

Password Storage Unit

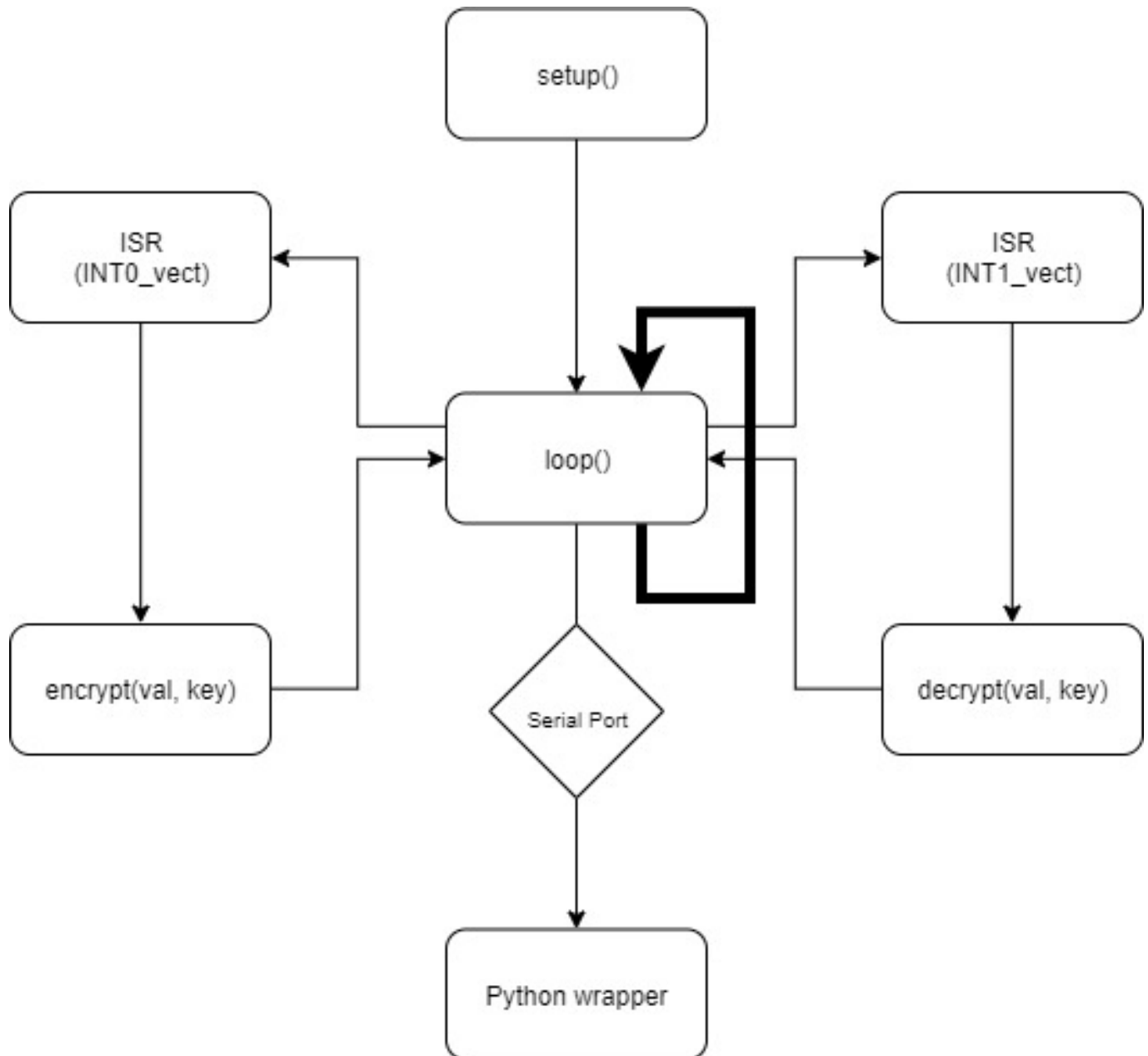
Introducere

Prezentarea pe scurt a proiectului:

- un dispozitiv portabil care face auto-complete la parole pentru conturile alese
- scopul proiectului este să ușureze autentificarea pe un site și să nu mai forțeze persoana să memoreze parole
- ideea de la care am plecat este funcționalitatea asemănătoare pe care o oferă Google
- consider că este util deoarece, personal, mi se întâmplă des să îmi uit parola de la un cont

Descriere generală

Schemă Bloc



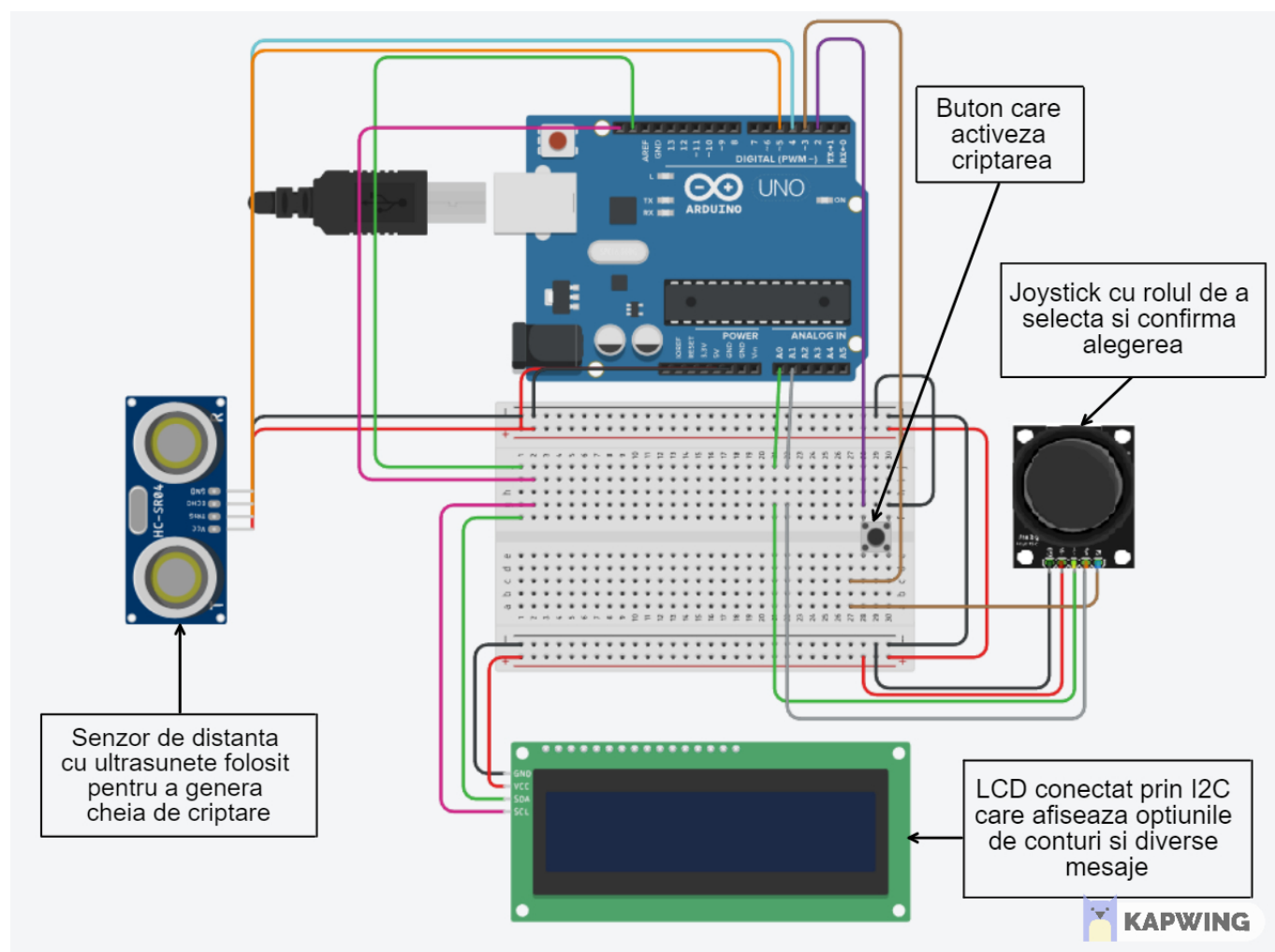
Hardware Design

Listă de piese

- 1 x plăcuță de dezvoltare Arduino Uno
- 1 x breadboard
- 1 x ecran LCD
- 1 x Joystick
- 1 x buton
- 1 x senzor distanță ultrasunete
- fire de conectare

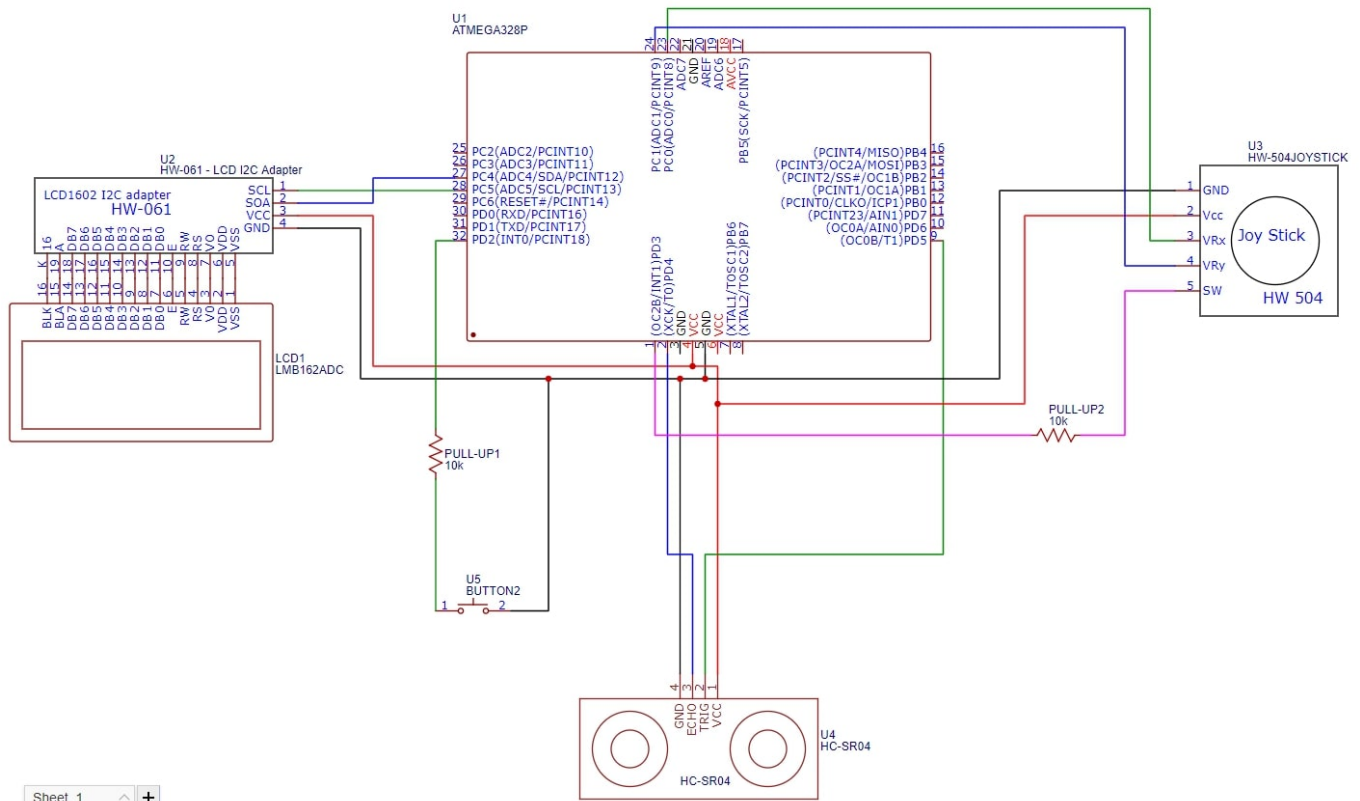
Schemă hardware și descrierea componentelor

Schemă hardware realizată cu Tinkercad si editată cu Kapwing

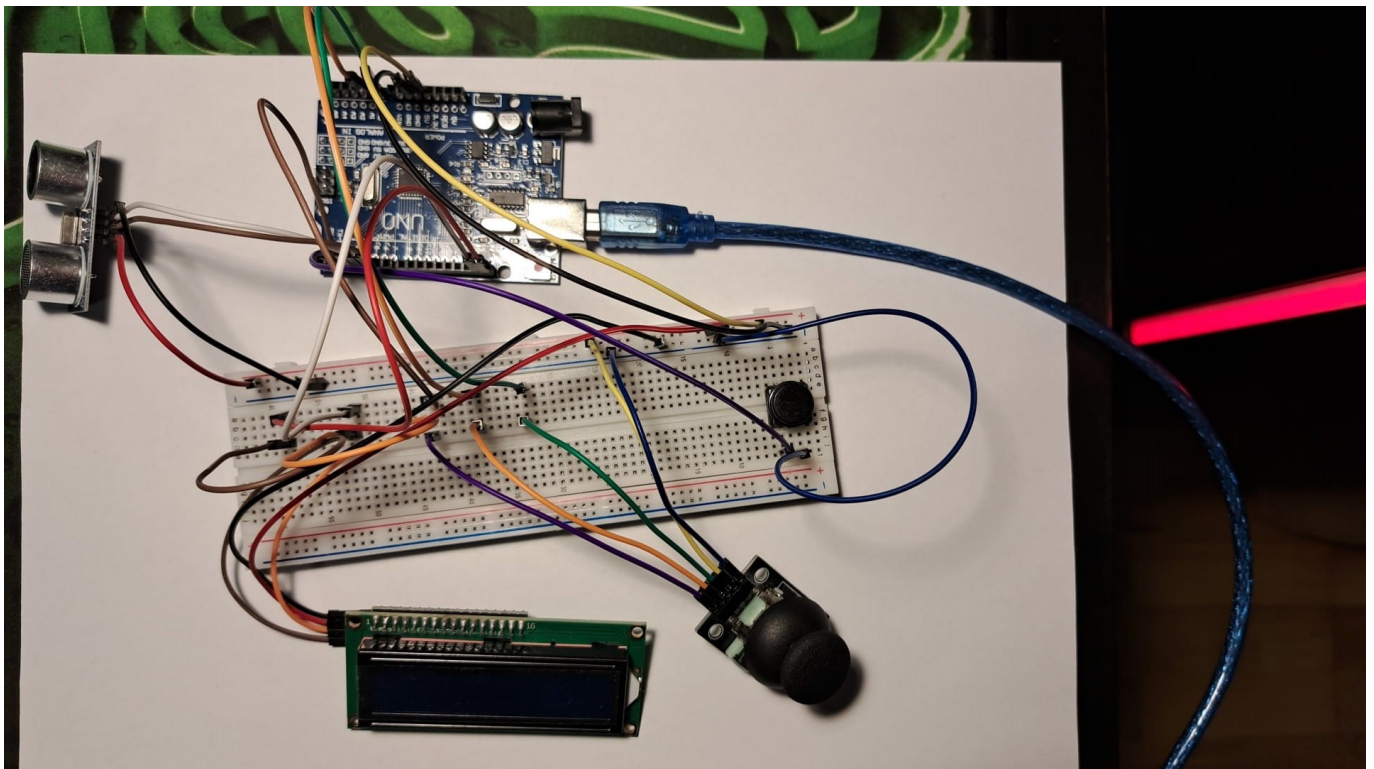


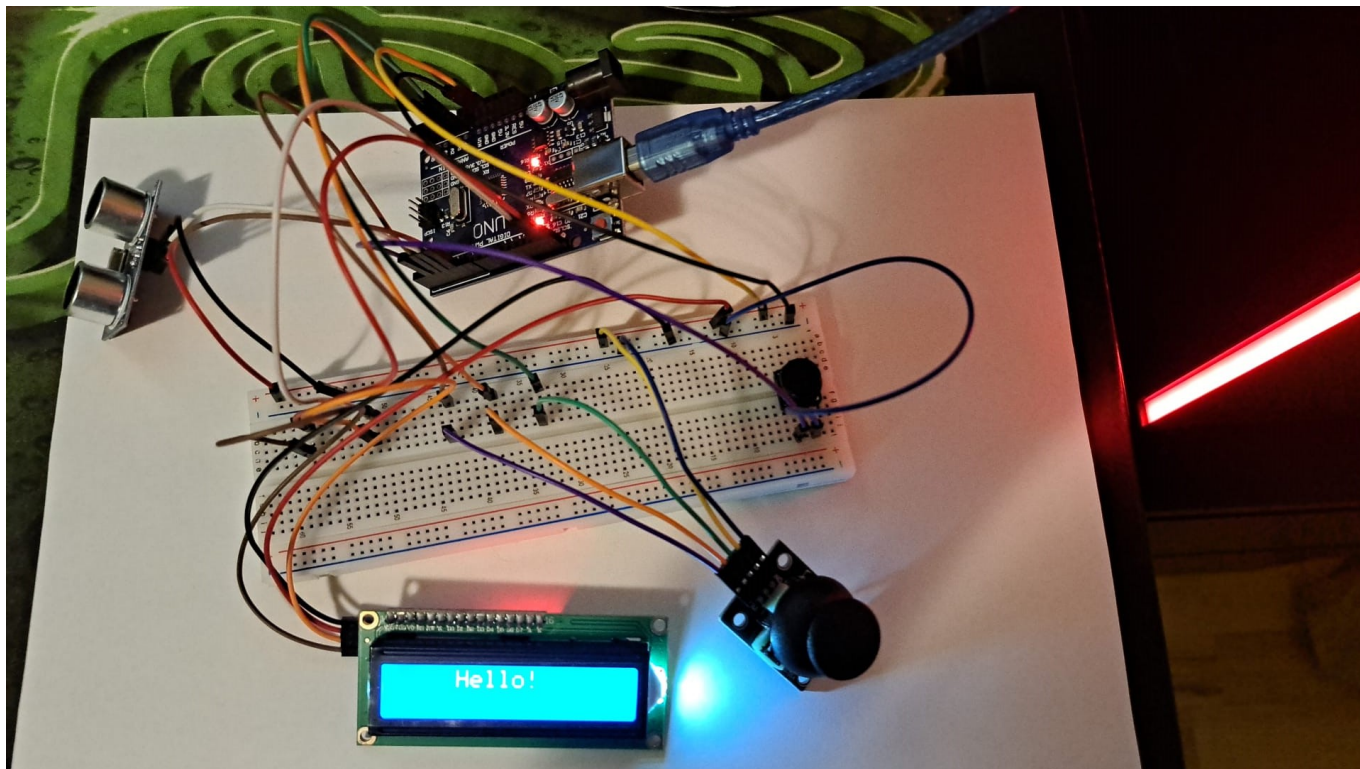
Schemă Electrică

Schemă electrică realizată cu EasyEDA 6.5.5



Poze asamblu hardware





Software Design

Descrierea codului aplicației:

1. mediu de dezvoltare: Arduino IDE 1.8.20
2. librării și surse 3rd-party:
 - LiquidCrystal_I2C.h (librarie 3rd-party pentru controlul unui display LCD conectat prin I2C)
3. algoritmi și structuri (plănuite):
 - structură de date folosită pentru a stoca informațiile despre cont (denumirea site-ului, username-ul și parola);
 - un algoritm de criptare pe baza unei chei generate de mine.

Descrierea wrapper-ului:

1. mediu de dezvoltare: VS Code
2. limbaj de programare folosit: Python3
3. librării și surse 3rd-party:
 - librăria Python "os" (folosită pentru a verifica pe ce sistem de operare rulează wrapper-ul)
 - librăria Python "serial" (folosită pentru a citi datele primite pe interfața serială)
 - librăria Python "selenium" (folosită pentru a automatiza procesul de deschidere a unei pagini web și completarea câmpurilor necesare)
 - chromedriver pentru a permite deschide unei ferestre de Chrome

Descrierea modulelor software

setup():

În funcția de setup() îmi inițializez seriala și lcd-ul, îmi setez modul fiecărui pin pe care îl folosesc, setez întreruperile necesare și încarc datele în vectorul de conturi.

ISR(INT0_vect):

Întreruperea generată de apăsarea butonului pentru a începe procesul de criptare al datelor. După ce generez noua cheie de criptare, folosindu-mă de senzorul de distanță cu ultrasunete, verific dacă datele nu sunt deja criptate, caz în care le voi decripta folosind vechea cheie, urmând ca pe urmă să le criptez cu noua cheie. Am introdus și un debounce de 200ms pentru a nu înregistra aceeași apăsare a butonului de două ori.

ISR(INT1_vect):

Întreruperea generată de apăsarea butonului din joystick care începe procesul de decriptare al datelor alese și trimiterea lor pe interfața serială. Asemănător întreruperii precedente, am adăugat un mecanism de debouncing.

encrypt(val, key):

Funcția de criptare, care primește valoarea ce trebuie criptată și cheia după care criptează. Deoarece bibliotecile cu funcții de criptare ocupă foarte mult din memoria dispozitivului, am ales să îmi implimentez propriul algoritm de criptare, bazat pe shiftarea caracterelor, asemănător Cifrului lui Cezar.

decrypt(val, key):

Funcția de decriptare, care primește o valoare criptată și cheia cu care să decripteze. Aici fac shiftarea inversă față de cea din funcția encrypt().

loop():

Funcția principală, care se execută într-o buclă infinită și se ocupa de coordonarea modulelor. Aici se regăsește și logica de alegere a contului, pe baza inputului de la joystick cât și afișarea mesajelor pe

ecranul LCD.

Python Wrapper:

Întrucât placa de dezvoltare Arduino Uno nu poate fi folosită ca un HID, am creat un wrapper scris în Python care citește informațiile transmise de arduino pe interfața serială și lansează o pagină de Chrome pentru site-ul respectiv unde completează credențialele cu informațiile primite. Acest wrapper a fost testat atât pe Windows cât și pe Linux.

Rezultate Obținute

Link catre Youtube: [Video prezentare](#)

Concluzii

În general nu am întâmpinat probleme în implementare. Mi-am bătut capul cu un folosirea Arduino Uno ca tastatură, însă modelul de placă pe care l-am achiziționat nu permite configurarea chipului de control USB, așa că a trebuit să improvizez, creându-mi un wrapper în Python care ascultă pe interfața serială (un Serial Monitor) și când primește date completează unde trebuie, în funcție de site. Asta m-a limitat la numărul de conturi pe care îl adaug, deoarece logarea pe fiecare site se face diferit și trebuie scrisă o nouă funcție în wrapper pentru fiecare site nou pe care îl introduc.

Download

Arhiva surse: [petredragos_proiectpm.zip](#)

Bibliografie/Resurse

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

Resurse Software

[Utilizare biblioteca selenium](#)

[Utilizare biblioteca serial](#)

Resurse Hardware

[Utilizare senzor distanță cu ultrasunete](#)

[Utilizare display LCD cu modul I2C](#)

[Conectare și utilizare joystick](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/amocanu/dragos.petre>



Last update: **2022/06/01 15:21**