**TECHNICAL UNIVERSITY OF MOLDOVA**
**FACULTY OF COMPUTERS, INFORMATICS**
**AND MICROELECTRONICS**
**DEPARTMENT OF SOFTWARE ENGINEERING**
**AND AUTOMATICS**

# Laboratory work nr. 1.2

## User interaction - LCD+Keypad STDIO

Created by: **Frunze Vladislav**
st. gr. FAF-212

Verified by: **Moraru Dumitru**
university lecturer

**Chișinău, 2024**

## THE TASK

Configure the application to work with the STDIO library through the serial interface for text exchange via LCD+Keypad. To design an MCU-based application for detecting a code from a 4x4 keyboard, verifying the code and displaying a message on an LCD.

- for a valid code, a green LED should light up, for an invalid code, a red LED.
- use STDIO to scan the keyboard and display on the LCD.

Recommendation:

- it is recommended to use an IDE with arduino support, which allows working with several files. For example, Eclipse;
- for Validation it is recommended to use a simulator, e.g. Proteus;
- the functionalities for each peripheral equipment (led, button, lcd, keypad) should be created in separate files, for the purpose of reuse in the other laboratories;
- use CamelCase coding rules;

## THE PROGRESS OF THE WORK

### The flowchart of the program

In the below diagram, it is represented the flow of the main program. It has three steps. First, it describes the definitions of the main constants. Next, it sets up the communication between the keypad and the LCD. Furthermore, it enters a loop that waits for keypad input and processes it accordingly.

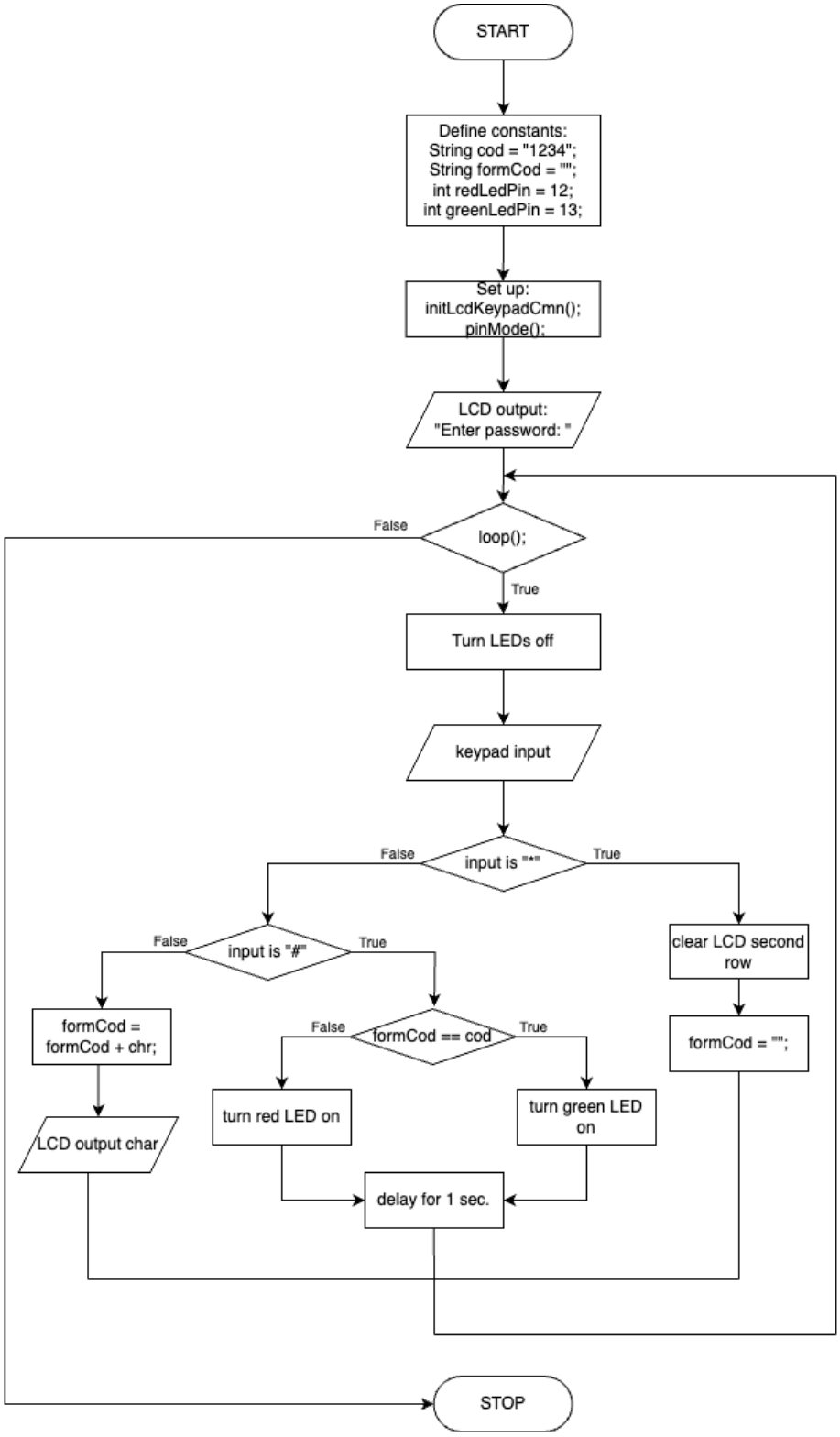


**Figure 1** The flowchart of the program

**The description of the main functions used to perform the tasks**

setup() - Initializes the system, including LCD, keypad, LED pins, and prints a prompt to enter a password to the first row of the LCD.

loop() - Continuously checks for input and handles the entered characters accordingly.

turnLedOff() - Turns off the specified LED pin.

turnLedOn() - Turns on the specified LED pin.

clearLcdSecondRow() - Clears the second row of the LCD display.

initLcdKeypadCmn() - Initializes the LCD, and sets the standart input and output to the custom functions.

printf() - Prints formatted data to the standard output which, in this case, is the LCD.

pinMode() - Configures the specified pin to behave as either an input or an output.

scanf() – Reads from the standard input, which in this case is the keypad.

delay() - Pauses the program to show the LED light for a specified number of milliseconds.

**The block diagram of program/s**

The block diagram describes a software that works by reading input from the keypad until the sequence of characters inputted form a pre-defined cod. When the user presses "#", the program checks if the formed cod matches the pre-defined one and if it does it lights up the green led, otherwise the red led.

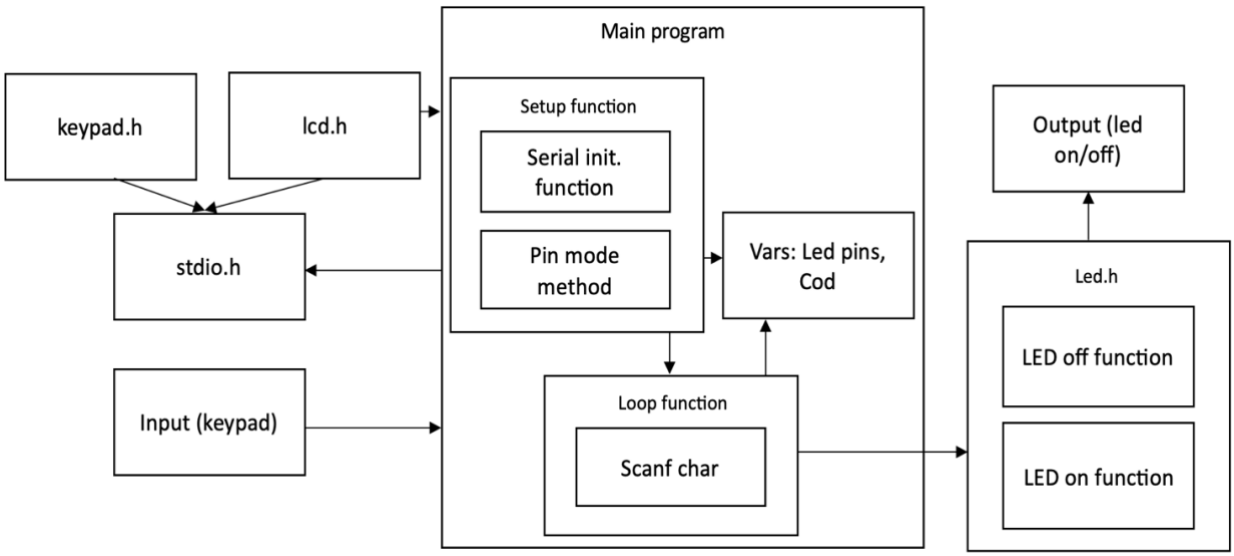


**Figure 2** The block diagram

**The simulated electrical schematic**

The schematic shows the connection of a 4x4 matrix keypad, an LCD, and two LEDs to an Arduino Uno microcontroller. The keypad has 16 keys arranged in a 4x4 grid, which is going to be used as the standard input. The two LEDs are going to be used as an indicator of valid (green) and invalid (red) codes. The LED is going to display a label and the inputted chars.

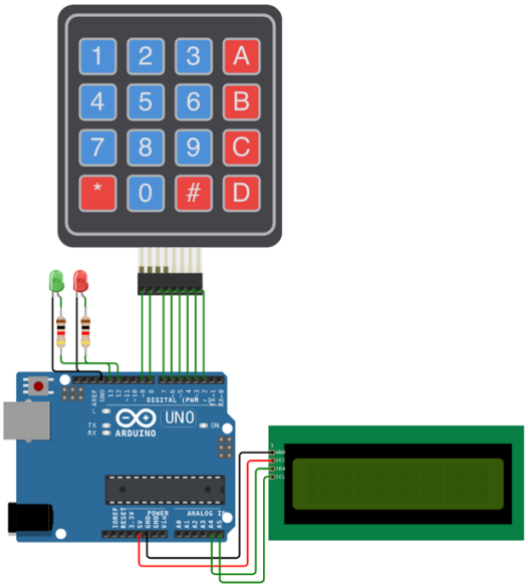


**Figure 3** Electrical schematic in Wokwi

**The screenshot of running the simulation program**

The following figure shows a running simulation. The LCD shows “Enter password:” label, and below the “1234” inputted code. Then it is clicked “#”. The inputted code is compared with the pre-defined code in the program, and it matches it as the green LED is turned on.

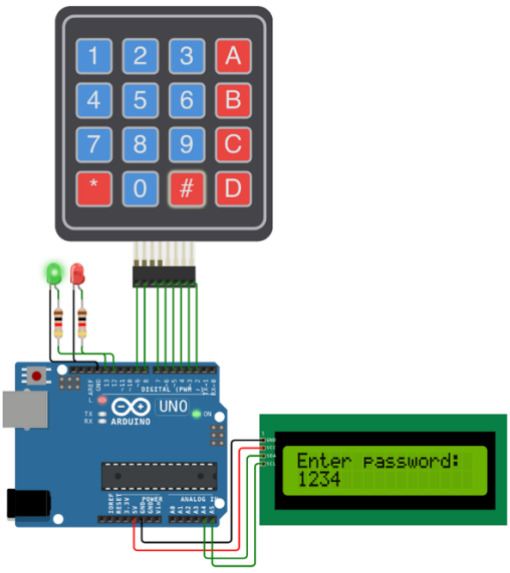


**Figure 4** Running the simulation program

## CONCLUSIONS

Arduino is a quite helpful tool when it is necessary to create a system where user interaction is crucial. It helps you unite different components together for a practical and convenient user interaction.

In this particular laboratory work, it was created a simple lock system, which can be used for doors to secure the entrance in a building or for simple safes.

In addition, Arduino hardware and its API show a high level of flexibility as you can connect an immense number of components in a complex system.

## BIBLIOGRAPHY

**1** ARDUINO: *Getting Started with Arduino*. Arduino official site, ©2024 [quote 14.02.2024]. Access link: https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/

**2** WOKWI: *Arduino Uno*. Wokwi simulator site, ©2024 [quote 14.02.2024]. Access link: https://wokwi.com/projects/new/arduino-uno

**3** ARDUINO: *Liquid Crystal Displays (LCD) with Arduino*. Arduino official site, ©2024 [quote 13.02.2024]. Access link: https://docs.arduino.cc/learn/electronics/lcd-displays/

**4** WOKWI: *Membrane Keypad Reference*. Wokwi official site, ©2024 [quote 13.02.2024]. Access link: https://docs.wokwi.com/parts/wokwi-membrane-keypad

## APPENDIX 1: Source code of the program

*sketch.ino*

```
#include <string.h>
#include "stdio.h"
#include "led.h"

char ext = '*';
char enter = '#';
String cod = "1234";
String formCod = "";
int redLedPin = 12;
int greenLedPin = 13;
int showLight = 2000;

void setup() {
  initLcdKeypadCmn();
  printf("Enter password: ");
  pinMode(greenLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
};

void loop() {
  turnLedOff(greenLedPin);
  turnLedOff(redLedPin);

  char chr;
  scanf("%c", &chr);

  if (chr == ext) {
    clearLcdSecondRow();
    formCod = "";
  } else if (chr == enter) {
    if (formCod == cod) {
      turnLedOn(greenLedPin);
    } else {
      turnLedOn(redLedPin);
    }
    delay(showLight);
  } else {
    formCod = formCod + chr;
    printf("%c", chr);
  }
}
```

*stdio.h*

```
#include <stdio.h>
#include "keypad.h"
#include "lcd.h"


void lcdPutChar(char c, FILE *f){
  if (col == lcdColumns) {
    if (row == usedRow) {
      clearLcdSecondRow();
    } else {
      col = firstCol; row++;
    }
  }
  lcd.setCursor(col, row);
  lcd.print(c);
  col++;
}

void keypadGetChar(FILE *f){
  char key;
  do {key = keypad.getKey();}
  while (!key);
```

```
  return key;
}

void initLcdKeypadCmn(){
  lcd.init();
  lcd.backlight();
  FILE *myStream = fdevopen(lcdPutChar, keypadGetChar);
  stdin = stdout = myStream;
}
```

### *keypad.h*

```
#include <Keypad.h>

const uint8_t ROWS = 4;
const uint8_t COLS = 4;
char keys[ROWS][COLS] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};
uint8_t colPins[COLS] = { 5, 4, 3, 2 }; // Pins connected to C1, C2, C3, C4
uint8_t rowPins[ROWS] = { 9, 8, 7, 6 }; // Pins connected to R1, R2, R3, R4
Keypad keypad = Keypad(
  makeKeymap(keys),
  rowPins, colPins,
  ROWS, COLS
);
```

### *lcd.h*

```
#include <LiquidCrystal_I2C.h>

#define i2cAddr    0x27
#define lcdColumns 16
#define lcdLines   2

int col = 0;
int row = 0;
int firstCol = 0;
int usedRow = 1;

LiquidCrystal_I2C lcd(i2cAddr, lcdColumns, lcdLines);

void clearLcdSecondRow() {
  lcd.setCursor(firstCol, usedRow);
  for (int i = 0; i < lcdColumns; i++) {
    lcd.print(" ");
  }
  col = firstCol;
}
```

### *led.h*

```
#include <Arduino.h>
#include "led.h"


bool turnLedOn(int ledPin) {
    digitalWrite(ledPin, HIGH);
}

bool turnLedOff(int ledPin) {
    digitalWrite(ledPin, LOW);
}
```