

SafeMe

Nume: Cocoara Oana

Grupa: 336CB

Introducere

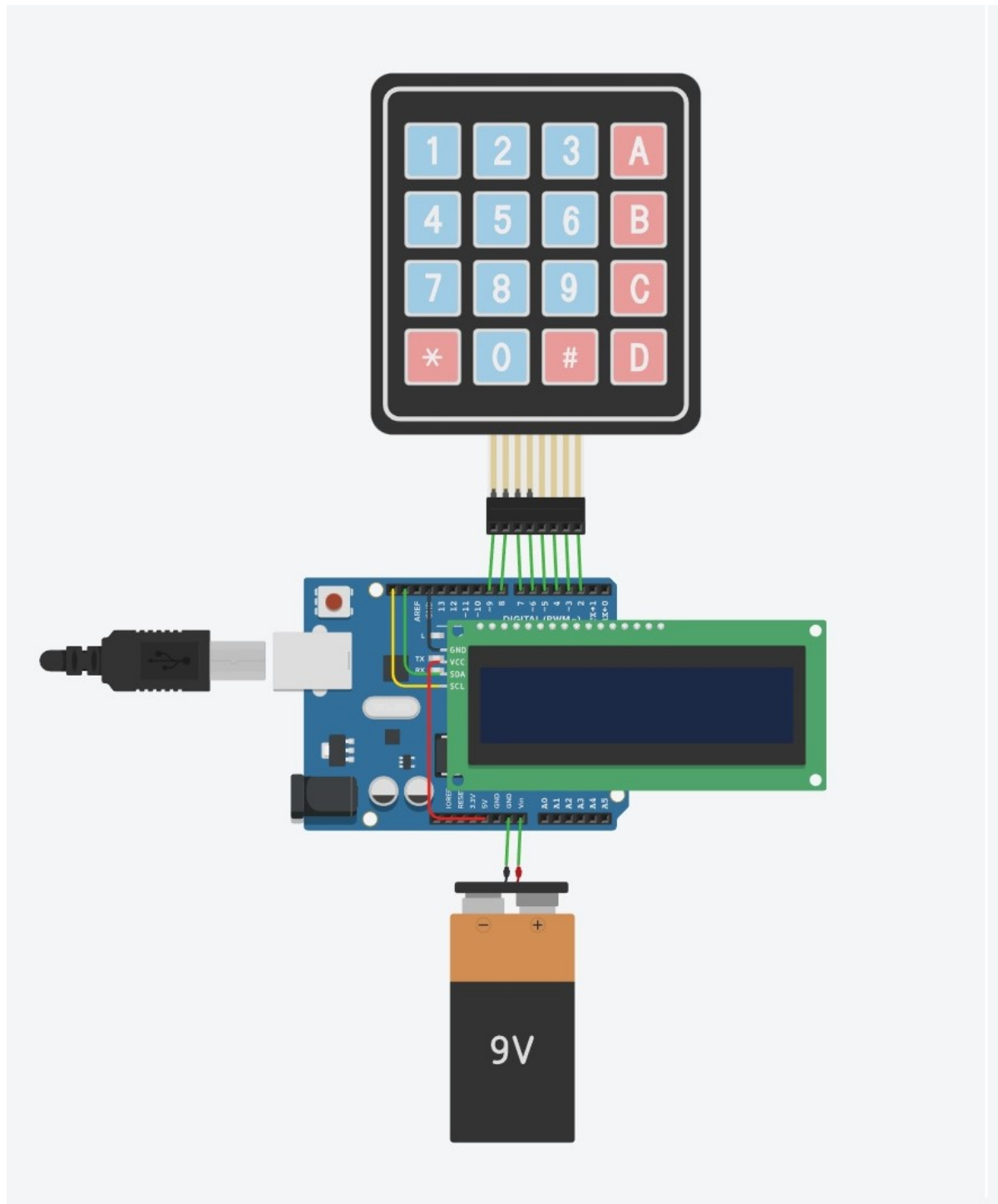


SafeMe este un proiect care imbina partea de embedded cu cea de criptografie. Acest sistem va permite utilizatorului sa creeze sau sa decripteze datele sale.

Sistemul propus este atractiv si util deoarece este compact si portabil, dimensiunile reduse vor permite integrarea acestuia in alte sisteme intr-un mod mult mai usor, iar faptul ca este portabil il face usor de transportat in diferite locatii, deci mai usor de utilizat.

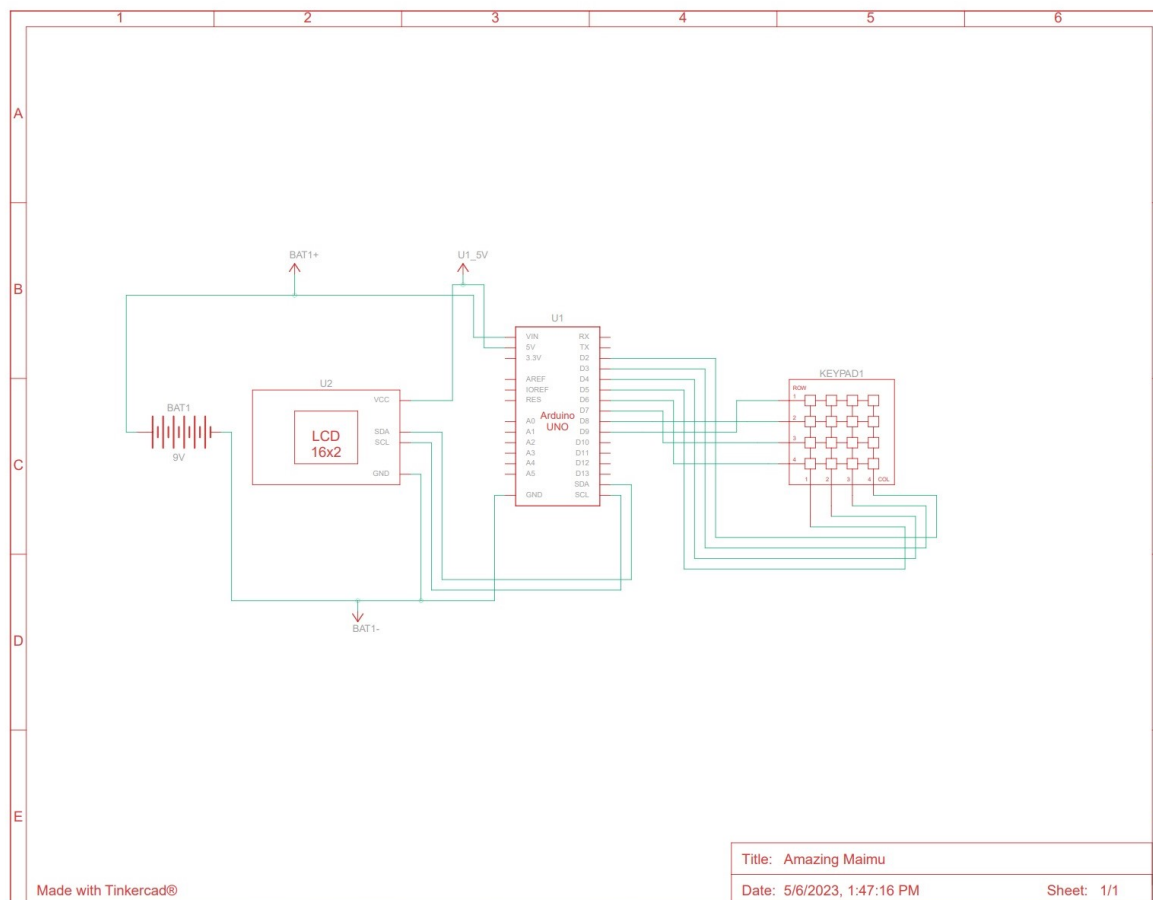
Descriere generală

Un sistem care cripteaza/decripteaza mesajele care vin pe interfata seriala. Algoritmul de criptare este un One Time Pad unde expansiunea cheii se face prin repetarea ei. Accesul la chei se face prin intermediul unor utilizatori presetati, ale caror drepturi sunt accesate folosind PIN-ul propriu de 4 cifre.



Hardware Design

Schema hardware a proiectului (mai trebuie adăugat shieldul de host USB).



Piese ce vor fi folosite în realizarea proiectului sunt:

- Placa Arduino
- Fire
- Modul Afisare
- Shield USB host
- Breadboard
- Alimentare cu baterie

Software Design

Mod de functionare:

- Meniu initial de login: tot ce se poate face e sa se aleaga un user pentru logare: se cere PIN-ul de 4 cifre, care se introduce de la tastatura. La login efectuat cu succes, globala active_user se schimba

la indexul userului respectiv. Va fi folosita la alegerea cheii de criptare si la verificarea posibilitatii decriptarii unui cyphertext.

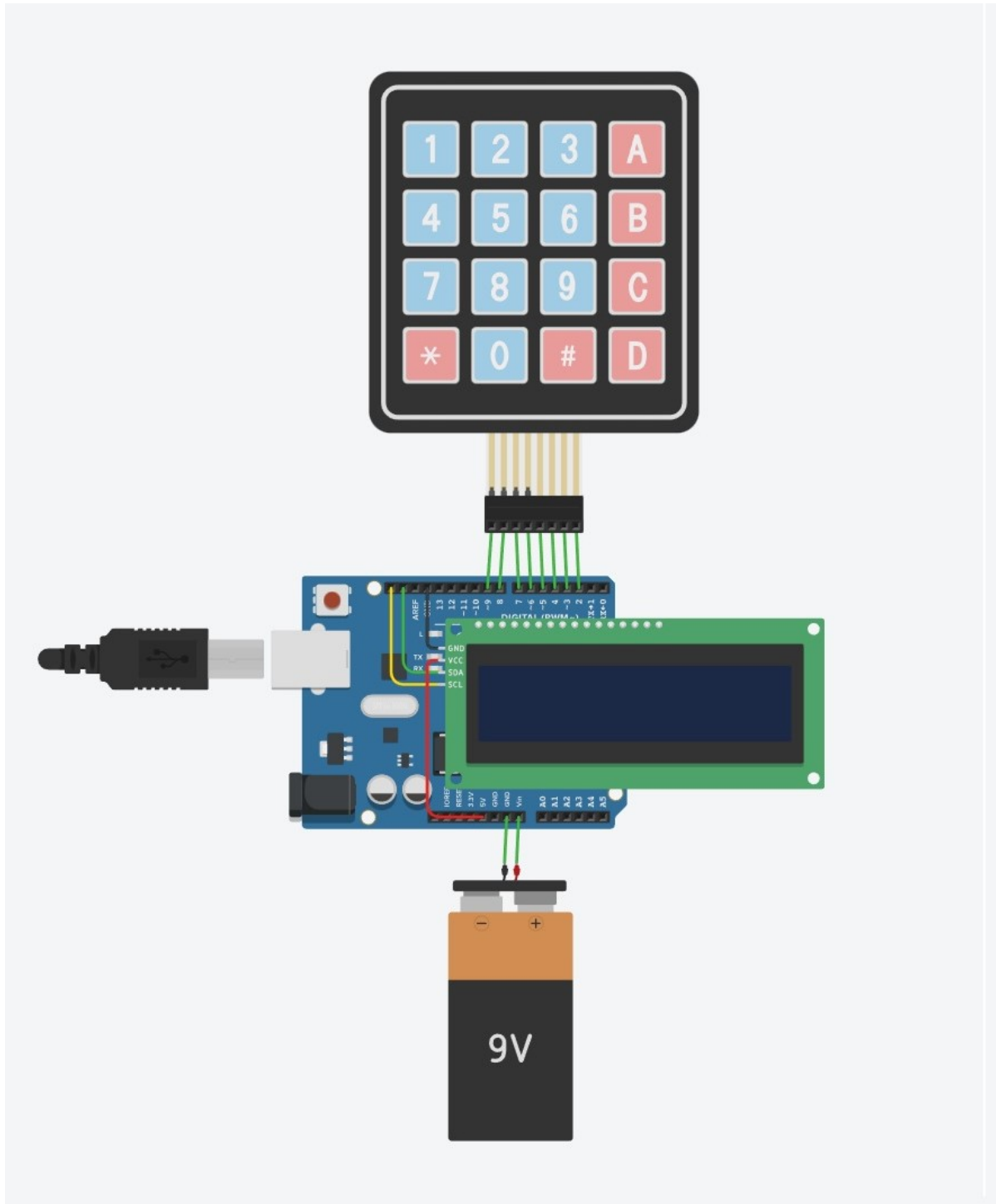
Meniul de utilizare efectiva de catre un user: optiuni pentru criptare, decriptare si logout.

- Criptare: asteapta input de la interfata seriala. Acesta se cripteaza (aproape) OTP (expansiunea cheii este doar repetarea ei) folosind cheia cu indexul active_user si rezultatul este intors la seriala (primul caracter identifica cheia folosita).
- Decriptare: se verifica daca userul curent poate decripta cyphertextul dat (in functie de primul caracter, despre care am mai zis): userii normali pot folosi doar propria cheie, admin le poate folosi pe toate.
- Logout: se seteaza active_user la -1 si se revine in meniul initial.

Rezultate Obținute

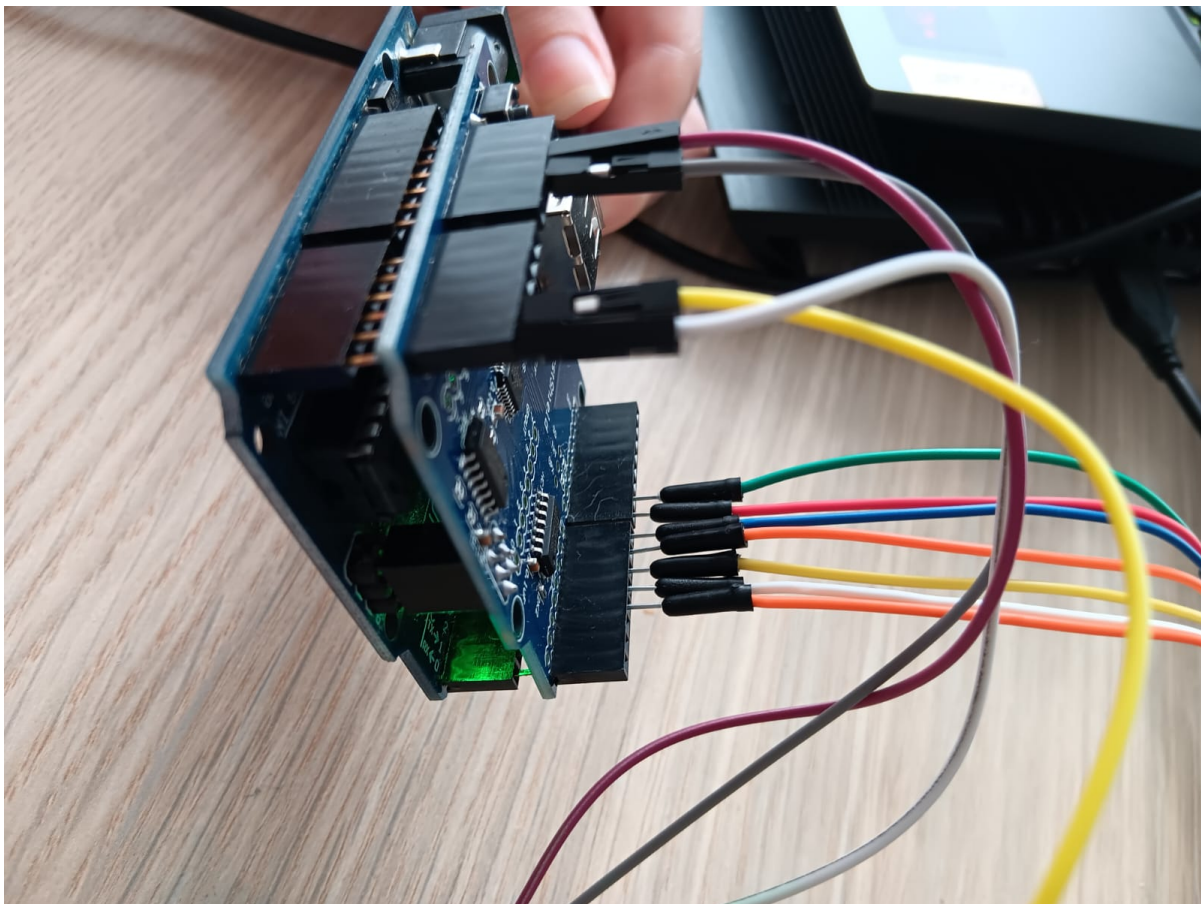
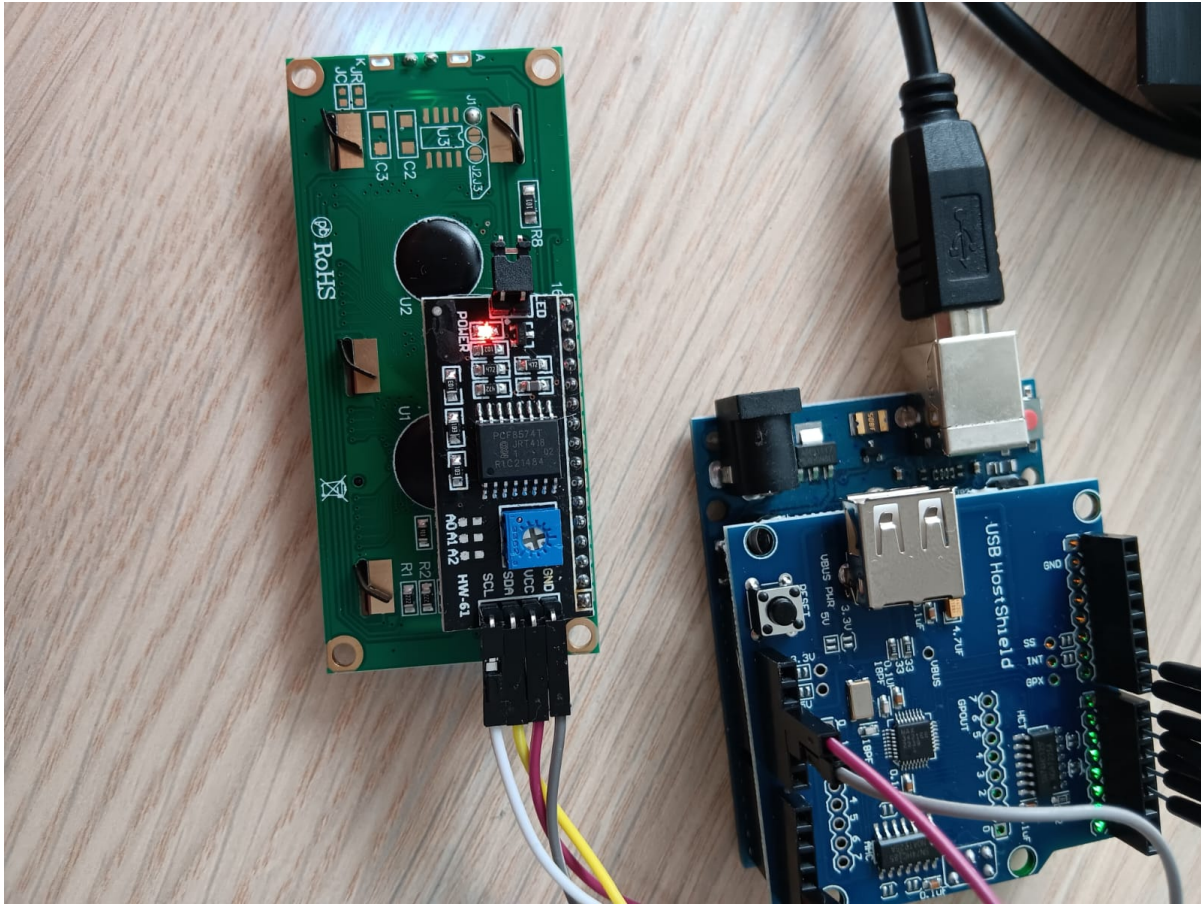
Care au fost rezultatele obținute în urma realizării proiectului vostru.

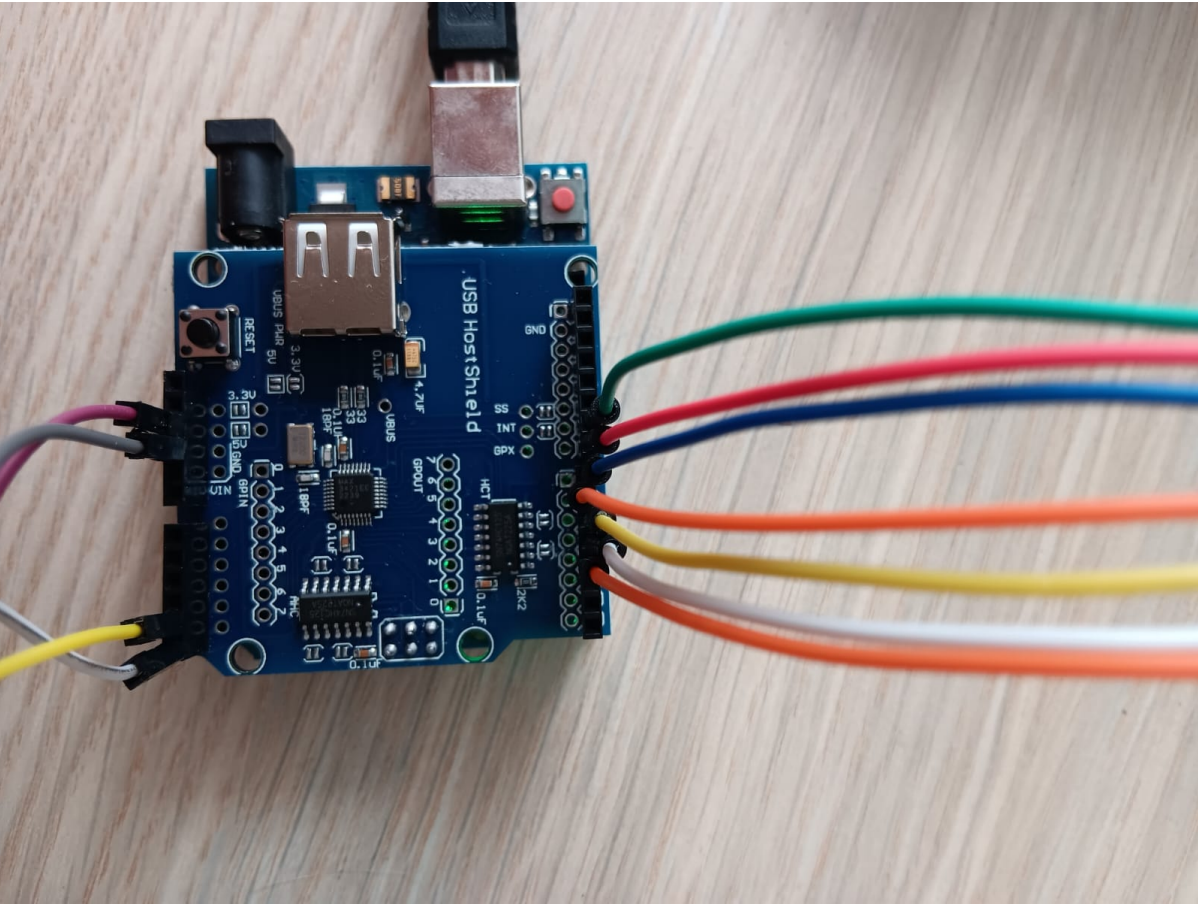
- Schema proiectului (lăste shieldul in aceasta schema):

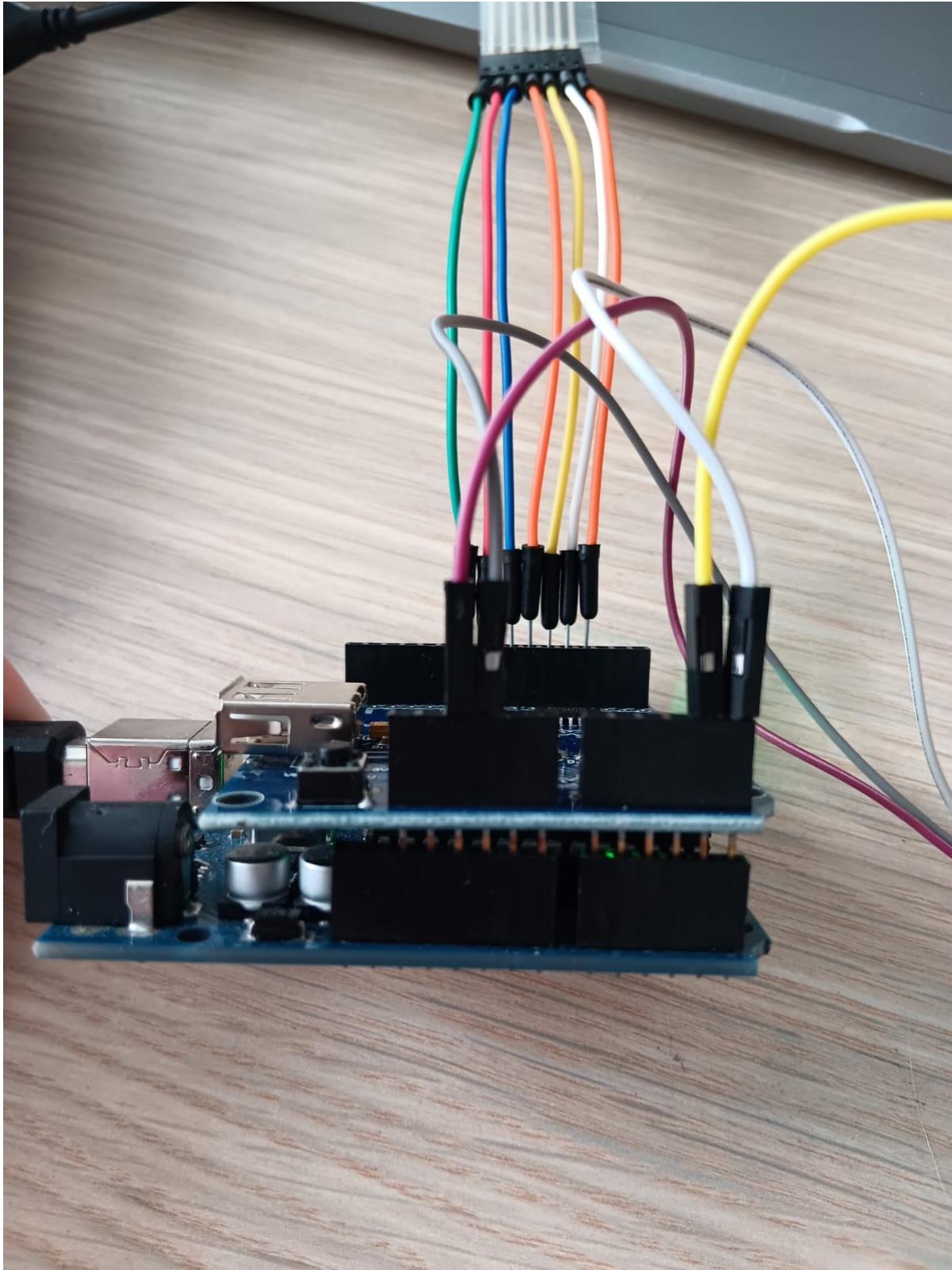


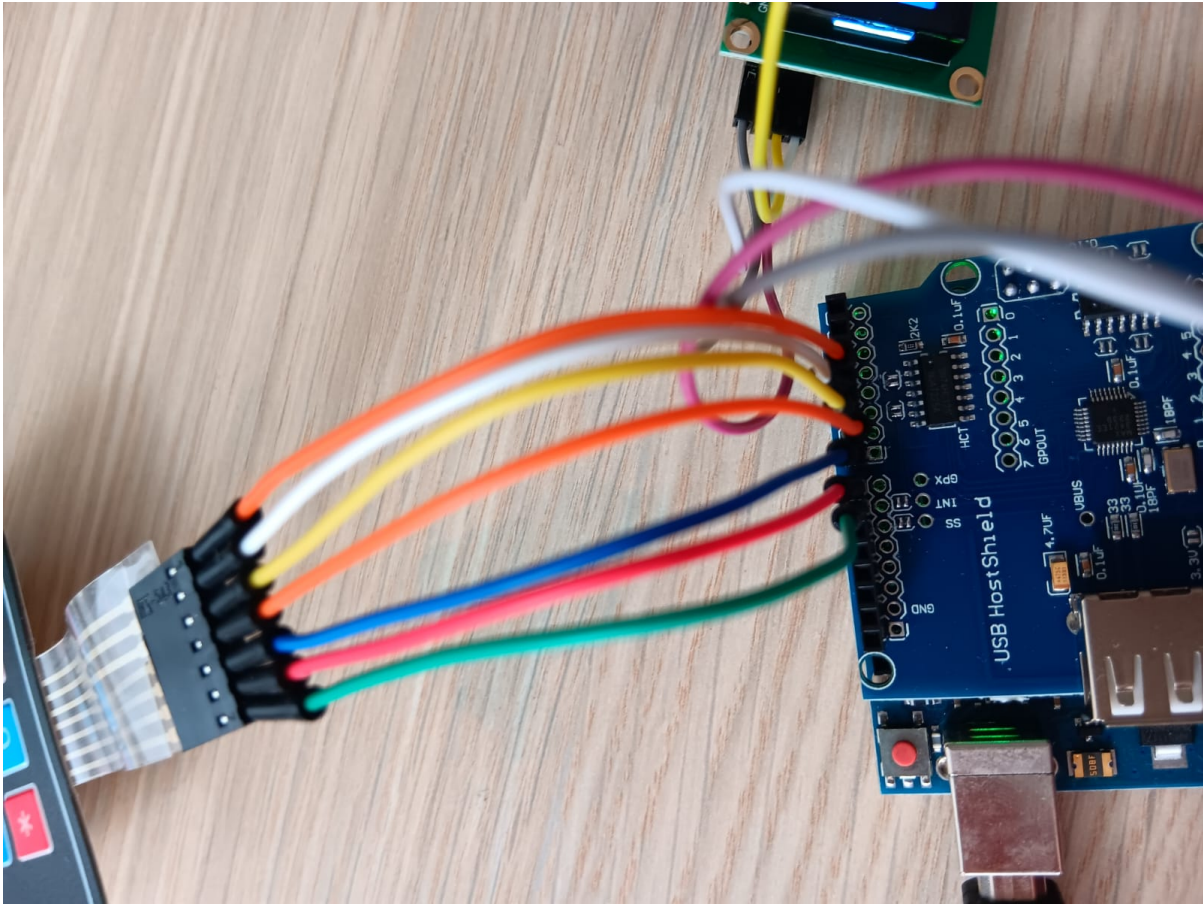
- Implementare hardware a proiectului si rezultatul testarii componentelor:



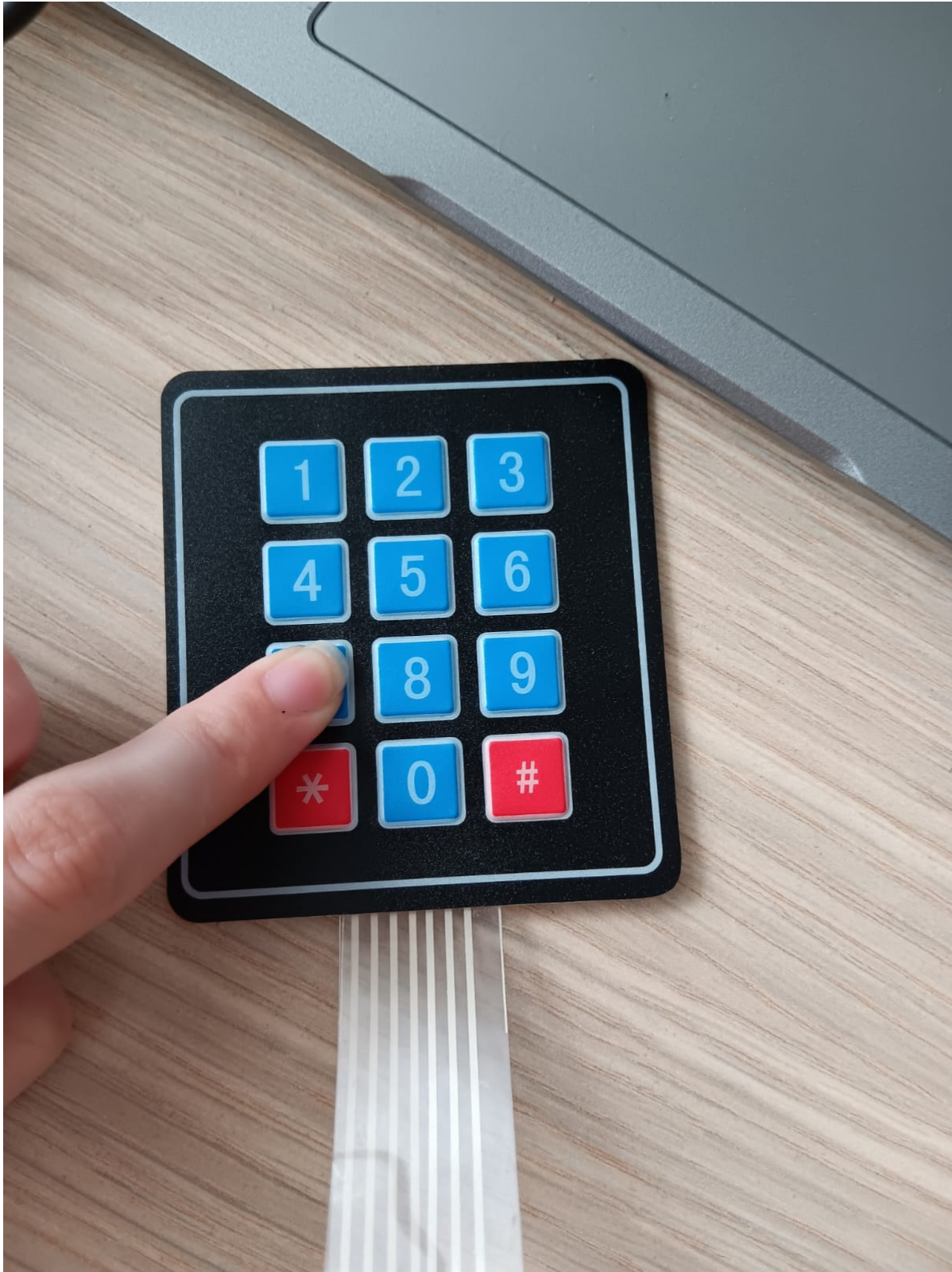






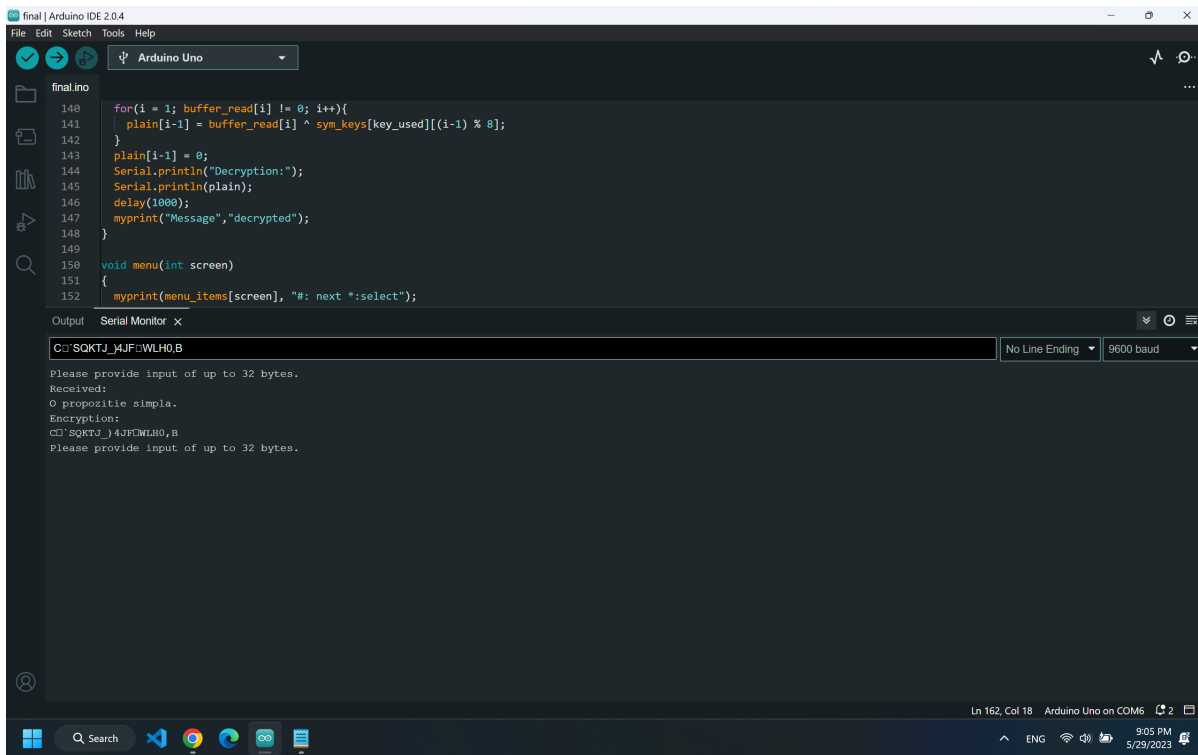








- Exemplu de functionalitate: logare cu unul dintre utilizatori, dupa care am ales optiunea de criptare, am adaugat textul pe care doream sa il criptez. La final am ales sa sa decriptez textul criptat - am ales optiunea de decriptare din meniul utilizatorului.

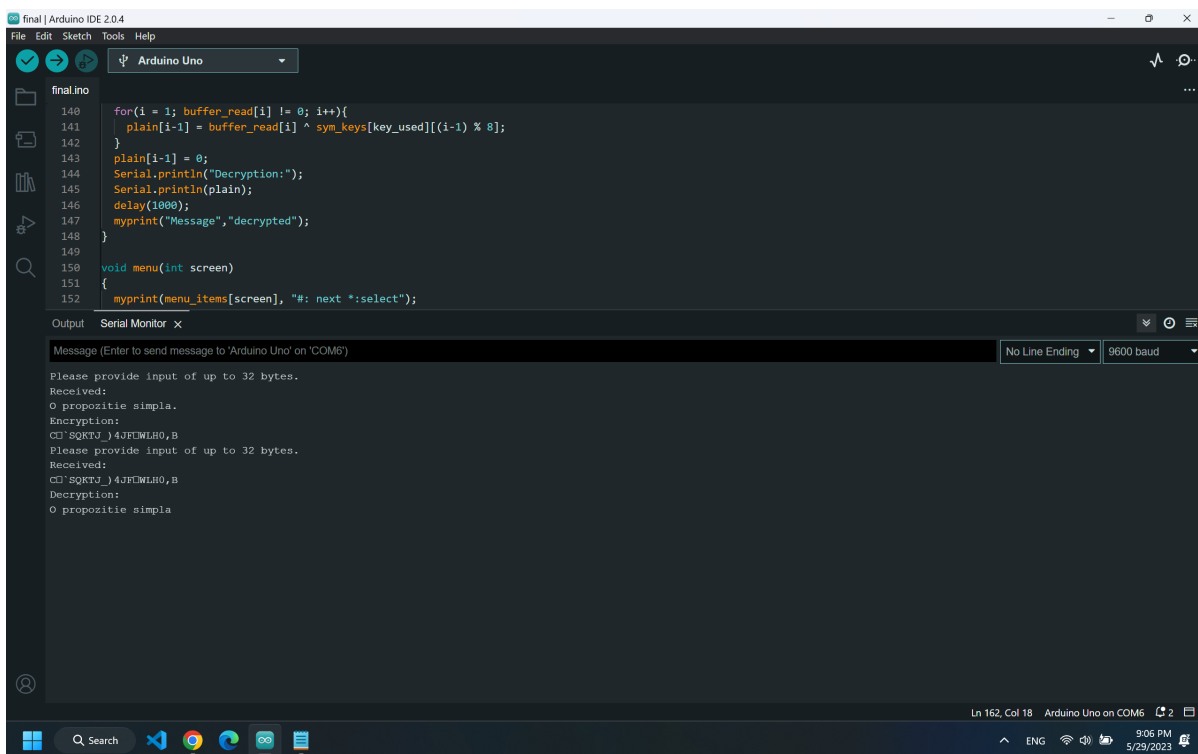


The screenshot shows the Arduino IDE 2.0.4 interface. The code in `final.ino` is as follows:

```
140 for(i = 1; buffer_read[i] != 0; i++){
141   plain[i-1] = buffer_read[i] ^ sym_keys[key_used][(i-1) % 8];
142 }
143 plain[i-1] = 0;
144 Serial.println("Decryption:");
145 Serial.println(plain);
146 delay(1000);
147 myprint("Message","decrypted");
148 }
149
150 void menu(int screen)
151 {
152   myprint(menu_items[screen], "#: next *:select");
```

The Serial Monitor shows the following output:

```
C:\SOKTJ_MJF:WLH0,B
Please provide input of up to 32 bytes.
Received:
O propositie simpla.
Encryption:
C:\SOKTJ_MJF:WLH0,B
Please provide input of up to 32 bytes.
```



The screenshot shows the Arduino IDE 2.0.4 interface with the same code as the first screenshot. The Serial Monitor shows the following output:

```
Message (Enter to send message to 'Arduino Uno' on 'COM6')
Please provide input of up to 32 bytes.
Received:
O propositie simpla.
Encryption:
C:\SOKTJ_MJF:WLH0,B
Please provide input of up to 32 bytes.
Received:
C:\SOKTJ_MJF:WLH0,B
Decryption:
O propositie simpla
```

Concluzii

- Nu a mers shieldul pt ca cel mai probabil nu e suportat de biblioteci(eu am luat o placa generica si posibil sa existe diferente). Am inlocuit cu scriere si citire de la interfata seriala. Criptarea se face folosind One Time Pad cu chei de 8 octeti.

- Primul caracter din cyphertext ajuta la identificarea cheii folosite. Indexarea cheilor incepe de la 0, ceea ce ar duce la a printa nimic criptat cu cheia 0. Solutie: adaug 67 la indexul cheii folosite.
- Problema (oarecum in acelasi registru cu cea anterioara): Folosind chei cu caractere uzuale, se poate ivi cazul in care am $X \text{ xor } X$, ceea ce da 0, iar printarea va vedea asta ca un end of string. Asadar, am folosit caractere mai putin folosite in chei.
- Alte probleme simple: read nu este o functie blocanta si a trebuit sa fac un loop blocant cu `Serial.available()`.

Download

[safe_me.zip](#)

Jurnal

Secțiune de jurnal în care se poate urmări progresul proiectului.

- Descrierea proiectului 2023/05/07
- Conectarea și testarea componentelor 2023/05/16
- Nu a mers USB HostShield-ul și readaptarea proiectului 2023/05/27
- Finalizarea părții software a aplicației 2023/05/28
- Realizarea documentației finale și încărcarea lor pe site 2023/05/29
- Prezentarea proiectului 2023/05/30

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

- Aici se poate găsi codul folosit pentru testarea componentelor - https://github.com/alexandra70/PM_SafeMe/tree/main/task0
- În realizarea codului de mai sus am folosit informații care apar în următoarele surse:
 - <https://github.com/fmalpartida/New-LiquidCrystal/tree/master/thirdparty%20libraries/SoftI2CMaster>
 - <https://forum.arduino.cc/t/ide-2-0-3-compilation-error-no-connection-established/1066586/2>
 - https://github.com/johnrickman/LiquidCrystal_I2C/tree/master/examples
 - <https://www.circuitgeeks.com/arduino-i2c-lcd-tutorial/>
 - <https://arduino.stackexchange.com/questions/56517/formatting-strings-in-arduino-for-output>

- <https://lastminuteengineers.com/arduino-keypad-tutorial/>
- <https://robojax.com/learn/arduino?vid=robojax-keypad-4x3>
- Codul aplicatiei - https://github.com/alexandra70/PM_SafeMe/tree/main/final
- In realizarea codului de mai sus am folosit informatii care apar in urmatoarele surse:
 - <https://forum.arduino.cc/t/read-char-over-serial-solved/153697/6>
 - <https://forum.arduino.cc/t/reading-from-serial-to-array-char-string/42712>
 - <https://www.engineersgarage.com/moving-text-on-16x2-lcd-with-arduino/>
 - <https://circuits4you.com/2018/03/08/arduino-convert-string-to-character-array/>
 - <https://www.delftstack.com/howto/arduino/arduino-wait-for-input/>
- In incercarea de a face sa functioneze USB HostShied am folosit urmatoarele surse:
 - <https://forum.arduino.cc/t/write-and-read-memory-stick-with-arduino-host-shield/398451>
 - <https://github.com/greiman/UsbFat>
 - <https://github.com/greiman/UsbFat/issues/2>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/iotelea/safeme>



Last update: **2023/05/29 19:24**