MINISTERUL EDUCAȚIEI NAȚIONALE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL AUTOMATICA

SINTEZA
proiectului de laborator cu titlul:

DOZATOR DE MEDICAMENTE

Autor: Andreea Balan

1. Cerinţele temei:

Una dintre cele mai mari probleme pentru persoanele în vârstă sau bolnave este să își ia medicamentele la timp și în dozele corecte. Uitarea sau greșelile în administrarea medicamentelor pot duce la consecințe grave asupra sănătății, cum ar fi reacții adverse, complicarea afecțiunilor existente sau chiar deces. Pentru a remedia această problemă, un dozator automat de medicamente poate fi o soluție utilă. Dozatorul poate fi programat să elibereze medicamentele în timpul zilei, asigurând astfel o administrare precisă și la timp a acestora. Această tehnologie poate fi de ajutor în special pentru cei care au nevoie de o dozare strictă și regulată a medicamentelor și pentru persoanele în vârstă care pot avea dificultăți în a-și aminti să ia medicamentele.

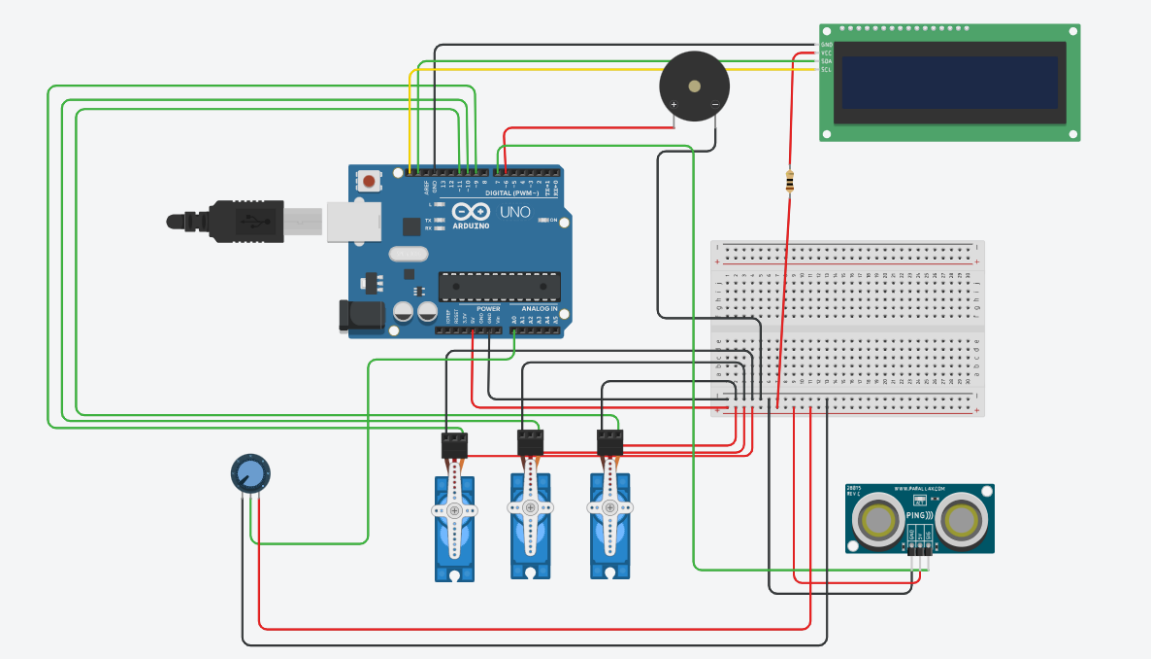
2. Soluţii alese:

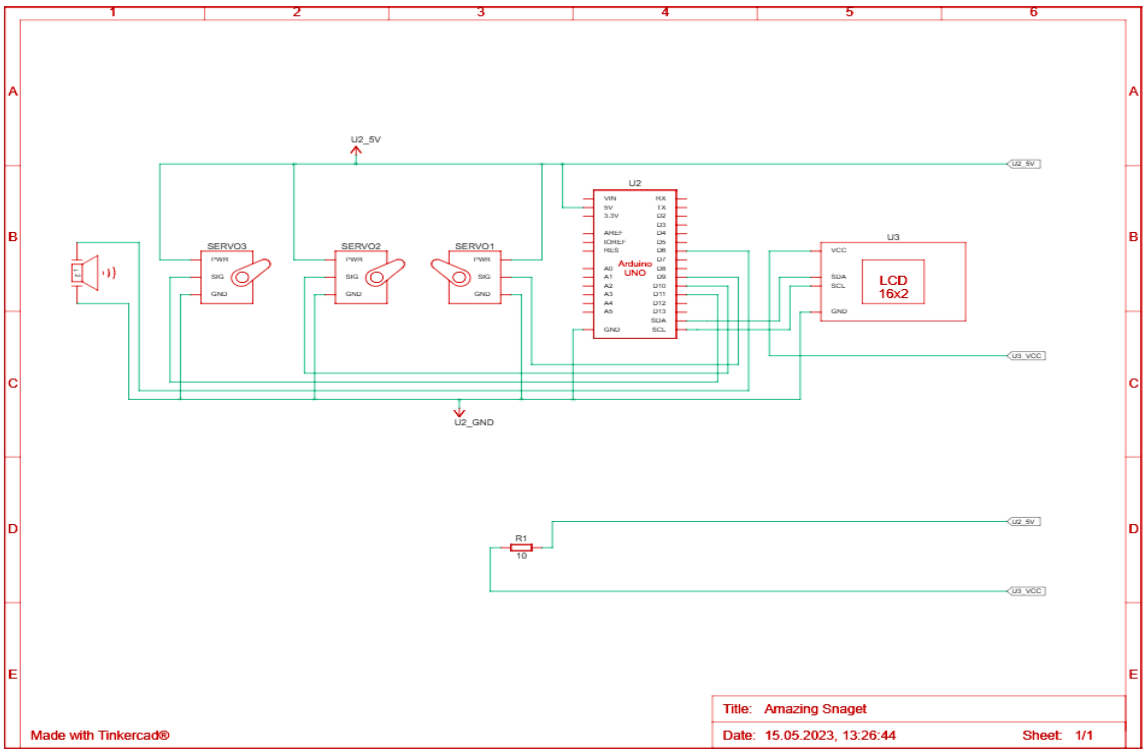
Proiectul se bazează pe un algoritm simplu. În primul rând, se memorează timpul de la pornirea plăcii Arduino, apoi se setează intervalul de timp la care trebuie luat fiecare medicament, dupa se verifica o data la 1 minut daca intervalul de timp trecut minus ultimul interval de timp la care s-a luat medicamentul este egal cu intervalul de timp dorit. Dacă această condiție este îndeplinită, se dă comanda spre Servo motorul medicamentului respectiv, acesta își schimbă poziția astfel încât să lase pastila să cadă si scoate un sunet pentru a atenționa utilizatorul, iar pe display este afișat numele medicamentului care a fost eliberat. Utilizatorul este atenționat de semnalul sonor, este sigur că ia medicamentul potrivit datorita afișajului de pe display și își ia tratamentul la timp, astfel procesul de a urma o prescripție de la medic este mult mai ușor.

3. Rezultate obţinute:

S-a obținut un sistem de dozat medicamente automat și eficient. Proiectul este ușor de instalat și de utilizat și poate fi personalizat în funcție de nevoile specifice ale utilizatorului care îl folosește. Schema generală a aplicației este prezentată într-o diagramă, iar fiecare componentă este detaliată într-o descriere tehnică.

- schema generală aplicației





- descriere a fiecărei componente implementate, la nivel de modul/procedura

  - **Arduino Uno** - Placa de dezvoltare Arduino Uno este utilizată ca microcontroler în această aplicație. Este programată să numere timpul de

la pornirea sa și să verifice condițiile necesare astfel încat să pornească Servo motoarele, buzzer-ul si display-ul.

- **Rezistență** - Rezistența este o componentă electrică utilizată pentru a limita curentul electric care trece prin display. Este conectată in serie cu display-ul si este folosită pentru a preveni arderea acestuia.

- **Servo motorul** - Un servo motor este un dispozitiv mecanic-electronic ce transformă semnalele electrice primite de la un controller intr-o mișcare mecanică precisă și controlată. În proiect sunt 3 Servo motoare care se mișcă la un unghi de 180̊ pentru a elibera medicamentul, apoi revine in pozitia initial.

- **Display** LCD I2C - este un afișaj cu cristale lichide cu interfață I2C, care permite transmiterea datelor prin intermediul a doar doi pini. Acest tip de display poate afișa text și grafice pe două linii cu un număr variabil de caractere pe linie, în funcție de model. În proiect display-ul afișează numele medicamentului care a fost eliberat.

- **Potențiometru** - este un instrument pentru variația potențialului electric; in circuit este folosit ca o substituție pentru un senzor de greutate, care ar indica cand dozatorului este gol si se afișeaza un mesaj corespunzator.

Name	Quantity	Component
U2	1	Arduino Uno R3
SERVO1 SERVO2 SERVO3	3	Positional Micro Servo
U3	1	MCP23008-based, 32 LCD 16 x 2 (I2C)
PIEZO1	1	Piezo
R1	1	10 Ω Resistor
PING1	1	Ultrasonic Distance Sensor
Rpot1	1	250 MΩ Potentiometer

4. Cod sursă

```
#include <Adafruit_LiquidCrystal.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

// Define servo pins
const int servoPin1 = 9;
const int servoPin2 = 10;
const int servoPin3 = 11;

// Define servo objects
Servo servo1;
Servo servo2;
Servo servo3;

// Define pill dispensing intervals
const int pill1DispenseInterval = 1;
const int pill2DispenseInterval = 2;
const int pill3DispenseInterval = 3;

//Last dispense
unsigned long pill1LastDispense = 0;
unsigned long pill2LastDispense = 0;
unsigned long pill3LastDispense = 0;

// Define buzzer pin
const int buzzerPin = 6;

// Initialize the display
Adafruit_LiquidCrystal lcd_1(0);

const int potentiometerPin = A0;

void setup() {
  // Attach servos to pins
  servo1.attach(servoPin1);
  servo2.attach(servoPin2);
  servo3.attach(servoPin3);

  // Set initial servo positions
  servo1.write(0);
  servo2.write(0);
  servo3.write(0);
```

```
  // Initialize buzzer pin
  pinMode(buzzerPin, OUTPUT);

   // Initialize the display
   lcd_1.begin(16,2);

  // Print message on the LCD
  lcd_1.setCursor(0, 0);
  lcd_1.print("Pill Dispenser");


   // Initialize the serial communication
  Serial.begin(9600);

  // Initialize the potentiometer pin as an input
  pinMode(potentiometerPin, INPUT);
}

void loop() {
  // Get current time
  unsigned long currentMillis = millis();
  int currentMinute = (currentMillis / 60000) % 60;

// Read the potentiometer value
  int potentiometerValue = analogRead(potentiometerPin);
// Print the potentiometer value to the serial monitor
  Serial.print("Potentiometer Value: ");
  Serial.println(potentiometerValue);

  // Check if it's time to dispense pill 1
  if (currentMinute - pill1LastDispense==pill1DispenseInterval &&
potentiometerValue!=0) {
    // Dispense pill 1
    servo1.write(180);
    delay(1000);
    servo1.write(0);

     // Play sound and display message
    tone(buzzerPin, 1000, 1000);
    lcd_1.setCursor(0,1);
    lcd_1.setBacklight(1);
    lcd_1.print("Pill 1 dispensed");
    delay(5000);
    lcd_1.clear();
```

```
    pill1LastDispense = currentMinute;
  }
else if(potentiometerValue==0)
    {
      lcd_1.setCursor(0,1);
      lcd_1.setBacklight(1);
      lcd_1.print("Dispenser is empty");
      delay(5000);
      lcd_1.clear();
    }

  // Check if it's time to dispense pill 2
  if (currentMinute - pill2LastDispense==pill2DispenseInterval &&
potentiometerValue!=0) {
    // Dispense pill 2
    servo2.write(180);
    delay(1000);
    servo2.write(0);

     // Play sound and display message
    tone(buzzerPin, 1000, 1000);
    lcd_1.setCursor(0,1);
    lcd_1.setBacklight(1);
    lcd_1.print("Pill 2 dispensed");
    delay(5000);
    lcd_1.clear();

    pill2LastDispense = currentMinute;
  }
else if(potentiometerValue==0)
    {
      lcd_1.setCursor(0,1);
      lcd_1.setBacklight(1);
      lcd_1.print("Dispenser is empty");
      delay(5000);
      lcd_1.clear();
    }

  // Check if it's time to dispense pill 3
  if (currentMinute - pill3LastDispense==pill3DispenseInterval &&
potentiometerValue!=0) {
    // Dispense pill 3
    servo3.write(180);
    delay(1000);
    servo3.write(0);
```

```
      // Play sound and display message
      tone(buzzerPin, 1000, 1000);
      lcd_1.setCursor(0,1);
      lcd_1.setBacklight(1);
      lcd_1.print("Pill 3 dispensed");
      delay(5000);
      lcd_1.clear();

      pill3LastDispense = currentMinute;
    }
  else if(potentiometerValue==0)
     {
       lcd_1.setCursor(0,1);
       lcd_1.setBacklight(1);
       lcd_1.print("Dispenser is empty");
       delay(5000);
       lcd_1.clear();
     }

   // Wait 1 minute before checking again
   delay(3600);

  //Reset timer after 24h
   if(currentMinute == 1440)
     currentMinute = 0;
     }
```

5. Testări şi verificări:

Pentru a utiliza aplicația, utilizatorul trebuie să o conecteze la alimentare și să modifice în cod intervalul în care se iau medicamentele după cum este necesar și să o lase să funcționeze în mod automat.

Data: 15.05.2023