

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**  
**Universitatea Tehnică a Moldovei**  
**Facultatea Calculatoare, Informatică și Microelectronică**  
**Departamentul Ingineria Software și Automatică**  
**Programul de studii: Tehnologia informației**



## **RAPORT**

**Disciplina „IoT – Internetul Lucrurilor”**  
**Tema: Achiziția de informații.**

**Student(ă):** \_\_\_\_\_ **Vlașitchi Ștefan , TI-212**

**Coordonator universitate:** \_\_\_\_\_ **Lupan Cristian, asist.univ.**

**Chișinău, 2024**

**Scopul:** 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).

**Obiective:**

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

**Introducere**

Internetul Lucrurilor (IoT) reprezintă o tehnologie emergentă care facilitează interconectarea dispozitivelor fizice prin intermediul internetului, permițând astfel schimbul de date și controlul automatizat al acestora. În cadrul acestui laborator, se urmărește implementarea unei aplicații bazate pe un microcontroler (MCU) care să preia semnale de la senzori și să afișeze parametrii fizici corespunzători pe un terminal serial și/sau un afișaj LCD. Această interacțiune cu utilizatorul este esențială în prototipuri și testarea aplicațiilor IoT, oferind o metodă eficientă de monitorizare a datelor de la senzori în timp real.

Obiectivul principal al acestui laborator este de a implementa o aplicație simplă pentru achiziția și afișarea datelor de la senzori, utilizând un senzor de lumină (LDR) și un senzor de temperatură și umiditate (DHT22). Aceste date vor fi prelucrate și afișate atât pe un terminal serial, cât și pe un afișaj LCD, oferind o interfață ușor de utilizat pentru monitorizarea parametrilor de mediu.

**Materiale utilizate**

- Microcontroler Arduino UNO – Microcontrolerul principal utilizat pentru achiziția și prelucrarea datelor de la senzori.
- Senzor de lumină (LDR) – Un senzor analogic utilizat pentru măsurarea intensității luminii (în LUX).
- Senzor DHT22 – Un senzor digital utilizat pentru măsurarea temperaturii și umidității.
- Cabluri de conectare și rezistențe
- Componente necesare pentru conectarea senzorilor și a afișajului la microcontroler.
- Simulator Wokwi – Simulator utilizat pentru validarea funcționalității aplicației.

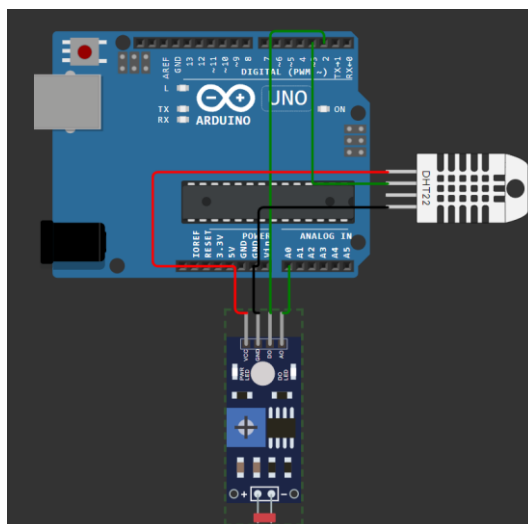
**Metodologie**

1. Achiziția semnalului de la senzori:
  - LDR: Valorile analogice preluate de la senzorul LDR sunt convertite în tensiune (V) și ulterior în intensitate a luminii (LUX), utilizând o funcție de conversie specifică.
  - DHT22: Valorile de temperatură și umiditate sunt citite digital, folosind biblioteci dedicate senzorului DHT22.
2. Conversia semnalului:

- Conversia semnalului analogic de la LDR se face folosind ADC-ul microcontrolerului pentru a obține o valoare de tensiune, iar apoi aceasta este transformată în LUX.
3. Afișarea datelor:
- Datele obținute de la senzori sunt afișate atât pe un afișaj LCD, cât și pe terminalul serial al Arduino, pentru a permite utilizatorului monitorizarea în timp real.
4. Implementarea interfeței de utilizator:
- Comunicarea cu utilizatorul se face prin intermediul afișajului LCD și al interfeței seriale, oferindu-se informații clare despre lumina ambientală, temperatura și umiditatea curente.
5. Simulare și testare:
- Aplicația este simulată pe platforma Wokwi, permițând validarea corectitudinii conexiunilor și a funcționalităților înainte de implementarea fizică.

## Rezultate

Pentru acest laborator am creat o aplicație pentru MCU care primește informația de la careva senzori achiziția și afișarea datelor de la senzori într-un sistem IoT simplu. Utilizarea senzorilor LDR și DHT22 a permis monitorizarea parametrilor de mediu în timp real, iar implementarea interfeței seriale a oferit utilizatorului o metodă ușoară și rapidă de a vizualiza aceste date (Vezi Figura 1).



**Figura 1 – Schema circuitului Arduino**

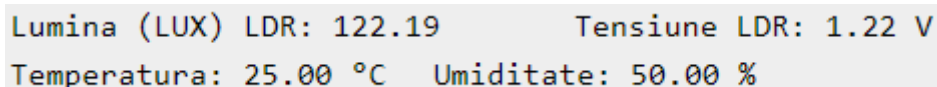
### Fragment cod sursa **main.ino**

```
void loop() {  
  // Citirea valorii analogice de la LDR  
  analogValue = analogRead(analogPin);  
  // Conversia ADC la tensiune (5V pentru Arduino)  
  tensiuneLDR = analogValue * (5.0 / 1023.0);  
  // Conversia tensiunii LDR la un parametru fizic (intensitatea luminoasă în lux)  
  lux = tensiuneLDR * 100; // Exprimat în lux (simplificat)  
  
  // Citirea datelor de la DHT22  
  umiditate = dht.readHumidity();  
  temperatura = dht.readTemperature();  
  
  // Verificăm dacă citirea senzorului DHT22 a avut succes  
  if (isnan(umiditate) || isnan(temperatura)) {  
    printf("Eroare la citirea DHT22!\n");  
  } else {  
    // Conversia valorilor float la șiruri de caractere pentru a fi afișate corect  
    dtostrf(lux, 6, 2, buffer); // Convertim lux-ul în șir  
    printf("Lumina (LUX) LDR: %s\t", buffer);  
  
    dtostrf(tensiuneLDR, 6, 2, buffer); // Convertim tensiunea LDR în șir  
    printf("Tensiune LDR: %s V\n", buffer);  
  
    dtostrf(temperatura, 6, 2, buffer); // Convertim temperatura în șir  
    printf("Temperatura: %s °C\t", buffer);  
  
    dtostrf(umiditate, 6, 2, buffer); // Convertim umiditatea în șir  
    printf("Umiditate: %s %%\n", buffer);  
  }  
  
  delay(2000); // Pauză de 2 secunde între citiri  
}
```

În cadrul acestui laborator, senzorii au fost integrați cu microcontrolerul pentru a prelua și afișa în timp real parametri fizici. Senzorul LDR a fost conectat la un pin analogic al microcontrolerului pentru a măsura intensitatea luminii ambientale, iar senzorul DHT22 a fost conectat la un pin digital, fiind responsabil de citirea temperaturii și umidității aerului.

Semnalele analogice de la LDR au fost preluate prin intermediul convertorului analog-digital (ADC) al microcontrolerului, transformându-se inițial într-o valoare de tensiune, care ulterior a fost convertită în intensitatea luminii, exprimată în LUX. În cazul senzorului DHT22, valorile de temperatură și umiditate au fost citite direct, utilizând o bibliotecă specifică pentru acest tip de senzor, fără a necesita conversii suplimentare.

Datele obținute de la senzori au fost afișate pe terminalul Serial Monitor al Arduino. Aceasta a permis utilizatorului acces facil la informațiile monitorizate. Pe Serial Monitor au fost afișate valorile de lumină în LUX, temperatura și umiditatea, împreună cu tensiunea măsurată de la senzorul LDR, asigurând astfel o vizualizare completă a parametrilor monitorizați (Vezi Fig. 2).



```
Lumina (LUX) LDR: 122.19      Tensiune LDR: 1.22 V  
Temperatura: 25.00 °C    Umiditate: 50.00 %
```

**Figura 2 – Rezultatul afișat pe Serial Monitor**

## Concluzie

Implementarea acestui laborator a demonstrat funcționarea corectă a unui sistem simplu de achiziție și afișare a datelor de la senzori, pe un microcontroler Arduino UNO. S-au obținut valori exacte pentru intensitatea luminii, temperatura și umiditatea mediului, care au fost prezentate în timp real pe un afișaj LCD și pe interfața serială.

Laboratoru a oferit o introducere esențială în lucrul cu senzori analogici și digitali, precum și în manipularea și afișarea datelor într-un sistem integrat IoT. Rezultatele au fost validate prin simularea corectă în Wokwi, ceea ce confirmă faptul că sistemul funcționează conform cerințelor specificate.

## Anexa A

### stdinout.h

```
#ifndef _STDINOUT_H
#define _STDINOUT_H

// no need to make an instance of this yourself
class initializeSTDINOUT
{
    static size_t initnum;
public:
    // Constructor
    initializeSTDINOUT();
};
```

```
// Call the constructor in each compiled file this header is included in
// static means the names won't collide
static initializeSTDINOUT initializeSTDINOUT_obj;

#endif
```

## **stdinout.cpp**

```
#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include <stdio.h>
#include "stdinout.h"

// Function that printf and related will use to print
static int serial_putchar(char c, FILE *f)
{
    if (c == '\n') {
        serial_putchar('\r', f);
    }

    return Serial.write(c) == 1 ? 0 : 1;
}

// Function that scanf and related will use to read
static int serial_getchar(FILE *)
{
    // Wait until character is available
    while (Serial.available() <= 0) { ; }

    return Serial.read();
}

static FILE serial_stdinout;

static void setup_stdin_stdout()
{
    // Set up stdout and stdin
    fdev_setup_stream(&serial_stdinout, serial_putchar, serial_getchar, _FDEV_SETUP_RW);
    stdout = &serial_stdinout;
    stdin = &serial_stdinout;
    stderr = &serial_stdinout;
}

// Initialize the static variable to 0
size_t initializeSTDINOUT::initnum = 0;

// Constructor that calls the function to set up stdin and stdout
initializeSTDINOUT::initializeSTDINOUT()
{
    if (initnum++ == 0) {
        setup_stdin_stdout();
    }
}
```

## **Anexa B**

### **Lab3.1.ino**

```
#include <DHT.h>
#include "stdinout.h"

// Definirea pinului pentru senzorul DHT22
#define DHTPIN 3
#define DHTTYPE DHT22
```

```

DHT dht(DHTPIN, DHTTYPE); // Crearea obiectului DHT pentru senzorul DHT22

// Definirea pinului pentru senzorul LDR
int analogPin = A0; // Pinul analogic pentru citirea semnalului AO de la LDR

int analogValue = 0;      // Valoare citită de la LDR
float tensiuneLDR = 0.0;  // Tensiune echivalentă LDR
float lux = 0.0;          // Intensitatea luminoasă în lux
float temperatura = 0.0;  // Temperatura citită de la DHT22
float umiditate = 0.0;    // Umiditatea citită de la DHT22

// Buffer pentru a stoca valorile convertite în string
char buffer[20];

void setup() {
    Serial.begin(9600);      // Inițializare Serial Monitor
    dht.begin();             // Inițializare senzor DHT22
    printf("Sistem initializat\n"); // Mesaj de confirmare pentru debug
}

void loop() {
    // Citirea valorii analogice de la LDR
    analogValue = analogRead(analogPin);

    // Conversia ADC la tensiune (5V pentru Arduino)
    tensiuneLDR = analogValue * (5.0 / 1023.0);

    // Conversia tensiunii LDR la un parametru fizic (intensitatea luminoasă în lux)
    lux = tensiuneLDR * 100; // Exprimat în lux (simplificat)

    // Citirea datelor de la DHT22
    umiditate = dht.readHumidity();
    temperatura = dht.readTemperature();

    // Verificăm dacă citirea senzorului DHT22 a avut succes
    if (isnan(umiditate) || isnan(temperatura)) {
        printf("Eroare la citirea DHT22!\n");
    } else {
        // Conversia valorilor float la șiruri de caractere pentru a fi afișate corect
        dtostrf(lux, 6, 2, buffer); // Convertim lux-ul în șir
        printf("Lumina (LUX) LDR: %s\t", buffer);

        dtostrf(tensiuneLDR, 6, 2, buffer); // Convertim tensiunea LDR în șir
        printf("Tensiune LDR: %s V\n", buffer);

        dtostrf(temperatura, 6, 2, buffer); // Convertim temperatura în șir
        printf("Temperatura: %s °C\t", buffer);

        dtostrf(umiditate, 6, 2, buffer); // Convertim umiditatea în șir
        printf("Umiditate: %s %%\n", buffer);
    }

    delay(2000); // Pauză de 2 secunde între citiri
}

```