```
/*
 * OperationProjekt.c
 *
 * Created: 2019-04-15 15:38:45
 * Author : ja1503an-s
 */
#define F_CPU 8000000UL
#include <util/delay.h>
#include <stdio.h>
#include <avr/io.h>
#include <time.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#include <string.h>

#define BUTTON1_PRESSED PINB & (1<<PB5) //Button is pressed
#define BUTTON2_PRESSED PINB & (1<<PB6) //Button is pressed
#define TWEEZERS PINB & (1<<PB7)

#define B_H PORTA |=(1<<PA3);
#define B_L PORTA &= ~(1<<PA3);

#define C_H PORTA |=(1<<PA4);
#define C_L PORTA &= ~(1<<PA4);

#define D_H PORTA |=(1<<PA5);
#define D_L PORTA &= ~(1<<PA5);

#define E_H PORTA |=(1<<PA2);
#define E_L PORTA &= ~(1<<PA2);

#define RS_H  PORTA |=(1<<PA0);
#define  RS_L PORTA &= ~(1<<PA0);

#define RW_H PORTA |=(1<<PA1);
#define RW_L PORTA &= ~(1<<PA1);

void clear_display(void);
void write_string(char string[]);
void init(void);
void buzzer(void);
void redDiod(void);
void timer(void);
```

```
void timeStart(void);
void timeEnd(void);
void penalty(void);
void displayOn(void);
void functionSet(void);
void displayOff(void);
void write_cmd(char c);
void write_char(char c);
void read_cmd(char c);
void write_string(char string[]);
void time_keep(void);
void timer0_init(void);
void button1(void);
void setDDRAM(char c);
void initVals(void);
void penalty(void);
void resetVals(void);
void highScore(void);
void button2(void);
void resetScoreboard(void);
void button1ForLetters(void);
void chooseLetter(void);
void button2ForLetters(void);
void readEeprom(void);
void writeToEeprom(void);

ISR(TIMER0_OVF_vect);
uint8_t  EEMEM EEPROMString[3];
int  EEMEM EEPROMINT;
char iForLetter;
uint8_t timerOverflowCount=0,sec;
uint8_t start,end,totalTime,penaltyTime;
int timerstart =1;
char result[5];
char scoreVal[5];
char highscoreToString[5];
int b1 = 0;
int n = 0;
int tioT;
int currentTime;
int penaltyDelay;
int highScore1;
int iForChoosing;
```

```
char newName[4];
typedef int bool;
#define true 1
#define false 0

bool pressedB1,notPressedB1,pressedB2,
scoreWritten,gamePlayed,gameReset,choosingLetters,pressedB2L,pressedB1L;


typedef struct{
	int playerScore;
	char player[3];
}Highscores;

Highscores arr[4];
int int_cmp (const void * a, const void * b)
{
	Highscores *scoreA = (Highscores *)a;
	Highscores *scoreB = (Highscores *)b;
	return ( scoreA->playerScore - scoreB->playerScore );
}
Highscores EEMEM EEStruct;
int main(void)
{

	TCNT0=0x00;
	TCCR0 = (1<<CS02);
	init();
	displayOn();
	functionSet();
	clear_display();
	timer0_init();
	sei();
	initVals();
	readEeprom();
	while(1)
	{

	while(BUTTON1_PRESSED){
		button1();
		timeStart();
		//buzzer();
```

```
        redDiod();
        clear_display();
        resetVals();
        setDDRAM(0b10000000);
        choosingLetters = true;
        write_string("Your time is: ");
}
        pressedB1 = false;

        /*************************************************************************/
        /* Spelet körs                                                           */
        /*************************************************************************/
        while (n == 1)
        {
                while (BUTTON1_PRESSED)
                {
                        button1();

                }
                pressedB1 = false;



                if (TWEEZERS)
                {
                        redDiod();
                        buzzer();
                        penalty();
                }

                currentTime = sec - start + penaltyTime;
                itoa(currentTime,result,10);
                setDDRAM(0xc0);
                write_char(result[0]);

                if (result[1]!=0)
                {
                        setDDRAM(0xc1);
                        write_char(result[1]);
                }
                if (result[2]!=0)
                {
                        setDDRAM(0xc2);
```

```
                write_char(result[2]);
            }
            scoreWritten = false;
            gamePlayed = true;
                while (BUTTON2_PRESSED)
                {
                    button2();
                }
                pressedB2 = false;
        }

        if(gamePlayed == true && gameReset == false){
            timeEnd();
            gamePlayed = false;
        }

        if(scoreWritten == false){
            clear_display();

            highScore();
            scoreWritten = true;
        }

        while (BUTTON2_PRESSED)
                {
                    resetScoreboard();
                }
                pressedB2 = false;
    }
}

void init(){
    DDRA = 0b11111111; //Port A output
    _delay_ms(2);
    DDRB = 0b11101111; //Port B biderictional for 0-2, output for 3-4 and input for 5
        _delay_ms(2);
    DDRB = 0xFF; //Port B output
        _delay_ms(2);
    DDRD = 0xFF; //Port D output
        _delay_ms(2);
      DDRB |= 1<<PB3;
        _delay_ms(2);
      DDRB |= 1<<PB4;
```

```
                _delay_ms(2);
            DDRB &= ~(1<<PB5); //ändrade från DDRD till DDRB
                _delay_ms(2);
}

void initVals(){
        pressedB1 = true;
        notPressedB1 = true;
        scoreWritten = false;
        penaltyDelay = -6;
        highScore1 = 10000;
        gamePlayed = false;
        pressedB2 = true;
        gameReset = false;
        choosingLetters = true;
        pressedB2L =true;
        pressedB1L = true;

        strcpy(arr[0].player,"NUL");
        arr[0].playerScore = 255;

        strcpy(arr[1].player,"NUL");
        arr[1].playerScore = 255;

        strcpy(arr[2].player,"NUL");
        arr[2].playerScore = 255;

        strcpy(arr[3].player,"NUL");
        arr[3].playerScore = 255;
        iForLetter = 0x65;
}
void buzzer(){
        PORTB ^= 1<<PB4;

}

void redDiod(){
        PORTB ^= 1<<PB3;
}

void highScore(){
        setDDRAM(0x80);
        write_string("Highscores!");
```

```
        qsort(arr, 4, sizeof(Highscores),int_cmp);
        if (arr[0].playerScore == 255)
        {
        setDDRAM(0xc0);
         write_string("No games played yet!");
        }else{


        setDDRAM(0xC0);
        itoa(arr[0].playerScore,scoreVal,10);
        write_char(arr[0].player[0]);
        write_char(arr[0].player[1]);
        write_char(arr[0].player[2]);
        write_char(0x20);
        write_string(scoreVal);
        if (arr[1].playerScore!= 255)
        {
        setDDRAM(0x94);
        itoa(arr[1].playerScore,scoreVal,10);
        write_char(arr[1].player[0]);
        write_char(arr[1].player[1]);
        write_char(arr[1].player[2]);
        write_char(0x20);
        write_string(scoreVal);
        }
        if (arr[2].playerScore!=255)
        {
        setDDRAM(0xD4);
        itoa(arr[2].playerScore,scoreVal,10);
        write_char(arr[2].player[0]);
        write_char(arr[2].player[1]);
        write_char(arr[2].player[2]);
        write_char(0x20);
        write_string(scoreVal);
        }

        }
        writeToEeprom();
}
void readEeprom(){
        for (int j=0;j<3;j++)
        {
        for (int i= 0; i<3;i++)
```

```
        {
                arr[j].player[i] = eeprom_read_byte((uint8_t*) i+(j*3));
        }
        }

        for (int i = 0;i<3;i++)
        {
                arr[i].playerScore = eeprom_read_byte((uint8_t*) i+9);
        }


}
void writeToEeprom(){
        int j = 0;
        for(int i= 0; i<3;i++)
        {
                eeprom_write_byte ((uint8_t*) 0+j*3, arr[i].player[0]);
                eeprom_write_byte ((uint8_t*) 1+j*3, arr[i].player[1]);
                eeprom_write_byte ((uint8_t*) 2+j*3 , arr[i].player[2]);
                j++;
        }

        for (int i=0; i<3;i++)
        {
                eeprom_write_byte((uint8_t*) 9+i, arr[i].playerScore);
        }

}
void button1(){
        if (pressedB1 == false)
        {

        if (b1==1)
        {
                b1=0;
        }else{
                b1 = 1;
        }
                switch(b1){
                        case 0: n = 0 ;
                        break;
                        case 1: n = 1 ;
                        break;
```

```
                }
        }
        pressedB1 = true;
}

void button2(){
        if(pressedB2 == false){
        if(n == 0){
                highScore1 = 10000;
        }else{
                n = 0;
        }
        }
        b1 =0;
        pressedB2 = true;
        gameReset = true;
}

// initialize timer, interrupt and variable
void timer0_init()
{
        // set up timer with prescaler = 256
        TCCR0 |= (1 << CS02);

        // initialize counter
        TCNT0 = 0;

        // enable overflow interrupt
        TIMSK |= (1 << TOIE0);

        // enable global interrupts


        // initialize overflow counter variable
        timerOverflowCount = 0;
}
void resetVals(){
        penaltyDelay = -6;
        penaltyTime = 0;
        gameReset = false;

}
ISR(TIMER0_OVF_vect)
```

```
{
	// keep a track of number of overflows
	timerOverflowCount++;
	if (timerOverflowCount >= 122)
	{
		timerOverflowCount = 0;
		sec++;
	}
}


void timeStart(){
	start = sec;
}
void timeEnd(){

		highScore1=currentTime;
		arr[3].playerScore = highScore1;
		clear_display();
		setDDRAM(0x80);
		write_string("Write your name");
		if(arr[2].playerScore>arr[3].playerScore){
			chooseLetter();
		}
		strcpy(arr[3].player,newName);
}

void chooseLetter(){

	iForChoosing = 0;
	iForLetter = 0x41;
	while (choosingLetters == true)
	{
		if(BUTTON2_PRESSED){
		while(BUTTON2_PRESSED){
		button2ForLetters();
	}
		}
		if(BUTTON1_PRESSED){
		while(BUTTON1_PRESSED){
		button1ForLetters();
	}
		}
```

```
        newName[iForChoosing]=iForLetter;
        if (iForChoosing == 0)
        {
                setDDRAM(0xc0);
                write_char(newName[iForChoosing]);
        }else if (iForChoosing == 1)
        {
                setDDRAM(0xc1);
                write_char(newName[iForChoosing]);
        }else if (iForChoosing == 2)
        {
                setDDRAM(0xc2);
                write_char(newName[iForChoosing]);
        }else{
                choosingLetters = false;
        }
        pressedB1L = false;
        pressedB2L = false;
        }


}

void resetScoreboard(){
        strcpy(arr[0].player,"NUL");
        arr[0].playerScore = 255;

        strcpy(arr[1].player,"NUL");
        arr[1].playerScore = 255;

        strcpy(arr[2].player,"NUL");
        arr[2].playerScore = 255;

        strcpy(arr[3].player,"NUL");
        arr[3].playerScore = 255;
        scoreWritten = false;
}

void button1ForLetters(){
        if(pressedB1L == false){
        iForChoosing++;
        }
        pressedB1L = true;
```

```
        iForLetter = 0x41;
}

void button2ForLetters(){
                if(pressedB2L == false){
                if (iForLetter<0x5A)
                {
                        iForLetter++;
                }else{
                iForLetter = 0x41;
           }


        }
        pressedB2L = true;
}
void penalty(){
        if (penaltyDelay+6 < currentTime)
        {
                penaltyTime = penaltyTime +5;
                penaltyDelay = currentTime;
        }
}

void displayOn(){
        write_cmd(0b00001100);
        _delay_ms(10);
}

void functionSet(){
        write_cmd(0b00111100);
        _delay_us(60);
}



void displayOff(){
        write_cmd(0b00001000);

}

void setDDRAM(char c){
        write_cmd(c);
```

```
	_delay_us(60);
}

void write_cmd(char c){
	RS_L;
	RW_L;
	E_H;
	PORTD = c;
	E_L;
	E_H;
}

void write_char(char c){
	RS_H;
	RW_L;
	E_H;
	PORTD =c;
	E_L;
	E_H;
	_delay_us(60);
}

void read_cmd(char c){
	RS_H;
	RW_H;
	E_H;
	PORTD = c;
	E_L;
	E_H;
}

void write_string(char string[]){
	int i = 0;
	while(string[i] !='\0'){
		write_char(string[i]);
		i++;
	}
}

void time_keep(){

}
```

```
void clear_display(){
        write_cmd(0b00000001);
        _delay_ms(4);
}
```