Internet sučelje za serijsku komunikaciju

Berečić, Antonio

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni prijediplomski studij Računarstvo

INTERNET SUČELJE ZA SERIJSKU KOMUNIKACIJU

Završni rad

Antonio Berečić

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju

Ocjena završnog rada na sveučilišnom prijediplomskom studiju

Ime i prezime pristupnika:	Antonio Berečić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4613, 27.07.2021.
JMBAG:	0165089296
Mentor:	izv. prof. dr. sc. Ivan Aleksi
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Internet sučelje za serijsku komunikaciju
Znanstvena grana završnog rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak završnog rada:	Temu rezervirao Antonio Berečić. Potrebno je izraditi internet stranicu koja omogućuje serijsku komunikaciju s mikroupravljačkim sustavima. Aplikacija treba omogućiti odabir baudrate brzine (9600, 115200, ...), serijski port, znakove za terminaciju prilikom slanja niza znakova (NL, CR, Both, None). Aplikaciju treba testirati uspostavljanjem izmjene informacije između računala i mikroupravljača u oba smjera, odnosno potrebno je testirati slanje i primanje podataka.

Datum prijedloga ocjene završnog rada od strane mentora:	12.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	25.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	26.09.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 26.09.2024.

Ime i prezime Pristupnika:	Antonio Berečić
Studij:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R4613, 27.07.2021.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Internet sučelje za serijsku komunikaciju**

izrađen pod vodstvom mentora izv. prof. dr. sc. Ivan Aleksi

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD

Serijska komunikacija igra ključnu ulogu u prijenosu podataka između različitih elektroničkih uređaja, posebno u kontekstu mikroupravljača i perifernih uređaja. Iako jednostavna u svojoj osnovi, serijska komunikacija često zahtijeva učinkovito i precizno upravljanje kako bi se osigurao točan prijenos podataka i funkcionalnost sustava.

Problem kojim se bavi ovaj završni rad je izrada internet sučelja koje omogućuje dvosmjernu komunikaciju između korisnika i serijskih uređaja. Ovaj zadatak obuhvaća nekoliko važnih aspekata, uključujući dizajn sučelja koje će korisnicima omogućiti interakciju sa serijskim uređajima putem mreže te osiguravanje pouzdane razmjene podataka.

U sljedećem poglavlju slijedi pregled i analiza aktualnih rješenja za navedeni problem te prednosti i mane koje one imaju. Nadalje u trećem poglavlju opisuje se proces dizajna i implementiranja predloženog internetskog sučelja te se u četvrtome poglavlju provodi analiza rezultata te metoda testiranja. Konačno, u posljednjem poglavlju se zaključuje cijeli rad.

1.1. Zadatak završnog rada

Zadatak završnog rada je izraditi internetsku stranicu koja omogućuje serijsku komunikaciju s mikroupravljačkim sustavima s mogućnostima promjene brzine slanja podataka, promjene serijskog porta i dodavanja znaka za novi red prilikom slanja niza znakova. Na kraju je potrebno testirati slanje i primanje podataka.

2. PREGLED POSTOJEĆIH RJEŠENJA

U ovom poglavlju je obrađen pregled aktualnih praktičnih dostignuća u području internet sučelja za serijsku komunikaciju. Cilj je pogledati postojeće metode i tehnologije koje se koriste za implementaciju sličnih sustava te identificirati ključne prednosti i mane. Analizirani su radovi koji se bave rješenjima za upravljanje serijskim uređajima te razvojem web sučelja za komunikaciju putem serijskih portova.

2.1. Suvremene metode i tehnologije

2.1.1. Web Serial API

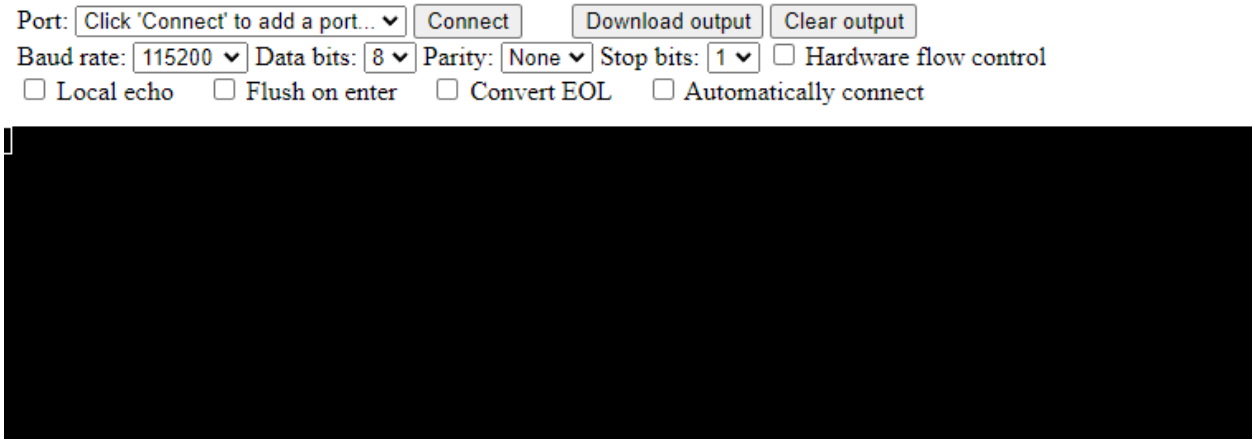
Web Serial API je moderna web tehnologija koja omogućava web aplikacijama direktan pristup serijskim portovima. Razvijen kao dio web standarda. Ovaj API omogućuje komunikaciju između web preglednika i serijskih uređaja, što omogućuje razvoj sofisticiranih web aplikacija koje mogu međusobno komunicirati s hardverom u stvarnom vremenu[1].

Značajke i funkcionalnosti:

- **Direktni pristup serijskim portovima**: Omogućava web aplikacijama da pristupaju serijskim portovima bez potrebe za dodatnim softverom ili posebnom konfiguracijom. Ovaj pristup omogućuje razvoj aplikacija koje mogu komunicirati sa serijskim uređajima direktno iz web preglednika.
- **Upravljanje parametrima komunikacije**: Omogućuje korisnicima da postave i prilagode ključne parametre serijske komunikacije, uključujući brzinu prijenosa podataka, broj podatkovnih bitova, paritet i stop bitove. Ovo pruža fleksibilnost u konfiguraciji komunikacije prema potrebama specifičnih uređaja.
- **Dvosmjerna komunikacija**: Podržava dvosmjernu komunikaciju, omogućujući korisnicima da šalju i primaju podatke u stvarnom vremenu. Ovo je korisno za aplikacije koje zahtijevaju interaktivnu komunikaciju sa serijskim uređajima.
- **Sigurnost i privatnost**: Dizajniran je s naglaskom na sigurnost i privatnost, zahtijevajući korisničku dozvolu za pristup serijskim portovima i implementirajući sigurnosne protokole kako bi zaštitio korisničke podatke i uređaje.

2.1.2. Google Chrome Labs Serial Terminal

Serial terminal je web aplikacija razvijena od strane Google Chrome Labs koja koristi Web Serial API za omogućavanje komunikacije sa serijskim uređajima direktno iz web preglednika. Ovaj alat predstavlja značajan korak naprijed u integraciji serijske komunikacije s modernim web tehnologijama[2].



Slika 2.1. Prikaz Google Chrome Labs serijskog terminala

Značajke i funkcionalnosti:

- **Spajanje na serijski port**: Omogućava korisnicima da se povežu s određenim serijskim portom, čime omogućuje direktnu komunikaciju sa serijskim uređajima bez potrebe za dodatnim softverom.
- **Preuzimanje izlaza**: Korisnici mogu preuzeti podatke sa serijskog porta u stvarnom vremenu, što omogućava učinkovito praćenje i analizu serijskih podataka.
- **Brisanje prozora**: Uključuje opciju za brisanje izlaza iz prozora, što pomaže u održavanju preglednosti i organiziranosti podataka.
- **Podešavanje parametara**: Omogućuje podešavanje brzine prijenosa podataka, broj bitova podataka, paritet, stop bitove i druge parametre, što omogućava prilagodbu serijske komunikacije specifičnim potrebama korisnika.

Prednosti:

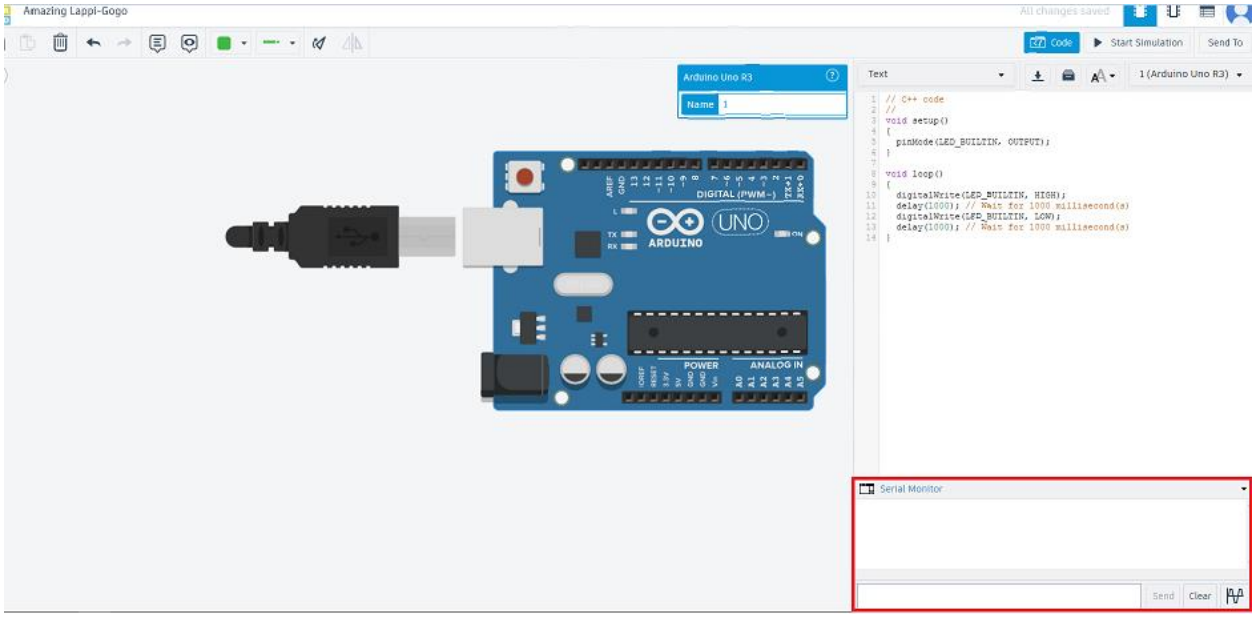
- **Napredne opcije**: Nudi raznovrstan broj opcija koje su obično prisutne u sofisticiranijim serijskim terminalima, uključujući detaljna podešavanja komunikacijskih parametara.

- **Jednostavnost korištenja**: Iako nudi napredne funkcionalnosti, sučelje je izuzetno jednostavno za korištenje. Korisnici mogu brzo pristupiti i upravljati serijskom komunikacijom bez potrebe za složenim konfiguracijama.

Serial Terminal predstavlja jedan od najnaprednijih alata u području web orijentiranih serijskih terminala. Kombinira napredne mogućnosti s iznimnom jednostavnošću korištenja, čime omogućuje korisnicima da na učinkovit način upravljaju serijskim komunikacijama izravno putem web preglednika. Ova kombinacija funkcionalnosti i pristupačnosti čini Serial Terminal značajnim rješenjem za korisnike koji zahtijevaju pouzdanu i fleksibilnu platformu za rad sa serijskim uređajima.

2.1.3. Tinkercad

Tinkercad je web aplikacija koju je razvio Autodesk i koja pruža niz alata za 3D modeliranje, elektroniku i kodiranje. Jedan od ključnih modula u Tinkercadu je Tinkercad Circuits, koji omogućuje simulaciju i testiranje elektroničkih sklopova, uključujući i serijsku komunikaciju[3].



Slika 2.2. Prikaz Tinkercad Circuits sučelja

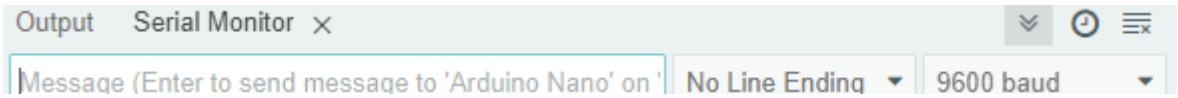
Integrirani serijski monitor u Tinkercad-u (Slika 2.2.) omogućuje korisnicima vizualizaciju podataka koji se šalju i primaju kroz serijski port u simuliranom okruženju. Ova funkcionalnost je izuzetno korisna za testiranje i otklanjanje pogrešaka u serijskoj komunikaciji. Međutim, u usporedbi s drugim serijskim monitorima, Tinkercad-ov serijski monitor ima značajne nedostatke u funkcionalnosti.

Kao što se može vidjeti na slici, Tinkercad-ov serijski monitor omogućava samo slanje podataka i brisanje prozora u kojem se prikazuju primljeni i poslani podaci (putem tipki SEND i CLEAR). Iako uključuje opciju "Toggle Graph" koja pruža vizualni prikaz serijskih podataka, što može biti korisno za analizu i razumijevanje podataka, mnoge ključne funkcionalnosti nedostaju.

Naime, serijski monitor u Tinkercad-u ne omogućuje promjenu brzine prijenosa podataka, broja podatkovnih bitova, pariteta i drugih komunikacijskih parametara. Ova ograničenja značajno smanjuju njegovu fleksibilnost i primjenjivost u kompleksnijim scenarijima serijske komunikacije, gdje su takve prilagodbe često neophodne za uspješno testiranje i otklanjanje pogreški.

2.1.4. Arduino IDE serial monitor

Arduino serial monitor je alat unutar Arduino IDE-a koji omogućuje praćenje i analizu podataka koji se šalju i primaju preko serijskog porta. Ovaj alat omogućuje korisnicima da šalju tekstualne poruke s računala na Arduino ploču i da primaju podatke s ploče na računalo. U serijskom monitoru, korisnici mogu vidjeti informacije u stvarnom vremenu, što je korisno za otkrivanje grešaka i testiranje. Koristi se za ispisivanje rezultata funkcija, praćenje varijabli i za interakciju s mikroupravljačem kroz serijsku komunikaciju[4].



Slika 2.3. Prikaz Arduino IDE serijskog monitora

Arduino serial monitor ima mogućnosti mijenjanja brzine slanja podataka, prikazivanje vremena u kojem je pojedina poruka poslana te razne terminacijske znakove poput "New Line", "Carriage Return" i "Both NL & CL". Međutim za korištenje ovog serijskog monitora potrebno je preuzeti aplikaciju Arduino IDE.

3. IZRADA INTERNET SUČELJA ZA SERIJSKU KOMUNIKACIJU

U ovom poglavlju razmatran je razvoj softvera za internet sučelje za serijsku komunikaciju, s ciljem omogućavanja učinkovitog upravljanja serijskim uređajima putem web preglednika. Programi i jezici koji su korišteni za izradu softvera su bili pregledani, uključujući njihove uloge i doprinos funkcionalnosti sučelja. Također, razmatrano je programsko rješenje potrebno za ostvarenje zadatka završnog rada. U idućem poglavlju opisano je provedeno testiranje kako bi se testirale funkcionalnosti.

3.1. Programska oprema

3.1.1. Visual Studio Code

Visual Studio Code (VS Code) je razvojno okruženje koje je steklo popularnost zbog svoje svestranosti i prilagodljivosti. Kao besplatan alat, VS Code pruža značajan set alata za razvoj, uključujući mogućnost rada s različitim programskim jezicima i tehnologijama. Njegovo korisničko sučelje je intuitivno, omogućavajući programerima brzi pristup svim potrebnim funkcionalnostima bez opterećivanja sučelja složenim opcijama.

Kroz ekstenzije i dodatke, VS Code omogućuje prilagodbu i proširenje svog funkcionalnog opsega, što ga čini pogodnim za širok raspon razvojnih potreba. Integrirani terminal i podrška za otkrivanje pogrešaka dodatno poboljšavaju učinkovitost rada, čineći ga vrijednim alatom za razvoj modernih web aplikacija i drugih softverskih rješenja[5].

3.1.2. JavaScript

JavaScript je dinamički, višenamjenski programski jezik koji je ključan za razvoj web aplikacija. Kao jezik koji se koristi za izgradnju interaktivnih i dinamičnih web stranica, JavaScript omogućuje razvoj kompleksnih funkcionalnosti na klijentskoj strani bez potrebe za ponovnim učitavanjem stranice.

Kao jedan od osnovnih jezika za razvoj weba, JavaScript se koristi u kombinaciji s HTML-om i CSS-om kako bi se poboljšalo korisničko iskustvo na internetu. Njegove mogućnosti uključuju manipulaciju DOM-a (Document Object Model), upravljanje događajima i asinkrono učitavanje podataka preko AJAX-a i Fetch API-ja. Također, JavaScript podržava različite biblioteke i frameworke poput **React**, Angular i Vue.js, koji omogućuju brži razvoj i bolju organizaciju koda.

Zbog svoje svestranosti, JavaScript se koristi ne samo za klijentsku stranu web aplikacija već i za server-side razvoj putem Node.js platforme. Ova funkcionalnost omogućuje kreiranje cijelog niza aplikacija, od jednostavnih web stranica do kompleksnih server-side rješenja, koristeći isti jezik kroz cijeli razvojni proces[6].

3.1.3. React

React je popularna JavaScript biblioteka za izgradnju korisničkih sučelja, posebno za single-page aplikacije gdje je potreban brzi i dinamičan prikaz podataka. Razvijena od strane Facebooka. React omogućuje razvoj komponentnih baza koje omogućuju jednostavno kreiranje i održavanje složenih korisničkih sučelja.

React koristi pristup zasnovan na komponentama, gdje su korisnička sučelja podijeljena na manje, samostalne komponente koje se mogu ponovno koristiti. Ove komponente mogu imati svoje vlastite stanje i logiku, što omogućuje efikasno upravljanje složenim interakcijama u aplikaciji. Pored toga, React koristi virtualni DOM (Virtual Document Object Model), što znači da umjesto direktnog manipuliranja stvarnim DOM-om, React najprije vrši promjene u virtualnom DOM-u, a zatim optimizira i ažurira stvarni DOM samo kada je to nužno. Ovaj pristup poboljšava performanse aplikacije i omogućuje brži odgovor na korisničke akcije.

Jedna od ključnih značajki React-a je njegova sposobnost da se integrira s drugim alatima i bibliotekama, kao što su Redux za upravljanje stanjem ili React Router za navigaciju, što omogućuje kreiranje robusnih i skalabilnih web aplikacija. Također, React podržava razvoj mobilnih aplikacija kroz React Native, omogućujući razvoj aplikacija koje izgledaju i funkcioniraju kao nativne aplikacije na iOS i Android platformama[7].

3.1.4. HTML i CSS

HTML (HyperText Markup Language) i CSS (Cascading Style Sheets) su osnovne tehnologije za razvoj web stranica.

HTML se koristi za strukturiranje sadržaja na web stranici. Omogućuje definiranje elemenata kao što su naslovi, paragrafe, slike i hiperveze te pruža osnovnu strukturu dokumenta pomoću različitih tagova i atributa[8].

CSS se koristi za stiliziranje i oblikovanje HTML elemenata. Omogućuje kontrolu nad izgledom elemenata na stranici, uključujući boje, fontove, razmake i raspored. CSS omogućuje stvaranje atraktivnih i responzivnih dizajna koji se prilagođavaju različitim uređajima i veličinama ekrana[9].

Kombinacija HTML-a i CSS-a omogućuje razvoj funkcionalnih i estetski privlačnih web stranica, gdje HTML pruža strukturu sadržaja, a CSS upravlja njegovim izgledom.

3.2. Programsko rješenje

Programsko rješenje za internet sučelje serijske komunikacije je razvijeno korištenjem JavaScript-a i React biblioteke. Postupak kojim je omogućeno stvaranje sučelja za serijsku komunikaciju je opisan u daljnjem dijelu.

3.2.1. Uvoz i inicijalizacija

```
import React, { useState, useEffect } from 'react';
import './App.css';
import io from 'socket.io-client';
const socket = io('http://localhost:3001');
```

Slika 3.1. Uključivanje biblioteka i inicijalizacija

Na početku koda su bile uvezane potrebne biblioteke i inicijalizirana potrebna WebSocket veze:

- React biblioteka je uvezena za korištenje React komponenti i "hookova".
- CSS datoteka je uvezena za stiliziranje HTML stranice.
- Socket.io-client je uvezen za uspostavljanje WebSocket veze.
- Nova WebSocket veza s poslužiteljem, u ovom slučaju localhost (odnosno vlastito računalno), je stvorena pomoću "const socket", što je omogućilo dvosmjernu komunikaciju između klijenta i poslužitelja.

3.2.2. Stanje komponenti

```
const [message, setMessage] = useState('');
const [terminalOutput, setTerminalOutput] = useState('');
const [baudRate, setBaudRate] = useState('9600');
const [port, setPort] = useState(null);
const [writer, setWriter] = useState(null);
const [textDecoder] = useState(new TextDecoder());
const [addLine, setAddLine] = useState(false);
```

Slika 3.2. Definiranje stanja komponenti

React "useState" hook je korišten za definiranje različitih stanja potrebnih za funkcioniranje aplikacije:

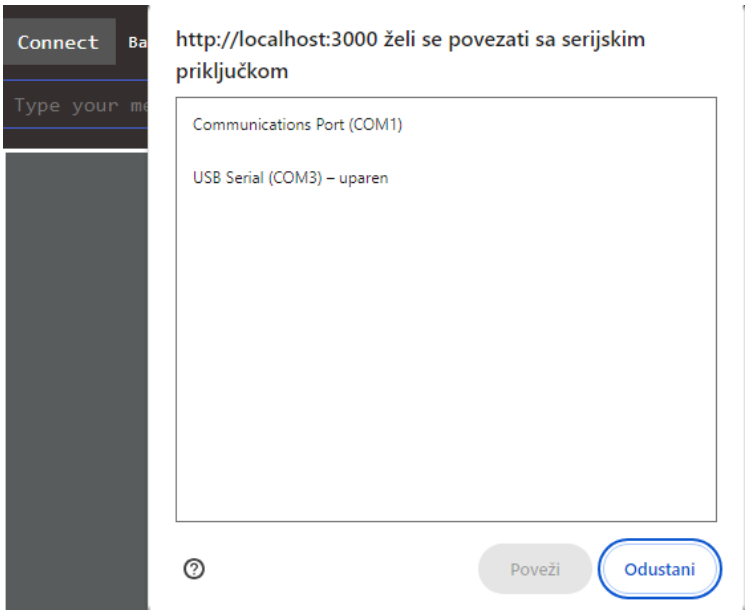
- Stanje poruka je definiralo varijablu koja pohranjuje tekst koju korisnik unosi i planira poslati putem serijskog porta.
- Stanje izlaza terminala je omogućilo pohranjivanje primljenih podataka sa serijskog porta koji su se prikazivali u terminalu aplikacije.
- Stanje brzine slanja poruke ili "BaudRate" pohranjuje brzinu prijenosa podataka za serijsku vezu. Početna vrijednost je 9600, a korisnik može promijeniti prema potrebama.
- Stanje serijskog porta pohranjuje referencu na trenutno odabrani serijski port, u početku je postavljen na "null" a kasnije se ažurira kada korisnik odabere port.
- Stanje pisača omogućuje slanje podataka na serijski port, stanje se ažurira kada se veza uspostavi i pisač se inicijalizira.
- Stanje dekodera teksta se koristi za pretvaranje primljenih bajtova u tekstualni format. Ovo stanje je ključno za interpretaciju poruka sa serijskog porta.
- Stanje dodavanja novog retka koristi se da se označi treba li dodavati znak novog retka pri slanju podataka.

3.2.3. useEffect za Serijski Port

```
useEffect(() => {
    if (port) {
        listenToPort();
    }

    return () => {
        if (port) {
            port.close();
        }
    };
}, [port]);
```

Slika 3.3. Funkcija useEffect za serijski priključak

Ovaj effectHook pokreće se svaki put kada se stanje port promijeni. Vrlo je važan jer pri svakoj promijeni stanja porta se aktivira. Ako je port postavljen, poziva se funkcija listenToPort koja počinje čitati podatke s porta. U "return" funkciji port se zatvara kada komponenta prestane biti prikazana ili se port mijenja.



Slika 3.4. Prikaz sučelja za odabir serijskog priključka

3.2.4. useEffect za pomicanje terminala

```
useEffect(() => {
    const terminalDiv = document.getElementById("serialResults");
    terminalDiv.scrollTop = terminalDiv.scrollHeight;
}, [terminalOutput]);
```

Slika 3.5. Funkcija useEffect za terminal

Zbog problema s automatskim prikazivanjem novih podataka, gdje je bilo potrebno listanje prema dolje kako bi se prikazali, dodan je ovaj efectHook. Hook se pokreće svaki put kada se stanje "terminalOutput" ažurira te automatski pomiče sadržaj terminala na dno kako bi se uvijek prikazivali najnoviji podaci.

3.2.5. Funkcija connectSerial

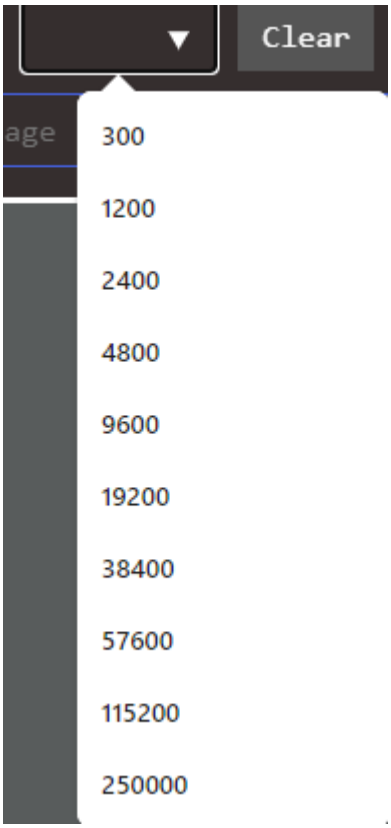
```
const connectSerial = async () => {
    try {
        const selectedPort = await navigator.serial.requestPort();
        const baudRate = parseInt(document.getElementById("baud").value, 10);
        await selectedPort.open({ baudRate });

        const textEncoderStream = new TextEncoderStream();
        const writableStreamClosed = textEncoderStream.readable.pipeTo(selectedPort.writable);
        const writer = textEncoderStream.writable.getWriter();

        setPort(selectedPort);
        setWriter(writer);

        listenToPort();
    } catch (error) {
        console.error("Serial Connection Failed", error);
        alert("Serial Connection Failed");
    }
};
```

Slika 3.6. Funkcija connectSerial

Funkcija connectSerial služi za uspostavljanje veze sa serijskim portom. Koristi se asinkrona funkcija kako bi se omogućilo „pauziranje“ dok se čekaju rezultati bez blokiranja ostatka aplikacije, isto tako omogućuje se primjena "try" i "catch" blokova za rukovanje greškama. Ovo uključuje nekoliko ključnih koraka:

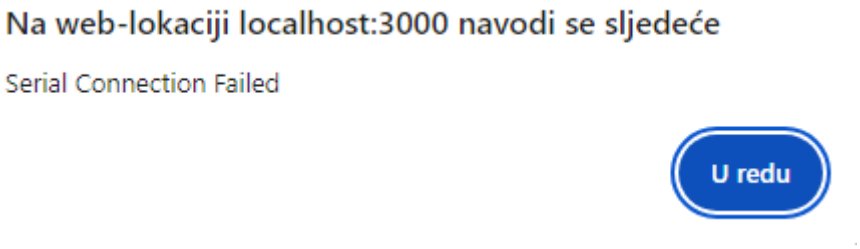
- Proces započinje odabirom serijskog porta od strane korisnika, čime se omogućuje aplikaciji da šalje i prima podatke s serijskog uređaja.
- Serijski port zahtijeva postavljanje brzine prijenosa podataka, poznate kao "baudRate". Nakon odabira porta, funkcija dohvaća brzinu prijenosa podataka iz korisničkog odabira

sa web stranice. Odabir brzine prijenosa podataka ključan je za usklađivanje brzine prijenosa podataka između aplikacije i serijskog uređaja.



Slika 3.7. Sučelje za odabir brzine slanja podataka

- U daljnjem dijelu postavljaju se pisači koji kodiraju tekst u bajtove prije slanja preko serijskog porta. Omogućuje se slanje podatka te se pisač konfigurira za slanje podataka na serijski port. Nakon toga pokreće se funkcija "listenToPort" kako bi se započelo čitanje podataka sa serijskog porta, što omogućuje aplikaciji da prima i obrađuje podatke.
- Na kraju, ako dođe do grešaka, funkcija hvata iznimke i prikazuje poruku o grešci.



Slika 3.8. Prikaz poruke o grešci pri spajanju

3.2.6. Funkcija listenToPort

```
const listenToPort = async () => {
    if (port) {
        const reader = port.readable.getReader();
        let isFirstMessage = true;
        try {
            while (true) {
                const { value, done } = await reader.read();
                if (done) {
                    break;
                }
                const decodedMessage = textDecoder.decode(value);

                if (isFirstMessage) {
                    isFirstMessage = false;
                    continue;
                }

                setTerminalOutput(prev => prev + decodedMessage);
            }
        } catch (error) {
            console.error("Error reading from port", error);
        } finally {
            reader.releaseLock();
        }
    }
};
```

Slika 3.9. Funkcija listenToPort

Funkcija listenToPort služi za kontinuirano praćenje i čitanje podataka koji dolaze sa serijskog porta. Njezina implementacija sastoji se od nekoliko ključnih koraka:

- Stvaranje čitatelja za čitanje podataka sa serijskog porta, čime se omogućuje asinkrono čitanje podataka.
- Varijabla "isFirstMessage" služi kao zastavica za preskakanje prve poruke.

```
��f□�����□□x□��□□□xff�����
```

Slika 3.10. Prikaz znakova koji se javljaju kao prva poruka

- Textdecoder koristi za dekodiranje dolaznih bajtova u čitljiv tekst.
- Nakon dekodiranja podataka, pomoću funkcije "setTerminalOutput" prikazuju se podaci u korisničkom sučelju.

- Catch blok rukuje greškama ukoliko dođe do istih prilikom čitanja podataka te nakon završetka čitanja, čitatelj se oslobađa što omogućuje pravilno upravljanje resursima.

3.2.7. Funkcija sendSerialLine

```
const sendSerialLine = async () => {
    let dataToSend = message;
    if (addLine) dataToSend += "\n";
    await writer.write(dataToSend);
    setMessage('');
};
```

Slika 3.11. Funkcija sendSerialLine

Funkcija "sendSerialLine" prvo priprema podatke koje treba poslati, koristeći stanje "message" koje sadrži poruku unesenu od strane korisnika. Ukoliko je opcija "addLine" uključena, dodaje se znak za novi red ('\n') na kraj poruke. Nakon pripreme podataka, pomoću pisača koji je ranije inicijaliziran, podaci se prenose na serijski port u obliku bajtova. Na kraju uspješnog slanja podataka, stanje "message" se resetira na prazno, čime se korisniku omogućuje unos nove poruke za slanje bez brisanja prethodne.

3.2.8. Dodatne funkcije

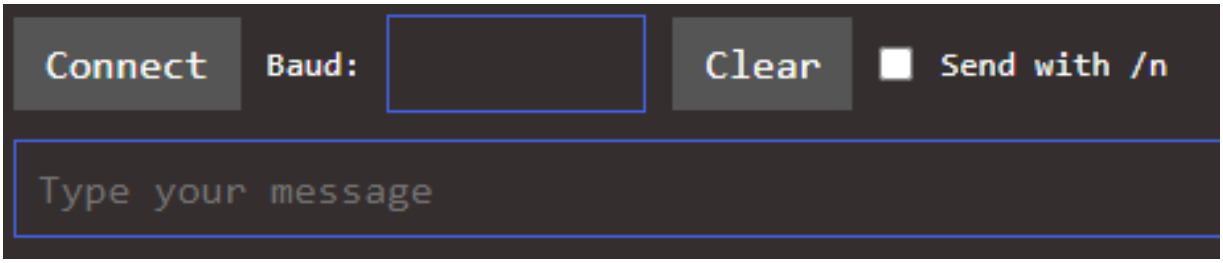
```
const changeBaudRate = (e) => {
    setBaudRate(e.target.value);
};

const clearTerminal = () => {
    setTerminalOutput('');
};

const handleCheckboxChange = (e) => {
    setAddLine(e.target.checked);
};
```

Slika 3.12. Dodatne funkcije

Funkcija "changeBaudRate" omogućuje jednostavno mijenjanje brzine prijenosa podataka korisnicima pomoću "setBaudRate" stanja i korisnički odabrane brzine prijenosa.

Funkcija "clearTerminal" služi za čišćenje prikaza terminala u aplikaciji. Pomaže korisnicima u uklanjanju prethodnih podatka, čime se poboljšava preglednost novih informacija.

Funkcija "handleCheckboxChange" kontrolira hoće li se pri slanju poruke na serijski port dodat znak za novi red ('\n'). Omogućuje korisnicima da prilagode formatiranje podataka koji se šalju.



Slika 3.13. Prikaz sučelja terminala

4. TESTIRANJE PROGRAMSKOG RJEŠENJA

U ovom poglavlju provedeno je testiranje programskog rješenja. Cilj poglavlja bio je potvrditi da rješenje ispunjava zahtjeve, odnosno da se ustanovi uspješno povezivanje te primanje i slanje podataka.

4.1. Arduino IDE

Arduino IDE je ključan alat za programiranje Arduino ploča, posebno kada se koristi za serijsku komunikaciju putem UART (Universal Asynchronous Receiver/Transmitter) protokola. UART omogućava jednostavnu serijsku razmjenu podataka između Arduino mikroupravljača i drugih uređaja poput računala, senzora ili drugih mikroupravljača.

Kroz Arduino IDE, korisnici mogu programirati Arduino ploče za serijsku komunikaciju pomoću funkcija kao što su Serial.begin(), Serial.print() i Serial.read(). Funkcija Serial.begin(baud_rate) postavlja brzinu prijenosa podataka, dok Serial.print() i Serial.println() omogućuju slanje podataka u serijskom formatu, a Serial.read() omogućuje čitanje primljenih podataka. Serijski monitor u IDE-u omogućuje praćenje i analizu podataka koji se šalju i primaju putem serijskog porta, što pomaže u testiranju i otkrivanju pogrešaka.

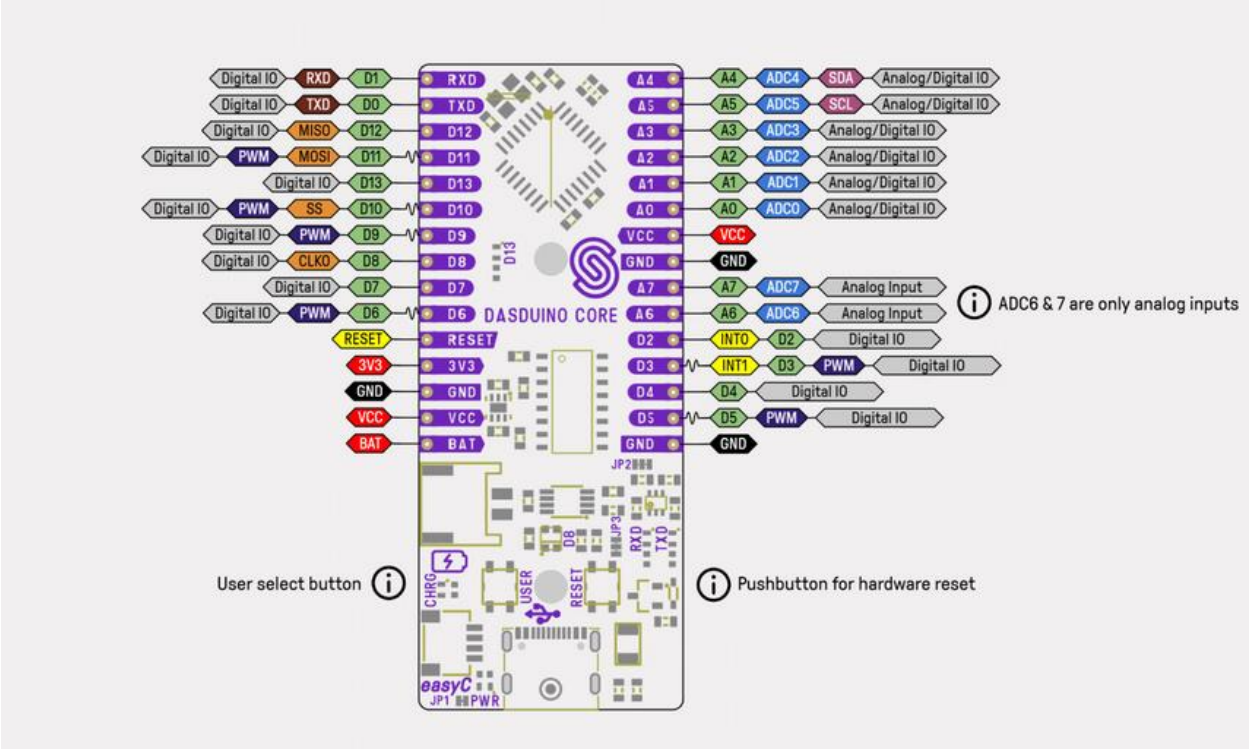
U kontekstu različitih projekata, UART komunikacija igra važnu ulogu u integraciji Arduino ploča s vanjskim uređajima, omogućujući učinkovitu razmjenu podataka i kontrolu različitih komponenti. Arduino IDE omogućuje razvoj i optimizaciju serijskih komunikacijskih protokola koji su ključni za uspjeh mnogih elektroničkih rješenja.

4.2. Dasduino CORE

Dasduino Core je specijalizirana razvojna ploča koja se temelji na proširenju Arduino platforme. Dizajnirana je za specifične primjene i nudi unaprijeđene funkcionalnosti u odnosu na standardne Arduino ploče. Pločica je bazirana na mikroprocesoru Atmega328.

Komuniciranje s mikroprocesorom ostvaruje se putem UART,SPI ili I2C protokola, međutim u našem slučaju koristi se UART. Programiranje pločice ostvaruje se preko USB-c priključka koji se spaja na računalo i programira putem Arduino IDE programskog okružja.

Funkcionira na nazivnom naponu od 5V, sastoji se od 22 nožica od kojih su 14 digitalnih, a 8 analognih.



Slika 4.1. Prikaz priključnih nožica Dasduino CORE pločice[10]

4.3. Programsko rješenje

4.3.1. Setup funkcija

```
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ;
  }
}
```

Slika 4.2. Setup funkcija

Gornjim kodom inicijalizira se brzina prijenosa podataka od 9600 bit/s (bauda) što postavlja serijski port za komunikaciju s računalom na ovu brzinu.

Linija "while (!Serial)" omogućuje da serijska veza bude potpuno uspostavljena prije nego što se nastavi s izvršavanjem ostatka koda.

4.3.2. Loop funkcija

```
void loop() {
  Serial.println("Hello from Arduino!");

  if (Serial.available() > 0) {
    char incomingByte = Serial.read();
    Serial.print("I received: ");
    Serial.println(incomingByte);
  }
  delay(1000);
}
```

Slika 4.3. Loop funkcija

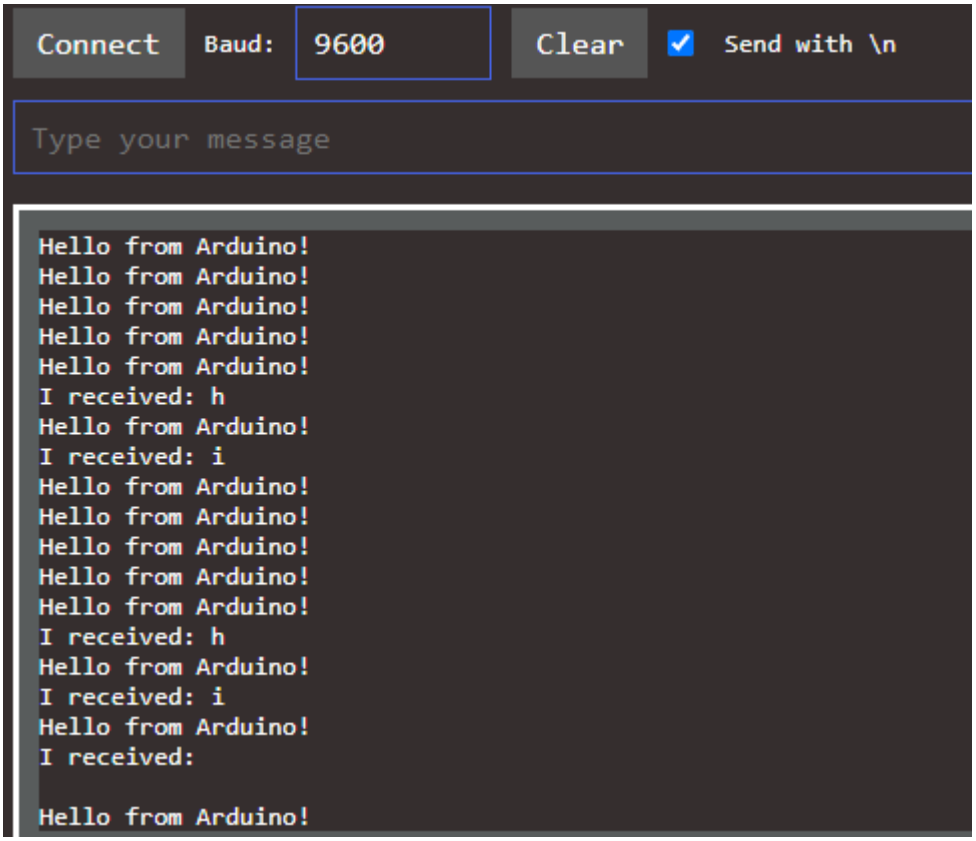
Linija "Serial.println(..)" šalje poruku na serijski monitor svake sekunde te ukoliko je slanje podataka uspješno na serijskom monitoru na web stranici bi trebali vidjeti istu poruku.

Nadalje, provjerava se postoji li nešto za pročitati iz serijskog ulaza, ako postoji, podaci se čitaju i spremaju na varijablu "incomingByte". Na kraju, na serijskom monitoru prikazuje se poruka "I received: " zajedno sa primljenim znakovima iz serijskog ulaza, a zatim se dodaje novi red.

Učitavanjem ovog programa na Dasduino Core trebali bi moći ustanoviti radi li serijski terminal na web stranici kako treba i jesu li funkcionalnosti zadovoljene.

4.4. Testiranje programa na web stranici

Pokretanjem web stranice i spajanjem Dasduino Core-a na računalo, prvo je potrebno povezati se na stranicu putem tipke "connect".



Slika 4.4. Prikaz internetskog sučelja serijske komunikacije

Možemo primijetiti iz slike 4.4. da nakon uspješnog spajanja nisu vidljivi simboli prve preskočene poruke iz slike 3.10. Nadalje, svake sekunde podaci su se uspješnom slali sa serijskog uređaja na serijski monitor samom pojavom poruka „Hello from Arduino!“. Slanjem niza znakova „hi“ primijećeno je da serijski monitor prima te znakove ispravno i ispisuje ih jedan po jedan. Također, pri drugom slanju, primijećeno je da se šalje i prazan red zbog označene opcije "Send with \n".

Iz ovog primjera zaključeno je da je glavna funkcionalnost slanja i primanja podataka uspješno implementirana te da je omogućeno slanje zajedno sa novim redom i promjena brzine prijenosa.

5. ZAKLJUČAK

U ovom radu razvijeno je internetsko sučelje za serijsku komunikaciju koje je uspješno ispunilo postavljene ciljeve. Tijekom izrade provedena je detaljna analiza postojećih rješenja, uključujući različite serijske monitore i tehnologije poput Web Serial API. Ova analiza omogućila je razumijevanje prednosti i mana postojećih alata te je poslužila kao temelj za razvoj vlastitog sučelja.

Praktični dio rada, izveden u Visual Studio Code-u pomoću JavaScripta i biblioteke React, uz HTML i CSS, doveo je do stvaranja funkcionalnog internetskog sučelja. Testiranjem s Arduino IDE-om i Dasduino Core-om potvrđena je ispravnost sučelja i uspješna komunikacija između sučelja i serijskih uređaja.

Iako su postignuti značajni rezultati, postoji prostor za poboljšanje sučelja. Predloženo je dodavanje opcija za odabir podatkovnih bitova, pariteta i stop bitova, što bi omogućilo veću fleksibilnost i prilagodbu u konfiguriranju serijske komunikacije. Također, implementacija mogućnosti preuzimanja izlaznih podataka omogućilo bi bolju analizu i pregled podataka.

LITERATURA

[1] MDN Web Docs. (2024). Web Serial API, dostupno na: https://developer.mozilla.org/en-US/docs/Web/API/Web_Serial_API [5. rujna 2024]

[2] Google Chrome Labs (2019.) Serial Terminal. dostupno na: https://googlechromelabs.github.io/serial-terminal [5. rujna 2024]

[3] Autodesk, Inc. (2023). Tinkercad. Dostupno na: https://www.tinkercad.com [5. rujna 2024].

[4] Arduino LLC (2023). Arduino IDE (Integrated Development Environment**)** (Verzija 2.1.0), dostupno na: https://www.arduino.cc/en/software [5. rujna 2024]

[5] Microsoft Corporation (2023). Visual Studio Code (Verzija 1.81), dostupno na: https://code.visualstudio.com/ [5. rujna 2024]

[6] Meta Platforms, Inc. (2023). React (Verzija 18.2.0), dostupno na: https://reactjs.org/ [5. rujna 2024]

[7] ECMA International, ECMAScript 2023 (ES14) (2023). Javascript, dostupno na: https://www.ecma-international.org/publications-and-standards/standards/ecma-262/ [5. rujna 2024]

[8] World Wide Web Consortium (W3C) (2014). HTML5, dostupno na: https://www.w3.org/TR/html5/ [5. rujna 2024]

[9] World Wide Web Consortium (W3C) (2014). CSS3, dostupno na: https://www.w3.org/Style/CSS/Overview.en.html [5. rujna 2024]

[10] Soldered, Dasduino CORE, Dasduino Core Pinout drawing [online], Soldered, Osijek, 2022, dostupno na: https://soldered.com/productdata/2022/04/Dasduino-CORE.png [6. rujna 2024]

SAŽETAK

Zadatak ovog završnog rada bio je razvoj internetskog sučelja za serijsku komunikaciju s ciljem komunikacije između računala i serijskih uređaja. Smjernice za rješavanje problema uključivale su analizu postojećih rješenja poput serijskih monitora i tehnologija kao što je Web Serial API te razvoj vlastitog sučelja koristeći JavaScript, biblioteku React, HTML i CSS. Postignuti rezultati uključuju uspješno razvijeno internetsko sučelje koje omogućuje funkcionalnu komunikaciju između serijskog uređaja i web stranice. Razvijeno sučelje je jednostavno za korištenje, što omogućuje korisnicima lako upravljanje serijskim uređajima bez potrebe za dodatnim alatima. Testiranje je potvrdilo ispravnost sučelja i njegovu sposobnost za slanje i primanje podataka. Zaključno, projekt je uspješno ostvario postavljene ciljeve i otvorio mogućnosti za daljnja unaprjeđenja.

Ključne riječi: Internetsko sučelje, JavaScript, Serijska komunikacija, Web Serial API

ABSTRACT

Title: Internet interface for serial communication

The task of this final paper was to develop an internet interface for serial communication with the goal of facilitating communication between a computer and serial devices. The approach to solving the problem involved analyzing existing solutions such as serial monitors and technologies like the Web Serial API, and creating a custom interface using JavaScript, the React library, HTML, and CSS. The results achieved include a successfully developed internet interface that enables functional communication between the serial device and the web page. The interface is user-friendly, allowing users to easily manage serial devices without the need for additional tools. Testing confirmed the correctness of the interface and its capability to send and receive data. In conclusion, the project successfully met its objectives and opened possibilities for further improvements.

Keywords: Internet interface, JavaScript, Serial communication, Web Serial API