

Proiect Arduino

Programare Orientată pe Obiecte

Tema 7. Ceas si Alarma digitala

Studenti:

Lungu Vanessa Elena 4.1

Martinov Ana-Luciana 4.1

Dani Bulaichi 1.2

An 2 grupa 4 si 1

Timișoara, 2020

Cerințele inițiale:

La pornire ceasul porneste de la 00:00:00 (hh:mm:ss) si se incrementeaza cu o secunda.

Folosind butonul select se selecteaza: modul normal care afiseaza ceasul pe randul 1 si alarma pe randul 2, modul de setare a ceasului si modul de setare al alarmei.

Butoanele left/right muta cursorul care indica valoare care v-a fi setata iar butoanele up/down seteaza valoarea corespunzatoare selectiei.

Pentru ceas setam ora si minutul iar pentru alarma ora, minutul si starea de active sau inactiv.

Utilizand un Arduino si un LCD Keypad Shield sa se implementeze un ceas si alarma digitala. SE INTERZICE UTILIZAREA FUNCTIEI DELAY() !!!

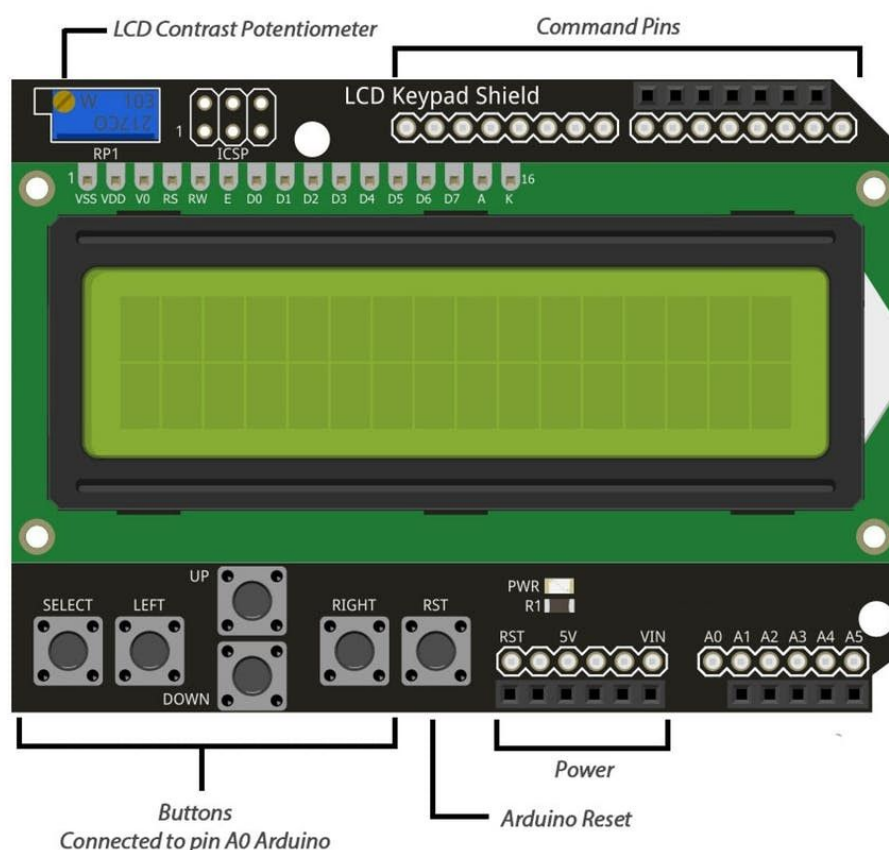


Fig.1.Schema hardware a sistemului

În Figura 2 următoare este prezentată schema bloc a softwarului care comandă. Ceasul cu alarma (schema bloc a fost realizată folosind schemele UML generate automat în Visual Studio 2015):

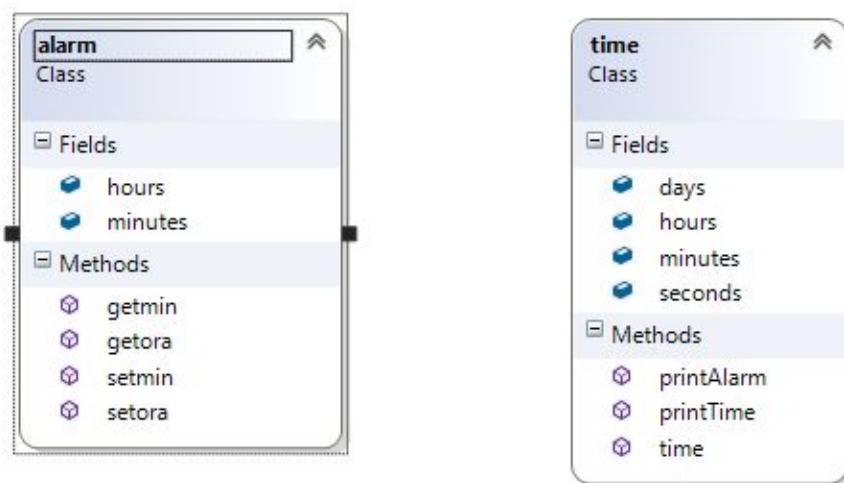


Fig.2.Schema UML a sistemului

Din schema UML se poate observa clasa **time** care are rolul de a seta pinii care sunt conectați prin shield la microcontroler. Clasa oferă funcții pentru interfațarea alarmei setata si a timpului.

Concluzii

În urma realizării acestui proiect ne-am aprofundat cunoștințele teoretice a limbajului C++, punându-le în practică în domeniul microcontrolelor.

Pe parcursul proiectului au fost întâmpinate și depășite mai multe dificultăți atât legate de munca în echipă cât și tehnice. Au existat probleme de comunicare, conflicte de programul fiecaruia, diferențe de viziune și de împărțire a sarcinilor. Aceste probleme de comunicare au fost depășite și am învățat să comunicăm mai bine între noi.

Nu au existat probleme tehnice. Materialele au fost folosite cu ușurință și nu s-a folosit funcția DELAY().

Codul sursă a programului

```
#include <LiquidCrystal.h>
#include "LCDKeypad.h"
```

```

#define DAYS 0
#define HOURS 1
#define MINUTES 2
#define SECONDS 3

// The LCD screen
LCDKeypad lcd;

class alarm
{
public:
    int minutes = 0;
    int hours = 12;
    void setmin(int min){this->minutes=min;}
    int getmin(){return minutes;}
    void setora(int ora){this->hours=ora;}
    int getora(){return hours;}

}alm;

// The time model
class time {
public:
    unsigned int days;
    unsigned int hours;
    unsigned int minutes;
    unsigned int seconds=1;
    time(){
        days = 0;
        hours = 0;
        minutes = 0;
        seconds = 1;
    }
    // Print the time on the LCD
    void printTime();

    void printAlarm();
}t;

void time::printTime(){
    // Set the cursor at the beginning of the second row

```

```

    lcd.setCursor(0,1);
    char time[17];
    sprintf(time, " %02i:%02i:%02i", hours, minutes, seconds);
    lcd.print(time);
}

```

```

void time:: printAlarm() {
    // Set the cursor at the begining of the second row
    lcd.setCursor(9,1);
    char time[17];
    // alm.setora(hours);
    // alm.setmin(minutes);
    sprintf(time, " %02i:%02i", alm.getora(), alm.getmin());
    lcd.print(time);
}
unsigned int setting = 0;

```

```

void setup() {
    // Set up the LCD's number of columns and rows:
    lcd.begin(16,2);

    // Set the cursor at the begining of the first row
    lcd.setCursor(0,0);

    // Print a text in the first row
    lcd.print("Seteaza ");
}

```

```

void loop() {
    // Increase the time model by one second
    incTime();

    // Print the time on the LCD
    t.printTime();

    t.printAlarm();
    alarmtick();
    // Listen for buttons for 1 second
    buttonListen();
}

```

```

void buttonListen() {

```

```
// Read the buttons five times in a second
for (int i = 0; i < 5; i++) {

    // Read the buttons value
    int button = lcd.button();

    switch (button) {

        // Right button was pushed
        case KEYPAD_RIGHT:
            setting++;
            break;

        // Left button was pushed
        case KEYPAD_LEFT:
            setting--;
            break;

        // Up button was pushed
        case KEYPAD_UP:
            switch (setting) {
                case HOURS:
                    alm.hours++;
                    if(alm.hours==23) alm.hours=0;

                    break;
                case MINUTES:
                    alm.minutes++;
                    if(alm.minutes==60) alm.minutes=0;
                    break;
            }
            break;

        // Down button was pushed
        case KEYPAD_DOWN:
            switch (setting) {
                case HOURS:
                    alm.hours--;
                    if (alm.hours == -1) alm.hours = 23;
                    break;
                case MINUTES:
                    alm.minutes--;
                    if (alm.minutes == -1) alm.minutes = 59;
                    break;
            }
            break;
    }
}
```

```

    alm.minutes--;
    if (alm.minutes == -1) alm.minutes = 59;
    break;

}
}

setting %= 4;
printSetting();

t.days %= 100;
t.hours %= 24;
t.minutes %= 60;
t.seconds %= 60;
t.printTime();

// Wait one fifth of a second to complete
while(millis() % 200 != 0);
}
}

// Print the current setting
void printSetting() {
    lcd.setCursor(9,0);

    switch (setting) {
    case HOURS:
        lcd.print("Ora ");
        break;
    case MINUTES:
        lcd.print("Min");
        break;
    case SECONDS:
        lcd.print("Sec");
    }
}

void alarmtick(){

    if((t.hours==alm.getora())&(t.minutes==alm.getmin()))
    {
        lcd.setCursor(0,0);

```

```

lcd.print("Wake Up");
delay(5000);
lcd.setCursor(0,0);
lcd.print("Seteaza");

}
}

```

// Increase the time model by one second

```

void incTime() {
    // Increase seconds
    t.seconds++;

    if (t.seconds == 60) {
        // Reset seconds
        t.seconds = 0;

        // Increase minutes
        t.minutes++;

        if (t.minutes == 60) {
            // Reset minutes
            t.minutes = 0;

            // Increase hours
            t.hours++;

            if (t.hours == 24) {
                // Reset hours
                t.hours = 0;

                // Increase days
                t.days++;
            }
        }
    }
}

```