

# Pametni HDMI razdjelnik

---

**Filipović, Franjo**

**Master's thesis / Diplomski rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:917707>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-26**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**

**INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Diplomski sveučilišni studij**

**PAMETNI HDMI RAZDJELNIK**

**Diplomski rad**

**Franjo Filipović**

**Osijek, 2023.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit**

Osijek, 15.09.2023.

**Odboru za završne i diplomske ispite****Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime Pristupnika:</b>	Franjo Filipović
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
<b>Mat. br. Pristupnika, godina upisa:</b>	D-47ARK, 13.10.2020.
<b>OIB studenta:</b>	39794995553
<b>Mentor:</b>	doc. dr. sc. Ivan Vidović
<b>Sumentor:</b>	,
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	izv. prof. dr. sc. Tomislav Matić
<b>Član Povjerenstva 1:</b>	doc. dr. sc. Ivan Vidović
<b>Član Povjerenstva 2:</b>	izv. prof. dr. sc. Ivan Aleksi
<b>Naslov diplomskog rada:</b>	Pametni HDMI razdjelnik
<b>Znanstvena grana diplomskog rada:</b>	<b>Arhitektura računalnih sustava (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	U ovom radu potrebno je istražiti postojeća rješenja za upravljanje HDMI razdjelnikom. Nakon istraživanja potrebno je implementirati vlastito rješenje koje će omogućiti udaljeno upravljanje HDMI razdjelnikom. Potrebno je implementirati mogućnost ručnog i automatskog odabira (vremenski intervali) HDMI izvora koji će se u određenom trenutku prosljeđivati na izlaz HDMI razdjelnika.
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	15.09.2023.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:



**FERIT**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

## IZJAVA O ORIGINALNOSTI RADA

Osijek, 04.10.2023.

Ime i prezime studenta:

Franjo Filipović

Studij:

Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije

Mat. br. studenta, godina upisa:

D-47ARK, 13.10.2020.

Turnitin podudaranje [%]:

2

Ovom izjavom izjavljujem da je rad pod nazivom: **Pametni HDMI razdjelnik**

izrađen pod vodstvom mentora doc. dr. sc. Ivan Vidović

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 studenta:

# Sadržaj

1. UVOD.....	1
2. PREGLED STANJA PODRUČJA .....	2
2.1. HDMI tehnologija.....	2
2.2. Što je HDMI razdjelnik?.....	3
2.3. Pregled postojećih rješenja.....	4
3. REALIZACIJA PAMETNOG HDMI RAZDJELNIKA.....	7
3.1. Analiza predloženog sustava.....	7
3.2. Sklopovlje sustava.....	9
3.3. Algoritmi upravljanja.....	13
4. TESTIRANJE SUSTAVA.....	22
4.1. Rezultati testiranja.....	24
5. ZAKLJUČAK .....	29
LITERATURA .....	31
Popis i opis upotrijebljenih oznaka i kratica .....	33
Sažetak.....	34
Abstract .....	35
Životopis.....	36

# 1. UVOD

HDMI (engl. *High-Definition Multimedia Interface*) razdjelnici pružaju fleksibilnost povezivanja većeg broja ulaznih i izlaznih uređaja, no često dolaze s nedostacima koji na neki način umanjuju korisničko iskustvo. S obzirom na to da postoje različite potrebe korisnika za upravljanjem HDMI signalom, također postoji širok raspon razdjelnika, od jednostavnih modela s ručnim prebacivanjem do složenijih AV (engl. *Audio/video*) prijemnika sa širokim spektrom ulaza. Dok AV prijemnici nude optimizaciju zvuka i slike te centralizirano upravljanje, njihova složenost i cijena nisu prihvatljivi za široku primjenu. S druge strane, iako su HDMI razdjelnici pristupačniji, često dolaze s problemima poput nedostatka automatskog prebacivanja ili potrebe za dodatnim daljinskim uređajima. Ovaj rad se usredotočuje na razvoj pametnog HDMI razdjelnika s mogućnošću povezivanja više izvora na jedan izlazni uređaj, kombinirajući praktičnost jednostavnog preklopnika s funkcionalnostima usmjerenima na bolju korisničku interakciju i udaljeno upravljanje.

Zadatak ovog diplomskog rada je istražiti postojeća rješenja upravljanja HDMI razdjelnikom te implementirati vlastito rješenje. Ovo rješenje trebalo bi omogućiti udaljeno upravljanje te ručni i automatski odabir izvora koji će se u određenom trenutku prosljeđivati na izlaz HDMI razdjelnika. Pri tome je također važan odabir komponenata sklopovlja za upravljanje i povezivanje u funkcionalan i praktičan sustav. Cilj je postići udaljeno i automatsko upravljanje razdjelnikom na nekoliko načina, izravnim odabirom željenog izvora, postavljanjem vremenskog intervala promjene izvora i postavljanjem vremenske oznake prebacivanja.

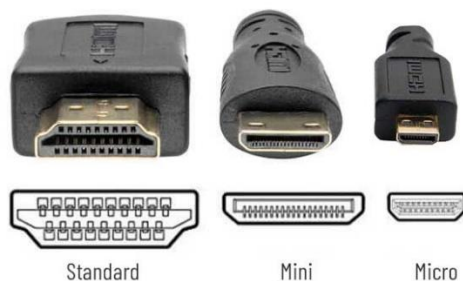
Rad je strukturiran u pet poglavlja, u ovome poglavlju iznesen je uvod u temu i problematiku zadatka, drugo poglavlje pruža teorijsku podlogu, objašnjava korištene tehnologije i pojmove te pregled postojećih rješenja. Treće poglavlje je ono središnje koje sadrži analizu predloženog sustava, opisuje izvedbu rješenja, primijenjene komponente sklopovlja i njihovu povezanost, metode i tehnologije korištene kod pisanja programskog koda i algoritama upravljanja. Četvrto poglavlje sadrži postignute rezultate konačnog rješenja, daje opis rezultata i njihova testiranja, dok je u posljednjem, petom poglavlju iznesen zaključak.

## 2. PREGLED STANJA PODRUČJA

U ovom poglavlju obrađen je pregled stanja područja vezanog za temu pametnog HDMI razdjelnika. Kroz tri odjeljka pobliže su objašnjeni sami pojmovi i tehnologije, funkcije i rad razdjelnika te analiza postojećih rješenja i uređaja sličnoga tipa i svrhe.

### 2.1. HDMI tehnologija

HDMI sučelje nastalo je suradnjom tvrtki Hitachi, Panasonic, Philips, Silicon Image, Sony, Thomson i Toshiba 2002. godine kada počinju s razvojem HDMI 1.0 standarda za prijenos video i audio sadržaja za HDTV (televiziju visoke rezolucije) [1]. Standard je omogućio povezivanje uređaja za prijenos video sadržaja bez kompresije i audio sadržaja s kompresijom ili bez nje, bio je kompatibilan s tada još novim DVI (engl. *Digital Visual Interface*) sučeljem za prijenos digitalnog signala, a imao je mogućnost prijenosa audio podataka i prijenosa istih signala bez gubitka kvalitete videa. Također, novi standard predstavio je i fizički manje konektore u više različitih veličina za praktičnije povezivanje raznih uređaja. 2009. godine objavljena je HDMI 1.4 verzija, prva koje podržava prijenos 4K videa [2]. Najnoviji standardi 2.0 i 2.1 omogućuju još veću maksimalnu širinu pojasa s 18 odnosno 48 Gbit/s te veće rezolucije i broj okvira (sličica) po sekundi, fps (engl. *frames per second*). Priključci se sastoje od 19 nožica (engl. *pin*) i dolaze u tri veličine, to su tip A (*Standard*), tip C (*Mini*) i tip D (*Micro*) (Sl. 2.1.) [3].



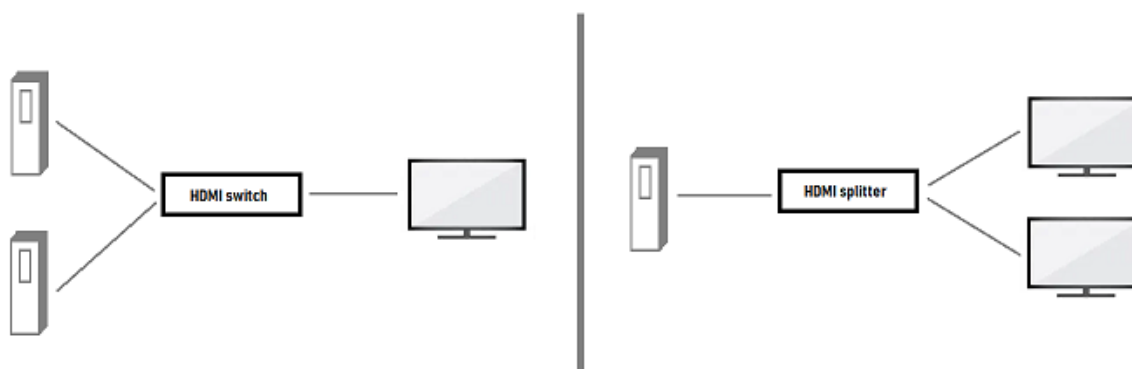
Sl. 2.1. Vrste HDMI priključaka. [3]

Tipovi kablova također se razlikuju po karakteristikama i mogućnostima pa tako kablovi kategorije 1 pružaju 4.95 Gbit/s i rezoluciju 1080i ili 720p, kategorija 2 podržava 10.2 Gbit/s i rezoluciju 1080p ili 4K pri 30 Hz dok kategorija 3 pruža 18 i 48 Gbit/s te rezolucije 4K pri 60 Hz, 8K pri 60 Hz i 4K

pri 120 Hz. HDMI koristi i HDCP (engl. *High-bandwidth Digital Content Protection*) zaštitu digitalnog sadržaja koja sprječava kopiranje i prijenos preko neautoriziranih uređaja [3].

## 2.2. Što je HDMI razdjelnik?

Prilikom prijenosa audiovizualnog sadržaja HDMI kablom, ovisno o situaciji i raspodjeli postojećih uređaja, poželjno je imati mogućnost odabira ulaznih i izlaznih uređaja bez iskopčavanja i prespajanja kablova. HDMI razdjelnici služe za prijenos digitalnog sadržaja na više izlaza (monitora, projektor) te za promjenu ulaza pritiskom tipkala na samom razdjelniku ili pomoću infracrvenog daljinskog uređaja. Razdjelnik (engl. *splitter*) u užem smislu služi za prijenos sadržaja s jednog ulaznog uređaja na više izlaznih uređaja, dijeli ulazni signal na dva jednaka signala; dostupan je kao pasivan uređaj bez pojačavanja signala te zadovoljava prijenos na do dva izlaza, dok se kod više izlaza i prilikom korištenja dugih kablova signal treba pojačati te se koriste aktivni razdjelnici. HDMI preklopnik (engl. *switch*) omogućuje odabir jednoga od više spojenih ulaznih uređaja i prijenos audiovizualnog sadržaja na jedan izlazni uređaj. Na slici 2.2. prikazana je razlika HDMI razdjelnika tipa *switch* (preklopnik) i *splitter*. Također postoje i razdjelnici koji povezuju funkcije prijenosa videa na više izlaza i odabira između više mogućih ulaza (engl. *matrix*) [4].

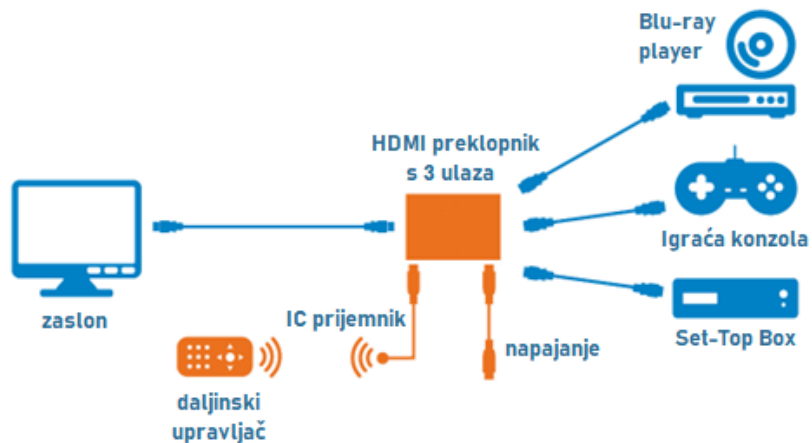


Sl. 2.2. Razlika između HDMI razdjelnika (lijevo switch, desno splitter).

Najčešće korišteni uređaj je preklopnik, a pogodan je u slučaju malog broja HDMI priključaka na izlaznom uređaju i s većim brojem ulaznih uređaja koji se koriste u kućnim TV postavima. Preklopnik omogućuje spajanje osobnog računala, igračih konzola, STB (engl. *set-top box*) uređaja i *Blu-ray* ili *DVD-playera* na jedan zaslon. Na slici 2.3. prikazan je način spajanja preklopnika i raspored ulaznih



i izlaznih uređaja, s vlastitim napajanjem preklopnika i prijemnikom za IR (infracrveni, engl. *infrared*) daljinski upravljač.



Sl. 2.3. Prikaz povezivanja 3u1 HDMI preklopnika. [5]

### 2.3. Pregled postojećih rješenja

HDMI razdjelnici koji su u ponudi najčešće nude mogućnost povezivanja od dva pa do pet i više ulaznih uređaja, većina ih ima omogućen odabir ulaznog kanala pomoću tipkala na razdjelniku, pri čemu svjetleća dioda pokazuje ulaz koji je trenutno povezan. Također kao rješenje postoje i AV prijemnici, kompleksniji uređaji za sustave kućnog kina. AV prijemnik radi kao pojačivač signala i pruža prijenos s optimiziranom kvalitetom slike i zvuka, uključuje mogućnost upravljanja velikog broja ulaza preko jednog daljinskog uređaja i spajanje više uređaja različitih formata (npr. digitalni i analogni audio, kompozitni video, S-Video, komponentni i SCART video te HDMI i USB). Za razliku od AV prijemnika, HDMI razdjelnik tj. preklopnik je jednostavniji i jeftiniji uređaj manjih dimenzija koji služi za odabir tj. spajanje jednog od više povezanih HDMI ulaza na jedan izlazni uređaj. Razdjelnici također mogu imati CEC (engl. *Consumer Electronic Control*) značajku za upravljanje preko jednog daljinskog upravljača, no češće je u ponudi dodatni eksterni prijemnik za infracrveni signal s posebnog daljinskog upravljača za odabir ulaza. Automatsko prebacivanje (engl. *auto-switching*) je značajka pomoću koje HDMI razdjelnik samostalno prepoznaje i usmjerava aktivni ulazni signal. Kada se na njega spoji novi uređaj, razdjelnik s ovom značajkom automatski prebacuje na taj novi ulaz. Često u slučaju više aktivnih ulaza odabire se nasumičan spojeni uređaj, a također se

događa i da spojeni ulazi nemaju sposobnost isključivanja HDMI signala pa ih razdjelnik prepoznaje kao aktivne i kada to nisu. Nedostaci „običnijih“ HDMI razdjelnika su ručno prebacivanje, potreba korištenja posebnog daljinskog uređaja, nedostatak automatskog prebacivanja te dodatnih funkcija i značajki naprednog upravljanja (npr. postavljanje vremenskih intervala prebacivanja signala, uključivanje željenog ulaza ili neke funkcije u određenom trenutku). U nastavku je navedeno nekoliko komercijalnih rješenja slične primjene s pregledom njihovih značajki i specifikacija.

*Manhattan 4K 3-Port HDMI Switch*, razdjelnik s tri ulaza i jednim izlazom koji omogućuje prijenos HDMI signala s rezolucijom 4K pri 30 Hz, podržava automatsko prebacivanje prilikom povezivanja novoga ulaza i ručno prebacivanje ulaza pomoću tipkala na razdjelniku. Također sadrži prijemnik za infracrveni daljinski upravljač za prebacivanje izvora te vlastito napajanje od 5 V [5]. Razdjelnik sličan ovome poslužio je kao osnova za nadogradnju i izradu sustava pametnog HDMI razdjelnika u sklopu ovog diplomskog rada.

*Extron DTP CrossPoint 84* (Sl. 2.4.), 8x4 4K *matrix* prezentacijski preklopnik s mogućnošću skaliranja širokog raspona video signala na stope izlaza visoke rezolucije. Može pratiti stanje i upravljati audiovizualnim uređajima koristeći standardnu Ethernet mrežu za lakšu daljinsku kontrolu i postavke sustava te dvosmjerni RS-232 i IR bez potrebe za dodatnim kablovima. Također pruža mogućnost daljinskog upravljanja putem web preglednika na raznim uređajima ili putem *Extron* sustava upravljanja [6].



Sl. 2.4. *Extron DTP CrossPoint 84*. [6]

Drugi uređaj je *DM-NVX-D30* proizvođača *Crestron Electronics, Inc.*, prijemnik i AV dekodeer preko IP-a (engl. *Internet Protocol*) dizajniran za napredne audiovizualne medije, podržava prijenos 4K videa pri 60 Hz s 4:4:4 chroma poduzorkovanjem preko standardnog gigabitnog Etherneta. Korisničko

sučelje i upravljanje postavkama dostupno je preko web preglednika te se uređajem može upravljati putem RS-232, IR signala i CEC-a. Podržava HDR (engl. *High dynamic range*) i HDCP 2.3 [7].

Primjer ugrađenog sustava za upravljanje ulaznim uređajima u sklopu televizora je *LG OLED evo C3* 4K pametni televizor. Sustav nudi poboljšano korisničko iskustvo svojim naprednim pametnim daljinskim upravljačem i značajkama upravljanja prebacivanja ulaznih signala. Sadrži ugrađeno inteligentno prepoznavanje glasa i kompatibilnost s raznim komercijalnim sustavima glasovnih naredbi za kontrolu televizora i povezanih uređaja. Uz pojednostavljenu navigaciju i primjenu značajki s umjetnom inteligencijom sustav omogućuje jednostavan pristup različitim pružateljima medijskih usluga te poboljšava iskustvo igranja videoigara s eARC (engl. *Enhanced Audio Return Channel*), VRR (engl. *Variable Refresh Rate*) i ALLM (engl. *Auto Low Latency Mode*) potporom za HDMI 2.1 [8].

Kompleksniji sustav koji se može prilagoditi za primjenu u specifičnim okruženjima je prijenos HDMI signala preko IP-a pomoću HDMI predajnika i prijemnika. Primjer pružatelja takvog sustava je *Just Add Power* [9], koji nudi širok izbor uređaja za prijenos audiovizualnog sadržaja preko IP-a. HDMI predajnici spajaju se na ulazne uređaje i pretvaraju HDMI signal s njih u IP pakete za prijenos preko standardnog Ethernet. Mrežni preklopnik služi za distribuciju IP signala, s mogućnošću varijacija prijena jedana na jedana, jedana na više ili više na više ulaznih signala na izlazne uređaje. Izlazni uređaji primaju signale pomoću HDMI prijemnika koji pretvaraju IP pakete u HDMI signal. Takvi sustavi pružaju skalabilnost i fleksibilnost primjene te mogu koristiti napredne sustave upravljanja, skaliranje rezolucije, audio integraciju itd.

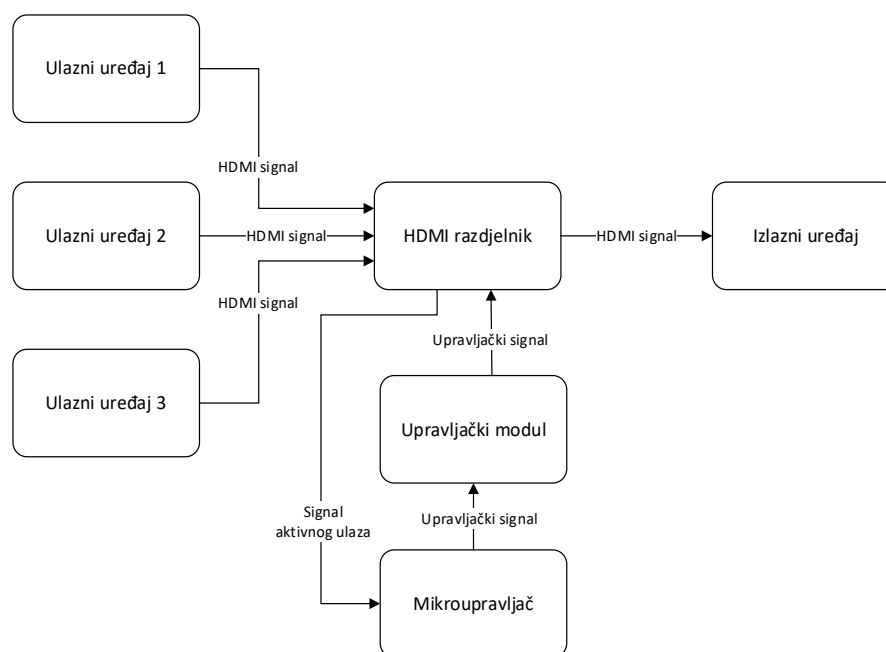
Većina navedenih primjera, osim prvoga HDMI razdjelnika, su složeniji uređaji viših razina cijene te su namijenjeni za napredne i poprilično skupe sustave prijena audiovizualnih sadržaja visoke kvalitete, postave kućnog kina, optimizaciju iskustva igranja videoigara i sl. Rješenje izrađeno u ovom diplomskom radu rješava neke od nedostataka običnog HDMI razdjelnika i koncentrira se na poboljšanja na jednostavnijoj razini s nižim financijskim ulaganjima. Realiziran je sustav pametnog HDMI razdjelnika s manjim adaptacijama sklopovlja i dodatnim opcijama udaljenog upravljanja orijentiranim prema naprednijoj softverskoj pozadini i mogućnostima.

### 3. REALIZACIJA PAMETNOG HDMI RAZDJELNIKA

Izvedba rješenja sastoji se od dva glavna dijela, to su sklopovlje i programski kod. U prijedlogu sustava i njegovoj analizi iznesena je okvirna ideja rješenja s njegovim glavnim komponentama i funkcionalnostima. U poglavljima koja slijede detaljnije je opisan svaki dio ovog sustava, pružajući sveobuhvatno razumijevanje njegova rada.

#### 3.1. Analiza predloženog sustava

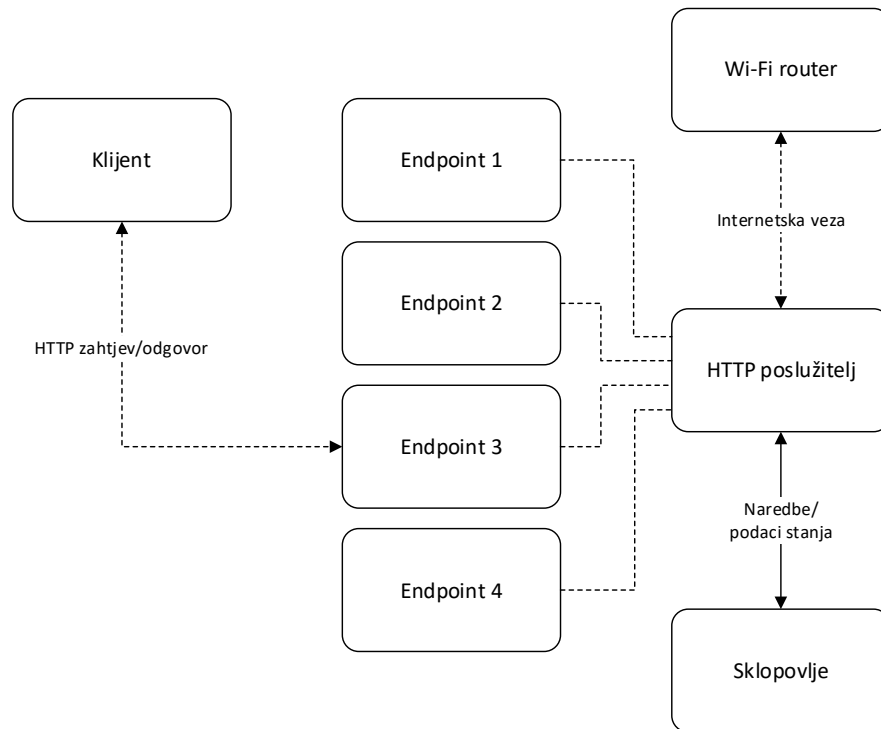
Predloženo rješenje izrađeno u sklopu ovog diplomskog rada zasnovano je na nadogradnji komercijalno dostupnog HDMI razdjelnika s tri ulazna priključka i ručnim odabirom ulaza pomoću tipkala. Sustav je nadograđen mikročipom koji podržava Wi-Fi bežičnu komunikaciju i omogućuje udaljeno upravljanje razdjelnikom preko sklopa za prebacivanje ulaza. Upravljanje razdjelnikom najpraktičnije je realizirati pomoću mikroupravljačke pločice s mogućnošću bežičnog pristupa internetu. Takve pločice također imaju jednostavno sučelje za razvoj i povezivanje preko prikladnih nožica, s USB priključkom za programiranje te ugrađenom regulacijom napajanja. Na slici 3.1. prikazan je blokovski dijagram prijedloga rješenja sklopovlja sustava. HDMI razdjelnik je gotovi proizvod komercijalno dostupan s osnovnim funkcionalnostima povezivanja većeg broja ulaznih uređaja i reprodukcije signala na izlaznom uređaju.



Sl. 3.1. Blokovski dijagram predloženog sklopovskog rješenja.

U prijedlogu rješenja prikazana su tri ulazna uređaja priključena na HDMI razdjelnik koji prosljeđuje signal s aktivnog ulaza na izlazni uređaj, najčešće televizor, monitor ili projektor. Mikroupravljač je središnji dio cjelokupnog sustava i služi za upravljanje izmjene aktivnog ulaza, na njemu se izvodi programski kod te u određenom trenutku preko GPIO (engl. *general-purpose input/output*) konektora aktivira upravljački modul. Upravljački modul odnosno sklop za prebacivanje ulaza djeluje kao aktuator tipkala HDMI razdjelnika te prebacuje na sljedeći dostupan ulazni uređaj.

Na funkcionalnoj razini sustav se sastoji od HTTP poslužitelja (engl. *server*) koji je postavljen na mikroupravljačkoj jedinici s mogućnošću bežičnog povezivanja na internet preko Wi-Fi routera. S druge strane server pruža API (engl. *Application Programming Interface*) u obliku nekoliko krajnjih točaka (engl. *endpoint*) na koje klijenti mogu slati zahtjeve za prebacivanje HDMI ulaza. Nakon primljenog zahtjeva mikroupravljač izvodi algoritam po kojemu se treba izvršiti određena naredba realizirana fizičkom vezom i signalom prema sklopovlju. Upravljanje HDMI ulazom preko sklopovlja sustava i HDMI razdjelnika također zahtjeva i povratne informacije o stanju aktivnog ulaza koje razdjelnik šalje mikroupravljaču. Na slici 3.2. prikazan je sustav s glavnim funkcionalnim dijelovima za pametno upravljanje HDMI razdjelnikom.

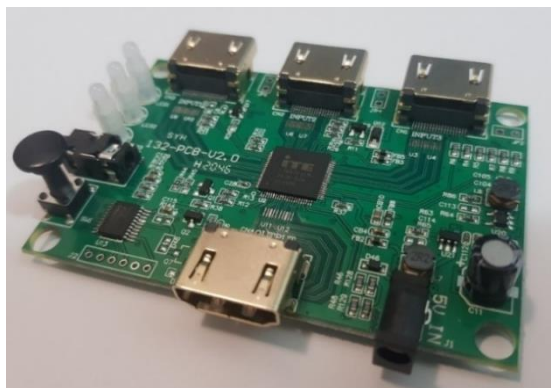


Sl. 3.2. Blokovski dijagram sustava na funkcionalnoj razini.

## 3.2. Sklopovlje sustava

Osnovna komponenta sklopovlja uključuje komercijalni 3u1 HDMI razdjelnik dok je srž inteligencije i automatizacije ovog sustava u WeMos D1 mini pločici s ESP8266 mikroupravljačem koji je ključan za procesiranje i upravljanje funkcionalnostima pametnog HDMI razdjelnika. Tipkalo razdjelnika integrirano je s 5 V relejom, povezujući ga u upravljani krug koji je normalno zatvoren. Ova konfiguracija pruža sučelje gdje se signali s HDMI razdjelnika izravno prenose na ESP8266 koji prima informacije o trenutno aktivnom ulazu. Sklopovsko rješenje sustava upotpunjeno je izravnim povezivanjem primarnog kruga releja s ESP8266 mikroupravljačem, integrirajući obje komponente sklopovlja, upravljačku i upravljaju.

**HDMI razdjelnik** korišten u realizaciji sustava je gotovi kupljeni uređaj s tri ulazna i jednim izlaznim priključkom te je u funkciji preklopnika (engl. *switch*) za prebacivanje HDMI signala s povezanih ulaznih uređaja (Sl. 3.3.). Uređaj ima funkciju automatskog prebacivanja na posljednji spojeni ulaz, sadrži tipkalo za prebacivanje na sljedeći dostupan ulaz i indikatore trenutno aktivnog ulaza. Razdjelnik podržava 4K rezoluciju sa stopom osvježavanja (engl. *refresh rate*) 60 Hz, ima mogućnost vlastitog napajanja od 5 V i priključak za spajanje prijemnika za IR daljinski upravljač.



Sl. 3.3. Prikaz sklopa korištenoga HDMI razdjelnika.

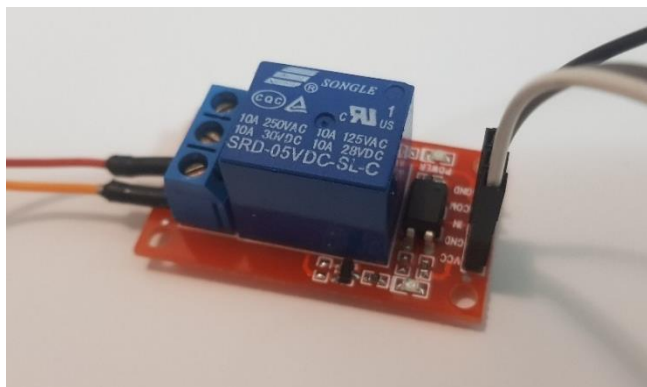
**ESP8266 WeMos D1 mini** pločica predstavlja mikroupravljačku jedinicu sustava, sastoji se od ESP8266 mikročipa koji podržava Wi-Fi bežičnu komunikaciju te omogućuje udaljeno upravljanje razdjelnikom preko sklopa za prebacivanje ulaza. Pločica također pruža mogućnost praktičnog povezivanja perifernih komponenata preko nožica, jednostavne je arhitekture s tipkalom za resetiranje te napajanjem od 3.3 V preko USB priključka. Korištena ESP8266 pločica je popularna zbog malih

dimenzija, niske cijene i mogućnosti jednostavnog udaljenog upravljanja uređajima u sustavima interneta stvari (engl. *Internet of Things*). Dodatna svrha ESP8266 mikročipa je programiranje HTTPS (engl. *Hypertext Transfer Protocol Secure*) poslužitelja na njemu što je ključno za udaljeno upravljanje putem računala ili mobilnog uređaja. WeMos D1 mini pločica prikazana je na slici 3.4.



Sl. 3.4. ESP8266 mikroupravljačka pločica (WeMos D1 mini). [10]

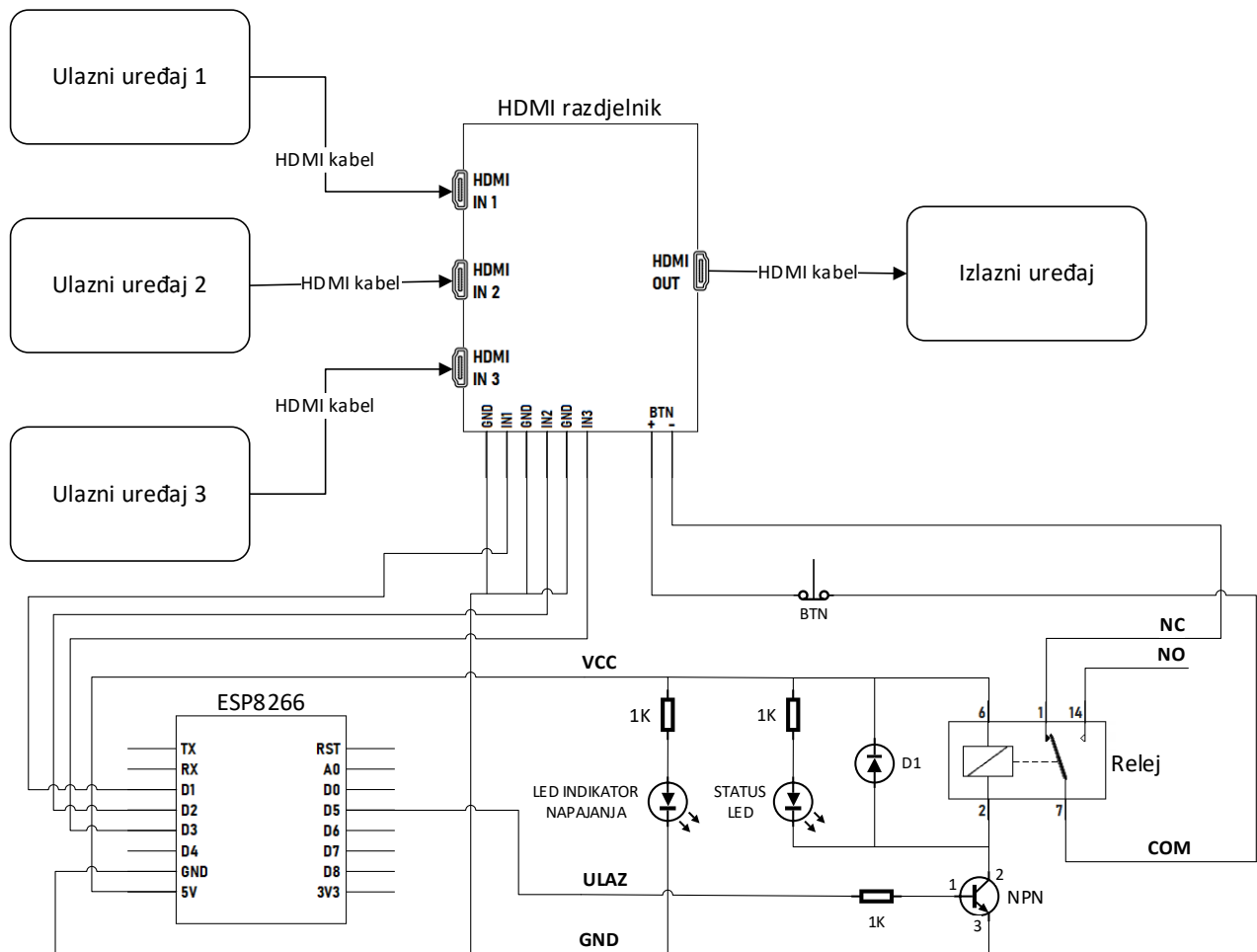
**Relej modul** korišten je radi mogućnosti odvajanja dvaju električnih krugova sustava. Tipkalo za odabir ulaznog signala na razdjelniku takvo je da se njegovim pritiskom prekida strujni krug (engl. *push-to-break button*), nakon otpuštanja tipkala razdjelnik prima signal iz ponovno zatvorenog strujnog kruga te prebacuje na ulaz sa sljedećeg uređaja ako je spojen i aktivan. Upravljanje takvim tipkalom tj. impulsom kojim se mijenja ulaz moguće je izvesti pomoću elektromagnetskog sklopnog releja. On ostvaruje upravljanje jednog strujnog kruga drugim, povezivanjem mikroupravljačke pločice na upravljačku stranu i ulaznim signalom s nje kontrolira se strujni krug upravljane strane. Upravljanu ili sekundarnu stranu moguće je upravljati na dva načina, normalno otvoreno (engl. *normally open*) i normalno zatvoreno (engl. *normally closed*). Upravljeni krug može biti u zatvorenom normalnom stanju kada kroz njega teče struja te se on prekida signalom s mikroupravljača, dok se kod normalno otvorenog kruga aktivacijom releja s upravljačke strane zatvara i krug sa sekundarne strane. Za sustav realiziran u sklopu ovog diplomskog rada, gdje se želi zadržati i fizičko tipkalo na uređaju i dodatno upravljati preko upravljača, potreban je normalno zatvoreni (NC) način rada te je sklopni relej s upravljačkim naponom od 5 V prikladno rješenje za upravljanje preko ESP8266 mikroupravljača. Slika 3.5. prikazuje relej u sklopu modula s dodatnim optičkim izolatorom.



**Sl. 3.5.** Sklopni relej modul 5 V.

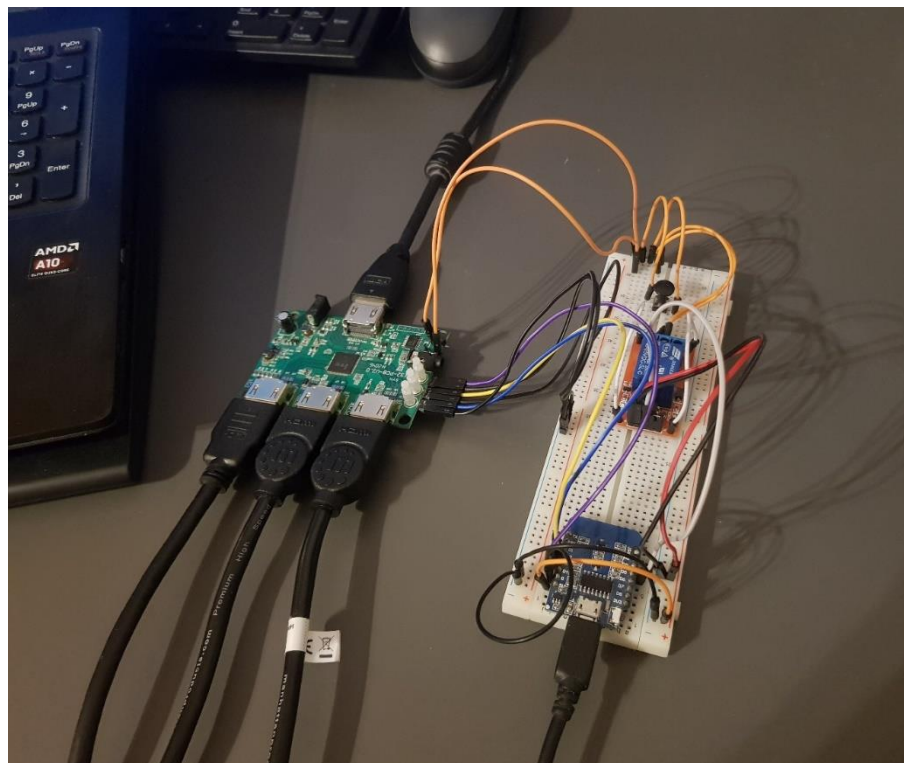
Detaljan shematski prikaz sklopovskog rješenja daje uvid u fizičke konekcije između elemenata sklopovlja (Sl. 3.6.). Upravljački dio sastoji se od ESP8266 i relej modula. Preko D5 GPIO nožice mikroupravljač šalje signal kojim aktivira relej, otvara i potom zatvara strujni krug upravljane strane čime djeluje kao tipkalo za prebacivanje ulaza. U upravljanom krugu koji je normalno zatvoren također se nalazi i fizičko *push-to-break* tipkalo koje i dalje služi svrsi ručnog prebacivanja. Relejni modul osim releja sadrži i dodatne komponente za signalizaciju stanja releja pomoću svjetlećih dioda te za zaštitu strujnih krugova. HDMI razdjelnik šalje signal na nožicu trenutno aktivnog ulaza te je povezan s ESP8266 preko GPIO nožica D1, D2 i D3 na koje prima signal potreban u obradi unutar algoritama upravljanja.





**Sl. 3.6.** Shematski prikaz sklopovskog rješenja.

Prikaz gotovog sklopovskog rješenja sustava uključujući pločicu HDMI razdjelnika, WeMos D1 mini mikroupravljačku pločicu, relej modul te povezane HDMI priključke spojenih uređaja nalazi se na slici 3.7.



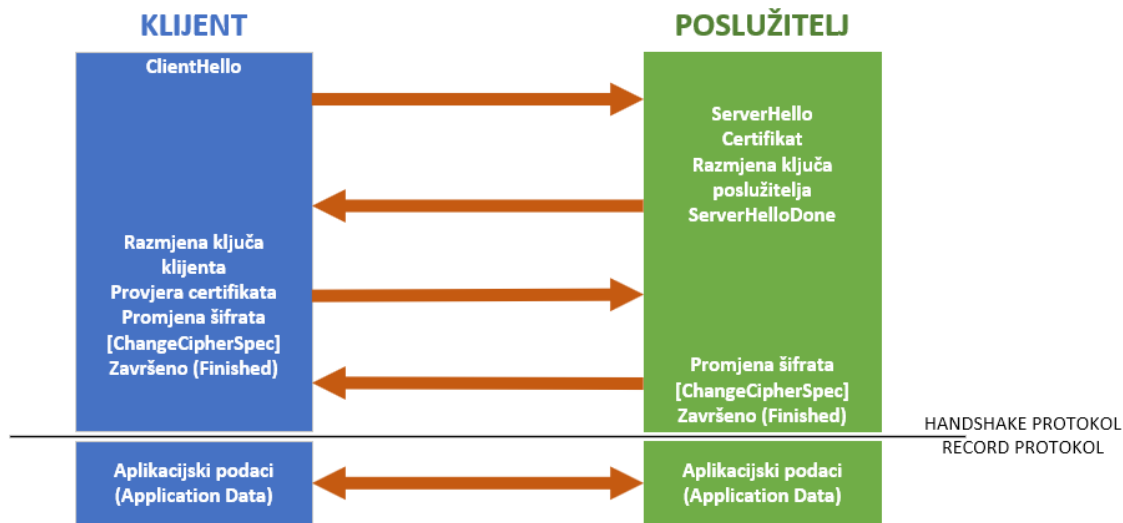
Sl. 3.7. Sklopovsko rješenje sustava.

### 3.3. Algoritmi upravljanja

Softverski dio sustava sastoji se od specijaliziranih biblioteka i funkcija s ciljem postizanja mogućnosti kontroliranja funkcije prebacivanja ulaza na HDMI razdjelniku. Prilikom izrade programskog rješenja korišteno je razvojno okruženje koje se sastoji od *Visual Studio Code* programa za uređivanje izvornog koda te većeg broja proširenja od kojih je najvažnije *Arduino* [11]. Ovo proširenje kao verzija *Arduino IDE* unutar *Visual Studio Code* platforme pogodno je za programiranje ESP8266 s dodatnim *esp8266* paketom za ove mikroupravljačke pločice. ESP8266 u ovom sustavu ne služi samo kao upravljačka jedinica, već i kao HTTPS poslužitelj koji sa SSL (engl. *Secure Sockets Layer*) certifikatom i privatnim ključem jamči sigurnost komunikacije.

HTTPS je protokol sličan HTTP-u ali sa SSL/TLS (engl. *Transport Layer Security*) enkripcijom podataka te javnim i privatnim ključem. SSL Certifikat je digitalna potvrda koja potvrđuje identitet web poslužitelja, a izdaje ga certifikacijsko tijelo, CA (engl. *Certification Authority*). Javni ključ dio je certifikata i dostupan je svima, dok je privatni ključ tajni i posjeduje ga samo poslužitelj. Klijent

preko web preglednika pristupa adresi poslužitelja koji potvrđuje svoj identitet pomoću certifikata. Nakon provjere (engl. *handshake*) SSL certifikata klijent kreira svoj tajni ključ i šifrira ga pomoću javnog ključa te ga dijeli s poslužiteljem (Sl. 3.8.). HTTPS poslužitelj dešifrira primljeni ključ pomoću svog privatnog ključa, na taj način stvaraju se ključevi sesije i ostvaruje se siguran kanal za komunikaciju između dviju strana [12]. U ovom programskom rješenju HTTPS poslužitelj je kreiran pomoću biblioteke *ESP8266WebServerSecure*.



Sl. 3.8. Razmjena poruka tijekom SSL handshake autentifikacije. [13]

Kako bi se osigurala operativna višestranost, ESP8266 je programiran za primanje POST zahtjeva klijenta na četiri različite krajnje točke. Te krajnje točke odgovaraju četirima različitim načinima rada (moda, engl. *mode*) s kojima HDMI razdjelnik može raditi. Ovakav modularni pristup nudi fleksibilnost, ostvarujući skalabilnost za potencijalna buduća poboljšanja i dodatne načine rada.

Na slici 3.9. prikazani su važniji dijelovi *setup* funkcije, ona predstavlja početne postavke programske skice te sadrži konfiguraciju GPIO nožica, inicijalizaciju serijske komunikacije, povezivanje na Wi-Fi pomoću *ESP8266WiFi* biblioteke, postavljanje NTP (engl. *Network Time Protocol*) vremena te pokretanje HTTPS poslužitelja.

```

1  pinMode(relayPin, OUTPUT);
2  pinMode(in1Pin, INPUT_PULLUP);
3  pinMode(in2Pin, INPUT_PULLUP);
4  pinMode(in3Pin, INPUT_PULLUP);
5  Serial.begin(115200); // Initialize serial communication
6
7  if (!connectToWifi()) {
8      delay(60000);
9      ESP.restart();
10 }
11
12 // Set up time
13 Serial.println("Setting up time...");
14 configTime(gmtOffset, daylightOffset, ntpServer, "time.nist.gov");
15
16 server.getServer().setRSACert(
17     new BearSSL::X509List(https_certificate),
18     new BearSSL::PrivateKey(https_private_key)
19 );
20 // Cache SSL sessions to accelerate the TLS handshake.
21 server.getServer().setCache(&serverCache);
22
23 // Start the server
24 Serial.println("Starting server...");
25 server.begin();
26 Serial.println("HTTPS server started");

```

**Sl. 3.9.** Konfiguracija setup funkcije.

Prije samog pokretanja poslužitelja u *setup* funkciji potrebno je također postaviti i rukovanje HTTP zahtjevima. Kada klijent pristupi početnom URL-u (engl. *Uniform Resource Locator*) odnosno web adresi poslužitelja, pritom šalje GET zahtjev te poslužitelj vraća zadani odgovor. Za pristup načinima rada i njihovim krajnjim točkama, klijent šalje POST zahtjev koji odgovarajući način rada obrađuje. Funkcija *readActiveInput* pomoću *digitalRead* funkcije čita vrijednosti GPIO nožica i zatim vraća broj trenutno aktivnoga ulaza (1, 2 ili 3) ili 0 u slučaju da nijedan ulaz nije aktivan (Sl. 3.10.).

```

1 void serverRouting()
2 {
3   server.on("/", HTTP_GET, [](()
4     { server.send(200, F("text/html"),
5       F("Welcome to the Smart HDMI switch Web Server")); });
6   server.on(F("/mod1"), HTTP_POST, mod1);
7   server.on(F("/mod2"), HTTP_POST, mod2);
8   server.on(F("/mod3"), HTTP_POST, mod3);
9   server.on(F("/mod4"), HTTP_POST, mod4);
10 }
11
12 int readActiveInput()
13 {
14   // Return the active input number (1, 2, or 3) or 0 if no input is active
15   val1 = digitalRead(in1Pin);
16   val2 = digitalRead(in2Pin);
17   val3 = digitalRead(in3Pin);
18
19   if (val1 == LOW){
20     delay(1000);
21     return 1;
22   }
23   if (val2 == LOW){
24     delay(1000);
25     return 2;
26   }
27   if (val3 == LOW){
28     delay(1000);
29     return 3;
30   }
31   else{
32     return 0;
33   }
34 }

```

**Sl. 3.10.** Funkcije za rukovanje HTTP zahtjevima i čitanje aktivnog ulaza.

Za uspješno prebacivanje ulaza potrebno je postaviti uvjete i uskladiti opcije prebacivanja s trenutnim stanjem aktivnog ulaza. Funkcija *handleSwitching* (Sl. 3.11.) prilikom odabira željenog ulaza na koji se treba prebaciti prima novi odabrani ulaz te ga uspoređuje s trenutnim aktivnim ulazom, ako je novi odabir različit od aktivnoga, izvodi se sekvenca prebacivanja ulaza i uspoređivanja s aktivnim dok se ne iscrpe moguće opcije. Ishodi pokušaja prebacivanja su uspješno prebacivanje na traženi ulaz ili ostanak odnosno vraćanje na inicijalni ulaz ako odabrani ulaz nije dostupan.

```

1 void switchInput()
2 {
3     // Switch the input
4     digitalWrite(relayPin, HIGH);
5     delay(timeDelay);
6     digitalWrite(relayPin, LOW);
7     delay(timeDelay);
8 }
9
10 void handleSwitching(int selectedInput) {
11     int initialInput = readActiveInput();
12     int attempts = 0;
13
14     while (readActiveInput() != selectedInput && attempts < 2) { // Restrict to two attempts
15         switchInput(); // Try to switch the input
16         attempts++;
17     }
18
19     // If we are not on the desired input and not on the initial input either, attempt to revert back
20     if (readActiveInput() != selectedInput && readActiveInput() != initialInput) {
21         switchInput();
22     }
23
24     if (readActiveInput() == selectedInput) {
25         Serial.print(F("IN"));
26         Serial.print(selectedInput);
27         Serial.print(F(" uključen!\r\n"));
28     } else if (readActiveInput() == initialInput) {
29         Serial.print(F("Nema spojenog uređaja na IN"));
30         Serial.print(selectedInput);
31         Serial.print(F(". Vraćen na početni ulaz IN"));
32         Serial.print(initialInput);
33         Serial.print(F(".\r\n"));
34     } else {
35         Serial.print(F("Nema spojenog uređaja na IN"));
36         Serial.print(selectedInput);
37         Serial.print(F(" i ne može se vratiti na početni ulaz.\r\n"));
38     }
39 }

```

Sl. 3.11. Funkcije za rukovanje prebacivanjem ulaza.

Upravljanje prebacivanjem potrebno je kod sva četiri načina rada. Za različite načine rada, klijent šalje različite vrijednosti na određenu krajnju točku. Svaka krajnja točka sastoji se od adrese web poslužitelja i jedinstvenog nastavka oblika „/modN“ za svaki od  $N$  načina rada. Zahtjevi koje klijent šalje su kreirani HTTP POST metodom koja je namijenjena za slanje podataka poslužitelju, omogućuje slanje većeg broja podataka te pruža sigurnost jer se podaci šalju unutar tijela zahtjeva. Načini rada ostvareni su izradom datoteka za svaki od njih te implementacijom *ArduinoJson* biblioteke [14]. Datoteke sadrže funkcije za obradu zahtjeva, primaju zahtjeve s tijelom u JSON (engl. *JavaScript Object Notation*) formatu i potom deserijaliziraju primljene podatke. Svi načini rada imaju sličnu formu, na početku je provjera sadržaja tijela zahtjeva i kreiranje dinamičkog JSON dokumenta

odgovarajuće veličine potrebnog za raščlanjivanje (engl. *parsing*) primljenih podataka. Nakon toga slijedi provjera greške prilikom deserijalizacije, pridjeljivanje sadržaja JSON dokumenta odgovarajućoj vrsti varijable (*JsonObject*, *JsonArray*) radi lakšeg pristupa te provjera metode zahtjeva. Glavni dio sastoji se od raščlanjivanja podataka zahtjeva i korištenja varijabli s primljenim vrijednostima s obzirom na uvjete prebacivanja HDMI ulaza. Na kraju se kreira odgovor, ovisno o uspješnosti primanja i obrade zahtjeva, koji poslužitelj šalje klijentu. Na slici 3.12. prikazan je prvi dio funkcije jednoga od načina rada, nakon njega slijedi specifični kod vezan za određeni način rada te na kraju odgovor poslužitelja oblika `server.send(201, F("application/json"), "{\\"status\\":\\"OK\\"}");`.

```
1 String postBody = server.arg("plain");
2 Serial.println(postBody);
3
4 DynamicJsonDocument doc(512);
5 DeserializationError error = deserializeJson(doc, postBody);
6 if (error)
7 {
8     // if the file didn't open, print an error:
9     Serial.print(F("Error parsing JSON "));
10    Serial.println(error.c_str());
11
12    String msg = error.c_str();
13    server.send(400, F("text/html"),
14              "Error in parsing json body! <br>" + msg);
15 }
16 else
17 {
18     JsonObject postObj = doc.as<JsonObject>();
19
20     Serial.print(F("HTTP Method: "));
21     Serial.println(server.method());
22
23     if (server.method() == HTTP_POST)
24     {
25         if (postObj.containsKey("key"))
26         {
27
28             Serial.println(F("done.\r\n"));
```

**Sl. 3.12.** Prvi dio funkcije za raščlanjivanje podataka zahtjeva.

Sustav razvijen u sklopu ovog diplomskog rada omogućuje četiri načina rada koja su odabrana s ciljem da se omoguće tri glavne opcije prebacivanja ulaza. Te opcije su: (i) manualni, trenutni odabir ulaza, (ii) postavljanje jednog ili više perioda između prebacivanja ulaza, (iii) zakazivanja događaja odnosno postavljanje vremenske oznake prebacivanja na određeni ulaz.

U tablici 3.1. opisani su načini rada te varijable s ispravnim vrijednostima koje se šalju u zahtjevu korisnika.

**Tab. 3.1. Načini rada**

<b>Br.</b>	<b>Način rada</b>	<b>Varijable</b>	<b>Vrijednosti varijabli</b>
1	trenutni odabir ulaza	broj ulaza	1, 2, 3
2	postavljanje jednog perioda prebacivanja	period	“HH:mm:ss”
3	postavljanje perioda prebacivanja za pojedine ulaze	do tri perioda	“HH:mm:ss”
4	postavljanje vremenske oznake prebacivanja za pojedine ulaze	do tri vremenske oznake  broj ulaza	“yyyy-MM-dd HH:mm:ss”  1, 2, 3

Prvi način rada služi trenutnom manualnom prebacivanju ulaza, u principu to je emulacija daljinskog upravljača s bilo kojeg uređaja s internet vezom. Klijent šalje zahtjev s brojem željenog ulaza te se on interpretira kao cjelobrojna vrijednost i predaje funkciji za rukovanje prebacivanjem.

Drugi način rada namijenjen je za postavljanje samo jednoga intervala prebacivanja ulaza. To znači da poslužitelj prima jednu vremensku oznaku formata „HH:mm:ss“ te ju pretvara u vrijednost u milisekundama i nakon toga pokreće mjerač vremena pomoću funkcije *millis*. Nakon što vremenski interval istekne, prebacuje se na sljedeći dostupan ulaz i tako se izvodi u *loop* funkciji unutar koje se pozivaju i sve ostale funkcije za rukovanje načinima rada. Na slici 3.13. prikazano je postavljanje mjerača vremena i rukovanje prebacivanjem nakon isteka zadanog vremenskog intervala.



```

1  unsigned long startTime = 0;
2  bool timerActive = false;
3
4  void setTimer()
5  {
6      startTime = millis();
7      timerActive = true;
8  }
9
10 void mod2Handler(unsigned long durationMillis)
11 {
12     // Check if the timer is active and the time period has elapsed
13     if (timerActive && (millis() - startTime >= durationMillis))
14     {
15         startTime = millis();
16         // Switch the input
17         switchInput();
18     }
19 }

```

**Sl. 3.13.** Funkcije mjerača vremena i provjere isteka intervala prebacivanja.

Treći način rada temelji se na sličnom principu, ali prima više varijabli s vremenima prebacivanja za pojedinačne ulaze. HTTP zahtjev klijenta sadrži do tri vremenska intervala u formatu „*HH:mm:ss*“ te se u obradi i raščlambi podataka koristi *JSONArray* referencu polja. Pomoću *for* petlji prolazi se kroz polje s intervalima prebacivanja tj. periodima trajanja pojedinih ulaza te se dohvaća vrijeme za svaki ulaz i dodjeljuje mu ga se. Dobivene vrijednosti se potom pretvaraju u milisekunde i predaju funkciji za rukovanje intervalima i prebacivanje.

Postavljanje specifičnog vremena prebacivanja ostvareno je četvrtim načinom rada uz pomoć *Time* biblioteke. Klijent šalje varijable broja ulaza te datumske i vremenske oznake određenog oblika (engl. *datetime format*). Moguće je jednim zahtjevom postaviti vrijeme prebacivanja za svaki ulaz što se ponovno rješava varijablom polja kao i u prethodnom načinu rada. Kako bi sustav znao koje je vrijeme i kada treba izvršiti prebacivanje koristi se dohvaćanje i postavljanje lokalnog vremena s NTP poslužitelja. Nakon uspostavljenе internetske veze, ESP8266 sinkronizira vrijeme s vremenom jednog od NTP poslužitelja, među kojima su poznatiji „*pool.ntp.org*“ i „*time.nist.gov*“. Algoritam ovog načina rada za zakazivanje događaja (engl. *event*) koristi vektor i stvara raspored prebacivanja uz pomoć strukture koja sadrži podatke o zakazanom događaju. Podaci o zakazanom događaju prebacivanja uključuju vremensku oznaku (proteklo vrijeme u sekundama od tzv. Unixove epohe) i broj ulaznog izvora. Nakon raščlanjivanja podataka, dohvaća se lokalno vrijeme kojemu se dodaje ili oduzima vrijednost u odnosu lokalne vremenske zone na UTC (engl. *Coordinated Universal Time*)

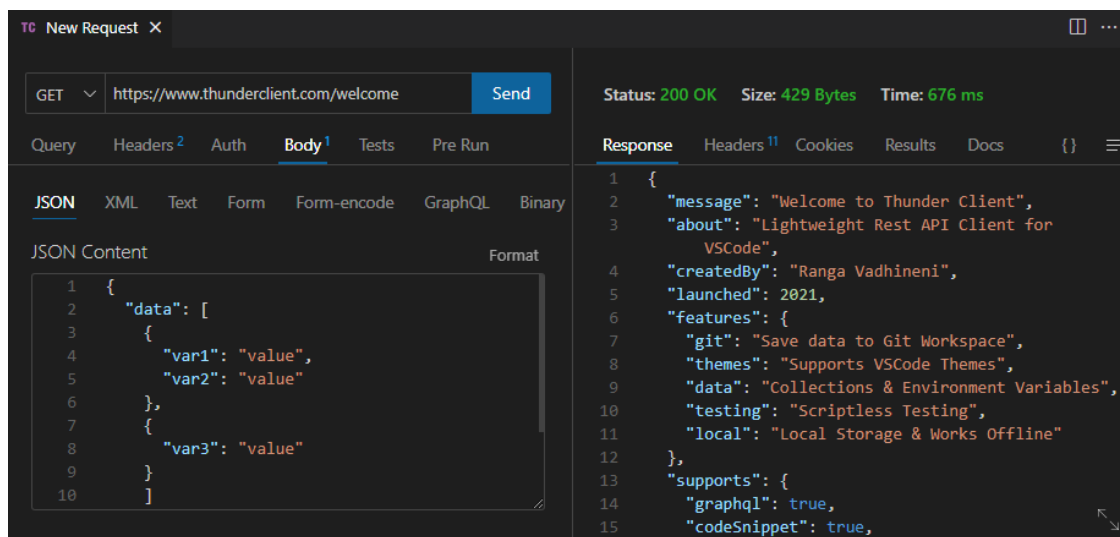
vrijeme nultog meridijana te s obzirom na ljetno računanje vremena. Slika 3.14. prikazuje funkciju za upravljanje rasporedom zakazanih događaja koja se poziva periodično unutar *loop* funkcije te vodi računa o vremenu izvođenja operacija prebacivanja.

```
1 struct SwitchEvent {
2     time_t timestamp;
3     int input;
4 };
5
6 std::vector<SwitchEvent> switchSchedule;
7 int nextSwitchIndex = 0;
8
9 void mod4Handler() {
10     if (switchSchedule.empty() || nextSwitchIndex >= switchSchedule.size()) {
11         return;
12     }
13
14     time_t nowUTC = time(nullptr);
15     time_t nowLocal = nowUTC + gmtoffset + daylightOffset;
16
17     if (nowLocal >= switchSchedule[nextSwitchIndex].timestamp) {
18         Serial.println("Current local time: " + String(nowLocal));
19         Serial.println("Next switch timestamp: " + String(switchSchedule[nextSwitchIndex].timestamp));
20         handleSwitching(switchSchedule[nextSwitchIndex].input);
21         nextSwitchIndex++;
22         if (nextSwitchIndex >= switchSchedule.size()) {
23             switchSchedule.clear();
24             nextSwitchIndex = 0;
25         }
26     }
27 }
```

Sl. 3.14. Dio programa za upravljanje rasporedom zakazanih događaja.

## 4. TESTIRANJE SUSTAVA

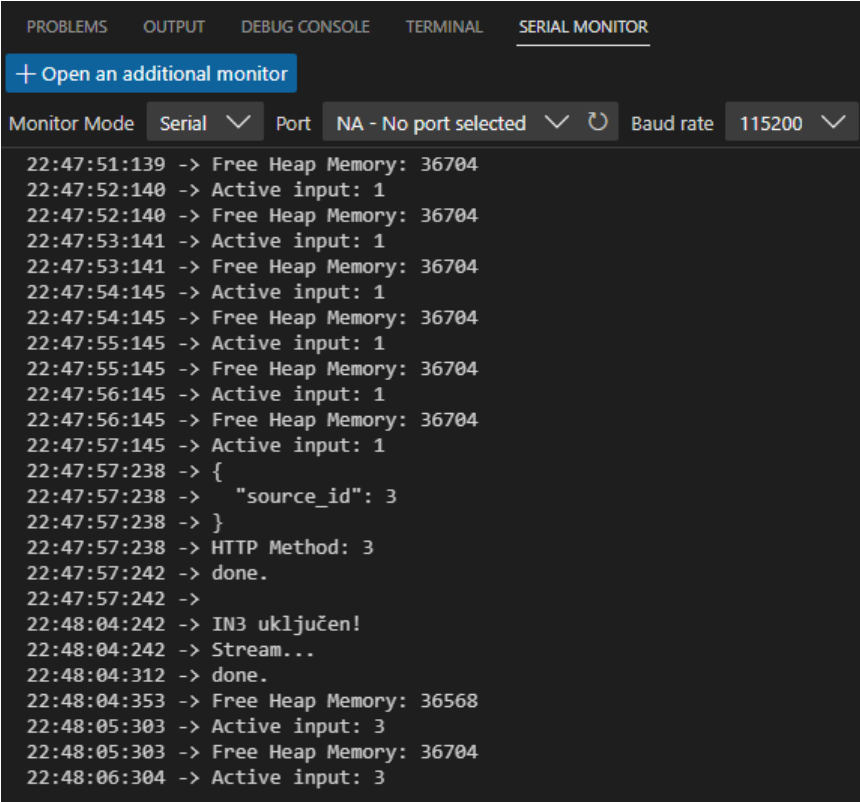
Prilikom testiranja sustava korišten je alat *Serial Monitor* u sklopu *Arduino* razvojnog okruženja, povezano sklopovlje sustava s računalima kao ulaznim uređajima i računalnim monitorom kao izlaznim uređajem. Mikroupravljačka pločica s ESP8266 integriranim sklopom programirana je preko USB kabla učitavanjem programske skice s računala. Za početak, testiranja su izvođena pomoću primitivnog koda slanjem vrijednosti preko serijskog monitora te jednostavnom simulacijom pritiska tipkala za prebacivanje ulaza. Svaki način rada zasebno je razvijen i testiran te nakon postavljanja HTTP poslužitelja prilagođen za primanje HTTP POST zahtjeva i dohvaćanje primljenih podataka u željenom obliku. Za slanje zahtjeva korištene su API platforme *Postman* i *Thunder Client* koji je praktičan i dostupan kao proširenje za *Visual Studio Code* (Sl. 4.1.). Ovi alati poslužili su kao sučelje za slanje zahtjeva na određenu krajnju točku poslužitelja s podacima unutar tijela zahtjeva u JSON formatu.



Sl. 4.1. Kreiranje zahtjeva pomoću Thunder Clienta.

Nakon uspostavljanja komunikacije i uspješnog slanja zahtjeva klijenta poslužitelju, potrebno je bilo prilagoditi algoritme u skladu s mogućnostima ESP8266 mikroupravljača, na primjer memorija, brzina izvođenja, kašnjenje te prema tome optimizirati kod. Nakon zasebnih testova načina rada potrebno je bilo uskladiti istovremenu obradu više zadataka bez preopterećenja *loop* funkcije. Složeniji načini rada kao što su postavljanje većeg broja varijabli za više ulaza i postavljanje

specifičnog vremena prebacivanja zahtijevali su izradu rasporeda sa zakazanim događajima čiju pravovremenu aktivaciju je potrebno pratiti. Također, prilikom testiranja javljao se određen broj iznimaka koje je trebalo uzeti u obzir, na primjer u situacijama kada nisu svi ulazi dostupni ili nema spojenih uređaja na ulaznim priključcima razdjelnika dolazilo je do pomaka u vrijednostima prilikom izvođenja algoritama prebacivanja. *Serial Monitor* poslužio je za pronalazak grešaka i nedostataka tj. debugiranje (engl. *debugging*) tijekom cijelog procesa razvoja. U dijelovima koda potrebno je ugraditi ispis poruka s određenim vrijednostima kako bi se znalo obrađuje li algoritam ispravne podatke i obavlja li posao kojemu je namijenjen. Također *Serial Monitor* je poslužio i kao dobra zamjena grafičkog sučelja za ispis trenutnog stanja sustava, ispis aktivnog ulaza, praćenje vremena i primljenih HTTP zahtjeva (Sl. 4.2.). Uz *esp8266* paket za *Arduino* razvojno okruženje dostupna značajka je i ispis poruka za debugiranje u stvarnom vremenu s više podataka i detalja koja je također bila od pomoći tijekom razvoja.

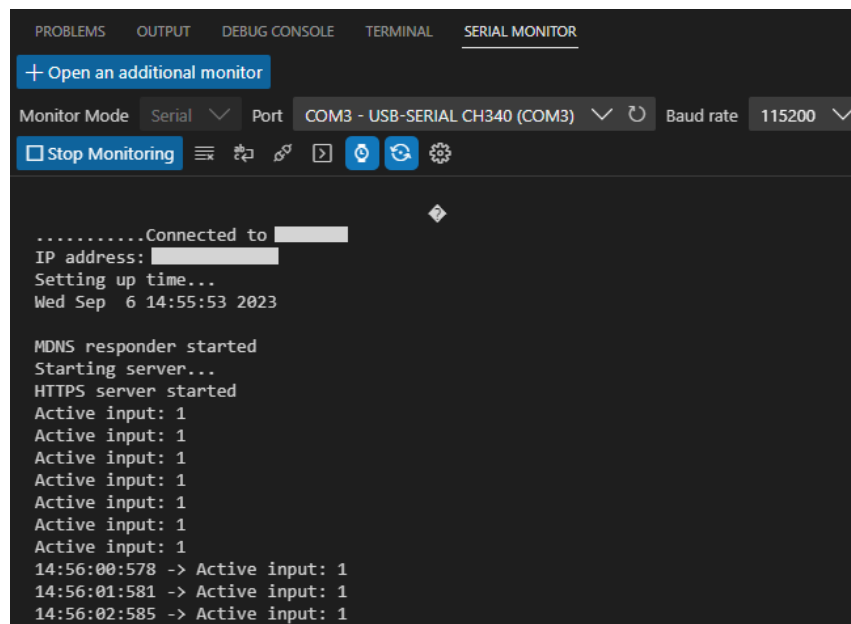


```
22:47:51:139 -> Free Heap Memory: 36704
22:47:52:140 -> Active input: 1
22:47:52:140 -> Free Heap Memory: 36704
22:47:53:141 -> Active input: 1
22:47:53:141 -> Free Heap Memory: 36704
22:47:54:145 -> Active input: 1
22:47:54:145 -> Free Heap Memory: 36704
22:47:55:145 -> Active input: 1
22:47:55:145 -> Free Heap Memory: 36704
22:47:56:145 -> Active input: 1
22:47:56:145 -> Free Heap Memory: 36704
22:47:57:145 -> Active input: 1
22:47:57:238 -> {
22:47:57:238 ->   "source_id": 3
22:47:57:238 -> }
22:47:57:238 -> HTTP Method: 3
22:47:57:242 -> done.
22:47:57:242 ->
22:48:04:242 -> IN3 uključen!
22:48:04:242 -> Stream...
22:48:04:312 -> done.
22:48:04:353 -> Free Heap Memory: 36568
22:48:05:303 -> Active input: 3
22:48:05:303 -> Free Heap Memory: 36704
22:48:06:304 -> Active input: 3
```

Sl. 4.2. Ispis poruka u *Serial Monitoru* prilikom testiranja rada sustava.

## 4.1. Rezultati testiranja

Početni uvjeti testiranja uključuju scenarij s tri ulazna uređaja povezana na HDMI razdjelnik te računalnim monitorom kao izlaznim uređajem. Uređaji su povezani s razdjelnikom preko HDMI kablova kategorije 3. ESP8266 mikroupravljač je napajan preko USB kabla dok je dio sklopa s HDMI razdjelnikom povezan bez vlastitog napajanja. Serijska komunikacija odvija se preko COM3 USB priključka s postavljenom brzinom 115200 bauda. Radnje prilikom uključivanja sustava su povezivanje ESP8266 na internet, postavljanje vremena i pokretanje HTTPS poslužitelja (Sl. 4.3.).

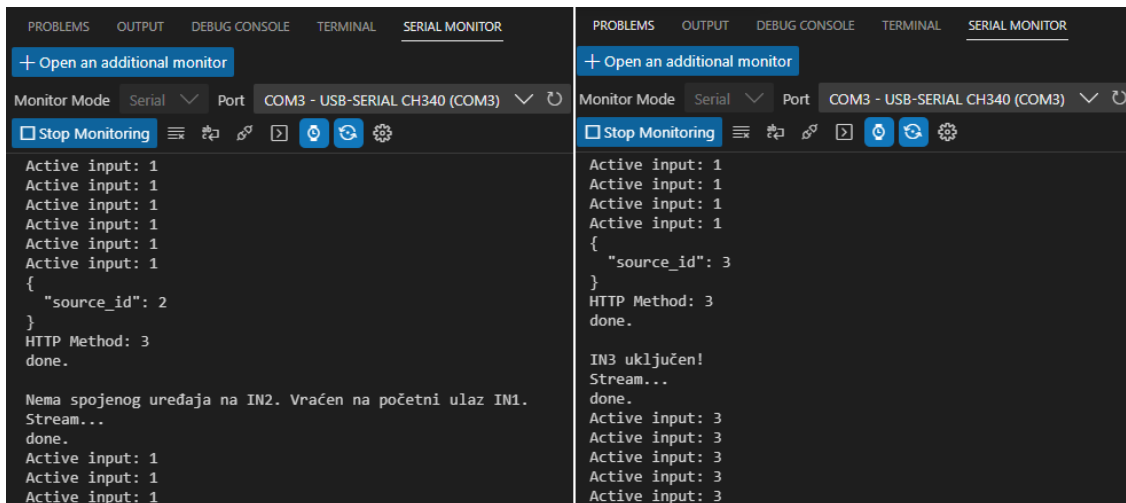


```
.....Connected to [redacted]
IP address: [redacted]
Setting up time...
Wed Sep 6 14:55:53 2023

MDNS responder started
Starting server...
HTTPS server started
Active input: 1
Active input: 1
Active input: 1
Active input: 1
Active input: 1
Active input: 1
Active input: 1
Active input: 1
14:56:00:578 -> Active input: 1
14:56:01:581 -> Active input: 1
14:56:02:585 -> Active input: 1
```

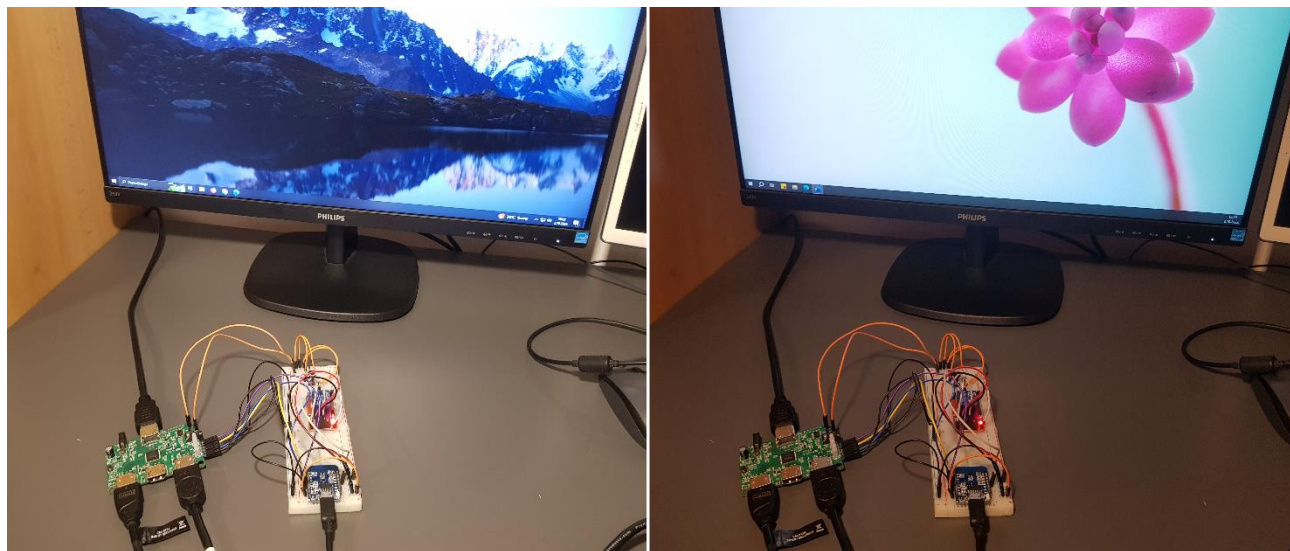
Sl. 4.3. Ispis u Serial Monitoru prilikom pokretanja ESP8266.

Prvi način rada koji predstavlja trenutno prebacivanje na odabrani ulaz testiran je u nekoliko scenarija od kojih su najbitniji onaj sa sva tri ulazna uređaja povezana i s barem jednim nepovezanim ulazom. Prebacivanje HDMI signala s ulaza izvršava se slanjem HTTP POST zahtjeva s brojem ulaza na krajnju točku prvog načina rada. Na slici 4.4. prikazan je ispis prilikom testiranja slučaja dva povezana ulazna uređaja na ulazima pod brojevima 1 i 3. Lijevo je prikazan pokušaj prebacivanja s aktivnog ulaza 1 na ulaz 2 na kojemu nema spojenog uređaja dok je desno prikaz ispisa uspješnog prebacivanja na ulaz 3.



**Sl. 4.4.** *Prebacivanje na nepovezan ulaz (lijevo) i uspješno prebacivanje (desno).*

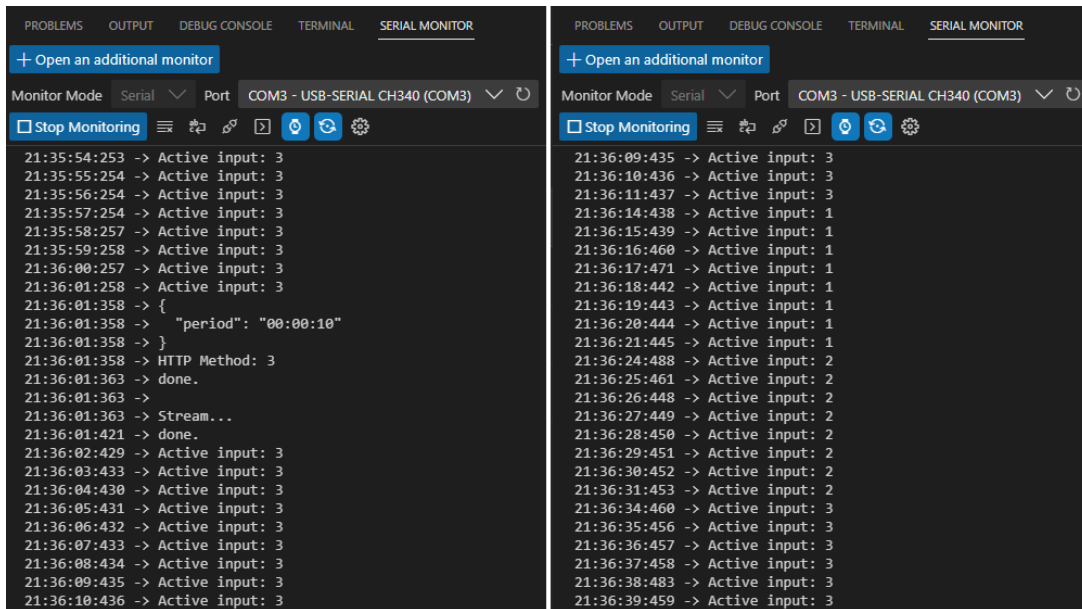
Na slici 4.5. prikazan je pametni HDMI razdjelnik i izlazni uređaj tj. računalni monitor prilikom prebacivanja na povezane HDMI ulaze, lijevo je prikaz HDMI signala s ulaznog uređaja povezanog na ulaz 1, a desno prikaz nakon prebacivanja na ulaz 3.



**Sl. 4.5.** *Monitor s prikazom HDMI signala povezanog ulaza 1 (lijevo) i ulaza 3 (desno).*

Drugi način rada uključuje postavljanje jednog perioda prebacivanja za sve povezane ulaze. Klijent šalje period prebacivanja koji poslužitelj dohvaća, pretvara u milisekunde i postavlja. U testnom primjeru zbog jednostavnosti se postavlja period prebacivanja od 10 sekundi. Na slici 4.6. lijevo je

prikazan ispis primljenog zahtjeva s periodom prebacivanja u primljenom formatu „00:00:10“ te desno ispis trenutno aktivnog ulaza koji se mijenja u intervalima primljene vrijednosti što se može vidjeti iz vremenskih oznaka.



Sl. 4.6. Ispis primljene vrijednosti perioda prebacivanja (lijevo) i prebacivanje ulaza (desno).

Treći načina rada je postavljanje posebnih perioda prebacivanja za pojedine ulaze. Klijent na krajnju točku trećeg načina rada šalje HTTP zahtjev s periodima trajanja za odabrane ulaze, nakon dohvaćanja vrijednosti sa strane poslužitelja, oni se pretvaraju u milisekunde te se postavljaju intervali prebacivanja. U primjeru testa na slici 4.7. povezana su dva ulazna uređaja na ulazima 1 i 3, dok je ulaz 2 slobodan. U testnom primjeru poslana su vrijednosti za sva tri ulaza ali se postavljaju vremena prebacivanja samo za ulaze sa spojenim uređajima. Desno na slici 4.7. može se vidjeti da se izvode prebacivanja na ulaze 1 i 3 s odgovarajućim vremenskim intervalima od 5 i 10 sekundi te se nepovezani ulaz 2 zanemaruje.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SERIAL MONITOR
+ Open an additional monitor
Monitor Mode Serial Port COM3 - USB-SERIAL CH340 (COM3)
Stop Monitoring
21:40:35:055 -> Active input: 3
21:40:36:056 -> Active input: 3
21:40:37:060 -> Active input: 3
21:40:38:058 -> Active input: 3
21:40:38:162 -> {
21:40:38:162 ->   "periods": [
21:40:38:162 ->     {
21:40:38:162 ->       "period1": "00:00:05"
21:40:38:165 ->     },
21:40:38:165 ->     {
21:40:38:168 ->       "period2": "00:00:07"
21:40:38:168 ->     },
21:40:38:170 ->     {
21:40:38:170 ->       "period3": "00:00:10"
21:40:38:173 ->     }
21:40:38:173 ->   ]
21:40:38:173 -> }
21:40:38:173 -> HTTP Method: 3
21:40:38:175 -> done.
21:40:38:175 ->
21:40:39:167 -> Stream...
21:40:39:227 -> done.
21:40:40:231 -> Active input: 3
21:40:41:233 -> Active input: 3
21:40:42:238 -> Active input: 3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SERIAL MONITOR
+ Open an additional monitor
Monitor Mode Serial Port COM3 - USB-SERIAL CH340 (COM3)
Stop Monitoring
21:40:38:175 ->
21:40:39:167 -> Stream...
21:40:39:227 -> done.
21:40:40:231 -> Active input: 3
21:40:41:233 -> Active input: 3
21:40:42:238 -> Active input: 3
21:40:43:234 -> Active input: 3
21:40:44:236 -> Active input: 3
21:40:45:251 -> Active input: 3
21:40:46:238 -> Active input: 3
21:40:47:239 -> Active input: 3
21:40:48:239 -> Active input: 3
21:40:49:241 -> Active input: 3
21:40:51:241 -> relay switched
21:40:53:242 -> Active input: 1
21:40:54:242 -> Active input: 1
21:40:55:241 -> Active input: 1
21:40:56:242 -> Active input: 1
21:40:57:243 -> Active input: 1
21:40:59:250 -> relay switched
21:41:01:244 -> Active input: 3
21:41:02:245 -> Active input: 3
21:41:03:247 -> Active input: 3
21:41:04:247 -> Active input: 3
21:41:05:248 -> Active input: 3

```

Sl. 4.7. Ispis primljenih vrijednosti (lijevo) i prebacivanje ulaza (desno).

Testiranje četvrtog načina rada odnosno postavljanje vremenske oznake prebacivanja za pojedini ulaz sastoji se od slanja HTTP zahtjeva na odgovarajuću adresu poslužitelja nakon čega se ispisuju zakazani događaji kao potvrda ispravnosti podataka i postavljanje rasporeda prebacivanja što je prikazano na slici 4.8.

```

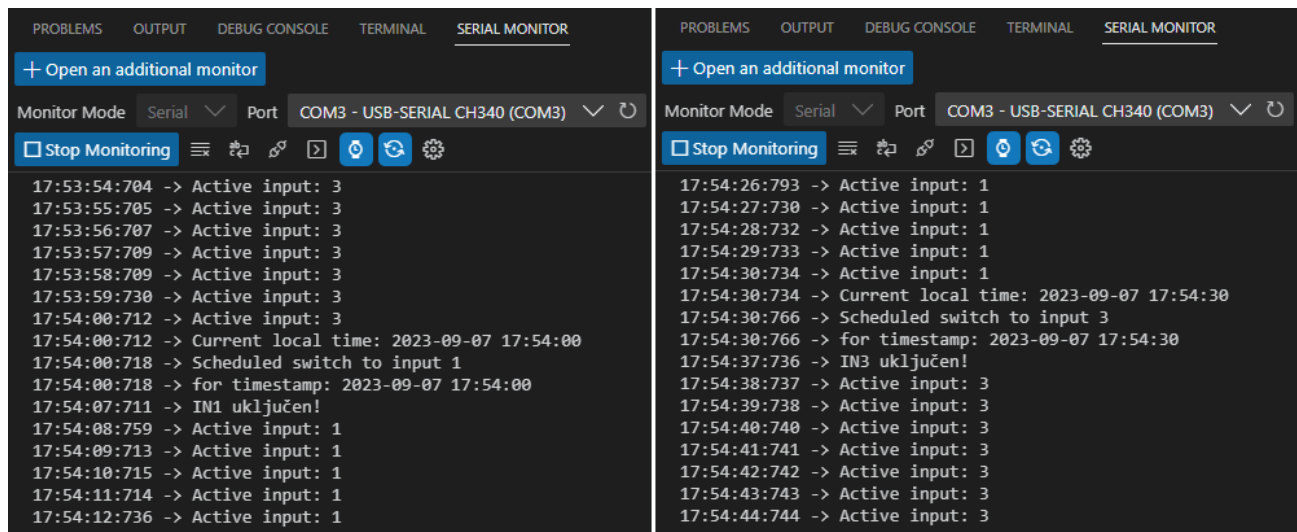
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SERIAL MONITOR
+ Open an additional monitor
Monitor Mode Serial Port COM3 - USB-SERIAL CH340 (COM3)
Stop Monitoring
17:53:22:922 -> Active input: 3
17:53:23:923 -> Active input: 3
17:53:24:925 -> Active input: 3
17:53:31:052 -> Active input: 3
17:53:31:072 -> Switch to input 1
17:53:31:072 -> scheduled for timestamp: 2023-09-07 17:54:00
17:53:31:078 -> Switch to input 3
17:53:31:078 -> scheduled for timestamp: 2023-09-07 17:54:30
17:53:32:678 -> Active input: 3
17:53:33:683 -> Active input: 3
17:53:34:700 -> Active input: 3
17:53:35:686 -> Active input: 3
17:53:36:686 -> Active input: 3
17:53:37:687 -> Active input: 3
17:53:38:688 -> Active input: 3
17:53:39:689 -> Active input: 3

```

Sl. 4.8. Postavljanje rasporeda prebacivanja ulaza.



U testnom primjeru poslana su vremenske oznake za sva tri ulaza s tim da je jedna od oznaka (za ulaz 2) prošla. HTTPS poslužitelj prima poslana vrijednosti i postavlja raspored zakazanih prebacivanja za spojene ulaze. Ulaz 2 se pritom preskače s obzirom na to da je vremenska oznaka koja mu je dodijeljena već protekla. Na slici 4.9. prikazan je ispis broja aktivnog ulaza te trenutnog lokalnog vremena i vremena prebacivanja na određeni ulaz u trenutku dostizanja njemu dodijeljene vremenske oznake. Za kraj je testiran rad više načina rada u isto vrijeme što je dalo pozitivne rezultate i pokazalo uspješnu integraciju svih dijelova koda u funkcionalnu cjelinu.



The image shows two side-by-side screenshots of a serial monitor application. Both screenshots show the same interface with a menu bar (PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, SERIAL MONITOR) and a toolbar with a 'Stop Monitoring' button and various icons. The main area displays a log of events. The left screenshot shows the initial state where input 3 is active. The right screenshot shows the state after a scheduled switch to input 1, with a subsequent switch back to input 3.

```
17:53:54:704 -> Active input: 3
17:53:55:705 -> Active input: 3
17:53:56:707 -> Active input: 3
17:53:57:709 -> Active input: 3
17:53:58:709 -> Active input: 3
17:53:59:730 -> Active input: 3
17:54:00:712 -> Active input: 3
17:54:00:712 -> Current local time: 2023-09-07 17:54:00
17:54:00:718 -> Scheduled switch to input 1
17:54:00:718 -> for timestamp: 2023-09-07 17:54:00
17:54:07:711 -> IN1 uključen!
17:54:08:759 -> Active input: 1
17:54:09:713 -> Active input: 1
17:54:10:715 -> Active input: 1
17:54:11:714 -> Active input: 1
17:54:12:736 -> Active input: 1

17:54:26:793 -> Active input: 1
17:54:27:730 -> Active input: 1
17:54:28:732 -> Active input: 1
17:54:29:733 -> Active input: 1
17:54:30:734 -> Active input: 1
17:54:30:734 -> Current local time: 2023-09-07 17:54:30
17:54:30:766 -> Scheduled switch to input 3
17:54:30:766 -> for timestamp: 2023-09-07 17:54:30
17:54:37:736 -> IN3 uključen!
17:54:38:737 -> Active input: 3
17:54:39:738 -> Active input: 3
17:54:40:740 -> Active input: 3
17:54:41:741 -> Active input: 3
17:54:42:742 -> Active input: 3
17:54:43:743 -> Active input: 3
17:54:44:744 -> Active input: 3
```

**Sl. 4.9.** Prebacivanje na odabrane ulaze u zakazana vremena.

## 5. ZAKLJUČAK

HDMI razdjelnik odnosno preklopnik omogućuje povezivanje različitih uređaja poput računala, igračih konzola, televizijskih prijemnika i drugih uređaja za reprodukciju medija na jedan izlazni uređaj bez potrebe za fizičkim prespajanjem kablova. Kroz analizu postojećih rješenja pronađen je prostor za poboljšanja osnovne funkcije HDMI razdjelnika koja se može proširiti i obogatiti bez potrebe za kupnjom kompleksnih uređaja s višestrukom svrhom.

U ovom diplomskom radu realiziran je sustav pametnog HDMI razdjelnika povezivanjem dodatnih elemenata sklopovlja i stvaranjem softverske podloge. Jedan od glavnih ciljeva bio je postizanje pametnog i udaljenog upravljanja tipkalom za prebacivanje ulaza te dodavanje novih funkcionalnosti i načina rada. Dodane komponente sklopovlja su ESP8266 mikroupravljačka pločica i 5 V relej modul kojim su odvojene upravljačka i upravljana strana sklopovlja. ESP8266 je središnji dio sustava i na njemu se izvode algoritmi upravljanja. HTTPS poslužitelj koji je postavljen na ESP8266 prima zahtjeve klijenta s podacima vezanima za četiri različita načina rada sustava. Ti načini rada uključuju funkcije trenutnog prebacivanja na odabrani ulaz razdjelnika, postavljanja perioda prebacivanja ulaza i postavljanja vremenske oznake prebacivanja.

Na razini sklopovlja, ESP8266 preko upravljačkog kruga kontrolira prebacivanje ulaza razdjelnika tako da signalom s GPIO nožice uključuje i isključuje relej. Upravljeni krug je normalno zatvoren te se on uključivanjem releja otvara, a isključivanjem ponovno zatvara. Tako kontrolirani relej modul djeluje kao tipkalo za prebacivanje ulaza. Razdjelnik je također izravno povezan s ESP8266 preko GPIO nožica na koje šalje signal trenutno aktivnoga ulaza. Tim signalom razdjelnik daje mikroupravljaču informaciju o trenutnom stanju sustava koja je potrebna u izvođenju algoritama za upravljanje prebacivanjem ulaza.

Na softverskoj razini, sustav funkcionira tako da poslužitelj od klijenta prima naredbe u obliku HTTP zahtjeva. Zahtjevi su kreirani POST metodom i sadrže vrijednosti broja ulaza ili perioda odnosno vremenske oznake za prebacivanje ulaza. HTTPS poslužitelj prima zahtjeve, raščlanjuje podatke u JSON formatu te ih obrađuje s obzirom na način rada kojemu su namijenjeni. Prvi način rada odnosi se na trenutno prebacivanje na ulaz poslan u zahtjevu. Drugi i treći način rada su postavljanje primljene vrijednosti jednoga odnosno više perioda prebacivanja ulaza. Posljednji četvrti način rada

obuhvaća primanje vremenske oznake za određeni ulaz i zakazivanje događaja prebacivanja za tu oznaku i ulaz.

Rezultati prilikom testiranja sustava su uspješno upravljanje HDMI razdjelnikom s četiri različita načina rada pomoću HTTPS poslužitelja preko bežičnog interneta. Testiranje uključuje sve scenarije stanja razdjelnika i spojenih uređaja; sa svim povezanim ulazima ili nepovezanim jednim ili više ulaza. Uzete su u obzir i iznimke prilikom slanja HTTP zahtjeva, s neispravnim metodama zahtjeva, neispravnim podacima, redosljedom zakazanih vremenskih oznaka i preskakanjem onih proteklih. Također, uspješno je omogućeno i istovremeno korištenje više načina rada.

Modularni pristup prilikom razvoja sustava ostavio je otvorenom mogućnost za dodatna poboljšanja i potencijalne nove načine rada. Njime se također nudi opcija dodavanja prilagođenih sučelja i web aplikacija za kontrolu, što donosi veću praktičnost i pristupačnost u budućnosti. U potencijalnoj nadogradnji trenutnog sustava također je moguća realizacija kompaktnijeg upravljačkog modula pomoću optičkih izolatora i tranzistora umjesto releja s komponentama sklopovlja povezanim na tiskanoj pločici te izrada odgovarajućeg kućišta sustava. U konačnici, ovaj pametni HDMI razdjelnik pokazuje kako jednostavne elektroničke komponente, kada su uparene s originalnim softverskim algoritmima, donose rješenja koja mogu potpuno izmijeniti korisničko iskustvo pružajući praktičan, pametan i siguran način upravljanja HDMI uređajima.

## LITERATURA

- [1] „HDMI cables – history and the story of the High Definition Multimedia Interface“, Pacroban, 2023., dostupno na: <https://www.pacroban.com/blogs/news/where-did-hdmi-originate> (pristupljeno 11. rujna 2023.)
- [2] „Does HDMI Support 4K? All You Need to Know“, Home Theater Academy, dostupno na: <https://hometheateracademy.com/does-hdmi-support-4k/> (pristupljeno 11. rujna 2023.)
- [3] „HDMI Explained“, Tripp Lite, Eaton, dostupno na: <https://tripplite.eaton.com/products/hdmi-cable-types> (pristupljeno 11. rujna 2023.)
- [4] „HDMI Splitter, Switch & Matrix - What are the Differences“, Electronic Notes, dostupno na: <https://www.electronics-notes.com/articles/audio-video/hdmi/switch-splitter-matrix-what-is-difference.php> (pristupljeno 11. rujna 2023.)
- [5] „4K 3-Port HDMI Switch“, Manhattan Products, dostupno na: <https://manhattanproducts.eu/products/manhattan-en-4k-3-port-hdmi-switch-207522> (pristupljeno 11. rujna 2023.)
- [6] „DTP CrossPoint 84 - 8x4 Scaling Presentation Matrix Switchers with DTP Extension“, Extron, dostupno na: <https://www.extron.com/product/dtpcp84ipcp#features> (pristupljeno 11. rujna 2023.)
- [7] „DM-NVX-D30 - DM NVX® 4K60 4:4:4 HDR Network AV Decoder“, Crestron, dostupno na: <https://www.crestron.com/Products/Video/DigitalMedia-Streaming-Solutions/Hardware-Encoders-Decoders/DM-NVX-D30> (pristupljeno 11. rujna 2023.)
- [8] „LG OLED evo C3 77 inch 4K Smart TV 2023“, LG, dostupno na: [https://www.lg.com/us/tvs/lg-oled77c3pua-oled-4k-tv#pdp\\_specs](https://www.lg.com/us/tvs/lg-oled77c3pua-oled-4k-tv#pdp_specs) (pristupljeno 11. rujna 2023.)
- [9] „Create a System - Building a system is simple“, Just Add Power, dostupno na: <https://www.justaddpower.com/build-a-system> (pristupljeno 11. rujna 2023.)
- [10] „WeMos D1 mini ESP8266 (V2.0)“, Grobotronics, dostupno na: <https://grobotronics.com/wemos-d1-mini-esp8266-v2.0.html?sl=en> (pristupljeno 11. rujna 2023.)

[11] „Arduino – Visual Studio Code extension for Arduino“, Microsoft, dostupno na: <https://marketplace.visualstudio.com/items?itemName=vsciot-vscode.vscode-arduino> (pristupljeno 11. rujna 2023.)

[12] „How does SSL work? | SSL certificates and TLS“, Cloudflare, dostupno na: <https://www.cloudflare.com/learning/ssl/how-does-ssl-work/> (pristupljeno 11. rujna 2023.)

[13] P. Kacherginsky, „TLS/SSL Protocol“, Medium, 2009., dostupno na: <https://iphelix.medium.com/tls-ssl-protocol-84a1ecef54bd> (pristupljeno 11. rujna 2023.)

[14] „ArduinoJson Assistant“, arduinojson.org, dostupno na: <https://arduinojson.org> (pristupljeno 11. rujna 2023.)

## Popis i opis upotrijebljenih oznaka i kratica

<b>Kratica</b>	<b>Puni naziv na engleskom jeziku</b>	<b>Prijevod na hrvatskom jeziku</b>
ALLM	Auto Low Latency Mode	Automatski način niske latencije
API	Application programming interface	Sučelje za programiranje aplikacija
AV	Audio/video	Audio/video
CA	Certification authority	Tijelo za izdavanje certifikata
CEC	Consumer Electronics Control	Kontrola potrošačke elektronike
DVI	Digital Visual Interface	Digitalno vizualno sučelje
eARC	Enhanced Audio Return Channel	Poboljšani audio povratni kanal
GPIO	General-purpose input/output	Ulaz/izlaz opće namjene
HDCP	High-bandwidth Digital Content Protection	Zaštita digit. sadržaja visoke propusnosti
HDMI	High-Definition Multimedia Interface	Multimedijsko sučelje visoke razlučivosti
HDR	High dynamic range	Visoki dinamički raspon
HDTV	High-definition television	Televizija visoke razlučivosti
HTTP	Hypertext Transfer Protocol	Protokol za prijenos hiperteksta
HTTPS	Hypertext Transfer Protocol Secure	Siguran protokol za prijenos hiperteksta
IP	Internet Protocol	Internetski protokol
IR	Infrared	Infracrveni
JSON	JavaScript Object Notation	Zapis JavaScript objekta
NC	Normally Closed	Normalno zatvoreno
NTP	Network Time Protocol	Mrežni vremenski protokol
SSL	Secure Sockets Layer	Sloj sigurnih krajnjih točaka
STB	Set-top box	Postavljena kutija/jedinica
TLS	Transport Layer Security	Sigurnost transportnog sloja
URL	Uniform Resource Locator	Uniformni/ujednačeni lokator sadržaja
USB	Universal Serial Bus	Univerzalna serijska sabirnica
UTC	Coordinated Universal Time	Koordinirano univerzalno vrijeme
VRR	Variable Refresh Rate	Promjenjiva stopa osvježavanja

## Sažetak

Diplomski rad „Pametni HDMI razdjelnik“ istražuje i implementira inovativno rješenje za poboljšanje upravljanja HDMI razdjelnikom odnosno preklopnikom. Temelji se na nadogradnji konvencionalnog uređaja s dodatkom ESP8266 mikroupravljača, omogućujući udaljeno upravljanje, zakazivanje događaja prebacivanja izvora i druge napredne funkcije. Kroz analizu postojećih rješenja, uočena su ograničenja standardnih razdjelnika i mogućnosti za unapređenje njihovih funkcija. Rad također uključuje koncepte HTTPS poslužitelja i SSL enkripcije, interneta stvari i upravljanja relej modulom te kako ih implementirati na HDMI razdjelniku. Sve to čini uređaj sigurnijim, praktičnijim i prilagodljivijim za različite potrebe korisnika.

**Ključne riječi:** HDMI razdjelnik, ESP8266, zakazivanje događaja, HTTPS poslužitelj, relej modul

## **Abstract**

### Smart HDMI switch

This master thesis titled "Smart HDMI switch" explores and implements an innovative solution for improved HDMI switch control. It is based on upgrading a conventional device with the addition of an ESP8266 microcontroller, enabling remote control, scheduling source switching events, and other advanced functions. By analyzing existing solutions, we can identify the limitations of standard switches and the potential for improving their functionality. The thesis also introduces concepts such as HTTPS server and SSL encryption, the Internet of Things, relay module-based control, and their implementation with an HDMI switch. All of these enhancements contribute to making the device more secure, practical, and adaptable for various user needs.

**Key words:** HDMI switch, ESP8266, scheduling, HTTPS server, relay module



## **Životopis**

Franjo Filipović rođen je 12. prosinca 1995. godine u Beču. Osnovnu školu završava u Vinkovcima kao i opći smjer gimnazije 2014. godine. Iste godine upisuje Elektrotehnički fakultet u Osijeku, današnji FERIT, gdje završava preddiplomski sveučilišni studij elektrotehnike, smjer Komunikacije i informatika. 2020. godine upisuje diplomski sveučilišni studij Automobilsko računarstvo i komunikacije.