

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

**Sveučilišni diplomski studij**

**SUSTAV ZA POSREDNO KORIŠTENJE ULAZNIH  
UREĐAJA**

**(eng. System for indirect use of input devices)**

**Diplomski rad**

**Luka Kruljac**

**Osijek, 2020**





## Sadržaj

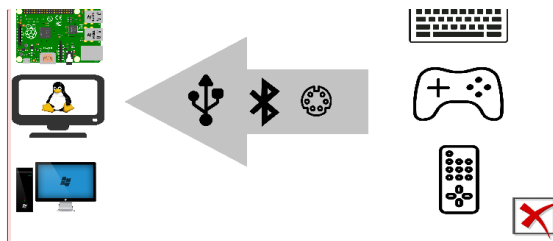
<b>1. UVOD .....</b>	<b>1</b>
<b>2. POSTOJEĆA RJEŠENJA.....</b>	<b>3</b>
2.1 Usporedba postojećih s novim rješenjem .....	4
<b>3. RAZVOJ APLIKACIJA NA POSREDNOM UREĐAJU.....</b>	<b>5</b>
3.1 WPF Programski okvir .....	6
3.2 MVVM Struktura .....	6
3.3 Komunikacijske veze.....	8
3.4 Uređaji .....	9
<b>4. UPRAVLJAČKI PROGRAM ZA SUSTAVE BAZIRANE NA UNIX-U .....</b>	<b>10</b>
4.1 Glavna funkcija programa .....	10
4.2 Parsiranje argumenata.....	10
4.3 Dretva za osluškivanje.....	10
4.4 Dretva za izvršavanje.....	10
4.5 Korištenje.....	10
<b>5. DEMO PROJEKT.....</b>	<b>11</b>
5.1 Hardware .....	11
5.2 Software.....	11
<b>6. TESTIRANJE.....</b>	<b>11</b>
<b>ZAKLJUČAK.....</b>	<b>12</b>
<b>LITERATURA .....</b>	<b>13</b>
<b>POPIS I OPIS UPOTRJEBLJENIH KRATICA .....</b>	<b>14</b>
<b>SAŽETAK.....</b>	<b>15</b>
Ključne riječi .....	15
<b>ABSTRACT .....</b>	<b>15</b>

Key words .....	15
<b>ŽIVOTOPIS.....</b>	<b>16</b>
<b>PRILOZI.....</b>	<b>17</b>
<b>ELEKTRONIČKA VERZIJA RADA.....</b>	<b>18</b>

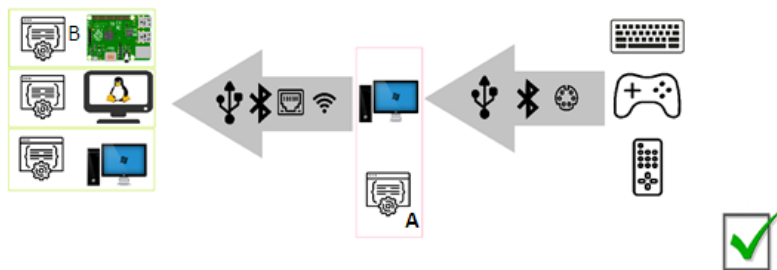
## 1. Uvod

Zadatak ovog diplomskog rada je razviti programsku podršku (eng. software) za posredno korištenje ulaznih uređaja (eng. using input devices indirectly). Software obuhvaća čitavo rješenje (eng. solution) u obliku više projekata koji u cjelini omogućuju korištenje određenog ulaznog uređaja na ciljanom uređaju (eng. target device) bez da su uređaji direktno spojeni. Veza između njih uspostavljena je preko posrednog uređaja koje je sa oba spomenuta uređaja direktno povezan nekom vezom.

Za uspješan rad ovakvog koncepta potrebno je izraditi upravljački software (eng. driver) na strani ciljanog uređaja, odnosno više njih ovisno o platformama ciljanih uređaja koje se žele koristiti te aplikaciju koja se izvršava na posrednom uređaju, a koja za ulogu ima uspostavljenje komunikacijskog puta između ulaznog i ciljanog uređaja odnosno pruža sučelja zaslonskih virtualnih uređaja. Slika 1.1 prikazuje uobičajen način povezivanja ulaznih uređaja PS2/USB/Bluetooth vezom sa ciljanim uređajima, međutim isti nije uvijek moguć te se zadatak ovog rada bavi načinom povezivanja koji je prikazan na Slika 1.2.



Slika 1.1 Uobičajen način povezivanja uređaja



Slika 1.2 Željeni način povezivanja uređaja

Commented [KL(DDCAV1]: Obije slike grafički obraditi

Da bi se razjasnilo uvođenje određene kompleksnosti iz prethodnog ulomka potrebno je shvatiti motive ovog projekta. Postoje situacije kada načini povezivanja sa Slika 1.1 nije moguć iz specifičnih razloga. Neki od njih su nabrojani:

➤ Integrirani ulazni uređaji

Ako se za primjer promatra prijenosno računalo, njegovu integriranu tipkovnicu nije moguće (barem ne na jednostavan način) od spojiti od prijenosnog računala te ju zatim spojiti i koristiti na nekom drugom uređaju.

➤ Ciljani uređaj je mobilan

Ukoliko je ciljani uređaj „na kotačima“, potrebno je pomjerati i ulazni uređaj. Za primjer uzeti učenički projekt automobila pokretan RPI računalom, a koji se upravlja tipkovnicom ili kontrolerom (eng. gampad).

➤ Ciljani uređaj je dislociran ili fizički nedostupan

Ponekad nije moguće spojiti ulazni uređaj jer smo udaljeni od njega, nemamo fizički pristup, ili ako imamo nemam mogućnost spajanja, primjerice, zauzeti su svi USB portovi

➤ Ulazni uređaj u fizičkom obliku nije dostupan

Aplikacija na posrednom računalu nudi i tzv. zaslonske uređaje, gdje kliktanje po sučelju simulira pritisak odgovarajuće tipke pa posjedovanje ulaznog uređaja nije nužno.

Ovaj rad obuhvaća cjelokupni proces razvoja sustava, od analize problema, usporede postojećih rješenja, prkeo razvoja do testiranja. Kroz drugo poglavlje predstavljene su postojeće tehnologije koje imaju sličnu ulogu kao i njihove prednosti i mane u odnosu na koncept sustava iz naslova ovog rada. U poglavljima tri i četiri opisani su postupci razvoja sustava, te funkcioniranje dvije njavažnije komponente sustava:

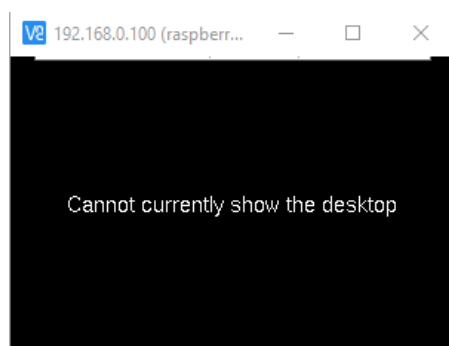
1. aplikacija na posrednom uređaju, obrađeno kroz treće poglavlje;
2. upravljački program za ciljanu platformu, obrađeno kroz četvrto poglavlje

Također, u sklopu rada pripremljen je i demo sustava čija izrada će bit predstavljena u poglavlju pet, dok su rezultati i analiza provedenog testa opisani u šestom poglavlju ovoga rada.

## 2. Postojeća rješenja

Projekt ovog rada po puno toga nije unikata, postoji mnogo rješenja koja obrađuju slične probleme, također za svaki problem nabrojan u poglavlju 1.2 postoji rješenje, no takva rješenja ne rješavaju sve nabrojene probleme u jednoj cjelini ili ih ne rješavaju na efikasan način, odnosno postavljaju dodatne preduvjete koji možda ne mogu biti ispunjeni u svakoj situaciji kada je potrebno upravljati ciljanom platformom.

Primjerice, VNC (eng. Virtual Network Computing) alat iz [2] objedinjuje sve funkcionalnosti koje su obrađene u ovom projektu, štoviše i puno više nego li je obrađeno ovim projektom. Međutim, ovo rješenje ima određena ograničenja po pitanju zahtijeva te ne taj način u određenim situacijama ne može parirati projektu iz ovog rada. Jedan od nedostataka je što ciljani uređaj mora imati spojen ili virtualno montirani display vidljivo na Slika 2.1, osim toga ovo podrazumijeva isključivo TCP/IP vezu između ciljanog i posrednog uređaja.



**Slika 2.1**VNC pogreška pri pokušaju spajanja na sustava bez definiranog displaya

Osim VNC tu je i SSH (eng. Secure shell) rješenje iz [3] koji također pruža puno više mogućnosti i ima puno širu primjenu, međutim neka ograničenja iz ovog projekta u određenim situacijama daju prednosti projektu iz naslova, štoviše isključivo s SSH protokolom neke od problema iz 1.2 nije niti moguće riješiti. Ograničenje je postavljeno na vezu, SSH za vezu između uređaja podržava TCP/IP i UART. Veći problem ovog alata je što nije moguće koristiti uređaje poput gamepada, barem ne analogni dio, a i pritisak više tipki istodobno nije podržan. Nadalje, aplikacija na strani ciljanog uređaja mora voditi računa o obradi signala s obzirom da SSH proslijeđuje ulaze samo u otvorenom terminalu, a ne čitavom operacijskom sustavu.



## 2.1 Usporedba postojećih s novim rješenjem

Commented [KL(DDCAV2]: TODO

TODO

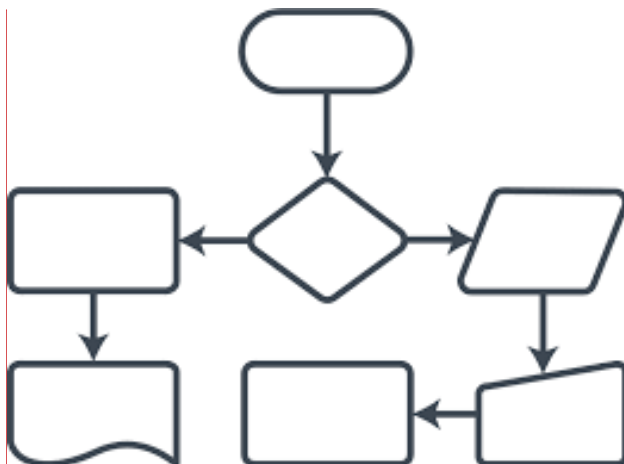
Nije potreban display – VNC;Nije potreban dodatan hardware;Omogućuje veći domet koristeći TCP/IP;I dalje je potreban SSH – minimalno jedan puta;Smanjeno korištenje resurca;Za razliku od SSH realtime, multibutton;Mogućnost proširenja na bilo koji uređaj

### 3. Razvoj aplikacija na posrednom uređaju

Pogledom na Sliku 1.2 vidljivo je da u tako postavljenom sustavu posredni uređaj zahtijeva software, komponenta označena slovom A, koji će omogućiti uređaju da postane posrednik između ulaznog uređaja i ciljane platforme. Da bi software odradio svoju ulogu postoji nekoliko zahtijeva među kojima su tri ključna:

1. Oslušivanje strujanja (eng. stream) ulaznih uređaja
2. Osiguranje komunikacijskog kanala s ciljanom platformom
3. Generiranje i slanje poruke kroz komunikacijski kanal

Osim tri ključna zahtijeva, zadani su zahtijevi poput filtriranja određenih uređaja te omogućavanje zaslonskih virtualnih uređaja. Zahtijevi ove komponente kao i njihova međusobna interakcija prikazani su na UML dijagramu na slici 3.1.



Slika 3.1 UML dijagram aplikacije

U sklopu ovog prototipa posredni uređaji ograničeni su na uređaje s windows operacijskim sustavom. S obzirom da windows operacijski sustav omogućava grafičko korisničko sučelje (eng. GUI) te uz činjenicu da velika većina korisnika preferira GUI aplikacije u usporedbi s CLI (eng. Comman Line Interface) aplikacijama, logičan izbor za tip ove komponente, odnosno aplikacije koja očekuje interakciju s korisnicima je windows desktop aplikacija.

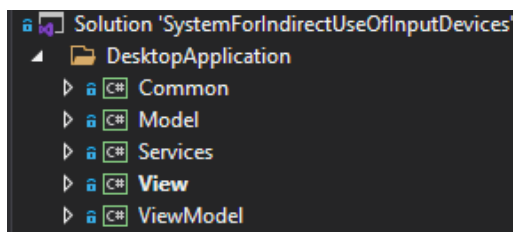
**Commented [KL(DDCAV3)]:** Pripremiti dijagram aplikacije u MS Visio ili sličnom alatu. Box koji predstavlja filtriranje, generiranje poruka, uspostavu linka i sl, te s ulaznim odnosno izlaznim streamom

### 3.1 WPF Programski okvir

Za izradu windows desktop aplikacije u ovom projektu korišten je programski okvir (eng. framework) otvorenog koda (eng. open source) WPF (eng. Windows Presentation Foundation). Ovaj framework temeljen je na vektorskom prikazu koji uz široki set značajki pomoću XAML jezika (eng. Extensible Application Markup Language) omogućuje poprilično jednostavnu izgradnju GUI-a neovisno o rezoluciji, te popratnog pozadinskog koda (eng. backend code) u jezicima iz .NET okruženja (C#, F#, VB). Neke od ključnih značajki WPF jesu da koristi DirectX, odnosno hardversko ubrzanje za iscrtavanje grafičkih elementa, omogućuje povezivanje podataka (eng. data binding) pomoću čega se lako odvajaju cijeline prikaza (eng. view) od cjelina podataka (eng. models). S obzirom da je Microsoft autor frameworka, Visual Studio je zasigurno najbolji IDE (eng. Integrated Development Environment) za razvoj WPF projekta te je on i korišten tijekom razvoja. Od programskih jezika, korišten je C# za backend kao i neke predkompajlirane (eng. precompiled) dinamičke biblioteke koje su razvijene C/C++ jezikom.

### 3.2 MVVM Struktura

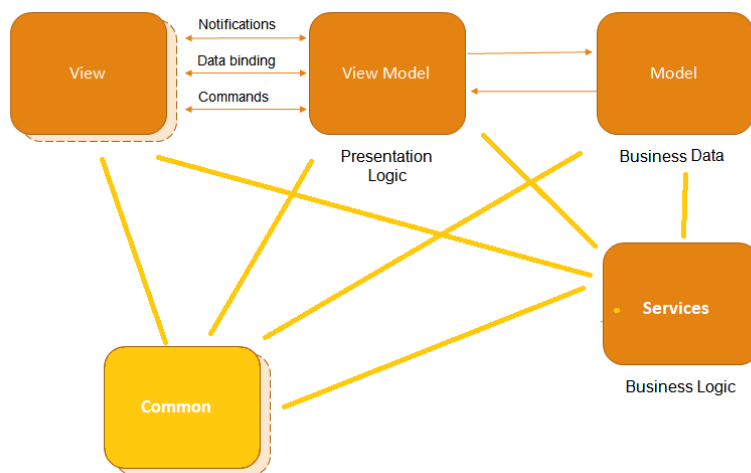
Pri razvoju aplikacije, da bi se odvojili programski blokovi ispravno je koristiti neki od --- (eng. design pattern), kao što su MVC, MVVM, MVP itd.. Korištenje design patterna olakšava i ubzava sam proces razvoja, omogućuje lakše testiranje i pojednostavljuje potencijalne izmjene u idućim nadogradnjama. Za ovaj projekt odabrana je MVVM struktura te su uz osnovne MVVM komponente dodane i komponente servisa (eng. Services) te zajedničke komponente (eng. Common). Ovaj set predstavlja čitavo rješenje (eng. Solution) aplikacije, prikazano na Slika 3.3, dok je na Slika 3.3. Komponente Common, Model, Services i ViewModel predstavljaju .NET Framework Class Library projekte čiji produkti su datoteke tipa .dll, dok je View komponenta Windows aplikacija na dnu hijerarhije čiji je produkt izvršna .exe datoteka.



Slika 3.2 Solution tree **view**

Commented [KL(DDCAV4)]: Ne znam kako ovo prevesti na hrv

Commented [KL(DDCAV5)]: Prijevod



Slika 3.3 Hijerarhijski prikaz projekata

#### Common

Ovaj projekt sadrži dvije klase koje omogućuju korištenje MVVM strukture te korištenje data bindinga. BaseView.cs je apstraktna klasa koju nasljeđuju sve klase modela. Nasljeđivanjem ove klase omogućuje se izvršavanje **event handlera** koji će reagirati na promjene vrijednosti odnosno bit će pozvan u Set metodi svojstava koji se prikazuju na View strani. Ovaj mehanizam omogućuje da svaka promjena svojstava u pozadinskom kodu bude automatski promijenjena i u GUI-u. Ovaj mehanizam prikazan je na DelegateCommand.cs definira objekt korišten unutar onih ViewModela čiji View sadrži neke komande te tako omogućuje povezivanje određene funkcije iz pozadinskog koda s određenom interaktivnom GUI komponentom. Sama datoteka sadrži dvije različite klase koje se razlikuju po tipu odnosno omogućuju povezivanje funkcija void tipa ili bilo kojeg generičkog tipa.

```

public string Status
{
    get { return _Status; }
    set { _Status = value; RaisePropertyChanged("Status"); }
}

```

Kod 3.2 Upotreba metode u model klasi

Kod 3.1 BaseViewModel klasa

Commented [KL(DDCAV6)]: Bolji grafički prikaz

Commented [LK7]: „Rukovoditelj događaja“....

Jednostavno smatram da neke termine nema ni smisla prevoditi

Model

ViewModel

**View**

Bla bla

### 3.3 Uredaji

#### 4. Upravljački program za sustave bazirane na UNIX-u

## 4.1 Glavna funkcija programa

## 4.2 Parsiranje argumenata

### 4.3 Dretva za osluškivanje

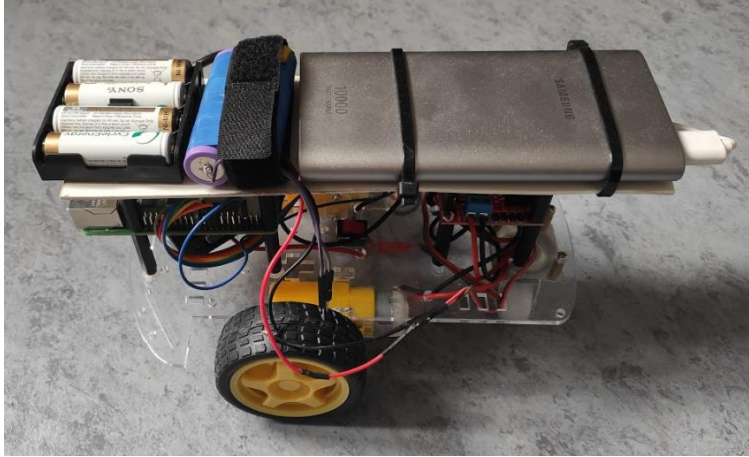
#### 4.4 Dretva za izvršavanje

## 4.5 Korištenje

[illegible]

## 5. Demo projekt

## 5.1 Hardware



## 5.2 Software

[illegible]

## 6. Testiranje

[illegible]



## Zaključak

Pri donošenju zaključka o ovome susustavu treba naglasiti kako je isti stvoren u domeni diplomskog rada jednog studenta te je produkt svega samo prototip s temlnim elementima pa prema tome sustav u ovoj fazi nije na razini spremnoj za komercijalnu upotrebu.

Tijekom izrade ovog rada vođeno je računo o arhitekturi softwarea na obje komponente, te su time postavljeni temelji za lakše unapređenje i ubacivanje novih pogodnosti kao što su novi načini komunikacije ili novi tipovi uređaji

**Commented [KL(DCAV8):** Mislim da je fer ograditi se od potencijalnih usporedbi ovog sustava s npr. VNC softwareom iza kojeg stoji povijest od skoro 20 godina i tvrtka od 100+ zaposlenih.

## Literatura

- [1] T. Matić, „Diplomski\_završni\_predložak“;  
dostupno na <https://loomen.carnet.hr/mod/resource/view.php?id=160448>,  
zadnja posjeta na 28.11.2020.
- [2] VNC User Guide  
dostupno na [https://www.realvnc.com/en/connect/downloads/VNC\\_User\\_Guide.pdf](https://www.realvnc.com/en/connect/downloads/VNC_User_Guide.pdf)  
zadnja posjeta na 28.11.2020.
- [3] SSH  
dostupno na <https://www.ssh.com/ssh/protocol/>  
zadnja posjeta na 28.11.2020.

## Popis i opis upotrjebljenih kratica

VNC	Virtual Network Computing
SSH	Secure Shell
RPI	Raspberry Pi
TCP/IP	Transimision Control Protocol/Internet Protocol
WPF	Windows Presnetation Foundation
MS	Microsoft
VS	Visaul Studio
CLI	Comand Line Interface
GUI	Graphical User Interface
XAML	Extensible Application Markup Language
IDE	Integrated Development Environment
MVC	Model View Controller
MVVM	Model View ViewModel
MVP	Model View Presenter

## Sažetak

[illegible]

## Ključne riječi

Udaljeno upravljanje;RPI; Ulazni uređaji; Uinput biblioteka

## Abstract

[illegible]

### Key words

## Remote control; RPI; Input devices; Uninput library

## Životopis

Luka Kruljac rođen je 25.3.1997. u Đakovu. Odrastao u Gašincima gdje je i pohađao Osnovnu školu J.A.Ćolnća od 1. do 4. razreda. Osnovnu školu od 5. do 8. razreda pohađa u istoimenoj školi u Satnici Đakovačkoj. Godine 2010. upisuje prirodoslovno-matematičku gimnaziju A.G.Matoš u Đakovu, istu završava 2015. godine kada upisuje program vojnog kadeta, smjer Vojno inženjerstvo. Program kadeta napušta iz osobnih razloga te u listopadu 2015. godine upisuje sveučilišni preddiplomski studij elektrotehnike na Elektrotehničkom fakultet Osijek. Za vrijeme preddiplomskog studija 2016. i 2017. odraduje prakse u tvrtki Siemens Convergence Creators, a 2018. u tvrtki Inchoo. Nakon završenog preddiplomskog studija komunikacija i informacijskih tehnologija upisuje diplomski studij Automobilskog računarstva i komunikacija na istom fakultetu. Tijekom studija bio je član, a zatim i predsjednik Studentskog Zbora, te član studentskog ogranka IEEE Osijek, gdje također u jednom periodu obnaša neke od dužnosti uprave ogranka. Sudionik je studentskog Work&Travel programa u SAD-u tijekom ljeta 2019., a po za povratku se programa u jesen iste godine počinje raditi kao stipendist u tvrtki RT-RK koju napušta nakon gotovo godinu dana, zatim, od jeseni 2020. godine sudjeluje na Erasmus Internship programu u tvrtki Infineon Technologies u Austriji gdje i danas radi kao software & component validation engineer.

U Osijeku, 11.9.2021.

Luka Kruljac



## Prilozi

1. Klasni dijagram čitavog rješenja

## Elektronička verzija rada

1. Na linku <https://github.com/lkruljac/diplomskiV2> može se pronaći izvorni kod čitavog projekta
2. Na linku <https://github.com/lkruljac/diplomskiV2/tree/master/Diplomski> može se pronaći elektronička verzija ovog dokumenta
3. Na linku  
[https://drive.google.com/drive/folders/19gSh3u\\_O7lwJ6STntR2eCAQP0J6EXJQw](https://drive.google.com/drive/folders/19gSh3u_O7lwJ6STntR2eCAQP0J6EXJQw) može se pronaći demo prikazi sustava