Realizacija sustava upravljanja pametnom kućom primjenom programibilnog logičkog kontrolera (PLC-a)

Šimunec, Domagoj

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:756343

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Domagoj Šimunec

Zagreb, 2021

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor: Student:

Izv. prof. dr. sc. Danijel Pavković, dipl. ing. Domagoj Šimunec

Zagreb, 2021

Izjavljujem da sam ovaj rad radio samostalno koristeći znanja i vještine stečene tijekom studiranja i prijašnjeg obrazovanja uz pomoć navedenih literatura.

Zahvaljujem se profesoru i mentoru dr.sc. Danijelu Pavkoviću i komentoru dr.sc. Matiji Krznaru na izvanrednoj kolegijalnosti te na svoj pruženoj pomoći.

Također se zahvaljujem svojim bližnjima i prijateljima na pruženoj podršci tijekom studiranja.

Domagoj Šimunec

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: **602 - 04 / 21 - 6 / 1**	
Ur.broj: **15 - 1703 - 21 -**	

ZAVRŠNI ZADATAK

Student: **Domagoj Šimunec** Mat. br.: 0035210183

Naslov rada na hrvatskom jeziku: **Realizacija sustava upravljanja pametnom kućom primjenom programibilnog logičkog kontrolera (PLC-a)**

Naslov rada na engleskom jeziku: **Smart house control system implementation by means of programmable logic controller (PLC)**

Opis zadatka:

U današnje vrijeme izrazito je aktualan razvoj tzv. pametnih kuća koje integriraju različite senzore i uređaje u svrhu poboljšanja kvalitete života. S obzirom na iznimne performanse suvremenih PLC sustava moguće ih je iskoristiti kao upravljačke uređaje za automatizaciju pametne kuće. U radu je potrebno napraviti sljedeće:

1. Proučiti relevantnu dokumentaciju programibilnog logičkog kontrolera (PLC-a) FX5U-32MT/ESS i grafičkog operaterskog terminala GS2107-WTBD proizvođača tvrtke Mitsubishi Electric.
2. Osmisliti elektroničko sučelje prekidača za svjetlo, motora za dizanje roleta i senzora temperature prema PLC-u.
3. Realizirati upravljački program na PLC-u (sustav logičkog upravljanja) i grafičko sučelje prema korisniku unutar operaterskog terminala.
4. Realizirati softversko mrežno sučelje temeljeno na TCP/IP ili sličnom protokolu za prijenos stanja PLC-a na mrežu i daljinsko upravljanje.

U radu, također, treba navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
30. studenoga 2020.

Datum predaje rada:
1. rok: 18 veljače 2021.
2. rok (izvanredni): 5. srpnja 2021.
3. rok: 23. rujna 2021.

Predviđeni datumi obrane:
1. rok: 22.2. – 26.2.2021.
2. rok (izvanredni): 9.7.2021.
3. rok: 27.9. – 1.10.2021.

Zadatak zadao:
Danijel Pavković
Izv. prof. dr. sc. Danijel Pavković

Predsjednik Povjerenstva:
Prof. dr. sc. Branko Bauer

Sadržaj

Sadržaj I

Popis slika III

Popis tablica V

Popis oznaka VI

Sažetak VII

Summary VIII

1 Uvod 1

2 Opis komponenti PLC sustava 5

2.1 Mitsubishi FX5U-32MT/ESS 5

2.2 HMI-GS2107-WTBD 7

2.3 GX Works3 8

2.4 GT Designer 9

3 Elektroničko sučelje 10

3.1 Konceptualni prikaz pametne kuće 10

3.2 Upravljanje rasvjetom 11

3.3 Upravljanje roletom 11

3.4 Regulacija temperature 11

3.4.1 Mjerni član 12

3.4.2 Izvršni član 15

3.4.3 Regulacijski član 15

3.4.4 Pulsno širinska modulacija 17

3.5 Upravljačka pločica 17

3.5.1 Model upravljačke pločice 19

3.6 Električna shema spajanja modela 20

4 Upravljački program 21

4.1 Sustav logičkog upravljanja 21
4.1.1 Upravljanje rasvjetom 21
4.1.2 Upravljanje roletom 22
4.1.3 Regulacija temperature 23
4.1.4 Oznake korištene unutar upravljačkog programa 24
4.2 Grafičko sučelje 25
5 Mrežno sučelje temeljeno na TCP/IP protokolu 28
5.1 Modbus 28
5.2 TCP/IP protokol 29
5.2.1 Arhitektura TCP/IP protokola 30
5.2.2 Aplikacijski sloj 31
5.2.3 Transportni sloj 32
5.2.4 Internet sloj 33
5.2.5 Sloj povezivanja 34
5.3 Ostvarivanje komunikacije s PLC-om 35
5.4 Primjeri komunikacije 36
5.4.1 Čitanje izlaza 36
5.4.2 Čitanje temperature 37
5.4.3 Promjena izlaza 38
5.4.4 Upisivanje reference temperature 39
6 Laboratorijski postav 40
7 Zaključak 41
8 Literatura 42

Popis slika

Slika 1-1 Integrirani sustav pametne kuće [5] 2
Slika 1-2 Sustavi automatizacije pametne kuće [6] 3
Slika 1-3 Protokoli komunikacije [2] 4
Slika 2-1 Mitsubishi FX5U-32MT/ESS [8] 5
Slika 2-2 HMI-GS2107-WTBD [10] 7
Slika 2-3 Vanjske dimenzije HMI-a [9] 7
Slika 2-4 Izgled prozora GX Works3 8
Slika 2-5 Izgled prozora GT Designer 9
Slika 3-1 Prikaz tlocrta prostorija sa elementima sustava 10
Slika 3-2 Regulacijski krug 12
Slika 3-3 Karakteristika NTC otpornika [15] 13
Slika 3-4 NTC u senzor spoju djelitelja napona [15] 13
Slika 3-5 Burkert 6223 proporcionalni ventil [16] 15
Slika 3-6 Relej upravljačke pločice 18
Slika 3-7 Redne stezaljke 18
Slika 3-8 Skica elemenata i raspored priključaka na upravljačkoj pločici 19
Slika 3-9 Električna shema upravljanja pločice 19
Slika 3-10 Električna shema spajanja modela 20
Slika 4-1 Program za upravljanje rasvjetom 21
Slika 4-2 Program za upravljanje roletom 22
Slika 4-3 Program unutar funkcijskog bloka RoleteLogika 23
Slika 4-4 Program regulacije temperature 23
Slika 4-5 ST kod unutar funkcijskog bloka konverzije temperature 24
Slika 4-6 ST kod PI regulatora 24
Slika 4-7 Globalne oznake upravljačkog programa 25
Slika 4-8 Lokalne oznake konverzije temperature 25
Slika 4-9 Lokalne oznake regulatora 25
Slika 4-10 Početni zaslon 26
Slika 4-11 Zaslon prostorije 1 26
Slika 4-12 Zaslon rasvjete 27
Slika 4-13 Zaslon rolete 27
Slika 4-14 Zaslon regulacije temperature 28

Slika 5-1 Enkapsulacija podataka [22] 31
Slika 5-2 Arhitektura TCP/IP modela sa protokolima [21] 35
Slika 5-3 Mapa pyModbusTCP paketa [23] 36
Slika 5-4 Čitanje izlaza 36
Slika 5-5 Čitanje temperature prostorije 37
Slika 5-6 Promjena izlaza 38
Slika 5-7 Upisivanje reference temperature 39
Slika 6-1 Laboratorijski postav 40

Popis tablica

Tablica 1 Tablica stanja T bistabila 11
Tablica 2 Metoda Ziegler-Nichols 17
Tablica 3 Oznake i funkcije releja 19

Popis oznaka

Oznaka	Mjerna jedinica	Opis
A,B,C		Steinhart-Hart koeficijenti
e		regulacijsko odstupanje
$G_R(s)$		prijenosna funkcija regulatora
K_P		proporcionalno pojačanje
K_R		pojačanje regulatora
R	Ω	otpor
R_{10k}		otpornik
R_{NTC}	Ω	vrijednost NTC otpornika
T	K	Temperatura
T_i	s	integralno vrijeme
U_{AD}	V	napon AD pretvornika
U_{izl}	V	izlazni napon
Ucc	V	napon napajanja otporničkog djelila
U_{mj}	V	mjerni napon
w		vodeća veličina
x		regulirana veličina
X_p		proporcionalno područje
X_{pkr}		kritično proporcionalno područje

Sažetak

Ovaj završni rad obrađuje sustav pametne kuće implementiran pomoću programibilnog logičkog kontrolera (PLC). Prikazani su osnovni dijelovi, načini komunikacije te upravljani procesi unutar sustava. Osmišljeno je i razrađeno električno sučelje za upravljanje rasvjetom, roletom prozora te regulaciju temperature. Također je izveden upravljački program i grafičko sučelje prema korisniku koji su implementirani na laboratorijskom postavu. Laboratorijski postav sadrži upravljačke sklopove tvrtke Mitsubishi Electric tj. PLC FX5U-32MT/ESS i grafički operatorski terminal GS2107-WTBD. Također je prikazana komunikacija s PLC-om preko Modbus TCP/IP protokola i Pyton programa pomoću kojeg je moguće čitati i mijenjati stanja izlaza i registara PLC-a. Suština ovog rada je utvrditi mogućnosti PLC-a pri upravljanju pametnom kućom te prikazati upravljanje pomoću mreže.

Ključne riječi: pametna kuća, programibilni logički kontroler, Modbus TCP/IP protokol

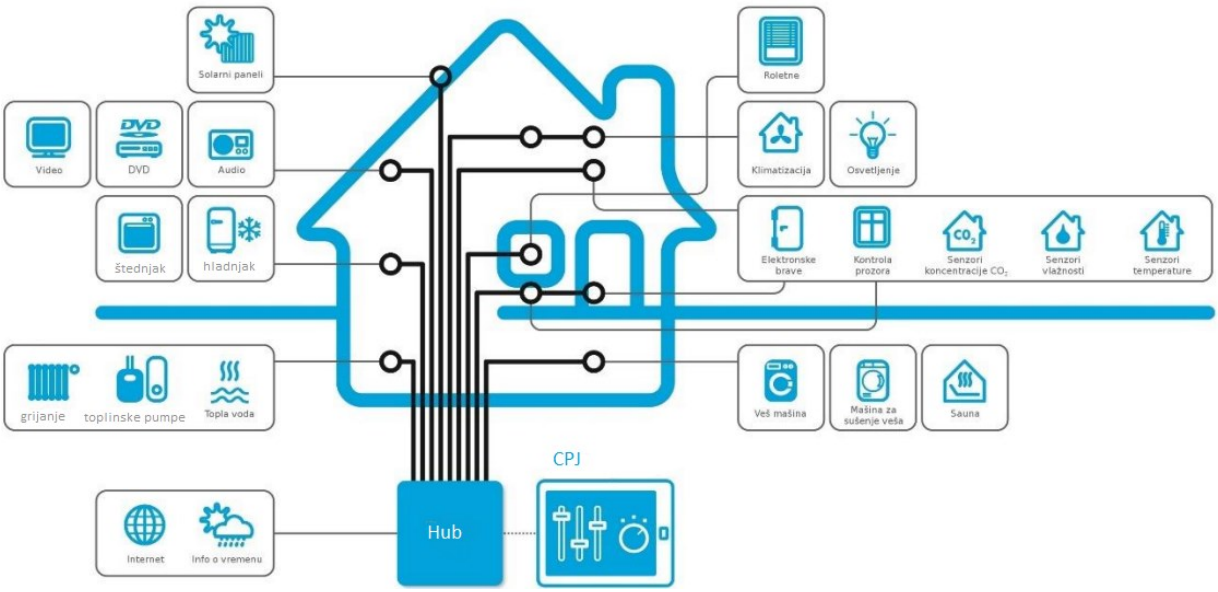
Summary

This final paper deals with a smart home system implemented using a programmable logic controller (PLC). The basic parts, methods of comunicaton and managed processes within the system are presented. An eletrical interface for lighting control, window blinds and temperature regulation have been designed and developed. A driver and graphical user interface were also developed and implemented in the laboratory setup. The laboratory setup contains Mitsubishi Electric control circuits, ie. PLC FX5U-32MT/ESS and the GS2107-WTBD graphics operator terminal. Communication with the PLC via the Modbus TCP/IP comunication is also shown, with the help of Pyton program which enables the read and change of the output states and registers of the PLC. The purpose of this paper is to determine the capabilities of the PLC in smart home managment and to show the possibility of remote control thru network.

Keywords: smart house, programmable logic controller, Modbus TCP/IP protocol.

1 Uvod

Pametna kuća je integrirani sustav uređaja u domaćinstvu kojima se upravlja na različite načine ovisno o potrebama stanara kuće. Takvi uređaji su umreženi, što im omogućava međusobnu razmjenu podataka, daljinsku kontrolu i ostale korisne primjene. Sa kućnom automatizacijom uređaji mogu aktivirati jedan drugoga ili uz pomoć ručne aktivacije preko aplikacije ili glasovnom naredbom. Pametne kuće su budućnost izgradnje domova zbog brzog i opširnog razvoja novih tehnologija, koje će omogućiti potpunu automatizaciju i autonomiju same kuće te tako poboljšati kvalitetu i komfor života. Pojam pametne kuće odnosi se na sustav gospodarenja energijom i utjecaja na okoliš primjenom različitih senzora i sustava za automatizaciju. Kao primjer možemo uzeti rasvjetu koja se može upaliti ili ugasiti ovisno o dobu dana ili navikama spavanja stanara. Također je korisno programirati sustav grijanja na način da se kuća zagrije na ugodnu temperaturu netom prije dolaska stanara (npr. povratak s posla) bez suvišnog trošenja energije. Nadalje pametne kuće mogu doprinijeti većoj sigurnosti sa uređajima interneta stvari (*engl. Internet of Things, IoT*) kao što su sigurnosni sustavi i kamere. Internet stvari (IoT) opisuje mrežu fizičkih objekata ili stvari koje su opremljene senzorima, softverima i drugim tehnologijama u svrhu povezivanja i razmjene podataka s drugim uređajima i sustavima pomoću interneta. Predstavlja mrežnu strukturu u kojoj fizičke i virtualne stvari svih vrsta međusobno komuniciraju. Spajanje uređaja može biti bežično i omogućava nove mogućnosti poput međusobne interakcije različitih sustava te noviteta unutar njihove kontrole, praćenja i pružanja usluga [1-4]. Na slici 1-1 prikazan je integrirani sustav pametne kuće.



Slika 1-1 Integrirani sustav pametne kuće [5]

Kućna automatizacija radi putem mreže uređaja koji su povezani s internetom putem različitih komunikacijskih protokola poput Wi-Fi, Bluetooth, ZigBee i sl. Kroz elektronička sučelja moguće je preko kontrolera upravljati uređajima, bilo sa glasovnim naredbama, korisničkim sučeljima ili aplikacijama. Mnogi od tih uređaja imaju senzore koji nadziru promjene u kretanju, temperaturi, svjetlosti i sl. Sustav preko senzora dobiva informacije o okolini uređaja koje se zatim procesiraju unutar centralne jedinice (mikroprocesor, računalo). Kako bi se izvršile neke promjene u okolini centralna jedinica aktivira aktivatore, fizičke mehanizme poput prekidača za svjetlo, motoriziranih ventila ili motora. Neki sustavi zahtijevaju središnji sustav, dok se neke mobilne aplikacije direktno vežu sa IoT uređajem preko rutera, ako se ne želi opterećivati kućni ruter, poželjno je koristiti središnji sustav[3].

Neke od funkcija koje se često upravljaju su: grijanje, osvjetljenje, alarmni sustavi, bijela (hladnjaci, pećnice, perilice) i smeđa (televizori, radio, kućno kino) tehnika, ventilacija, komunikacijski uređaji i sl. Skraćeno svi sustavi vezani za jedno kućanstvo. Na slici 1-2 prikazani su sustavi kojima se često upravlja u pametnoj kući.

Slika 1-2 Sustavi automatizacije pametne kuće [6]

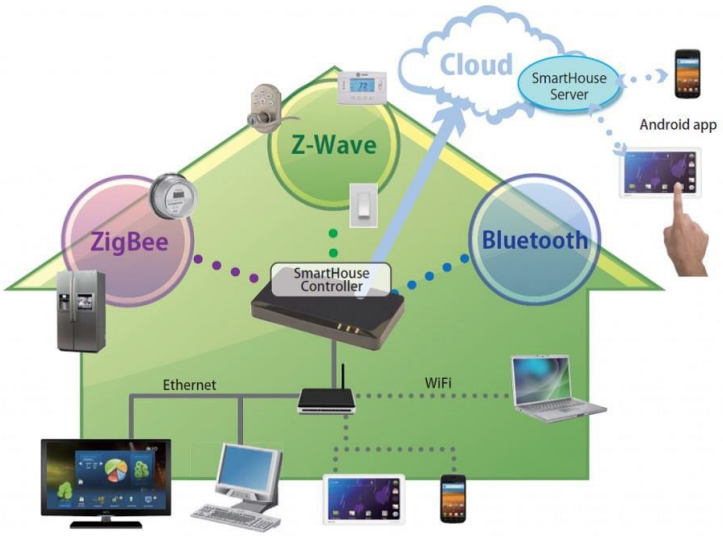
Automatizacija kuće odvija se na tri razine:

- praćenje – korisnik provjerava stanje uređaja i njihove okoline (npr. praćenje temperature jačine svjetlosti i sl.),
- kontrola – korisnik bežično upravlja uređajima zbog dodatnih informacija o okolini (npr. zakretanje kamere zbog povećanja nadzora),
- automatizacija – korisnik postavlja referentne vrijednosti, sustav međusobno komunicira i aktivira određene aktivatore (npr. regulacija temperature).

Način na koji uređaji međusobno komuniciraju i povezuju na internet ovisi o komunikacijskom protokolu. Pa tako razlikujemo više vrsta protokola korištenih u pametnim kućama:

- WiFi – najčešći oblik komunikacije. IoT uređaji međusobno komuniciraju preko lokalne internetske mreže,
- Z-Wave – bežična tehnologija koja komunicira pomoću mikrovalova frekvencije 908.42 MHz,
- ZigBee – slično kao i Z-Wave, ZigBee je bežična mreža male snage koja se temelji na standardu otvorene IP adrese
- Bluetooth –komunikacija se temelji na bluetooth vezi [3].

Na slici 1-3 prikazana je komunikacija različitim protokolima.



Slika 1-3 Protokoli komunikacije [2]

Za većinu kućanstva WiFi protokol je dovoljan, ali ako se radi o ozbiljnim projektima pametnih kuća, poželjno je razmišljati o drugim protokolima. Prednosti pametnih kuća su daljinski pristup, komfor, energentska učinkovitost te razne ostale pogodnosti. Kada navodimo nedostatke pametnih kuća prvo mjesto zauzima cijena. Svi uređaji su puno skuplji zbog dodatnih značajki, npr. kod pametne žarulje koja posjeduje daljinsko upravljanje, različite boje, promjena jakosti svjetla i sl. Nadalje problem je i sigurnost jer su svi ti uređaji povezani sa internetom te ih je moguće kompromitirati. Dakako u to ulazi i problem privatnosti ako se napadač uspije povezati sa video nadzorom pametne kuće. Posljednji problem je nepotpuno razrađena nova tehnologija s potencijalnim greškama, gdje može doći do problema povezivosti ili odziva uređaja. Taj problem se rješava novim softverskim ažuriranjima te daljnjim razvojem i napretkom tehnologije [3].

Rad je organiziran kako slijedi.

U poglavlju 2 opisani su dijelovi korištenog upravljačkog sustava pametne kuće temeljenog na PLC-u, te popratnih alata potrebnih za programiranje. U poglavlju 3 prikazan je konceptualni prikaz pametne kuće sa svim elementima, parametrima, elektronikom, instalacijama i shemama spajanja. U poglavlju 4 prikazan je i objašnjen upravljački program te izgled i funkcionalnost grafičkog sučelja prema korisniku. U poglavlju 5 objašnjen je Modbus i TCP/IP protokol te je pokazano upravljanje registrima PLC-a preko računala uz pomoć Pyton skripte. U poglavlju 6 opisan je laboratorijski postav, dok je u poglavlju 7 prikazan zaključak i prijedlozi za daljnji rad.

2 Opis komponenti PLC sustava

2.1 Mitsubishi FX5U-32MT/ESS

PLC (*engl. Programmable logic controller*) je programabilni logički kontroler, tj. industrijsko računalo koje se sastoji od memorije, procesora te industrijskih ulaza i izlaza. Ti ulazi nisu miševi i tipkovnice kao kod običnih računala nego su to tipkala, sklopke te razne vrste pretvornika i senzora. PLC se najviše koristi u industriji kao osnovni dio upravljačkih automatskih sustava. Program PLC-a lako je promjenjiv te pogodan za brzo rješenje problema. PLC je digitalno računalo i njegov program se izvršava ciklički i sastoji se od 3 faze. Prva faza je čitanje vrijednosti ulaznih varijabli, zatim slijedi izvršavanje zadanog programskog koda te na kraju ispisivanje rezultata logičkih operacija na izlaze PLC-a. Projektiran je za teške uvjete rada, otporan na vibracije, temperaturne promjene i električne smetnje [7].

U ovom radu koristi se PLC tvrtke Mitsubishi. Mitsubishi Electric nudi dvije serije kompaktnih kontrolera: FX5(iQ-F) i FX3. U ovom radu koristi se serija MELSEC iQ-F koju odlikuju napredne funkcije, unaprijeđena sigurnost i intuitivno programiranje. Na slici 2-1 prikazan je osnovni PLC na koji je moguće, ovisno o potrebi, dodavati ekspanzijske module.

Slika 2-1 Mitsubishi FX5U-32MT/ESS [8]

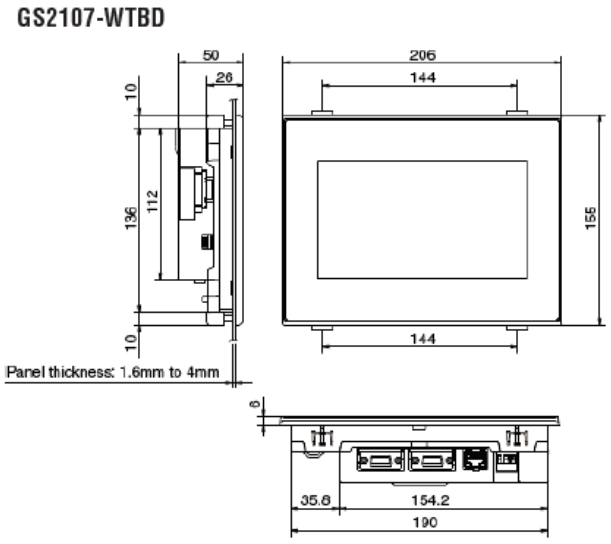
Neke od glavnih karakteristika ovog PLC-a su:

- podrška za analogne signale, ulaze i izlaze (*engl. Analog signals I/O support*), FX5U-32MT/ESS je opremljen s mogućnostima upravljanja analognim signalima bez upotrebe „ladder" logike u postavkama parametara,
- podrška za SD karticu, pogodna prilikom ažuriranja PLC-a na terenu, spremanje programa na karticu i distribucija na ostale kontrolere. Također se može koristiti i za „data logging",
- RUN/STOP/RESET sklopka, PLC može biti isključen bez isključivanja glavnog napajanja za učinkovito traženje grešaka,
- ethernet priključak – osobni, komercijalni i industrijski standard za lako i efikasno prebacivanje podataka, nužan za mrežnu komunikaciju i povezivanje sa HMI-em (*engl. Human Machine Interface*),
- ugrađenu RS-485 (Modbus) komunikaciju, podržava pouzdano i efikasno spajanje s ostalim PLC-ima, senzorima, modemima i slično,
- sistemska sabirnica visoke brzine (*engl. High-speed System Bus*) pruža brzi prijenos podataka između računala i PLC-a. Nova arhitektura pruža i do 150 puta veću brzinu nego na FX3U, garantirana brzina čak i sa ekspanzijskim modulima,
- pozicioniranje, mogućnost pozicioniranja u četiri osi za servo i step motore,
- sigurnosne značajke, ograničavanje pristupa PLC-u pomoću zaporke,
- integrirani „real-time clock",
- prikaz aktivnih ulaza/izlaza pomoću LED dioda [8].

2.2 HMI-GS2107-WTBD

Korisničko sučelje (*engl. Human Machine Interface, HMI*) povezuje korisnika i upravljačku jedinicu industrijskog sustava. HMI posjeduje elektroničke komponente za signalizaciju i upravljanje sustava automatike. HMI modul može prevesti podatke industrijskih procesa i sustava u jezik koji korisnik razumije. Pomoću HMI-a korisnik može promatrati sheme sustava, pokretati i upravljati sustavom, te vršiti nadzor sustava (upozorenja, pogreške, grafovi i sl.). HMI modul povezan je sa PLC-om te je na njemu moguće prikazati virtualne elemente poput prekidača ili indikatora koji su prije morali biti žičano povezani sa PLC-om. Mitsubishi Electric tvrtka nudi mogućnosti promjene izgleda zaslona kao i mnoge druge napredne funkcije u softveru za konfiguriranje HMI modula tj. GT Designer3. U ovom radu koristi se GS2107-WTBD korisničko sučelje preko kojeg se simuliraju neki ulazni i izvršni elementi, određuju reference te prati ponašanje naše pametne kuće. Koristi se „transparent“ mod komunikacija između PLC-a i HMI-a. U ovoj vrsti komunikacije HMI je povezan pomoću USB kabela sa računalom, a PLC i HMI su međusobno povezani Ethernet vezom [9]. Na slici 2-2 možemo vidjeti izgled HMI-a, a na slici 2-3 vanjske dimenzije koje su važne prilikom montaže ekrana na zid prostorije.

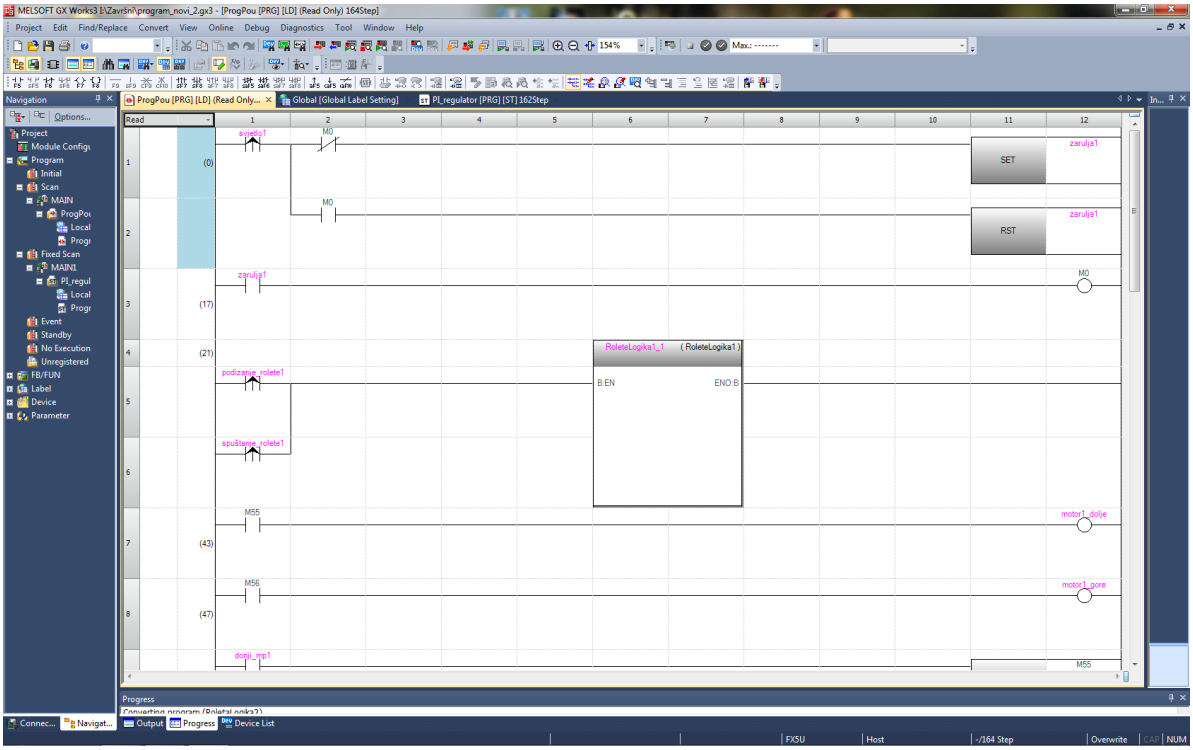
Slika 2-2 HMI-GS2107-WTBD [10]



Slika 2-3 Vanjske dimenzije HMI-a [9]

2.3 GX Works3

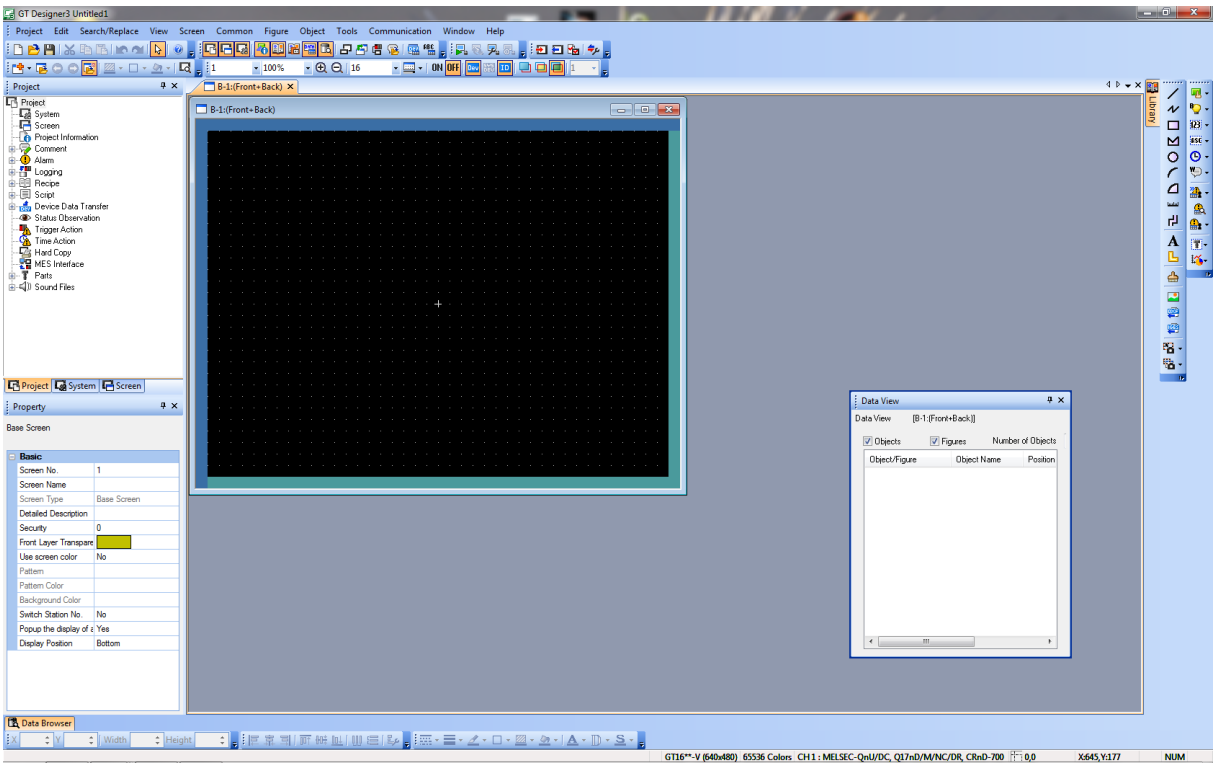
GX Works3 je najnovija generacija softvera za programiranje i održavanje koji nudi Mitsubishi Electric posebno dizajniran za upravljački sustav serije MELSEC iQ-R i MELSEC iQ-F koji se koristi u ovome radu. Softver uključuje mnoštvo novih značajki i tehnologija koje osiguravaju brzo i efikasno rješenje inženjerskih problema. GX Works3 sadrži ažuriranu knjižnicu objekata unaprijed instaliranih unutar programa te knjižnicu modula koji su bili dostupni u vrijeme izdavanja programa. Podržava glavne IEC jezike. Različiti programski jezici (ladder, structured text, function block) mogu se koristiti unutar programa istovremeno uz jednostavnu i brzu promjenu jezika pomoću kartice izbornika. Varijable i uređaji koji se koriste u svakom programu mogu se dijeliti na više platformi s podrškom za prilagodbu funkcionalnih blokova. U ovom radu GX Works 3 se koristi za programiranje upravljačkog programa pametne kuće [11].



Slika 2-4 Izgled prozora GX Works3

2.4 GT Designer

GT Designer 3 je softver za dizajniranje i konfiguraciju zaslona HMI-a. Optimizacija polja za unos podataka, prozora dijaloga kao i pomoći pri postavljanju elemenata pridonosi smanjenju vremena razvoja i traženoj fleksibilnosti programa. Nakon završetka razvoja zaslona moguće je pokrenuti simulaciju koja programerima omogućuje uklanjanje grešaka i provjeru projekata unutar stvarnog vremena. HMI je moguće koristiti i za upravljanje PLC-a uz dizajn zaslona sa potrebnim kontaktima koji su povezani sa kontaktima PLC-a. Jedan od naprednijih alata je „Utilize Data Screen“ unutar kojih se mogu pronaći unaprijed izrađeni zasloni za napredne servo, robotske i generalno automatizacijske sustave. Na kraju je potrebno samo povezati varijable što unaprjeđuje programiranje, odnosno ubrzava proces programiranja i izrade sustava [12].

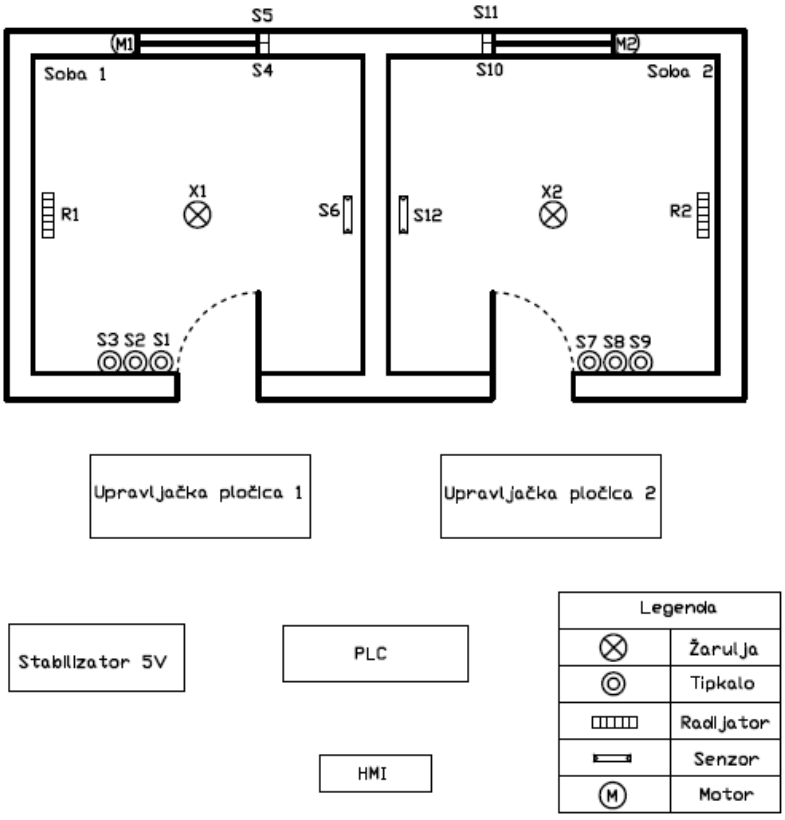


Slika 2-5 Izgled prozora GT Designer

3 Elektroničko sučelje

3.1 Konceptualni prikaz pametne kuće

Konceptualni prikaz pametne kuće podijeljen je na dvije prostorije identičnih elemenata i svojstava. U svakoj sobi nalazi se prozor sa roletom, a podizanje i spuštanje rolete je prvi sustav. Upravljanje se vrši preko tipkala na zidu ili preko grafičkog sučelja. Drugi sustav je jednostavna upali/ugasi rasvjeta koja se također upravlja tipkalom na zidu ili preko grafičkog sučelja. Treći sustav je regulacija temperature u prostoriji čiju vrijednost mjeri senzor temperature. Referentnu vrijednost temperature moguće je promijeniti u jednoj od opcija na grafičkom sučelju. Svaka prostorija ima svoju upravljačku pločicu na koju se spajaju sve električne veličine vezane za tu prostoriju. Na pločici se nalaze releji kojima se aktiviraju određene komponente. Upravljačka pločica se određenim brojem žica spaja na ulaze i izlaze PLC-a. Ukupan broj ulaza je 12 (2 analogna, 10 digitalnih), a izlaza 8. HMI se preko UTP/Ethernet kabla spaja na PLC. Na slici 3-1 prikazan je tlocrt dvije prostorije sa svim pripadnim elementima spomenutih sustava.



Slika 3-1 Prikaz tlocrta prostorija sa elementima sustava

3.2 Upravljanje rasvjetom

Upravljanje rasvjetom vrši se preko tipkala na zidu ili preko grafičkog sučelja. Tipkalom S1 upravlja se rasvjetom u sobi 1, dok se za svjetlo u sobi 2 koristi tipkalo S7. Za promjenu stanja rasvjete koristi se samo jedno tipkalo kako bi se smanjio broj ulaza u PLC. Za taj način upravljanja koristi se logika T bistabila koja je implementirana u upravljačkom programu.

Tablica 1 Tablica stanja T bistabila

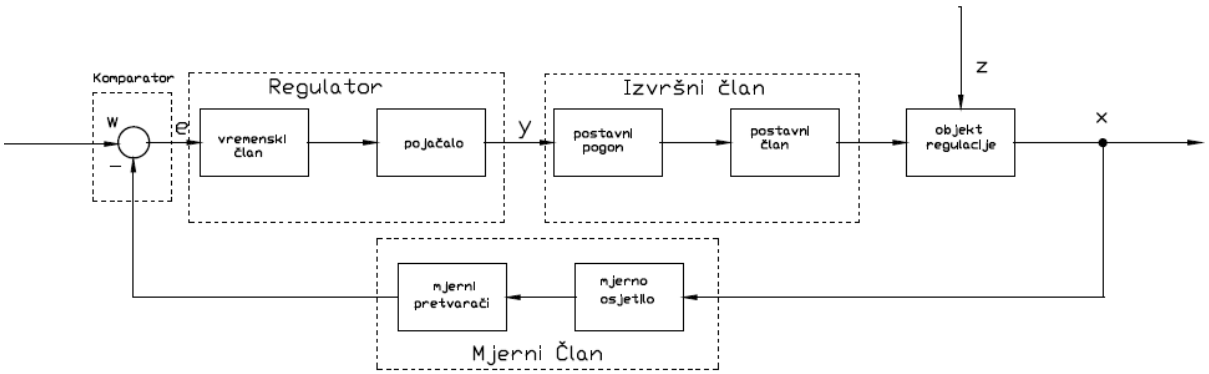
T	Q_{n+1}
0	Qn
1	$\overline{Qn}$

3.3 Upravljanje roletom

Upravljanje roletom prozora u prostoriji 1 vrši se pomoću tipkala S2 gdje se daje signal motoru za spuštanje rolete. Tipkalo S3 daje signal motoru za podizanje rolete. Tipkala su spojena u logici RS bistabila, npr. prilikom pritiska tipkala za dolje, kretnja rolete prema gore se prekida i obrnuto. Analogno tome na isti način djeluje i roleta u prostoriji 2 uz tipkalo S7 (tipka za dolje) i tipkala S8 (tipka za gore). Navedene tipke za gore i dolje nalaziti će se i u opciji upravljanjem prozora na grafičkom sučelju. Na oba prozora nalaze se mikro prekidači koji prepoznaju gornju krajnju točku (S5 i S11) i donju krajnju točku (S4 i S10) rolete. Mikro prekidači služe kao sigurnost i kao detektirani položaj kompletno zatvorene ili otvorene rolete. Pomak rolete ostvarujemo pomoću elektromotora koji je direktno postavljen na osovinu oko koje se namata roleta. Pretpostavlja se da motor već na sebi ima elektroniku za njegovo upravljanje te da ima samo 2 priključka.

3.4 Regulacija temperature

Regulacija je proces u kojem regulirana veličina djeluje povratno na ulaznu veličinu, održavajući željeno stanje. Regulirana veličina *x* na izlazu staze dolazi najprije do mjernog člana koji se sastoji od mjernog osjetila i pretvarača. Mjerno osjetilo ima zadatak da mjeri trenutačnu vrijednost regulirane veličine i vodi je dalje u prikladnom obliku. Regulirana veličina iz mjernog člana ulazi u komparator gdje se uspoređuje sa vodećom veličinom *w*. Razlika tih veličina je regulacijsko odstupanje *e*. Zatim taj signal prolazi kroz regulator te se preoblikuje sa zahtjevima regulacijskog kruga. Nadalje regulacijski signal vodi se dalje u izvršni član koji direktno ili indirektno djeluje na reguliranu veličinu [13]. Na slici 3-2 nalazi se blok dijagram regulacijskog kruga.



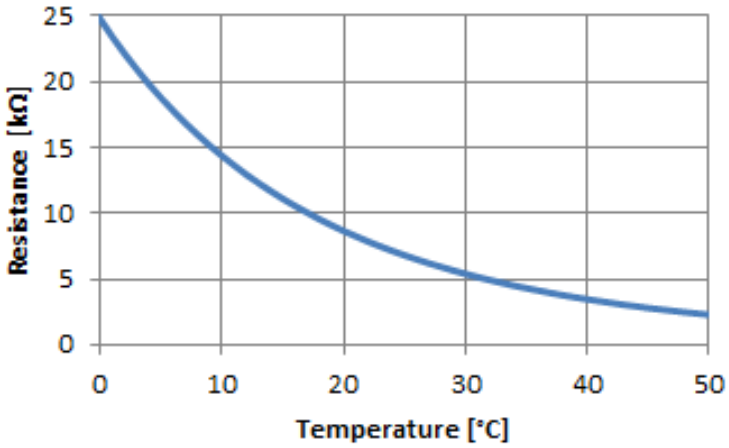
Slika 3-2 Regulacijski krug

3.4.1 Mjerni član

Na modelu pametne kuće mjerno osjetilo je termistor. Na tlocrtu prostorije označen S6 i S12. Termistor je električni termootpornik zasnovan na poluvodičkom materijalu. Električni otpor termistora mijenja se s promjenom temperature. Postoje dvije vrste termistora [14]:

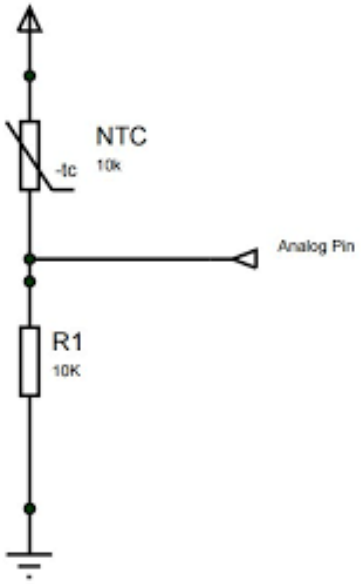
1. PTC (*engl. Positive Temperature coefficient*) termistori koji imaju pozitivni koeficijent temperaturne promjene otpora, s rastom temperature povećava se i električni otpor.
2. NTC (*engl. Negative Temperature coefficient*) termistori koji imaju negativan koeficijent temperaturne promjene otpora tj. s povećanjem temperature električni otpor se smanjuje.

NTC otpornik, koji spada u grupu NTC termistora, je otpornik promjenjivog otpora koji se mijenja u ovisnosti o temperaturi. Koristi se u vrlo puno uređaja zbog svoje male cijene i jednostavnosti. Prednost NTC otpornika je veliki raspon temperatura koje može mjeriti, a ono puno važnije je što se električna shema vrlo lako i brzo realizira pomoću djelitelja napona. Slika 3-3 prikazuje karakteristiku NTC otpornika odnosno ovisnost otpora o temperaturi [15].



Slika 3-3 Karakteristika NTC otpornika [15]

U ovom radu se kao senzor temperature koristi se NTC otpornik nazivnog iznosa 10kΩ. NTC otpornik spajamo u seriju sa otpornikom od 10kΩ. Tako se ostvaruje spoj djelitelja napona (Slika 3-4) u kojem se mjeri napon na konstantnom otporniku. Napon je promjenjiv s otporom NTC senzora. Točnije prilikom porasta temperature smanjuje se otpor NTC-a, pa se time i mijenja njegov pad napona, što rezultira porastom napona na konstantnom otporniku prema Kirchhoffovom zakonu za napone. Taj se napon dovodi na AD pretvornik gdje se analogan signal napona mijenja u digitalni.[15].



Slika 3-4 NTC u senzor spoju djelitelja napona [15]

Kako bi mogli mjeriti temperaturu pomoću izabranog senzora potrebno je odrediti matematičku ovisnost temperature i otpora. Za prikaz matematičke ovisnosti koristi se Steinhart-Hart jednadžba koja glasi:

$$\frac{1}{T} = A + BlnR + C(\ln(R))^3, \quad (1)$$

Gdje su:

T-temperatura, K

R-otpor, Ω

A,B,C -Steinhart-Hart koeficijenti koji ovise o modelu termistora

Koeficijenti iz jednadžbe (1) određuju se na osnovi podataka, različite vrijednosti otpora povezane sa pripadnim temperaturama. Kako bi se povratnoj petlji regulacije unutar programa na PLC-u dobio podatak o mjerenoj temperaturi izraženoj u °C potrebno je prilagoditi matematički izraz. Iz Slike 3-4 možemo zaključiti da je izlazni napon jednak:

$$U_{izl} = U_{cc}\frac{R_{10k}}{R_{10k} + R_{NTC}} \quad (2)$$

Gdje je:

U_{izl}	napon na konstantnom otporniku, V
U_{cc}	napon napajanja, V
R_{NTC}	vrijednost NTC otpornika, Ω
R_{10k}	vrijednost konstantnog otpornika, Ω

Formula koja povezuje digitalni broj i napon prikazana je izrazom:

$$ADV = \frac{U_{mj}}{U_{AD}}4000 \quad (3)$$

Koristi se vrijednost od 4000 jer je na taj broj normiran 12 bitni AD pretvornik unutar PLC-a. Nakon uvrštavanja (2) u jednadžbu (3) pri čemu je U_{mj}=U_{izl}, dobije se izraz za vrijednost otpora.

$$R_{NTC} = R_{10k}\left(\frac{2000}{ADV} - 1\right) \quad (4)$$

Vrijednost otpora koji se dobije izrazom (4) uvrštava se u Steinhart-Hart jednadžbu (1) i dobije se vrijednost temperature izraženu u kelvinima [15].

3.4.2 Izvršni član

U slučaju regulacije temperature izvršne članove predstavljat će ventili na radijatorima R1 i R2 koji zagrijavaju prostoriju do željene temperature. Izmjena topline na radijator vrši se preko zagrijanog medija (najčešće vodom) iz kotlovnice. Regulacija topline vrši se indirektno preko regulacije protoka medija kroz radijator. Regulaciju protoka ostvarujemo sa proporcionalnim ventilom. Prednost ovakvog ventila je što se može upravljati pomoću PWM-a čime se rješava problem statičkog trenja unutar ventila. Ventil je u stacionarnom stanju čvrsto zatvoren te nije potreban dodatan ventil za zatvaranje. Temperaturno područje rada ventila je između -10°C i +90°C, što je idealno za sustave grijanja vodom [16]. Na slici 3-5 prikazan je proporcionalan ventil za moguću upotrebu.

Slika 3-5 Burkert 6223 proporcionalni ventil [16]

3.4.3 Regulacijski član

Regulacijski član je dio upravljačkog kruga unutar kojeg se izvršavaju određene logičke i upravljačke funkcije. Regulacijski član sastoji se od 3 člana:

- komparatora koji uspoređuje referentnu i reguliranu veličinu a često je s njim povezan i davač referentne veličine,
- regulatora unutar kojeg se regulacijski signal preoblikuje vremenski ovisno o djelovanju. Ta djelovanja preoblikuju ulazni signal na karakterističan način što utječe na performanse regulatora. Razlikujemo tri vrste djelovanja. Proporcionalno djelovanje koje razmjerno reagira na nagle promjene regulacijskog odstupanja i ubrzava odziv

zatvorenog kruga. Veći iznosi proporcionalnog pojačanja mogu smanjiti stupanj prigušenja. Samostalno proporcionalno djelovanje u većini slučajeva ne može odstraniti trajno regulacijsko odstupanje. Integralno djelovanje većinom može ukloniti trajno regulacijsko odstupanje čime se unaprjeđuje stacionarna točnost regulacijskog sustava. Također djeluje i na brzinu odziva ali može narušiti stabilnost regulacijskog sustava jer uvodi fazno kašnjenje u prijenosnu funkciju otvorenog kruga. Derivacjisko djelovanje djeluje u odnosu na trend kretanja regulacijskog odstupanja. Time ubrzava odziv te djeluje stabilizirajuće uvođenjem faznog prethođenja u prijenosnu funkciju otvorenog kruga,

- regulacijskog pojačala čija je svrha pojačanje izlaznog regulacijskog signala. Stupanj pojačanja čini omjer izlaznog i ulaznog signala. Imperativ pojačala je da uz što manju tromost ima što veće pojačanje [15].

Spajanjem I-regulatora i P-regulatora dobivamo PI-regulator gdje do izražaja dolaze prednosti obaju djelovanja. P-djelovanje osigurava brzu reakciju, dok I-djelovanje otklanja sporo ali potpuno trajno regulacijsko odstupanje. Ovaj tip regulatora koristimo kad sa samim djelovanjem P-regulatora ne možemo dovoljno smanjiti trajno regulacijsko odstupanje, a da ne ugrozimo stabilnost ili kad bi sam I-regulator uzrokovao nestabilnost, odnosno kad bi odveć presporo reagirao. PI-regulator je pogodan za staze sa mrtvim vremenom i za staze s izjednačenjem višeg reda. Zato je PI najčešće korišten tip regulatora u procesnoj regulaciji [13]. Prijenosna funkcija PI-regulatora glasi:

$$G_R(s) = \frac{Y(s)}{E(s)} = K_P\left[1 + \frac{1}{T_i s}\right] \qquad (5)$$

U regulaciji temperature koristimo PI-regulator zbog navedenih prednosti. Parametre regulatora moguće je podesiti prema Ziegler-Nicholsovoj metodi. Ova se metoda primjenjuje kod staza višeg reda koje se mogu prikazati kombinacijom člana s mrtvim vremenom i aperiodskog člana, kod kojih vremenski odziv nije poznat. Postupak namještanja je sljedeći [13]:

- regulator se namjesti na čisto P-djelovanje, integralno djelovanje se postavi kao $T_i=0$,
- pojačanje K_p poveća se dok se ne pojave oscilacije, odnosno do granice stabilnosti. Tada se određuje kritično proporcionalno područje X_{pkr} kao i kritični period $T_{kr}=2\pi/\omega$,
- na osnovi tih dvaju vrijednosti i tablice 2, namjesti se prema tipu regulatora, u našem slučaju PI, proporcionalno područje X_p i integralno vrijeme T_i.

Tablica 2 Metoda Ziegler-Nichols

	X_p	T_i	T_d
P	2,0	-	-
PI	2,2	0,85	-
PID	1,7	0,5	0,12

Nedostatak ove metode je što se regulacijski krug dovodi do granice oscilacija, a to ponekad nije dozvoljeno [13].

3.4.4 Pulsno širinska modulacija

Pulsno širinska modulacija (*engl. Pulse width modulation – PWM*) korisna je tehnika za upravljanje analognim i digitalnim krugovima pomoću digitalnih izlaza. PWM se koristi u raznim aplikacijama poput mjerenja, komunikacije, upravljanja i konverzije. Digitalnim upravljanjem analognih sklopova cijena sustava i potrošnja energije se znatno smanjuje. Nadalje, puno mikroprocesora i logičkih računala imaju ugrađeno PWM upravljanje što olakšava implementaciju. PWM je zapravo digitalan način generiranja analognih signala. PWM signal je digitalan jer u svakom trenutku vremena izlazni signal ili ima maksimalnu vrijednost ili minimalnu vrijednost. Izlazna analogna vrijednost ovisi o ponavljanju serija „on“ ili „off“ impulsa. „On-time“ je vrijeme u kojem vlada impuls maksimalne vrijednosti, a „off-time“ je vrijeme u kojem vlada minimalna vrijednost izlaznog napona ili struje unutar ciklusa. Ako je dana dovoljna širina pojasa, svaka analogna vrijednost može se generirati pomoću PWM-a. Vrlo bitna stavka PWM upravljanja je i frekvencija modulacije. Npr. ako želimo upravljati rasvjetom sa 50% radnog ciklusa pri frekvenciji od 10Hz to znači da je rasvjeta upaljena 50 ms, a zatim ugašena 50 ms što i vizualno daje do znanja da je frekvencija modulacije premala. Većina tereta, bilo induktivnih ili kapacitivnih, radi ispravno na puno većoj frekvenciji modulacije od 10 Hz. Tipična vrijednost je od 1kHz do 200kHz. Prednosti PWM-a su ekonomičnost, ušteda prostora te otpornost na smetnje [17-18]. U radu PWM se koristi za upravljanje proporcionalnim ventilom pri regulaciji protoka medija u radijator.

3.5 Upravljačka pločica

Upravljačka pločica je poveznica između instalacija u prostoriji i PLC-a. Pločica ima svoje posebno izmjenično napajanje od 220V (kućna mreža), te stalni napon od 5V dobiven preko stabilizatora napona. Na PLC se preko određenog broja žica odvode samo električne veličine bitne za upravljački program. Svaka prostorija ima svoju zasebnu upravljačku pločicu sa relejima za upravljanje elementima rasvjete i motora roleta te elektronike za čitanje stanja

termistora. Zasebnim upravljačkim pločicama postiže se modularna izvedba upravljanja pametnom kućom tj. moguće je proširivanje broja prostorija dodavanjem upravljačkih pločica i potrebnih elemenata. Pritom je samo potrebno pripaziti na dostupan broj ulaza ili izlaza na PLC-u. Na upravljačkoj pločici se uz releje i sklop za čitanje elektronike nalaze i sabirnice ili redne stezaljke. Preko releja se uključuju ili isključuju pojedine komponente našeg sustava. Zavojnicom releja upravlja se preko izlaza PLC-a, dok su kontakti spojeni na element i na izvor kućne mreže. Na slici 3-7 prikazan je relej koji je moguće iskoristiti na upravljačkoj pločici.

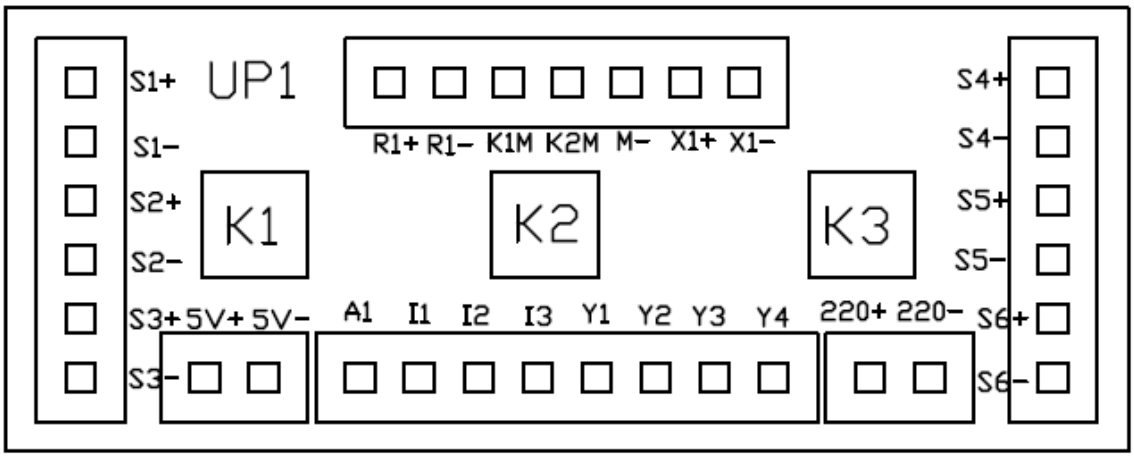
Slika 3-6 Relej upravljačke pločice

Po rubovima pločice nalaze se redne stezaljke (*engl. Terminal block connectors*) pomoću kojih se na vrlo jednostavan i brzi način mogu spojiti ožičenja našeg sustava. Redne stezaljke na sebi imaju utor u koji se stavi žica očišćene izolacije, te se pomoću vijka s gornje strane čvrsto pritegne odvijačem kako bi se osigurao dobar spoj. Na slici 3-8 prikazan je jedan tip rednih stezaljki koji omogućuje direktno lemljenje na upravljačku pločicu preko nožica s donje strane.

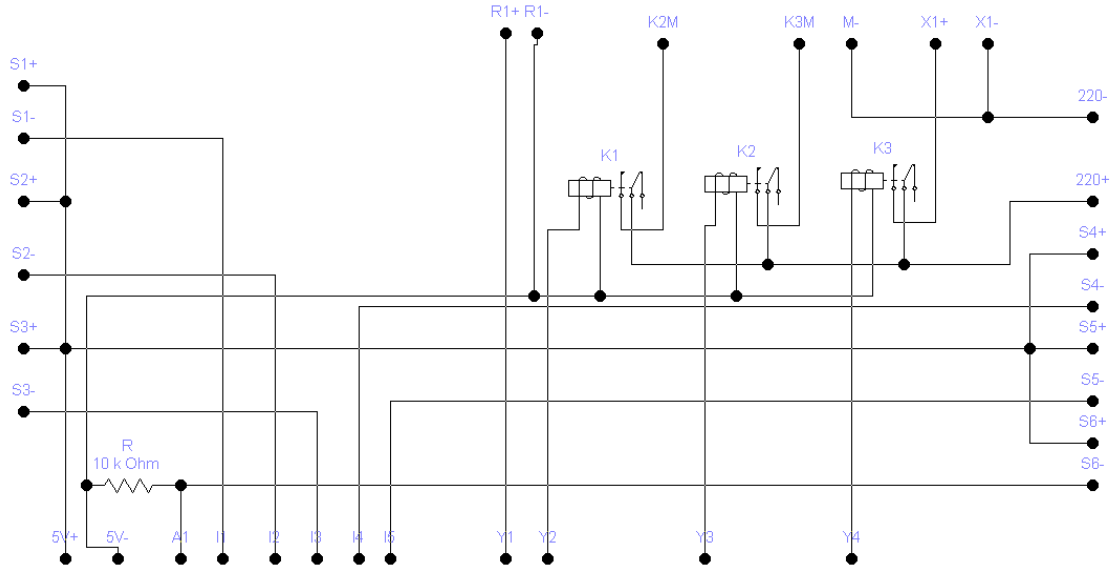
Slika 3-7 Redne stezaljke

3.5.1 Model upravljačke pločice

Na slici 3-8 prikazan je predložen izgled upravljačke pločice. U tablici 3 nalaze se objašnjenja funkcija releja, a na slici 3-9 vidimo električnu shemu upravljačke pločice.



Slika 3-8 Skica elemenata i raspored priključaka na upravljačkoj pločici



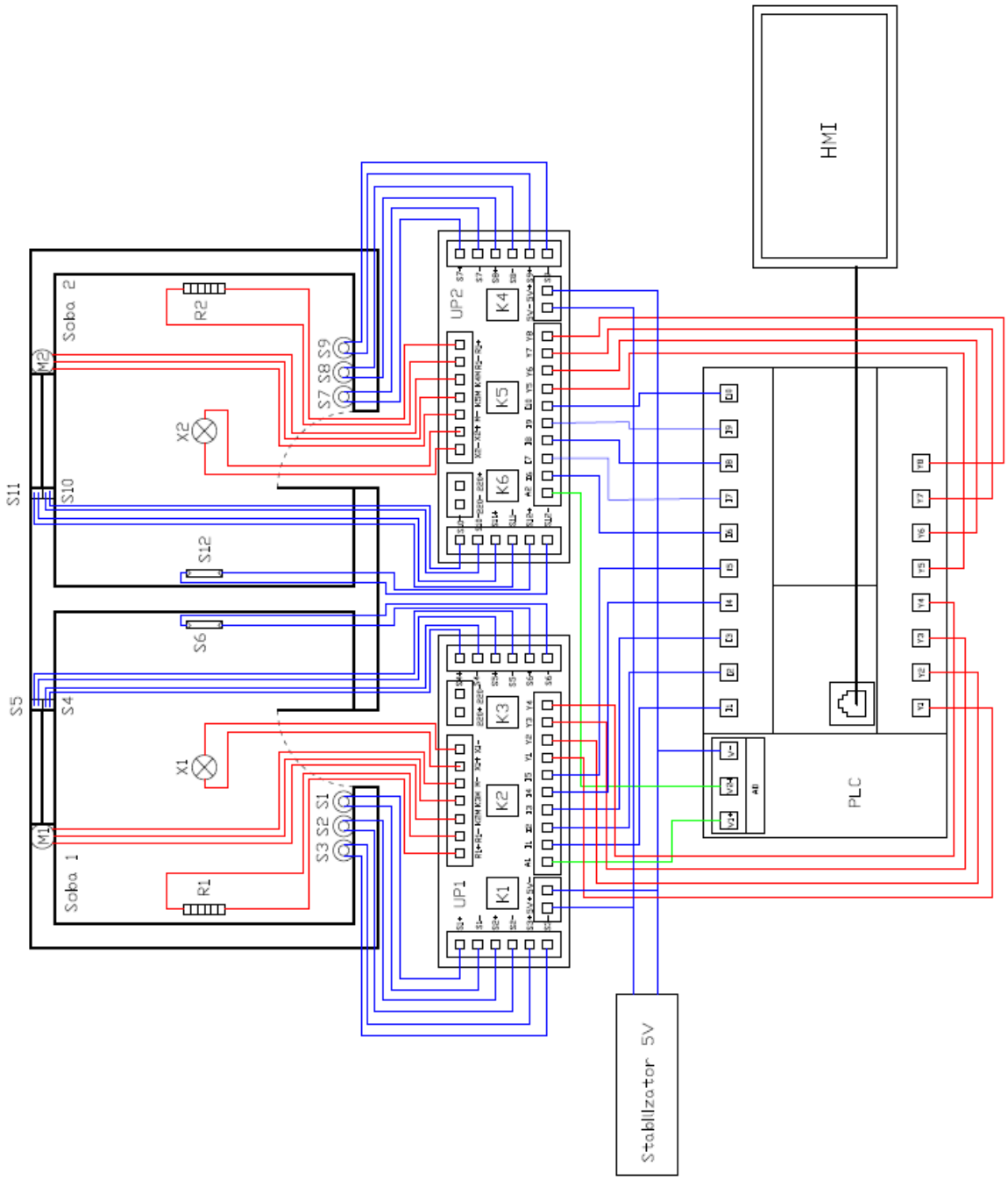
Slika 3-9 Električna shema upravljanja pločice

Tablica 3 Oznake i funkcije releja

Oznaka releja	Funkcija releja za prostoriju 1
K1	Motor , smjer rolete prema dolje
K2	Motor, smjer rolete prema gore
K3	Rasvjeta

3.6 Električna shema spajanja modela

Komponente sustava spajaju se kako je prikazano na slici 3-10.



Slika 3-10 Električna shema spajanja modela

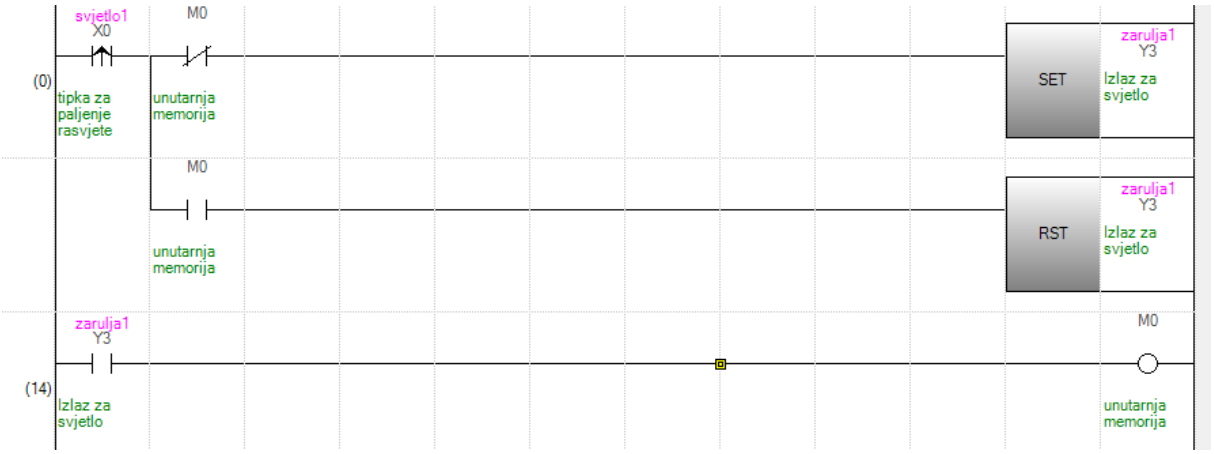
4 Upravljački program

4.1 Sustav logičkog upravljanja

Za potrebe programiranja koristimo već spomenuti programski paket „GX Works3“. Sustav logičkog upravljanja izveden je preko FBD/LD programiranja. FBD/LD je u suštini spoj dva načina programiranja odnosno kombinacija funkcijskih blokova i ladder logike zajedno sa ST programiranjem čiji je kod sličan kodu C jezika. Struktura programa dijeli se na tri razine. Prva razina je glavni program koji se izvodi ciklički u zadanom vremenu. Druga razina su „interupt“ programi koji se izvršavaju unutar vremenskog intervala. Treća razina su funkcijski blokovi u kojima su zapisani manji programi, a u glavnom programu se koriste kao jedan element programske rutine [15]. U sljedećim točkama zasebno se opisuju dijelovi glavnog programa za upravljanje pojedinih sustava. Zbog minimalnih razlika (samo u adresama ulaza i izlaza) prikazuje se samo program za jednu prostoriju.

4.1.1 Upravljanje rasvjetom

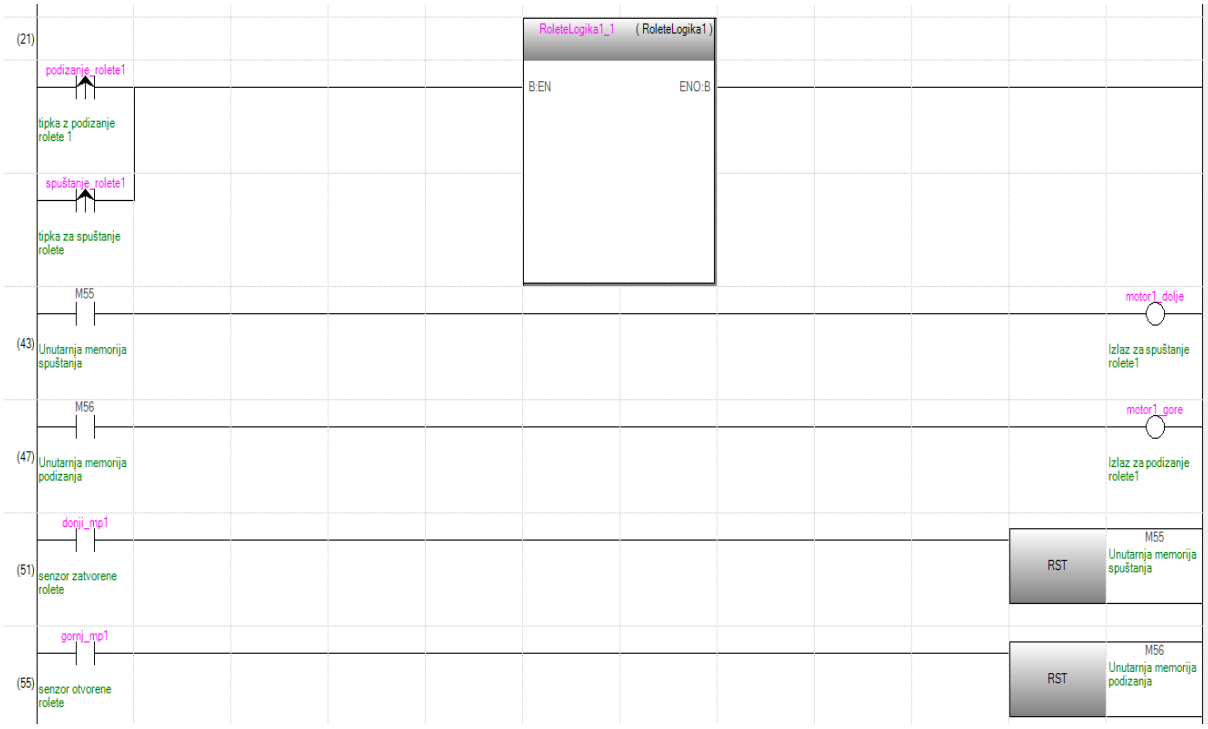
Upravljanje rasvjetom ostvaruje se preko tipkala. Stanje tipkala se prenosi samo tijekom rastućeg brida signala. Signal putuje samo preko mirnog kontakta M0 do instrukcije Set gdje se izlaz Y3 postavlja u visoko stanje. Kada je Y3 u visokom stanju tada i unutarnja memorija poprima visoko stanje. Prilikom sljedećeg pritiska na tipkalo, signal može proći samo preko radnog kontakta unutarnje memorije M0 te tako resetira izlaz Y3 u nisko stanje a time također i stanje unutarnje memorije. Ovim programom postiže se naizmjenično paljenje i gašenje rasvjete jednim tipkalom što je zapravo logika T bistabila.



Slika 4-1 Program za upravljanje rasvjetom

4.1.2 Upravljanje roletom

Na slici 4-2 prikazan je upravljački program rolete. Upravljanje roletom postiže se pomoću dva tipkala. Tipkama za podizanje odnosno spuštanje rolete. Pritiskom na jedno od tipkala program ulazi u funkcijski blok „Roletna logika“. Unutar funkcijskog bloka ovisno o pritisnutom tipkalu program ulazi u uvjetovane potprograme. Uvjete podizanja i spuštanja roleta određuju stanja unutarnjih memorija podizanja i spuštanja. U programu se nalaze i kontakti senzora krajnjih položaja koji resetiraju stanje pripadajućih izlaza. Na slici 4-3 prikazan je program funkcijskog bloka pisan u ST jeziku.



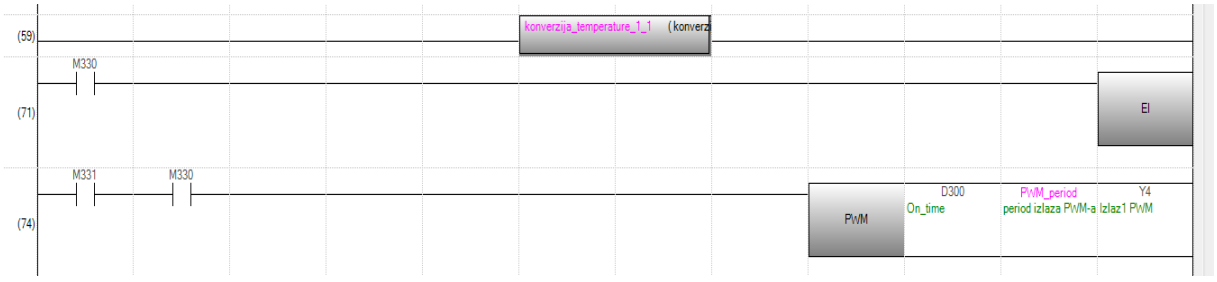
Slika 4-2 Program za upravljanje roletom

```
IF spuštanje_rolete2 THEN // Uvjet pritiska tipke za spuštanje

    IF NOT M57 AND NOT M58 THEN   //Ako roleta ne putuje ni gore ni dolje , roleta može krenuti prema dolje
        M57 := TRUE;
        M58 := FALSE;



        ELSIF M58 AND NOT M57 THEN   // Ako roleta ide gore i ne ide dolje, onda sve staje na pritisak tipke za dolje
        M57 := FALSE;
        M58 := FALSE;

        ELSIF M57 AND NOT M58 THEN   // Ako roleta ide dolje i opet se stisne tipka za dolje, onda ne radi nista
        ;

    END_IF;

END_IF;


IF podizanje_rolete2 THEN //Uvjet pritiska tipke za podizanje


    IF NOT M57 AND NOT M58 THEN   // Ako roleta ne putuje ni gore ni dolje , roleta može krenuti prema gore
        M57 := FALSE;
        M58 := TRUE;



        ELSIF M57 AND NOT M58 THEN   //Ako roleta ide dolje i ne ide gore, onda sve staje na pritisak tipke za gore
        M57:= FALSE;
        M58 := FALSE;

        ELSIF NOT M57 AND M58 THEN   // Ako roleta ide gore i opet se stisne tipka za gore onda ne radi nista
        ;


    END_IF;

END_IF;
```

Slika 4-3 Program unutar funkcijskog bloka RoleteLogika

4.1.3 Regulacija temperature

Regulacija temperature postiže se pomoću PI regulatora i PWM-a. PI regulator je implementiran unutar „interupt“ rutine. Interupt rutinu potrebno je omogućiti unutar glavnog programa, a to se ostvaruje uvjetom na unutarnjoj memoriji M330 koji se nalazi na zaslonu temperature (slika 4-11). Uvjet zapravo služi za paljenje ili gašenje regulacije temperature unutar prostorije. Na slici 4-4 nalazi se dio glavnog programa za regulaciju temperature.



Slika 4-4 Program regulacije temperature

Nadalje prije računanja greške unutar regulatora potrebno je prilagoditi signal senzora u mjernu jedinicu referentne veličine. To se postiže unutar funkcijskog bloka „konverzija temperature“. Na slici 4-4 prikazan je kod za prilagodbu signala senzora unutar funkcijskog bloka konverzije temperature.

```
INT2FLT(TRUE,SD6021,Vo1);
Rntc1:=Rconst*((2000.0/Vo1)-1.0);

LogRntc1:=LN(Rntc1);

T_kelvin1:=(1.0/(konst_1+konst_2*LogRntc1+konst_3*LogRntc1*LogRntc1*LogRntc1));

T_celzij1:=T_kelvin1-273.15;

mj_temperatura1:=T_celzij1;
```

Slika 4-5 ST kod unutar funkcijskog bloka konverzije temperature

Unutar prikazanog koda vrši se konverzija signala pomoću izvedenih jednadžbi (4) i (1). Dobiven rezultat sprema se kao mjerena temperatura prostorije i šalje u regulator. PI regulator također je isprogramiran pomoću ST koda. Unutar koda regulatora računa se regulacijsko odstupanje i postavna veličina. Nakon izračuna postavne veličina sprema se kao „on_time“ PWM-a. Na slici 4-6 prikazan je ST kod PI regulatora unutar interupt programa.

```
//PI regulator
    Greska := temperatura_1 - mj_temperatura1;   // greska
    proportional := KP*Greska;
    integral := KI*(integral + Greska*DeltaTime);
    U_PI := proportional+integral;
    // ograničenje izlaza
    IF U_PI > 20.0 THEN
        U_PI := 20.0;
        integral := 20.0 - proportional;
    ELSIF U_PI <= 0.0 THEN
        U_PI := 0.0;
        integral := proportional;
    END_IF;
    //Omogućavanje funkcije PWM-a
    IF U_PI>0.01 THEN
        M331:=1;
        ELSE
        M331:=0;
    END_IF;

FLT2INT(TRUE,U_PI,D300); //Spremnaje vrijednosti u on_time PWM-a
```

Slika 4-6 ST kod PI regulatora

4.1.4 Oznake korištene unutar upravljačkog programa

Sljedeće slike prikazuju iznose, adrese, oblik podataka i objašnjenja korištenih oznaka unutar upravljačkog programa.

	Label Name	Data Type		Class	Assign (Device/Label)	Initial Value	Constant	Comment
1	svjetlo1	Bit	...	VAR_GLOBAL	X0			tipka za paljenje svjetla 1
2	spuštanje_rolete1	Bit	...	VAR_GLOBAL	X1			tipka za spuštanje rolete 1
3	podizanje_rolete1	Bit	...	VAR_GLOBAL	X2			tipka za podizanje rolete 1
4	donji_mp1	Bit	...	VAR_GLOBAL	X3			donji senzor krajnjeg položaja 1
5	gornj_mp1	Bit	...	VAR_GLOBAL	X4			gornji senzor krajnjeg položaja 1
6	svjetlo2	Bit	...	VAR_GLOBAL	X5			tipka za paljnje svjetla 2
7	spuštanje_rolete2	Bit	...	VAR_GLOBAL	X6			tipka za spuštanje rolete 2
8	podizanje_rolete2	Bit	...	VAR_GLOBAL	X7			tipka za podizanje rolete 2
9	donji_mp2	Bit	...	VAR_GLOBAL	X10			donji senzor krajnjeg položaja 2
10	gornji_mp2	Bit	...	VAR_GLOBAL	X11			gornji senzor krajnjeg položaja 2
11	mj_temperatura1	FLOAT [Single Precision]	...	VAR_GLOBAL	D402			Mjerena temperatura prostorije 1
12	temperatura_1	FLOAT [Single Precision]	...	VAR_GLOBAL	D404			Zadana temperatura sobe 1
13	mj_temperatura2	FLOAT [Single Precision]	...	VAR_GLOBAL	D406			Mjerena temperatura prosotrije 2
14	temperatura2	FLOAT [Single Precision]	...	VAR_GLOBAL	D408			Zadana temperatura sobe 2
15	ventil1	Bit	...	VAR_GLOBAL	Y4			izlaz za proporcionalni ventil 1
16	motor1_dolje	Bit	...	VAR_GLOBAL	Y1			izlaz za smjer motora dolje 1
17	motor1_gore	Bit	...	VAR_GLOBAL	Y2			izlaz za smjer motora gore 1
18	zarulja1	Bit	...	VAR_GLOBAL	Y3			izlaz za svjetlo 1
19	zarulja2	Bit	...	VAR_GLOBAL	Y0			izlaz za svjetlo 2
20	motor2_gore	Bit	...	VAR_GLOBAL	Y5			izlaz za smjer motora gore 2
21	motor2_dolje	Bit	...	VAR_GLOBAL	Y6			izlaz za smjer motora dolje 2
22	ventil2	Bit	...	VAR_GLOBAL	Y7			izlaz za ventil 2
23	PWM_period	Word [Signed]	...	VAR_GLOBAL_CONST			20	period izlaza PWM-a

Slika 4-7 Globalne oznake upravljačkog programa

	Label Name	Data Type		Class	Initial Value	Constant	Comment
1	konst_1	FLOAT [Single Precision]	...	VAR_CONSTANT		2.1085e-3	Steinhart–Hart koeficijent
2	konst_2	FLOAT [Single Precision]	...	VAR_CONSTANT		0.79793e-4	Steinhart–Hart koeficijent
3	konst_3	FLOAT [Single Precision]	...	VAR_CONSTANT		6.5351e-7	Steinhart–Hart koeficijent
4	Rconst	FLOAT [Single Precision]	...	VAR_CONSTANT		10000	Otpor
5	Rntc1	FLOAT [Single Precision]	...	VAR			NTC otpor
6	LogRntc1	FLOAT [Single Precision]	...	VAR			logaritam otpora
7	T_celzij1	FLOAT [Single Precision]	...	VAR			temperatura u celzijima
8	T_kelvin1	FLOAT [Single Precision]	...	VAR			temperatura u kelvinima
9	Vo1	FLOAT [Single Precision]	...	VAR			napon AD konverzije

Slika 4-8 Lokalne oznake konverzije temperature

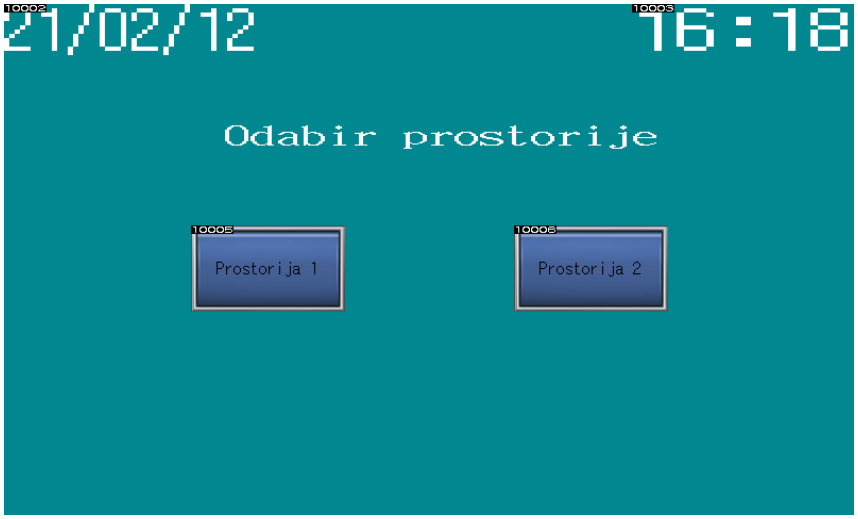
	Label Name	Data Type		Class	Initial Value	Constant	Comment
1	proportional	FLOAT [Single Precision]	...	VAR			proporcionalno djelovanje
2	Greska	FLOAT [Single Precision]	...	VAR			regulajsko odstupanje
3	integral	FLOAT [Single Precision]	...	VAR			integralno djelovanje
4	PWM_time_on	FLOAT [Single Precision]	...	VAR			on time PWM-a
5	DeltaTime	FLOAT [Single Precision]	...	VAR_CONSTANT		0.02	trajanje interuppta
6	KP	FLOAT [Single Precision]	...	VAR_CONSTANT		20	proporcionalno pojačanje
7	KI	FLOAT [Single Precision]	...	VAR_CONSTANT		0.1	integralno pojačanje
8	U_PI	FLOAT [Single Precision]	...	VAR			postavna veličina

Slika 4-9 Lokalne oznake regulatora

4.2 Grafičko sučelje

Grafičko sučelje prema korisniku (HMI) programira se putem prethodno spomenutog alata GT Designer3. Pomoću grafičkog sučelja moguće je upaliti ili ugasiti svjetlo, podizati ili spuštati rolete ili upisati referencu temperature obje prostorije. To se postiže spajanjem prekidača i upisnih registara sa pojedinim kontaktima i registrima unutar glavnog programa PLC-a. Grafičko sučelje podijeljeno je na 9 funkcionalnih zaslona. Prvi je početni zaslon a ostali su podijeljeni po četiri za svaku prostoriju. Prebacivanje između zaslona postiže se funkcijskim tipkama. Pošto su zasloni i funkcije obje prostorije srodne, prikazat ćemo grafičko sučelje samo jedne prostorije.

Na slici 4-10 prikazan je početni zaslon na kojem se odabire prostorija u kojoj se želi obaviti neka radnja.

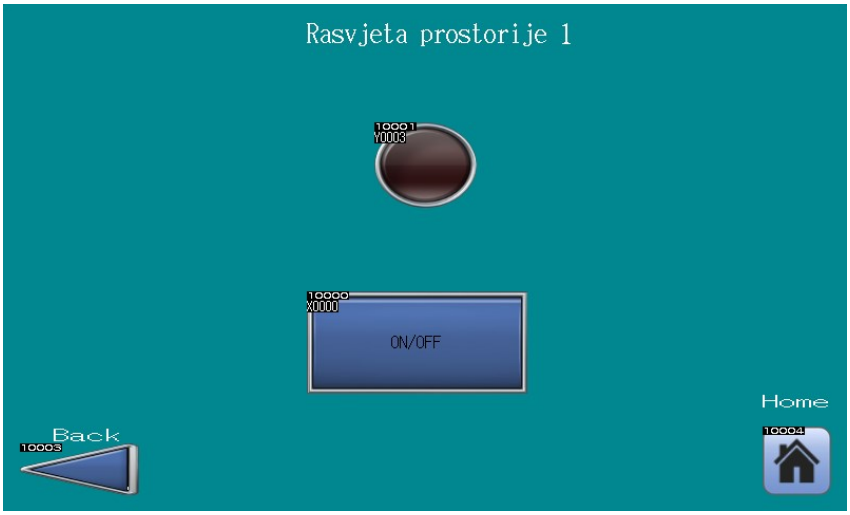


Slika 4-10 Početni zaslon

Nakon odabira prostorije prikazuje se zaslon prostorije (slika 4-11). Na zaslonu prostorije odabire se sustav kojim želimo upravljati.

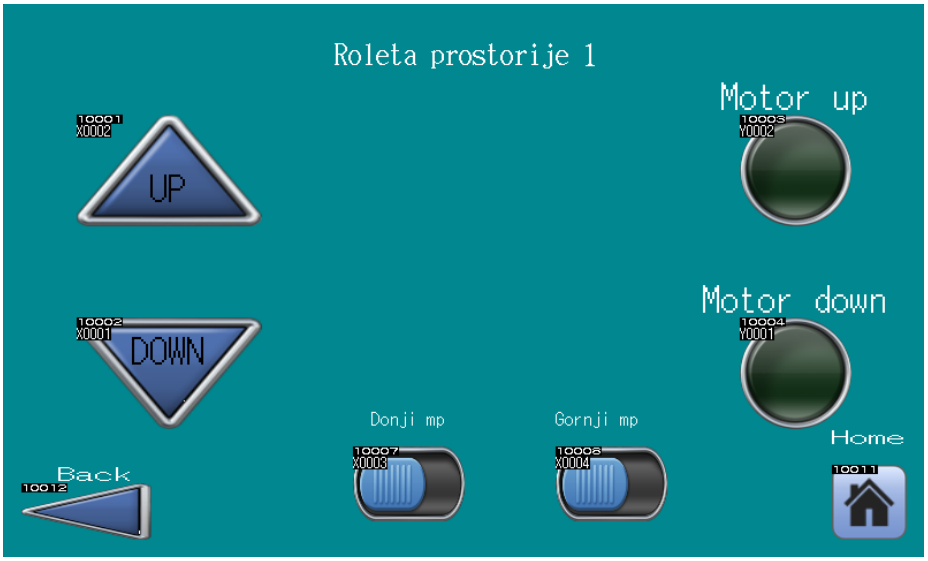
Slika 4-11 Zaslon prostorije 1

Ako smo odabrali sustav upravljanja rasvjetom otvara se zaslon rasvjete (slika 4-12). Na zaslonu rasvjete nalazi se tipka za paljenje ili gašenje svjetla. Na zaslonu se također nalazi i prikaz rasvjetnog tijela kako bi imali bolji uvid u stanje rasvjete.



Slika 4-12 Zaslon rasvjete

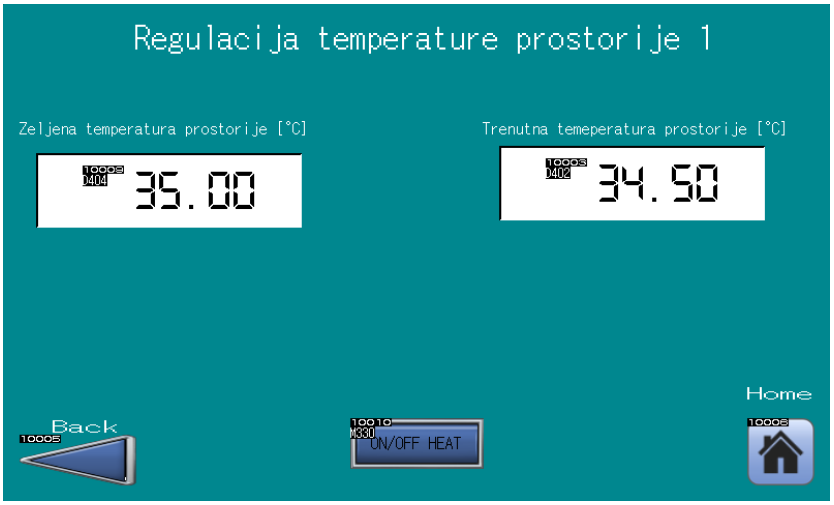
Ako smo odabrali sustav upravljanja roletom prikazuje se zaslon rolete (slika 4-13). Na zaslonu rolete nalaze se tipke za podizanje i spuštanje rolete. Također na ekranu se nalaze i svjetleće diode koje signaliziraju stanje pojedinog izlaza. U svrhu ispitivanja funkcionalnosti programa postavljeni su virtualni senzori krajnjih položaja roleta, „Donji mp“ i „Gornji mp“ koji se aktiviraju ručno.



Slika 4-13 Zaslon rolete

Nadalje ako smo odabrali sustav regulacije temperature prikazuje se zaslon temperature (slika 4-14). Na zaslonu se nalazi prikaz trenutne temperature unutar prostorije te prostor za upis

željene temperature unutar nje. Također se na zaslonu nalazi tipka ON/OFF HEAT pomoću koje možemo potpuno isključiti grijanje unutar prostorije.



Slika 4-14 Zaslon regulacije temperature

5 Mrežno sučelje temeljeno na TCP/IP protokolu

5.1 Modbus

Modbus je protokol za podatkovnu komunikaciju koju je izvorno objavio Modicon (danas Scheinder Eletric) 1979. godine za upotrebu komunikacije u svojih programibilnim logičkim kontrolerima. Modbus je praktički postao standardni komunikacijski protokol sa širokom primjenom u povezivanju industrijskih elektroničkih uređaja. Modbus je postao popularan u industrijskim okruženjima jer je otvoreno dostupan i moguće ga je koristit bez naknade. Razvijen je za industrijske aplikacije i relativno je jednostavan za primjenu i održavanje u usporedbi s drugim standardima. Postavlja vrlo malo ograničenja, osim na veličinu datagrama (paketa) na format podataka koji se prenose. Modbus protokol koristi znakovne serijske komunikacijske linije, Ethernet ili paket internetskih protokola kao transportni sloj. Modbus podržava komunikaciju i sa više uređaja povezanih na isti kabel ili Ethernet mrežu. Na primjer, može postojati uređaj koji mjeri temperaturu, te drugi uređaj za mjerenje vlažnosti spojen na isti kabel, oba prenoseći mjerenja na isto računalo. Modbus se često koristi za povezivanje računala za nadzor postrojenja /sustava s udaljenom terminalnom jedinicom (*RTU*) unutar sustava nadzora i kontrole te sakupljanja podataka (*SCADA*). Kao primjer u elektroenergetskoj industriji. Mnoge vrste podataka dobivaju naziv iz industrijskog upravljanja tvorničkim

uređajima, poput upotrebe Ladder logike u pogonskim relejima. Prema tome jedan fizički izlaz naziva se zavojnica, a jedan fizički ulaz naziva se diskretni ulaz ili kontakt [19].

Ograničenja:

- Modbus je dizajniran krajem 1970-tih za komunikaciju s PLC-ima pa je broj različitih podataka ograničen razumijevanju PLC-a toga vremena. Veliki binarni objekti nisu podržani.
- Ne postoji standardni način da čvorište pronađe opis podatkovnog objekta.
- Modbus je nadređeni/podređeni protokol pa ne postoji način da terenski uređaj prijavi iznimku osim preko Ethernet TCP/IP protokola tzv. „open modbus“. Glavno čvorište mora rutinski anketirati svaki terenski uređaj i tražiti promjene u podacima. To troši mrežno vrijeme i smanjuje propusnost podataka.
- Modbus je ograničen na adresiranje 247 uređaja na jednoj podatkovnoj vezi, što ograničava broj terenskih uređaja spojenih sa roditeljskom stanicom.
- Modbus prijenos mora biti susjedno slijedan.
- Modbus protokol ne pruža sigurnost od neovlaštenih naredbi ili presretanja podataka [19].

Verzije Modbus protokola postoje za serijsku komunikaciju, Ethernet i protokole koji podržavaju IP. Postoje mnoge vrste Modbus protokola poput [19]:

- Modbus RTU.
- Modbus ASCII.
- Modbus TCP/IP ili Modbus TCP.
- Modbus preko UDP-a.
- Modbus Plus (Modbus+, MB+ ili MBP).
- Pemex Modbus.
- Enron Modbus.

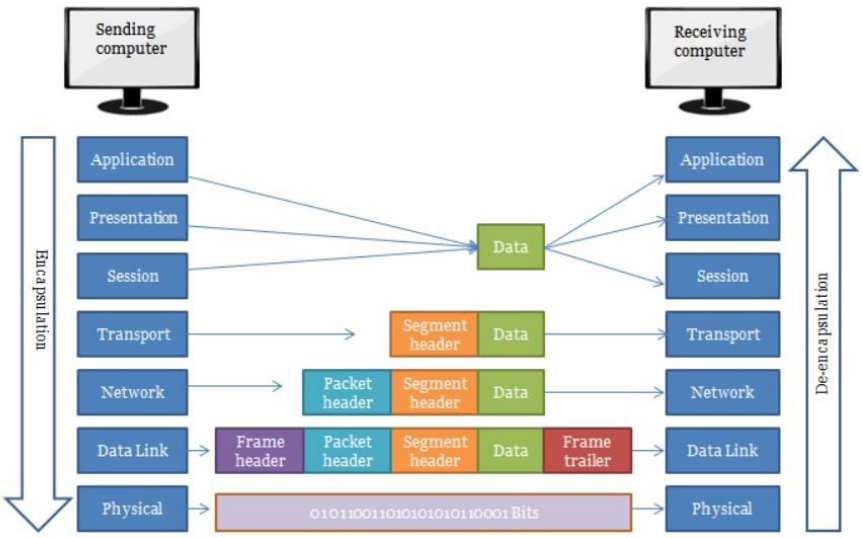
5.2 TCP/IP protokol

Internetski protokolni paket je konceptualni model i skup komunikacijskih protokola koji se koriste na internetu i sličnim računalnim mrežama. Općenito je poznat kao TCP/IP protokol jer se temelji na protokolu za kontrolu prijenosa (TCP) i internetskog protokola (IP). Prijašnje verzije bile su poznate kao i Department of Defense protokol jer je njegov razvoj nastao od

strane DARPA-e. Paket internetskih protokola omogućuje „end to end“ komunikaciju koja određuje kako se podaci trebaju pakirati, adresirati, prenositi, usmjeravati i primati [20]. Spomenuti model omogućuje povezivanje računala neovisno o tipu računalne opreme i operacijskih sustava koje koriste te je prihvaćen kao standard. Zbog jednostavnog načina adresiranja TCP/IP protokol omogućuje povezivanje i komunikaciju svih uređaja neovisno o tipu mrežne opreme na fizičkoj razini i razini prijenosnog medija. Ovaj skup protokola jedan je od najčešće korištenih mrežnih protokola jer omogućuje povezivanje mreža sa različitim arhitekturama [21].

5.2.1 Arhitektura TCP/IP protokola

TCP/IP protokol opisuje postupak pri kojem se informacija s jednog računala prenosi na drugi. Format informacije mijenja se kod prelaska na niži nivo. Podaci višeg sloja se enkapsuliraju. Na podatke višeg sloja nadodaje se zaglavlje sloja kroz koje je podatak prošao. Enkapsulacija, koja je prikazana na slici 5-1, na podatak dodaje informacije važne protokolu kako bi taj podatak uspješno i u cijelosti stigao do ciljanog računala ili mreže. TCP/IP protokol sastavljen je od četiri funkcionalna sloja. Prvi korak je enkapsulacija podataka unutar transportnog sloja TCP/IP protokola. Unutar njega se na zaprimljene informacije dodaje zaglavlje toga sloja, a novonastali podatak naziva se segment. Kreirani segment prosljeđuje se mrežnom sloju na daljnju obradu. Unutar mrežnog sloja stvara se zaglavlje za pristigli segment. Zaglavlje mrežnog sloja sadrži informacije koje su potrebne za adresiranje i usmjeravanje. Spajanjem zaglavlja mrežnog sloja i segmenta nastaje paket. Paket se prosljeđuje dalje podatkovnom sloju. Paket zajedno sa zaglavljem podatkovnog sloja naziva se „frame“. Fizički sloj prima „frameove“ podatkovnog sloja te im mijenja oblik u pogodan za prijenos određenim prijenosnim medijem. Nakon dolaska podataka na zadano odredište vrši se postupak obrnut od navedenog. Unutar svakog sloja uklanja se zaglavlje tog sloja i tako redom do izvornog oblika [21].



Slika 5-1 Enkapsulacija podataka [22]

5.2.2 Aplikacijski sloj

Aplikacijski sloj je najviši sloj TCP/IP modela koji uključuje protokole koje koristi većina aplikacija za pružanje korisničkih usluga ili razmjenu podataka aplikacija putem mrežnih veza uspostavljenih protokolima niže razine. To može uključivati neke osnovne usluge mrežne podrške, poput protokola usmjeravanja i konfiguracije domaćina [20]. Primjeri tih protokola su [21]:

- „Protokol za prijenos informacija na Internetu (*engl. Hypertext Transfer Protocol*, HTTP)“,
- „protokol za siguran prijenos informacija na Internetu (*engl. Hypertext Transfer Protocol Secure*, HTTPS)“,
- „protokol za preuzimanje i postavljanje datoteka na poslužitelj (*engl. File Transfer protocol*, FTP)“,
- „protokol namijenjen za povezivanje na udaljeni uređaj, te izvršavanje naredba i funkcija preko mreže (*engl. Secure Shell*, SSH)“,
- „protokol zadužen za doznavanje IP adresa iz imena računala i obratno (*engl. Domain Name System*, DNS)“,
- „protokol zadužen za upravljanje aktivnim mrežnim uređajima (*engl. Simple Network Managment Protocol,* SNMP)“,
- „protokol za slanje elektroničke pošte (*engl. Simple Mail Transfer Protocol,* SMTP)“,

- „protokol za sinkronizaciju satova na mrežnim uređajima i računalima (*engl. Network Time Protocol,* NTP)“.

Podaci kodirani prema protokolima aplikacijskog sloja inkapsulirani su u jedinice protokola transportnog sloja koje zauzvrat koriste protokole nižeg sloja za ostvarenje stvarnog prijenosa podataka. TCP/IP model ne uzima u obzir specifičnosti oblikovanja i prezentiranja podataka i ne definira dodatne slojeve između aplikacijskog i transportnog sloja. Takve funkcije spadaju u područje knjižnica i sučelja za programiranje aplikacija. Protokoli aplikacijskog sloja općenito tretiraju protokole transportnog sloja (i nižeg) kao crnu kutiju. Ti protokoli pružaju stabilnu mrežnu vezu preko koje se može komunicirati, iako su aplikacije obično svjesne ključnih kvaliteta veze transportnog sloja kao što su IP adrese krajnje točke i port brojevi. Protokoli aplikacijskog sloja često su povezani s određenim aplikacijama klijent-server, a uobičajene usluge imaju dobro poznate brojeve priključaka. Transportni sloj i niži slojevi ne vode brigu o specifičnostima protokola aplikacijskog sloja. Ruteri i mrežni preklopnici obično ne ispituju inkapsulirani promet već samo pružaju kanal za njega. Nadalje, TCP/IP model razlikuje korisničke protokole od protokola podrške. Protokoli za podršku pružaju usluge sustavu mrežne infrastrukture. Korisnički protokoli koriste se za stvarne korisničke aplikacije. Na primjer, FTP je korisnički protokol, a DNS protokol podrške [20].

5.2.3 Transportni sloj

Transportni sloj uspostavlja osnovne podatkovne kanale koje aplikacije koriste za razmjenu podataka specifičnih za zadatak. Sloj uspostavlja povezanost od domaćina do domaćina u obliku usluga prijenosa poruka „end to end“ koji su neovisni o osnovnoj mreži i neovisni o strukturi korisničkih podataka i logistici razmjene informacija. Povezivost na transportnom sloju može se kategorizirati kao orijentirana na vezu, implementirana u TCP-u ili bez veze, implementirana u UDP-u. Protokoli u ovom sloju mogu pružati kontrolu pogrešaka, segmentaciju, kontrolu protoka, kontrolu zagušenja i adresiranje aplikacija (brojevi porta). U svrhu pružanja specifičnih procesnih kanala za prijenos aplikacija, sloj uspostavlja koncept mrežne luke. Ovo je numerirana logička konstrukcija dodijeljena posebno za svaki od komunikacijskih kanala koji su potrebni aplikaciji. Za mnoge vrste usluga ti su brojevi porta standardizirani tako da klijentska računala mogu pristupiti određenim uslugama serverskog računala bez uključivanja SDP protokola ili mape usluga. Budući da IP pruža samo dostavu najveće propusnosti, zato neki protokoli transportne razine pružaju pouzdanost. TCP je protokol usmjeren na vezu koja rješava brojne probleme s pouzdanošću u pružanju toka bajtova:

- podaci stižu u redu,
- podaci imaju minimalnu pogrešku (tj. ispravnost),
- duplicirani podaci se odbacuju,
- izgubljeni ili odbačeni paketi se ponovno šalju,
- uključuje kontrolu zagušenja prometa.

Noviji protokol za kontrolu prijenosa (*SCTP*) također je pouzdan, orijentiran na povezivanje transportnog mehanizma. Orijentiran je na tok poruka, a ne na protok bajtova poput TCP-a i pruža višestruke multipleksirane tokove kroz jednu vezu. Također pruža više mrežnu podršku, u kojoj kraj veze može biti predstavljen s više IP adresa (koje predstavljaju više fizičkih sučelja), tako da ako jedna ne uspije, veza se ne prekida. U početku je razvijen za telefonske aplikacije. Pouzdanost se također može postići pokretanjem IP-a preko pouzdanog protokola za podatkovnu vezu, kao što je kontrola podataka visoke razine (*HDLC*). „User Datagram Protocol“ (UDP) je datagram protokol bez veze. Poput IP-a i to je nepouzdan protokol koji se orijentira na najbolju propusnost. Pouzdanost se rješava otkrivanjem pogrešaka pomoću algoritma kontrolne sume. UDP se obično koristi za aplikacije kao što su streaming mediji (audio, video, glas preko IP-a itd.) gdje je pravovremeni dolazak važniji od pouzdanosti ili za jednostavne programe za upit/odgovor kao što su DNS pretraživanja, pri čemu je potreba postavljanja pouzdane veze nerazmjerno velika. Transportni protokol u stvarnom vremenu (RTP) je datagram protokol koji se koristi preko UDP-a i dizajniran je za podatke u stvarnom vremenu kao što su streaming mediji. Aplikacije na bilo kojoj datoj mrežnoj adresi razlikuju se po TCP ili UDP priključku. Prema dogovoru, određene dobro poznate luke povezane su s određenim aplikacijama [20].

5.2.4 Internet sloj

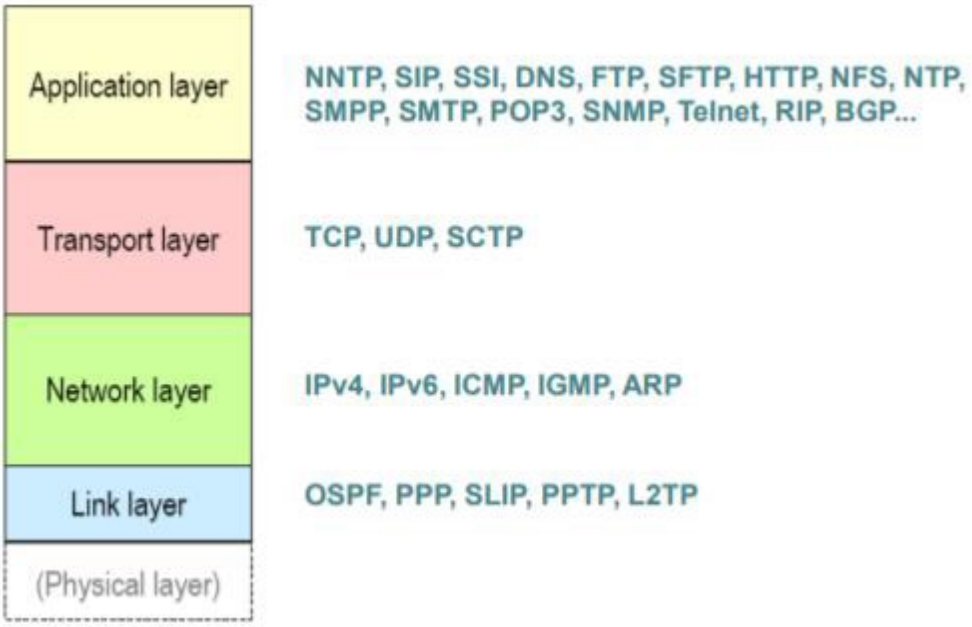
Internetsko umrežavanje zahtijeva slanje podataka s izvorne mreže na odredišnu mrežu. Taj se proces naziva „routing“ ili usmjeravanje, a slijedi ga adresiranje i identifikacija domaćina pomoću hijerarhijskog sustava IP adresiranja. Internetski sloj pruža nepouzdanu mogućnost prijenosa datagrama između domaćina smještenih na potencijalno različitim IP mrežama i to prosljeđivanjem datagrama na odgovarajući ruter za daljnje prenošenje na odredište. Internetski sloj je odgovoran za slanje paketa preko potencijalno više mreža. Ovom funkcionalnošću internetski sloj omogućuje internetsko umrežavanje, međusobno povezivanje različitih IP mreža i u osnovi uspostavlja Internet. Internetski sloj ne razlikuje različite protokole transportnog sloja. IP prenosi podatke za niz različitih protokola gornjeg sloja. Svaki se od ovih protokola identificira jedinstvenim brojem protokola [20]. Za uspješno izvršavanje navedenih

zadataka, internet sloj mora poznavati strukturu podmreže, te put kroz istu podmrežu. Neki od glavnih protokola koji egzistiraju unutar internet sloja su [21]:

- standardni internetski protokol, sa funkcijama adresiranja i usmjeravanja (*engl. Internet Protocol,* IP),
- protokol koji reagira kod neočekivanih događaja (*engl. Internet Control Message Protocol*, ICMP),
- protokol za pronalazak veze između IP adresa i fizikalnih veza (*engl. Address Resolution Protocol*, ARP) i (*engl. Reverse Address Resolution Protocol*, RARP),
- protokol za dopuštanje grupnog adresiranja (*engl. Internet Protocol Managment Group*, IGMP).

5.2.5 Sloj povezivanja

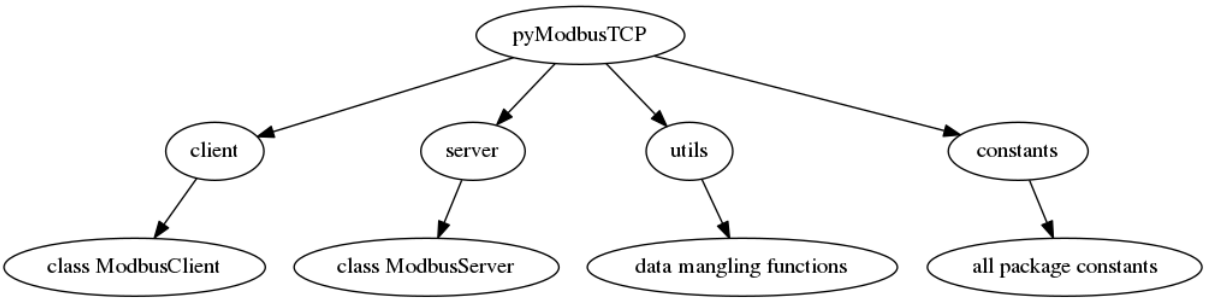
Protokoli sloja povezivanja djeluju u okviru lokalne mrežne veze na koju je priključen domaćin. Taj se režim na TCP/IP jeziku naziva poveznicom i najniži je funkcionalan sloj paketa. Poveznica uključuje sve domaćine dostupne bez prelaska rutera. Veličina veze stoga se određuje mrežnim dizajnom hardvera. U principu, TCP/IP dizajniran je da bude neovisan o hardveru i može se implementirati povrh gotovo svake tehnologije sloja povezivanja. To uključuje ne samo hardverske implementacije, već i virtualne slojeve povezivanja poput virtualnih privatnih mreža i mrežnih tunela. Sloj povezivanja koristi se za premještanje paketa između sučelja internetskog sloja dva različita domaćina na istoj vezi. Procesima prijenosa i primanja paketa na vezi može se upravljati u upravljačkom programu uređaja za mrežnu karticu, kao i u firmwareu ili pomoću specijaliziranih skupa čipova. Oni izvode funkcije, poput pripreme paketa internetskog sloja za prijenos i konačno prenose okvire na fizički sloj preko prijenosnog medija. TCP/IP model uključuje specifikacije za prevođenje metoda mrežnog adresiranja korištenih u internetskom protokolu na adrese sloja veza, poput adresa za kontrolu pristupa medijima (MAC). Međutim, za sve ostale aspekte ispod te razine implicitno se pretpostavlja da postoje i nisu izričito definirani u TCP/IP modelu [20]. Na slici 4-2 prikazani su funkcionalni slojevi sa korištenim protokolima.



Slika 5-2 Arhitektura TCP/IP modela sa protokolima [21]

5.3 Ostvarivanje komunikacije s PLC-om

Za ostvarivanje komunikacije s PLC-om, temeljene na Modbus TCP/IP protokolu koristimo gotov paket pyModbusTCP [23]. Ovaj paket daje pristup modbus/TCP serveru pomoću ModbusClient objekta. Paket se sastoji od 4 klase. Client, sever, utilis i constants klase. Klasa server služi za sigurno pristupanje bitovima i riječima. Naredbe poput „start()“, „stop()“ i „is_run()“ koriste se za pokretanje, zaustavljanje ili provjeru stanja servera. Unutar klase client opisuju se funkcije za povezivanje sa serverom te objašnjenje naredbi za mijenjanje stanja memorija PLC-a. Naredbe poput read_single_coil ili write_single_coil koje se koriste za čitanje ili pisanje stanja pojedinih zavojnica detaljno su objašnjenje unutar klase client. Klasa constants sadrži popis svih modbus funkcija za čitanje ili pisanja u registar. Također sadrži i posebne kodove i kodove pogrešaka. Unutar klase utilities prikazane su i objašnjene modbus funkcije za upravljanje podacima. Pomoću tih funkcija moguće je prebacivati vrijednosti iz jednog formata u drugi. Npr. val_int integer u bits list [23]. Na slici 5-3 prikazana je mapa pyModbusTCP paketa.



Slika 5-3 Mapa pyModbusTCP paketa [23]

5.4 Primjeri komunikacije

5.4.1 Čitanje izlaza

```
##################
#Citanje izlaza
##################

from pyModbusTCP.client import ModbusClient
import time
import struct
SERVER_HOST = "192.168.0.110"
SERVER_PORT = 502


c = ModbusClient()

# uncomment this line to see debug message
#c.debug(True)

# define modbus server host, port
c.host(SERVER_HOST)
c.port(SERVER_PORT)
#addr odreduje adresu zavojnice te program prikazuje njeno stanje,toggle odreduje
#koliko adresa u nizu zelimo procitati
#Adrese zavojnica
#1-spuštanje rolete prostorije 1
#2-podizanje rolete prostorije 1
#3-rasvjeta prostorije 1
#5-podizanje rolete prostorije 2
#6-spuštanje rolete prostorije 2
#7=rasvjeta prostorije 2
addr=2
toggle=1
c.open()
print('Stanje izlaza=')
bits=c.read_coils(addr,toggle)
print('Stanje izlaza='+str(bits))

```

Slika 5-4 Čitanje izlaza

5.4.2 Čitanje temperature

```
##################
#Citanje temperature prostorije 1
##################

from pyModbusTCP.client import ModbusClient
import time
import struct
SERVER_HOST = "192.168.0.110"
SERVER_PORT = 502


c = ModbusClient()

# uncomment this line to see debug message
#c.debug(True)

# define modbus server host, port
c.host(SERVER_HOST)
c.port(SERVER_PORT)

c.open()

t1=c.read_holding_registers(402,2)
print(t1)
packed_string=struct.pack("HH",*t1)
unpacked_float=struct.unpack("f",packed_string)[0]
print('Temperatura prostorije 1:')
print(unpacked_float)

t2=c.read_holding_registers(404,2)
packed_string=struct.pack("HH",*t2)
unpacked_float=struct.unpack("f",packed_string)[0]
print('Referentna temperatura prostorije 2:')
print(unpacked_float)

```

Slika 5-5 Čitanje temperature prostorije

5.4.3 Promjena izlaza

```
##################
#Promjena izlaza
##################

from pyModbusTCP.client import ModbusClient
import time
import struct
SERVER_HOST = "192.168.0.110"
SERVER_PORT = 502


c = ModbusClient()

# uncomment this line to see debug message
#c.debug(True)

# define modbus server host, port
c.host(SERVER_HOST)
c.port(SERVER_PORT)
#addr odreduje adresu zavojnice te program mijenja njeno stanje,toggle odreduje
#u koje stanje zelimo promijeniti"
#Adrese zavojnica"
#8247-spuštanje rolete prostorije 1"
#8248-podizanje rolete prostorije 1"
#8249-rasvjeta prostorije 1"
#8250-podizanje rolete prostorije 2"
#6-spuštanje rolete prostorije 2"
#0=rasvjeta prostorije 2"
addr=8247
toggle=1
c.open()
c.write_single_coil(addr,toggle)
bits=c.read_coils(addr,1)
print('Novo stanje izlaza='+str(bits))
```

Slika 5-6 Promjena izlaza

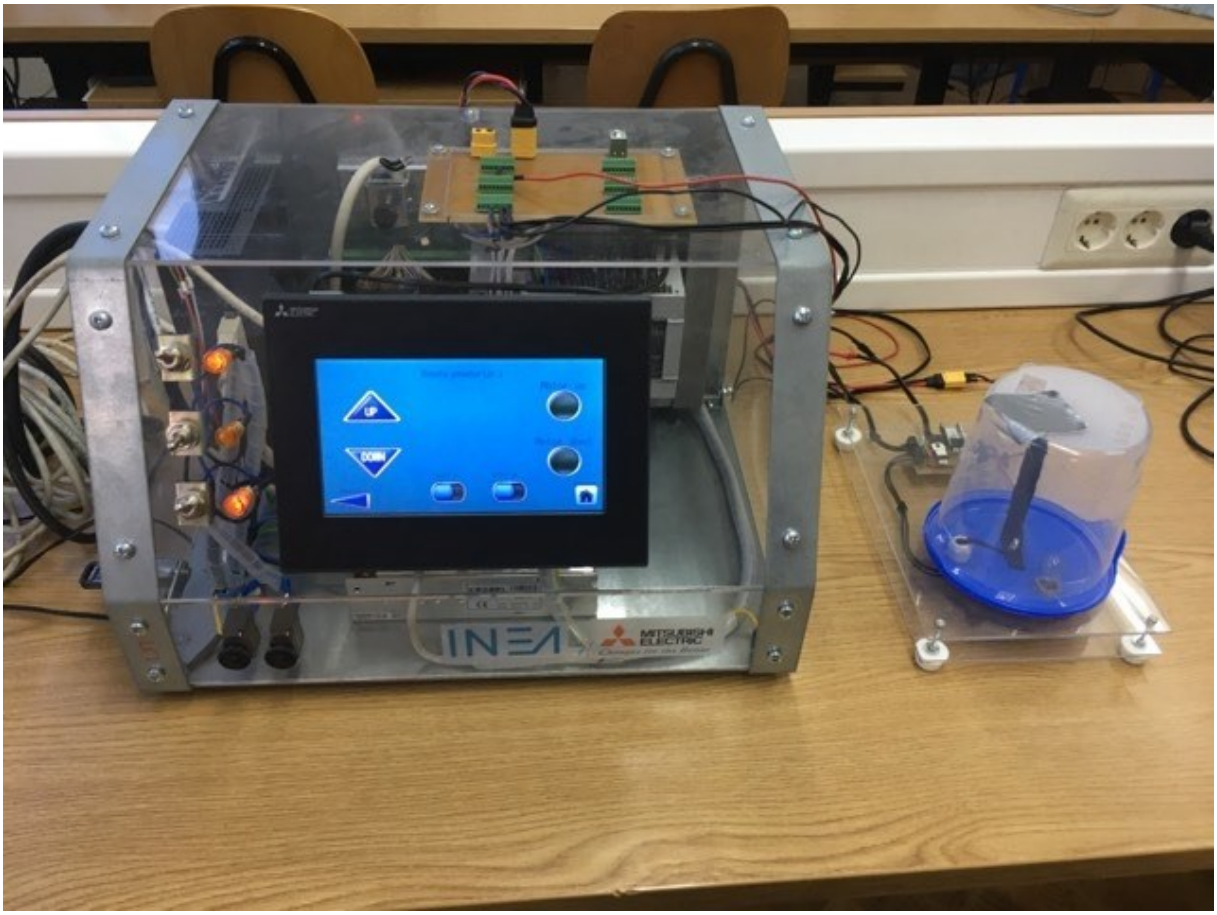
5.4.4 Upisivanje reference temperature

```
##################
#Upisvanje referentne temperature prostorije 1
##################

from pyModbusTCP.client import ModbusClient
import time
import struct
SERVER_HOST = "192.168.0.110"
SERVER_PORT = 502


c = ModbusClient()

# uncomment this line to see debug message
#c.debug(True)

# define modbus server host, port
c.host(SERVER_HOST)
c.port(SERVER_PORT)
c.open()
#referentna temperatura prostorije 1
ref_temp1=35
packed_float=struct.pack("f",ref_temp1)
unpacked_string=struct.unpack("HH",packed_float)
print(unpacked_string)

c.write_single_register(404,unpacked_string[0])
c.write_single_register(405,unpacked_string[1])

#referentna temperatura prostorije 2
ref_temp2=25
packed_float=struct.pack("f",ref_temp2)
unpacked_string=struct.unpack("HH",packed_float)
print(unpacked_string)

c.write_single_register(408,unpacked_string[0])
c.write_single_register(409,unpacked_string[1])
```

Slika 5-7 Upisivanje reference temperature

6 Laboratorijski postav

Za potrebe ovog rada koristi se PLC/HMI postav unutar laboratorija Fakulteta strojarstva i brodogradnje. Upravljački programi izrađeni su na računalu fakulteta i prebačeni na postav. Svi elementi ulaznih varijabli su virtualni, zajedno sa aktivatorima i izvršnim elementima sustava. Izuzetak je sustav regulacije temperature u kojem umjesto proporcionalnog ventila kao izvršnog elementa koristimo toplinsku komoru. Toplinska komora sadrži NTC otpornik kao senzor temperature i električnu žarulju kao izvor topline. Snaga koja protječe kroz žarulju simulira protok koji bi prolazio unutar proporcionalnog ventila. Na slici 6-1 prikazan je korišteni laboratorijski postav.



Slika 6-1 Laboratorijski postav

7 Zaključak

U ovome završnom radu obrađena je implementacija sustava pametne kuće temeljenog na programibilnom logičkom kontroleru (PLC), grafičkim upravljačkim ekranom (HMI) i mrežnim komunikacijskim protokolom Modbus. Objašnjene su osnovne komponente sustava, način komunikacije i upravljanja procesima unutar sustava. Razrađeni su potrebni električni sklopovi za prihvat i slanje signala, te je pokazan jedan on mogućih načina implementacije upravljanja rasvjetom, roletom prozora te regulacijom temperature. Također je izveden upravljački program i grafičko sučelje prema korisniku koji su implementirani na laboratorijskom postavu. Pokazana je komunikacija s PLC-om preko Modbus TCP/IP protokola koristeći biblioteku programskog jezika Python. U konačnici je pokazano da PLC može relativno jednostavno i pouzdano biti centralna komponenta pri upravljanju pametnom kućom, uz moguću vanjsku kontrolu putem mreže. U budućnosti se može razvijati upravljanje putem interneta, budući da se obrađena komunikacija putem Modbusa može proširiti primjerice implementacijom programskih skripti u alatu za generiranje web stranica putem Pythona.

8 Literatura

[1] Iva Bošnjak, Tehnologija pametne kuće, Filozofski fakultet u Rijeci, Rijeka, 2018

[2] Sanjin Rade, Pametna kuća, Filozofski fakultet u Rijeci, Rijeka, 2015

[3] https://www.security.org/home-automation/ , Pristupljeno: 31. siječnja 2021

[4] https://en.wikipedia.org/wiki/Internet_of_things , Pristupljeno: 15. veljače 2021

[5] https://zir.nsk.hr/islandora/object/politehnikapu:90/preview/ Pristupljeno: 16. veljače 2021

[6] https://medicaleducationonline.org/home-automation-system-the-total-control-is-under-you/ , Pristupljeno 16. veljače 2021

[7] https://en.wikipedia.org/wiki/Programmable_logic_controller, Pristupljeno: 14. siječnja 2021

[8] https://docs.rs-online.com/4ee4/0900766b813df617.pdf, Pristupljeno: 15. siječnja 2021

[9] https://eu3a.mitsubishielectric.com/fa/en/dl/10639/jy997d52901d.pdf, Pristupljeno: 15. siječnja 2021

[10] https://www.inea.hr/produkti/mitsubishi-electric/hmi-paneli/serija-got-simple/ Pristupljeno: 15. siječnja 2021

[11] https://www.mitsubishielectric.com/fa/products/cnt/plceng/smerit/gx_works3/programming.html Pristupljeno: 15.siječnja 2021

[12] https://us.mitsubishielectric.com/fa/en/products/visualization/human-machine-interfaces-hmis-got/engineering-software/gt-works3-version1/gt-works3-version1, Pristupljeno: 15. siječnja 2021

[13] T. Šurina, Automatska regulacija, Školska knjiga, Zagreb, 1981

[14] https://en.wikipedia.org/wiki/Thermistor/ , Pristupljeno: 29. siječnja 2021

[15] P. Šarac, Projektiranje i implementiranje digitalnih regulatora PID tipa za regulaciju temperature toplinske komore, FSB, Zagreb, 2020

[16] https://www.burkert.com/en/content/download/7480/229630/file/Product-Overview-Proportional-Valves.pdf?id=44 , Pristupljeno: 29. siječnja 2021

[17] http://homepage.cem.itesm.mx/carbajal/Microcontrollers/ASSIGNMENTS/readings/ARTICLES/barr01_pwm.pdf?fbclid=IwAR2_BJ7cOKpjVcqnEOoy6iBDdpmyL3u_i-l8m9vsSoDZEJ507WMEIgDdcf0, Pristupljeno: 29. siječnja 2021

[18] https://en.wikipedia.org/wiki/Pulse-width_modulation, Pristupljeno: 29. siječnja 2021

[19] https://en.wikipedia.org/wiki/Modbus/ , Pristupljeno: 3. veljače 2021

[20] https://en.wikipedia.org/wiki/Internet_protocol_suite/ , Pristupljeno: 3. veljače 2021

[21] Perić, Jelena, Analiza funkcija protokola transportnog sloja TCP/IP složaja, Fakultet prometnih znanosti, Zagreb, 2019

[22] https://www.computernetworkingnotes.com/ccna-study-guide/data-encapsulation-and-de-encapsulation-explained.html/ , Pristupljeno: 4. veljače 2021

[23] https://pypi.org/project/pyModbusTCP/ , Pristupljeno: 11. veljače 2021