< Implementarea unei masini de scris in alfabetul Braille folosind Arduino >

# Documentul de proiectare

Student: Ene Teodor-Lucian

AIA IV

Grupa 1

Cuprins

[Documentul de proiectare i](#_Toc164027168)

[Cuprins ii](#_Toc164027169)

[1. Introducere 1](#_Toc164027170)

[1.1 Scopul documentului 1](#_Toc164027171)

[2. Prezentare generală și abordări de proiectare 2](#_Toc164027172)

[2.1 Prezentare generală 2](#_Toc164027173)

[2.2 Presupuneri/ Constrângeri/ Riscuri 2](#_Toc164027174)

[3. Considerațiii de proiectare 4](#_Toc164027175)

[3.1 Obiective și linii directoare (ghiduri) 5](#_Toc164027176)

[3.2 Metode de dezvoltare 6](#_Toc164027177)

[3.3 Strategii de arhitectură 6](#_Toc164027178)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 8](#_Toc164027179)

[4.1 Vedere logică 8](#_Toc164027180)

[4.2 Arhitectură hardware 9](#_Toc164027181)

[4.3 Arhitectură software 9](#_Toc164027182)

[4.4 Arhitectura informațiilor 10](#_Toc164027183)

[4.5 Arhitectura de comunicații interne 11](#_Toc164027184)

[5. Proiectarea sistemului 12](#_Toc164027185)

[5.1 Interfețe utilizator 12](#_Toc164027186)

[5.2 Proiectarea interfețelor cu utilizatorul. 14](#_Toc164027187)

[6. Scenarii de utilizare 16](#_Toc164027188)

[7. Proiectare de detaliu 17](#_Toc164027189)

[8. Controale pentru verificarea integrității sistemului 19](#_Toc164027190)

## Introducere

* Numele complet al proiectului: Implementarea unei mașini de scris în alfabetul Braille folosind Arduino
* Acronim: BrailleWriter
* Situația existentă și propusă: Proiectul propus vizează dezvoltarea unei mașini de scris în alfabetul Braille utilizând platforma Arduino Mega și alte componente hardware și software specificate. Situația existentă este că proiectul este în stadiul de proiectare și dezvoltare.
* Evoluția așteptată a documentului: Documentul de Proiectare a Sistemului va evolua pe măsură ce proiectul progresează. Inițial, va conține o prezentare generală a cerințelor și obiectivelor proiectului, arhitectura sistemului și specificațiile detaliate ale componentelor. Pe măsură ce proiectul avansează, documentul va fi actualizat pentru a reflecta modificările și ajustările aduse designului și implementării sistemului.
* Considerații referitoare la securitate sau confidențialitate: Documentul de Proiectare a Sistemului poate conține informații despre arhitectura și funcționalitățile sistemului, precum și despre specificațiile hardware și software. În scopul protejării informațiilor confidențiale și a securității sistemului, este important să se limiteze accesul la acest document numai membrilor autorizați ai echipei de dezvoltare și să se implementeze măsuri adecvate de securitate cibernetică pentru protejarea datelor și a sistemului în ansamblu.

### Scopul documentului

Scopul Documentului de Proiectare a Sistemului este de a oferi o descriere detaliată a arhitecturii și specificațiilor tehnice ale proiectului propus "Implementarea unei mașini de scris în alfabetul Braille folosind Arduino" (acronim: BrailleWriter). Acest document este adaptat pentru a răspunde nevoilor specifice ale acestui proiect și servește drept ghid pentru proiectarea, dezvoltarea și implementarea sistemului.

Principalele obiective ale Documentului de Proiectare a Sistemului sunt:

1. Definirea arhitecturii sistemului;
2. Specificarea detaliată a componentelor;
3. Descrierea interfeței om-mașină;
4. Proiectarea datelor asociate cu sistemul;
5. Considerații referitoare la securitate și confidențialitate.

## Prezentare generală și abordări de proiectare

Această secțiune descrie principiile și strategiile care vor fi utilizate ca ghiduri în momentul proiectării și implementării sistemului.

### Prezentare generală

Proiectul are ca scop dezvoltarea unei mașini de scris specializate, capabilă să transforme textul introdus de utilizator în simboluri Braille și să le imprime pe hârtie. Această mașină de scris este destinată utilizatorilor cu deficiențe de vedere și are ca scop facilitarea comunicării scrise în rândul acestora.

Abordarea sau Organizarea de Bază a Proiectării:

Proiectarea sistemului se bazează pe o arhitectură modulară și integrată, care include atât componentele hardware, cât și cele software, pentru a asigura o funcționare eficientă și fiabilă. Abordarea proiectării este centrată pe nevoile utilizatorului și pe obiectivele de performanță și utilizabilitate.

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

Hardware asociat:

* Funcționarea corectă a dispozitivelor hardware, cum ar fi Arduino Mega, display-ul LCD 1602, tastatura 4x4 și motoarele pas cu pas, este crucială pentru funcționarea adecvată a sistemului. Orice defectare sau incompatibilitate a acestor componente poate afecta performanța și funcționalitatea sistemului.

Software asociat:

* Sistemul depinde de software-ul încorporat pe Arduino Mega pentru a controla și gestiona dispozitivele hardware și pentru a executa logica de procesare a textului și de interfață utilizator.
* Utilizarea unui IDE (Integrated Development Environment) adecvat pentru dezvoltarea și încărcarea codului pe Arduino este o presupunere esențială pentru dezvoltatorii implicați în proiect.

Sistem de operare:

* Sistemul nu are nevoie de un sistem de operare complex, deoarece este bazat pe un microcontroler Arduino. Totuși, este important să se asigure compatibilitatea cu mediile de dezvoltare și încărcare a codului pentru Arduino.

Caracteristicile utilizatorilor finali:

* Se presupune că utilizatorii finali ai sistemului BrailleWriter sunt persoane din invatamant care lucreaza cu copii cu deficiențe de vedere care sunt familiare cu alfabetul Braille și care doresc să comunice în scris folosind acest sistem.
* Interfața utilizatorului trebuie să fie concepută astfel încât să fie accesibilă și ușor de înțeles pentru această categorie de utilizatori, având în vedere nevoile și capacitățile lor.

Posibile modificări ale funcționalității:

* Este posibil ca în timpul dezvoltării și testării să fie identificate necesități suplimentare sau îmbunătățiri ale sistemului. Proiectul trebuie să fie flexibil și adaptabil pentru a face față acestor schimbări și cerințelor viitoare ale utilizatorilor.
* De asemenea, există posibilitatea ca în viitor să fie necesare actualizări ale software-ului pentru a remedia eventuale probleme sau pentru a adăuga noi funcționalități, iar sistemul trebuie să fie proiectat astfel încât să permită actualizări ușoare și fără probleme.

## Considerațiii de proiectare

Cerințele de proiect:

Este esențial să se definească clar și să se înțeleagă cerințele proiectului, inclusiv funcționalitățile necesare, interfețele utilizatorului, performanța dorită și orice alte specificații tehnice sau de design. Acest lucru va oferi o bază solidă pentru dezvoltarea soluției de design.

Compatibilitatea hardware și software:

Trebuie să se confirme că toate componentele hardware și software selectate sunt compatibile între ele și pot funcționa împreună pentru a atinge obiectivele proiectului. Este important să se efectueze teste și evaluări pentru a evita problemele de interoperabilitate în viitor.

Alegerea componentelor hardware adecvate:

Se impune o evaluare atentă a diferitelor opțiuni de componente hardware disponibile pentru a selecta cele mai potrivite pentru proiect. Acestea includ microcontrolerul Arduino Mega, display-ul LCD 1602, tastatura 4x4, motoarele pas cu pas și electromagnetul.

Interacțiunea utilizatorului:

Trebuie să se ia în considerare modul în care utilizatorii vor interacționa cu sistemul și să se proiecteze o interfață utilizator intuitivă și ușor de utilizat, adaptată nevoilor și capacităților utilizatorilor finali.

Securitatea și confidențialitatea datelor:

Este important să se asigure că datele utilizatorilor sunt protejate împotriva accesului neautorizat și că sistemul respectă standardele și reglementările privind securitatea și confidențialitatea datelor.

Testarea și validarea soluției:

Înainte de implementarea finală a soluției, aceasta trebuie să fie supusă unor teste riguroase pentru a verifica funcționalitatea corectă, performanța și fiabilitatea sistemului. Este esențial să se valideze soluția în condiții reale de utilizare pentru a identifica și remedia eventuale probleme.

### Obiective și linii directoare (ghiduri)

* Eficiența și performanța: Prioritatea este acordată eficienței și performanței sistemului pentru a asigura o experiență fluidă și fără probleme pentru utilizatori. Acest lucru implică optimizarea algoritmilor și a codului pentru a obține o viteză ridicată de procesare a textului și a deplasării hârtiei, fără a compromite calitatea sau exactitatea.
* Utilizabilitatea: Accentul este pus pe crearea unei interfețe utilizator intuitive și ușor de folosit, care să ofere o experiență plăcută utilizatorilor, inclusiv celor cu deficiențe de vedere. Linii directoare de utilizabilitate sunt aplicate pentru a asigura accesibilitatea și navigarea simplă prin meniuri și opțiuni.
* Fiabilitatea și stabilitatea: Un alt obiectiv important este asigurarea fiabilității și stabilității sistemului, pentru a preveni eșecurile și erorile în timpul utilizării. Acest lucru implică testarea riguroasă a sistemului și implementarea măsurilor de gestionare a erorilor și recuperare în caz de probleme.
* Optimizarea resurselor: Se urmărește utilizarea eficientă a resurselor hardware și software disponibile pentru a maximiza performanța și pentru a minimiza consumul de memorie și energie. De exemplu, se pot alege algoritmi și tehnici de programare care să ofere un echilibru optim între viteză și utilizarea memoriei.
* Consistența și standardizare: Se aplică linii directoare și convenții de codare pentru a asigura consistența și standardizarea în întregul cod sursă al sistemului. Acest lucru facilitează înțelegerea și întreținerea codului de către membrii echipei și contribuie la reducerea erorilor și a ambiguităților.
* Adaptabilitatea și extensibilitatea: Sistemul este proiectat să fie adaptabil și extensibil, permițând adăugarea și modificarea facilă a funcționalităților în viitor. Acest lucru se realizează prin utilizarea unei arhitecturi modulare și a unei codificări flexibile.
* Compatibilitatea și portabilitatea: Se acordă atenție compatibilității și portabilității sistemului, pentru a asigura interoperabilitatea cu diferite configurații hardware și software și pentru a permite utilizarea sistemului într-o varietate de medii și contexte.

### Metode de dezvoltare

Pentru designul sistemului și al software-ului pentru proiectul "Implementarea unei mașini de scris în alfabetul Braille folosind Arduino" (BrailleWriter), este adoptată o abordare orientată pe obiecte. Această abordare este aleasă datorită flexibilității și modularității sale, ceea ce permite o proiectare și implementare eficientă a sistemului.

În cadrul abordării orientate pe obiecte, se utilizează limbajul de programare C++ și mediul de dezvoltare Arduino IDE pentru codificarea și implementarea soluției. Principiile de proiectare și programare orientată pe obiecte sunt aplicate pentru a organiza și structura codul în clase și obiecte, fiecare responsabil pentru funcționalități specifice ale sistemului.

Deși nu este necesar să se utilizeze formal o metodă publicată pentru proiectul BrailleWriter, se iau în considerare anumite aspecte ale UML (Unified Modeling Language) pentru a realiza diagrame de proiectare a arhitecturii sistemului și a relațiilor între componentele sale. Acest lucru ajută la clarificarea și comunicarea conceptelor și interacțiunilor dintre diferitele module și componente ale sistemului.

Contingențe care ar putea apărea în designul sistemului și al software-ului includ:

* Lipsa acordurilor de interfață cu agențiile externe: Dacă sistemul trebuie să se integreze cu alte sisteme sau dispozitive externe, lipsa acordurilor de interfață sau a documentației adecvate poate afecta proiectarea și implementarea. În acest caz, este necesar să se facă eforturi suplimentare pentru a obține informațiile necesare sau pentru a găsi alternative și soluții de rezervă.
* Arhitecturi instabile: Schimbările neprevăzute în arhitecturile sau platformele de dezvoltare pot afecta planurile de proiectare și implementare. Este important să se monitorizeze și să se adapteze la modificările în timp real și să se identifice soluții alternative sau planuri de rezervă pentru a face față acestor contingente.

### Strategii de arhitectură

1. Utilizarea limbajului de programare C++ și a mediului de dezvoltare Arduino IDE: Această decizie a fost luată pentru a beneficia de simplitatea și eficiența limbajului C++ și pentru a utiliza mediul de dezvoltare Arduino IDE, care oferă unelte și biblioteci specifice pentru dezvoltarea aplicațiilor bazate pe Arduino.

2. Structurarea codului în clase și obiecte: Am decis să folosim o abordare orientată pe obiecte pentru a organiza codul în clase și obiecte, fiecare responsabil pentru funcționalități specifice ale sistemului, cum ar fi controlul dispozitivelor hardware, interfața utilizator și procesarea textului.

3. Reutilizarea de biblioteci și componente software existente: Am ales să folosim biblioteci și componente software existente, cum ar fi bibliotecile pentru comunicarea cu dispozitivele hardware și pentru procesarea textului, pentru a accelera dezvoltarea și pentru a asigura fiabilitatea și eficiența sistemului.

4. Planuri pentru extinderea și îmbunătățirea software-ului: Am planificat să dezvoltăm și să îmbunătățim continuu software-ul pentru a adăuga noi funcționalități și pentru a îmbunătăți performanța și utilizabilitatea sistemului pe baza feedback-ului utilizatorilor și a cerințelor suplimentare.

5. Paradigma interfeței utilizatorului: Interfața utilizatorului este concepută pentru a fi intuitivă și ușor de utilizat, cu meniuri clare și opțiuni de navigare, adaptată nevoilor și capacităților utilizatorilor finali, inclusiv celor cu deficiențe de vedere.

6. Detectarea și recuperarea erorilor: S-au implementat mecanisme de detectare și recuperare a erorilor pentru a gestiona situațiile neașteptate și pentru a asigura o funcționare stabilă și fiabilă a sistemului.

7. Gestionarea memoriei și resurselor: Politici și strategii adecvate de gestionare a memoriei și a resurselor sunt implementate pentru a optimiza utilizarea acestora și pentru a evita consumul excesiv sau ineficient.

8. Comunicarea și sincronizarea: S-au adoptat mecanisme adecvate de comunicare și sincronizare între diferitele componente ale sistemului pentru a asigura o interacțiune corectă și coordonată între acestea.

## Arhitectura Sistemului și Proiectarea Arhitecturii

Interfața utilizatorului (UI): Această componentă gestionează interacțiunea cu utilizatorul și afișează informații pe ecranul LCD și primește comenzi de la tastatură. UI traduce intrările utilizatorului în comenzi pentru celelalte componente ale sistemului și afișează rezultatele proceselor.

Procesarea textului și convertirea în Braille: Această componentă primește intrările textuale de la interfața utilizatorului și le procesează pentru a fi convertite în alfabetul Braille. Acest lucru implică maparea caracterelor și cuvintelor la echivalentele lor Braille și generarea instrucțiunilor necesare pentru poziționarea electromagnetului pentru a crea puncte Braille pe hârtie.

Controlul hardware-ului: Această componentă gestionează controlul dispozitivelor hardware, cum ar fi motoarele pas cu pas pentru deplasarea hârtiei, electromagnetul pentru crearea punctelor Braille și interacțiunea cu tastatura și ecranul LCD. Acesta primește comenzi de la procesarea textului și de la interfața utilizatorului și le traduce în comenzi specifice pentru dispozitivele hardware.

Comunicarea între componente: Pentru a asigura o comunicare eficientă între componente, se utilizează tipare de proiectare cum ar fi Observer pentru a notifica componente despre modificările în stare și Strategy pentru a selecta algoritmi diferiți pentru procesare textului în funcție de cerințe specifice.

Deși există mai multe modalități de a decompune sistemul în componente și subsisteme, această abordare a fost aleasă pentru a separa responsabilitățile și pentru a permite extinderea și modificarea individuală a fiecărei componente. Prin separarea clară a funcționalității, fiecare componentă poate fi dezvoltată, testată și întreținută independent, ceea ce duce la un sistem mai modular și mai ușor de gestionat.

### Vedere logică

În cadrul proiectului vederile logice relevante sunt documentate în Documentul de Cerințe și Documentul de Proiectare a Sistemului. Aceste documente descriu cerințele funcționale și non-funcționale ale sistemului, precum și detaliile designului și arhitecturii acestuia.

Documentul de Cerințe conține o listă detaliată a cerințelor sistemului, inclusiv funcționalitățile dorite, interfețele utilizatorului, performanța așteptată și orice alte specificații tehnice sau de design. Acesta servește drept bază pentru dezvoltarea și implementarea sistemului, asigurându-se că soluția finală îndeplinește nevoile și așteptările stabilite.

Documentul de Proiectare a Sistemului prezintă o descriere detaliată a arhitecturii sistemului, includând modul în care funcționalitatea este împărțită în componente și subsisteme, precum și interacțiunile dintre acestea. Acesta oferă o înțelegere generală a modului în care sistemul a fost proiectat să funcționeze și cum sunt gestionate datele și procesele în cadrul acestuia.

Aceste documente sunt actualizate și revizuite pe măsură ce proiectul progresează și oferă o bază solidă pentru dezvoltarea și implementarea sistemului BrailleWriter.

### Arhitectură hardware

Arduino Mega: Este componenta centrală a sistemului, care controlează toate celelalte componente hardware. Arduino Mega este responsabil pentru procesarea textului, generarea instrucțiunilor pentru deplasarea hârtiei și acționarea electromagnetului pentru crearea punctelor Braille.

Display LCD 1602 cu interfață I2C: Acesta este utilizat pentru afișarea informațiilor către utilizator, inclusiv textul introdus și instrucțiunile de operare.

Tastatură 4x4: Tastatura este folosită pentru introducerea textului și a comenzilor de către utilizator.

Motoare pas cu pas 28BYJ-48 cu driver ULN2003: Acestea sunt utilizate pentru deplasarea electromagnetului și a hârtiei în sus și în jos, în timp ce un rând este scris.

Electromagnet de tragere: Acesta este utilizat pentru crearea punctelor Braille pe hârtie prin tragerea unui ac la locul potrivit.

### Arhitectură software

1. Software-ul încorporat pe Arduino Mega:

- \*\*Codul de control al mașinii de scris:\*\* Acesta este codul principal care gestionează procesarea textului, deplasarea hârtiei și acționarea electromagnetului pentru crearea punctelor Braille. Este scris în limbajul de programare C++ folosind mediul de dezvoltare Arduino IDE.

2. Software pentru interfața utilizatorului (UI):

Codul pentru interacțiunea cu tastatura și afișarea pe ecran LCD: Acest software este responsabil pentru interacțiunea cu utilizatorul prin intermediul tastaturii și afișajului LCD. Este scris în limbajul de programare C++ și folosește biblioteci specifice pentru a comunica cu aceste dispozitive.

3. Software pentru conversia textului în Braille:

Codul de conversie a textului: Acesta este software-ul responsabil pentru transformarea textului introdus de utilizator în echivalente Braille. Acesta poate implica maparea caracterelor și cuvintelor la echivalentele lor Braille și generarea instrucțiunilor necesare pentru controlul electromagnetului.

4. Software de simulare și de testare (opțional):

Codul pentru simularea și testarea sistemului: Pentru dezvoltare și depanare, pot fi dezvoltate instrumente de simulare și testare pentru a verifica funcționalitatea și performanța sistemului BrailleWriter.

### Arhitectura informațiilor

1. Textul introdus de utilizator: Aceasta este informația de bază care va fi introdusă manual de către utilizator folosind tastatura. Acest text va fi prelucrat de sistem și transformat în puncte Braille pe hârtie.

2. Instrucțiunile de operare și setările utilizatorului: Utilizatorul poate introduce comenzi sau setări specifice pentru controlul mașinii de scris, cum ar fi setările de aliniere, spațiere sau formatarea textului. Aceste informații sunt introduse de utilizator și sunt stocate temporar în memoria sistemului până când sunt utilizate pentru a controla procesul de scriere.

3. Datele de configurare și de calibrare a sistemului: Informații precum viteza și direcția de deplasare a hârtiei, coordonatele punctelor Braille și setările pentru controlul electromagnetului sunt stocate în sistem pentru a asigura funcționarea corectă și precisă a mașinii de scris.

4. Starea și diagnosticarea sistemului: Informații despre starea sistemului și eventualele erori sau probleme întâlnite pot fi stocate pentru a ajuta la diagnosticarea și depanarea problemelor în timpul utilizării.

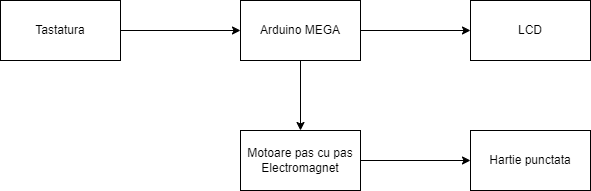
5. Informații sensibile: În funcție de utilizarea și contextul sistemului, textul introdus de utilizator ar putea include informații sensibile, cum ar fi date personale sau informații medicale. Deși sistemul BrailleWriter în sine nu este conceput pentru a gestiona informații sensibile în mod specific, este important să se ia măsuri adecvate pentru a proteja confidențialitatea și securitatea datelor atunci când sunt utilizate astfel de informații.

Datele sunt furnizate sistemului manual de către utilizator, prin introducerea lor directă prin intermediul tastaturii. Nu există import automat de date sau conectivitate la alte surse de date externe în cadrul sistemului BrailleWriter.

### Arhitectura de comunicații interne

În cadrul sistemului nu există o rețea de comunicații complexă, deoarece sistemul este în mare parte autonom și încorporat. Totuși, se poate considera interfața dintre componentele hardware și software ca fiind o rețea simplă de comunicații.

Arhitectura de comunicații implementată este una de tip point-to-point, în care componentele sistemului comunică direct între ele, fără a fi necesar un echipament intermediar, cum ar fi un router sau un switch.



În această diagramă simplificată, componentele hardware (tastatura, Arduino Mega, motoarele pas cu pas, electromagnetul și display-ul LCD) sunt conectate direct între ele. Tastatura furnizează date către Arduino Mega, care le procesează și controlează componentele hardware pentru a executa operațiile necesare. Apoi, rezultatele sunt afișate pe display-ul LCD și/sau produse fizic pe hârtie.

## Proiectarea sistemului

### Interfețe utilizator

Utilizatorii finali:

* Descriere: Utilizatorii finali sunt persoanele care folosesc mașina de scris Braille pentru a introduce și a imprima text în alfabetul Braille.
* Număr total de utilizatori anticipați: Depinde de mediul în care este utilizat sistemul, dar poate fi estimat între 1 și 10 utilizatori, în funcție de context.
* Număr maxim de utilizatori simultani: De obicei, doar un utilizator poate folosi sistemul în același timp, deoarece este o mașină de scris personală.
* Numărul de utilizatori externi: Nu se aplică.

Personalul de suport operațional:

* Descriere: Personalul de suport operațional este responsabil pentru întreținerea și asistența tehnică a sistemului BrailleWriter în cazul în care apar probleme tehnice sau de funcționare.
* Număr total de utilizatori anticipați: Poate fi estimat la 1 sau 2 persoane, în funcție de necesități.
* Număr maxim de utilizatori simultani: De obicei, doar un membru al personalului de suport operațional interacționează cu sistemul în același timp pentru a rezolva problemele.
* Numărul de utilizatori externi: Nu se aplică.

Personalul de întreținere:

* Descriere: Personalul de întreținere este responsabil pentru mentenanța fizică și repararea echipamentelor hardware ale sistemului BrailleWriter, cum ar fi înlocuirea componentelor defecte.
* Număr total de utilizatori anticipați: Poate fi estimat la 1 sau 2 persoane, în funcție de necesități.
* Număr maxim de utilizatori simultani: De obicei, doar un membru al personalului de întreținere lucrează la sistem în același timp pentru a efectua întreținerea sau repararea.
* Numărul de utilizatori externi: Nu se aplică.

#### Intrări

Tastatura 4x4:

Descriere: Utilizatorul introduce textul și comenzile folosind o tastatură matriceală 4x4. Tastatura este utilizată pentru introducerea textului în alfabetul standard și pentru a selecta opțiuni sau comenzi specifice.

Ecranul LCD 1602:

Descriere: Ecranul LCD 1602 afișează textul introdus de utilizator și informațiile de stare ale sistemului. Acesta oferă o interfață vizuală pentru utilizator pentru a vedea ceea ce a fost introdus și pentru a primi feedback de la sistem.

Maparea către fluxurile de date la nivel înalt poate fi reprezentată astfel:

Tastatura 4x4 --> Arduino Mega --> Procesare și control sistem --> Ecran LCD 1602

Aspectul ecranului de intrare de date (GUI) pentru utilizator constă în ecranul LCD 1602, care afișează un rând de text pentru introducerea utilizatorului și mesaje de stare sau de feedback.

Criteriile de editare pentru elementele de date includ:

* Textul introdus: Oricare text alfanumeric standard este acceptat.
* Comenzile și opțiunile: Se pot introduce anumite comenzi sau opțiuni specifice, cum ar fi aliniere, spațiere sau formatarea textului.
* Lungimea: Limitată la lungimea ecranului LCD 1602.
* Obligatoriu/opțional: Textul introdus este de obicei obligatoriu, dar comenzile opționale pot fi furnizate de utilizator.

#### Ieșiri

Ecranul LCD 1602:

* Cod/Nume: Ecran principal
* Conținut: Afișează textul introdus de utilizator și informații de stare ale sistemului, cum ar fi confirmările comenzilor și mesajele de eroare.
* Scopul ieșirii: Furnizează feedback vizual utilizatorului cu privire la acțiunile întreprinse și la starea sistemului.
* Utilizatori principali: Utilizatorii finali ai sistemului.

Rapoarte:

* Cod/Nume: Raport de activitate
* Conținut: Acest raport poate include informații despre activitatea utilizatorului, cum ar fi textul introdus, comenzile efectuate și starea sistemului în timpul utilizării.
* Scopul ieșirii: Furnizează o înregistrare a activității utilizatorului și a stării sistemului pentru scopuri de auditare sau de referință.
* Utilizatori principali: Personalul de suport operațional, personalul de întreținere.

Interfețe grafice și ecrane de afișare a datelor:

* Cod/Nume: Interfață de configurare
* Conținut: Afișează opțiunile de configurare disponibile și permite utilizatorului să modifice setările sistemului, cum ar fi viteza de deplasare a mașinii de scris sau setările de formatare a textului.
* Scopul ieșirii: Furnizează utilizatorului o interfață grafică pentru a interacționa cu sistemul și a modifica setările într-un mod intuitiv.
* Utilizatori principali: Utilizatorii finali ai sistemului.

### Proiectarea interfețelor cu utilizatorul.

Interfața utilizatorului pe ecranul LCD 1602:

* Descriere: Această interfață permite utilizatorului să interacționeze cu sistemul și să primească feedback vizual. Utilizatorul poate introduce text și comenzi folosind tastatura și poate vedea rezultatele pe ecranul LCD 1602.
* Elemente de interfață:
* Afișaj pentru textul introdus și mesaje de stare.
* Indicatoare pentru starea sistemului (cum ar fi bateria).
* Prompt-uri pentru comenzi și opțiuni disponibile.
* Mesaje de eroare în cazul introducerii incorecte a datelor sau a comenzilor.
* Funcționalități:
* Afișare text introdus și comenzi.
* Confirmarea comenzilor și afișarea mesajelor de stare.
* Avertizarea în cazul unor erori sau a unor situații anormale.

Interfața de configurare:

* Descriere: Această interfață permite utilizatorului să modifice setările sistemului, cum ar fi viteza de deplasare a mașinii de scris sau setările de formatare a textului. Interfața poate fi accesată prin intermediul unor comenzi specifice sau direct de pe ecranul LCD, în funcție de implementare.
* Elemente de interfață:
* Opțiuni de configurare, cum ar fi viteza de deplasare sau formatarea textului.
* Prompt-uri pentru introducerea datelor de configurare.
* Confirmări pentru modificările efectuate.
* Funcționalități:
* Modificarea setărilor sistemului.
* Confirmarea și salvarea modificărilor.
* Afișarea mesajelor de confirmare sau avertizare în cazul unor erori sau a unor acțiuni nereușite.

## Scenarii de utilizare

Funcționalitatea generală a sistemului este de a permite utilizatorilor să introducă și să imprime text în alfabetul Braille folosind un dispozitiv hardware bazat pe Arduino. Sistemul oferă o interfață intuitivă și ușor de utilizat pentru introducerea textului și configurarea diverselor setări.

Scenariile operaționale pentru sistemul BrailleWriter ar putea include:

1. Scenariu de introducere a textului:

- Utilizatorul pornește sistemul BrailleWriter.

- Pe ecranul LCD 1602, utilizatorul vede un prompt pentru introducerea textului.

- Utilizatorul folosește tastatura 4x4 pentru a introduce textul dorit.

- Pe ecranul LCD, textul introdus este afișat pentru confirmare.

- Utilizatorul finalizează introducerea textului, apoi apasă o comandă specifică pentru a imprima textul în alfabetul Braille.

2. Scenariu de configurare a setărilor:

- Utilizatorul accesează interfața de configurare.

- Pe ecranul LCD 1602, utilizatorul vede opțiuni pentru configurarea sistemului, cum ar fi viteza de deplasare a mașinii de scris.

- Utilizatorul selectează opțiunile dorite folosind tastatura și confirmă modificările.

- Sistemul salvează setările configurate și le aplică în funcționarea ulterioară a sistemului.

3. Scenariu de raportare a activității:

- Personalul de suport operațional accesează sistemul pentru a vizualiza rapoartele de activitate.

- Pe ecranul LCD 1602, personalul vede opțiunea de afișare a rapoartelor.

- Utilizatorul selectează raportul dorit și îl afișează.

- Sistemul afișează raportul de activitate, inclusiv informații despre textul introdus, comenzile efectuate și starea sistemului.

## Proiectare de detaliu

Pentru a construi și integra efectiv componentele hardware, codifica și integra componentele software și interconecta segmentele hardware și software într-un produs funcțional, echipa de dezvoltare va avea nevoie de următoarele informații și proceduri:

1. Componente hardware:

- Specificațiile complete ale fiecărei componente hardware, inclusiv Arduino Mega, display-ul LCD 1602 cu interfață I2C, tastatura 4x4, motoarele pas cu pas, electromagnetul și releul.

- Instrucțiuni detaliate de asamblare pentru fiecare componentă hardware și conexiunile necesare între acestea.

- Schemă și diagrame de conexiune pentru interfațele hardware și modulele electronice, cum ar fi modulul de drivere pentru motoarele pas cu pas și releul pentru alimentarea electromagnetului.

- Informații despre alimentarea cu energie și sursele de putere necesare pentru fiecare componentă.

2. Componente software:

- Codul sursă pentru software-ul Arduino, inclusiv codul pentru controlul motoarelor pas cu pas, interacțiunea cu tastatura și afișajul LCD, gestionarea datelor introduse și procesarea comenzilor utilizatorului.

- Instrucțiuni de instalare și configurare a mediului de dezvoltare Arduino IDE și a oricăror biblioteci necesare.

- Documentație detaliată a funcționalităților software și a interfețelor de programare a aplicațiilor (APIs) pentru integrare și dezvoltare ulterioară.

3. Integrarea componentelor hardware și software:

- Proceduri și ghiduri pentru testarea și depanarea fiecărei componente hardware și a funcționalităților software în parte.

- Instrucțiuni pentru integrarea și testarea interacțiunilor dintre componentele hardware și software.

- Proceduri pentru calibrarea și optimizarea sistemului, cum ar fi ajustarea vitezei de deplasare a mașinii de scris sau setările de sensibilitate ale tastaturii.

4. Interconectarea segmentelor hardware și software:

- Diagrame detaliate ale fluxului de date între componentele hardware și software, inclusiv comunicarea dintre Arduino Mega și modulele periferice.

- Configurarea și testarea conexiunilor de comunicație, cum ar fi conexiunea I2C între Arduino și ecranul LCD sau gestionarea interfețelor de comunicare serială pentru trimiterea și primirea datelor între diferite componente.

- Proceduri pentru testarea integrată a întregului sistem și a funcționalităților sale, asigurându-se că toate componentele hardware și software funcționează corect împreună.

5. Combinarea pachetelor separate externe:

- Identificarea și selecția adecvată a pachetelor externe necesare pentru a îndeplini cerințele sistemului.

- Proceduri de instalare și configurare a acestor pachete externe, precum drivere pentru dispozitivele hardware, biblioteci de software sau alte componente suplimentare.

- Testarea și integrarea pachetelor externe în sistemul existent, asigurându-se că funcționează în mod corespunzător și nu afectează negativ alte funcționalități.

## Controale pentru verificarea integrității sistemului

1. Securitate internă:

- Accesul la datele critice trebuie restricționat utilizatorilor/operatorilor în funcție de tipurile de acces necesare pentru îndeplinirea sarcinilor lor.

- Implementarea unui sistem de autentificare și autorizare care să permită accesul utilizatorilor doar la datele și funcționalitățile la care au drepturi legale.

- Criptarea datelor sensibile pentru a proteja integritatea și confidențialitatea acestora în timpul tranzitului și în stocare.

2. Proceduri de audit:

- Definirea și implementarea procedurilor de audit care să respecte cerințele de control, raportare și perioade de reținere pentru rapoartele operaționale și de management.

- Generarea de jurnale de audit care să înregistreze toate activitățile relevante, inclusiv accesul la date, modificările și acțiunile utilizatorilor/operatorilor.

- Păstrarea înregistrărilor de audit pentru o perioadă specificată conform reglementărilor și cerințelor organizaționale.

3. Piste de auditare a aplicațiilor:

- Implementarea unor mecanisme de auditare a aplicațiilor pentru a monitoriza dinamic accesul la datele critice designate.

- Utilizarea de instrumente de monitorizare a activității aplicațiilor pentru a detecta și a preveni activitățile neautorizate sau abuzive.

4. Tabele standard pentru validarea câmpurilor de date:

- Definirea și implementarea unor tabele standard pentru validarea câmpurilor de date în baza de date.

- Asigurarea că datele introduse în sistemele sunt conforme cu formatele și regulile specificate în tabelele standard.

5. Procese de verificare pentru adăugarea, ștergerea sau actualizarea datelor critice:

- Implementarea unor procese de verificare și aprobare pentru orice adăugare, ștergere sau actualizare a datelor critice.

- Aplicarea unui principiu de separare a responsabilităților pentru a evita conflictele de interese și abuzul de putere în gestionarea datelor critice.

6. Capacitatea de a identifica informațiile de auditare:

- Implementarea unui sistem de identificare a informațiilor de auditare, care să includă identificarea utilizatorului, a terminalului de rețea, dată, oră și datele accesate sau modificate.

- Utilizarea unor unelte și tehnologii adecvate pentru a gestiona și a accesa eficient informațiile de auditare atunci când este necesar.