Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrarea de laborator nr. 3.1

La disciplina “Internetul Lucrurilor”

**Tema:** **Senzori Achiziția de informații**

A efectuat: st. gr. SI-211 Adrian Chihai

A verificat: Valentina Astafi

**Chișinău – 2024**

**Definire Problema**

Sa se realizeze o aplicație in baza de MCU care va prelua un semnal de la o sursa de semnal, si va afișa parametrul fizic la un terminal (LCD si/sau Serial).   
Fiecare student va selecta un senzor fie analogic fie digital (nu binar) din PDF atașat sau: http://www.37sensors.com/

* Sa se achiziționeze semnalul de la senzor;
* Să se afișeze datele pe afișor LCD și / sau Serial.

**2 Procesul realizării lucrării**

**2.1 Descrierea funcțiilor**

 **measureDistance()**:

* Această funcție este utilizată pentru a colecta date de la senzorul ultrasonic HC-SR04.
* Ea declanșează un impuls ultrasonic prin pinul **Trig** și măsoară durata semnalului reflectat pe pinul **Echo**.
* Durata semnalului este convertită în distanță folosind formula: Distanța˘ (cm)=Durata˘ (µs)×0.0342\text{Distanță (cm)} = \frac{\text{Durată (µs)} \times 0.034}{2}Distanța˘ (cm)=2Durata˘ (µs)×0.034​
* Returnează distanța măsurată în centimetri.

 **setup()**:

* Funcția inițializează pinii **Trig** și **Echo** ai senzorului ultrasonic.
* Inițializează afișajul LCD I2C și afișează mesajul „Distanta:” pe primul rând al ecranului.
* Configurează portul Serial pentru a putea trimite date către monitorul serial.

 **loop()**:

* Este bucla principală a programului, care rulează continuu.
* Măsoară distanța utilizând funcția measureDistance().
* Afișează valoarea distanței măsurate atât pe LCD, cât și în monitorul serial.
* Include o întârziere de 500 ms între măsurători pentru a evita actualizările prea rapide.

 **LCD Initialization (lcd.init() și afișare)**:

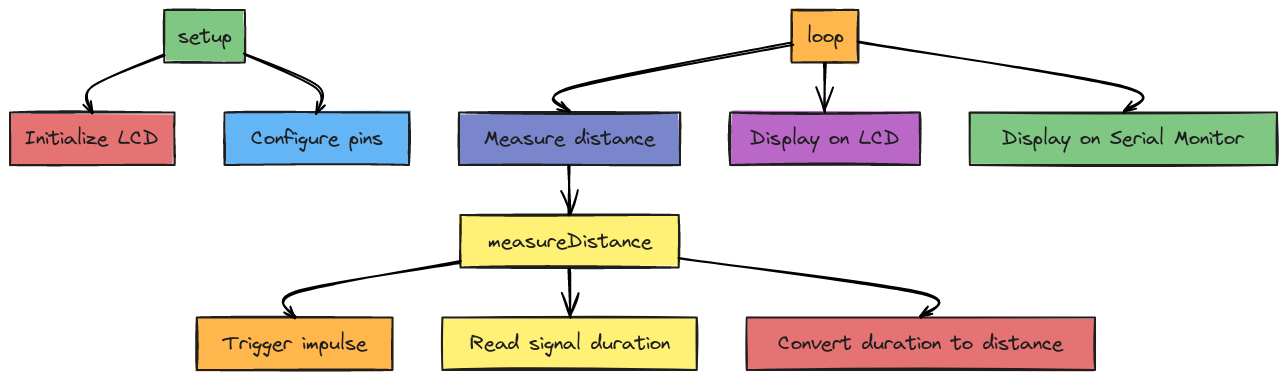
* Folosește biblioteca LiquidCrystal\_I2C pentru a controla afișajul LCD prin interfața I2C.
* Activează iluminarea de fundal a LCD-ului și setează cursorul pe rândul corespunzător pentru a afișa mesajele.

 **Delay (delay(500))**:

* Introduce o întârziere de 500 ms între măsurători pentru a limita frecvența de actualizare a afișajului și a monitorului serial.

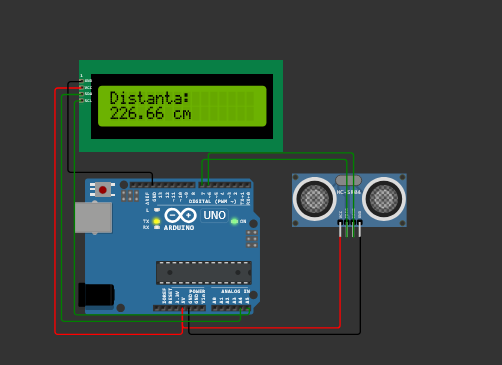
**2.2 Diagrama programului**

Figura 2.1 reprezintă fluxul și interacțiunea dintre configurare, bucla principală și funcțiile de măsurare a distanței într-un program Arduino care utilizează un senzor HC-SR04 și un LCD pentru afișarea rezultatelor.

**Figura 2.1** Flow-ul programului

**2.2 Circuitul programului**

În figura 2.2 este reprezentat circuitul pentru achiziția informației cu ajutorul senzorului HC-SR04



**Fig. 2.2** Circuitul asamblat si pornit

**Concluzie**

În cadrul laboratorului am folosit conceptele IoT utilizând un senzor cu ultrasunete (HC-SR04) pentru a măsura distanțele și un LCD pentru a afișa rezultatele. Un microcontroler a fost programat pentru a capta și procesa datele senzorului, convertindu-le într-un parametru fizic măsurabil.

Proiectul a integrat hardware și software pentru a realiza o achiziție eficientă a datelor, calcularea distanței folosind formule de unde sonore și afișarea rezultatelor pe un LCD și un monitor serial.

**Anexa 1**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

// Configurare LCD (adresă I2C, 16 coloane, 2 rânduri)

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Pini pentru senzorul HC-SR04

const int trigPin = 7;

const int echoPin = 6;

// Variabilă pentru stocarea distanței

float distance = 0.0;

void setup() {

  // Configurare pini HC-SR04

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  // Inițializare LCD

  lcd.init(); // Inițializează LCD-ul

  lcd.backlight();

  lcd.setCursor(0, 0);

  lcd.print("Distanta:");

  // Inițializare Serial Monitor

**Serial**.begin(9600);

}

void loop() {

  // Măsurarea distanței

  distance = measureDistance();

  // Afișare pe LCD

  lcd.setCursor(0, 1);

  lcd.print(distance);

  lcd.print(" cm    "); // Spațiu pentru a șterge eventuale caractere rămase

  // Afișare în Serial Monitor

**Serial**.print("Distanta: ");

**Serial**.print(distance);

**Serial**.println(" cm");

  delay(500); // Așteptare 500 ms înainte de următoarea măsurare

}

float measureDistance() {

  // Declanșare impuls pe TRIG

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  // Citire durată semnal de pe ECHO

  long duration = pulseIn(echoPin, HIGH);

  // Conversie durată în distanță (în cm)

  float distance = (duration \* 0.034) / 2;

  return distance;

}

**BIBLIOGRAPHY**

1. RANDOM NERD TUTORIALS: Guide for DS18B20 Temperature Sensor with Arduino. Copyright © 2013-2024 RandomNerdTutorials.com [quote 15.03.2024]. Access link:https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/