



# DOCUMENTAȚIE MT

Profesor coordonator : Floroian Laura

Nume : Ivascu Andrei Iulian

Programul de studii : Automatica

Grupa : 4LF412

An : 2022-2023

## Cuprins

1.Introducere .....	2
2.Arhitectură .....	2
3.Realizarea părții HW .....	5
4.Realizarea părții SW .....	6
5.Ansamblu final .....	11
6.Anexe .....	12

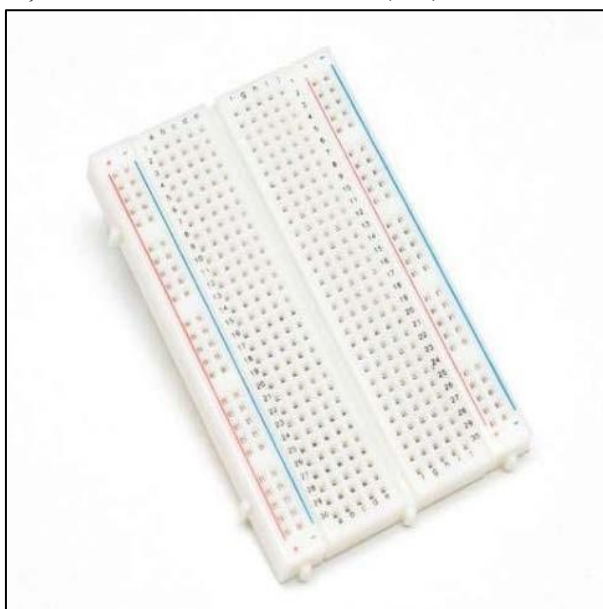
# 1.Introducere

Ideea a pornit de la realizarea unui senzor folosind Arduino Uno, care masoara viteza unui obiect care trece prin fata senzorului.

Această idee pe care am pus-o în aplicare a fost realizata cu succes şi am avut nevoie de următoarele componente: Arduino Uno, IR Sensor x 2, 16×2 LCD Display, 100R Resistor, 4.7k Resistor, 1k Resistor, Male to Male Jumper Wires, Battery 9v, Battery clip şi nişte linii de cod scrise in IDE-ul de Arduino.

## 2.Arhitectură

Pentru început, aveţi nevoie de un Breadboard (2.1).



*Fig.2. 1 Breadboard*

După care, aveţi nevoie de o plăcuţă Arduino (2.2), o baterie de 9V (2.3) , un 2 senzori de tip IR (2.4), un LCD Display 16x2 (2.5), niste cabluride tip Male to Male Jumper (2.6), si clema bateriei (2.7).

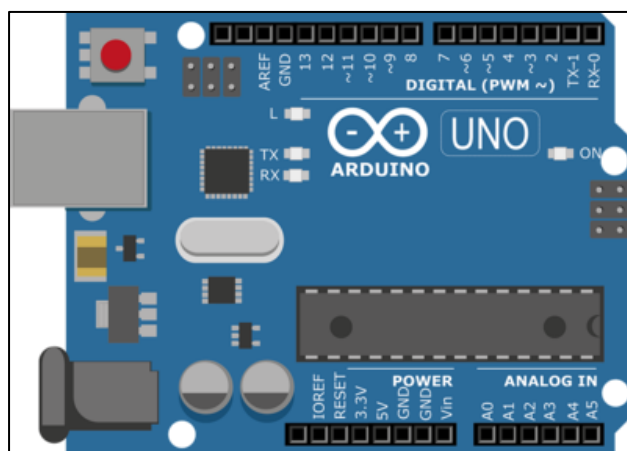


Fig.2. 2 Plăcuță Arduino

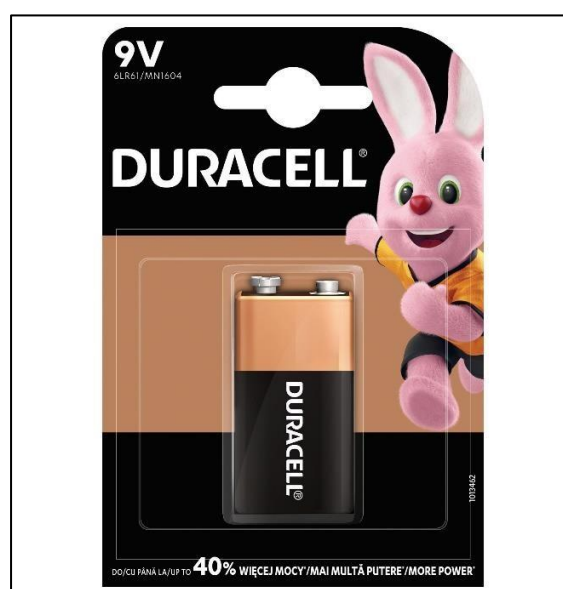


Fig.2. 3 Baterie 9V



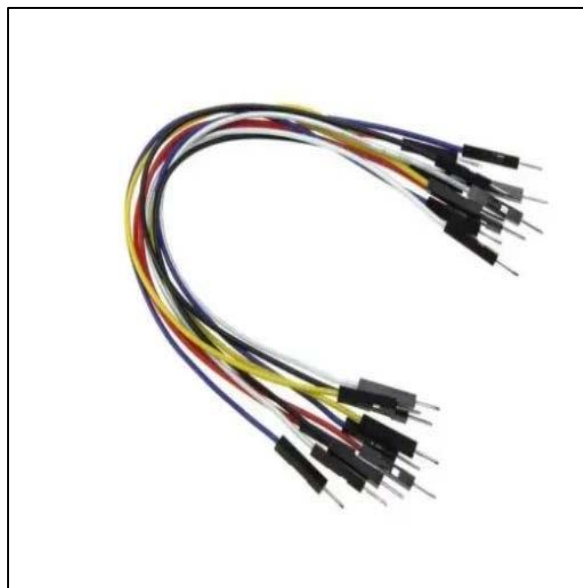
Fig.2. 4 IR Sensor

4



Fig.2. 5 16x2 LCD Display

Fig.2. 6 Male-Male Jumper Wires



*Fig.2. 7 Battery clip*



5

### 3.Realizarea părţii HW

Pentru realizarea partii HW, am atasat mai jos o imagine cu circuitul pe care se poate vedea întreaga arhitectura a proiectului.

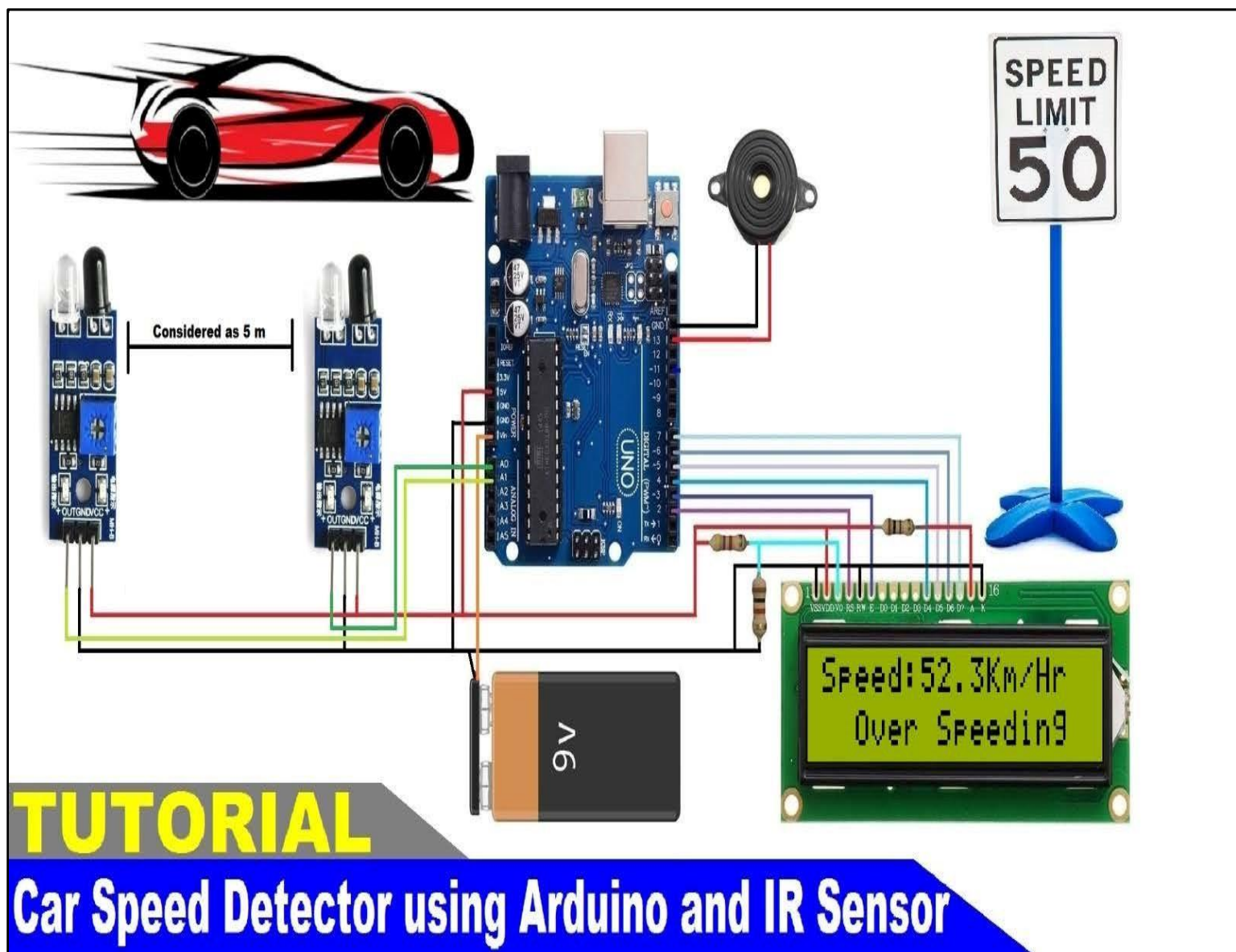


Fig 2.8

## 4.Realizarea părţii SW

Mai jos este atasat intreg codul care a fost folosit pentru Arduino IDE .



```
#include<LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

int timer1;
int timer2;
float Time;

int flag1 = 0;
int flag2 = 0;

float distance = 5.0;
float speed;

int ir_s1 = A0;
int ir_s2 = A1;
int buzzer = 13;

void setup(){
  pinMode(ir_s1, INPUT);
  pinMode(ir_s2, INPUT);
  pinMode(buzzer, OUTPUT);

  lcd.begin(16,2); lcd.clear();
  lcd.setCursor(0,0); lcd.print("
Masurari si traductoare ");
  lcd.setCursor(0,1);
  lcd.print("Masurator de viteza");
  delay(2000); lcd.clear();
}
```

7

```
void loop() {
  if(digitalRead (ir_s1) == LOW && flag1==0){timer1 = millis(); flag1=1;}

  if(digitalRead (ir_s2) == LOW && flag2==0){timer2 = millis(); flag2=1;}

  if (flag1==1 && flag2==1){    if(timer1 >
timer2){Time = timer1 - timer2;} else if(timer2 >
timer1){Time = timer2 - timer1;}
Time=Time/1000;//convert millisecond to second
```



```
speed=(distance/Time);//v=d/t
speed=speed*3600;//multiply by seconds per hr
speed=speed/100000;//division by cm per Km
}

if(speed==0){ lcd.setCursor(0, 1); if(flag1==0 &&
flag2==0){lcd.print("Nimic detectat..");}
else{lcd.print("Calculeaza viteza... ");}
}
else{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Viteza:");
    lcd.print(speed,1);
    lcd.print("Km/H ");
    lcd.setCursor(0, 1);
    if(speed > 5){lcd.print(" Limita Depasita "); digitalWrite(buzzer, HIGH);}
    else{lcd.print(" Viteza Normala "); }
    delay(3000);
    digitalWrite(buzzer, LOW);
    speed = 0;
    flag1 = 0;
    flag2 = 0;
}
}
```



## 5. Ansamblu final

În imaginea de mai jos, se poate vedea asamblarea finală a proiectului.

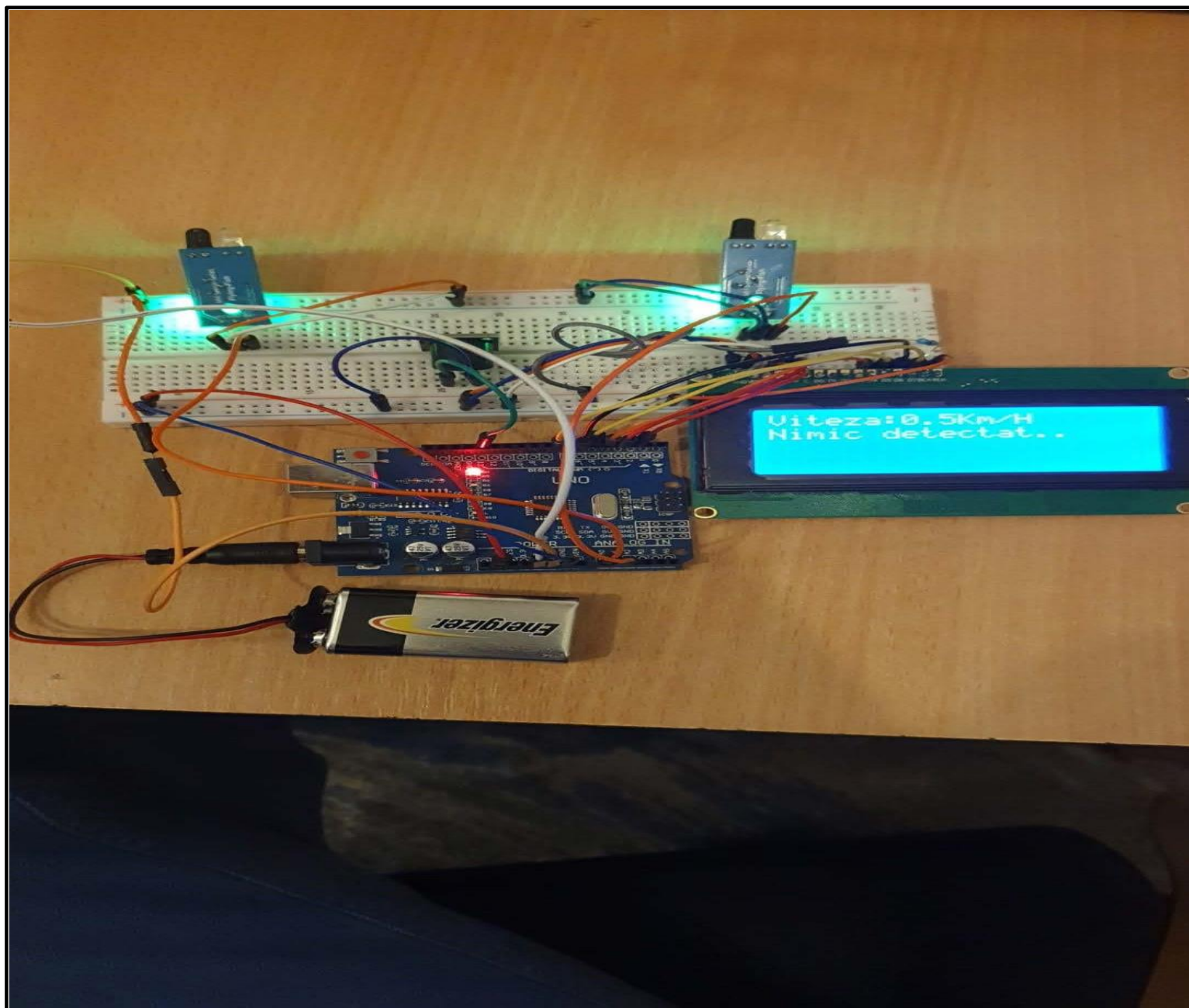


Fig. 2.9



## 6. Anexe

1. <https://www.arduino.cc/en/software>
2. <https://www.youtube.com/watch?v=n9TMs46DpGM>