

Ministerul Educației și Cercetării  
Universitatea Tehnică a Moldovei  
Facultatea Calculatoare, Informatică și Microelectronică  
Departamentul Ingineria Software și Automatică

# Raport

Curs: Internetul lucrurilor

Tema: Interacțiunea cu utilizatorul - Serial STDIO.  
LCD+Keypad STDIO.

A elaborat:

Chirita Stanislav

A verificat:

Astafi Valentina

Chișinău 2024

## Lucrare de laborator nr. 1

### Partea nr 1.1: Interactiunea cu utilizatorul: STDIO - Interfata seriala

#### Definire problema

1. Sa se configureze aplicatia pentru lucrul cu biblioteca STDIO prin interfata seriala pentru schimbul de text prin terminal.
2. sa se proiecteze o aplicatie in baza de MCU care ar receptiona comenzi de la terminal prin interfata seriala pentru a seta starea unui LED.
  - *led on* pentru aprindere si
  - *led off* pentru stingere.,
  - sistemul trebuie sa raspunda cu mesaje text despre confirmarea comenzii
  - pentru schimbul de text prin terminal a se utiliza biblioteca STDIO

#### Obiective:

- configurarea aplicației pentru lucrul cu biblioteca STDIO;
- crearea schemei și codului conform sarcinii propuse;
- înțelegerea sistemului creat și procesele ce se întâmplă.

#### Introducere

Problema abordată în acest studiu de caz este configurarea aplicațiilor pentru comunicarea cu un microcontroler (MCU) prin interfețe seriale pentru schimbul de text prin terminal. Scopul este de a permite utilizatorului să furnizeze comenzi prin intermediul terminalului pentru a controla starea unui LED conectat la MCU și să răspundă cu mesaje text care confirmă comenzile.

Domeniul relevant pentru această problemă este dezvoltarea aplicațiilor încorporate și comunicarea cu dispozitivele hardware folosind un microcontroler. În acest caz, ne vom concentra pe utilizarea interfețelor seriale pentru a permite comunicarea între un terminal și un MCU pentru a controla un LED.

În această lucrare sunt 2 probleme actuale și anume configurarea aplicației pentru lucrul cu biblioteca STDIO prin interfața serială și proiectarea aplicației pentru controlul unui LED bazat pe comenzi primite prin interfața serială. Pentru a realiza prima problemă este necesară o configurare corespunzătoare a aplicației pentru a utiliza biblioteca STDIO pentru a gestiona comunicarea serială cu terminalul. Aceasta implică înțelegerea

protocolului de comunicare serială și implementarea funcționalităților necesare pentru schimbul de texte între aplicație și terminal. Pentru a doua problemă este esențial să proiectăm și să implementăm aplicația astfel încât să poată interpreta comenzile primite de la terminal prin interfața serială și să controleze corespunzător starea LED-ului. Aceasta implică gestionarea comenzilor "led on" și "led off" și asigurarea unei răspunsuri adecvate prin mesaje text de confirmare. Un ultim lucru important este conexiunea cu o aplicație ce funcționează cu Arduino pentru a efectua și executa sistemul în real. Din referința [2] a fost analizat pașii de conectare cu aplicația Arduino IDE și rularea unui proiect de început [1].

#### Arduino:

- Arduino este o platformă hardware open-source bazată pe microcontrolere. Aceasta oferă un mediu de dezvoltare și un set de biblioteci pentru a crea proiecte electronice interactive.
- Microcontrolerele Arduino au pinuri digitale și analogice pe care puteți conecta diverse componente, cum ar fi LED-uri, senzori, ecrane, etc.
- Dezvoltarea cu Arduino implică scrierea codului în limbajul Arduino (o variantă a C/C++) și încărcarea acestuia pe placă folosind Arduino IDE sau altă aplicație similară.

#### Comunicarea Serială:

- Comunicarea serială este procesul de transmitere și recepție a datelor pe un singur fir, un bit la un moment dat. Unul dintre cele mai comune protocoale de comunicare serială este UART (Universal Asynchronous Receiver/Transmitter).
- Baud rate reprezintă viteza de comunicație în biți pe secundă și trebuie să fie setată corect pe ambele părți ale comunicării pentru a asigura o comunicare eficientă.
- Arduino folosește portul serial pentru a comunica cu un computer sau cu alte dispozitive externe, ceea ce permite monitorizarea și controlul proiectelor.

#### Biblioteca STDIO:

- Biblioteca STDIO oferă funcții pentru manipularea fluxurilor standard de intrare și ieșire, cum ar fi tastatura și ecranul, în limbajul C/C++.
- Funcții precum printf() permit formatarea și afișarea textului într-un mod flexibil și ușor de utilizat.
- scanf() poate fi folosită pentru citirea și interpretarea datelor de la utilizator.

#### Interfața cu Utilizatorul prin LCD:

- Un ecran LCD (Liquid Crystal Display) este un dispozitiv de afișare folosit pentru a prezenta informații într-un format text sau grafic.

- Interfața cu un LCD implică inițializarea ecranului, configurarea cursorului și afișarea de texte sau simboluri pe ecran.
- Biblioteci precum "LiquidCrystal" în Arduino facilitează comunicarea și controlul unui ecran LCD.

#### Lucrul cu Taste (Keypad):

- Un keypad sau tastatură este un dispozitiv cu multiple butoane sau taste care pot fi utilizate pentru a introduce date sau pentru a interacționa cu un sistem.
- Conectarea unui keypad la Arduino necesită alocarea pinurilor corespunzătoare și citirea valorilor butoanelor apășate.

#### Controlul unui LED:

- LED-urile (Light Emitting Diodes) sunt dispozitive electronice care emit lumină atunci când sunt alimentate cu curent electric.
- Controlul unui LED implică stabilirea stării sale, fie prin aplicarea unei tensiuni pentru aprindere (HIGH), fie prin eliminarea tensiunii pentru stingere (LOW).

#### Logica de programare și flux de control:

- Programarea implică dezvoltarea algoritmilor și logicii pentru a realiza sarcinile dorite.
- Folosirea structurilor de decizie (if/else) permite programului să ia decizii bazate pe condiții specifice, cum ar fi verificarea codului introdus sau controlul stării LED-urilor.

#### Conexiuni Hardware:

- Este esențial să înțelegeți cum să conectați corect componente hardware la placa Arduino, asigurându-vă că pinurile corespund și că legăturile sunt sigure și stabile.
- Documentația specifică a componentelor și a plăcii Arduino trebuie consultată pentru a efectua conexiunile corecte.

Pentru efectuarea sarcinii propuse vor fi necesare următoarele elemente pentru construirea circuitului:

1. Placa Arduino;
2. Un led, cazul dat galben;
3. Un rezistor;
4. Fire de conexiune.

Pașii de executare a creării sistemului:

1. Led-ul este conectat cu catodul ce e destinat pentru semnul minus la secțiunea GND – ground, iar celălalt capăt mai lung e anodul ce reprezintă + și e conectat la numărul 13 prin fire explicat pe pași mai jos;
2. De pe placa Arduino de la GND se duce la secțiune minus(-) un fir de conexiune, apoi de pe această linie se duce la poziția 14b astfel că următoarea poziție 15c să fie anodul led-ului;
3. De pe placă de la 13 se introduce un fir la poziția 9b trimitând energie secțiunii 10c; Toată descrierea creării circuitului descris este prezentată în figura 1.

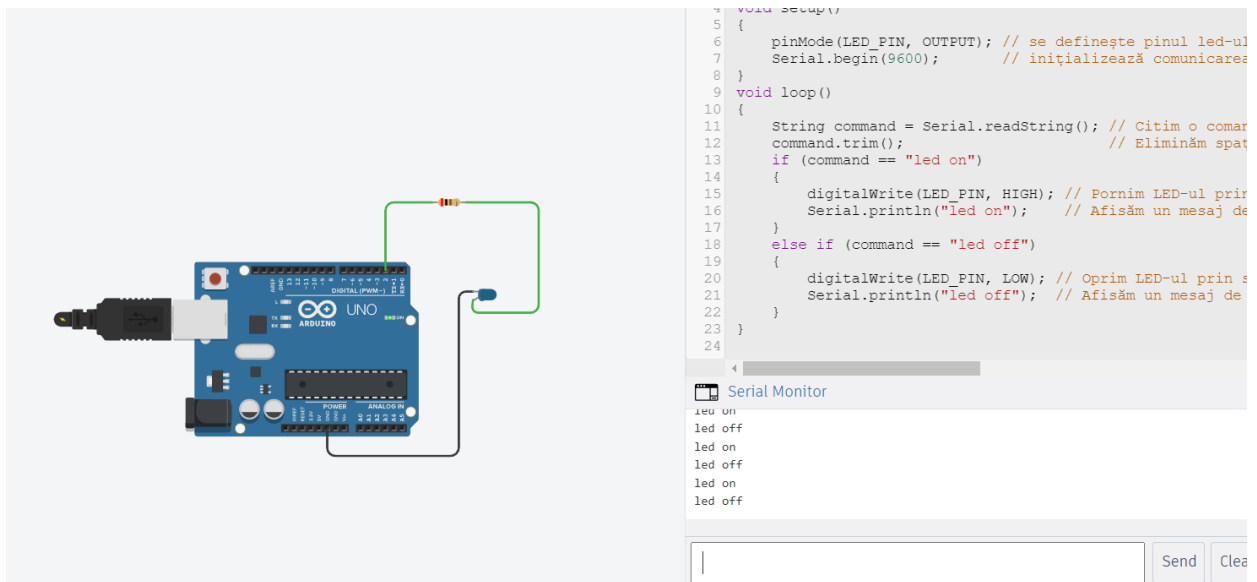


Figura 1 – Schema sarcinii 1

În continuare urmează codului pentru sarcina dată, unde fiecare linie va conține un comentariu despre rolul sau acțiunea pe care o exercită pentru îndeplinirea cerinței.

```
#include <stdio.h> // Include librăria stdio pentru funcțiile de input/output
#include <Arduino.h>
#define LED_PIN 13 // definește numărul LED-ului utilizat
// Funcție care redirecționează output-ul către serial
int serial_putc(char c, FILE *f) {
    Serial.write(c);
    return 0;
}
// Funcție care redirecționează input-ul din serial
int serial_getc(FILE *f) {
    while (!Serial.available()) {} // Așteaptă până sunt disponibile date
    return Serial.read();
}
// Declară fluxurile de I/O pentru redirecționare
FILE serial_stdout;
FILE serial_stdin;
void lab1_setup() {
    // Configurează pinul pentru LED ca ieșire
```

```

pinMode(LED_PIN, OUTPUT);

// Inițializează comunicația serială la 9600 baud rate
Serial.begin(9600);

// Asociază funcțiile de I/O cu fluxurile STDIO
fdev_setup_stream(&serial_stdout, serial_putc, NULL, _FDEV_SETUP_WRITE);
fdev_setup_stream(&serial_stdin, NULL, serial_getc, _FDEV_SETUP_READ);

// Setează stdin și stdout pentru fluxurile seriale
stdout = &serial_stdout;
stdin = &serial_stdin;

// Mesaj de confirmare că sistemul este gata
printf("Sistemul este gata. Scrie 'led on' sau 'led off'.\n");
}
void lab1_loop() {
    char command[20]; // Buffer pentru citirea comenzilor
    char state[5];
    // Așteaptă o comandă din partea utilizatorului
    scanf("%19s", command); // Citește o comandă de maxim 19 caractere
    scanf("%4s", state);

    strcat(command, " ");
    strcat(command, state);

    if (strcmp(command, "led on") == 0 ) {
        digitalWrite(LED_PIN, HIGH); // Aprinde LED-ul
        printf("LED-ul a fost aprins.\n");
    }
    else if (strcmp(command, "led off") == 0) {
        digitalWrite(LED_PIN, LOW); // Stingă LED-ul
        printf("LED-ul a fost stins.\n");
    }
    else {
        printf("Comanda necunoscută. Folosește 'led on' sau 'led off'.\n");
    }
}

```

Explicarea logicii executării codului.

- String command = Serial.readString(): citește date de pe portul serial și le stochează într-un obiect de tip String numit command;
- command.equalsIgnoreCase("led on"): Verifică dacă șirul de caractere stocat în variabila command este egal cu "led on", ignorând diferențele dintre litere mari și mici. Dacă șirul este "led on", se va executa funcția ledOn() care aprinde LED-ul.
- digitalWrite(13, HIGH);: Aceasta este instrucțiunea care aprinde efectiv LED-ul conectat la pinul 13, trimițând un semnal HIGH (tensiune mare) către acest pin.
- Serial.println("LED is ON");: Această linie trimite un mesaj înapoi prin portul serial pentru a confirma că LED-ul a fost aprins. Mesajul va apărea în monitorul serial.

- `else if (command.equalsIgnoreCase("led off"))`:: Aceasta este o altă condiție de control de flux care verifică dacă șirul `command` este egal cu `"led off"`. Dacă da, se va executa funcția `ledOff()`, care stinge LED-ul.
- `digitalWrite(13, LOW)`:: Aceasta este instrucțiunea care stinge LED-ul conectat la pinul 13, trimițând un semnal LOW (tensiune joasă) către acest pin.
- `Serial.println("LED is OFF")`:: Această linie trimite un mesaj înapoi prin portul serial pentru a confirma că LED-ul a fost stins. Mesajul va apărea în monitorul serial.
- `else { Serial.println("Unknown command. Use 'led on' or 'led off'"); }`: Dacă niciuna dintre condițiile `"led on"` sau `"led off"` nu este adevărată, codul intră în ramura `else`, afișând un mesaj care informează utilizatorul că comanda nu este validă și indică comenzile disponibile, `"led on"` sau `"led off"`.

## Partea nr 1.2 : Interacțiunea cu Utilizatorul: LCD + Keypad

### Definire problema

1. Sa se configureze aplicatia pentru lucrul cu libraia STDIO prin interfata seriala pentru schimbul de text prin LCD+Keypad.
2. sa se proiecteze o aplicatie in baza de MCU pentru detectarea unui cod de la o tastatura 4x4, sa verifice codul si sa afiseze mesaj la un LCD.
  - pentru cod valid sa se aprinda un led de culoare verde, pentru cod invalid, un led de culoare rosie.
  - A se utiliza STDIO pentru scanarea tastaturii si afisare la LCD.

### Obiective:

- configurarea aplicației pentru lucrul cu librăria STDIO;
- crearea schemei și codului conform sarcinii propuse;
- conectarea LCD și a Keypad-ului;
- înțelegerea sistemului creat și procesele ce se întâmplă.

### Introducere

Problema abordată în acest studiu de caz este configurarea aplicațiilor pentru comunicarea cu un microcontroler (MCU) prin interfețe seriale pentru schimbul de text prin terminal. Scopul este de a permite

utilizatorului să furnizeze comenzi prin intermediul terminalului pentru a controla starea unui LED conectat la MCU și să răspundă cu mesaje text care confirmă comenzile.

Domeniul relevant pentru această problemă include tehnologiile încorporate și dezvoltarea de aplicații pentru microcontrolere. Acesta implică înțelegerea principiilor de funcționare a tastaturilor matrice, a LCD-urilor și a interacțiunii cu acestea prin intermediul unui microcontroler. În plus, se explorează și utilizarea bibliotecii STDIO pentru gestionarea intrărilor și ieșirilor textuale.

În această lucrare sunt 3 probleme actuale și anume configurarea aplicației pentru lucrul cu biblioteca STDIO prin interfața serială, proiectarea aplicației pentru detectarea unui cod de la o tastatură 4x4 și primirea unui feedback vizual și textual cu ajutorul unui LCD. Pentru a realiza primei probleme este ajustarea codului pentru a utiliza librăria STDIO pentru scanarea tastaturii și afișarea rezultatelor pe un LCD. Acesta implică înțelegerea modului în care să se citească și să se proceseze datele de la tastatură folosind STDIO și apoi să se afișeze rezultatele pe un LCD. Pentru a doua problemă are loc proiectarea logicii pentru a citi și interpreta codul introdus de utilizator prin tastatură, apoi să verifice dacă acesta este valid sau nu. Această problemă necesită înțelegerea tastaturilor matrice și a metodelor de scanare a acestora. Ultima problema este despre implementarea unei funcționalități pentru a afișa rezultatele (cod valid sau invalid) pe un ecran LCD și a controla LED-urile în funcție de acest rezultat. Este necesară cunoașterea modului în care să se comunice cu un LCD și cum să se controleze LED-urile .

Pentru efectuarea sarcinii propuse vor fi necesare următoarele elemente pentru construirea circuitului:

1. Placa Arduino;
2. O tastatura 4x4;
3. Un LCD I2C;
4. Fire de conexiune.

Pașii de executare a creării sistemului:

1. Se conectează LCD în modul următor: de la GND de pe placă se duce un fir către secțiunea denumită GND la LCD, de la 5V de pe placă se duce la VCC de pe LCD, de la SDA și SCL de pe placă se conectează corespunzător cu cele de pe LCD;
2. Pentru tastatură se conectează patru pini pentru coloane și alți patru pini pentru rânduri, unde pe baza lor se va determina ce element de pe tastatură a fost apăsător;

Toată descrierea creării circuitului descris este prezentată în figura 4.



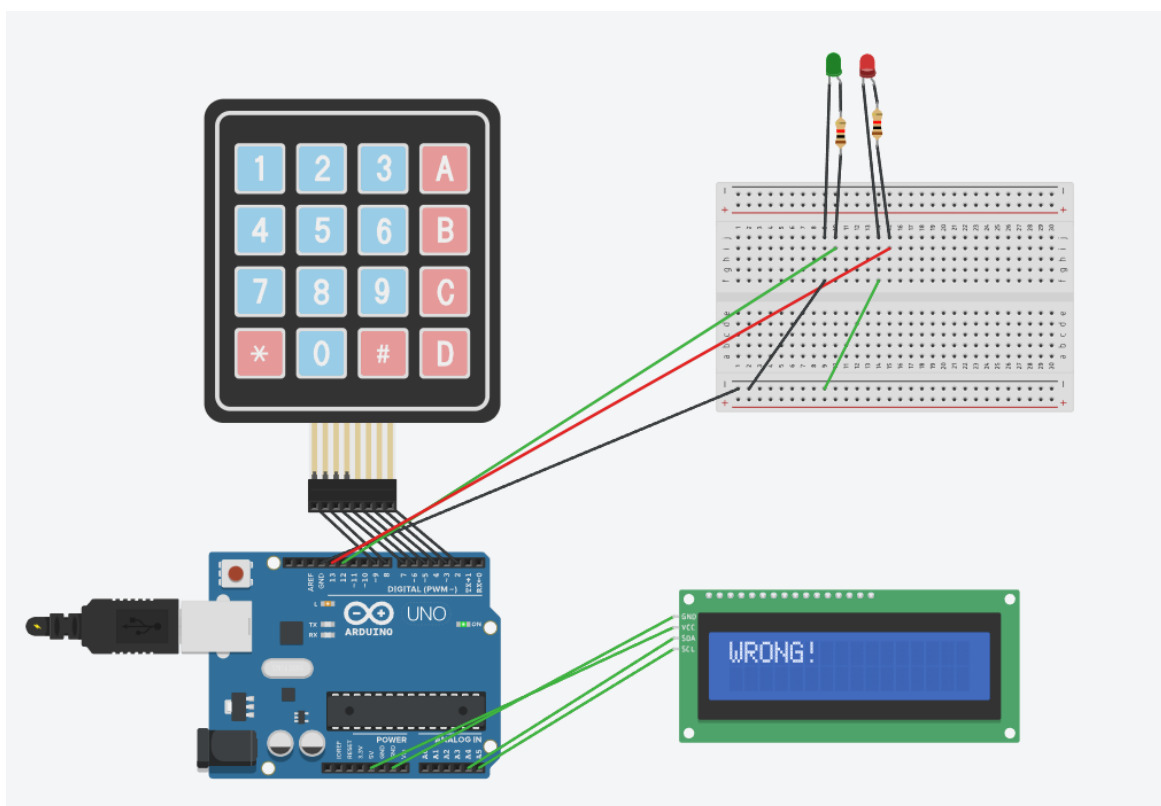


Figura 2 – Schema sarcinii 1.2

Lucrarea abordează trei probleme principale: configurarea aplicației pentru utilizarea librăriei STDIO prin interfața serială, implementarea logicii pentru detectarea unui cod de la o tastatură 4x4 și afișarea feedback-ului vizual și textual pe un LCD. Prima problemă constă în adaptarea codului pentru a utiliza STDIO în scanarea tastaturii și afișarea datelor pe LCD. A doua problemă implică citirea și validarea codului introdus de utilizator prin tastatură, ceea ce presupune cunoașterea tastaturilor matrice. A treia problemă se referă la afișarea rezultatului pe LCD și controlul LED-urilor în funcție de validitatea codului.

## Concluzie:

Am îndeplinit obiectivele stabilite, configurând și implementând cu succes aplicațiile pentru ambele interfețe. Am realizat atât proiectarea schemelor, cât și scrierea codului necesar pentru a asigura o interacțiune eficientă și intuitivă cu utilizatorul, utilizând terminalul serial, tastatura 4x4 și LCD-ul. Astfel, aplicația oferă un control clar și funcționalitate precisă în gestionarea comenzilor și feedback-ului vizual.

## Bibliografie

1. Keypad Library for Arduino [citat 12.09.2024] Disponibil:  
<https://playground.arduino.cc/Code/Keypad/>
2. Introducere în Arduino [citat 12.09.2024] Disponibil:  
<https://sites.google.com/site/arduinoelectronicasiprogramare/prima-lectie/5-introductere-in-arduino>
3. Arduino LCD I2C - Tutorial with Arduino Uno [citat 12.09.2024] Disponibil:  
[https://www.youtube.com/watch?v=EAeuxjtkumM&t=139s&ab\\_channel=AymaanRahman](https://www.youtube.com/watch?v=EAeuxjtkumM&t=139s&ab_channel=AymaanRahman)
4. How to Wire and Program a Button [citat 7.09.2024] Disponibil: <https://docs.arduino.cc/built-in-examples/digital/Button>
5. Introducere în Arduino [citat 12.09.2024] Disponibil:  
<https://sites.google.com/site/arduinoelectronicasiprogramare/prima-lectie/5-introductere-in-arduino>