//LCD Program

#define *F\_CPU* 8000000UL /\* Define CPU Frequency e.g. here 8MHz \*/

#include <avr/io.h> /\* Include AVR std. library file \*/

#include <util/delay.h> /\* Include inbuilt defined Delay header file \*/

#define LCD\_Data\_Dir DDRD /\* Define LCD data port direction \*/

#define LCD\_Data\_Port PORTD /\* Define LCD data port \*/

#define LCD\_Command\_Dir DDRB /\* Define LCD command port direction register \*/

#define LCD\_Command\_Port PORTB /\* Define LCD data port \*/

#define RS PB0 /\* Define Register Select (data/command reg.)pin \*/

#define RW PB1 /\* Define Read/Write signal pin \*/

#define EN PB2 /\* Define Enable signal pin \*/

#define KEY\_PRT PORTA //keyboard PORT

#define KEY\_DDR DDRA //keyboard DDR

#define KEY\_PIN PINA //keyboard PIN

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

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

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

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

unsigned char colloc, rowloc;

void LCD\_Command(unsigned char cmnd)

{

LCD\_Data\_Port= cmnd;

LCD\_Command\_Port &= ~(1<<RS); /\* RS=0 command reg. \*/

LCD\_Command\_Port &= ~(1<<RW); /\* RW=0 Write operation \*/

LCD\_Command\_Port |= (1<<EN); /\* Enable pulse \*/

*\_delay\_us*(1);

LCD\_Command\_Port &= ~(1<<EN);

*\_delay\_ms*(3);

}

void LCD\_Char (unsigned char char\_data) /\* LCD data write function \*/

{

LCD\_Data\_Port= char\_data;

LCD\_Command\_Port |= (1<<RS); /\* RS=1 Data reg. \*/

LCD\_Command\_Port &= ~(1<<RW); /\* RW=0 write operation \*/

LCD\_Command\_Port |= (1<<EN); /\* Enable Pulse \*/

*\_delay\_us*(1);

LCD\_Command\_Port &= ~(1<<EN);

*\_delay\_ms*(1);

}

void LCD\_Init (void) /\* LCD Initialize function \*/

{

LCD\_Command\_Dir = 0xFF; /\* Make LCD command port direction as o/p \*/

LCD\_Data\_Dir = 0xFF; /\* Make LCD data port direction as o/p \*/

*\_delay\_ms*(20); /\* LCD Power ON delay always >15ms \*/

LCD\_Command (0x38); /\* Initialization of 16X2 LCD in 8bit mode \*/

LCD\_Command (0x0C); /\* Display ON Cursor OFF \*/

//LCD\_Command (0x06); /\* Auto Increment cursor \*/

LCD\_Command (0x01); /\* Clear display \*/

LCD\_Command (0x80); /\* Cursor at home position \*/

}

void LCD\_String (char \*str) /\* Send string to LCD function \*/

{

int i=0;

for(i=0;str[i]!=0;i++) /\* Send each char of string till the NULL \*/

{

LCD\_Char(str[i]);

}

}

void LCD\_Clear()

{

LCD\_Command (0x01); /\* clear display \*/

LCD\_Command (0x80); /\* cursor at home position \*/

}

//Embedded C Program for interfacing Keypad to AVR Microcontroller and display key pressed on LCD

/\*4x4 Keypad Interfacing with ATmega16/32 \*/

char keyfind()

{

while(1)

{

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

KEY\_PRT = 0xFF;

/\* now check for rows \*/

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

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 0;

break;

}

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

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 1;

break;

}

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

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 2;

break;

}

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

colloc = (KEY\_PIN & 0x0F);

if(colloc != 0x0F)

{

rowloc = 3;

break;

}

}

if(colloc == 0x0E){

LCD\_Clear();

return(keypad[rowloc][0]);

}

else if(colloc == 0x0D){

LCD\_Clear();

return(keypad[rowloc][1]);

}

else if(colloc == 0x0B){

LCD\_Clear();

return(keypad[rowloc][2]);

}

else{

LCD\_Clear();

return(keypad[rowloc][3]);

}

}

int main(void)

{

LCD\_Init();

LCD\_String("Press a key");

while(1)

{

LCD\_Command(0xc0);

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

}

}