

```
/*
 * GccApplication1.c
 *
 * Created: 2019-04-09 10:07:53
 * Author : an2563gi-s
 */
#define F_CPU 16000000UL
#define F_SCL 100000UL

#include <avr/io.h>
#include <util/twi.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

//Defines display
#define E_D 7 //PA7
#define RW_D 6 //PA6
#define RS_D 5 //PA5
#define PD2 2

#define row1 0x00
#define row2 0x40
#define row3 0x14
#define row4 0x54

//Defines RTC
#define RTC_READ (0b11011111)
#define RTC_WRITE (0b11011110)

#define rtc_SEC 0x00
#define rtc_MIN 0x01
#define rtc_HOUR 0x02
#define rtc_WKDAY 0x03
#define rtc_DATE 0x04
#define rtc_MTH 0x05
#define rtc_YEAR 0x06
#define CONTROL 0x07
#define OSCTRIM 0x08

#define A_0_SEC 0x0A
#define A_0_MIN 0x0B
#define A_0_HOUR 0x0C
#define A_0_WKDAY 0x0D
#define A_0_DATE 0x0E
#define A_0_MTH 0x0F

// Defines time-variables
char s1;
char s10;
char m1;
char m10;
char h1;
char h10;
char date1 = 1;
char date10;
char weekday;
char month1 = 1;
char month10;
char year1;
```

```

char year10;

char a_co_s1;
char a_co_s10;
char a_co_m1 = 0;
char a_co_m10 = 0;
char a_co_h1 = 0;
char a_co_h10 = 0;
char a_co_weekday = 1;
char a_co_date1 = 0;
char a_co_date10 = 0;
char a_co_month1 = 0;
char a_co_month10 = 0;

// VARIABLES
char read_key , read_val, int_val;
volatile uint16_t pushed;
char AlarmOnOff = 0;
char CoffeeOnOff = 0;
char data;
int button = 0;
#define on 1
#define off 0
#define GREEN_LED 7 //Coffee pin

//METHODS
void changeTime(char year10, char year1, char month10, char month1, char date10 , char date1, char weekday, char h10, char
h1, char m10, char m1, char s10, char s1);
void initClock(char year10, char year1, char month10, char month1 , char date10, char date1, char weekday, char h10, char h1,
char m10, char m1, char s10, char s1);
void clockSetup(char inputVal, char address);
void setDataFlows();
void setKeyboard();
void startDisplay();
void clearDisplay();
void cursorSetPos(char location);
void writeChar(char c);
void writeText();
void writeWelcome();
void writeTime();
void writeAlarmStatus();
void writeInt();
void setAlarmClock(int status, char month10, char month1, char date10, char date1, char weekday, char h10, char h1, char
m10, char m1);
void initAlarmClock(char month10, char month1, char date10, char date1, char weekday, char h10, char h1, char m10, char m1,
char s10, char s1);
void disp_time();
void disp_coffeeAlarm();
void read_keypad();
void lampOn();
void lampOff();
void alarmSetUp();
int read_keypadAlarm();
void setLargeDisplay();
void buzzerOn();
void buzzerOff();

// RTC Setup
void clockSetup(char inputVal, char address){
    TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_START) {

```

```

        while(1);
    }

    TWDR = RTC_WRITE;
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_MT_SLA_ACK ) {
        while(1);
    }

    TWDR = address;
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_MT_DATA_ACK ) {
        while(1);
    }

    TWDR = inputVal;
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_MT_DATA_ACK ) {
        while(1);
    }

    TWCR = (1<<TWINT) | (1<<TWEN) | (1<<TWSTO);
}

void setOscilation(int input){

    TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_START) {
        while(1);
    }

    TWDR = RTC_WRITE;
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_MT_SLA_ACK ) {
        while(1);
    }

    TWDR = 0;
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_MT_DATA_ACK ) {
        while(1);
    }

    TWDR = 0b00000000 | (input<<7);
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1 << TWINT)));
    if((TWSR & 0xF8) != TW_MT_DATA_ACK ) {
        while(1);
    }

    TWCR = (1<<TWINT) | (1<<TWEN) | (1<<TWSTO);
}

char getTime(char RTCAddress) {

    TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);

```

```

while (!(TWCR & (1 << TWINT)));
if((TWSR & 0xF8) != TW_START) {
    while(1);
}

TWDR = RTC_WRITE;
TWCR = (1<<TWINT) | (1<<TWEN);
while (!(TWCR & (1 << TWINT)));
if((TWSR & 0xF8) != TW_MT_SLA_ACK ) {
    while(1);
}

TWDR = RTCAddress;
TWCR = (1<<TWINT) | (1<<TWEN);
while (!(TWCR & (1 << TWINT)));
if((TWSR & 0xF8) != TW_MT_DATA_ACK ) {
    while(1);
}

TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);
while (!(TWCR & (1 << TWINT)));
if((TWSR & 0xF8) != TW_REP_START) {
    while(1);
}

TWDR = RTC_READ;
TWCR = (1<<TWINT) | (1<<TWEN);
while (!(TWCR & (1 << TWINT)));
if((TWSR & 0xF8) != TW_MR_SLA_ACK ) {
    while(1);
}

TWCR = (1<<TWINT) | (0<<TWEA) | (1 << TWEN);
while (!(TWCR & (1 << TWINT)));

data = TWDR;

TWCR = (1<<TWINT) | (1<<TWEN) | (1 << TWSTO);
return data;
}

// Main method
int main(void)
{
    setDataFlows();
    setKeyboard();
    TWSR = 0b00000000;
    TWBR = 2;
    changeTime(1,9, 0,5, 2,7, 1, 1,3, 2,0, 0,0);
    startDisplay();
    setLargeDisplay();
    clearDisplay();
    writeWelcome();
    sei();

    while (1)
    {
        writeAlarmStatus();
        disp_time();
        disp_coffeeAlarm();
        if (pushed == 1) {
            read_keypad();
        }
    }
}

```

```

        pushed = 0;
    }
}

//INTERRUPTS
ISR(INT0_vect) {
    read_val = PINA & 0b00011110;
    read_val = read_val >> 1;
    pushed = 1;
}

ISR(INT1_vect) {
    if(AlarmOnOff==1 && CoffeeOnOff==1){
        lampOn();
        buzzerOn();
    } else if(AlarmOnOff==1){
        buzzerOn();
    }
}

//DATAFLOWS
void setDataFlows() {
    DDRB = 0b11111111;
    DDRA = 0b11100000;
    DDRD |= (1<<7);
    PORTA|= 0b00011110;
    PORTB = 0b00000000;
    PORTC =
    PORTD = 0b00000100;
    DDRC |= (1 << GREEN_LED); //| (1 << RED_LED);
}

void setKeyboard(){
    MCUCR = 0b00001111;
    GICR = 0b11000000;
}

void setLargeDisplay(){
    _delay_ms(100);
    PORTB = 0b00111100;
    PORTA |= (1 << E_D);
    PORTA &= ~(1<<E_D);
}

//DISPLAY
void startDisplay() {
    _delay_ms(100);
    PORTB = 0b00001101;
    PORTA |= (1 << E_D);
    PORTA &= ~(1<<E_D);
}

void clearDisplay() {
    PORTB = 0b00000001;
    PORTA |= (1 << E_D);
    PORTA &= ~(1<<E_D);
    _delay_ms(2);
}

void writeChar(char ch) {
    //Skriver ut ett specifikt tecken på displayen

```

```

        PORTA |= (1 << RS_D);
        PORTB = ch;
        _delay_ms(10);
        PORTA |= (1 << E_D);
        _delay_ms(10);
        PORTA &= ~(1<<E_D);
        PORTA &= ~(1<<RS_D);
        _delay_ms(10);
    }

void writeText(char string[]) {
    //Skriver ut en följd av tecken
    int i = 0;

    while (string[i] !='\0'){
        writeChar(string[i]);
        i++;
    }
}

void writeWelcome() {
    clearDisplay();
    writeText("WELCOME!");
}

void writeAlarmStatus(){
    cursorSetPos(row4);
    writeText("Clock:");
    if (AlarmOnOff == 1){
        writeText("On ");
    } else {
        writeText("Off");
    }
    cursorSetPos(0x5E);
    writeText("Coffee:");
    if (CoffeeOnOff == 1){
        writeText("On ");
    } else {
        writeText("Off");
    }
}

void writeTime(){
    cursorSetPos(row1);
    writeInt(h10);
    writeInt(h1);
    writeChar(':');
    writeInt(m10);
    writeInt(m1);
    writeChar(':');
    writeInt(s10);
    writeInt(s1);
    writeChar(' ');
    writeInt(date10);
    writeInt(date1);
    writeChar('/');
    writeInt(month10);
    writeInt(month1);
    writeText("/20");
    writeInt(year10);
    writeInt(year1);
}

```

```

void writeInt(int h){
    if(h/10==0){
        writeChar('0' + h%10);
    }else{
        writeInt(h/10);
        writeInt(h%10);
    }
}

void cursorSetPos(char location){
    PORTB = 0b10000000 | location;
    PORTA |= (1 << E_D);
    PORTA &= ~(1<<E_D);
    _delay_ms(2);
}

//RTC - Time and Alarm

void disp_time(){
    s1 = getTime(rtc_SEC) & 0b00001111;
    s10 = (getTime(rtc_SEC) & 0b01110000)>>4;
    m1 = getTime(rtc_MIN) & 0b00001111;
    m10 = (getTime(rtc_MIN) & 0b01110000)>>4;
    h1 = getTime(rtc_HOUR) & 0b00001111;
    h10 = (getTime(rtc_HOUR) & 0b00110000)>>4;
    date1 = getTime(rtc_DATE) & 0b00001111;
    date10 = (getTime(rtc_DATE) & 0b00110000)>>4;
    weekday = getTime(rtc_WKDAY) & 0b00000111;
    month1 = getTime(rtc_MTH) & 0b00001111;
    month10 = (getTime(rtc_MTH) & 0b00010000)>>4;
    year1 = getTime(rtc_YEAR) & 0b00001111;
    year10 = (getTime(rtc_YEAR) & 0b11110000)>>4;
    writeTime();
}

void disp_coffeeAlarm(){
    cursorSetPos(row3);
    writeInt(a_co_h10);
    writeInt(a_co_h1);
    writeChar(':');
    writeInt(a_co_m10);
    writeInt(a_co_m1);

    writeChar(' ');
    writeInt(a_co_date10);
    writeInt(a_co_date1);
    writeChar('/');
    writeInt(a_co_month10);
    writeInt(a_co_month1);
    writeText("/20");
    writeInt(year10);
    writeInt(year1);

    cursorSetPos(0x25);
    if (a_co_weekday == 1) {
        writeText("Mon");
    } else if (a_co_weekday == 2) {
        writeText("Tue");
    } else if (a_co_weekday == 3) {
        writeText("Wed");
    }
}

```

```

    } else if (a_co_weekday == 4) {
        writeText("Thu");
    } else if (a_co_weekday == 5) {
        writeText("Fri");
    } else if (a_co_weekday == 6) {
        writeText("Sat");
    } else if (a_co_weekday == 7) {
        writeText("Sun");
    }
}

void changeTime(char year10, char year1, char month10, char month1, char date10, char date1, char weekday, char h10, char
h1, char m10, char m1, char s10, char s1) {
    initClock(year10, year1, month10, month1, date10, date1, weekday, h10, h1, m10, m1, s10, s1);
}

void setAlarmClock(int status, char month10, char month1, char date10, char date1, char weekday, char h10, char h1, char
m10, char m1) {
    if(status == on){
        initAlarmClock(month10, month1, date10, date1, weekday, h10, h1, m10, m1, 0, 0);
        AlarmOnOff = 1;
    } else {
        AlarmOnOff = 0;
    }
    // Här ska alarm kaffe vara med om man valt det alternativet
}

void initAlarmClock(char month10, char month1, char date10, char date1, char weekday, char h10, char h1, char m10, char m1,
char s10, char s1) {
    char sInput = (s10 << 4) | s1;
    clockSetup(sInput, A_0_SEC);
    char mInput = (m10 << 4) | m1;
    clockSetup(mInput, A_0_MIN);
    char hInput = (h10 << 4) | h1;
    clockSetup(hInput, A_0_HOUR);
    char wInput = (1 << 7) | (1 << 6) | (1 << 5) | (1 << 4) | weekday;
    clockSetup(wInput, A_0_WKDAY);
    char dateInput = (date10 << 4) | date1;
    clockSetup(dateInput, A_0_DATE);
    char monthInput = (month10 << 4) | month1;
    clockSetup(monthInput, A_0_MTH);
    char alarmActivate = (1 << 4);
    clockSetup(alarmActivate, CONTROL);
    //setOscilation(1);
}

void initClock(char year10, char year1, char month10, char month1, char date10, char date1, char weekday, char h10, char h1,
char m10, char m1, char s10, char s1) {
    char sInput = (1 << 7) | (s10 << 4) | s1;
    char mInput = (m10 << 4) | m1;
    clockSetup(mInput, rtc_MIN);
    char hInput = (h10 << 4) | h1;
    clockSetup(hInput, rtc_HOUR);
    char wInput = weekday;
    clockSetup(wInput, rtc_WKDAY);
    char dateInput = (date10 << 4) | date1;
    clockSetup(dateInput, rtc_DATE);
    char monthInput = (month10 << 4) | month1;
    clockSetup(monthInput, rtc_MTH);
    char yInput = (year10 << 4) | year1;
    clockSetup(yInput, rtc_YEAR);
    clockSetup(sInput, rtc_SEC);
}

```



```

        //setOscilation(1);
    }

void alarmSetUp(){
    char pos = 0x14;
    cursorSetPos(pos);
    int stop = 0;
    while (stop != 1) {
        if (pushed == 1) {
            button = read_keypadAlarm();
            if (button != -1) {
                if ((pos >= 0x14) && (pos <= 0x25)){
                    switch (button) {
                        case 12:
                            if (pos > 0x14) {
                                if (pos == 0x25) {
                                    pos = 0x1e;
                                } else if ((pos == 0x1d) || (pos == 0x1a) ||
(pos== 0x17)) {

                                    pos = pos - 2;
                                } else {
                                    pos--;
                                }
                            }
                            cursorSetPos(pos);
                            button = (-1);
                            break;

                        case 13:
                            if (pos < 0x25) {
                                if (pos == 0x1e) {
                                    pos = 0x25;
                                } else if ((pos == 0x15) || (pos == 0x18) || (pos
== 0x1b)) {

                                    pos = pos + 2;
                                } else {
                                    pos++;
                                }
                            }
                            cursorSetPos(pos);
                            button = (-1);
                            break;

                        case 1:
                            if (pos == 0x14) {
                                a_co_h10 = button;
                                cursorSetPos(pos);
                                writeInt(a_co_h10);
                                pos++;
                                cursorSetPos(pos);
                                button = (-1);
                                break;
                            }
                            else if (pos == 0x15) {
                                a_co_h1 = button;
                                cursorSetPos(pos);
                                writeInt(a_co_h1);
                                pos = pos + 2;
                                cursorSetPos(pos);
                                button = -1;
                                break;
                            }
                    }
                }
            }
        }
    }
}

```

```

else if (pos == 0x17) {
    a_co_m10 = button;
    cursorSetPos(pos);
    writeInt(a_co_m10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1a) {
    a_co_date10 = button;
    cursorSetPos(pos);
    writeInt(a_co_date10);
    pos++;
    cursorSetPos(pos);
    button = (-1);
    break;
}
else if (pos == 0x1b) {
    a_co_date1 = button;
    cursorSetPos(pos);
    writeInt(a_co_date1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1d) {
    a_co_month10 = button;
    cursorSetPos(pos);
    writeInt(a_co_month10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1e) {
    a_co_month1 = button;
    cursorSetPos(pos);
    writeInt(a_co_month1);
    button = -1;
    pos = 0x25;
    cursorSetPos(pos);
    break;
}
else if (pos == 0x25) {
    a_co_weekday = button;
    cursorSetPos(pos);
    writeText("Mon");
    button = -1;
    break;
}
break;

```

```

case 2:
if (pos == 0x14) {
    a_co_h10 = button;
    cursorSetPos(pos);
    writeInt(a_co_h10);
    if (a_co_h1 > 3) {
        a_co_h1 = 0;
        cursorSetPos(0x15);
        writeInt(a_co_h1);
    }
    pos++;
    cursorSetPos(pos);
    button = (-1);
    break;
}
else if (pos == 0x15) {
    a_co_h1 = button;
    cursorSetPos(pos);
    writeInt(a_co_h1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x17) {
    a_co_m10 = button;
    cursorSetPos(pos);
    writeInt(a_co_m10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1a) {
    a_co_date10 = button;
    cursorSetPos(pos);
    writeInt(a_co_date10);
    pos++;
    cursorSetPos(pos);
    button = (-1);
    break;
}
else if (pos == 0x1b) {
    if (a_co_date10 < 3) {
        a_co_date1 = button;
        cursorSetPos(pos);
        writeInt(a_co_date1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}

```

```

}
else if (pos == 0x1e) {
    a_co_month1 = button;
    cursorSetPos(pos);
    writeInt(a_co_month1);
    pos = 0x25;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x25) {
    a_co_weekday = button;
    cursorSetPos(pos);
    writeText("Tue");
    button = -1;
    break;
}
break;

case 3:
if (pos == 0x15) {
    a_co_h1 = button;
    cursorSetPos(pos);
    writeInt(a_co_h1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x17) {
    a_co_m10 = button;
    cursorSetPos(pos);
    writeInt(a_co_m10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1a) {
    if((a_co_month10 == 0) && (a_co_month1 == 2)) {
        break;
    } else {
        a_co_date10 = button;
        cursorSetPos(pos);
        writeInt(a_co_date10);
        if (a_co_date1 > 1) {
            a_co_date1 = 0;
            cursorSetPos(0x1b);
            writeInt(a_co_date1);
        }
        pos++;
        cursorSetPos(pos);
        button = (-1);
    }
}

```

```

        }
        break;
    }
    else if (pos == 0x1b) {
        if (a_co_date10 < 3) {
            a_co_date1 = button;
            cursorSetPos(pos);
            writeInt(a_co_date1);
            pos = pos + 2;
            cursorSetPos(pos);
            button = -1;
        }
        break;
    }
    else if (pos == 0x1e) {
        if (a_co_month10 == 0) {
            a_co_month1 = button;
            cursorSetPos(pos);
            writeInt(a_co_month1);
            pos = 0x25;
            cursorSetPos(pos);
            button = -1;
        }
        break;
    }
    else if (pos == 0x25) {
        a_co_weekday = button;
        cursorSetPos(pos);
        writeText("Wed");
        button = -1;
        break;
    }
    break;

    case 4:
    if (pos == 0x15) {
        if (a_co_h10 < 2){
            a_co_h1 = button;
            cursorSetPos(pos);
            writeInt(a_co_h1);
            pos = pos + 2;
            cursorSetPos(pos);
            button = -1;
        }
        break;
    }
    else if (pos == 0x17) {
        a_co_m10 = button;
        cursorSetPos(pos);
        writeInt(a_co_m10);
        pos++;
        cursorSetPos(pos);
        button = -1;
        break;
    }
    else if (pos == 0x18) {
        a_co_m1 = button;
        cursorSetPos(pos);
        writeInt(a_co_m1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }

```

```

        break;
    }
    else if (pos == 0x1b) {
        if (a_co_date10 < 3) {
            a_co_date1 = button;
            cursorSetPos(pos);
            writeInt(a_co_date1);
            pos = pos + 2;
            cursorSetPos(pos);
            button = -1;
        }
        break;
    }
    else if (pos == 0x1e) {
        if (a_co_month10 == 0) {
            a_co_month1 = button;
            cursorSetPos(pos);
            writeInt(a_co_month1);
            pos = 0x25;
            cursorSetPos(pos);
            button = -1;
        }
        break;
    }
    else if (pos == 0x25) {
        a_co_weekday = button;
        cursorSetPos(pos);
        writeText("Thu");
        button = -1;
        break;
    }
    break;

case 5:
if (pos == 0x15) {
    if (a_co_h10 < 2){
        a_co_h1 = button;
        cursorSetPos(pos);
        writeInt(a_co_h1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x17) {
    a_co_m10 = button;
    cursorSetPos(pos);
    writeInt(a_co_m10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}

```

```

}
else if (pos == 0x1b) {
    if (a_co_date10 < 3) {
        a_co_date1 = button;
        cursorSetPos(pos);
        writeInt(a_co_date1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x1e) {
    if (a_co_month10 == 0) {
        a_co_month1 = button;
        cursorSetPos(pos);
        writeInt(a_co_month1);
        pos = 0x25;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x25) {
    a_co_weekday = button;
    cursorSetPos(pos);
    writeText("Fri");
    button = -1;
    break;
}
break;

case 6:
if (pos == 0x15) {
    if (a_co_h10 < 2){
        a_co_h1 = button;
        cursorSetPos(pos);
        writeInt(a_co_h1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1b) {
    if (a_co_date10 < 3) {
        a_co_date1 = button;
        cursorSetPos(pos);
        writeInt(a_co_date1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
}

```

```

        break;
    }
    else if (pos == 0x1e) {
        if (a_co_month10 == 0) {
            a_co_month1 = button;
            cursorSetPos(pos);
            writeInt(a_co_month1);
            pos = 0x25;
            cursorSetPos(pos);
            button = -1;
        }
        break;
    }
    else if (pos == 0x25) {
        a_co_weekday = button;
        cursorSetPos(pos);
        writeText("Sat");
        button = -1;
        break;
    }
    break;

case 7:
if (pos == 0x15) {
    if (a_co_h10 < 2){
        a_co_h1 = button;
        cursorSetPos(pos);
        writeInt(a_co_h1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1b) {
    if (a_co_date10 < 3) {
        a_co_date1 = button;
        cursorSetPos(pos);
        writeInt(a_co_date1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x1e) {
    if (a_co_month10 == 0) {
        a_co_month1 = button;
        cursorSetPos(pos);
        writeInt(a_co_month1);
        pos = 0x25;
        cursorSetPos(pos);
        button = -1;
    }
}

```



```

        }
        break;
    }
    else if (pos == 0x25) {
        a_co_weekday = button;
        cursorSetPos(pos);
        writeText("Sun");
        button = -1;
        break;
    }
    break;

case 8:
if (pos == 0x15) {
    if (a_co_h10 < 2){
        a_co_h1 = button;
        cursorSetPos(pos);
        writeInt(a_co_h1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1b) {
    if (a_co_date10 < 3) {
        a_co_date1 = button;
        cursorSetPos(pos);
        writeInt(a_co_date1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x1e) {
    if (a_co_month10 == 0) {
        a_co_month1 = button;
        cursorSetPos(pos);
        writeInt(a_co_month1);
        pos = 0x25;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
break;

case 9:
if (pos == 0x15) {
    if (a_co_h10 < 2){
        a_co_h1 = button;
        cursorSetPos(pos);

```

```

        writeInt(a_co_h1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1b) {
    if((a_co_month10 == 0) && (a_co_month1 == 2)) {
        break;
    } else {
        if (a_co_date10 < 3) {
            a_co_date1 = button;
            cursorSetPos(pos);
            writeInt(a_co_date1);
            pos = pos + 2;
            cursorSetPos(pos);
            button = -1;
        }
    }
    break;
}
else if (pos == 0x1e) {
    if (a_co_month10 == 0) {
        a_co_month1 = button;
        cursorSetPos(pos);
        writeInt(a_co_month1);
        pos = 0x25;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
break;

case 0:
if (pos == 0x14) {
    a_co_h10 = button;
    cursorSetPos(pos);
    writeInt(a_co_h10);
    pos++;
    cursorSetPos(pos);
    button = (-1);
    break;
}
else if (pos == 0x15) {
    a_co_h1 = button;
    cursorSetPos(pos);
    writeInt(a_co_h1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}

```

```

}
else if (pos == 0x17) {
    a_co_m10 = button;
    cursorSetPos(pos);
    writeInt(a_co_m10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x18) {
    a_co_m1 = button;
    cursorSetPos(pos);
    writeInt(a_co_m1);
    pos = pos + 2;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1a) {
    a_co_date10 = button;
    cursorSetPos(pos);
    writeInt(a_co_date10);
    pos++;
    cursorSetPos(pos);
    button = (-1);
    break;
}
else if (pos == 0x1b) {
    if (a_co_date10 != 0) {
        a_co_date1 = button;
        cursorSetPos(pos);
        writeInt(a_co_date1);
        pos = pos + 2;
        cursorSetPos(pos);
        button = -1;
    }
    break;
}
else if (pos == 0x1d) {
    a_co_month10 = button;
    cursorSetPos(pos);
    writeInt(a_co_month10);
    pos++;
    cursorSetPos(pos);
    button = -1;
    break;
}
else if (pos == 0x1e) {
    if (a_co_month10 != 0) {
        a_co_month1 = button;
        cursorSetPos(pos);
        writeInt(a_co_month1);
        button = -1;
        pos = 0x25;
        cursorSetPos(pos);
    }
    break;
}
break;

```

case 69:

```
        setAlarmClock(on, a_co_month10, a_co_month1, a_co_date10,  
a_co_date1, a_co_weekday, a_co_h10, a_co_h1, a_co_m10, a_co_m1);  
        cursorSetPos(row2);  
        writeText("Do you want coffe?");  
        stop = 1;  
        break;  
    }  
    pushed = 0;  
}  
  
}  
  
}  
  
}  
  
}
```

```
//KEYBOARD  
void read_keypad(){
```

```
    read_val = PINA & 0b00011110;  
    read_val = read_val >> 1;  
  
    switch (read_val)  
    {  
        case 0x00:  
            break;  
  
        case 0x01:  
            break;  
  
        case 0x02:  
            read_val = 255;  
            alarmSetUp();  
            break;  
  
        case 0x03:  
            break;  
  
        case 0x04:  
            break;  
  
        case 0x05:  
            CoffeeOnOff = off;  
            break;  
  
        case 0x06:  
            lampOn();  
            CoffeeOnOff = on;  
            cursorSetPos(row2);  
            writeText("Coffee is brewing!");  
            break;  
  
        case 0x07:  
            CoffeeOnOff = on;  
            cursorSetPos(row2);  
            writeText("           ");  
            break;  
  
        case 0x08:  
            break;  
  
        case 0x09:  
            break;
```

```

        case 0x0a:
        CoffeeOnOff =off;
        lampOff();
        break;

        case 0x0b:
        break;

        case 0x0c:
        break;

        case 0x0d:
        buzzerOff();
        AlarmOnOff = off;
        break;

        case 0x0e:
        break;

        case 0x0f:
        CoffeeOnOff = off;
        cursorSetPos(row2);
        writeText("          ");
        break;
    }
}

```

```

int read_keypadAlarm(){

    int_val = PINA & 0b00011110;
    int_val = int_val >> 1;

    switch (int_val)
    {
        case 0x00:

            return 1;
            break;

        case 0x01:

            return 2;
            break;

        case 0x02:

            return 3;
            break;

        case 0x04:

            return 4;
            break;

        case 0x05:

            return 5;
            break;

        case 0x06:

```

```

        return 6;
        break;

    case 0x07:

        return 69;
        break;

    case 0x08:

        return 7;
        break;

    case 0x09:

        return 8;
        break;

    case 0x0a:

        return 9;
        break;

    case 0x0c:

        return 12;
        break;

    case 0x0d:

        return 0;
        break;

    case 0x0e:
        ;
        return 13;
        break;
    }
    return (-1);
}

//BUZZER - alarm
void buzzerOn(){
    PORTD |= (1<<7);
}

void buzzerOff(){
    PORTD &= ~(1<<7);
}

//LAMP (later coffeboiler)
void lampOff(){
    PORTC &=~(1<<PC7);
    cursorSetPos(row2);
    writeText("          ");
}

void lampOn(){
    PORTC |= (1<<PC7);
}

```