Experiment-7

<u>Aim</u> Hands-on experimentation of 4x4 matrix keyboard interfacing with ATMega32 in C.

<u>Objectives:</u> After successfully completing 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.

<u>Equipment required:</u>

- Windows7 or later based host computer
- ATMega32 Development Board
- USBasp Programmer
- Jumper Wires
- LCD

<u>Software required:</u>

- AVR Studio7 installation setup
- USBasp driver installation setup

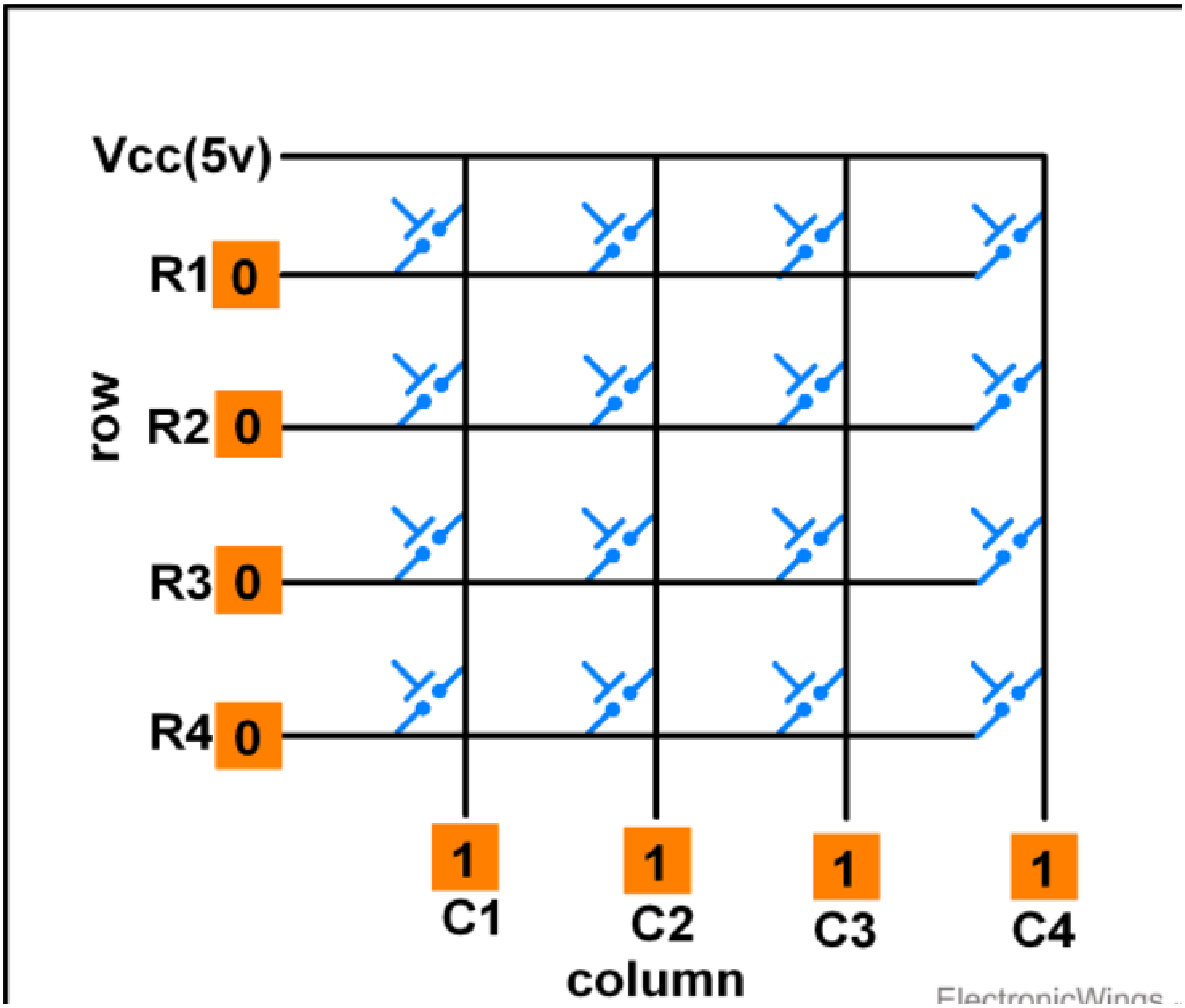
<u>Theory:</u>

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.

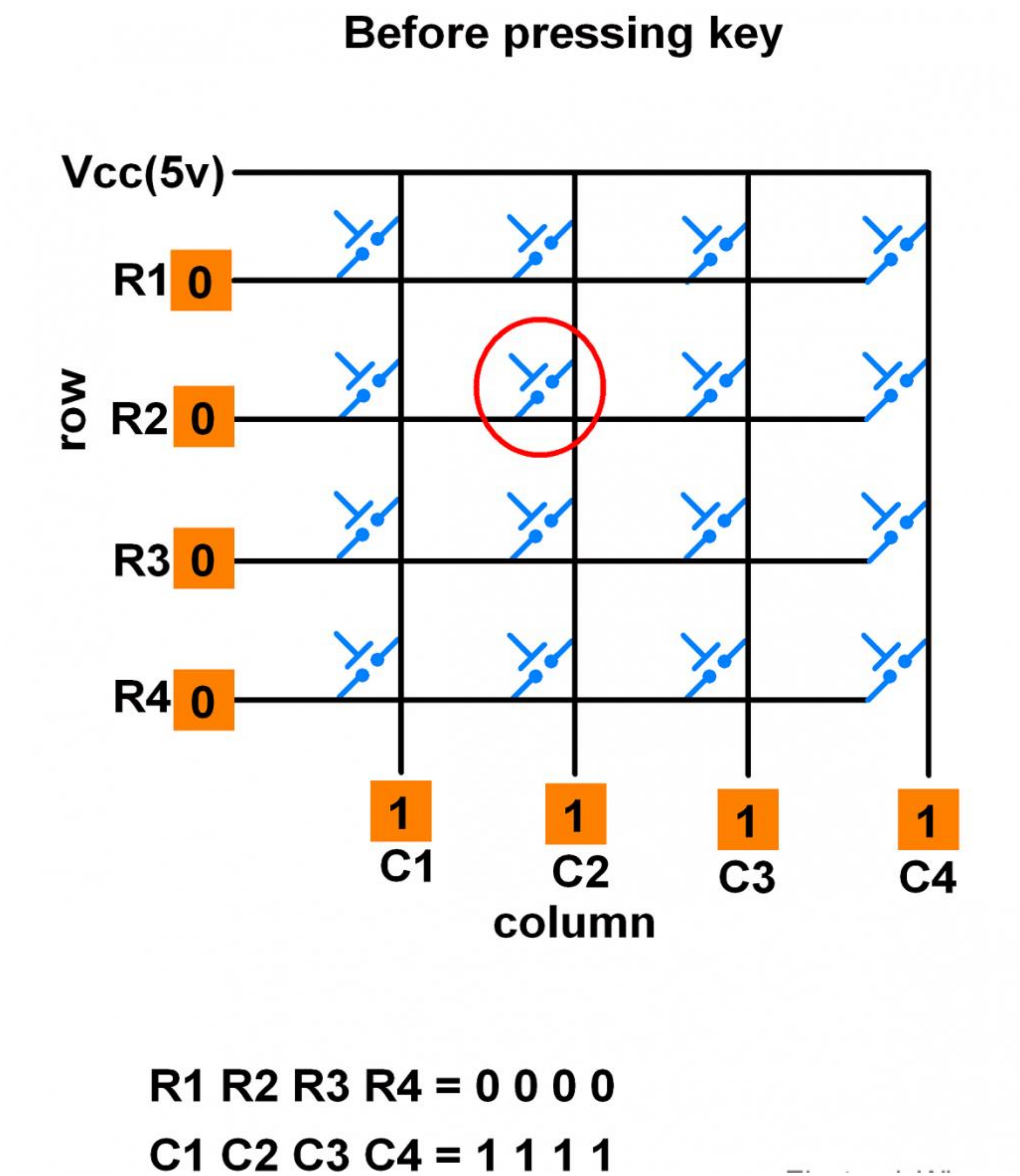
4x4 Keypad Matrix



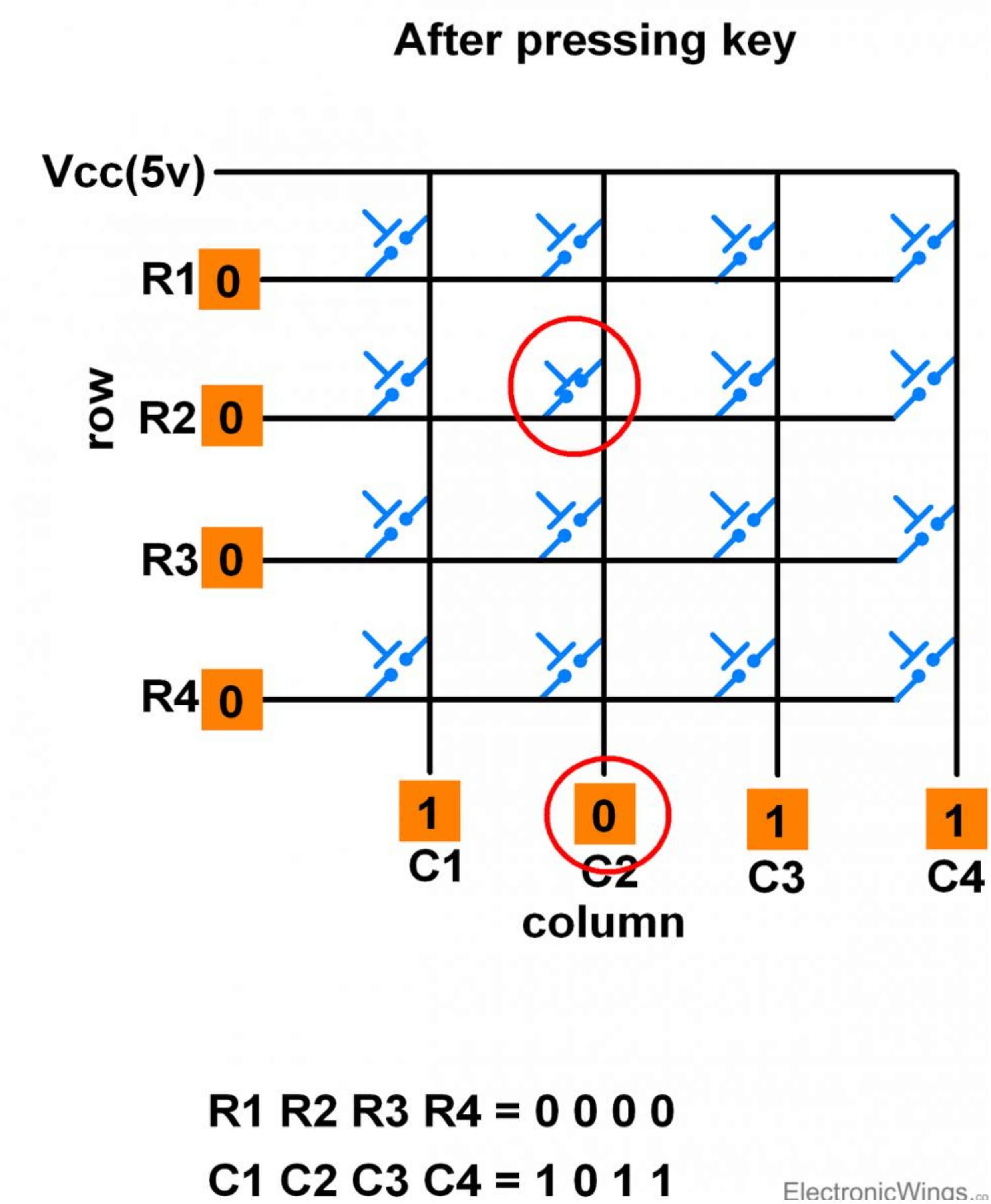
Keypad Matrix Working

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

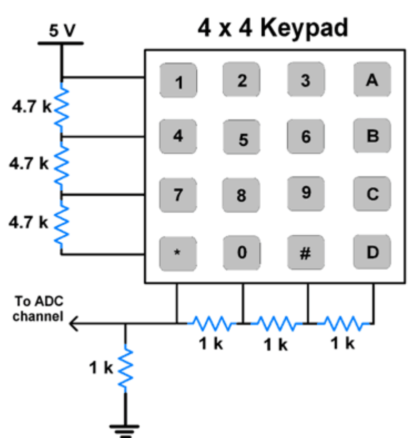
Starting from the top row, the microcontroller will ground it by providing a low to row R1 only.

- 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

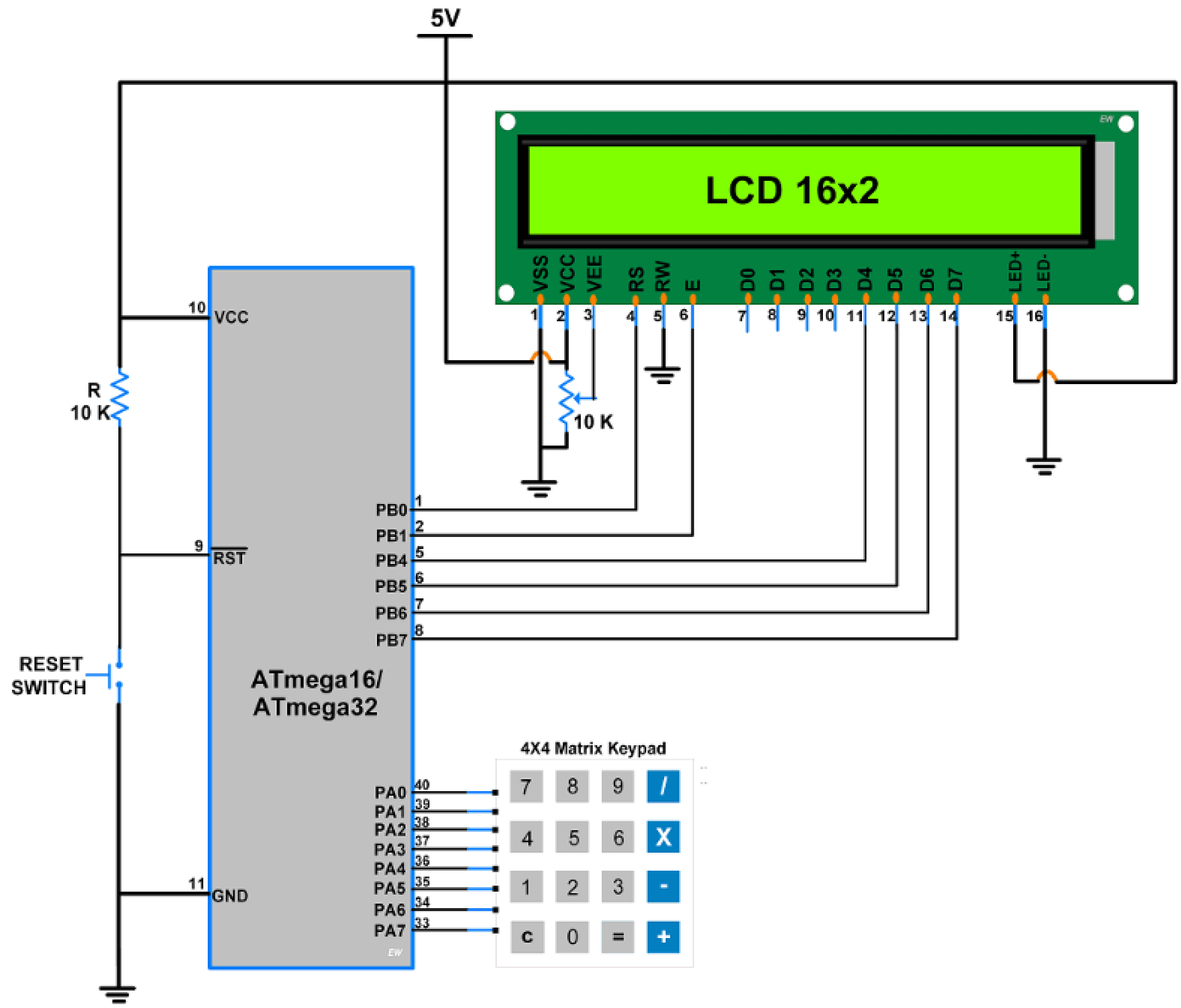
It is possible to interface Keypad of any size with just one Analog pin. It is based on voltage divider network.

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:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#define LCD PORTA
// LCD data port connected to PORTC
#define EN 7
#define RW 6
#define RS 5
```

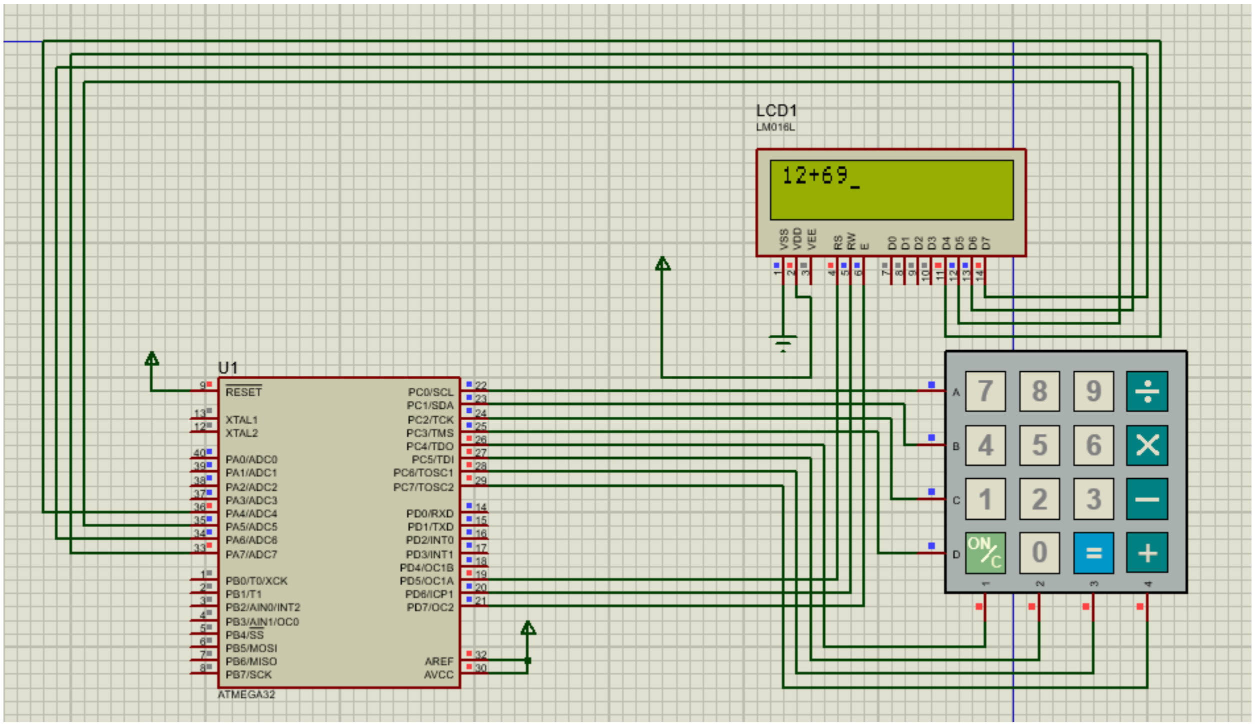
```
unsigned char keypad();
void lcdcmd(unsigned char cmd)
{
	PORTD &= ~(1 << RS); // RS=0 for command
	PORTD &= ~(1 << RW); // RW=0 for write
	LCD = cmd & 0xF0; // send upper nibble
	PORTD |= (1 << EN); // EN=1 for H to L pulse
	_delay_ms(1);
	PORTD &= ~(1 << EN); // EN=0 for H to L pulse
	LCD = cmd << 4; // send low nibble
	PORTD |= (1 << EN); // EN=1 for H to L pulse
	_delay_ms(1);
	PORTD &= ~(1 << EN);
}
void lcddata(unsigned char data)
{
	PORTD |= (1 << RS); // RS=1 for data //RW=0 for write
	PORTD &= ~(1 << RW);
	LCD = data & 0xF0; // send upper nibble
	PORTD |= (1 << EN); // EN=1 for H to L pulse
	_delay_ms(1);
	PORTD &= ~(1 << EN); // EN=0 for H to L pulse
	LCD = data << 4; // send low nibble
	PORTD |= (1 << EN); // EN=1 for H to L pulse
	_delay_ms(1);
	PORTD &= ~(1 << EN);
}
void lcd_init(){
	DDRA = 0xFF; // define output LCD port
	DDRD = 0xFF; // define RS,EN and RW pin as output
	PORTD &= ~(1 << EN); // initialize en = 0
	lcdcmd(0x33);
	lcdcmd(0x32);
	lcdcmd(0x28); // LCD in 4 bit mode
	lcdcmd(0x0E); // display on cursor on
	lcdcmd(0x01); // clear LCD
	_delay_ms(2);
}
int main(void)
{
```

```
	unsigned char x;
	DDRC = 0x0F; // Make PCO to PC3 = O/P and PC4 to PC7 = 1/P
	_delay_ms(1);
	PORTC = 0xF0;
	lcd_init();
	while (1)
	{
		PORTC = 0xF0; // Make all 4 columns 1 and all 4 rows 0
		_delay_ms(25);
		if (PINC != 0xF0)
		{
			x = keypad();
			lcddata(x);
		}
}
return 0;
}
unsigned char keypad()
{
	PORTC = 0b11111110; // make first row 0
	if ((PINC & (1 << PINC4)) == 0)
	{
		_delay_ms(125);
		return '7';
	}
	else if ((PINC & (1 << PINC5)) == 0)
	{
		_delay_ms(125);
		return '8';
	}
	else if ((PINC & (1 << PINC6)) == 0)
	{
		_delay_ms(125);
		return '9';
	}
	else if ((PINC & (1 << PINC7)) == 0)
	{
		_delay_ms(125);
		return '/';
	}
```

```
PORTC = 0b11111101; // make second row 0
 if ((PINC & (1 << PINC4)) == 0)
 {
 	_delay_ms(125);
 	return '4';
 }
 else if ((PINC & (1 << PINC5)) == 0)
 {
 	_delay_ms(125);
 	return '5';
 }
 else if ((PINC & (1 << PINC6)) == 0)
 {
 	_delay_ms(125);
 	return '6';
 }
 else if ((PINC & (1 << PINC7)) == 0)
 {
 	_delay_ms(125);
 	return '*';
 }
 PORTC = 0b11111011; // make third row 0
 if ((PINC & (1 << PINC4)) == 0)
 {
 	_delay_ms(125);
 	return '1';
 }
 else if ((PINC & (1 << PINC5)) == 0)
 {
 	_delay_ms(125);
 	return '2';
  }
  else if ((PINC & (1 << PINC6)) == 0)
  {
  	_delay_ms(125);
  	return '3';
  }
  else if ((PINC & (1 << PINC7)) == 0)
  {
  	_delay_ms(175);
```

```
            return '-';
        }
        PORTC = 0b11110111; // make forth row 0
        if ((PINC & (1 << PINC4)) == 0)
        {
            _delay_ms(125);
            return 'C';
        }
        else if ((PINC & (1 << PINC5)) == 0)
        {
            _delay_ms(125);
            return '0';
        }
        else if ((PINC & (1 << PINC6)) == 0)
        {
            _delay_ms(125);
            return '=';
        }
        else if ((PINC & (1 << PINC7)) == 0)
        {
            _delay_ms(125);
             return '+';
         }
         return 0;
    }
```

OUTPUT:



CONCLUSION:

After Performing this experiment I am able to program the Keypad.

Experiment-7 Post Lab Exercise

Student Name: Aryan Langhanoja
Enrollment No: 92200133030

Answer the following questions:

1) In reading the columns of a matrix, if no key is pressed we should get all in binary notation?
(a) 0
(b) 1
(c) F
(d) 7

Ans:- (a) - 0
Explanation: When a key is pressed, it connects the row and column, pulling the voltage level down to 0 (low). So, if no key is pressed in a keyboard matrix, we should generally get all 1s (high).

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

Ans:- (b) ground pins of the port one at a time

3) Which is the application for keypad interfacing?

- Security Systems: Keypads are often used in security systems to enter access codes or passwords.
- Electronic Door Locks: Keypads are integrated into electronic door locks for secure entry.
- ATMs (Automated Teller Machines): Keypads on ATMs allow users to input their PINs for transactions.
- Industrial Control Panels: Keypads are used in industrial settings for machine control and parameter input.
- Calculator and Numeric Entry Devices: Keypads are essential components in calculators and other numeric entry devices.
- Menu Navigation in Electronic Devices: Keypads are used in consumer electronics for menu navigation and user input.
- Home Automation: Keypads are employed in home automation systems for controlling various devices.
- POS (Point of Sale) Systems: Keypads are used in POS terminals for entering transaction amounts and other details.
- Telecommunication Devices: Keypads are found in telephones and other communication devices for dialing numbers and inputting information.

- Educational Kits: Keypads are used in educational kits for learning about microcontroller programming and interfacing.

4) What is the use of 4x4 keypad?

- User Input: The primary purpose of a 4x4 keypad is to allow users to input information easily by pressing the keys corresponding to specific characters, numbers, or functions.
- Password Entry: 4x4 keypads are commonly used for entering passwords or access codes in security systems, electronic locks, and other secure applications.
- Menu Navigation: In electronic devices or systems with menus, a 4x4 keypad can be used to navigate through different options and make selections. Numeric Entry: The keypad provides a convenient interface for entering numeric values,
  making it useful in calculators, cash registers, and other devices that require numerical input.
- Control Interfaces: 4x4 keypads are often integrated into control panels for industrial machines or systems, allowing operators to input commands or parameters.
- DIY Electronics Projects: Hobbyists and electronics enthusiasts often use 4x4 keypads in do-it-yourself (DIY) projects for various applications, such as home automation or robotics.
- Educational Purposes: 4x4 keypads are used in educational settings to teach students about microcontroller programming, interfacing, and digital input methods.

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

- User Feedback with LCD Integration: The LCD displays information to provide feedback to the user about the key they pressed or the result of the keypad input. This visual feedback enhances the user interface and makes the system more user-friendly.
- Displaying Characters: The LCD can be used to display characters, numbers, or symbols corresponding to the keys pressed on the keypad. This is particularly useful when entering passwords, codes, or any other alphanumeric information.
- Output Presentation: If the 4x4 keypad is used for controlling a system or entering commands, the LCD can be employed to show the current status, menu options, or any relevant information regarding the system's operation.
- Debugging and Testing: The LCD can be used for debugging and testing purposes. It allows the programmer or user to visualize intermediate results or debug information during the development phase.
- Enhancing User Interaction: The combination of a 4x4 keypad and an LCD enhances the overall interaction with the system. Users can see and confirm their input, making the system more intuitive and responsive.
- Educational Purposes: When used in educational projects, integrating an LCD provides a tangible and visible output, aiding students in understanding the connection between keypad inputs and displayed information.

Conclusion:-

In simple terms, a 4x4 keypad is an input device with 4 rows and 4 columns, and buttons are placed at the intersections of rows and columns. To identify which button is pressed, the microcontroller sequentially grounds each row and checks the columns. If a column reads as 0, a key in that row is pressed.