

Digital Ruler

Introducere

Acest proiect constă într-o ruletă digitală dotată cu un senzor ultrasonic, construită cu ajutorul unei plăci Arduino Nano și cu un ecran LCD pentru afișarea distanței. Ruleta digitală utilizează senzorul ultrasonic pentru a măsura distanța între ruletă și obiectul detectat. Această distanță este afișată pe ecranul LCD în timp real. Scopul proiectului este de a oferi o modalitate precisă și ușor de utilizat pentru măsurarea distanțelor, oferind o alternativă digitală și interactivă la ruletele tradiționale. Ideea a pornit de la dorința de a combina tehnologia digitală cu conceptul de măsurare a distanțelor, oferind o soluție atât practică, având în vedere faptul că este o modalitate simplă și precisă de a măsura distanțe în diverse contexte, cât și distractivă pentru utilizatori.

Descriere generală



Ruleta digitală funcționează astfel:

Senzorul ultrasonic emite unde sonore către obiectul din față sa și măsoară timpul necesar pentru ca acesta unde să se întoarcă la senzor. Pe baza timpului de întoarcere a undelor sonore, se calculează distanța dintre senzor și obiectul detectat. Această distanță este apoi afișată pe ecranul LCD atașat la placa Arduino Nano. Utilizatorul poate citi distanța direct de pe ecranul LCD în timp real.

Hardware Design

Piese pe care le folosesc pentru acest proiect sunt:

- * o plăcuță Arduino Nano
- * un senzor ultrasonic HC-SR04
- * un display LCD
- * male & female header

* un potențiomtru

* un conector - 12V

Schema electrică

Schema a fost realizată pe platforma [TinkerCad](#).



Software Design

Acest proiect a fost implementat folosind [Arduino IDE](#).

Am folosit o singură bibliotecă, și anume [Liquid Crystal](#), cu ajutorul căreia se realizează conexiunea cu display-ul LCD.

Codul folosit pentru proiect este următorul:

```
#include <LiquidCrystal.h>

// Inițializăm biblioteca LiquidCrystal cu pinii conectați la LCD
LiquidCrystal lcd(9, 8, 4, 5, 6, 7); // Pinii Arduino (RS = 9, E = 8, D4 = 4,
D5 = 5, D6 = 6, D7 = 7)

#define Trig 2 // Pinul pentru semnalul de declanșare (Trig) al sensorului
HC-SR04
#define Echo 3 // Pinul pentru semnalul de ecou (Echo) al sensorului
HC-SR04

// Setup-ul pentru ecranul LCD
void setup() {
  lcd.begin(16, 2); // Configurăm ecranul LCD cu 16 coloane și 2 rânduri

  // Configurăm pinul Trig ca ieșire utilizând registre
  DDRD |= (1 << DDR2); // Setăm bitul corespunzător pinului 2 (Trig) ca
ieșire

  // Configurăm pinul Echo ca intrare utilizând registre
  DDRD &= ~(1 << DDR3); // Resetăm bitul corespunzător pinului 3 (Echo)
pentru a-l seta ca intrare

  // Afișăm mesajul de introducere (bun venit) pe ecranul LCD
  lcd.setCursor(2, 0);
  lcd.print("SA MASURAM");
  lcd.setCursor(4, 1);
```

```
lcd.print("DISTANTA!");
delay(2000); // Așteptăm 2 secunde pentru a permite citirea mesajului
lcd.clear();
}

// Trimitem un impuls de declanșare către senzorul ultrasonic
void loop() {
  // Setăm pinul Trig la LOW pentru 4 microsecunde
  PORTD &= ~(1 << PORTD2); // Setăm pinul Trig la LOW
  delayMicroseconds(4);

  // Setăm pinul Trig la HIGH pentru 10 microsecunde
  PORTD |= (1 << PORTD2); // Setăm pinul Trig la HIGH
  delayMicroseconds(10);

  PORTD &= ~(1 << PORTD2); // Setăm pinul Trig înapoi la LOW

  // Măsurăm durata semnalului de ecou
  long t = pulseIn(Echo, HIGH);

  // Calculăm distanța în cm și inch
  long cm = t / 58;
  long inch = t / 148;

  // Afișăm distanța măsurată pe ecranul LCD în centimetri
  lcd.setCursor(0, 0);
  lcd.print("Distanța: ");
  lcd.print(cm);
  lcd.print("cm ");

  // Afișăm distanța măsurată pe ecranul LCD în inch
  lcd.setCursor(0, 1);
  lcd.print("Distanța: ");
  lcd.print(inch);
  lcd.print("inch ");
}
```

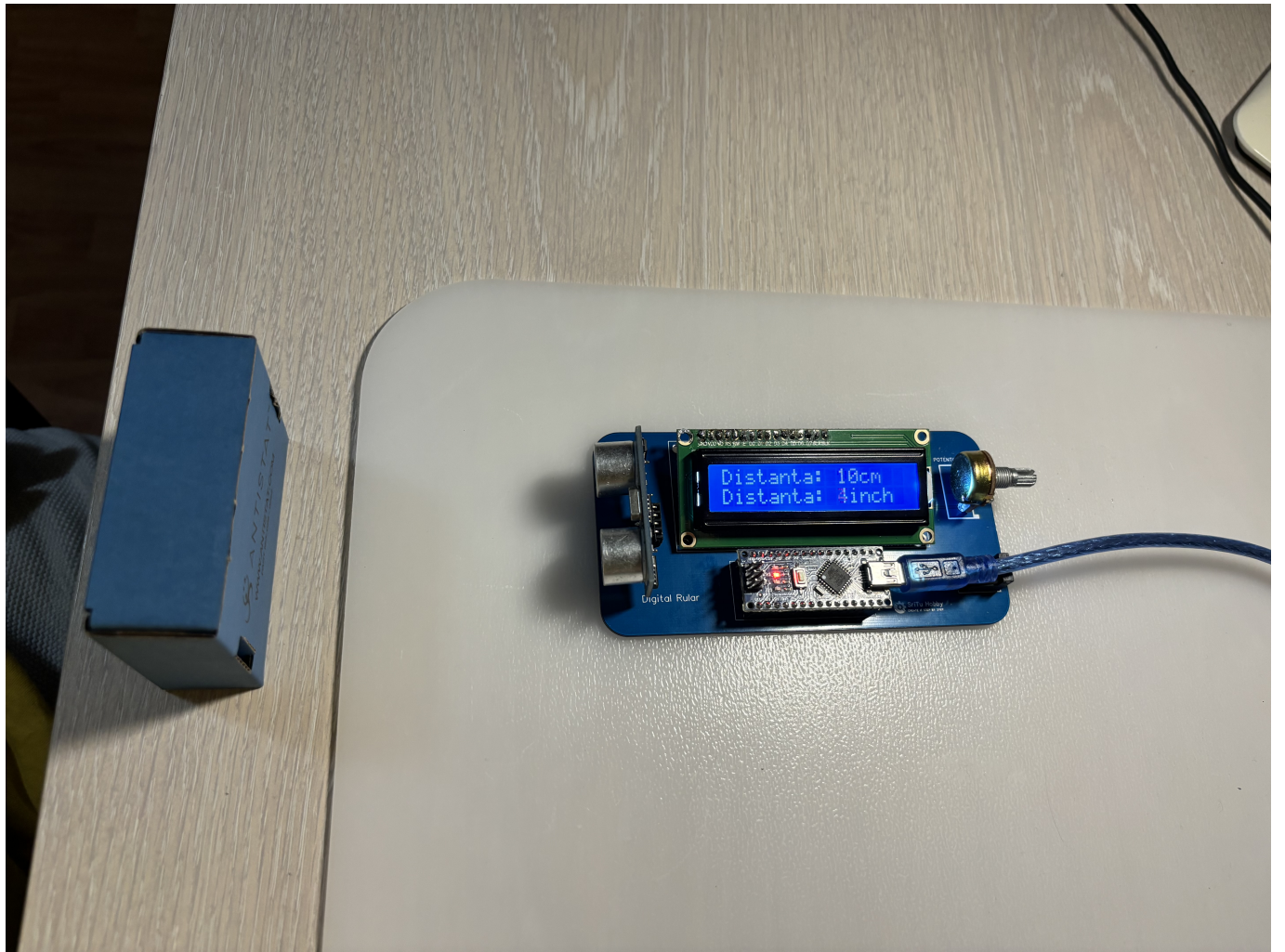
Rezultate Obținute

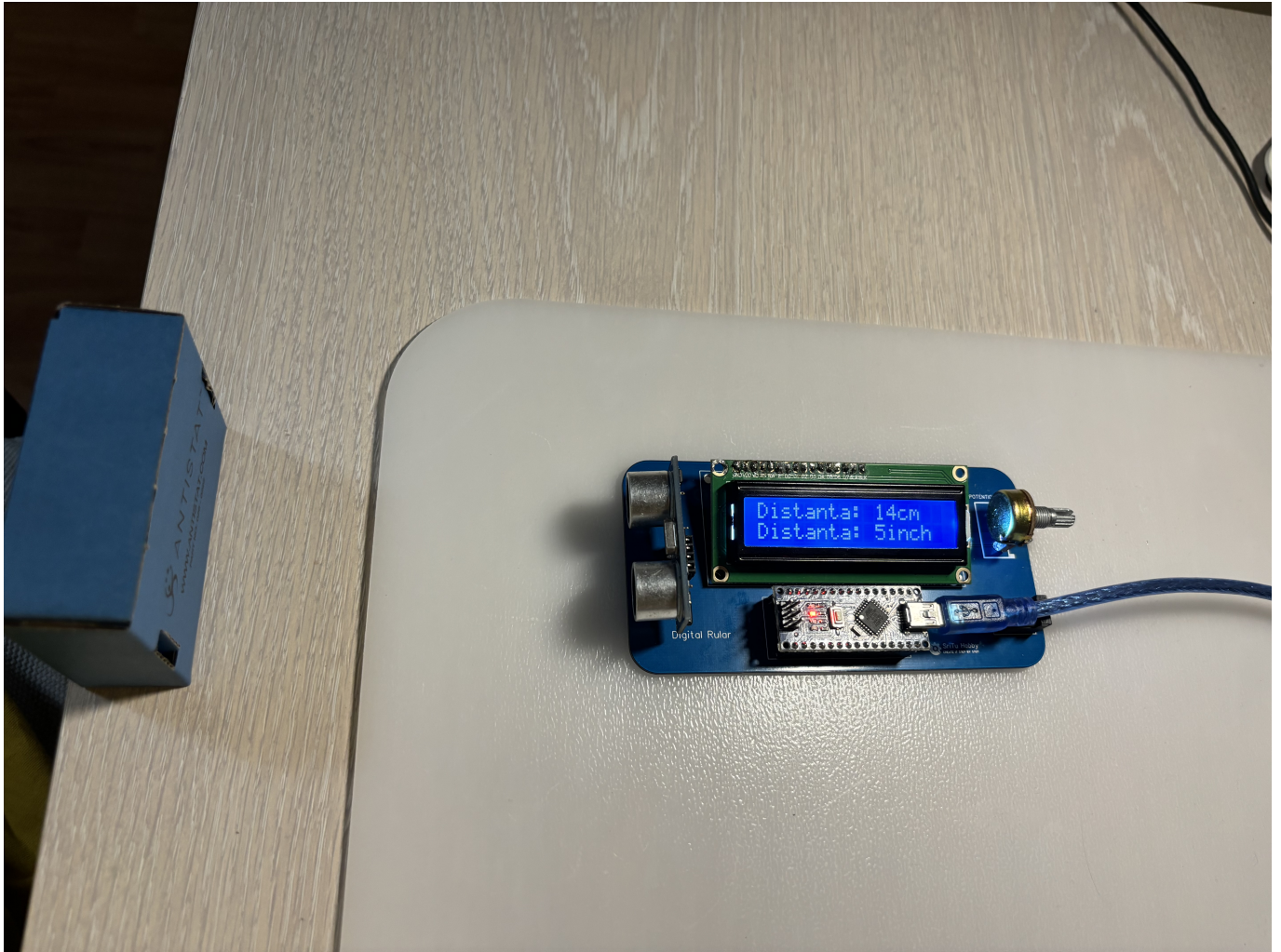
În urma implementării, ruleta digitală funcționează conform așteptărilor.

[Vedeți aici](#) un videoclip în care arăt funcționalitatea acesteia.









Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/silviu_vlad.cretu



Last update: **2024/05/28 19:23**