



Keypad is most widely used input device to provide input from the outside world to the [microcontroller](#). The keypad makes an application more users interactive. The concept of interfacing a keypad with the [ATmega16](#) is similar to interfacing it with any other microcontroller. The article of [Interfacing keypad with 8051](#) can be referred for detailed description of the methodology used here.

**This article explains the interfacing of a 4x3 keypad with [AVR microcontroller](#) (ATmega16) and displaying the output on a [LCD](#).**

The algorithm and detailed explanation for [keypad interfacing](#) is given in above mentioned article. The brief steps to interface the keypad with [AVR](#) are written below:

1. Configure the row pins or column pins.
2. Make all output pins to low and input pins to high.
3. Keep monitoring the port value, where the key pad is connected.

```
while(1)
```

```
{
```

```

        PORTD=0xF0;           //set all the input to one
        value=PIND;           //get the PORTD value in variable "value"
        if(value!=0xF0)       //if any key is pressed value changed
        {
                check1();
                check2();
                check3();
                check4();
        }
}

```

4. If there is any change in port value, make one of the output pin of port to zero and rest all high.

```

void check1(void)
{
        //DDRD = 0xF0;
        pad = 0b11111110;
        //pad &= (0<<r1);
        _delay_us(10);
        if(bit_is_clear(PIND,c1))
                LCD_write('1');
        else if(bit_is_clear(PIND,c2))
                LCD_write('2');
        else if(bit_is_clear(PIND,c3))
                LCD_write('3');
}

```

5. If any of input pin found zero, write the particular pin data to [LCD](#), else continue with the step

## Source Code :

```

/ Program to get input from keypad and display it on LCD.
#include<avr/io.h>
#include<util/delay.h>

#define pad PORTD
#define r1 PD0
#define r2 PD1
#define r3 PD2
#define r4 PD3

#define c1 PD4
#define c2 PD5

```

```

#define c3 PD6

void check1(void);
void check2(void);
void check3(void);
void check4(void);

#define LCD_DATA PORTA          //LCD data port

#define ctrl PORTB
#define en PB2                  //enable signal
#define rw PB1                  //read/write signal
#define rs PB0                  //resister select signal

void LCD_cmd(unsigned char cmd);
void init_LCD(void);
void LCD_write(unsigned char data);

unsigned int press;

int main()
{
    unsigned char value;
    DDRA=0xff;                  //LCD_DATA port as output port
    DDRB=0x07;                  //signal as out put
    DDRD=0x0F;
    pad=0xf0;
    init_LCD();                  //initialization of LCD
    LCD_write_string("press a key");
    LCD_cmd(0xc0);

    while(1)
    {
        PORTD=0xF0;              //set all the input to one
        value=PIND;              //get the PORTD value in variable "value"
        if(value!=0xf0)          //if any key is pressed value changed
        {
            check1();
            check2();
            check3();
            check4();
        }
    }
    return 0;
}

void check1(void)

```

```

{
    //DDRD = 0xf0;
    pad = 0b11111110;
    //pad &= (0<<r1);
    _delay_us(10);
    if(bit_is_clear(PIND,c1))
        LCD_write('1');
    else if(bit_is_clear(PIND,c2))
        LCD_write('2');
    else if(bit_is_clear(PIND,c3))
        LCD_write('3');
}

```

```

void check2(void)
{
    pad=0b11111101;
    /pad &= (0<<r2);
    _delay_us(10);
    if(bit_is_clear(PIND,c1))
        LCD_write('4');
    else if(bit_is_clear(PIND,c2))
        LCD_write('5');
    else if(bit_is_clear(PIND,c3))
        LCD_write('6');
}

```

```

void check3(void)
{
    pad=0b11111011;
    //pad &= (0<<r3);
    _delay_us(10);
    if(bit_is_clear(PIND,c1))
        LCD_write('7');
    else if(bit_is_clear(PIND,c2))
        LCD_write('8');
    else if(bit_is_clear(PIND,c3))
        LCD_write('9');
}

```

```

void check4(void)
{
    pad = 0b11110111;
    //pad &= (0<<r4);
    _delay_us(10);
    if(bit_is_clear(PIND,c1))
        LCD_write('#');
    else if(bit_is_clear(PIND,c2))

```

```

    LCD_write('0');
    else if(bit_is_clear(PIND,c3))
        LCD_write('*');
}

void init_LCD(void)
{
    LCD_cmd(0x38);      //initializtion of 16X2 LCD in 8bit mode
    _delay_ms(1);

    LCD_cmd(0x01);      //clear LCD
    _delay_ms(1);

    LCD_cmd(0x0E);      //cursor ON
    _delay_ms(1);

    LCD_cmd(0x80);      // ---8 go to first line and --0 is for 0th position
    _delay_ms(1);
    return;
}

void LCD_cmd(unsigned char cmd)
{
    LCD_DATA=cmd;
    ctrl =(0<<rs)|(0<<rw)|(1<<en); // making RS and RW as LOW and EN as HIGH
    _delay_ms(1);
    ctrl =(0<<rs)|(0<<rw)|(0<<en); // making RS, RW , LOW and EN as LOW
    _delay_ms(50);
    return;
}

void LCD_write(unsigned char data)
{
    LCD_DATA= data;
    ctrl = (1<<rs)|(0<<rw)|(1<<en); // making RW as LOW and RS, EN as HIGH
    _delay_ms(1);
    ctrl = (1<<rs)|(0<<rw)|(0<<en); // making EN and RW as LOW and RS HIGH
    _delay_ms(50);          // give a 10 milli second delay to get thigs executed
    return ;
}

void LCD_write_string(unsigned char *str)    //take address vaue of the string in pionter *str
{
    int i=0;
    while(str[i]!='\0')          // loop will go on till the NULL charaters is soon in string
    {

```

```

        LCD_write(str[i]);           // sending data on CD byte by byte
        i++;
    }
    return;
}

```



#### [Preset](#)

A preset is a three legged electronic component which can be made to offer varying resistance in a circuit. The resistance is varied by adjusting the rotary control over it. The adjustment can be done by using a small screw driver or...

- 19,907-Reads



#### [LCD](#)

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments...](#)

- 50,884-Reads



#### [ATmega16](#)

**ATmega16** is an 8-bit high performance microcontroller of Atmel's Mega [AVR](#) family with low power consumption. Atmega16 is based on enha...

- 24,059-Reads