```
/*
 * LCD_Testing.c
 *
 * Created: 2022-05-12 13:48:03
 * Author : ra5327ru-s
 */
#define F_CPU 16000000UL
#include <stdio.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include  <util/delay.h>
#include <time.h>

#define KEY_PRT PORTA
#define KEY_DDR DDRA
#define KEY_PIN PINA
#define LCD_Data_Dir DDRB
#define LCD_Command_Dir DDRD
#define LCD_Data_Port PORTB
#define LCD_Command_Port PORTD
#define BUZZ PD2
#define SENSOR PC0
#define RS PD4
#define RW PD5
#define EN PD6

int alarm_activated = 0;
int triggered = 0;


void LCD_Command(unsigned char cmnd)
{

	LCD_Data_Port = cmnd;
	LCD_Command_Port &= ~ (1<<RS);
	LCD_Command_Port &= ~ (1<<RW);
	LCD_Command_Port |= (1<<EN);
	_delay_us(1);
	LCD_Command_Port &= ~ (1<<EN);
	_delay_ms(3);
}


void LCD_Char(unsigned char_data){
    LCD_Data_Port = char_data;
    LCD_Command_Port |= (1<<RS);
    LCD_Command_Port &= ~ (1<<RW);
    LCD_Command_Port |= (1<<EN);
    _delay_us(1);
    LCD_Command_Port &= ~(1<<EN);
    _delay_ms(1);

}

void LCD_Init(void){
    LCD_Command_Dir = 0xFF;
    LCD_Data_Dir = 0xFF;
    _delay_ms(20);

    LCD_Command(0x38);
    LCD_Command(0x0C);
    LCD_Command(0x06);
    LCD_Command(0x01);
    LCD_Command(0x80);
    _delay_ms(2);
    }

void LCD_String(char*str){

    int i;
    for(i = 0; str[i]!=0;i++){
        LCD_Char(str[i]);
    }
}



void LCD_String_xy (char row, char pos, char*str){

    if(row == 0 && pos < 16){
        LCD_Command((pos & 0x0F) | 0x80);
        LCD_String(str);
    }
        else if(row == 1 && pos < 16){
        LCD_Command((pos & 0x0F) | 0xC0);
        LCD_String(str);
        }
```

```
       }

void LCD_Clear(){

       LCD_Command(0x01);
       LCD_Command(0x80);

}


unsigned char keypad [4][4] = { {'9', '8', '7', '*'},
                                                         {'6', '5' , '4', '/'},
                                                         {'3' , '2' ,'1', '-'},
                                                         {'R' , '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]);

}
void sound(int length){
    LCD_Clear();
    LCD_String_xy(0,0, "Popo pull up");
    LCD_String_xy(1,0,"Skriv in kod");
    for(int i = 0; i < length; i++){
        LCD_Command_Port |= (1<<BUZZ);
        _delay_ms(300);
        LCD_Command_Port &= ~(1<<BUZZ);
        _delay_ms(300);
        if(alarm_activated == 0){
            break;
        }

    }
}

void alarm_triggered(void) {
    DDRC &= ~(1<<SENSOR);
    int temp = PINC & 0x01;
    if(temp == 0x01){
        triggered = 1;
    } else {
        triggered = 0;
    }

}


ISR(PCINT2_vect){
    alarm_triggered();
    if(alarm_activated == 1 && triggered == 1){
        sound(5);
        main();
    }
}


int main(void)
{
```

```
PCICR = 0b00000100;
PCMSK2 = 0b00000001;
LCD_Init();

LCD_String_xy(0,0, "Ange kod:");

int i = 0;
char correct[6] = {'2', '2', '0', '2','2','4'};
char inserted[6] = { ' ' };
int right = 0;
sei();

while(1){
	inserted[i] =  keyfind();
	LCD_Char(inserted[i]);


	if(i == 5){
		for(i = 0; i < 6; i++){
			if(inserted[i] == correct[i]){
				right = 1;
			} else{
				right = 0;
				break;
			}
		}
		if (right == 1){
			_delay_ms(200);
			LCD_Clear();
			LCD_String_xy(0,0, "Korrekt kod");
			_delay_ms(500);
			if(alarm_activated == 0){
				alarm_activated = 1;
				LCD_Clear();
				LCD_String_xy(0,0, "Larmet");
				LCD_String_xy(1, 0, "Aktiverat");
				_delay_ms(700);
				main();
			} else {
				LCD_Clear();
				LCD_String_xy(0,0, "Larmet");
				LCD_String_xy(1,0, "Avaktiverat");
				alarm_activated = 0;
				_delay_ms(700);
```

```
                    main();
                }
            } else{
                _delay_ms(200);
                LCD_Clear();
                LCD_String_xy(0,0, "Fel kod");
                _delay_ms(700);
                if(alarm_activated == 1){
                    sound(5);
                }

                main();
            }
        }

        i+=1;
    }
}
```