Appendix B

```
/*
 * ArkadBasket.c
 *
 * Created: 2017-03-24 13:16:38
 * Author : ine13plo
 */

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
int anybuttonpressed=0;
int gamestarted=0;
int irsensor=0;
int points=0;
int boolint=0;
int boolint2=0;
int boolint3=0;
int greyButtonPressed=0;
int yellowButtonPressed=0;
int menustarted=0;
int arcademode=0;
int pointhit=0;
int arcadehighscore=0;
int normalhighscore=0;

void display_cmd(){
	PORTD= 0b10000000;			// RW, RS -> 0, E -> 1
	PORTD= 0b00000000;			// RW, RS -> 0, E -> 0
	_delay_ms(10);
}

void display_setup(){
	DDRA = 0b11111100;
	PORTA = 0b11111100;
	DDRB = 0b11111111;			// Sätter pinnarna för port B till ettor (ut)
	DDRD = 0b11100000;			// Sätter Pinne 19-21 till ettor
	_delay_ms(10);

	//Function Set
	PORTB= 0b00111000;			// Sätter interface datalängd, antal linjer etc. //ändrade här nu har vi fyra
rader
	display_cmd();
	_delay_ms(10);

	//Display on
	PORTB = 0b00001100;			// Inställningar för displayen
	display_cmd();
```

```
        _delay_ms(10);

        //Entry mode set
        PORTB = 0b00000110;
        display_cmd();
        _delay_ms(10);
}

void display_clear(){
        PORTB= 0b00000001;
        _delay_ms(10);
        display_cmd();
}

void write_string(char str[]) {
        int i = 0;
        while( str[i] != '\0') {            // \0 null character in ASCII
        PORTB = str[i];
        _delay_ms(10);
        PORTD= 0b10100000;                  // RS -> 1, E -> 1
        _delay_ms(10);
        PORTD= 0b00100000;                  // RS -> 1, E -> 0
        i++;
        }
}


/****************************************************************
 * Interrupt
 */

void setup_interrupt() {
                MCUCR = MCUCR | 0b00001111;
                GICR = GICR | 0b11000000;

                sei();
}

        ISR(INT1_vect){
                anybuttonpressed=1;
                if((menustarted==1)&&(gamestarted==0)){
                switch(PINA){
                        case 0b11111101:            //(Av någon anledning sätts PINA2-PINA7 till ettor när programmet
                                                    //startas även fast de inte används.)
                                _delay_ms(10);
                                greyButtonPressed=1;
                                break;
                        case 0b11111110:
                                _delay_ms(10);
                                yellowButtonPressed=1;
```

```
                              break;
                         }
                    }
               }

               ISR(INT0_vect){
               irsensor=1;
               }



               /************************************************************
 * KLOCKAN
 */
volatile int count;
double measured_time = 0.0;
double time_in_seconds =0.0;
double total_time_in_seconds=0.0;

volatile double count2;
double measured_time_2 = 0.0;
double time_in_seconds_2 =0.0;

//Overflow Interrupt (för klockan)
          ISR(TIMER0_OVF_vect) {   // TIMER0_OVF_vect
                    count2++;
                    if(gamestarted==1){
                              count++;            // Räknas upp vid varje interrupt. I det här fallet xxx per sekund
                    }
          }

void clock_setup() {
          TCCR0 = TCCR0 | 0b00000101; // Prescaler 1024   (Sida 81,83 i databladet.)
          TIMSK = TIMSK | 0b00000001; // Overflow interrupt aktivt.  (Sida 83 i databladet.)
}


void start_race(){
          count = 0;
          points=0;
}

void starttext(){
          count2 = 0;
          display_clear();
          PORTB = 0b11000000;                                   // Byt rad till rad 2
          _delay_ms(20);
          display_cmd();
          _delay_ms(10);
          write_string("     GET READY");
```

```
        while(count2/61.2745098<3){
        //Inget ska hända i tre sekunder
        }

    display_clear();
    count2 = 0;

    for (int i = 3; i>0 ; i--){
         PORTB = 0b11000000;                        // Byt rad till rad 2
         _delay_ms(20);
         display_cmd();
         _delay_ms(10);
         char str[5];
         sprintf(str, "%d", i);                 //Int till char
         write_string("        ");
         write_string(str);

        while(count2/61.2745098<1){
            //Inget ska hända i en sekund
        }
        count2 = 0;
        display_clear();
    }
}

void main(void){
    display_setup();

    display_clear();

    setup_interrupt();

    clock_setup();

    start_race();

    write_string("  ***************  ");

    // Byt rad till rad 2
    PORTB = 0b11000000;
    _delay_ms(20);
    display_cmd();
    _delay_ms(10);

    write_string("  *ARCADE BASKET*");

    // Byt rad till rad 3
    PORTB = 0b10010100;
    _delay_ms(20);
    display_cmd();
    _delay_ms(10);
    write_string("  ***************  ");

    while(count2/61.2745098<3){
            //Inget ska hända i tre sekunder
    }

    display_clear();
    write_string(" Press a button to");
    write_string("       start!");
_delay_ms(300);

    while (1) {

        if((anybuttonpressed==1)&&(gamestarted==0)&&(menustarted==0)){
            display_clear();
            menustarted=1;
            write_string("Arcade mode: YELLOW Normal mode: GREY");
        }

         if((yellowButtonPressed==1)&&(menustarted==1)&&(gamestarted==0)){
            display_clear();

            // Byt rad till pos 44
            PORTB = 0b11000100;
            _delay_ms(20);
            display_cmd();
            _delay_ms(10);
             write_string("Arcade mode");

             arcademode=1;
             starttext();
             gamestarted=1;
             anybuttonpressed=0;
             menustarted=0;
         }

         if((greyButtonPressed==1)&&(menustarted==1)&&(gamestarted==0)){
             display_clear();

              PORTB = 0b11000100;
              _delay_ms(20);
              display_cmd();
              _delay_ms(10);
              write_string("Normal mode");

             starttext();
             gamestarted=1;
              anybuttonpressed=0;
              menustarted=0;
```

```
 }

 if(irsensor==1){
            points++;
            irsensor=0;
            pointhit=1;
 }


measured_time=count;
time_in_seconds=measured_time/61.2745098 ;

if(gamestarted==1){
 if(total_time_in_seconds<30){                                   //Slutar skriva ut efter 30 s.
 if(time_in_seconds>=1){                                         //Vad som görs varje sekund
            if(arcademode==1){
                        if((points%3==0) && (pointhit==1)){
                                    total_time_in_seconds-=3;
                                    PORTB = 0b10010100;                     // Byt rad till rad 3
                                    _delay_ms(20);
                                    display_cmd();
                                    _delay_ms(10);
                                    write_string("Extra time!");
                                    _delay_ms(10000);
                                    _delay_ms(10000);
                                    pointhit=0;
                        }
            }

            display_clear();
            char str[5];
            sprintf(str, "%d", points);                     //Int till char
            write_string("Points: ");
            write_string(str);

            time_in_seconds=0.0;
            count=0.0;
            total_time_in_seconds ++;
            char time_str[2];
            dtostrf(total_time_in_seconds, 2, 0, time_str); // Skapar char-array av double (3e parametern
                                                            //anger antal decimaler)
            write_string(" Time: ");
            write_string(time_str);
           }
 }
}


 //Verkar onödigt att ha display_clear i egen metod, men det funkar inte annars.
 if(total_time_in_seconds>=30 && boolint3==0){
    display_clear();
    boolint3=1;
    boolint2=1;
}

if(boolint2==1 && boolint==0){
    display_clear();
    if((arcademode==1)&&(points>arcadehighscore)){

            arcadehighscore=points;

            for(int i=0;i<7;i++){
             PORTB = 0b11000000;                   // Byt rad till rad 2
             _delay_ms(20);
             display_cmd();
             _delay_ms(10);

             write_string("     HIGHSCORE!");
             _delay_ms(2700);
             display_clear();
             _delay_ms(2000);
            }
            display_clear();
        }

         if((arcademode==0)&&(points>normalhighscore)){

             normalhighscore=points;

             for(int i=0;i<7;i++){
                 PORTB = 0b11000000;               // Byt rad till rad 2
                 _delay_ms(20);
                 display_cmd();
                 _delay_ms(10);

                 write_string("     HIGHSCORE!");
                 _delay_ms(2700);
                 display_clear();
                 _delay_ms(2000);
             }
             display_clear();
         }

    char str[5];
    sprintf(str, "%d", points);          //Int till char
    write_string("  TOTAL POINTS: ");
    write_string(str);

    // Byt rad till rad 2
    PORTB = 0b11000000;
```

```
		_delay_ms(20);
		display_cmd();
		_delay_ms(10);
		write_string("    Highscore: ");

		if(arcademode==1){
			char str[3];
		   sprintf(str, "%d", arcadehighscore);			//Int till char
		   write_string(str);
		}

		if(arcademode==0){
			char str[3];
			sprintf(str, "%d", normalhighscore);			//Int till char
			write_string(str);
		}


		// Byt rad till rad 3
		PORTB = 0b10010110;
		_delay_ms(20);
		display_cmd();
		_delay_ms(10);
		write_string("Press any button");

		//byt till rad 4
		PORTB = 0b11011001;
		_delay_ms(20);
		display_cmd();
		_delay_ms(10);
		write_string("to restart");
		anybuttonpressed=0;
		boolint=1;
	}

	if((boolint==1)&&(anybuttonpressed==1)){
		greyButtonPressed=0;
		yellowButtonPressed=0;
		points=0;
		count=0;
		total_time_in_seconds=0.0;
		measured_time=0.0;
		time_in_seconds=0.0;
		boolint=0;
		boolint2=0;
		boolint3=0;
		arcademode=0;
		pointhit=0;
		menustarted=0;
		gamestarted=0;
```

```
                                        arcademode=0;
                                }
                        }
                }
```