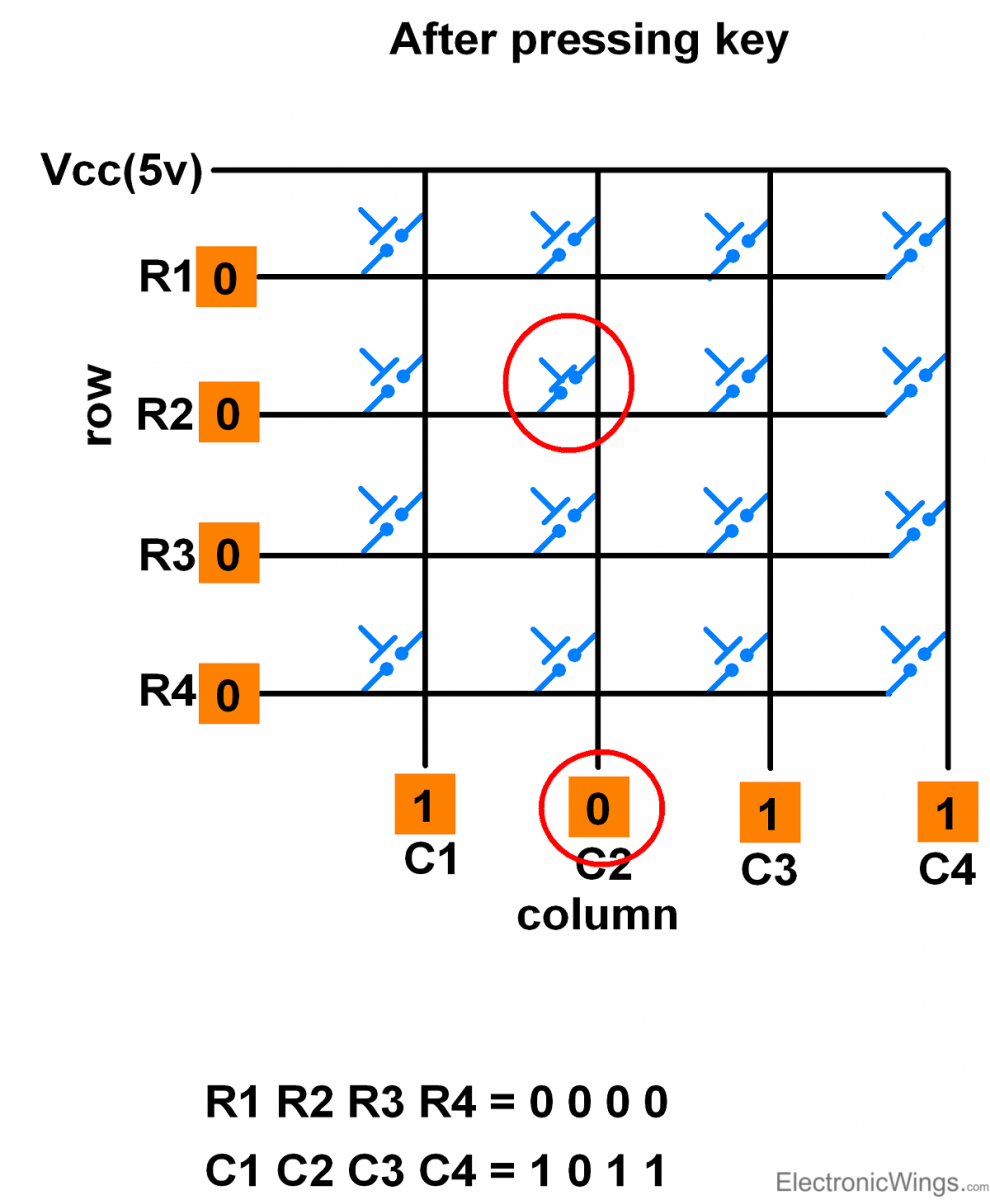
# Keypad Matrix Working

**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**

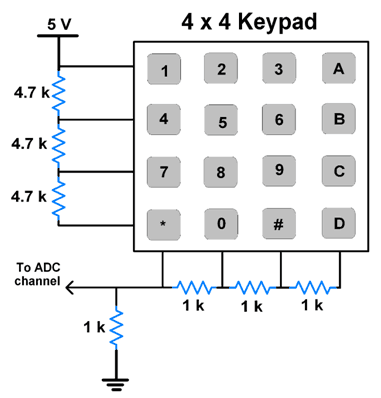
Starting from the top row, the microcontroller will ground it by providing a low to row R1 only.

* 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**

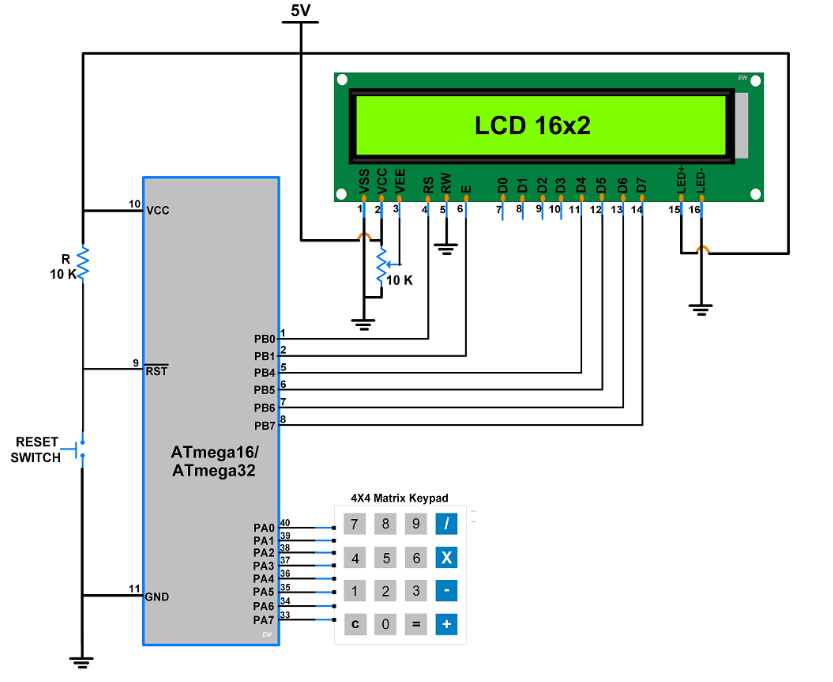
It is possible to interface Keypad of any size with just one Analog pin. It is based on voltage divider network.

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



**Note:** Resistor value combinations can be different.

**NOTE:** Here, in the development board, 4x4 keypad matrix is a on chip peripheral so we do not have to interface an external keypad with our development board.



CODE:

#include "LCD16x2\_4bit.h"

#include <avr/io.h>

#include <util/delay.h>

#define KEY\_PRT PORTA

#define KEY\_DDR DDRA

#define KEY\_PIN PINA

unsigned char keypad[4][4] = {{'7','8','9','/'},

{'4','5','6','\*'},

{'1','2','3','-'},

{' ','0','=','+'}};

unsigned char colloc, rowloc;

char keyfind()

{

while(1)

{

KEY\_DDR = 0xF0; /\* set port direction as input-output \*/

KEY\_PRT = 0xFF;

do

{

KEY\_PRT &= 0x0F; /\* mask PORT for column read only \*/

asm("NOP");

colloc = (KEY\_PIN & 0x0F); /\* read status of column \*/

}while(colloc != 0x0F);

do

{

do

{

\_delay\_ms(20); /\* 20ms key debounce time \*/

colloc = (KEY\_PIN & 0x0F); /\* read status of column \*/

}while(colloc == 0x0F); /\* check for any key press \*/

\_delay\_ms (40); /\* 20 ms key debounce time \*/

colloc = (KEY\_PIN & 0x0F);

}while(colloc == 0x0F);

/\* now check for rows \*/

KEY\_PRT = 0xEF; /\* check for pressed key in 1st row \*/

asm("NOP");

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 0;

break;

}

KEY\_PRT = 0xDF; /\* check for pressed key in 2nd row \*/

asm("NOP");

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 1;

break;

}

KEY\_PRT = 0xBF; /\* check for pressed key in 3rd row \*/

asm("NOP");

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 2;

break;

}

KEY\_PRT = 0x7F; /\* check for pressed key in 4th row \*/

asm("NOP");

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 3;

break;

}

}

if(colloc == 0x0E)

return(keypad[rowloc][0]);

else if(colloc == 0x0D)

return(keypad[rowloc][1]);

else if(colloc == 0x0B)

return(keypad[rowloc][2]);

else

return(keypad[rowloc][3]);

}

int main(void)

{

LCD\_Init();

LCD\_String\_xy(1,0,"Press a key");

while(1)

{

LCD\_Command(0xc0);

LCD\_Char(keyfind()); /\* Display which key is pressed \*/

}

}

OUTPUT:

CONCLUSION:

**Experiment-9 Post Lab Exercise**

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enrollment No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Answer the following questions:**

1. In reading the columns of a matrix, if no key is pressed we should get all in binary notation?
2. 0
3. 1
4. F
5. 7

2) To identify that which key is being pressed, we need to:

a) ground all the pins of the port at a time

b) ground pins of the port one at a time

c) connect all the pins of the port to the main supply at a time

d) none of the mentioned

3) Which is the application for keypad interfacing?

4) What is the use of 4x4 keypad?

5) What was the use of LCD in this experiment?