# MINISTERUL EDUCAŢIEI, CULTURII ȘI CERCETĂRII

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică**

 **Departamentul Ingineria Software și Automatică**

Raport

**la disciplina Internetul lucrurilor IOT**

**Tema: Interacțiunea cu utilizatorul**

**A efectuat: st. gr.TI-204 Agatiev Dumitru**

**A verificat: asist. univ. D.Litra**

Chișinău 2023

# Definirea problemei

1. Sa se proiecteze o aplicate in baza de MCU care ar receptiona comenzi de la terminal prin interfata seriala pentru a seta starea unui LED;
   * led onpentru aprindere si led offpentru stingere., sistemul trebuie sa raspunda cu mesaje text despre confirmarea comenzii ;
   * pentru schimbul de text prin terminal a se utiliza libraia STDIO;
2. Sa se proiecteze o aplicatie in baza de MCU pentru detectarea unui cod de la o tastarura 4x4, sa verifice codul si sa afisese 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

* Implementarea a 2 aplicatii de interactiune cu utilizatorul in baza de microcontroller folosind limbajul C++ si biblioteca Arduino;
* Simularea produsului program obtinut folosind un software de simulare (ex. Proteus, Tinkercad) sau simularea cu ajutorul unui circuit real;
* Utilizarea bibliotecii STDIO pentru scanarea tastaturii si afisarea la LCD si pentru schimbul de text in terminal.

# Introducere

Internetul lucrurilor (IOT) reprezinta un concept ce are la baza conectarea unor dispositive, servicii si sisteme automate intr-o retea, folosind internetul. Un sistem bazat pe internetul lucrurilor are ca scop colectarea si transmiterea datelor in timp real. Este constituit din 3 componente:

* dispozitive smart – colecteaza date din mediul intern, interactiunea utilizatorului si trimite datele prin intermediul internetului;
* aplicatii IOT – set de servicii si produse program ce prelucreaza si transmite datele primite de la dispozitivele IOT;
* interfata utilizatorului – sunt folosite pentru manipularea dispozitivelor prin interventia utilizatorului.

Problema principala a lucrarii de laborator este interactiunea cu utilizatorul. Aceasta consta din elemente cu ajutorul carora utilizatorii pot comunica cu sistemele IOT, introducand diverse comenzi. Acestea sunt prelucrate ulterior de catre software-ul sistemului. Exista diverse posibilitati prin care utilizatorul poate interactiona cu o solutie IOT. Cele mai raspandite dintre ele sunt:

* receptionarea notificarilor in mod automat – in cazul unei operatiuni finalizate cu succes sau in urma aparitiei unei erori;
* monitorizarea informatiei in mod proactiv – se pot folosi dispozitive precum telefon mobil sau computer pentru monitorizarea informatiei;
* controlul sistemului la distanta – prin intermediul unei aplicatii mobile utilizatorul poate stinge lumina unei lampi sau stinge caldura in apartament;

# Materiale si metode

Pentru realizarea lucrarii de laborator au fost folosite urmatoarele materiale:

* LED – utilizat ca indicator pentru a indica utilizarea cu succes a echipamentelor electronice si codul sursa pentru realizarea sarcinilor;
* breadboard-ul Arduino – se foloseste pentru dezvoltarea circuitelor electronice si conexiunea acestor circuite cu microcontroller-ul;
* tastatura – tastatura 4x4 se conecteaza cu microcontroller-ul prin intermediul a 8 port-uri;
* LCD – display-ul 16x2 este format dintr-o matrice LCD, in care fiecare pixel este constituit din cateva molecule de cristale lichide si 2 filtre de polarizare. Afiseaza output-ul indicat in program;
* button – dispozitiv de input plasat pe breadboard. La apasarea acestuia se declanseaza aprinderea LED-ului ca output;

# Sarcina 1:

In urma efectuarii sarcinii a fost obtinut rezultatul asteptat. Astfel, in urma introducerii comenzii “led on” are loc aprinderea LED-ului, iar in urma introducerii comenzii “led off” are loc stingerea LED- ului. Prin apelul metodei Serial.begin(9600) are loc setarea portului serial la o rata de trasmitere a datelor de 9600 bps (bits per second).

Citireaa comenzilor de la terminal are loc prin apelul metodei Serial.readString(), iar afisarea raspunsului catre utilizator are loc prin apelul metodei Serial.println(“string”). In figura 1 se poate observa schema asamblata pe platforma tinkercad.

#include <stdio.h>

int ledPin = 2;

void setup() {

  pinMode(ledPin, OUTPUT);

  Serial.begin(9600);

}

void loop() {

  if (Serial.available() > 0) {

    String command = Serial.readStringUntil('\n');

    if (command == "led on") {

      digitalWrite(ledPin, HIGH);

      Serial.println("LED is on");

    } else if (command == "led off") {

      digitalWrite(ledPin, LOW);

      Serial.println("LED is off");

    }

  }

}

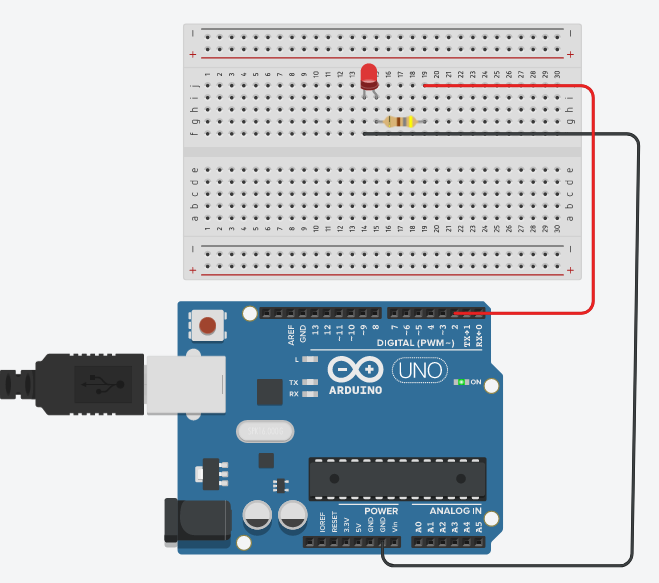


Figura 1. Schema ce prezinta sarcina 1

# Sarcina 2:

In urma analizei rezultatului obtinut s-a constat faptul ca in urma introducerii unui cod corect de la keypad pe LCD se afiseaza raspunsul correct, iar in urma intoroducerii unui cod de raspuns gresit, se afiseaza un mesaj de eroare. Pentru a obtine acest rezultat are loc initierea tastelor keypad-ului. Ca parametri constructorul Keypad se seteaza matricea tastelor prin apelul funtiei makeKeymap, numarul de coloane, numarul de linii si pinii pentru fiecare linie si pentru fiecare coloanal a matricei keypad-ului.

In metoda setup are loc initializarea LCD-ului si setarea iluminarii tastaturii prin metoda lcd.backlight(), intrucat LCD-ul nu produce lumina, ci o obtine prin intermediul unei surse.

In cadrul metodei enterpassword in cadrul fiecarei iteratii din ciclu se verifica daca utilizatorul nu a tastat simbolul “\*” pe keypad. Daca da, atunci are loc validarea codului.

In cazul unui cod valid, lcd-ul afiseaza un mesaj de success. In caz contrar, lcd-ul afiseaza un mesaj de eroare. In figura 2 se poate observa schema asamblata pe platforma tinkercad.

#include <Keypad.h>

#include <stdio.h>

#include <Adafruit\_LiquidCrystal.h>

#define MAX\_PASS\_LEN 10

char pass[MAX\_PASS\_LEN] = "1111";

char currPass[MAX\_PASS\_LEN];

int rLed = 12;

int gLed = 13;

unsigned long timing;

Adafruit\_LiquidCrystal lcd(0);

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {11, 10, 9, 8};

byte colPins[COLS] = {7, 6, 5, 4};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

int posMainPass = 0, posCurrPass = 0;

int keyOut = '\*';

static FILE my\_stream = {0};

int lcdPutChar(char ch, FILE\* f) {

if (ch != '\0') lcd.print(ch);

}

int keypadGetChar(FILE\* f) {

char key;

do {

key = keypad.getKey();

} while (key == 0);

return key;

}

void setup() {

Serial.begin(9600);

lcd.begin(16, 2);

pinMode(rLed, OUTPUT);

pinMode(gLed, OUTPUT);

fdev\_setup\_stream(&my\_stream, lcdPutChar,

keypadGetChar, \_FDEV\_SETUP\_RW);

stdin = &my\_stream;

stdout = &my\_stream;

}

void loop() {

lcd.clear();

printf("Enter Password");

enterPassword(currPass);

if (compare()){

blink(gLed, "Correct Password");

}

else{

blink(rLed, "Wrong Password");

}

}

void enterPassword(char pass[]) {

int pos = 0;

while (true) {

char key;

scanf("%c", &key);

if (key == keyOut) {

for (int i = 0; i < realLength(); i++)

printf("%s", pass[i]);

break;

}

pass[pos] = key;

lcd.setCursor(pos, 1);

printf("\*");

pos++;

}

}

bool compare(void) {

int len = realLength();

for (int i = 0; i < len; i++) {

if (currPass[i] != pass[i])

return false;

}

return true;

}

void blink(int led, char msg[]) {

lcd.clear();

lcd.setCursor(0, 0);

printf(msg);

if (led == 12) {

lcd.setCursor(0, 1);

printf("again");

digitalWrite(led, HIGH);

delay(500);

digitalWrite(led, LOW);

loop(); }

else {

while (true){

digitalWrite(led, HIGH);

delay(500);

loop();

}

}

}

int realLength(void) {

int count = 0;

while (pass[count++]) {}

return count;

}

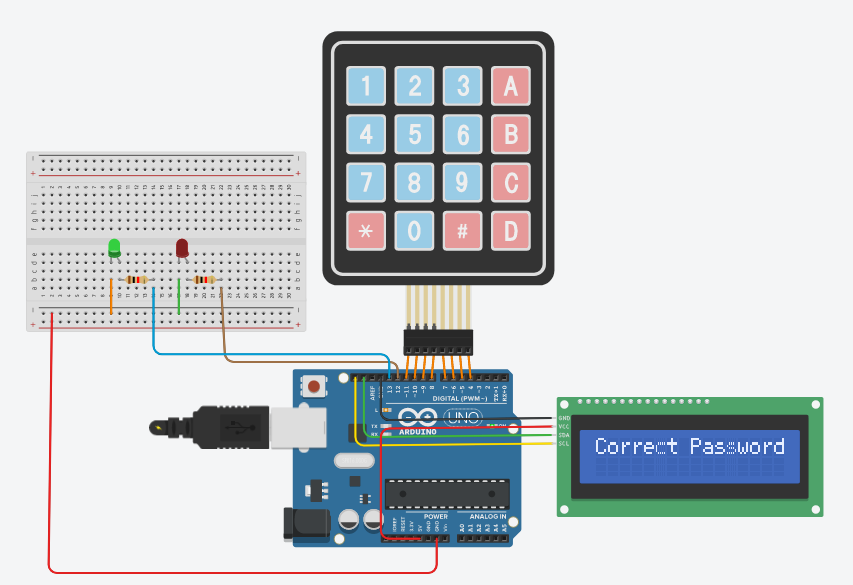


Figura 2. Schema ce prezinta sarcina 2

# Concluzie

După desfășurarea laboratorului, am reușit să dezvoltăm cu succes două aplicații practice bazate pe un microcontroller, folosind biblioteca Arduino. Aceste aplicații au implicat integrarea și utilizarea unei game variate de dispozitive periferice, precum butoane, tastatură (keypad), LED-uri și ecrane LCD. De asemenea, am lucrat cu placa Arduino și cu breadboard-ul pentru a crea simulări de circuite.

Scopul principal al acestor simulări de circuite a fost de a crea un mediu în care să putem detecta semnalele de intrare de la dispozitivele periferice și să obținem rezultate corespunzătoare ca ieșire. Acest lucru ne-a permis să experimentăm cu diferite scenarii și să înțelegem modul în care microcontrollerul poate interacționa cu mediul înconjurător.

Pentru a facilita comunicarea între microcontroller și dispozitivele periferice, am utilizat obiectul Serial din biblioteca Arduino. Aceasta a fost o componentă crucială pentru citirea și scrierea datelor în terminal, permițându-ne să monitorizăm și să controlăm fluxul de informații între componentele noastre.

În concluzie, acest laborator a fost o experiență educatională valoroasă în care am dobândit competențe practice în dezvoltarea și utilizarea microcontrollerelor, precum și în crearea de circuite simulate pentru a rezolva diverse probleme. Această abordare practică ne-a ajutat să înțelegem mai bine cum pot fi folosite microcontrollerele și cum pot fi integrate în aplicații reale.