Källkod

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#define red_diode PD6
#define green_diode PD5
#define E PD0
#define RW PD1
#define RS PD4
unsigned int password = 1;
unsigned int count = 0;
unsigned int larm_status = 0;
unsigned int alarmactive = 0;
unsigned int data_available;
unsigned int pinCounter = 0;
unsigned char pinCode[4] = {'1', '2', '3', '4'};
unsigned char pinCodeTry[4];
unsigned char val;
unsigned int charnbr = 0;
unsigned int new_input = 0;
unsigned int sec = 0;
unsigned int min = 15;
unsigned int hour = 10;
unsigned int clock = 49911;
int timevect[8];

        char read_keypad(char code){
            if(code == 0b00000000){
                return '0';
                }else if(code==0b00000100){
                return '1';
                }else if(code==0b00001000){
                return '2';
                }else if(code==0b00001100){
                return '3';
                }else if(code==0b00010000){
                return '4';
                }else if(code==0b00010100){
                return '5';
                }else if(code==0b00011000){
                return'6';
                }else if(code==0b00011100){
                return '7';
                }else if(code==0b00100000){
```

```
                return '8';
                }else if(code==0b00100100){
                return '9';
                }else if(code==0b00101000){
                return 'A';
                }else if(code==0b00101100){
                return 'B';
                }else if(code==0b00110000){
                return 'C';
                }else if(code==0b00110100){
                return 'D';
                }else if(code==0b00111000){
                return 'E';
                }else if(code==0b00111100){
                return 'F';
            }

        }
int main(void)
{
    INIT();
    _delay_ms(5);
    val=0xff;
    _delay_ms(5);
    start_Display();
    _delay_ms(5);
    set_Display();
    _delay_ms(5);
    clear_display();
    _delay_ms(5);
    write_welcome();

    while (1)
    {


        if(larm_status == 2){
            alarmtriggered();
        }

        while(new_input == 1){
            char input = read_keypad(val);


            if(input == 'A' ){
```

```
    int pincount = 0;
    while(pincount < 3){
        clear_display();
        write_activatelarm();
        new_input = 0;
        for(int k =0; k<4; k++){

            while(new_input == 0){
            }

            char pin = read_keypad(val);
            pinCodeTry[k] = pin;
            write_char('*');
            new_input = 0;
        }
        int wrong = 0;
        for(int i=0; i<4;i++){
            if(pinCodeTry[i] !=pinCode[i]){
            wrong = 1;
            }
        }
        if(wrong == 0){
            clear_display();
            alarm_ON();
            new_input = 0;
            pincount = 3;
        } else {
            pincount++;
            write_wrongcode();
            if(pincount == 3){
                clear_display();
                larm_status = 2;
                new_input =0;
            }
        }
    }
}
if(input == 'D'){
    int pincount = 0;
    while(pincount < 3){
        clear_display();
        write_deactivatelarm();
        new_input = 0;
        for(int k =0; k<4; k++){

                while(new_input == 0){
                }

            char pin = read_keypad(val);
            pinCodeTry[k] = pin;
            write_char('*');
        new_input = 0;
        }
        int wrong = 0;
        for(int i=0; i<4;i++){
            if(pinCodeTry[i] !=pinCode[i]){
                wrong = 1;
            }
        }
        if(wrong == 0){
            clear_display();
            alarm_OFF();
            new_input = 0;
            pincount = 3;
            } else {
            pincount++;
            write_wrongcode();
            _delay_ms(5000);
            if(pincount == 3){
                clear_display();
                larm_status = 2;
                new_input =0;
            }
        }
    }
}

if(input == 'C'){
    int pincount = 0;
    while(pincount < 3){
    new_input = 0;
        clear_display();
        write_oldcode();
        for(int k =0; k<4; k++){

                while(new_input == 0){
                }

            char pin = read_keypad(val);
            pinCodeTry[k] = pin;
            write_char('*');
            new_input = 0;
```

```
                    }
                    int wrong = 0;

                    for(int i=0; i<4;i++){
                        if(pinCodeTry[i] !=pinCode[i]){
                            wrong = 1;
                        }
                    }
                    if(wrong == 0){
                        clear_display();
                        write_newcode();
                        new_input = 0;
                        for(int i=0;i<4;i++){

                            while(new_input == 0){
                            }


                            char newPin = read_keypad(val);
                            pinCode[i] = newPin;
                            write_char(newPin);
                            new_input = 0;
                        }
                        new_input = 0;
                        pincount = 3;
                        clear_display();
                    } else {
                        pincount++;
                        write_wrongcode();
                        _delay_ms(5000);
                        if(pincount == 3){
                            clear_display();
                            larm_status = 2;
                            new_input =0;
                        }
                    }
                }
            }
        }
    }
}

    ISR(TIMER1_OVF_vect) {
            sec++;
            if(sec == 60){
                min++;
                sec = 0;
                if(min == 60){
                    hour++;
                    min = 0;
                    if(hour == 24){
                        hour = 0;
                    }
                }
            }
    }

    ISR(INT0_vect){
        val = PINA & 0b00111100;
        new_input = 1;
    }

    ISR(INT1_vect){
        if(larm_status==1){
            larm_status = 2;
        }
    }




    void INIT(){ // Sätter initialtillstånden
        DDRB = 0b11111111; //vi sätter databussen ut
        DDRA = 0b00000000; //vi sätter a-porten in
        DDRD = 0b11110011; //vi sätter RS. RW, E ut
        DDRC = 0b00000000; // C-porten in
        PORTD = 0b00001100; //vi sätter RS till 0 och E till 0 även interupt 1
        GICR = 0b11000000;
        MCUCR = 0b0001111;
        TCCR1B =0b00000101; //inställning för klockan (vet ej om bit 6 ska vara 0
eller 1)
        TCNT1 = clock;
        TIMSK = 0b00000100; // Ska sätta hur ofta interuptet skjer, hur??
        sei();


    }

    void display_int(int time){
        int time1 = time/10;
        itoa(time1, timevect, 10);
        write_char(timevect[0]);
        int time2 = time%10;
        time1 = itoa(time2, timevect, 10);
        write_char(timevect[0]);
}

void showTime(int hour, int min, int sec){
        display_int(hour);
        write_char(':');
        display_int(min);
        write_char(':');
        display_int(sec);

}

void clear_display(){
        _delay_ms(10);

        PORTB = 0b00000001;
        PORTD |= _BV(E);
        PORTD &= ~_BV(E);
        _delay_ms(10);
}
void change_curser(){
        char location = 0x40;
        if(location < 0b01111111){
        PORTB = 0b10000000 | location;
        PORTD |= _BV(E);
        PORTD &= ~_BV(E);
        }
        }

void set_Display(){
        PORTB = 0b00111100; // Ska sista två vara nollor? - i datablad
        PORTD |= _BV(E);
        PORTD &= ~_BV(E);
}

void start_Display(){ // Används även denna för off? Behövs off?
        PORTB = 0b00001111;
        PORTD |= _BV(E); // Sätter E till 1
        PORTD &= ~_BV(E); // Sätter E till 0
}

void write_char(char ch){
        _delay_ms(5);
        PORTD |= _BV(RS); // Sätter RS hög
        _delay_ms(5);
        PORTB = ch;
        _delay_ms(5);
        PORTD |= _BV(E); // Sätter E hög
        _delay_ms(5);
        PORTD &= ~_BV(E); // Sätter E låg
        _delay_ms(5);
        PORTD &= ~_BV(RS); // Sätter rs låg
}

void write_int(int i){
        _delay_ms(5);
        PORTD |= _BV(RS); // Sätter RS hög
        _delay_ms(5);
        PORTB = i;
        _delay_ms(5);
        PORTD |= _BV(E); // Sätter E hög
        _delay_ms(5);
        PORTD &= ~_BV(E); // Sätter E låg
        _delay_ms(5);
        PORTD &= ~_BV(RS); // Sätter rs låg
}

void write_cmd(char cmd){
        _delay_ms(5);
        PORTB = cmd;
        PORTD &= ~ _BV(PD4); // Sätter RS låg
        PORTD &= ~ _BV(PD0); // Sätter E låg
        PORTD |= _BV(PD4); // Sätter RS hög
        PORTD |= _BV(PD0); // Sätter E hög
}

void write_AlarmON(){
        clear_display();
        _delay_ms(5);
        write_char('A');
        write_char('L');
        write_char('A');
        write_char('R');
        write_char('M');
        write_char(' ');
        write_char('O');
        write_char('N');
}

void write_AlarmOFF(){
        clear_display();
        _delay_ms(5);
        write_char('A');
        write_char('L');
        write_char('A');
        write_char('R');
        write_char('M');
        write_char(' ');
        write_char('O');
        write_char('F');
        write_char('F');
}

void write_wrongcode(){
        clear_display();
        _delay_ms(5);
        write_char('W');
        write_char('R');
        write_char('O');
        write_char('N');
        write_char('G');
        write_char(' ');
        write_char('C');
        write_char('O');
        write_char('D');
        write_char('E');
}



        void write_oldcode(){
            clear_display();
            _delay_ms(5);
            write_char('O');
            write_char('L');
            write_char('D');
            write_char(' ');
            write_char('C');
            write_char('O');
            write_char('D');
            write_char('E');
            write_char(':');
            change_curser();
        }
        void write_newcode(){
            clear_display();
            _delay_ms(5);
            write_char('N');
            write_char('E');
            write_char('W');
            write_char(' ');
            write_char('C');
            write_char('O');
            write_char('D');
            write_char('E');
            write_char(':');
            change_curser();
        }
        void write_welcome(){
            clear_display();
            _delay_ms(5);
            write_char('W');
            write_char('E');
            write_char('L');
            write_char('C');
            write_char('O');
            write_char('M');
            write_char('E');
            write_char(' ');
            write_char('T');
            write_char('O');
            change_curser();
            write_char('L');
            write_char('A');
            write_char('R');
            write_char('M');
            write_char(' ');
            write_char('M');
            write_char('A');
            write_char('S');
            write_char('T');
            write_char('E');
            write_char('R');
        }
        void write_activatelarm(){
            clear_display();
            _delay_ms(5);
            write_char('A');
            write_char('C');
            write_char('T');
            write_char('I');
            write_char('V');
            write_char('A');
            write_char('T');
            write_char('E');
            write_char(' ');
            write_char('L');
            write_char('A');
            write_char('R');
            write_char('M');
            write_char(':');
            change_curser();
        }
        void write_deactivatelarm(){
            clear_display();
            _delay_ms(5);
            write_char('D');
            write_char('E');
            write_char('A');
            write_char('C');
            write_char('T');
            write_char('I');
            write_char('V');
            write_char('A');
            write_char('T');
            write_char('E');
            write_char(' ');
            write_char('L');
            write_char('A');
            write_char('R');
            write_char('M');
            write_char(':');
            change_curser();
        }

        void triggered_at(){
            clear_display();
            _delay_ms(5);
            write_char('T');
            write_char('R');
            write_char('I');
            write_char('G');
            write_char('G');
            write_char('E');
            write_char('R');
            write_char('E');
            write_char('D');
            write_char(' ');
```

```
            write_char('A');
            write_char('T');
            write_char(':');
            change_curser();
            _delay_ms(5);
        }

// LED
    void greenDiode_ON(){
        PORTD |= _BV(green_diode);
        PORTD &= ~_BV(PD0);
        PORTD |= _BV(PD0);
    }

        void redDiode_ON(){
            PORTD |= _BV(red_diode);
            PORTD &= ~_BV(PD0);
            PORTD |= _BV(PD0);
        }

        void greenDiode_OFF(){
            PORTD &= ~ _BV(green_diode);
            PORTD &= ~_BV(PD0);
            PORTD |= _BV(PD0);
        }

        void redDiode_OFF(){
            PORTD &= ~ _BV(red_diode);
            PORTD &= ~_BV(PD0);
            PORTD |= _BV(PD0);
        }
        void summer_ON(){
            PORTD |= _BV(PD7);
            PORTD &= ~_BV(E);
            PORTD |= _BV(E);
        }
        void summer_OFF(){
            PORTD &= ~ _BV(PD7);
            PORTD &= ~_BV(PD0);
            PORTD |= _BV(PD0);
            }

    void alarm_ON(){
        larm_status = 1;
        greenDiode_OFF();
        redDiode_ON();
        write_AlarmON();
}

void alarm_OFF(){
        redDiode_OFF();
        greenDiode_ON();

        larm_status = 0;
        summer_OFF();
        write_AlarmOFF();
}

void alarmtriggered(){
        clear_display();
        triggered_at();
        showTime(hour, min,sec);
while(read_keypad(val) != 'D'){
        summer_ON();
                redDiode_ON();
                _delay_ms(1000);
                summer_OFF();
        redDiode_OFF();
                _delay_ms(1000);
        }
        if(larm_status == 2){
                summer_ON();
                redDiode_ON();
        }
}
```