|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ  DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI |  |

PROIECT DE DIPLOMĂ

Nicolas Sebastian Manea

COORDONATOR ȘTIINȚIFIC

Prof. Dr. Ing. Ioan Stelian Lemeni

Septembrie 2023

CRAIOVA

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ  DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI |  |

*Configurator pentru matrice de LED-uri*

Nicolas Sebastian Manea

COORDONATOR ȘTIINȚIFIC

Prof. Dr. Ing. Ioan Stelian Lemeni

Septembrie 2023

CRAIOVA

*„Învățătura este o comoară care își urmează stăpânul pretutindeni.”*

Proverb popular

**DECLARAȚIE DE ORIGINALITATE**

Subsemnatul Manea Nicolas Sebastian student la specializarea Calculatoare cu predare în limba română din cadrul Facultății de Automatică, Calculatoare și Electronică a Universității din Craiova, certific prin prezenta că am luat la cunoștință de cele prezentate mai jos și că îmi asum, în acest context, originalitatea proiectului meu de licenţă:

* cu titlul Configurator pentru matrice de LED-uri
* coordonată de Ioan Stelian Lemeni
* prezentată în sesiunea Septembrie 2023

La elaborarea proiectului de licenţă, se consideră plagiat una dintre următoarele acţiuni:

* reproducerea exactă a cuvintelor unui alt autor, dintr-o altă lucrare, în limba română sau prin traducere dintr-o altă limbă, dacă se omit ghilimele şi referinţa precisă,
* redarea cu alte cuvinte, reformularea prin cuvinte proprii sau rezumarea ideilor din alte lucrări, dacă nu se indică sursa bibliografică,
* prezentarea unor date experimentale obţinute sau a unor aplicaţii realizate de alţi autori fără menţionarea corectă a acestor surse,
* însuşirea totală sau parţială a unei lucrări în care regulile de mai sus sunt respectate, dar care are alt autor.

Pentru evitarea acestor situaţii neplăcute se recomandă:

* plasarea între ghilimele a citatelor directe şi indicarea referinţei într-o listă corespunzătoare la sfărşitul lucrării,
* indicarea în text a reformulării unei idei, opinii sau teorii şi corespunzător în lista de referinţe a sursei originale de la care s-a făcut preluarea,
* precizarea sursei de la care s-au preluat date experimentale, descrieri tehnice, figuri, imagini, statistici, tabele et caetera,
* precizarea referinţelor poate fi omisă dacă se folosesc informaţii sau teorii arhicunoscute, a căror paternitate este unanim cunoscută și acceptată.

Data, Semnătura candidatului,

13/09/2023

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  Facultatea de Automatică, Calculatoare şi Electronică  Departamentul de Calculatoare și Tehnologia Informației | Aprobat la data de  …………………  Şef de departament,  Prof. dr. ing.  Nicolae Enescu |

**PROIECTUL DE DIPLOMĂ**

|  |  |
| --- | --- |
| Numele și prenumele studentului/-ei: |  |
| Enunțul temei: | Configurator pentru matrice de led-uri. Scopul principal al acestui proiect este de a creea un site cu ajutorul căruia se poate programa o matrice de led-uri 8x8. |
| Datele de pornire: | Dezvoltarea unui site pentru un configurator. |
| Conținutul proiectului: | 1. Introducere 2. Domeniul de cunoștințe si instrumente 3. Design-ul si implementarea 4. Termenii de utilizare 5. Concluzii 6. Bibliografie 7. Referințe Web |
| Material grafic obligatoriu: | 1. Documentația Proiectului 2. Prezentare PowerPoint 3. Imagini Reprezentative |
| Consultații: | Periodice |
| Conducătorul științific  (titlul, nume și prenume, semnătura): | Profesor Dr. Ing. Stelian Ioan Lemeni |
| Data eliberării temei: | 11.10.2022 |
| Termenul estimat de predare a proiectului: | 14.09.2023 |
| Data predării proiectului de către student și semnătura acestuia: | 11.09.2023 |

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  Facultatea de Automatică, Calculatoare şi Electronică  Departamentul de Calculatoare și Tehnologia Informației |  |

**REFERATUL CONDUCĂTORULUI ȘTIINȚIFIC**

|  |  |
| --- | --- |
| Numele și prenumele candidatului/-ei: |  |
| Specializarea: | Calculatoare cu predare in limba romana |
| Titlul proiectului: | Configurator pentru matrice de LED-uri |
| Locația în care s-a realizat practica de documentare (se bifează una sau mai multe din opțiunile din dreapta): | În facultate ☒ |
| În producție □ |
| În cercetare □ |
| Altă locație: |

În urma analizei lucrării candidatului au fost constatate următoarele:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nivelul documentării | | Insuficient  □ | Satisfăcător □ | Bine  □ | Foarte bine  □ |
| Tipul proiectului | | Cercetare  □ | Proiectare  □ | Realizare practică □ | Altul |
| Aparatul matematic utilizat | | Simplu  □ | Mediu  □ | Complex □ | Absent  □ |
| Utilitate | | Contract de cercetare □ | Cercetare internă □ | Utilare  □ | Altul |
| Redactarea lucrării | | Insuficient  □ | Satisfăcător □ | Bine  □ | Foarte bine  □ |
| Partea grafică, desene | | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Realizarea practică | Contribuția autorului | Insuficientă  □ | Satisfăcătoare □ | Mare  □ | Foarte mare  □ |
| Complexitatea  temei | Simplă  □ | Medie  □ | Mare  □ | Complexă  □ |
| Analiza cerințelor | Insuficient  □ | Satisfăcător □ | Bine  □ | Foarte bine  □ |
| Arhitectura | Simplă  □ | Medie  □ | Mare  □ | Complexă  □ |
| Întocmirea specificațiilor funcționale | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Implementarea | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Testarea | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Funcționarea | Da  □ | Parțială  □ | Nu  □ | |
| Rezultate experimentale | | Experiment propriu  □ | | Preluare din bibliografie  □ | |
| Bibliografie | | Cărți | Reviste | Articole | Referințe web |
| Comentarii  și  observații | |  | | | |

În concluzie, se propune:

|  |  |
| --- | --- |
| ADMITEREA PROIECTULUI  □ | RESPINGEREA PROIECTULUI  □ |

Data, Semnătura conducătorului științific,

**REZUMATUL PROIECTULUI**

Scopul acestui cod este de a crea o interfață care permite utilizatorilor să genereze cod pentru o matrice LED (WS2812) conectată la un modul Arduino. Interfața constă dintr-o listă derulabilă, un tabel și diverse butoane și câmpuri de introducere.

Structura HTML a paginii este împărțită în diferite secțiuni folosind elemente div și clase CSS. Stilul CSS definește aspectul și aspectul elementelor. Fundalul este setat la un gradient liniar, iar o imagine este utilizată ca imagine de fundal.

Secțiunea listă derulabilă conține o listă cu un titlu. Fiecare element din listă reprezintă o stare salvată a matricei LED. Elementele din listă pot fi selectate făcând clic pe ele, iar o stare selectată poate fi ștearsă folosind butonul "Ștergeți". Starea selectată este evidențiată vizual.

Secțiunea tabelului conține o masă 8x8 reprezentând matricea LED. Fiecare celulă de tabel reprezintă un LED individual. Celulele pot fi apăsate pentru a selecta o culoare și a seta o valoare de timp pentru LED-ul corespunzător. Valorile de culoare și timp selectate sunt aplicate celulelor. Culoarea este, de asemenea, utilizată pentru a seta culoarea de fundal a celulei, iar culoarea textului este ajustată pe baza raportului de contrast pentru a asigura lizibilitatea.

Secțiunea inferioară a containerului conține diverse butoane și un câmp de introducere. Intrarea selectorului de culori permite utilizatorului să selecteze o culoare pentru LED. Butonul "Aplicați" aplică valorile de culoare și timp selectate celulelor selectate. Butonul "Ștergeți" resetează celulele selectate la starea lor implicită. Butonul "Descărcați" este utilizat pentru a descărca codul generat. Butonul "Ștergeți" șterge starea selectată din lista derulabilă. Câmpul de introducere permite utilizatorului să introducă o valoare de timp în secunde pentru celulele selectate.

Codul JavaScript adaugă interactivitate interfeței. Se ocupă de evenimente precum clic pe celulele tabelului, clic pe butoane și glisarea și plasarea stărilor în lista derulabilă. Când se face clic pe o celulă de tabel, valorile de culoare și timp selectate sunt aplicate celulei. Butonul "Aplicați" este activat atunci când este selectată cel puțin o celulă. Butonul "Ștergeți" elimină starea selectată din listă și resetează grila. Butonul "Aplicați" salvează starea curentă a grilei în listă. Scopul acestui cod este de a crea o interfață care permite utilizatorilor să genereze cod pentru o matrice LED (WS2812) conectată la un modul Arduino. Interfața constă dintr-o listă derulabilă, un tabel și diverse butoane și câmpuri de introducere.

Structura HTML a paginii este împărțită în diferite secțiuni folosind elemente div și clase CSS. Stilul CSS definește aspectul și aspectul elementelor. Fundalul este setat la un gradient liniar, iar o imagine este utilizată ca imagine de fundal.

Secțiunea listă derulabilă conține o listă cu un titlu. Fiecare element din listă reprezintă o stare salvată a matricei LED. Elementele din listă pot fi selectate făcând clic pe ele, iar o stare selectată poate fi ștearsă folosind butonul "Ștergeți". Starea selectată este evidențiată vizual.

Secțiunea tabelului conține o masă 8x8 reprezentând matricea LED. Fiecare celulă de tabel reprezintă un LED individual. Celulele pot fi apăsate pentru a selecta o culoare și a seta o valoare de timp pentru LED-ul corespunzător. Valorile de culoare și timp selectate sunt aplicate celulelor. Culoarea este, de asemenea, utilizată pentru a seta culoarea de fundal a celulei, iar culoarea textului este ajustată pe baza raportului de contrast pentru a asigura lizibilitatea.

Secțiunea inferioară a containerului conține diverse butoane și un câmp de introducere. Intrarea selectorului de culori permite utilizatorului să selecteze o culoare pentru LED. Butonul "Aplicați" aplică valorile de culoare și timp selectate celulelor selectate. Butonul "Ștergeți" resetează celulele selectate la starea lor implicită. Butonul "Descărcați" este utilizat pentru a descărca codul generat. Butonul "Ștergeți" șterge starea selectată din lista derulabilă. Câmpul de introducere permite utilizatorului să introducă o valoare de timp în secunde pentru celulele selectate.

Codul JavaScript adaugă interactivitate interfeței. Se ocupă de evenimente precum clic pe celulele tabelului, clic pe butoane și glisarea și plasarea stărilor în lista derulabilă. Când se face clic pe o celulă de tabel, valorile de culoare și timp selectate sunt aplicate celulei. Butonul "Aplicați" este activat atunci când este selectată cel puțin o celulă. Butonul "Ștergeți" elimină starea selectată din listă și resetează grila. Butonul "Aplicați" salvează starea curentă a grilei în listă. Butonul "Descărcați" generează codul pe baza stărilor salvate și permite utilizatorului să îl descarce. Codul include, de asemenea, funcții pentru a restabili o stare salvată și pentru a gestiona funcționalitatea drag and drop pentru lista derulabilă.

În general, acest cod oferă o interfață ușor de utilizat pentru generarea codului pentru o matrice LED conectată la un modul Arduino. Interfața permite utilizatorilor să seteze vizual culori și valori de timp pentru LED-uri individuale și să gestioneze mai multe stări salvate. Butonul „Descărcare” generează codul pe baza stărilor salvate și permite utilizatorului să-l descarce. Codul include, de asemenea, funcții pentru a restabili o stare salvată și a gestiona funcționalitatea de glisare și plasare pentru lista derulabilă.

În general, acest cod oferă o interfață ușor de utilizat pentru generarea codului pentru o matrice LED conectată la un modul Arduino. Interfața permite utilizatorilor să seteze vizual culori și valori de timp pentru LED-uri individuale și să gestioneze mai multe stări salvate.

***Termenii cheie***: LED, Configurator, Arduino, Interfață.**MULȚUMIRI**

Aș dori să-mi exprim cea mai profundă recunoștință față de stimatul meu profesor, Ioan Stelian Lemeni, pentru îndrumarea neprețuită, sprijinul neclintit și încurajarea continuă pe parcursul dezvoltării acestui proiect. Expertiza lor, cunoștințele extinse și dedicarea pentru predare au fost esențiale în modelarea înțelegerii mele și îmbunătățirea calității acestei lucrări.

De asemenea, aș dori să exprim sincera mea apreciere stimaților membri ai Facultății de Automatică, Calculatoare și Electronica Craiova, al căror angajament față de excelența în educație și cercetare mi-a oferit un mediu academic excepțional. Resursele, facilitățile și oportunitățile oferite de facultate au fost cruciale în finalizarea cu succes a acestui proiect.

În plus, aș dori să recunosc contribuțiile colegilor mei de clasă și colegilor mei care au oferit perspective valoroase și feedback constructiv, îmbogățind calitatea generală a acestui demers.

În cele din urmă, aș dori să-mi exprim sincere mulțumiri familiei și prietenilor mei pentru sprijinul și încurajarea neclintită pe parcursul acestei călătorii. Credința lor în abilitățile mele a fost o sursă constantă de motivație.

Acest proiect nu ar fi fost posibil fără eforturile colective și sprijinul tuturor celor menționați mai sus. Contribuțiile lor au fost vitale în modelarea creșterii mele academice și personale și sunt profund recunoscător pentru asistența lor neclintită.

Simțiți-vă liber să modificați și să personalizați această confirmare în funcție de cerințele dumneavoastră specifice și de natura proiectului dumneavoastră.

**CUPRINSUL**

[1 Introducere 1](#_Toc145273696)

[1.1 Scopul 1](#_Toc145273697)

[1.2 Motivația 2](#_Toc145273698)

[2 domeniul de cunostiințe și instrumente 4](#_Toc145273699)

[2.1 Arduino Uno 4](#_Toc145273700)

[2.2 HTML 5](#_Toc145273701)

[2.3 CSS 7](#_Toc145273702)

[2.4 WS2812 7](#_Toc145273703)

[3 Design-ul si Implementare 9](#_Toc145273704)

[3.1 Componenta Hardware 9](#_Toc145273705)

[3.2 Componenta Software 9](#_Toc145273706)

[3.2.1 Interfata cu utilizatorul 10](#_Toc145273707)

[3.2.2 Generarea codului 16](#_Toc145273708)

[3.2.3 Algoritmul de generare 20](#_Toc145273709)

[3.2.4 Script-ul pentru compialare 25](#_Toc145273710)

[4 Termeni de utilizare 27](#_Toc145273711)

[4.1 Autorii 27](#_Toc145273712)

[4.2 Licența de utilizare 27](#_Toc145273713)

[5 Concluzii 28](#_Toc145273714)

[6 Bibliografie 31](#_Toc145273715)

[7 Referințe web 33](#_Toc145273716)

[C. CD / DVD 34](#_Toc145273717)

[Index 35](#_Toc145273718)

**LISTA FIGURILOR**

[Figura 1. Arduino UNO R3 5](#_Toc145270174)

[Figura 2. Exemplu de „Hello World” 6](#_Toc145270175)

[Figura 3 8x8 LED WS2812 8](#_Toc145270176)

[Figura 4 Schema Hardware 8](#_Toc145270177)

[Figura 5 Structura proiectului 9](#_Toc145270178)

[Figura 6. Interfata cu utilizatorul 10](#_Toc145270179)

[Figura 7. Containarul de States 10](#_Toc145270180)

[Figura 8. Containarul Drept 12](#_Toc145270181)

[Figura 9 Continutul unei stari 14](#_Toc145270182)

[Figura 10. Event Listener pentru fiecare patrat 14](#_Toc145270183)

[Figura 11. Aplicarea culorii si valorii numerice 15](#_Toc145270184)

[Figura 12 Dictionarul format din culoare si timp 16](#_Toc145270185)

[Figura 13 Icon-ul pentru stergere 16](file:///C:\Users\Sebi\Desktop\Proiect%20Licenta\LED-Configurator\2023_ACE_Template_Documentatie_Proiect_DiplomaBUN.docx#_Toc145270186)

[Figura 14 EventListener-ul pentru sageata in sus 17](#_Toc145270187)

[Figura 15 EventListener-ul pentru sageata in jos 18](#_Toc145270188)

[Figura 16 Resetarea patratelor din matrice 18](#_Toc145270189)

[Figura 17 Butonul de download 18](#_Toc145270190)

[Figura 18 Elementele de initializare 20](#_Toc145270191)

[Figura 19 Iterarea prin sirul de stari 20](#_Toc145270192)

[Figura 20 Extragerea culorii si Timpului 21](#_Toc145270193)

[Figura 21 Extragerea valoriilor RGB 21](#_Toc145270194)

[Figura 22 Timpul Patratelor 22](#_Toc145270195)

[Figura 23 Obiectul Blob 22](#_Toc145270196)

[Figura 24 Crearea unui link de download 23](#_Toc145270197)

[Figura 25 Revocarea adresa URL 23](#_Toc145270198)

[Figura 26 Script-ul pentru rularea executabilului 24](#_Toc145270199)

**LISTA TABELELOR**

[Tabelul 1 Caracteristici RGB 9](#_Toc145355135)

[Tabelul 2 Caracteristici electrice 9](#_Toc145355136)

# Introducere

## Scopul

Interfața de generare a codului cu matrice LED este un instrument software versatil care găsește aplicații în diverse domenii. Capacitățile sale îl fac util atât în proiecte personale, cât și profesionale, oferind o gamă largă de posibilități de exprimare creativă și implementări funcționale. Câteva domenii cheie în care acest proiect poate fi utilizat includ:

Instalații de artă: artiștii și designerii pot folosi interfața de generare a codului cu matrice LED pentru a crea afișaje de lumină uimitoare și dinamice. Utilizând modele de culoare, animații și efecte personalizate, aceștia își pot aduce la viață viziunile artistice și pot captiva publicul cu instalații captivante.

Iluminat decorativ: software-ul poate fi folosit în configurații de iluminat decorativ pentru case, birouri, evenimente și spații publice. Utilizatorii pot proiecta și programa matrice LED pentru a produce modele atrăgătoare, efecte de iluminare ambientală și scheme de culori personalizate, îmbunătățind estetica și atmosfera oricărui mediu.

Publicitate și branding: Cu capacitatea de a genera conținut vizual captivant, interfața de generare a codului cu matrice LED poate fi utilizată în campanii de publicitate și branding. Matricele LED pot fi programate pentru a afișa logo-uri, mesaje promoționale sau conținut interactiv, atrăgând eficient atenția și lăsând o impresie de durată asupra spectatorilor.

Jocuri și divertisment: Software-ul poate contribui la experiențele de joc și divertisment, permițând crearea de afișaje cu matrice LED care se sincronizează cu jocul sau îmbunătățesc mediile imersive. Animațiile dinamice, indicațiile vizuale și indicatorii de stare pot fi programate pentru a îmbogăți experiențele de joc și pentru a crea configurații interactive de divertisment.

Educație și învățare: interfața de generare a codului cu matrice LED servește ca instrument educațional pentru învățarea conceptelor de programare, electronică și design vizual. Elevii și educatorii pot explora lumea programării Arduino și a matricelor LED, experimentând diferite configurații și efecte și dobândind cunoștințe practice într-un mod practic.

IoT și automatizarea locuinței: atunci când este integrată cu platforme IoT sau sisteme de automatizare a locuinței, interfața de generare a codului cu matrice LED poate controla matricele LED ca parte a unei configurații mai mari de casă inteligentă. Utilizatorii pot programa matricele pentru a afișa notificări, actualizări meteo sau date în timp real, adăugând un element interactiv și informativ caselor lor inteligente.

Prototiparea și dezvoltarea produsului: Pentru ingineri și dezvoltatori de produse, software-ul oferă o soluție de prototipare rapidă pentru testarea și validarea funcționalităților matricei LED. Le permite să genereze rapid fragmente de cod pentru integrarea matricei LED în prototipuri, permițând dezvoltarea și evaluarea eficientă a produselor.

În general, interfața de generare a codului cu matrice LED are aplicații diverse în artă, design, decorare, publicitate, jocuri, educație, IoT și dezvoltarea de produse. Versatilitatea și ușurința sa de utilizare îl fac un instrument valoros pentru oricine dorește să utilizeze matricei LED WS2812 conectate la modulele Arduino Uno în proiectele lor, indiferent de antecedentele sau nivelul de expertiză.

## Motivația

Proiectul de interfață de generare a codului cu matrice LED provine dintr-o motivație adânc înrădăcinată și un interes puternic în domeniu, reflectând pasiunea creatorului pentru tehnologie, creativitate și inovație. Proiectul a luat naștere din dorința de a simplifica și eficientiza procesul de programare a matricelor LED conectate la modulele Arduino Uno, dând în cele din urmă utilizatorilor puterea de a-și dezlănțui imaginația și de a-și aduce ideile la viață. Acest rezumat oferă o perspectivă asupra motivațiilor din spatele proiectului și a pasiunii creatorului pentru domeniu.

Fascinația creatorului pentru electronică, programare și design vizual a stârnit inițial ideea pentru interfața de generare a codului cu matrice LED. Provocarea de a programa și controla eficient matricele LED, în special modelele WS2812, l-a determinat pe creator să dezvolte o soluție software ușor de utilizat, care să atenueze complexitățile adesea asociate cu această sarcină. Prin crearea unei interfețe intuitive, scopul a fost de a permite utilizatorilor, indiferent de expertiza lor în programare, să valorifice întregul potențial al matricelor LED în proiectele lor.

Proiectul a fost, de asemenea, condus de dorința de a stimula creativitatea și de a încuraja experimentarea. Creatorul a recunoscut potențialul imens al matricelor LED ca mediu de exprimare artistică, iluminare decorativă și instalații interactive. Prin simplificarea procesului de generare a codului, interfața deschide utilizatorilor o lume de posibilități de a crea afișaje captivante din punct de vedere vizual, animații dinamice și efecte de iluminare personalizate. Proiectul urmărește să împuternicească indivizii să-și exploreze înclinațiile artistice, fie că sunt artiști, designeri sau pasionați, și să-și deblocheze potențialul creativ folosind matrice LED.

În plus, proiectul reflectă entuziasmul creatorului pentru educație și împărtășirea cunoștințelor. Interfața de generare a codului cu matrice LED servește ca instrument educațional, permițând studenților, educatorilor și entuziaștilor să se adâncească în domeniul programării Arduino, al electronicii și al designului vizual. Prin furnizarea unei platforme practice și intuitive pentru învățare, proiectul își propune să inspire și să împuternicească următoarea generație de inovatori, promovând o înțelegere profundă a principiilor din spatele programării cu matrice LED și stârnind interesul în domeniul mai larg al tehnologiei.

Interesul creatorului în domeniul matricelor LED și al programării Arduino se extinde dincolo de proiectele personale. Proiectul este condus de o viziune mai largă de promovare a colaborării și a angajamentului comunității. Oferind software-ul ca resursă gratuită și accesibilă pe internet, creatorul își propune să promoveze schimbul de cunoștințe, să încurajeze contribuțiile și să construiască o comunitate vibrantă în jurul programării cu matrice LED. Natura deschisă a proiectului invită utilizatorii din medii diverse să-și împărtășească cunoștințele, experiențele și fragmentele de cod, îmbogățind și mai mult experiența de învățare colectivă și alimentând inovația.

Pe scurt, proiectul de interfață de generare a codului cu matrice LED a fost motivat de o pasiune pentru tehnologie, creativitate și inovație. Interesul creatorului pentru electronică, programare și design vizual, combinat cu impulsul de a simplifica programarea matricei LED și de a stimula creativitatea, a culminat cu dezvoltarea unui instrument software intuitiv. Proiectul reflectă, de asemenea, un angajament față de educație, partajarea cunoștințelor și construirea comunității, cu scopul de a împuternici utilizatorii să-și exploreze aspirațiile artistice și tehnologice. În cele din urmă, proiectul de interfață de generare a codului cu matrice LED reprezintă o dovadă a pasiunii creatorului pentru domeniu și a dăruirii lor de a-i împuternici pe alții să-și realizeze viziunile creative.

# domeniul de cunostiințe și instrumente

Pentru realizarea acestui proiect, au fost studiate diferite concepte de Web design, ce includ limbaje de mark-up, dar si limbaje de programare cum ar fi JavaScript, dar si cunoștințe in domeniul microcontroalerelor si electronicii pentru realizarea si montarea componentelor electronice. Pe partea de micro s-a folosit un modul de Arduino UNO si o matrice de LED-uri 8×8(WS2812), iar ca limbaj de programare s-a folosit C, renumit pentru programarea eficienta a plăcuțelor de micro, împreună cu librăriile adiționale pentru programarea matricei de LED-uri.

## Arduino Uno

“Arduino Uno este o placă de microcontroler open-source bazată pe microcontrolerul Microchip ATmega328P (MCU) și dezvoltată de Arduino.cc și lansată inițial în 2010. Placa este echipată cu seturi de pini de intrare/ieșire (I/O) digitali și analogici care pot fi interfațați cu diferite plăci de expansiune (ecrane) și alte circuite. Placa are 14 pini I/O digitali (șase capabili de ieșire PWM), 6 pini I/O analogici și este programabilă cu Arduino IDE (Integrated Development Environment), printr-un cablu USB de tip B. Poate fi alimentat de un cablu USB sau de un conector butoi care acceptă tensiuni între 7 și 20 de volți, cum ar fi o baterie dreptunghiulară de 9 volți. Are același microcontroler ca placa Arduino Nano și aceiași anteturi ca placa Leonardo. Designul de referință hardware este distribuit sub o licență Creative Commons Attribution Share-Alike 2.5 și este disponibil pe site-ul Arduino. Fișierele de aspect și de producție pentru unele versiuni ale hardware-ului sunt, de asemenea, disponibile.Arduino Uno este o placă de microcontroler open-source bazată pe microcontrolerul Microchip ATmega328P (MCU) și dezvoltată de Arduino.cc și lansată inițial în 2010. Placa este echipată cu seturi de pini de intrare/ieșire (I/O) digitali și analogici care pot fi interfațați cu diferite plăci de expansiune (ecrane) și alte circuite. Placa are 14 pini I/O digitali (șase capabili de ieșire PWM), 6 pini I/O analogici și este programabilă cu Arduino IDE (Integrated Development Environment), printr-un cablu USB de tip B. Poate fi alimentat de un cablu USB sau de un conector butoi care acceptă tensiuni între 7 și 20 de volți, cum ar fi o baterie dreptunghiulară de 9 volți. Are același microcontroler ca placa Arduino Nano și aceiași anteturi ca placa Leonardo. Designul de referință hardware este distribuit sub o licență Creative Commons Attribution Share-Alike 2.5 și este disponibil pe site-ul Arduino. Fișierele de aspect și de producție pentru unele versiuni ale hardware-ului sunt, de asemenea, disponibile.”

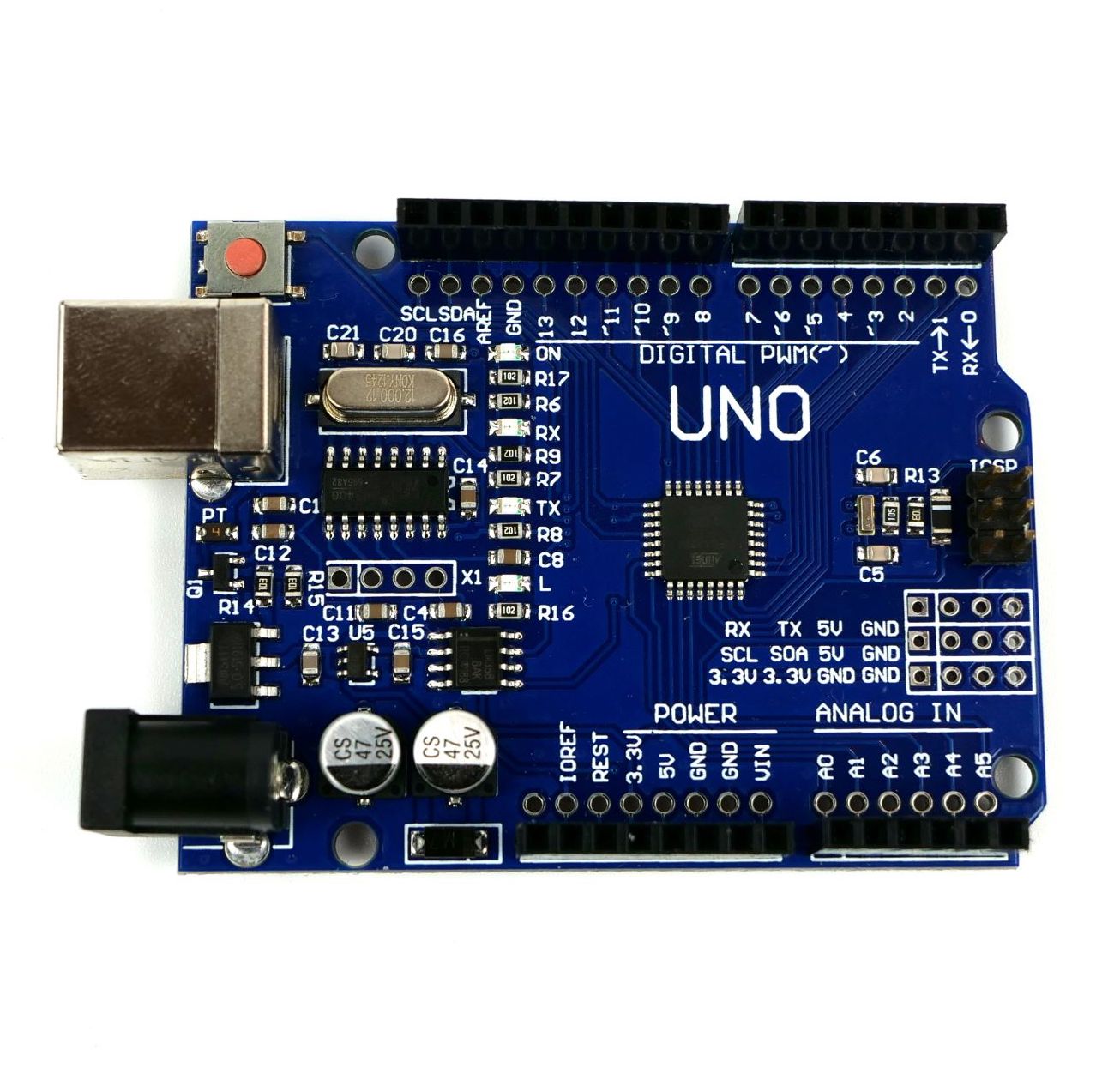


Figura 1. Arduino UNO R3

## HTML

„HyperText Markup Language sau HTML este limbajul standard de [marcare](https://en.wikipedia.org/wiki/Markup_language) pentru documentele concepute pentru a fi afișate într-un [browser web](https://en.wikipedia.org/wiki/Web_browser). Este adesea asistat de tehnologii precum [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) și [limbaje de scripting](https://en.wikipedia.org/wiki/Scripting_language), cum ar fi [JavaScript](https://en.wikipedia.org/wiki/JavaScript).

[Browserele web](https://en.wikipedia.org/wiki/Web_browser) primesc documente HTML de la un [server](https://en.wikipedia.org/wiki/Web_server) web sau de la stocarea locală și [redau](https://en.wikipedia.org/wiki/Browser_engine) documentele în pagini web multimedia. HTML descrie [structura unei pagini web semantic](https://en.wikipedia.org/wiki/Web_page) și a inclus inițial indicii pentru aspectul acesteia.

[Elementele HTML](https://en.wikipedia.org/wiki/HTML_element) sunt blocurile de construcție ale paginilor HTML. Cu construcții HTML, [imaginile](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) și alte obiecte, cum ar fi [formularele interactive](https://en.wikipedia.org/wiki/Fieldset), pot fi încorporate în pagina redată. HTML oferă un mijloc de a crea [documente structurate](https://en.wikipedia.org/wiki/Structured_document) prin denotarea [semanticii](https://en.wikipedia.org/wiki/Semantics) structurale pentru text, cum ar fi titluri, paragrafe, liste, [linkuri](https://en.wikipedia.org/wiki/Hyperlink), citate și alte elemente. Elementele HTML sunt delimitate de etichete, scrise folosind paranteze unghiulare. Etichete precum <img/> și <input/> introduc direct conținut în pagină. Alte taguri, cum ar fi <p> și </p> înconjoară și furnizează informații despre textul documentului și pot include taguri de sub elemente. Browserele nu afișează etichetele HTML, ci le utilizează pentru a interpreta conținutul paginii.

HTML poate încorpora programe scrise într-un limbaj de scripting, cum ar fi JavaScript, care afectează comportamentul și conținutul paginilor web. Includerea CSS definește aspectul și aspectul conținutului. Consorțiul World Wide Web (W3C), fost întreținător al HTML și actual întreținător al standardelor CSS, a încurajat utilizarea CSS în locul HTML explicit de prezentare din 1997. O formă de HTML, cunoscută sub numele de HTML5, este utilizată pentru a afișa conținut video și audio, utilizând în principal elementul <pânză>, împreună cu JavaScript.

Marcajul HTML constă din mai multe componente cheie, inclusiv cele numite etichete (și atributele acestora), tipuri de date bazate pe caractere, referințe de caractere și referințe de entitate. Etichetele HTML vin cel mai frecvent în perechi precum <h1> și < / h1>, deși unele reprezintă elemente goale și astfel sunt neasociate, de exemplu <img>. Prima etichetă dintr-o astfel de pereche este eticheta de început, iar a doua este eticheta de sfârșit (se mai numesc etichete de deschidere și etichete de închidere).

O altă componentă importantă este [declarația de tip document](https://en.wikipedia.org/wiki/Document_type_declaration) HTML, care declanșează redarea [modului standard](https://en.wikipedia.org/wiki/Standards_mode).

Următorul este un exemplu al [programului clasic "Hello, World!](https://en.wikipedia.org/wiki/%22Hello,_World!%22_program)":

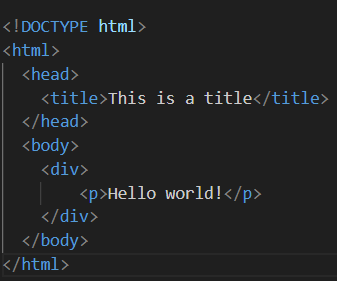
****

Figura 2. Exemplu de „Hello World”

Textul dintre <html> și </html> descrie pagina web, iar textul dintre <body> și </body> este conținutul vizibil al paginii. Textul de marcare <title>Acesta este un titlu</titlu> definește titlul paginii browserului afișat în [filele browserului](https://en.wikipedia.org/wiki/Browser_tab) și titlurile [ferestrelor](https://en.wikipedia.org/wiki/Window_(computing)), iar eticheta <div> definește o diviziune a paginii utilizată pentru o stilizare ușoară. Între <head> și </head>, un element <meta> poate fi utilizat pentru a defini metadatele paginii web.

Declarația de tip document <! DOCTYPE html> este pentru HTML5. Dacă o declarație nu este inclusă, diferite browsere vor reveni la "[modul quirks](https://en.wikipedia.org/wiki/Quirks_mode)" pentru redare.”

## CSS

„Foi de stil în cascadă (CSS) este un limbaj de foi de stil utilizat pentru descrierea prezentării unui document scris într-un limbaj de marcare, cum ar fi HTML sau XML (inclusiv dialecte XML precum SVG, MathML sau XHTML). CSS este o tehnologie de bază a World Wide Web, alături de HTML și JavaScript.

CSS este conceput pentru a permite separarea conținutului și a prezentării, inclusiv aspectul, culorile și fonturile. Această separare poate îmbunătăți accesibilitatea conținutului; să ofere mai multă flexibilitate și control în specificarea caracteristicilor de prezentare; permite mai multor pagini web să partajeze formatarea prin specificarea CSS relevant într-un fișier .css separat, ceea ce reduce complexitatea și repetarea conținutului structural; și activați memorarea în cache a fișierului .css pentru a îmbunătăți viteza de încărcare a paginii între paginile care partajează fișierul și formatarea acestuia.

Separarea formatării și a conținutului face, de asemenea, posibilă prezentarea aceleiași pagini de marcare în stiluri diferite pentru diferite metode de redare, cum ar fi pe ecran, în imprimare, prin voce (prin browser sau cititor de ecran bazat pe vorbire) și pe dispozitive tactile bazate pe Braille. CSS are, de asemenea, reguli pentru formatarea alternativă în cazul în care conținutul este accesat de pe un dispozitiv mobil.

Cascada de nume provine din schema de prioritate specificată pentru a determina ce regulă de stil se aplică dacă mai multe reguli se potrivesc cu un anumit element. Această schemă de priorități în cascadă este previzibilă.

Specificațiile CSS sunt menținute de World Wide Web Consortium (W3C). Internet media type (MIME type) text/css este înregistrat pentru utilizare cu CSS de RFC 2318 (martie 1998). W3C operează un serviciu gratuit de validare CSS pentru documentele CSS.

În plus față de HTML, alte limbaje de marcare acceptă utilizarea CSS, inclusiv XHTML, XML simplu, SVG și XUL. CSS este, de asemenea, utilizat în setul de instrumente widget GTK. "

## WS2812

„WS2812 este o sursă inteligentă de lumină LED de control pe care circuitul de control și cipul RGB sunt integrate într-un pachet de componente 5050. Acesta include un dispozitiv digital inteligent de blocare a datelor și un circuit de acționare de amplificare care remodelează semnalul. Includeți, de asemenea, un oscilator intern de precizie și o parte de control al curentului constant programabilă de tensiune de 12V, asigurând în mod eficient înălțimea culorii luminii punctului pixel consistentă.

Protocolul de transfer de date utilizează un singur mod de comunicare NZR. După resetarea pixel power-on, portul DIN primește date de la controler, primul pixel colectează datele inițiale pe 24 de biți apoi le trimite la zăvorul intern de date, celelalte date care se remodelează prin circuitul intern de amplificare a remodelării semnalului trimis la următorul pixel cascadă prin portul DO. După transmisia pentru fiecare pixel, semnalul pentru a reduce 24bit. Pixelul adoptă tehnologia de transmisie cu remodelare automată, ceea ce face ca numărul cascadei pixelilor să nu limiteze transmisia semnalului, ci doar să dependente de viteza de transmisie a semnalului.”



Figura 3 8x8 LED WS2812

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Emiting color | Model | Wavelength(nm) | Luminous intensity(mcd | Voltage(V) |
| Red | 13CBAUP | 620-625 | 390-420 | 2.0-2.2 |
| Green | 13CGAUP | 522-525 | 660-720 | 3.0-3.4 |
| Blue | 10R1MUX | 465-467 | 180-200 | 3.0-3.4 |

Tabelul 1 Caracteristici RGB

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | Symbol | Conditions | Min | Tpy | Max | Unit |
| Low voltage output current | IOL | ROUT |  | 18.5 |  | mA |
| Idout | VO=0.4V, DOUT | 10 |  |  | mA |
| Input current | II | VI=VDD/VSS |  |  | ±1 | mA |
| Input voltage level | VIH | DIN，SET | 0.7VDD |  |  | mA |
| VIL | DIN，SET |  |  | 0.3 VDD | mA |
| Hysteresis voltage | VH | DIN，SET |  | 0.35 |  | mA |

Tabelul 2 Caracteristici electrice

# Design-ul si Implementare

## Componenta Hardware

Pentru acest proiect s-a folosit un Arduino UNO conectat la o matrice de LED-uri(WS2812 8x8), conectata prin 3 fire principale:

* Pinul de intrare care va recepționa ce LED-uri se vor aprinde.
* Pinul de alimentare care alimentează LED-urile de pe placa.
* Pinul de GND, unde placa se conectează la plăcuța Arduino.

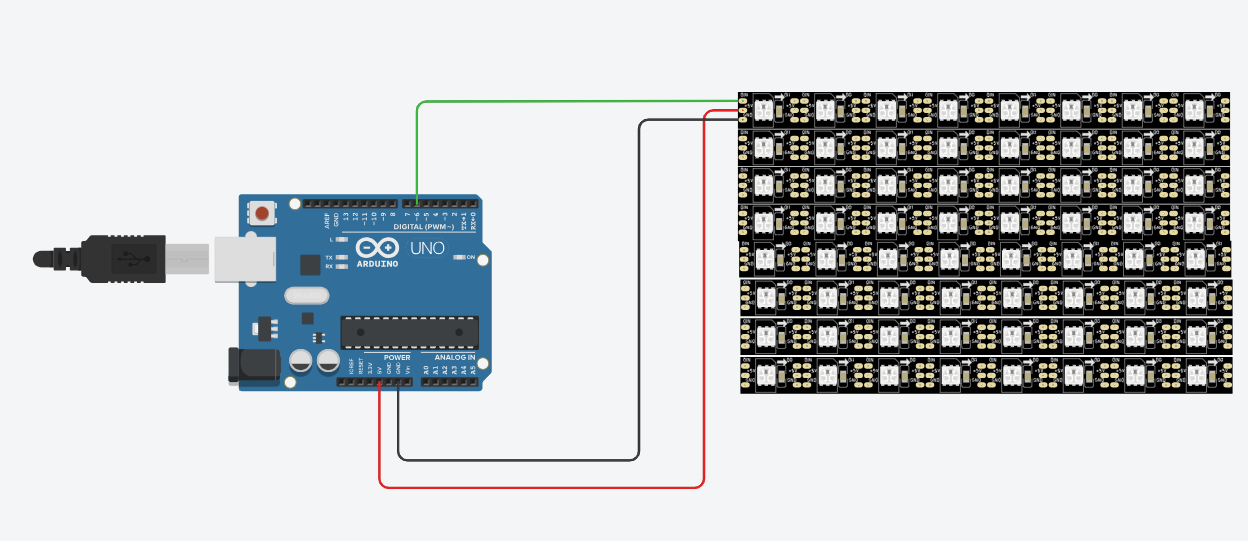


Figura 4 Schema Hardware

## Componenta Software

Pentru o interacțiune mai ușoară si mai prietenoasa cu utilizatorul, componenta software include un site HTML, alături de CSS and JavaScript, pentru crearea unui configurator pentru generarea codului C pentru plăcuța Arduino.

Proiectul va fi împărțit in 2 foldere, dintre care in unul se vor stoca imagini adiționale, iar in celălalt se vor stoca fișierele sursa. In folderul Pictures vor fi eventualele poze, iar in folderul Components vom avea 3 fișiere: fișierul HTML unde va fi codul pentru interfața, fișierul CSS unde va fi stocat codul pentru elementele HTML si fișierul Javascript, unde este stocat codul pentru interacțiunea cu utilizatorul si generarea fișierului .ino.

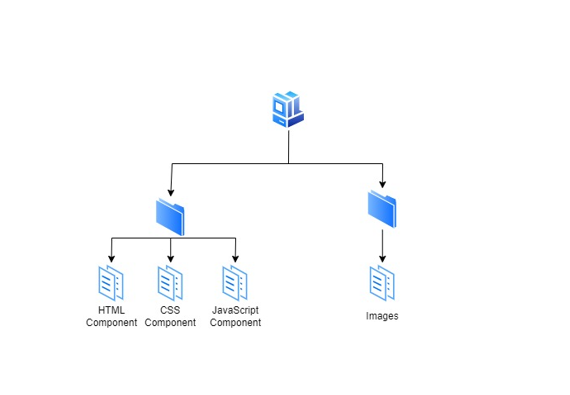


Figura 5 Structura proiectului

### Interfața cu utilizatorul

interfața cu utilizatorul a fost construita prin intermediul unui site, ușor accesibil pentru toate platformele, atât utilizatori de PC, cat si Android.

Principalele componente ale site-ului sunt cele 2 containere care conțin lista stărilor, stările memorează informațiile despre matrice de LED-uri, dar si o matrice de 8×8, simulând o copie a matricei.

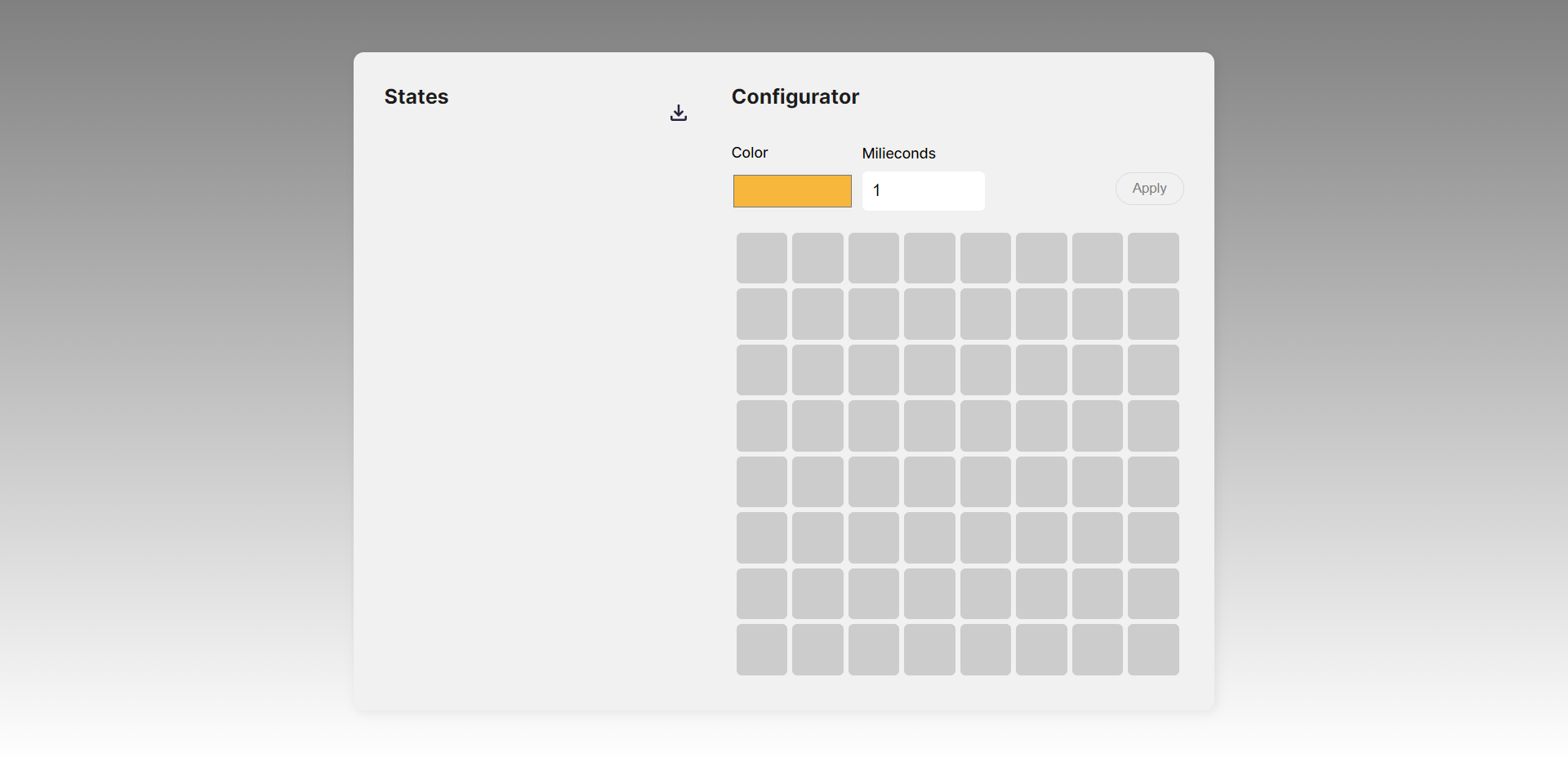


Figura 6. Interfata cu utilizatorul

Primul container, numit si containerul „States”, stochează stările create cu ajutorul configuratorului, introducând titlul, o pictogramă SVG și eventual elemente suplimentare pentru gestionarea stărilor de animație. Utilizatorii se pot aștepta să interacționeze cu această secțiune pentru a selecta stările pentru modificare, pentru a-și schimba ordinea și, eventual, pentru a le șterge. Pictograma SVG indică faptul că există acțiuni asociate cu această secțiune, îmbunătățind interactivitatea și implicarea utilizatorului. În general, această secțiune contribuie la natura prietenoasă a proiectului, permițând utilizatorilor să gestioneze și să personalizeze eficient animațiile cu matricea LED.

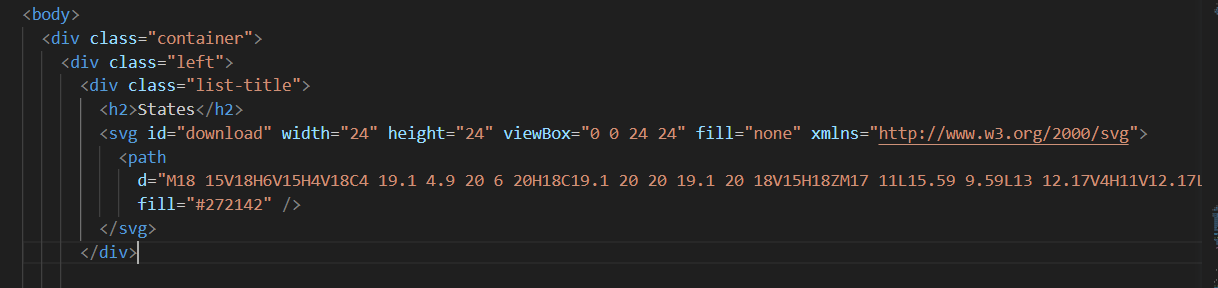


Figura 7. Containerul de States

* <div class="container">: Acest <div> servește ca un container care încapsulează întreaga secțiune legată de gestionarea statului. Acesta conține probabil alte componente și elemente care gestionează colectiv statele.
* <div class="left">: în interiorul containerului, există un alt <div> cu clasa rămasă. Acesta ar putea fi folosit pentru aranjarea și poziționarea elementelor, ajutând la structurarea estetică a interfeței.
* <div class="list-title">: în div-ul din stânga, acest <div> reprezintă secțiunea de titlu pentru lista de state. Are titlul „State” și o pictogramă SVG care indică probabil capacitatea de a descărca sau de a efectua acțiuni legate de lista de state.
* <h2>State</h2>: acest element <h2> afișează titlul „State”, indicând utilizatorului că următoarea secțiune se ocupă de gestionarea diferitelor stări de animație.
* <svg>: Acesta este un element SVG, care include un element cale pentru a crea o pictogramă. Pictograma SVG aici apare ca o săgeată îndreptată în jos, indicând acțiunea de descărcare. Atributul id „descărcare” ar putea fi folosit pentru gestionarea evenimentelor JavaScript.
* <path>: Acest element definește calea pictogramei SVG. Folosește o combinație de comenzi și coordonate pentru a crea forma săgeții.

Al doilea container este responsabil pentru configurarea matricei, utilizatorul poate selecta o culoare și o valoare de timp, poate aplica aceste setări celulelor LED individuale făcând clic pe ele și pot stoca aceste configurații ca stări. Îmbunătățește interactivitatea utilizatorului, permițând configurarea dinamică și vizualizarea animațiilor cu matrice LED. În plus, codul permite utilizatorilor să șteargă configurația unui pătrat, ștergând ora introdusă și făcând clic pe pătrat.

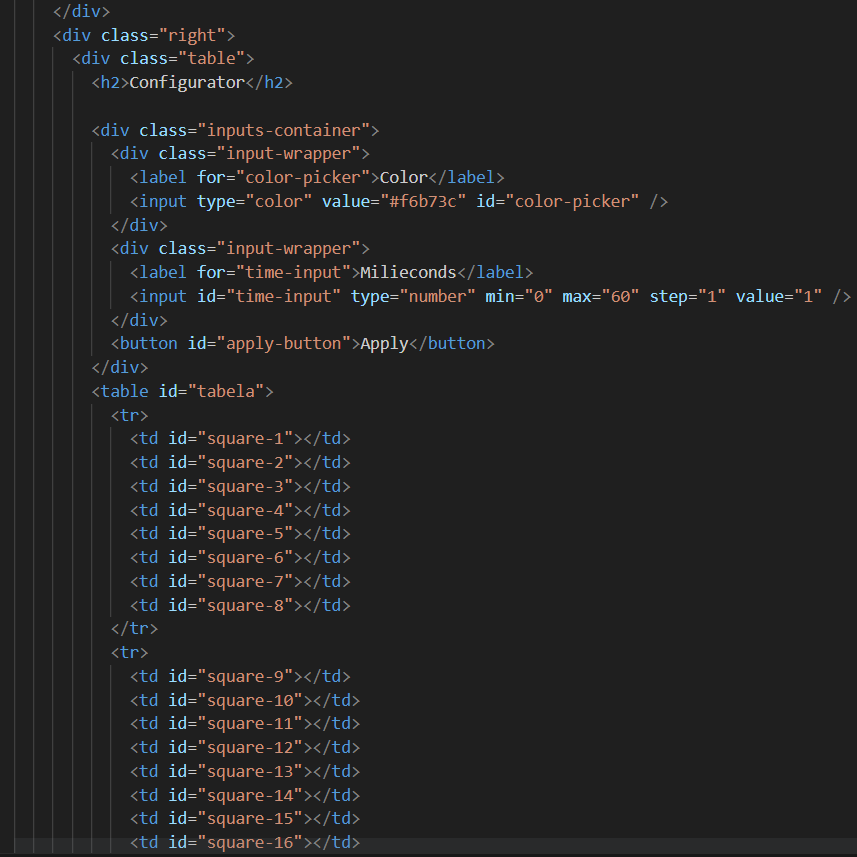


Figura 8. Containerul Drept

* <div class="right"> și <div class="table">: Aceste elemente <div> definesc containerul pentru secțiunea de configurare a matricei LED, creând o zonă structurată în care utilizatorii pot interacționa cu intrările de configurare și pot vizualiza matricea LED .
* <h2>Configurator</h2>: Acest element <h2> servește ca titlu pentru secțiunea de configurare, oferind un titlu clar care indică scopul acestei părți a interfeței.
* <div class="inputs-container">: Acest <div> grupează elementele de intrare și butonul „Aplicare” pentru configurarea matricei LED.
* <div class="input-wrapper"> împachetează intrarea de culoare.
* <label for="color-picker">Culoare</label> oferă o etichetă pentru introducerea selectorului de culori.
* <input type="color" value="#f6b73c" id="color-picker" /> creează o intrare de selectare a culorilor, inițializată cu culoarea implicită „#f6b73c”.
* <label for="time-input">Milisecunde</label> etichetează timpul de intrare.
* <input id="time-input" type="number" min="0" max="60" step="1" value="1" /> generează un număr de intrare pentru timp în milisecunde, cu un interval de 0 la 60 și un pas de 1. Valoarea inițială este setată la 1.
* <button id="apply-button">Aplicare</button>: Acest buton declanșează aplicarea setărilor de culoare și timp configurate la celula LED selectată atunci când este făcută clic.

Reprezentarea matricei LED (<table id="tabela">):

Acest tabel reprezintă vizual matricea LED cu 8 rânduri și 8 coloane, însumând 64 de celule LED.

Fiecare celulă LED este reprezentată ca un element <td> (date de tabel) cu un ID unic, cum ar fi pătrat-1, pătrat-2 și așa mai departe.

De asemenea pentru o trasabilitate mult mai buna a fiecărei stări, butonului de „Apply”, i s-a atribuit funcția de creare a unui <ul>, care va fi adăugat in lista stărilor. Pentru acest lucru, butonul de „Apply” va apela o funcție numita „saveState()”, care va genera pe baza input-urilor de la user starea respectiva. Funcția va creea un nou <ul> si i se va atribui un IdCounter, acesta fiind un ID unic pentru fiecare stare si un index. Indexul va fi cel afișat in lista stărilor pentru o trasabilitate mai buna si prelucrare eventuală. De asemenea pentru o interacțiune mai ușoară i se va mai atașa si alte elemente ajutatoare. Elementele vor fi in interiorul unui <div> pentru o așezare mai ordonata in pagina. In interiorul div-ului vor fi un <svg>, care va conține poza unui cos de gunoi, acesta realizând ștergerea elementului, o <img> cu o săgeata in sus pentru deplasarea in sus a stării in cadrul listei si încă o <img> cu o săgeata in jos pentru deplasarea in jos a stării in cadrul listei. La finalul acestei funcții elementul va fi adăugat într-o lista, cu ajutorul acestei liste vom vedea cum se poate deplasa in sus sau in jos starea respectiva.

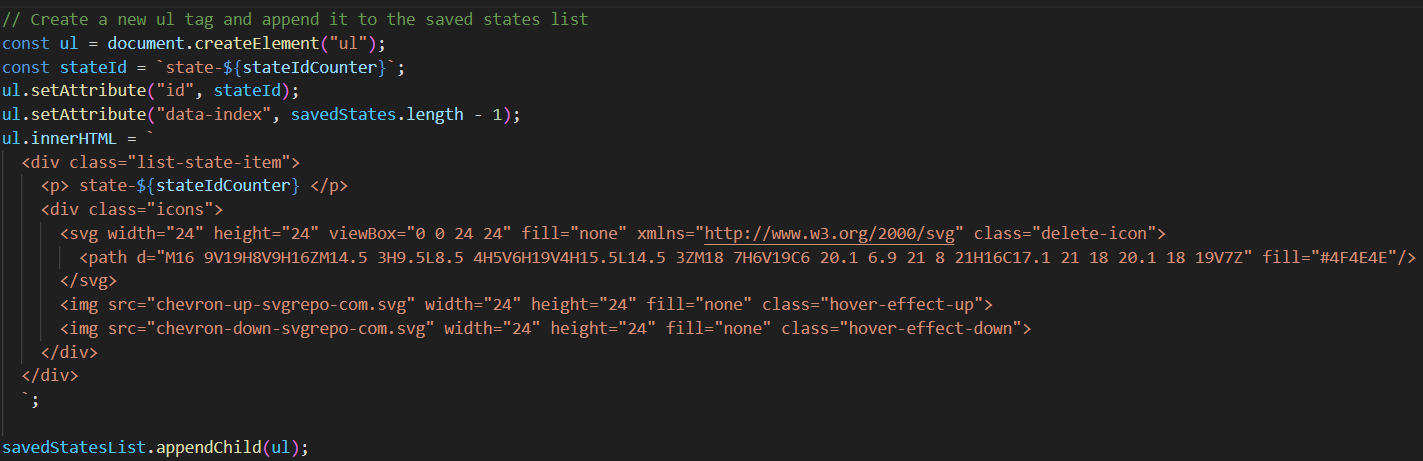


Figura 9 Conținutul unei stări

Fiecare pătrat in matrice are atașată un „event listener”, prin intermediul căruia la apăsarea printr-un click pe pătrat, acesta sa-si schimbe culoarea eventual in interiorul acestuia se va scrie valoarea in milisecunde. Inițial se va verifica daca un pătrat are sau nu are o culoare, deoarece butonul de „Apply” nu va funcționa daca nu este nici măcar o coloare in matrice.

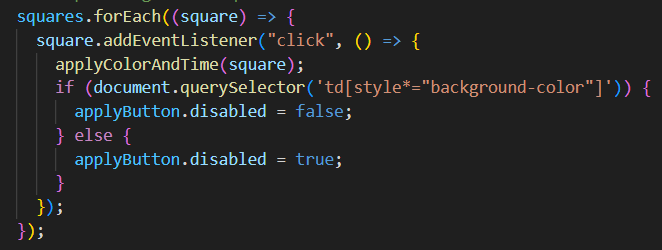


Figura 10. Event Listener pentru fiecare pătrat

De remarcat este culoarea pătratelor si culoarea valorii numerice din interiorul acesteia. Pentru evitarea confuziei, culoarea valorii alfanumerice va fi întotdeauna in contrast cu colarea selectata de utilizator.

Formula după care s-a calculat contrastul este:

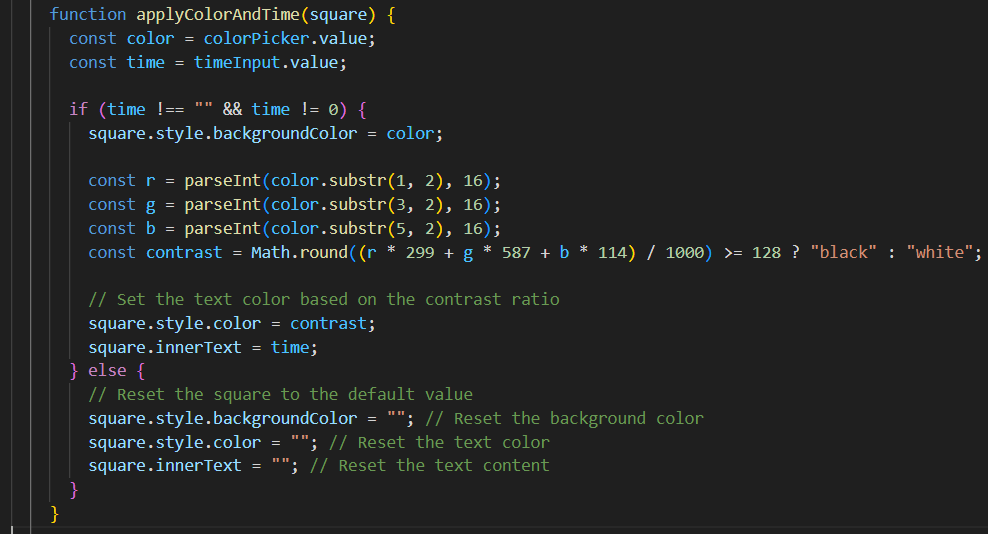


Figura 11. Aplicarea culorii si valorii numerice

### Generarea codului

Am văzut anterior partea grafica cu care utilizatorul interacționează si gestionează animațiile. De asemenea pentru aprinderea LED-urilor după dorința utilizatorului, programul trebuie sa genereze un cod C, ca acesta sa poată sa aprindă luminițele.

Pentru acest lucru s-a creat un algoritm pe baza input-urilor de la user, de menționat ca aprinderea se face într-o maniera lineara, unde fiecare stare trebuie sa-si termine absolut toate LED-urile de aprins după care va urma cealaltă.

Pentru ca stările matricei sa fie stocate in interiorul unui sir unidimensional, iar pe fiecare sub-stare este stocata sub forma unui dicționar care memorează in general 2 variabile: culoarea si timpul.

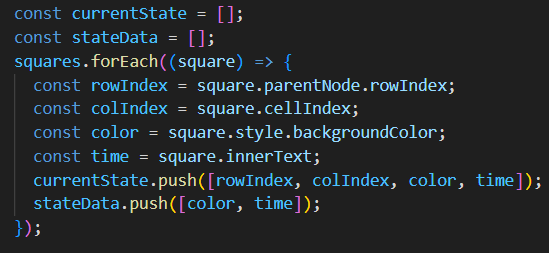


Figura 12 Dicționarul format din culoare si timp

După ce sunt stocate într-un sir unidimensional fiecare stări i se atribuie un <ul>, la acesta adăugându- se diferite atribute pentru o manipulare mai ușoară atât pentru interfața, cat si pentru prelucrarea acesteia mai târziu in cadrul programului.

Se creează un o constanta de tipul <ul> cu un ID unic, care se incrementează de fiecare data la crearea unuia nou, data-index cu ajutorul căruia putem șterge, muta elementul mai sus sau mai jos in cadrul listei, un paragraf, care afișează numărul acesteia, iar mai apoi un <div> care cuprinde următoarele imagini:

* O iconița de ștergere, care șterge respectivul <ul> din cadrul listei. Pentru ca sa evitam propagarea comenzii, s-a adăugat linia e.stopPropagation(), deoarece fără aceasta linie de cod s-ar fi afișat in matrice starea respectiva, ceea ce este incorect. In momentul in care starea a fost ștearsă din sir, se va apela funcția resetGrid(), pentru resetarea matricei in faza default.

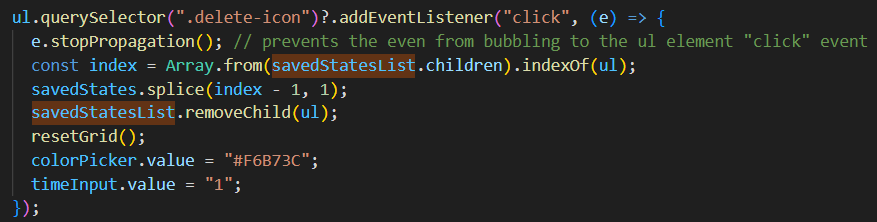


Figura 13 Icon-ul pentru stergere

* O iconiță care muta in sus in lista stărilor <ul> respectiv. De asemenea s-a folosit si in acest caz funcția de e.stopPropagation(), deoarece fără aceasta linie de cod s-ar fi afișat in matrice starea respectiva, ceea ce este incorect. Ca acest lucru sa fie realizabil, avem nevoie de de lista de elementelor <ul>, dar si de containerul de deasupra lui. Pentru acest lucru se verifica daca in lista exista un „frate” precedent, iar daca exista se inserează deasupra acestuia cu ajutorul funcții insertBefore() pentru containerul care are ca nume de clasa „list”.

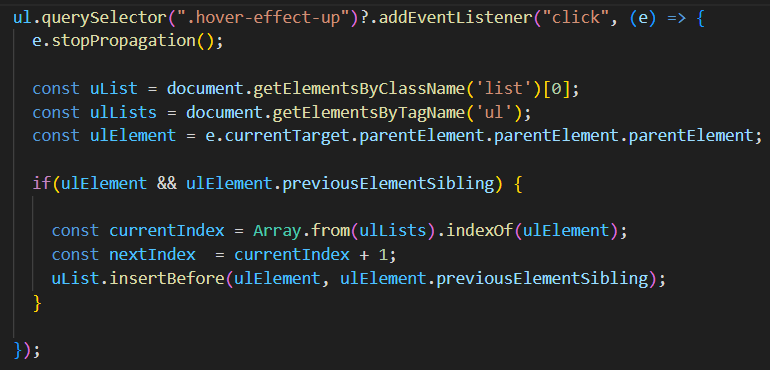


Figura 14 EventListener-ul pentru săgeata in sus

* O iconiță care muta in jos in lista stărilor <ul> respectiv. De asemenea s-a folosit si in acest caz funcția de e.stopPropagation(), deoarece fără aceasta linie de cod s-ar fi afișat in matrice starea respectiva, ceea ce este incorect. Ca acest lucru sa fie realizabil, avem nevoie de de lista de elementelor <ul>, dar si de containerul de deasupra lui. Pentru acest lucru se verifica daca in lista exista un „frate” următor, iar daca exista se inserează după acesta cu ajutorul funcții insertBefore() pentru containerul care are ca nume de clasa „list”.

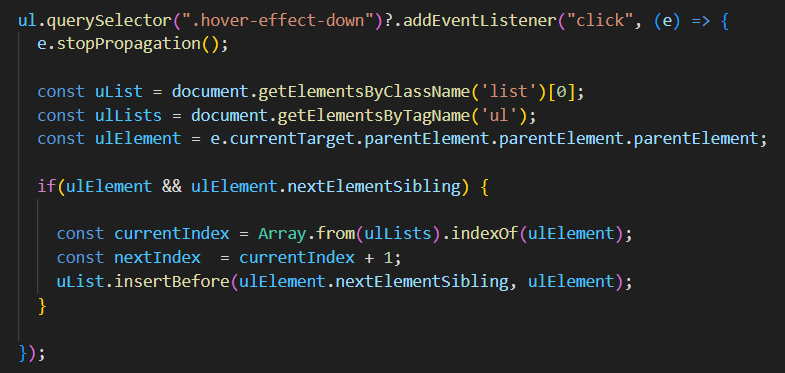


Figura 15 EventListener-ul pentru săgeata in jos

De precizat ca după fiecare ștergere a unei stării trebuie sa se facă o resetare de matrice a stărilor, iar pentru acest lucru se apelează un resetGrid(), care resetează fiecare pătrat din matrice.

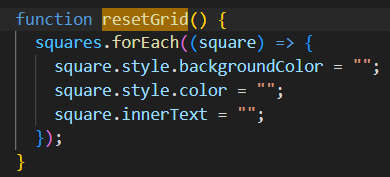


Figura 16 Resetarea pătratelor din matrice

Pentru generarea codului propriu zis s-a definit un buton de download, care la apăsarea acestuia instalează un fișier .ino pe calculatorul utilizatorului. In interiorul fișierului se afla codul C care va executa animațiile/comenzile introduse anterior de utilizator putând fii rulate pe plăcuța de Arduino.

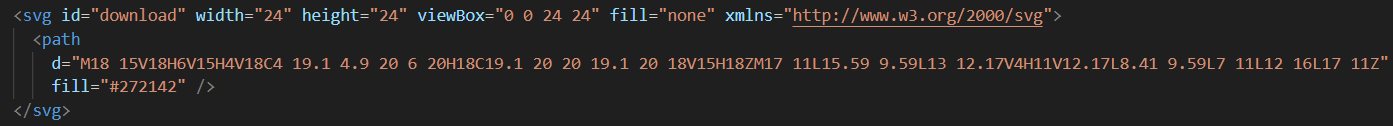


Figura 17 Butonul de download

Un EventListener este atașat butonului de download, in momentul in care utilizatorul apasă pe acesta codul va fi generat in interiorul unui sir de caractere. Inițial șirul va conține informații de baza cum ar fi librăria folosită pentru acest lucru, numărul total de LED-uri din matrice, șirul care conține toate LED-urile sau informații despre pinii folosiți, de precizat ca s-a folosit pinul numărul 6 pentru acest lucru, numai acesta poate sa fie folosit momentan. In funcția void setup() sunt puse alte informații adiționale pentru început, de menționat ca aceasta funcție(void setup()) este apelata când un program începe, ea fiind folosita pentru inițializarea de variabile, pini, librarii. Funcția setup() va rula o singura data, după fiecare reset sau pornire a plăcii.

### Algoritmul de generare

Pentru generarea codului C s-a dezvoltat un algoritm care prelucrează stările date ca input de către user si le transforma in cod C. Pentru acest lucru s-a mai adăugat in plus o librărie care folosește ca structura de date o coada de prioritate, deoarece vom avea nevoie sa le prelucram in ordine crescătoare. La începutul funcții a fost definit un comparator care ne va ajuta sa ordonam elementele din interiorul cozii de prioritate in ordine crescătoare. In prima etapa s-a definit un string care stochează lucrurile de baza pentru inițializare, cum ar fi:

* Librăriile folosite: #include<FastLED.h>
* Definirea pinului de ieșire: #define LED\_PIN 6
* O constanta care definește numărul total de LED-uri(folosirea acesteia a fost opțională): #define NUM\_LEDS 64
* Un sir unidimensional care corespunde fiecărui LED din matrice, numai in felul acesta se poate acționa fiecare led independent: CRGB leds[NUM\_LEDS]
* Funcția pentru dezactivarea tuturor LED-urilor, acesta trebuie apelata întotdeauna la început ca sa ne asiguram ca nu este nicio culoare pornita: void clearMatrix(). In interiorul acestei funcții se iterează prin toate LED-urile si sunt inițializate cu valoarea „CRGB::Black”, aceasta valoare fiind folosita pentru dezactivarea unui LED.
* Funcția de inițializare, care se apelează întotdeauna o singura data când placa de Arduino este pornita. In interiorul acesteia este definita funcția pentru definirea matricii „FastLED.addLeds<WS2812, LED\_PIN, GRB>(leds, NUM\_LEDS)”, care ia ca argumente modelul matricii, pinul de ieșire, formatul color, matricea de LED-uri menționată anterior si numărul total.
* Luminozitatea generala, de preferat a fost valoarea de 50, deoarece la 100 culorile începeau sa devina foarte greu de distins intre ele.
* Funcția de inițializare este apelata aici pentru setarea LED-urilor.
* Funcția FastLED.show() este folosita pentru actualizarea LED-urilor. In momentul in care este apelata tot ce s-a setat pentru matrice este afișat.
* Într-un final s-a adăugat funcția de void loop(), unde plăcuța Arduino v-a intra aici de fiecare data si v-a efectua codul din interiorul acesteia.

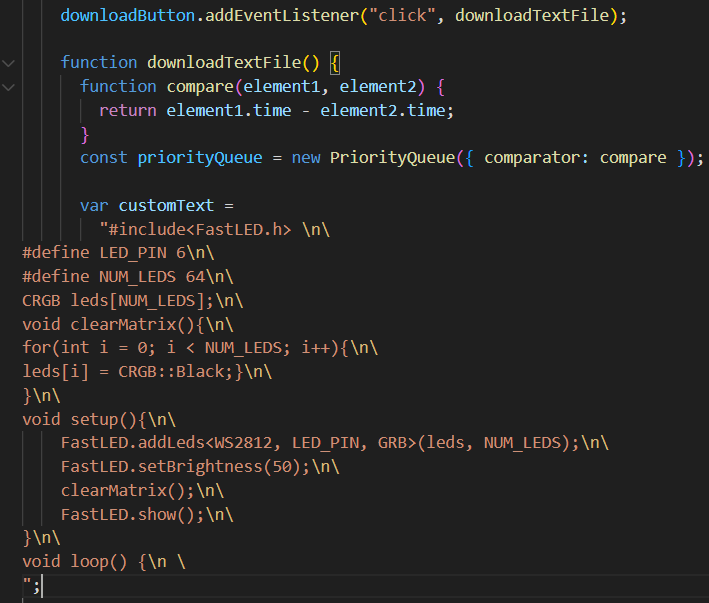


Figura 18 Elementele de inițializare

Pentru generarea propriu zisa se va folosi șirul de stării menționat in capitolele anterioare care stochează toate stările care au fost date de către user.

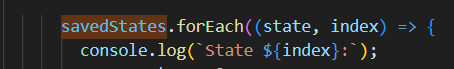


Figura 19 Iterarea prin șirul de stări

Din fiecare element din savedState, se extrage o stare, in interiorul acestui element se stochează 64 de elemente fiecare corespunzând unui pătrat din matrice. In interiorul acesteia se afla un tuple de date: culoare pătratului, care la rândul acesteia este formata din 3 elemente si timpul de afișare. Fiecare stocând o valoare intre 0 si 255(1 byte) aceste 3valori corespund unei culori(Roșu, Verde si Albastru).

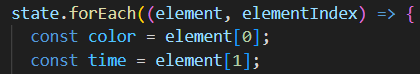


Figura 20 Extragerea culorii si Timpului

Având in vedere ca valorile culorilor au fost stocate in formatul RGB, vom avea nevoie de a le extrage din variabila color, ele au fost stocate sub forma unui sir de caractere, pentru acest lucru s-a folosit expresiile regulate(RegEx). Expresiile regulate sunt deseori folosite pentru extragerea unei secvențe de caractere după un anumit criteriu. In cazul nostru s-a folosit “\d+/g”. Unde “\d” semnifica un caracter numeric, “+” este folosit pentru unul sau mai multe apariții, iar “/g” este folosit pentru căutare globala, adică nu se oprește la prima apariție.

In momentul in care valorile au fost extrase se verifica daca șirul este valid, iar daca este valid cele 3 culori sunt extrase din text si sunt puse într-o coada de prioritate, așa cum s-a menționat mai sus, împreuna cu timpul, dar si cu un counter pentru a memora numărul pătratului.

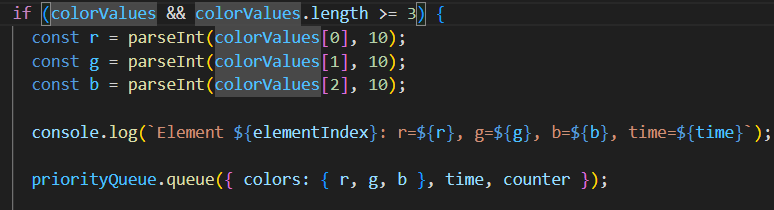


Figura 21 Extragerea valorilor RGB

După extragerea culorilor ele vor fi stocate in textul custom. In textul custom se va adăuga la fiecare stare pătratele care au fost inițializate cu culoarea respective, avantajul este ca putem sa inițializăm acum pătratele si după sa le afișam când dorim.

Pentru afișarea lor in ordinea dorita se vor afișa absolut toate, doar timpul de afișare va fi diferit. Prima oara se va pune un delay() cu timpul cel mai mic, după ce timpul acestuia s-a scurs se va opri cu ajutorul unui CRGB::Black. Pentru următorul pătrat nu se va mai pune un delay() cu timpul lui, ci dimpotrivă se va pune diferența dintre timpul acestuia si timpul scurs de la pătratul anterior. In felul in afara de primul pătrat, la restul se vor pune diferențele dintre cel prezent si anterior.

Figura 22 Timpul Pătratelor

Formula după care s-a calculat:

Unde delay(NumberOfSquare(N)), este timpul pentru pătratul respectiv

La final timpul calculat va fi stocat într-o variabila previousTime, care va fi actualizata de fiecare data pentru alt pătrat.

Acum ca avem stocat codul generat pentru matricea de LED-uri într-un sir de caractere este timpul sa-l importam sub forma unui fișier de tipul “ino”. Fișierele “ino” sunt in general folosite de către Arduino Integrated Development Enviroment, având ca si conținut codul sursa.

Pentru a face acest lucru vom crea o instanță de Blob(Binary Large Object), este un tip de obiect specia in JavaScript folosit pentru a reprezenta date binare, cum ar fi text sau fișiere binare. Acesta va lua ca argument obiectul Blob creat anterior.

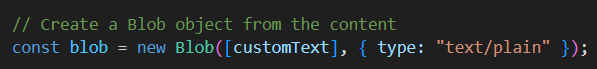


Figura 23 Obiectul Blob

Aici, un nou element de ancorare HTML (<a>) este creat în mod programatic. Acest element va servi drept link pe care utilizatorul face clic pentru a descărca fișierul.

Funcția createObjectURL generează o adresă URL unică care reprezintă conținutul obiectului blob. Această adresă URL este atribuită atributului href al linkului de descărcare, legându-l efectiv la datele Blob-ului.

Atributul de descărcare al link-ului de descărcare este setat la numele fișierului dorit pentru fișierul descărcat. În acest caz, este „LEDFile.ino”.

Această linie declanșează în mod programatic un eveniment de clic pe linkul de descărcare. Când un utilizator face clic pe un link cu atributul de descărcare setat, îi solicită să descarce fișierul legat.

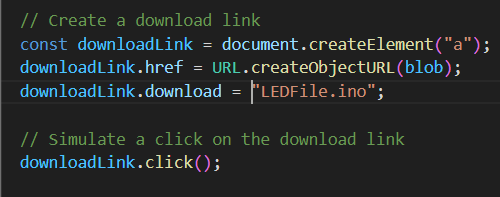


Figura 24 Crearea unui link de download

După ce descărcarea fișierului a fost inițiată, este o practică bună să revocați (eliberați) adresa URL a obiectului creat cu URL.createObjectURL. Acest lucru ajută la eliberarea resurselor asociate cu adresa URL.

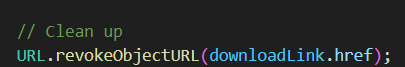


Figura 25 Revocarea adresa URL

### Script-ul pentru compilare

De regula, fișierele .ino sunt specifice arduino development, iar acest lucru este puțin defavorabil pentru unii utilizatori, acest lucru necesitând instalarea obligatorie a Arduino Development. In acest caz este posibil ca unii utilizatori sa nu dorească acest lucru, iar pentru acest lucru in prima instanță se va folosi Arduino CLI.

Arduino CLI este o soluție all-in-one care oferă plăci/Manager de bibliotecă, generator de schițe, detectarea plăcilor, încărcare și multe alte instrumente necesare pentru a utiliza orice placă și platformă compatibilă Arduino din linia de comandă sau interfețele mașinii.

Pe lângă faptul că este un instrument de sine stătător, Arduino CLI este inima tuturor programelor oficiale de dezvoltare Arduino (Arduino IDE, Arduino Web Editor). Părți din această documentație se aplică și acestor instrumente.

Cu ajutorul Arduino CLI, utilizatorul va putea rula fișierul .ino fără sa fie nevoie de alte lucruri adiționale.

Pentru rularea acestuia se va folosi un script custom de Windows unde utilizatorul va putea sa-si seteze calea spre executabilul de Arduino CLI, dar si fișierul .ino unde este pus. Adițional utilizatorul poate sa selecteze portul de comunicare, in general default este COM3.

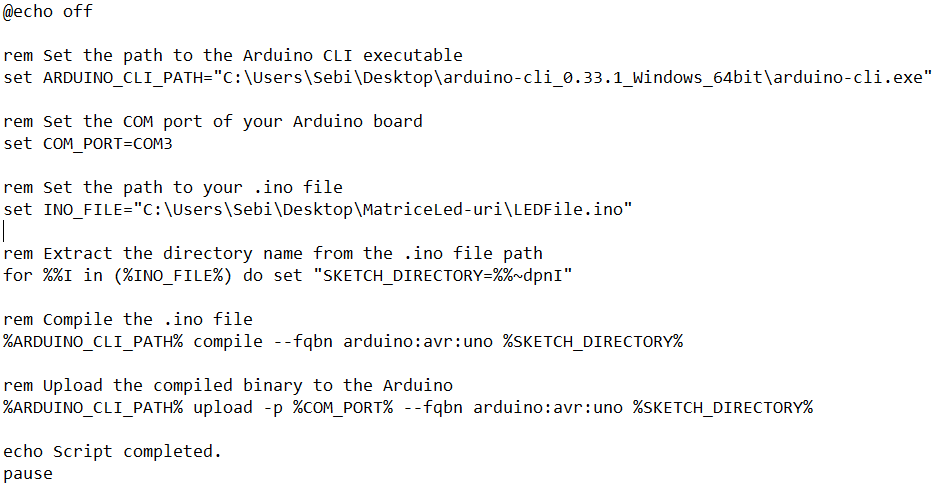


Figura 26 Script-ul pentru rularea executabilului

# Termeni de utilizare

## Autorii

Acest document a fost creat de Manea Nicolas Sebastian pentru o înțelegere mai concreta a asupra proiectului creat. De asemenea acest document a fost scris sub îndrumarea Dr. Ing. Ioan Stelian Lemeni, coordonatorul științific al acestui proiect.

## Licența de utilizare

Nu există restricții de utilizare. Documentul nu este constrâns de nicio licență. Sugestiile de îmbunătățire pot fi adresate autorului pe adresa de e-mail: maneasebastiannicolas@gmail.com.

# Concluzii

Considerând varietatea de subiecte abordate pentru realizarea acestui proiect, putem trage câteva concluzii pe baza subiectelor abordate si proiectului:

Semnificația proiectului:

* Proiectul presupune crearea unei interfețe pentru generarea codului pentru matricele LED WS2812 conectate la modulele Arduino Uno.
* Acest instrument simplifică procesul de programare a animațiilor cu matrice LED, făcându-l mai accesibil atât pentru începători, cât și pentru utilizatorii experimentați.

Interfață ușor de utilizat:

* Interfața bazată pe web oferă utilizatorilor o modalitate intuitivă de a configura animațiile cu matrice LED.
* Utilizatorii pot selecta cu ușurință culorile, pot seta intervale de timp și își pot vizualiza animațiile pe o grilă matrice LED 8x8.

Aplicații din lumea reală:

* Capacitatea de generare de cod a proiectului are aplicații practice în diverse domenii, inclusiv IoT (Internet of Things), artă digitală și educație.
* Acesta permite utilizatorilor să creeze animații LED personalizate pentru afișaje, instalații de artă și iluminat decorativ.

Personalizare și flexibilitate:

* Utilizatorii au libertatea de a alege culorile, de a ajusta intervalele de timp și de a aplica aceste setări celulelor LED individuale.
* Butonul „Aplica” permite previzualizarea și personalizarea în timp real a animațiilor cu matrice LED.

Management de stări:

* Sistemul încorporează o caracteristică de gestionare a stării care permite utilizatorilor să salveze și să gestioneze diferite configurații de matrice LED.
* Acest lucru îmbunătățește gradul de utilizare facilitând reutilizarea și modificarea stărilor salvate.

Implicarea utilizatorului:

* Proiectul încurajează implicarea și creativitatea utilizatorilor, oferind o platformă interactivă pentru proiectarea animațiilor LED.
* Capacitatea de a șterge configurațiile pătrate prin ștergerea timpului de intrare sporește confortul utilizatorului.

Sursă deschisă și gratuită:

* Proiectul pune accent pe deschidere și accesibilitate, fiind disponibil gratuit pe internet, fără nicio restricție de licență.
* Acest lucru promovează utilizarea și colaborarea pe scară largă în cadrul comunităților de matrice Arduino și LED.

Motivație și interes:

* Proiectul reflectă pasiunea dezvoltatorului pentru electronică, programare și tehnologia matricei LED.
* Motivația provine din dorința de a simplifica crearea de animație cu matrice LED și de a împărtăși această capacitate cu ceilalți.

Domeniu de aplicare și limitări:

* Scopul proiectului acoperă în primul rând dezvoltarea interfeței cu utilizatorul pentru configurarea animației cu matrice LED.
* Nu cuprinde aspectele hardware ale configurației matricei LED sau programarea detaliată pentru Arduino Uno.

Direcții viitoare:

* Luați în considerare potențialele îmbunătățiri viitoare, cum ar fi adăugarea unor funcții de animație mai avansate, suport pentru alte microcontrolere sau contribuții conduse de comunitate.
* Configurarea custom a pinilor de ieșire.
* Control de la distanta cu ajutorul unui modul HC05.
* Crearea unui back-end al site-ului pentru înregistrarea user-ilor si menținerea animațiilor create într-o baza de date.
* Simularea animațiilor.
* Configurarea custom a dimensiunilor matricii.
* Configurarea custom a nuanțelor de culoare.
* Configurarea custom a intensității luminii
* Configurarea custom a modului in care stările vor fi prelucrate. In proiectul prezentat, stările vor fi una după alta, mai exact starea următoare trebuie sa aștepte ca starea inițială sa termine animațiile la toate pătratele.

# Bibliografie

[JD11] – *HTML and CSS: Design and Build Websites* by Jon Duckett, disponibila pe https://www.amazon.com/HTML-CSS-Design-Build-Websites/dp/1118008189

[JD14] – *JavaScript and jQuery: Interactive Front-End Web Development* by Jon Duckett, disponibila pe https://www.amazon.com/JavaScript-JQuery-Interactive-Front-End-Development

[JW23] – *HTML and CSS: The Comprehensive Guide* by Jürgen Wolf, disponibila pe https://www.amazon.com/HTML-CSS-Comprehensive-J%C3%BCrgen-Wolf/dp/1493224220

[PA22] – *JavaScript: The Comprehensive Guide to Learning Professional JavaScript Programming (The Rheinwerk Computing)* by [Philip Ackermann](https://www.amazon.com/Philip-Ackermann/e/B0B524JWS9/ref=dp_byline_cont_book_1), disponibila pe https://www.amazon.com/JavaScript-Comprehensive-Learning-Professional-Programming/dp/1493222864

[MM20] – *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects* by  [Michael Margolis](https://www.amazon.com/Michael-Margolis/e/B004XUYD1U/ref=dp_byline_cont_book_1), [Brian Jepson](https://www.amazon.com/Brian-Jepson/e/B000ARA238/ref=dp_byline_cont_book_2) (Author), [Nicholas Weldin](https://www.amazon.com/s/ref=dp_byline_sr_book_3?ie=UTF8&field-author=Nicholas+Weldin&text=Nicholas+Weldin&sort=relevancerank&search-alias=books) (Author), disponibila pe https://www.amazon.com/Arduino-Cookbook-Recipes-Enhance-Projects/dp/149190352X

[BF22] – *Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques, 4th Edition* by [Ben Frain](https://www.amazon.com/Ben-Frain/e/B00BIK2USQ/ref=dp_byline_cont_book_1), disponibila pe https://www.amazon.com/Responsive-Web-Design-HTML5-CSS/dp/180324271X

[PS16] *– Practical Electronics for Inventors, Fourth Edition 4th Edition* by Paul Scherz, disponibila pe https://www.amazon.com/Practical-Electronics-Inventors-Fourth-Scherz/dp/1259587541

[CP21] – *Make: Electronics: Learning by Discovery: A hands-on primer for the new electronics enthusiast* by Charles Platt, disponibila pe https://www.amazon.com/Make-Electronics-hands-electronics-enthusiast/dp/1680456873

[TC22] – *Introduction to Algorithms, 4th edition* by Thomas H. Cormen, disponibila pe https://dl.ebooksworld.ir/books/Introduction.to.Algorithms.4th.Leiserson.Stein.Rivest.Cormen.MIT.Press.9780262046305.EBooksWorld.ir.pdf

[SS08] – *The Algorithm Design Manual* by Steven S. Skiena, disoponibila pe https://www.amazon.com/Algorithm-Design-Manual-Steven-Skiena/dp/1849967202

[DR88] – *C Programming Language, 2nd Edition* by [Brian W. Kernighan](https://www.amazon.com/Brian-W-Kernighan/e/B000AQ1TNQ/ref=dp_byline_cont_book_1), [Dennis M. Ritchie](https://www.amazon.com/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=Dennis+M.+Ritchie&text=Dennis+M.+Ritchie&sort=relevancerank&search-alias=books), disponibila pe https://www.amazon.com/Programming-Language-2nd-Brian-Kernighan/dp/0131103628

[RS20] – *Effective C: An Introduction to Professional C Programming* by Robert C. Seacord, disponibila pe https://www.amazon.com/Effective-Introduction-Professional-Robert-Seacord/dp/1718501048

[RM08] – *Clean Code: A Handbook of Agile Software Craftsmanship* by Robert C. Martin, disponibila pe <https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>

[RM17] – Clean Architecture: A Craftsman's Guide to Software Structure and Design by Robert C. Martin Series, disponibila pe https://www.amazon.com/Clean-Architecture-Craftsmans-Software-Structure/dp/0134494164

[MH18] – Eloquent JavaScript by Marijn Haverbeke

[KS14] – You Don't Know JS (book series) by Kyle Simpson

[JR21] – Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics by Jennifer Robbins

[LV15] – CSS Secrets: Better Solutions to Everyday Web Design Problems by Lea Verou

[DC08] – JavaScript: The Good Parts by Douglas Crockford

[DF20] – JavaScript: The Definitive Guide by David Flanagan

[PM14] – C Programming Absolute Beginner's Guide (3rd Edition) by Perry and Miller

[V07] – C Programming for the Absolute Beginner, Second Edition by Vine

[DT19] – The Pragmatic Programmer: Your Journey to Mastery by Andrew Hunt and David Thomas

[EG94] – Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Grady Booch

[MF99] – Refactoring: Improving the Design of Existing Code by Martin Fowler

[JD14] – Web Design with HTML, CSS, JavaScript and jQuery Set by Jon Duckett

[EM11] – Responsive Web Design by Ethan Marcotte

[SK13] – Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability by Steve Krug

# Referințe web

[1] – <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>

[2] – <https://en.wikipedia.org/wiki/HTML>

[3] – <https://en.wikipedia.org/wiki/CSS>

[4] – <https://en.wikipedia.org/wiki/JavaScript>

[5] – <https://en.wikipedia.org/wiki/Arduino>

[6] – <https://chat.openai.com/>

[7] – <https://stackoverflow.com/>

[8] – https://stackexchange.com/

[9] – <https://en.wikipedia.org/wiki/Visual_Studio_Code>

[10] – <https://en.wikipedia.org/wiki/GitHub>

[11] – https://www.udemy.com/course/the-complete-web-development-bootcamp/

# CD / DVD

Un repository pe Github al acestui proiect se poate găsii pe următorul site:

https://github.com/sebark7/LED-Configurator



# Index

A

Algoritmul de generare 20

Arduino Uno 4

Autorii 27

B

Bibliografie 31

C

CD/DVD 34

Componenta Hardware 9

Componenta Software 9

CONCLUZII 28

CSS 7

**CUPRINSUL** xiv

D

DESIGN-UL SI IMPLEMENTARE 9

Domenul de Cunostiinte Si Instrumente 4

G

Generarea codului 16

H

HTML 5

I

Interfata cu utilizatorul 10

Introducere 1

L

Licenta de utilizare 27

**LISTA FIGURILOR** xv

**LISTA TABELELOR** xvi

M

Motivatia 2

R

Referințe web 33

S

Scopul 1

Script-ul pentru compilare 25

T

TERMENI DE UTILIZARE 27

W

WS2812 7