

Ceas de birou smart

Introducere

Acest proiect implică dezvoltarea unui ceas de birou inteligent, nu doar un simplu ceas, ci unul care îmbină funcționalitatea cu inovația. Scopul său este de a afișa ora curentă și temperatura din cameră, oferind totodată o iluminare suficient de puternică pentru a elimina necesitatea unei alte surse de lumină pe birou, prin intermediul LED-urilor din jurul său.

Ideea pentru acest proiect a luat naștere din dorința de a crea ceva unic și util, combinată cu lipsa de iluminare adecvată pe biroul meu în timp ce mă gândeam la ce aș putea realiza. Am constatat că o sursă de lumină suplimentară ar fi extrem de benefică, mai ales pe timp de noapte, când lumina naturală lipsește.

Sunt convinsă că nu sunt singura persoană care se confruntă cu această problemă, iar acest ceas inteligent poate fi de mare ajutor pentru oricine are nevoie de o sursă de lumină eficientă pe birou. Proiectul meu are potențialul de a aduce un plus de confort și funcționalitate în spațiul de lucru al multor persoane.

Descriere generală

Funcționalități:

- Controlul LED-urilor
 - LED-urile RGB** își pot schimba culoarea în funcție de semnalele primite de la telecomanda IR.
 - LED-urile albe** își ajustează intensitatea în funcție de lumina ambientală (dacă este mai puțină lumină în jur, LED-urile luminează mai tare). Nivelul de lumină este măsurat cu ajutorul unui fotorezistor.
- Afișarea orei și temperaturii
 - Ora curentă** este obținută de la modulul RTC și afișată pe ecranul LCD.
 - Temperatura** este citită de la senzorul DS18B20 și afișată pe ecranul LCD.
- Controlul prin telecomanda IR
 - Diverse butoane ale telecomenzii IR pot schimba culoarea LED-urilor RGB sau pot porni/opri LED-urile albe.

Schema bloc:



Schema bloc este realizată în Figma, folosind icon-uri de pe www.flaticon.com.

Laboratoare folosite:

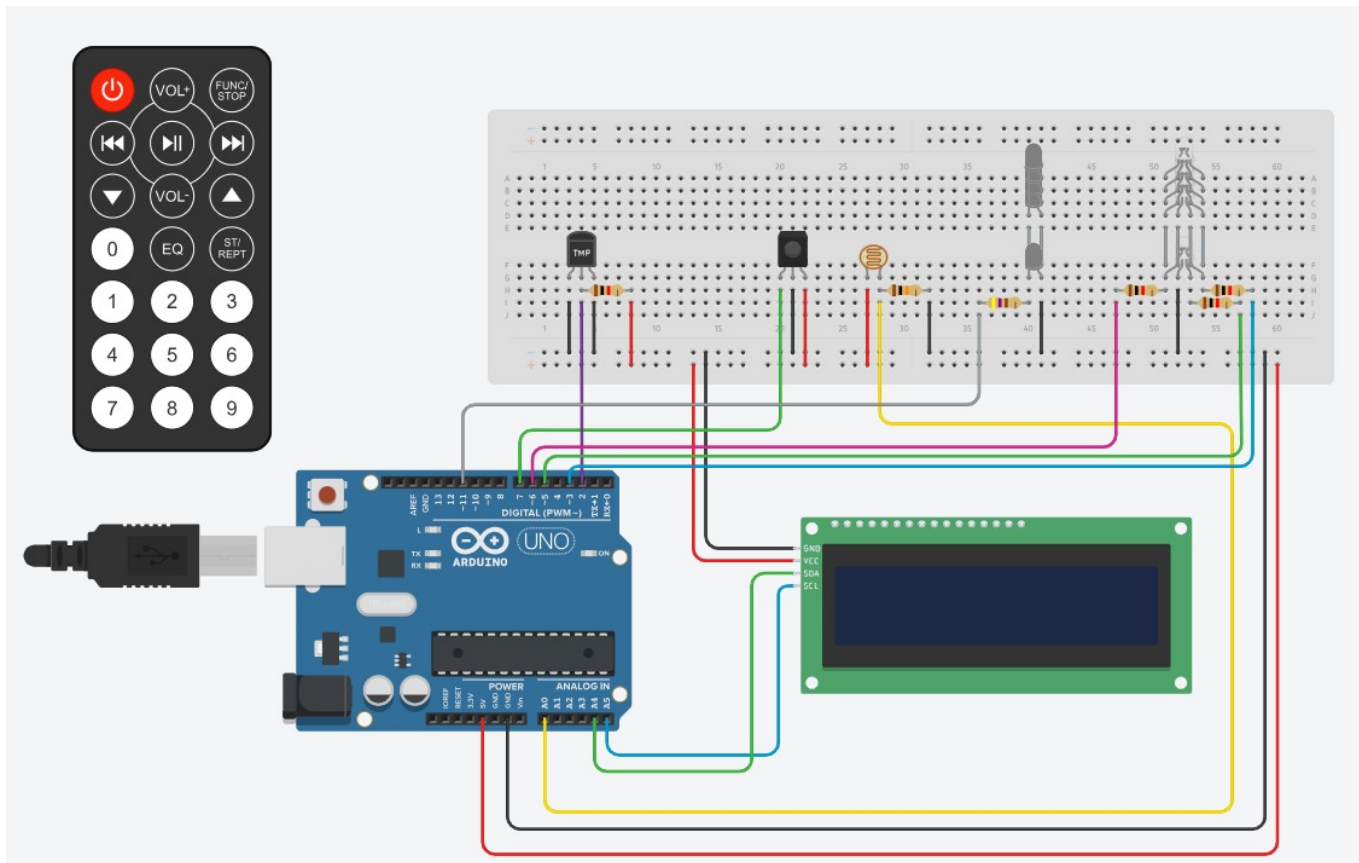
- Laboratorul 3: **PWM** - folosit pentru a regla intensitatea LED-urilor albe în funcție de lumina ambientală
- Laboratorul 4: **ADC** - ma folosesc de convertorul analogic-digital care este implementat pe arduino pentru a prelua semnalele analogice de la senzori si a le converti in valori numerice discrete pentru a putea face calcule cu acestea. De asemenea, transform si valori numerice in valori analogice, atunci cand doresc sa folosesc o anumita intensitate pentru LED-urile albe
- Laboratorul 6: **I2C** - datele trimise si primite de la senzorul RTC si de la ecranul LCD sunt prin interfata I2C

Hardware Design

Componente utilizate:

- Breadboard
- Aduino uno
- I2C display
- Senzor de temperatura
- Senzor infrarosu
- Telecomanda
- Fotorezistor
- LED-uri albe
- LED-uri RGB
- Rezistente
- Fire
- Modul RTC

Circuit proiect:

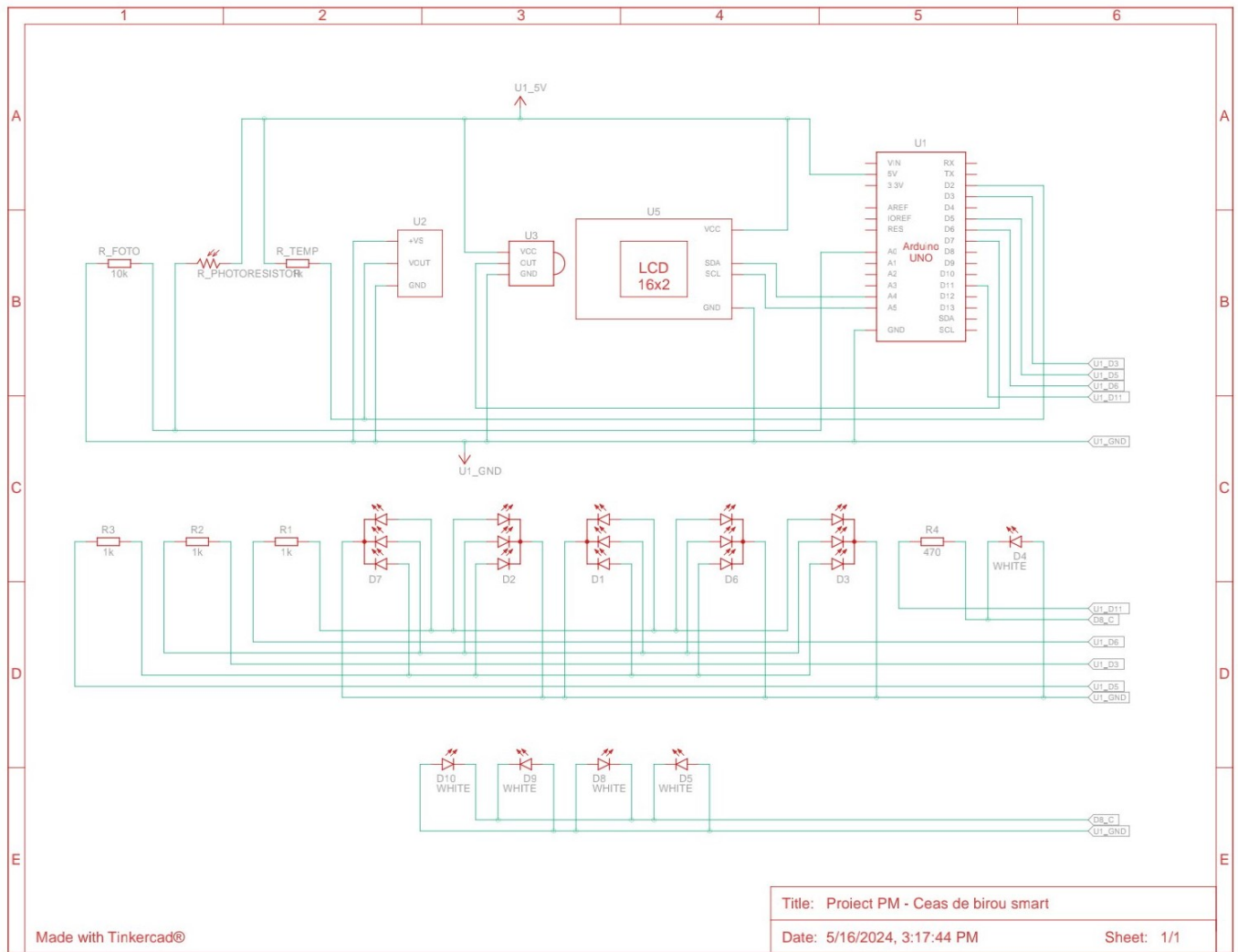


Schema proiectului este realizată în www.tinkercad.com.

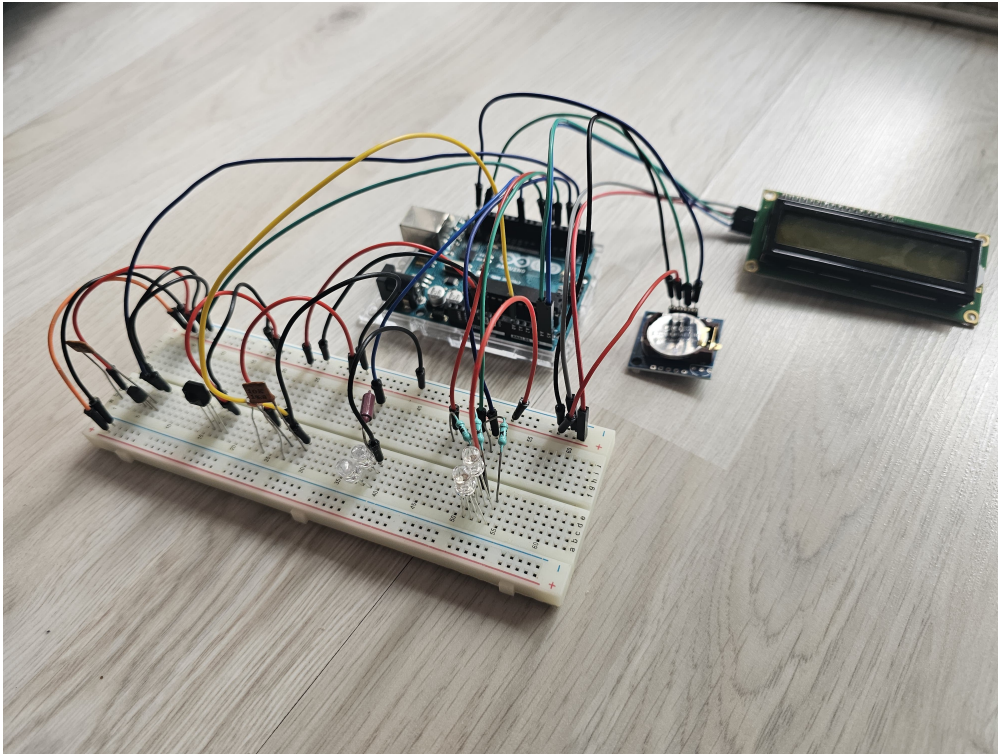
De pe această schemă **lipsește modulul RTC**, întrucât acesta nu există în Tinkercad.

De asemenea, LED-urile RGB au aici catodul comun, însă în realitate folosesc **LED-uri cu anod comun**.

Schematic:



Hardware-ul inainte de a fi bagat intr-o carcasa de ceas:



Software Design

Mediu de Dezvoltare

- Arduino IDE

Librării utilizate

- **DS1307.h** - folosita pentru a comunica cu modulul de RTC
- **Wire.h** - folosita pentru comunicatiile cu I2C
- **LiquidCrystal_I2C.h** - folosita pentru facilitarea afisarii datelor pe ecranul LCD
- **OneWire.h** - folosita pentru a facilita comunicarea cu senzorul de temperatura
- **DallasTemperature.h** - specifica senzorului de temperatura pe care l-am folosit (DS1820)
- **IRremote.hpp** - folosita pentru decodarea semnalelor primite de la senzorul de IR

Implementare

Dupa ce am adaugat aceste librării, urmeaza partea de implementare, care este realizata in 3 etape:

Initializari

Mai intai initializez componentele, mai exact specific pinii pe care ii va folosi fiecare componenta, alaturi de alte initializari acolo unde este cazul.

```
// LED-uri RGB
int ledRGB_R = 6; // pinul 6 (pwm)
int ledRGB_G = 5; // pinul 7 (pwm)
int ledRGB_B = 3; // pinul 3 (pwm)

// LED-uri albe
int led_white = 10; // pinul 10 (pwm)
int is_on = 1; // Variabila pentru a porni/opri ledurile

// LCD I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); // (adresa I2C, nr coloane, nr linii)

// Modul RTC
DS1307 rtc;

// Senzor IR
const int RECV_PIN = 7; // pinul 7
String code = ""; // Variabila pentru a stoca codul citit de senzor

// Senzor de temperatura DS18B20
int temp = 2; // pinul 2
OneWire oneWire(temp); // Instanta OneWire pentru a comunica cu device-uri
OneWire
DallasTemperature temp_sensor(&oneWire); // Pasez adresa instantei
```

setup()

Mai intai specific faptul ca atat pinii de la LED-urile RBG, cat si cei de la LED-urile albe sunt **output**.

```
pinMode(ledRGB_R, OUTPUT);
pinMode(ledRGB_G, OUTPUT);
pinMode(ledRGB_B, OUTPUT);

pinMode(led_white, OUTPUT);
```

Dupa ce am specificat acest lucru, setez ca LED-urile RGB sa fie stinse la pornirea ceasului, iar LED-urile albe sa fie aprinse.

```
digitalWrite(ledRGB_R, 1);
digitalWrite(ledRGB_G, 1);
digitalWrite(ledRGB_B, 1);
```

```
digitalWrite(led_white, 0);
```

Pentru a afisa pe ecranul LCD ora curenta si temperatura din camera in care se afla ceasul, fac un mic setup pe acesta in care afisez cuvintele specifice: pe primul rand "Ora" si pe al doilea rand "Temperatura [...] C".

```
lcd.init();  
lcd.backlight();  
lcd.print("Ora:");  
lcd.setCursor(0, 1);  
lcd.print("Temperatura:");  
lcd.setCursor(15, 1);  
lcd.print("C");
```

Pentru a putea afisa ora curenta, trebuie sa pornesc modulul RTC si sa ii introduc prima oara manual ora si data curenta, dupa care el va retine automat aceste date.

```
rtc.begin();  
rtc.set(50, 59, 9, 23, 05, 2024); //sec, min, hour, day, month, year  
rtc.start();
```

Mai ramane doar sa pornesc senzorul IR si senzorul de temperatura.

```
IrReceiver.begin(RECV_PIN, DISABLE_LED_FEEDBACK);  
temp_sensor.begin();
```

loop()

În această funcție se execută în mod repetitiv următoarele acțiuni:

1. Afisare pe LCD:

- Declaram variabile in care se retin datele de la senzorul RTC legate de ora si data curenta. Apoi **afisez pe ecran doar ora**, avand inasa grija in cazul in care una din variabilele "hour", "min" sau "sec" sunt de o singura cifra (0 - 9), caz in care mai trebuie adaugat un "0" in fata.

```
uint8_t sec, min, hour, day, month;  
uint16_t year;  
rtc.get(&sec, &min, &hour, &day, &month, &year);  
  
lcd.setCursor(5, 0); // Setează cursorul la linia 1, poziția 5  
if (hour < 10) {  
    lcd.print("0");  
    lcd.print(hour, DEC);  
} else {  
    lcd.print(hour, DEC);  
}  
}
```

```
lcd.print(':');
if (min < 10) {
  lcd.print("0");
  lcd.print(min, DEC);
} else {
  lcd.print(min, DEC);
}

lcd.print(':');
if (sec < 10) {
  lcd.print("0");
  lcd.print(sec, DEC);
} else {
  lcd.print(sec, DEC);
}
```

- Alatur de ora, pe ecranul LCD trebuie sa mai **afisez si temperatura**. Astfel, citesc de la senzor temperatura masurata si o afisez pe ecran la locul specific acesteia.

```
temp_sensor.requestTemperatures();
lcd.setCursor(12, 1);
lcd.print(temp_sensor.getTempCByIndex(0));
```

2. Pentru controlul LED-urilor avem la dispozitie o telecomanda si o fotorezistenta. Astfel,

- Pentru **LED-urile albe**: Se citeste nivelul de lumina de la pinul A0, la care este conectat fotorezistenta, iar in functie de acesta LED-urile lumineaza mai tare (nivel de lumina scazut) sau mai incet (nivel de lumina ridicat). Aceste LED-uri albe pot fi si oprite din telecomanda, deci vor lumina conform conditiilor de mai sus **doar daca** sunt aprinse (a fost apasat butonul de turn on si variabila `is_on` a fost setata pe 1).

```
int value = analogRead(A0);
if (is_on == 1) {
  if (value < 150) { // Daca este lumina slaba in jur
    analogWrite(led_white, 250); // LED-ul lumineaza mai intens
  } else if (value >= 150 && value < 300) {
    analogWrite(led_white, 200);
  } else if (value >= 300 && value < 500) {
    analogWrite(led_white, 150);
  } else if (value >= 500 && value < 700) {
    analogWrite(led_white, 100);
  } else {
    analogWrite(led_white, 50);
  }
} else {
  analogWrite(led_white, 0);
}
```

- Asa cum am mentionat mai sus, din telecomanda aprind si sting LED-urile albe, dar pot sa controlez si **LED-urile RGB**, putand fie sa le sting de tot, fie sa aprind o anumita culoare (rosu/verde/albastru/portocaliu/galben/mov). Pentru a realiza acest lucru, citesc mai intai codul

receptionat de senzorul IR, iar in functie de acesta setez anumite valori pentru pinii LED-urilor. Pentru culorile primare (red, green, blue) folosesc digital write-uri, iar pentru celelalte folosesc analog write-uri.

```
if (IrReceiver.decode()) {
  code = String(IrReceiver.decodedIRData.decodedRawData, HEX);
  if (code == "fb04ef00") { // rosu
    digitalWrite(ledRGB_R, 0);
    digitalWrite(ledRGB_G, 1);
    digitalWrite(ledRGB_B, 1);
  } else if (code == "fa05ef00") { // verde
    digitalWrite(ledRGB_R, 1);
    digitalWrite(ledRGB_G, 0);
    digitalWrite(ledRGB_B, 1);
  } else if (code == "f906ef00") { // albastru
    digitalWrite(ledRGB_R, 1);
    digitalWrite(ledRGB_G, 1);
    digitalWrite(ledRGB_B, 0);
  } else if (code == "f708ef00") { // portocaliu
    analogWrite(ledRGB_R, 0);
    analogWrite(ledRGB_G, 200);
    analogWrite(ledRGB_B, 255);
  } else if (code == "f609ef00") { // galben
    analogWrite(ledRGB_R, 0);
    analogWrite(ledRGB_G, 120);
    analogWrite(ledRGB_B, 255);
  } else if (code == "f40bef00") { // mov
    analogWrite(ledRGB_R, 0);
    analogWrite(ledRGB_G, 255);
    analogWrite(ledRGB_B, 0);
  } else if (code == "f50aef00") { // turn OFF LED-uri RBG
    digitalWrite(ledRGB_R, 1);
    digitalWrite(ledRGB_G, 1);
    digitalWrite(ledRGB_B, 1);
  } else if (code == "fc03ef00") { // turn OFF LED-uri albe
    is_on = 0;
  } else if (code == "fd02ef00") { // turn ON LED-ruie albe
    is_on = 1;
  }
  IrReceiver.resume();
}
```

Rezultate Obținute

Intr-un final am reusit sa dau viata unui ceas de birou care arata in felul urmator:



Putem sa aprindem LED-urile albe ca sa lumineze in functie de lumina ambientala:





Sau putem sa aprindem LED-urile RGB si sa le setam ce culoare dorim:



Fun fact: un astfel de ceas poate ajunge si la 100 de lei. Ma gandesc de acum incolo sa imi realizez singura anumite gadget-uri 😊

Concluzii

Acest proiect a fost o adevarata provocare pentru mine, mai ales din pricina faptului ca nu am mai facut asa ceva pana acum si la inceput am fost chiar sperziata de ceea ce as putea sa fac.

Pe parcurs a inceput sa imi placa provocarile oferite de acesta, simtind ca ia nastere din mainile mele. Am inceput usor usor sa il construiesc, mai adaugand o componenta, mai scriind o linie de cod.

Am avut norocul de a functiona totul bine si de a nu avea probleme cu piesele comandate (bine, senzorul de temperatura mai avea putin si lua foc, dar nu din vina lui 😊).

In final, a fost o experienta pe care ma bucur ca am avut-o si care cu siguranta este un inceput daca voi vrea pe viitor sa ma mai ocup de ceva din acest domeniu.

Download

[proiect_pm_simona_alexe.zip](#)

Bibliografie/Resurse

- Senzor de temperatura:
 - <https://www.instructables.com/How-to-use-DS18B20-Temperature-Sensor-Arduino-Tuto/>
 - <https://forum.arduino.cc/t/ds18b20-temperature-sensor-using-onewire-library/699347>
- Senzor infrarosu:
 - <https://www.instructables.com/Using-Infrared-Sensor-With-Arduino/>
 - <https://github.com/Arduino-IRremote/Arduino-IRremote#converting-your-2x-program-to-the-4x-version>
- Fotorezistor:
 - <https://projecthub.arduino.cc/tropicalbean/how-to-use-a-photoresistor-1143fd>
- Ecran LCD:
 - https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615
 - <https://arduinogetstarted.com/tutorials/arduino-lcd-i2c>
- LED-uri:
 - <https://docs.arduino.cc/built-in-examples/basics/Blink/>
 - https://www.optimusdigital.ro/ro/optoelectronice-led-uri/484-led-rgb-anod-comun.html?search_query=led+rgb+anod+comun&results=2
 - <https://docs.arduino.cc/built-in-examples/basics/Fade/>
- Senzor RTC:
 - <https://www.instructables.com/DS1307-Real-Time-Clock-RTC-With-Arduino/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/mihaela.alex0502>



Last update: **2024/05/27 14:14**