```c
/*
 * Kod_0.c
 *
 * Created: 2016-04-05 09:22:33
 *  Author: Oskar Strömberg & Olof Svanemur
 */

#include <util/delay.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>

//Permament datalagring
uint8_t highscore = 1;
#define HIGHSCORE_ADDRESS 0x00

//Temporär datalagring
int orderNbr = 0;
int level =  0;
int orderSize = 1;
int order[100];
char success = 1;

//Program
char gameRunning = 0;
int click = 0;
int buttonClicks = 0;

void SetEHigh(){
		PORTC = PORTC | 0b01000000;
}

void SetELow(){
		PORTC = PORTC & 0b10111111;
}

void SetRSHigh(){
		PORTC = PORTC | 0b00000001;
}

void SetRSLow(){
		PORTC = PORTC & 0b11111110;
}

void SetRWHigh(){
		PORTC = PORTC | 0b00000010;
}

void SetRWLow(){
		PORTC = PORTC & 0b11111101;
}

void SetAllHigh(){
		SetEHigh();
		SetRSHigh();
		SetRWHigh();
}
```

```
void SetAllLow(){
            SetELow();
            SetRSLow();
            SetRWLow();
}

//Set up för displayen
void SetUpDisplay(){
            /*
            char c = 0b00000011;
            _delay_ms(15);
            WriteCmd(c);
            _delay_ms(5);
            WriteCmd(c);
            _delay_ms(1);
            WriteCmd(c);
            _delay_ms(5);
            c = 0b00000010;
            WriteCmd(c);
            */

            char c = 0b00000010;
            _delay_ms(15);
            ExecuteCmd(c);
            c = 0b00101000; //0b00101000
            _delay_ms(5);
            ExecuteCmd(c);


            //c = 0b00000010;
            //WriteCmd(c);
            //c = 0b00100000;
            //ExecuteCmd(c);
            //c = 0b00001110;
            //ExecuteCmd(c);
            //c = 0b00000110;
            //ExecuteCmd(c);
}

//Tömmer skärmen
void ClearDisplay(){
            ExecuteCmd(0b00000001);
            _delay_ms(5);
            //SetUpDisplay();
            _delay_ms(5);
}

//Startar skärmen, går till första menyn med visat start av spel och nuvarande rekord
void StartDisplay(){
            char c = 0b00001111;
            ExecuteCmd(c);
            ClearDisplay();
}

//Stänger av skärmen
void StopDisplay(){
            ClearDisplay();
            ExecuteCmd(0b00001000);
```

```
}

//Skicka in kommando som ska utföras
void ExecuteCmd(char c){
            char c1 = (PORTB & 0xF0)|((c>>4)&0x0F); //Övre fyra bitarna
            WriteCmd(c1);
            char c2 = (PORTB & 0xF0)|(c&0x0F); //Nedre fyra bitarna
            WriteCmd(c2);
}

//Skriv kommando för att ställa in displayen.
void WriteCmd(char c){
            //Obs viktig sekvens
            SetRSHigh();
            SetRWHigh();
            SetELow();
            _delay_ms(1);
            SetRSLow();
            SetRWLow();
            _delay_ms(1);
            SetEHigh();
            _delay_ms(1);
            PORTB = c;
            SetELow();
            _delay_ms(1);
            SetRSHigh();
            SetRWHigh();
            _delay_ms(1);
            SetEHigh();
}

//Skriv data till RAM, för att skriva på displayen.
void WriteRam(char c){
            SetAllHigh();
            PORTB = c;
            SetRWLow();
            _delay_ms(1);
            SetELow();
            _delay_ms(1);
            SetEHigh();
            SetAllHigh();
}

//Skriv tecken på displayen.
void WriteChar(char c){
            char c1 = (PORTB & 0xF0)|((c>>4) & 0x0F);
            WriteRam(c1);
            char c2 = (PORTB & 0xF0)|(c & 0x0F);
            WriteRam(c2);
}

//Slumpar fram ordningen som dioderna ska blinka samt vilken färg som ska blinka
void Randomize()
{
            int place = 0;
            while(place < level){
                        order[place] = rand() % 16;
                        place++;
```

```
            }
}

void FirstBlueBlink(){
            PORTA = 0b00000001;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void FirstGreenBlink(){
            PORTA = 0b00000010;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void FirstRedBlink(){
            PORTA = 0b00000100;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void FirstWhiteBlink(){
            PORTA = 0b00000111;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void SecondBlueBlink(){
            PORTA = 0b00001000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void SecondGreenBlink(){
            PORTA = 0b00010000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void SecondRedBlink(){
            PORTA = 0b00100000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void SecondWhiteBlink(){
            PORTA = 0b00111000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}
```

```
void ThirdBlueBlink(){
            PORTA = 0b01000000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void ThirdGreenBlink(){
            PORTA = 0b10000000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            _delay_ms(500);
}

void ThirdRedBlink(){
            PORTD = 0b01000000;
            _delay_ms(1000);
            PORTD = 0b00000000;
            _delay_ms(500);
}

void ThirdWhiteBlink(){
            PORTA = 0b11000000;
            PORTD = 0b01000000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(500);
}

void FourthBlueBlink(){
            PORTD = 0b00100000;
            _delay_ms(1000);
            PORTD = 0b00000000;
            _delay_ms(500);
}

void FourthGreenBlink(){
            PORTB = 0b10000000;
            _delay_ms(1000);
            PORTB = 0b00000000;
            _delay_ms(500);
}

void FourthRedBlink(){
            PORTB = 0b01000000;
            _delay_ms(1000);
            PORTB = 0b00000000;
            _delay_ms(500);
}

void FourthWhiteBlink(){
            PORTB = 0b11000000;
            PORTD = 0b00100000;
            _delay_ms(1000);
            PORTB = 0b00000000;
            PORTD = 0b00000000;
```

```
            _delay_ms(500);
}

void AllBlink(){
            PORTA = 0b00010001;
            PORTB = 0b11000000;
            PORTD = 0b01100000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
            PORTA = 0b11100010;
            PORTB = 0b00000000;
            PORTD = 0b01100000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
            PORTA = 0b01111100;
            PORTB = 0b10000000;
            PORTD = 0b00000000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
            PORTA = 0b10001111;
            PORTB = 0b01000000;
            PORTD = 0b00000000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
}

void AllBlueBlink(){
            PORTA = 0b01001001;
            PORTB = 0b00000000;
            PORTD = 0b00100000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
}

void AllGreenBlink(){
            PORTA = 0b10010010;
            PORTB = 0b10000000;
            PORTD = 0b00000000;
            _delay_ms(500);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
```

```
}

void AllRedBlink(){
            PORTA = 0b00100100;
            PORTB = 0b01000000;
            PORTD = 0b01000000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
}

//Blinkar alla dioder vitt en gång
void AllWhiteBlink(){
            PORTA = 0b11111111;
            PORTB = 0b11000000;
            PORTD = 0b01100000;
            _delay_ms(1000);
            PORTA = 0b00000000;
            PORTB = 0b00000000;
            PORTD = 0b00000000;
            _delay_ms(100);
}

//Blinkar motsvarande diod grönt en gång
void GreenBlink(int button){
            switch(button){
                        case 0:
                        FirstGreenBlink();
                        break;

                        case 1:
                        SecondGreenBlink();
                        break;

                        case 2:
                        ThirdGreenBlink();
                        break;

                        case 3:
                        FourthGreenBlink();
                        break;

                        case 4:
                        AllGreenBlink();
                        break;
            }
}

//Blinkar motsvarande diod rött en gång
void RedBlink(int button){
            switch(button){
                        case 0:
                        FirstRedBlink();
                        break;

                        case 1:
```

```
                        SecondRedBlink();
                        break;

                        case 2:
                        ThirdRedBlink();
                        break;

                        case 3:
                        FourthRedBlink();
                        break;

                        case 4:
                        AllRedBlink();
                        break;
            }
}

//5 röda blinkningar med alla dioder vid förlust
void LoseBlink(){
            int i = 0;
            while(i < 5){
                        AllRedBlink();
                        i++;
            }
}

//5 gröna blinkningar med alla dioder vid vunnen nivå
void WinBlink(){
            int i = 0;
            while(i < 5){
                        AllGreenBlink();
                        i++;
            }
}

//Blinkar den framslumpade sekvensen
void BlinkSequence(){
            int i = 0;
            orderNbr = 0;
            while(i < orderSize){
                        i++;
                        switch(order[orderNbr]){
                                    case 0:
                                    FirstBlueBlink();
                                    break;

                                    case 1:
                                    FirstGreenBlink();
                                    break;

                                    case 2:
                                    FirstRedBlink();
                                    break;

                                    case 3:
                                    FirstWhiteBlink();
                                    break;
```

```
                                        case 4:
                                        SecondBlueBlink();
                                        break;

                                        case 5:
                                        SecondGreenBlink();
                                        break;

                                        case 6:
                                        SecondRedBlink();
                                        break;

                                        case 7:
                                        SecondWhiteBlink();
                                        break;

                                        case 8:
                                        ThirdBlueBlink();
                                        break;

                                        case 9:
                                        ThirdGreenBlink();
                                        break;

                                        case 10:
                                        ThirdRedBlink();
                                        break;

                                        case 11:
                                        ThirdWhiteBlink();
                                        break;

                                        case 12:
                                        FourthBlueBlink();
                                        break;

                                        case 13:
                                        FourthGreenBlink();
                                        break;

                                        case 14:
                                        FourthRedBlink();
                                        break;

                                        case 15:
                                        FourthWhiteBlink();
                                        break;
                                }
                                orderNbr++;
                        }
}

//Jämför tryckt knapp med vilken diod som lystes upp på nuvarande plats.
void ButtonCompare(int button){
                buttonClicks++;
                int diodNbr = order[orderNbr] / 4;
                if(button != diodNbr){
                                RedBlink(button);
```

```
                        success = 0;
                        LoseBlink();
                } else {
                        GreenBlink(button);
                }
                orderNbr++;
}

//Nollställer nivån inför nytt spel.
void SetUpGame(){
                gameRunning = 1;
                success = 1;
                orderNbr = 0;
                orderSize = 1;
                level = 0;
}

//Loop som väntar medans spelaren trycker på knapparna, avbryter då fel knapp trycks
// eller hela sekvensen har tryckts korrekt
void CheckButtons(){
                orderNbr = 0;
                buttonClicks = 0;
                while(success && (buttonClicks <= level)){
                        _delay_ms(10);
                }
}

//Startar en ny spelomgång
void Play(){
                SetUpGame();
                //SetPlayDisplay();
                AllBlink();
                while(success){
                        _delay_ms(500);
                        orderNbr = 0;
                        level++;
                        orderSize++;
                        SetPlayDisplay();
                        _delay_ms(2000);
                        Randomize();
                        BlinkSequence();
                        CheckButtons();
                        if(success){
                                WinBlink();
                        }
                }
                gameRunning = 0;
                if(level > highscore){
                        highscore = level;
                        eeprom_write_word((uint8_t*)HIGHSCORE_ADDRESS,highscore);
                }
                level = 0;
                SetMenuDisplay();
}

//Går till menyn.
void Menu(){
                gameRunning = 0;
```

```
            SetMenuDisplay();

}

//Skriver ut Avsluta
void WriteAvsluta(char c){
            ClearDisplay();
            ExecuteCmd(c); //0b10000001
            //WriteRam(0b00000000);
            WriteChar('A'); //A
            WriteChar('v'); //v
            WriteChar('s'); //s
            WriteChar('l'); //l
            WriteChar('u'); //u
            WriteChar('t'); //t
            WriteChar('a'); //a
}

//Skriver ut Start
void WriteStart(){
            ClearDisplay();
            //ExecuteCmd(0x80); //0b10000001
            //WriteRam(0b00000000);
            WriteChar('S');
            WriteChar('t');
            WriteChar('a');
            WriteChar('r');
            WriteChar('t');
            //WriteChar(0b00100111);
            //WriteChar(0b01000000);
}

//Skriver ut rekord
void WriteHighscore(char k){
            ExecuteCmd(k); //0b10000011, 0x94
            //WriteRam(0b00000000);
            WriteChar('R');
            WriteChar('e');
            WriteChar('k');
            WriteChar('o');
            WriteChar('r');
            WriteChar('d');
            WriteChar(':');

            //Beräkning av ascii för rekord
            int i = highscore/10;
            char c;
            if(i > 0){
                        char b = 48 + i;
                        WriteChar(b);
                        int rest = highscore%10;
                        c = 48 + rest;
            } else {
                        c = 48+highscore;
            }

            WriteChar(c);
}
```

```
//Skriver ut nivå
void WriteLevel(char k){
		ExecuteCmd(k); //0b10000001, 0xD4
		//WriteRam(0b00000000);
		WriteChar('N');
		WriteChar('i');
		WriteChar('v');
		WriteChar('a');
		WriteChar(':');

		//Beräkning av ascii för nivå
		int i = level/10;
		char c;
		if(i > 0){
			char b = 48 + i;
			WriteChar(b);
			int rest = level%10;
			c = 48 + rest;
		} else {
			c = 48+level;
		}

		WriteChar(c);
}

//Visar Display för spelomgång
void SetPlayDisplay(){
		WriteAvsluta(0x82);
		WriteHighscore(0x96);
		WriteLevel(0xD6);
}

//Visar menyns display
void SetMenuDisplay(){
		WriteStart();
		WriteHighscore(0x94);
		WriteLevel(0xD4);
}

//Sparar highscore på processorns permanenta minne. //Överflödig, ta bort!
void SaveHighscore(){
		//Sida 20, datablad processor
}

void SetDataDirection(){
		DDRA = 0b11111111;
		DDRB = 0b11001111;
		DDRC = 0b01000011;
		DDRD = 0b01100000;
}

int main(void)
{
		SetDataDirection();
		SetUpDisplay();

		ClearDisplay();
```

```
		StartDisplay();


		PIND = 0b00000000;
		DDRD = 0b01100000;
		GICR = 0b01000000;
		MCUCR = 0b00000011;
		sei();

		highscore = eeprom_read_word((uint8_t*)HIGHSCORE_ADDRESS);
		if (highscore==0xFFFF){
			highscore=0;
		}

		Menu();

		while(1)
		{

		}
}

ISR(INT0_vect){

		if(click == 1){
			return;
		}

		click = 1;
		cli();

		_delay_ms(50);

		switch(PIND){
			case 0b00010100:
				if(gameRunning){
					ButtonCompare(0);
				}

			break;

			case 0b00001100:
				if(gameRunning){
					ButtonCompare(1);
				}

			break;

			case 0b00000110:
				if(gameRunning){
					ButtonCompare(2);
				}

			break;

			case 0b00000101:
				if(gameRunning){
					ButtonCompare(3);
```

```
                        }

                break;

                case 0b10000100:
                        AllBlueBlink();
                        if(gameRunning){
                                sei();
                                click = 0;
                                success = 0;
                        } else {
                                sei();
                                click = 0;
                                Play();
                        }

                break;
        }

        sei();

        click = 0;
}
```