

```

/*
 * GccApplication1.c
 *
 * Created: 2018-04-20 15:19:59
 * Author : ine15jlu
 */

#define F_CPU 8000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#define greenDiode PA6
#define redDiode PA7
#define summer PC6
#define E_D PD6
#define RW_D PD1
#define RS_D PD0
#define RD_C PD4
#define WR_C PD5
#define ALE_C PD7
#define CS_C PC7

char sensor1 = 0;
char sensor2 = 0;
uint16_t temp1 = 0;
uint16_t temp2 = 0;
unsigned int alarmStatus = 0;
unsigned int newInput = 0;
char val;
char pinCode[4];
char pinCompare[4];
unsigned int wrongPin = 0;
unsigned int sec = 0;
unsigned int min = 0;
unsigned int hour = 0;
unsigned int triggerIR1 = 0;
unsigned int triggerIR2 = 0;
unsigned int triggerTemp1 = 0;
unsigned int triggerTemp2 = 0;
unsigned int secTrig = 0;
unsigned int minTrig = 0;
unsigned int hourTrig = 0;
volatile uint16_t TimerOverFlowCount;

/* Instructions for reading the keypad */
char readKeypad(char code){
    if(code == 0b00000000){
        return '0';
    }else if(code==0b00000001){
        return '1';
    }else if(code==0b00000010){
        return '2';
    }else if(code==0b00000011){

```

```

        return '3';
    }else if(code==0b00000100){
        return '4';
    }else if(code==0b00000101){
        return '5';
    }else if(code==0b00000110){
        return '6';
    }else if(code==0b00000111){
        return '7';
    }else if(code==0b00001000){
        return '8';
    }else if(code==0b00001001){
        return '9';
    }else if(code==0b00001010){
        return 'A';
    }else if(code==0b00001011){
        return 'B';
    }else if(code==0b00001100){
        return 'C';
    }else if(code==0b00001101){
        return 'D';
    }else if(code==0b00001110){
        return 'E';
    }else if(code==0b00001111){
        return 'F';
    }
}

ISR(TIMEROVVF_vect)
{
    //each time the timer0 overflows, this ISR will be executed to indicate timer0
    overflow
    //Increment the counter value by this operation.
    TimerOverflowCount++;

    if(TimerOverflowCount>= 33250)
    {
        sec++;
        if (sec == 60) {
            min++;
            sec = 0;
        }
        if (min == 60) {
            hour++;
            min = 0;
        }
        if (hour == 24) {
            hour = 0;
        }
        TimerOverflowCount=0; //reset the timerOverflow count back to
0.
    }
}

```

```

int main(void)
{
    INIT();
    timerINIT();
    ADMUX |= 1<<REFS0;
    ADCSRA |= (1<<ADEN) | (0<<ADPS2) | (1<<ADPS1) | (0<<ADPS0);
    sei();
    val = 0xff;
    startDisplay();
    setDisplay();
    clearDisplay();
    writeWelcome();
    _delay_ms(2000);
    writeSelect();

    while (1)
    {
        while(larmStatus){
            alarmActive();
        }

        while(newInput == 1) {
            char input = readKeypad(val);
            newInput = 0;
            if (input == 'A') {
                functionA();
            }
            if (input == 'B') {
                functionB();
            }
            if (input == 'C') {
                functionC();
            }
            if (input == 'D') {
                functionD();
            }
        }
    }
}

ISR(INT0_vect) {
    val = PINA & 0b00001111;
    newInput = 1;
}

ISR(INT1_vect) {
    sec++;
    if (sec == 60) {
        min++;
        sec = 0;
    }
    if (min == 60) {
        hour++;
    }
}

```

```

        min = 0;
    }
    if (hour == 24) {
        hour = 0;
    }
}

/*Sätter initialtillstånden*/
void INIT(){
    DDRB = 0b11111111; // vi sätter databussen ut
    DDRA = 0b11000000; // vi sätter A6, A7 ut, resten in
    DDRD = 0b11110011; // vi sätter RS. RW, E ut
    DDRC = 0b11000000; // C6, C7 ut, resten in
    PORTD = 0b00001100; // vi sätter RS till 0 och E till 0 även interupt 1
    GICR = 0b11000000;
    MCUCR = 0b00011111;
}

void timerINIT(){
    TCNT0 = 0;
    TimerOverFlowCount= 0;

    //Enable the Timer0 overflow interrupt flag
    TIMSK|=(1<<TOIE0);

    //Start the Timer0
    TCCR0|= (1<<CS00); //since we don't want any prescaling.
}

void rtcINIT() {
    PORTC &= ~_BV(CS_C);
    PORTD |= _BV(WR_C);
    PORTD |= _BV(RD_C);

    PORTB = 0b00010001;

    PORTD |= _BV(ALE_C);
    _delay_us(5);
    PORTD &= ~_BV(ALE_C);

    PORTB = 0b00011100;

    PORTD &= ~_BV(WR_C);
    _delay_us(5);
    PORTD |= _BV(WR_C);

    PORTB = 0b00010000;

    PORTD |= _BV(ALE_C);
    _delay_us(5);
    PORTD &= ~_BV(ALE_C);

    PORTB = 0b00001000;
}

```

```

        PORTD &= ~_BV(WR_C);
        _delay_us(5);
        PORTD |= _BV(WR_C);
    }

void waitForInput(){
    while(newInput == 0){
    }
}

void greenDiodeOn(){
    PORTA |= _BV(greenDiode);
}

void greenDiodeOff(){
    PORTA &= ~_BV(greenDiode);
}

void redDiodeOn(){
    PORTA |= _BV(redDiode);
}

void redDiodeOff(){
    PORTA &= ~_BV(redDiode);
}

void summerOn(){
    PORTC |= _BV(summer);
}

void summerOff(){
    PORTC &= ~_BV(summer);
}

uint16_t analogReadTemp1(){
    ADMUX = 0b11000100;
    _delay_ms(20);
    ADCSRA|=(1<<ADSC);
    while ( !(ADCSRA & (1<<ADIF)));
    ADCSRA|=(1<<ADIF);
    return (ADC);
}

uint16_t analogReadTemp2(){
    ADMUX = 0b11000101;
    _delay_ms(20);
    ADCSRA|=(1<<ADSC);
    while ( !(ADCSRA & (1<<ADIF)));
    ADCSRA|=(1<<ADIF);
    return (ADC);
}

void setDisplay(){
    PORTB = 0b00000010;
}

```

```

        PORTD |= _BV(E_D);
        PORTD &= ~_BV(E_D);
        _delay_us(100);
        PORTB = 0b00111100;
        PORTD |= _BV(E_D);
        PORTD &= ~_BV(E_D);
        _delay_us(100);
    }

void startDisplay(){
    _delay_ms(100);
    PORTB = 0b00001101;
    PORTD |= _BV(E_D);
    PORTD &= ~_BV(E_D);
}

void clearDisplay(){
    PORTB = 0b00000001;
    PORTD |= _BV(E_D);
    PORTD &= ~_BV(E_D);
    _delay_ms(2);
}

void changeCursor(){
    char location = 0x40;
    if(location < 0b01111111){
        PORTB = 0b10000000 | location;
        PORTD |= _BV(E_D);
        PORTD &= ~_BV(E_D);
        _delay_us(100);
    }
}

void writeChar(char ch){
    PORTD |= _BV(RS_D); // Sätter RS hög
    PORTB = ch;
    _delay_us(100);
    PORTD |= _BV(E_D); // Sätter E hög
    _delay_us(100);
    PORTD &= ~_BV(E_D); // Sätter E låg
    PORTD &= ~_BV(RS_D); // Sätter RS låg
    _delay_us(100);
}

void writeWelcome(){
    clearDisplay();
    writeChar('W');
    writeChar('E');
    writeChar('L');
    writeChar('C');
    writeChar('O');
    writeChar('M');
    writeChar('E');
    writeChar('!');
}

```

```
}  
  
void writeActivate(){  
    clearDisplay();  
    writeChar('A');  
    writeChar('C');  
    writeChar('T');  
    writeChar('I');  
    writeChar('V');  
    writeChar('A');  
    writeChar('T');  
    writeChar('E');  
    writeChar(' ');  
    writeChar('A');  
    writeChar('L');  
    writeChar('A');  
    writeChar('R');  
    writeChar('M');  
    writeChar(',');  
}
```

```
void writeWrongCode(){  
    clearDisplay();  
    writeChar('W');  
    writeChar('R');  
    writeChar('O');  
    writeChar('N');  
    writeChar('G');  
    writeChar(' ');  
    writeChar('P');  
    writeChar('I');  
    writeChar('N');  
    writeChar('!');  
}
```

```
void writeActivated(){  
    clearDisplay();  
    writeChar('A');  
    writeChar('C');  
    writeChar('T');  
    writeChar('I');  
    writeChar('V');  
    writeChar('A');  
    writeChar('T');  
    writeChar('E');  
    writeChar('D');  
}
```

```
void writeDeactivated(){  
    clearDisplay();  
    writeChar('D');  
    writeChar('E');  
    writeChar('A');  
    writeChar('C');
```

```
        writeChar('T');
        writeChar('I');
        writeChar('V');
        writeChar('A');
        writeChar('T');
        writeChar('E');
        writeChar('D');
    }
```

```
void writeEnterPin(){
    clearDisplay();
    writeChar('E');
    writeChar('N');
    writeChar('T');
    writeChar('E');
    writeChar('R');
    writeChar(' ');
    writeChar('P');
    writeChar('I');
    writeChar('N');
    writeChar(':');
    writeChar(' ');
}
```

```
void writeSelect(){
    clearDisplay();
    writeChar('S');
    writeChar('E');
    writeChar('L');
    writeChar('E');
    writeChar('C');
    writeChar('T');
    writeChar(' ');
    writeChar('F');
    writeChar('U');
    writeChar('N');
    writeChar('C');
    writeChar('T');
    writeChar('I');
    writeChar('O');
    writeChar('N');
    writeChar(':');
    changeCursor();
    writeChar('A');
    writeChar(',');
    writeChar(' ');
    writeChar('B');
    writeChar(',');
    writeChar(' ');
    writeChar('C');
    writeChar(' ');
    writeChar('O');
    writeChar('R');
    writeChar(' ');
    writeChar('D');
```



```
}
```

```
void writeCountdown(){  
    clearDisplay();  
    writeChar('5');  
    _delay_ms(1000);  
    clearDisplay();  
    writeChar('4');  
    _delay_ms(1000);  
    clearDisplay();  
    writeChar('3');  
    _delay_ms(1000);  
    clearDisplay();  
    writeChar('2');  
    _delay_ms(1000);  
    clearDisplay();  
    writeChar('1');  
    _delay_ms(1000);  
}
```

```
void writeSetTime(){  
    clearDisplay();  
    writeChar('S');  
    writeChar('E');  
    writeChar('T');  
    writeChar(' ');  
    writeChar('T');  
    writeChar('I');  
    writeChar('M');  
    writeChar('E');  
    writeChar(':');  
}
```

```
void writeAlarmTrig(){  
    clearDisplay();  
    writeChar('A');  
    writeChar('L');  
    writeChar('A');  
    writeChar('R');  
    writeChar('M');  
    writeChar(' ');  
    writeChar('T');  
    writeChar('R');  
    writeChar('I');  
    writeChar('G');  
    writeChar('G');  
    writeChar('E');  
    writeChar('R');  
    writeChar('E');  
    writeChar('D');  
    changeCursor();  
    writeChar('A');
```

```

writeChar('T');
writeChar(' ');
displayTime(hourTrig, minTrig, secTrig);
_delay_ms(300);
waitForInput();
newInput = 0;
clearDisplay();
if(triggerIR1){
    writeChar('I');
    writeChar('R');
    writeChar('1');
    writeChar(':');
    writeChar('Y');
    writeChar('E');
    writeChar('S');
} else {
    writeChar('I');
    writeChar('R');
    writeChar('1');
    writeChar(':');
    writeChar('N');
    writeChar('O');
    writeChar(' ');
}
writeChar(' ');
if(triggerIR2){
    writeChar('I');
    writeChar('R');
    writeChar('2');
    writeChar(':');
    writeChar('Y');
    writeChar('E');
    writeChar('S');
} else {
    writeChar('I');
    writeChar('R');
    writeChar('2');
    writeChar(':');
    writeChar('N');
    writeChar('O');
}
changeCursor();
if(triggerTemp1){
    writeChar('T');
    writeChar('1');
    writeChar(' ');
    writeChar(':');
    writeChar('Y');
    writeChar('E');
    writeChar('S');
} else {
    writeChar('T');
    writeChar('1');
    writeChar(' ');
    writeChar(':');
}

```

```

        writeChar('N');
        writeChar('O');
        writeChar(' ');
    }
    writeChar(' ');
    if(triggerTemp2){
        writeChar('T');
        writeChar('2');
        writeChar(' ');
        writeChar(':');
        writeChar('Y');
        writeChar('E');
        writeChar('S');
    } else {
        writeChar('T');
        writeChar('2');
        writeChar(' ');
        writeChar(':');
        writeChar('N');
        writeChar('O');
    }
    _delay_ms(300);
    waitForInput();
    newInput = 0;
}

```

```

void writeNoTrig(){
    clearDisplay();
    writeChar('A');
    writeChar('L');
    writeChar('A');
    writeChar('R');
    writeChar('M');
    writeChar(' ');
    writeChar('W');
    writeChar('A');
    writeChar('S');
    writeChar(' ');
    writeChar('N');
    writeChar('O');
    writeChar('T');
    changeCursor();
    writeChar('T');
    writeChar('R');
    writeChar('I');
    writeChar('G');
    writeChar('G');
    writeChar('E');
    writeChar('R');
    writeChar('E');
    writeChar('D');
    _delay_ms(300);
    waitForInput();
    newInput = 0;
}

```

```

void displayTime(int h, int m, int s){
    unsigned int place1 = h / 10;
    unsigned int place0 = h - (place1 * 10);
    writeChar(readKeypad(place1));
    writeChar(readKeypad(place0));
    writeChar(':');
    place1 = m / 10;
    place0 = m - (place1 * 10);
    writeChar(readKeypad(place1));
    writeChar(readKeypad(place0));
    writeChar(':');
    place1 = s / 10;
    place0 = s - (place1 * 10);
    writeChar(readKeypad(place1));
    writeChar(readKeypad(place0));
}

void alarmActive(){
    sensor1 = PINC & 0b00000001;
    sensor2 = PINC & 0b00000010;
    temp1 = analogReadTemp1();
    temp2 = analogReadTemp2();
    triggerIR1 = 0;
    triggerIR2 = 0;
    triggerTemp1 = 0;
    triggerTemp2 = 0;
    if (sensor1 == 1 || sensor2 == 2 || temp1 > 120 || temp2 > 120) {

        if(sensor1 == 1) {
            triggerIR1 = 1;
        }
        if(sensor2 == 2) {
            triggerIR2 = 1;
        }
        if(temp1 > 120) {
            triggerTemp1 = 1;
        }
        if(temp2 > 120) {
            triggerTemp2 = 1;
        }
        hourTrig = hour;
        minTrig = min;
        secTrig = sec;
        summerOn();
        while(larmStatus){
            deactivate();
        }
    }

    if(newInput == 1) {
        char input = readKeypad(val);
        newInput = 0;
        if (input == 'A') {

```

```

        deactivate();
    }
}

void deactivate(){
    writeEnterPin();
    _delay_ms(300);
    waitForInput();
    newInput = 0;
    char pin = readKeypad(val);
    pinCompare[0] = pin;
    writeChar(pin);
    for(int i = 1; i < 4; i++){
        waitForInput();
        newInput = 0;
        char pin = readKeypad(val);
        pinCompare[i] = pin;
        writeChar(pin);
    }
    for(int i = 0; i < 4; i++){
        if (pinCompare[i] != pinCode[i]){
            wrongPin = 1;
        }
    }
    _delay_ms(1000);
    if(wrongPin){
        wrongPin = 0;
        writeWrongCode();
        _delay_ms(1000);
        clearDisplay();
    } else {
        larmStatus = 0;
        summerOff();
        redDiodeOff();
        greenDiodeOn();
        writeDeactivated();
        _delay_ms(1000);
        writeSelect();
    }
}

void functionA(){
    writeActivate();
    changeCursor();
    writeChar('P');
    writeChar('I');
    writeChar('N');
    writeChar(':');
    writeChar(' ');
    for(int i = 0; i < 4; i++){
        waitForInput();
        newInput = 0;
        char pin = readKeypad(val);
        pinCode[i] = pin;
    }
}

```

```

        writeChar(pin);
    }
    _delay_ms(1000);
    writeCountdown();
    larmStatus = 1;
    greenDiodeOff();
    redDiodeOn();
    writeActivated();
    _delay_ms(1000);
    clearDisplay();
}

```

```

void functionB(){
    writeSetTime();
    _delay_ms(300);

    waitForInput();
    clearDisplay();
    writeChar('H');
    writeChar('O');
    writeChar('U');
    writeChar('R');
    writeChar(':');
    writeChar(' ');
    newInput = 0;
    char pin = readKeypad(val);
    unsigned int place1 = val;
    writeChar(pin);
    waitForInput();
    newInput = 0;
    pin = readKeypad(val);
    unsigned int place0 = val;
    writeChar(pin);
    hour = place1*10 + place0;

    waitForInput();
    clearDisplay();
    writeChar('M');
    writeChar('I');
    writeChar('N');
    writeChar(':');
    writeChar(' ');
    newInput = 0;
    pin = readKeypad(val);
    place1 = val;
    writeChar(pin);
    waitForInput();
    newInput = 0;
    pin = readKeypad(val);
    place0 = val;
    writeChar(pin);
    min = place1*10 + place0;

    waitForInput();
}

```

```
clearDisplay();
writeChar('S');
writeChar('E');
writeChar('C');
writeChar(':');
writeChar(' ');
newInput = 0;
pin = readKeypad(val);
place1 = val;
writeChar(pin);
waitForInput();
newInput = 0;
pin = readKeypad(val);
place0 = val;
writeChar(pin);
sec = place1*10 + place0;
_delay_ms(1000);
writeSelect();
```

```
}
```

```
void functionC(){
    while (!newInput){
        clearDisplay();
        displayTime(hour, min, sec);
        _delay_ms(100);
    }
    newInput = 0;
    writeSelect();
}
```

```
void functionD(){
    if(triggerIR1 || triggerIR2 || triggerTemp1 || triggerTemp2) {
        writeAlarmTrig();
    } else {
        writeNoTrig();
    }
    writeSelect();
}
```