

UNIVERZA V LJUBLJANI
Fakulteta za elektrotehniko

Klemen Žbontar

PAMETNO OGREVANJE PROSTOROV

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: doc. dr. Boštjan Murovec

Ljubljana, 2011

Zahvala

Na prvem mestu bi se rad zahvalil doc. dr. Boštjanu Murovcu za dostopnost za razpravo, nasvete, dodatna pridobljena znanja in moralno podporo. Zahvalil bi se sodelavcem v podjetju LPKF Laser & Elektronika d.o.o. za pomoč pri izdelavi tiskanega vezja. Prav tako bi se rad zahvalil družini in puncu za izkazano moralno podporo med časom študija ter nastajanja te diplomske naloge.

Povzetek

Diplomska naloga raziskuje možnost izboljšave klasične regulacije temperature prostora z uvedbo pametnega ogrevanja. Le-to je sestavljeno iz dveh ločenih sistemov, modula ter spletne strani, ki sta med seboj povezana preko internetnega omrežja. Prvi se uporablja za merjenje in regulacijo ambientne temperature, medtem ko drugi služi kot uporabniški vmesnik. Srce modula temelji na kombinaciji dveh Microchipovih proizvodov, in sicer mikrokrmilnika PIC18F4620 ter Ethernet krmilnika ENC28J60. Uporabljeni internetni protokoli so bili realizirani z uporabo odprtokodnega Microchipovega TCP/IP sklada.

Za uspešno delovanje aplikacije je poleg pravilno delujoče komunikacije potrebno natančno zaznavanje temperature prostora. Le-to je implementirano preko preciznega napetostnega temperaturnega senzorja LM35 v povezavi z uporabo instrumentacijskega ojačevalnika ter zunanega 12-bitnega AD pretvornika. Verifikacija rezultatov je bila opravljena s pomočjo referenčnega termometra proizvajalca Burster. Za krmiljenje grela se uporablja dvostopenjski regulacijski algoritem.

Uporabniški vmesnik v obliki spletne strani omogoča nastavljanje parametrov regulacijskega algoritma, ogled aktualnih podatkov o izmerjeni in želeni temperaturi ter še nekatere dodatne možnosti. Aplikacija je zasnovana tako, da modul strežniku periodično pošilja aktualne podatke o njegovem delovanju in hkrati proizveduje po novih regulacijskih parametrih. Poleg tega je naprednejšemu uporabniku omogočeno uporabljanje modula v razvojne namene.

Ključne besede:

Nadgradnja regulacije ogrevanja, komunikacija mikrokrmilnik-spletni strežnik, Microchip TCP/IP sklad, natančno merjenje temperature, PIC18F4620

Abstract

The thesis discusses the possibility for improving classical room temperature regulation with the introduction of smart heating. It consists of two separate systems, a module and a Web page, which are interconnected through an Internet network. The first is used for measurement and regulation of the ambient temperature and the second as a user interface. As a basis the module incorporates a combination of two Microchip products, the PIC18F4620 microcontroller and the ENC28J60 Ethernet controller. The used Internet Protocols were realised with the open-source TCP/IP Stack from Microchip.

Beside a correctly functioning communication it is necessary to accurately measure the room temperature for successful operation of the application. This is implemented by means of precise voltage temperature sensor, instrumentation amplifier and a 12-bit AD converter device. Verification of the results was conducted with a Burster reference thermometer.

The Web page as a user interface allows setting the regulation algorithm parameters, the examination of current measured and desired room temperature data and some other options. The application is designed in a way that the module periodically sends current data about its operation and concurrently enquires about new regulation parameters. Furthermore it is possible that this module is used as a development kit for advanced users.

Key words:

Heating regulation upgrade, communication microcontroller-Web server, Microchip TCP/IP Stack, accurate temperature measurement, PIC18F4620

Kazalo vsebine

1	Uvod.....	2
2	Računalniške mreže.....	4
2.1	Referenčni model ISO OSI.....	6
2.1.1	Sloji ISO OSI referenčnega modela.....	6
2.2	Model TCP/IP.....	8
2.2.1	Načelo ovojnice.....	9
2.3	Brezrazredno naslavljanje.....	10
2.3.1	Bloki brezrazrednih naslovov.....	11
2.4	IP protokol.....	11
2.4.1	Usmerjanje paketov.....	12
2.4.2	Podomrežno naslavljanje.....	12
2.5	ICMP protokol.....	13
2.6	ARP protokol.....	14
2.7	UDP protokol.....	15
2.7.1	Vrata in vtičnice.....	16
2.8	TCP protokol.....	17
2.8.1	Vzpostavljane in prekinjanje povezave.....	18
2.8.2	Nadzor pretoka.....	19
2.9	DHCP protokol.....	19
3	Delovanje in uporaba Microchip TCP/IP sklada.....	22
3.1	Delovanje sklada.....	23
3.1.1	Datoteka <i>main</i>	24
3.1.2	Kooperativna večopravnost.....	25
4	Zgradba in delovanje modula.....	28
4.1	Napajalni del.....	29
4.2	Ethernet krmilnik ENC28J60.....	31
4.3	Vhodno-izhodni del z mikrokrmilnikom.....	33
4.3.1	SPI protokol.....	34
4.4	Vmesnik za programiranje modula.....	35
4.5	Serijski EEPROM pomnilnik.....	36
5	Merjenje in regulacija temperature.....	38

5.1	Merjenje temperature	38
5.1.1	Zgradba merilnega vezja	39
5.1.2	Dimenzioniranje elementov vezja za zajem temperature.....	41
5.1.3	Izračun pogreška meritve temperature	43
5.2	Regulacija temperature	45
5.2.1	Splošno o sistemih vodenja	46
5.2.2	Delovanje dvopoložajnega regulatorja.....	47
5.2.3	Dvostopenjska regulacija temperature.....	48
5.2.4	Programska realizacija dvostopenjskega regulatorja.....	52
5.2.5	Komunikacija s strežnikom	54
5.2.6	Delovanje modula v primeru napake	56
6	Praktična izvedba aplikacije	58
7	Verifikacija rezultatov	62
8	Zaključek.....	64
	Literatura	66
	Izjava.....	70
A	Priloga – električna vezalna shema.....	72
B	Priloga – spisek uporabljenih elementov	82

Kazalo slik

Slika 1 – Shematski prikaz zgradbe in delovanja aplikacije.....	3
Slika 2 – Topologija omrežja	4
Slika 3 – tip omrežja CSMA/CD.....	5
Slika 4 – referenčni model ISO OSI.....	7
Slika 5 – model ISO OSI v realnosti	9
Slika 6 – Načelo ovojnice	10
Slika 7 – Brezrazredno naslavljanje	11
Slika 8 – ARP proces pridobivanja strojnega naslova	14
Slika 9 – Pošiljanje podatkov pri uporabi UDP protokola.....	15
Slika 10 – Vrata, ki jih uporabljata TCP in UDP protokola	16
Slika 11 – Pošiljanje podatkov pri uporabi TCP protokola	17
Slika 12 – Trosmerno rokovanje za vzpostavitev seje.....	18
Slika 13 – Drseče okno	19
Slika 14 – DHCP samodejno dodeljevanje IP naslova	20
Slika 15 – Modularna zgradba TCP/IP sklada	22
Slika 16 – Shematski prikaz zgradbe modula.....	28
Slika 17 – Graetzov mostič.....	29
Slika 18 – Delovanje diode	30
Slika 19 – Delovanje Graetzovega mostiča	30
Slika 20 – Tipična komunikacijska vezava ENC28J60	31
Slika 21 – Zunanje povezave Ethernet krmilnika ENC28J60	32
Slika 22 – Ojačevanje signalov med ENC28J60 in PIC18F4620.....	32
Slika 23 – SPI komunikacija med gospodarjem in večimi sužnji.....	34
Slika 24 – Primer SPI komunikacije gledano s strani sužnja	35
Slika 25 – Priključitev programatorja ICD preko konektorja RJ11	35
Slika 26 – Splošna zgradba merilnega sistema.....	38
Slika 27 – Shematska zgradba vezja	39
Slika 28 – Uporovni delilnik	43
Slika 29 – Bločni diagram zaprtozančnega regulacijskega sistema	46
Slika 30 – Bločni diagram zaprtozančnega sistema ogrevanja.....	47
Slika 31 – Nastanek dvopoložanega regulatorja iz zveznega P regulatorja	48
Slika 32 – Dvig temperature zaradi delovanja grela	49

Slika 33 – Potek veličin sistema brez upoštevanja mrtvega časa.....	50
Slika 34 – Potek veličin sistema pri upoštevanju histereze in mrtvega časa	51
Slika 35 – Bločni diagram regulacijske zanke.....	52
Slika 36 – Potek funkcije <i>TempRegulation()</i>	53
Slika 37 – Bločni diagram funkcije <i>TCPClient()</i>	54
Slika 38 – Zgrajen modul.....	58
Slika 39 – Prva stran spletne strani	59
Slika 40 – Nastavitve prikaza podatkov na naslovni strani	60

Kazalo tabel

Tabela 1 – Primer določanja omrežnega naslova	13
Tabela 2 – Datoteke sklada, potrebne za pravilno delovanje aplikacije	23
Tabela 3 – Glavne karakteristike mikrokrmilnika PIC18F4620	33
Tabela 4 – Rezultati primerjalnih meritev	62
Tabela 5 – Spisek uporabljeni elementov.....	82

Kazalo grafov

Graf 1 – Primerjava referenčne in izmerjene temperature.....	63
Graf 2 – Odstopanje izmerjene temperature od referenčne	63

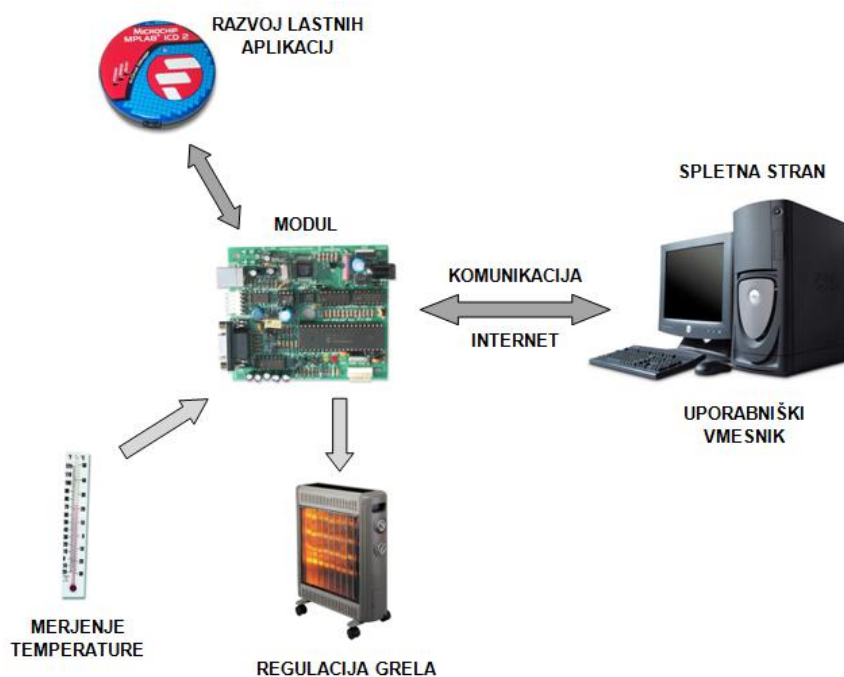
1 Uvod

Mnoga gospodinjstva uporabljajo za namene centralnega ogrevanja klasične termostate s statično nastavitvijo želene temperature prostora in časa ogrevanja, zaradi česar se lahko zgodi, da bomo ogrevali prazno hišo ali stanovanje in si s tem brezpomensko višali stroške ogrevanja.

V šoli ali podjetju z večjimi sejnimi prostori, ki niso neprekinjeno ali celo dokaj redko zasedeni si lahko zamislimo naslednji primer. Pri težnji po čim nižjih stroških ogrevanja ob uporabi klasičnih termostatov lahko sklepamo, da ne moremo aktivno prilagajati urnika ogrevanja vsem nepredvidljivim situacijam in s tem nismo zmožni znižati stroškov ogrevanja na minimum.

Situacijo je možno izboljšati s sistemom, ki profil ogrevanja prilagaja podatkom prostora, pridobljenim iz različnih senzorjev. Taki sistemi so že razviti in se uporabljajo v tako imenovanih pametnih hišah, njihova največja omejitev za splošne kupce pa je visoka cena, ki v veliki večini ne odtehta možnosti, ki jih taki sistemi ponujajo.

To diplomsko delo se navezuje predvsem na slednji primer pametnega ogrevanja ene od učilnic na srednji šoli. Glavno vodilo zamisli diplomske naloge je bilo dejstvo, da je na večini šol obstoječi sistem ogrevanja možno opazno izboljšati z uporabo relativno dostopnih in poceni električnih elementov. Cilj diplomske naloge je bil zasnovati, izdelati in sprogramirati modul, ki je sposoben zajemati temperaturo v prostoru in se na podlagi urnika razreda, pridobljenega s spletnega strežnika, odločati o ogrevanju prostora. Modul je bil zasnovan tako, da ga uporabnik, seznanjen s programiranjem mikrokontrolerov, poleg primarne naloge pametnega ogrevanja prostora, uporablja kot razvojno okolje za druge aplikacije, ki uporabljajo dostop do Interneta. Poleg modula je bila izdelana spletna stran, ki omogoča spremljanje in nastavljanje delovanja modula. Slika 1 prikazuje shematski prikaz zgradbe in delovanja aplikacije.



Slika 1 – Shematski prikaz zgradbe in delovanja aplikacije

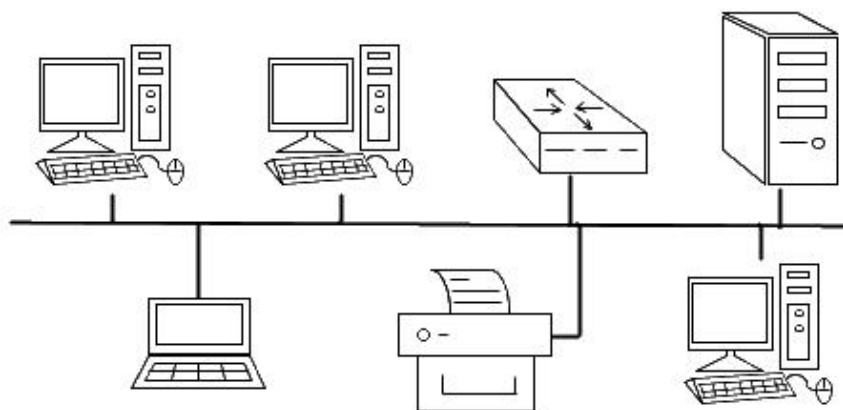
Celotna interakcija med uporabnikom in modulom poteka preko internetnega omrežja, zato si bomo najprej ogledali teoretično ozadje povezovanja naprav v računalniška omrežja. V nadaljevanju bomo to tematiko predstavili še iz praktičnega zornega kota, pri čemer se bomo osredotočili na implementacijo uporabljene programske opreme.

Sledila bo predstavitev delovanja poglavitnih delov modula. Del vezja za zajem temperature in programsko realizacijo regulacije si bomo ogledali podrobneje, saj je njuno delovanje kritično za pravilno delovanje aplikacije.

V zadnjem delu se bomo osredotočili na delovanje aplikacije kot celote, kjer bomo med drugim predstavili delovanje spletne strani ter na verifikacijo rezultatov.

2 Računalniške mreže

Računalniško omrežje [1, poglavja 2.1-2.4] ali krajše omrežje je definirano kot zbirka računalnikov in drugih naprav, ki so med seboj povezana s komunikacijskim kanalom za izmenjavo podatkov in souporabo perifernih enot, kot so tiskalniki, trdi diski in ostale naprave. Iz Slike 2 [2], ki grafično ponazarja omrežje, lahko razberemo, da brez organiziranosti pošiljanja podatkov obvezno prihaja do trčenj.

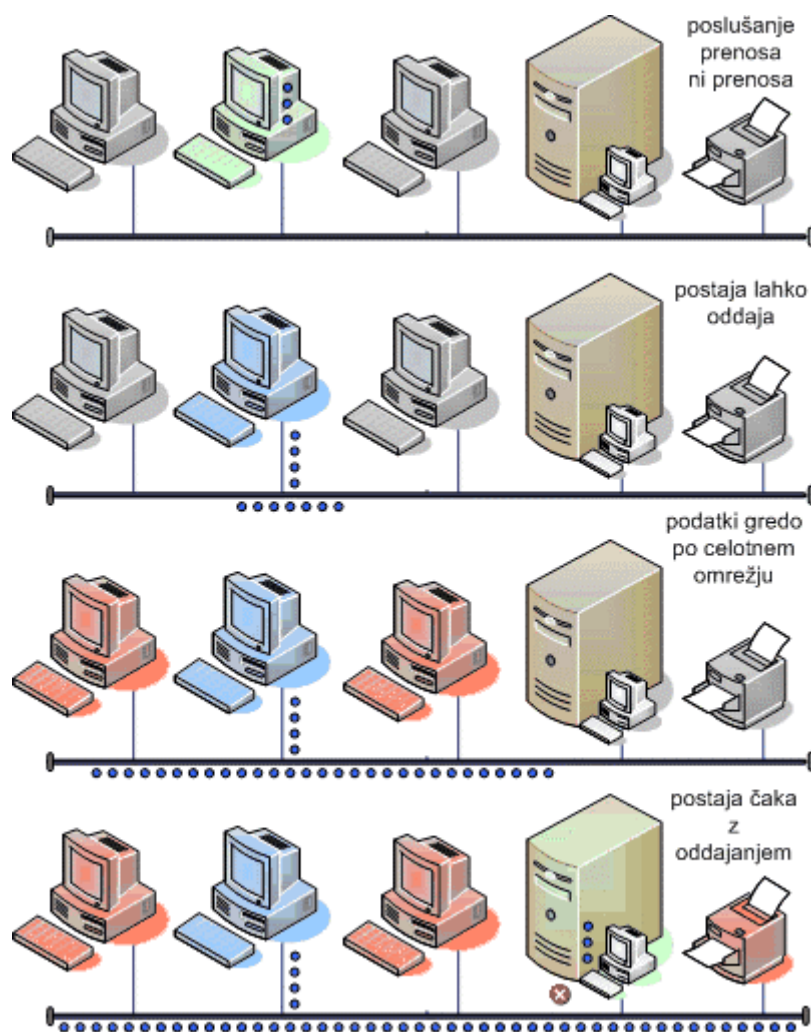


Slika 2 – Topologija omrežja

Število trčenj lahko občutno zmanjšamo z definiranjem pravil:

- detekcija prenosa (ang. Carrier Sense): pred začetkom oddajanja naprava najprej posluša, če že kdo oddaja in začne oddajati šele, ko je komunikacijski kanal prost,
- zaznavanje trka (ang. Collision Detect): če naprava zazna, da je prišlo do trka, počaka naključen čas in ponovno poskusi poslati podatke,
- maksimalna dovoljena dolžina kabla ali najkrajši dovoljeni čas oddajanja signala: čas oddajanja signala ne sme biti krajši od časa, potrebnega za razširitev podatkov po celem omrežju.

Vsa zgoraj omenjena pravila so združena v tipu omrežja *detekcija prenosa-dostop za vse-zaznavanje trka* (ang. **C**arrier **S**ense-**M**ultiple **A**ccess/**C**ollision **D**etect ali krajše **CSMA/CS**) [1, poglavje 2.4.8]. Princip delovanja bomo najlažje razumeli s pomočjo Slike 3 [3].



Slika 3 – tip omrežja CSMA/CD

Pri tem tipu omrežja vse naprave nadzorujejo omrežni promet na vodilu. Naprava, ki želi oddajati, najprej preveri prostost vodila. Ko je vodilo prosto, začne oddajati podatke. Druga naprava, ki želi oddajati, lahko zaradi končne hitrosti potovanja podatkov po vodilu zazna, da je le-to prosto in začne pošiljati podatke. To dejanje privede do trka, ki ga napravi zaznata in zato prekineta oddajanje. Le-to se ponovi po pretečenem naključnem času, kar zmanjša možnost ponovnega trka. Ker imajo vse naprave enakovreden dostop do vodila, začnejo pošiljati podatke tiste, ki zaznajo prosto vodilo.

Ta tip zasnove je utelešen v računalniškem omrežju, imenovanem *Ethernet* in se najbolj pogosto uporablja za komercialne namene. Z večanjem omrežja postaja vse večji tudi promet na vodilu, kar pripelje do večjega števila trkov in posledično večjega števila ponovnih pošiljanj podatkov. To se odraža v počasni odzivnosti omrežja. V

industrijskem okolju se večinoma uporabljata dva tipa omrežij. *Industrijski Ethernet* imenujemo Ethernet omrežje, ki je prilagojeno delovanju v industrijskem okolju. Njegova prednost je v razširjenosti naprav z Ethernet vmesnikom, kar znatno zniža stroške postavitve. *Žetonsko vodilo* (ang. Token bus) je zasnovano tako, da lahko oddaja samo naprava, ki je prejela »žeton«, kar prepreči trke. Pomanjkljivost te zasnove je komplicirana izvedba ponovne vzpostavitve reda po okvari in problem vključitve dodatne naprave v omrežje.

V primeru identičnih zmogljivosti vodila velja, da prenesemo več podatkov:

- preko Ethernet vodila, če je omrežje manj obremenjeno, saj je pri žetonskem vodilu potrebno čakati na »žeton«, čeprav je vodilo prazno in
- preko žetonskega vodila, če je omrežje bolj obremenjeno, saj pri Ethernet vodilu prihaja do več trkov.

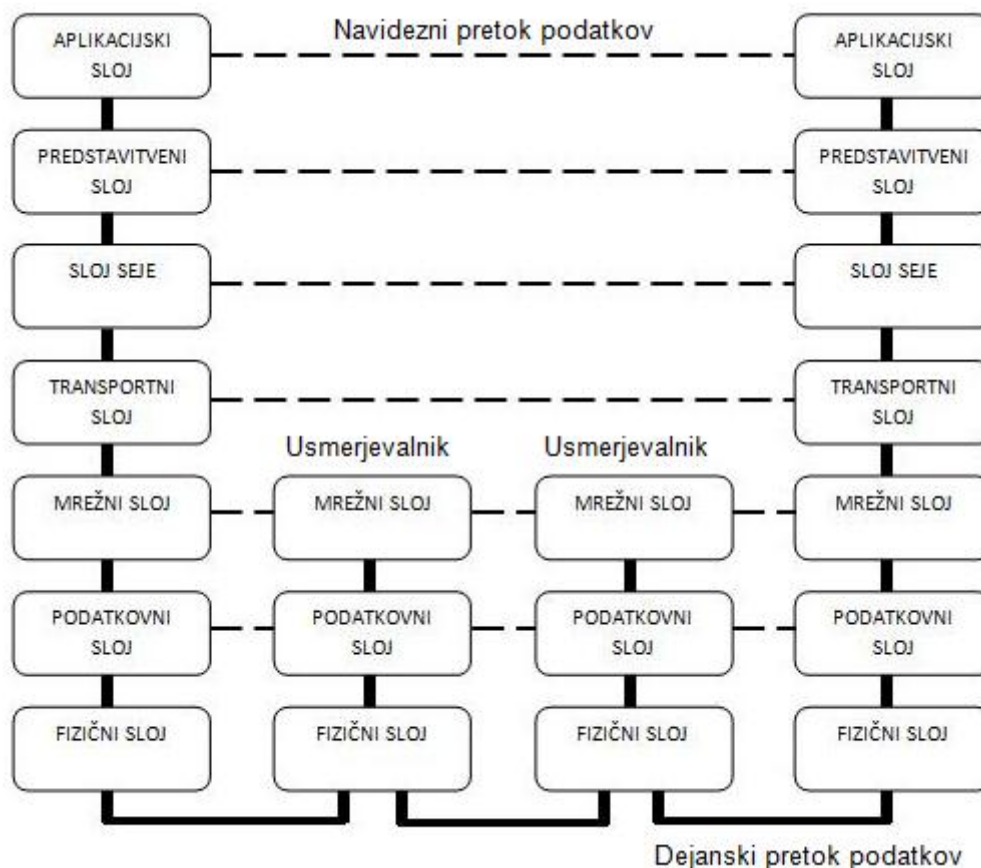
2.1 Referenčni model ISO OSI

V 80. letih je mednarodna organizacija za standardizacijo ISO (ang. International Organisation for Standardization) izdala dokument, imenovan Osnovni referenčni model za odprte systemske povezave (ang. The Basic Reference Model for Open Systems Interconnection) [1, poglavje 11.4.1], ki uvaja deljenje komunikacijskega sistema na sedem podsistemov, ki jih imenujemo sloji. V osnovi je bila ta struktura mišljena kot model za nov protokolni sklad, ki ni bil nikoli uveljavljen. Trenutno se ga uporablja v povezavi z ostalimi obstoječimi protokoli kot referenčno orodje ter za učenje omrežnih komunikacij.

2.1.1 Sloji ISO OSI referenčnega modela

Vsak sloj zagotavlja storitve sloju nad njim in hkrati uporablja storitve nižje ležečega sloja, istočasno pa neposredno komunicira z enako ležečim slojem na drugi napravi. To omogoča logično ali virtualno povezavo med enako ležečimi sloji, kot je prikazano na Sliki 4. V praksi poteka prava komunikacija med sosednjima slojema le na enem

računalniku, kjer na vsakem sloju programska oprema zagotavlja omrežne funkcije glede na skupek uporabljenih protokolov.



Slika 4 – referenčni model ISO OSI

- **FIZIČNI SLOJ** (ang. Physical Layer): Fizični sloj [1, poglavje 11.5] definira električne in mehanske lastnosti mrežnih naprav ali drugače, povezavo med mrežno napravo in prenosnim medijem (bakreni ali optični kabel) v kar so vključene tudi lastnosti kablov in konektorjev. Opravlja storitev fizičnega prenosa podatkov, ki je podvržen šumom. Najmanjša enota informacije so biti.
- **PODATKOVNI SLOJ** (ang. Data Layer): Podatkovni sloj [1, poglavje 11.5] odpravlja napake pri prenosu podatkov preko nezanesljivega komunikacijskega medija in s tem zagotavlja zanesljivo povezavo višjim slojem. Podatke razdrobi na manjše okvirje (ang. Frames) in jim doda mrežni naslov. Najmanjša enota informacije je okvir.

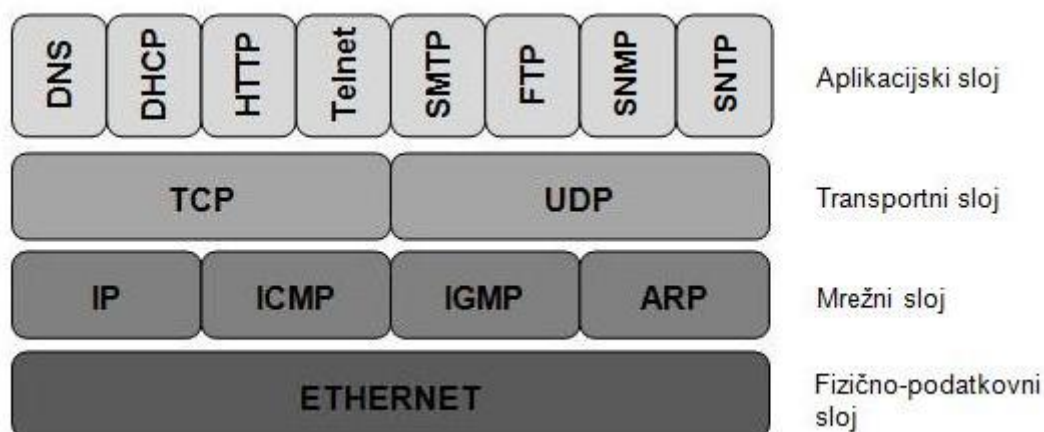
- **MREŽNI SLOJ** (ang. Network Layer): Mrežni sloj [1, poglavje 11.5] zagotavlja načine prenosa podatkovnih paketov iz enega v drugo omrežje, kar omogoča delovanje Interneta. Na tem nivoju delujejo usmerjevalniki, ki določajo poti potovanja podatkovnih paketov. Najmanjša enota informacije je paket ali datagram.
- **TRANSPORTNI SLOJ** (ang. Transport Layer): Transportni sloj [1, poglavje 11.5] skrbi za prenos podatkovnih paketov med končnimi uporabniki in višjim slojem zagotavlja zanesljiv prenos paketov. Najmanjša enota informacije je segment (ang. Segment).
- **SLOJ SEJE** (ang. Session Layer): Sloj seje [1, poglavje 11.5] nadzoruje vzpostavitev, vzdrževanje in prekinitev seje, to je komunikacije med končnimi napravami. Vzpostavlja različne tipe povezav. *Enosmerna* (ang. Simplex) uporablja eno oddajno in eno ali več sprejemnih naprav. *Izmenično dvosmerna* (ang. Half Duplex) omogoča napravi izmenično sprejemanje in oddajanje. Pri *dvosmerni* (ang. Full Duplex) lahko naprava hkrati oddaja in sprejema podatke. Večina omrežij tega sloja ne uporablja oziroma njegovo funkcijo realizira na druge načine.
- **PREDSTAVITVENI SLOJ** (ang. Presentation Layer): Predstavitveni sloj [1, poglavje 11.5] skrbi za uskladitev različnih načinov predstavitve podatkov. Sprejete podatke pretvori v obliko, primerno aplikaciji. Večina omrežij tega sloja ne uporablja več.
- **SLOJ APLIKACIJ** (ang. Application Layer): Aplikacijski sloj [1, poglavje 11.5] je vmesnik med končnim uporabnikom in komunikacijskim vmesnikom. V sloju aplikacij so storitve, ki jih kličejo aplikacije, kot so Mozilla Firefox in druge.

2.2 Model TCP/IP

Model TCP/IP ali Internetni sklad protokolov (ang. Internet Protocol Suite) opisuje množico splošnih napotkov implementacije mrežnih protokolov, ki omogočajo

napravam komuniciranje preko omrežij. Model TCP/IP ali krajše TCP/IP je dobil ime po dveh najbolj pogosto uporabljenih mrežnih protokolih, in sicer TCP protokola (ang. Transmission Control Protocol) ali protokola za nadzor prenosa in IP protokola (ang. Internet Protocol) ali internetnega protokola. TCP/IP zagotavlja povezljivost med končnimi uporabniki in predpisuje pravila za naslavljanje, usmerjanje, pošiljanje in sprejemanje točno določene oblike podatkov.

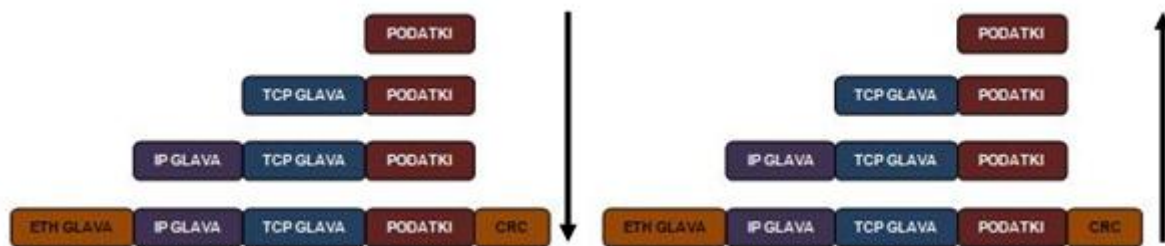
TCP/IP namesto sedem slojnega referenčnega modela ISO OSI uporablja štiri slojni model [1, poglavje 11.5.1, 4, poglavje 1.2], ki ga prikazuje Slika 5 [5, str. 3]. Zgornji trije sloji referenčnega modela ISO OSI so združeni v en sloj modela TCP/IP, ki ga imenujemo aplikacijski sloj. Naslednja dva sloja, transportni in mrežni sloj, sta enaka kot v referenčnem modelu ISO OSI, zadnja dva pa sta v modelu TCP/IP zopet združena v fizično-podatkovni sloj.



Slika 5 – model ISO OSI v realnosti

2.2.1 Načelo ovojnice

Načelo ovojnice [4, poglavje 1.6, 6] je metoda načrtovanja modularnih komunikacijskih protokolov, kjer vsak sloj podatkom, prejetim iz višjega sloja, doda svojo glavo (ang. Header) in v primeru Ethernet sloja tudi rep (ang. Trailer). Princip delovanja prikazuje Slika 6.



Slika 6 – Načelo ovojnice

Ko aplikacija zahteva prenos podatkov po omrežju, le-ti najprej potujejo skozi sloje. Vsak sloj prejete podatke razdrobi na enote informacij, ki so značilne za določeni sloj in jim doda informacije, potrebne za prenos paketa po omrežju. *Bistvo načela ovojnice je, da vsak sloj doda svojo glavo, pri tem pa ne pregleduje podatkov, ki jih je prejel.*

Na sprejemni strani potujejo podatki od najnižjega sloja proti aplikaciji. Vsak sloj prebere k paketu dodane informacije, jih odstrani in preostale podatke preda naslednjemu sloju.

2.3 Brezrazredno naslavljanje

Vsaka naprava, ki komunicira v omrežju, ima predpisan unikatni internetni naslov ali IP naslov (ang. IP Address) [4, poglavje 1.4, 7]. Le-ta je numerična oznaka, ki je dodeljena vsaki napravi v omrežju, katera za komunikacijo uporablja IP protokol. IP naslov je sestavljen iz dveh delov, in sicer *omrežnega naslova* (ang. Net ID) in *naslova gostitelja* (ang. Host ID). V uporabi so trije tipi IP naslovov:

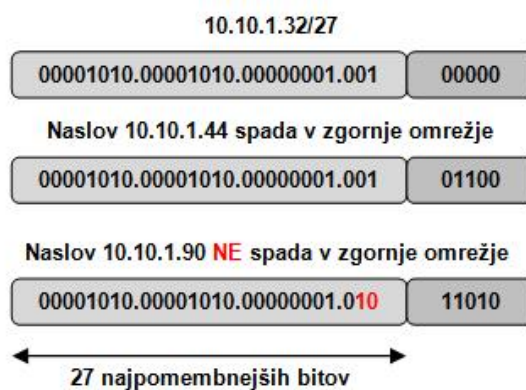
- naslovi za enega gostitelja (ang. unicast),
- naslovi za vse gostitelje na lokalnem omrežju (ang. broadcast) in
- naslovi za gostitelje, ki pripadajo skupini za oddajanje večim uporabnikom (ang. multicast).

Leta 1993 je bilo predstavljeno brezrazredno naslavljanje [1, poglavja 10.17-10.21], ki je nadomestilo naslovno arhitekturo, katera je temeljila na naslovnih razredih [1, poglavje 4]. Brezrazredno naslavljanje je z gospodarnejšim izkoriščanjem 32-bitnega naslovnega prostora IPv4 upočasnilo hitro izčrpavanje prostih IP naslovov. Temelji

na spremenljivi dolžini podomrežne maske (ang. Subnet Mask), ki omogoča delitev omrežij na poljubno velika podomrežja.

2.3.1 Bloki brezrazrednih naslovov

Brezrazredno naslavljanje temelji na bitni interpretaciji IP naslovov. Vsi naslovi, ki imajo enako začetno sekvenco bitov, so združeni v blok brezrazrednih naslovov (ang. CIDR block), ki določa omrežje. Posamezni naslovi imajo obliko štiridelnega decimalnega zapisa, ki mu sledi poševnica in številka od 0 do 32. Primer zapisa naslova in interpretacija uporabe sta prikazana na Sliki 7.



Slika 7 – Brezrazredno naslavljanje

Številka za poševnico ali predpona določa dolžino najpomembnejših bitov v naslovih, ki se ne spreminjajo, in posledično določa velikost omrežja. Manjša predpona pomeni večje omrežje in obratno. Če naslovi, ki se uporabljajo na določenem geografskem področju, spadajo v enak naslovni prostor, to znatno olajša usmerjanje paketov, kar bomo obdelali v naslednjem podpoglavju.

2.4 IP protokol

IP protokol [1, poglavje 7, 4, poglavje 3] je najpomembnejši komunikacijski protokol za usmerjanje IP datagramov med omrežji. Ker je zadolžen za usmerjanje in dostavljanje paketov preko mej posameznega omrežja samo na podlagi IP naslovov, je to primarni protokol, ki sestavlja Internet. Vsi podatki, ki izvirajo iz TCP, UDP,

ICMP in IGMP protokolov, so poslani v obliki IP datagramov. IP protokol je zasnovan kot nezanesljiv sistem dostavljanja datagramov, ki ne specificira poti potovanja posameznih datagramov. Nezanesljivost izvira iz dejstva, da nimamo nobenega zagotovila, da bo paket uspešno prišel do naslovnika. Protokol obravnava vsak paket neodvisno kot zaključeno celoto, kar pomeni, da usmerjevalniki ne hranijo informacij o poslanih paketih. S tem občutno zmanjšamo kompleksnost usmerjevalnikov, saj zagotavljanje zanesljive povezave prenesemo na višje sloje (TCP).

2.4.1 Usmerjanje paketov

Pri usmerjanju [4, poglavje 3.3] lahko govorimo o dveh vrstah paketov:

- paketi, ki so generirani lokalno in jih je potrebno poslati ter
- paketi, ki jih prejmemo od mrežnega vmesnika in jih je potrebno posredovati naprej (ang. forwarding).

IP sloj si pri usmerjanju paketov pomaga s tabelo usmerjanja (ang. Routing Table), ki je shranjena v pomnilniku naprave. Vsak vnos v tabeli je sestavljen iz naslovnega IP naslova, naslova usmerjevalnika na naslednjem koraku (ang. next-hop Router Address), dodatne informacije o naslovnem IP naslovu in podatkov o mrežnem vmesniku, preko katerega naj bi bili paketi poslani.

Usmerjanje deluje po načelu korak-za-korakom (ang. hop-by-hop), kar pomeni, da usmerjevalnik ne pozna celotne poti nekega paketa, razen v primeru, ko je naslovnik v istem omrežju kot pošiljatelj. Naloga usmerjevalnika je zagotavljanje IP naslova naslednjega koraka na poti paketa, ki naj bi bil bližje naslovniku.

2.4.2 Podomrežno naslavljanje

Podomrežje (ang. Subnetwork) [8] je logična podstruktura IP omrežja. Vse naprave v podomrežju naslavljam s skupnim, enakim najpomembnejšim delom bitov v njihovem IP naslovu. Posledično se IP naslov razdeli na dva dela, in sicer na omrežni naslov in preostali del ali naslov gostitelja. Slednji definira točno določeno napravo v podomrežju. Dolžino skupnega omrežnega naslova definiramo s podomrežno masko ali predpono, ki sledi IP naslovu. Predpona ima na mestu bitov, ki določajo omrežni

naslov, zapisane enice. Omrežje 192.168.2.1 s podomrežno masko 255.255.255.0 zapišemo v obliki 192.168.2.1/24.

Pred uporabo brezrazrednega naslavljanja smo omrežni naslov pridobili direktno iz IP naslova, sedaj pa za to potrebujemo tako IP naslov, kot tudi predpono, kar prikazuje primer v Tabeli 1.

Tabela 1 – Primer določanja omrežnega naslova

	Binarni zapis	Decimalni zapis
IP naslov	11000000.10101000.00000010.00001111	192.168.2.15
Predpona (/24)	11111111.11111111.11111111.00000000	255.255.255.0
Omrežni naslov	11000000.10101000.00000010.00000000	192.168.2.0
Naslov gostitelja	00000000.00000000.00000000.00001111	0.0.0.15

Omenimo dve večji prednosti uporabe podomreženja:

- struktura organizacije notranjega omrežja ni vidna zunanjim napravam, kar poveča varnost in
- podomreženje občutno skrajša tabele usmerjanja, saj zunanji usmerjevalniki poznajo samo IP naslov naprave, ki povezuje podomrežja v Internet. To pomeni, da ti usmerjevalniki potrebujejo samo en vpis v tabelo usmerjanja za dostop do kateregakoli gostitelja v podomrežjih.

2.5 ICMP protokol

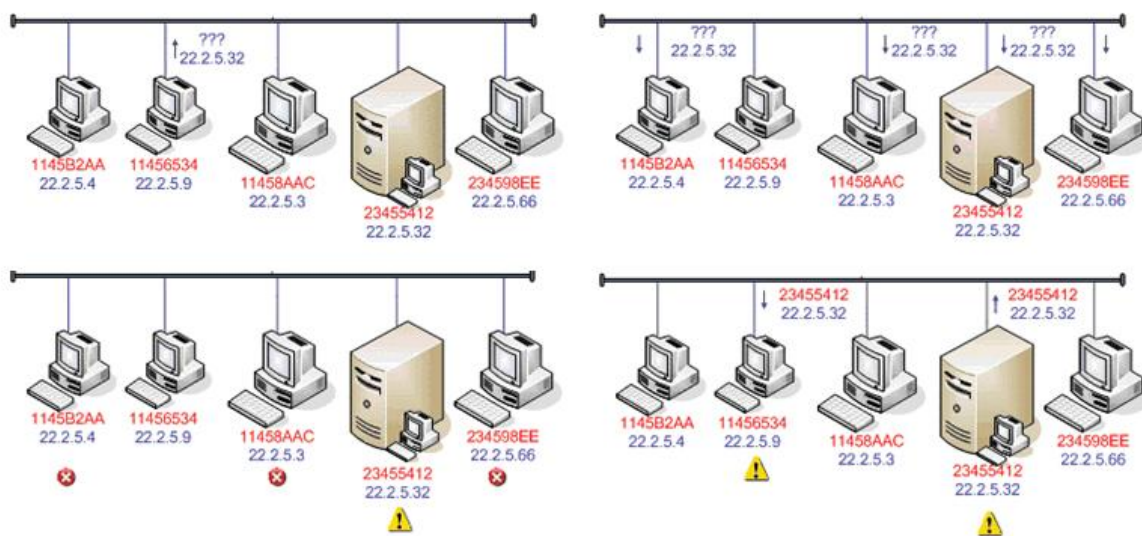
Internetni protokol za krmilna sporočila (ang. Internet Control Message Protocol) [1, poglavje 9] je eden bistvenih protokolov, uporabljenih v TCP/IP. Pogosto ga obravnavamo kot del mrežnega sloja. Njegova glavna naloga je pošiljanje sporočil o napakah, kot je nezmožnost povezave z gostiteljem ali usmerjevalnikom, poleg tega se uporablja tudi za pošiljanje poizvedovanj (ang. query). ICMP protokol se od transportnih protokolov (TCP in UDP) razlikuje v tem, da se ga tipično ne uporablja za izmenjavo podatkov med gostitelji, prav tako pa ga ne uporabljajo aplikacije, ki so v interakciji z uporabnikom (izjemi sta funkciji *ping* in *traceroute*, ki se uporabljata za

diagnostične naloge). ICMP sporočila se pošiljajo kot IP datagrami, kar pomeni, da je dostava le-teh nezanesljiva.

ICMP sporočilo vsebuje podatke, ki določajo transportni protokol in uporabniški proces, ki je povzročil napako.

2.6 ARP protokol

Protokol za pretvorbo naslovov (ang. Address Resolution Protocol) [1, poglavje 5, 4, poglavje 4] je mrežni protokol, ki se uporablja za določanje neznanih strojnih naslovov iz znanih IP naslovov. To je pomembno, ker fizično-podatkovni sloj uporablja lastne 48-bitne naslove za naslavljanje paketov. Delovanje ARP protokola bomo najlažje razložili s pomočjo Slike 8 [9].



Slika 8 – ARP proces pridobivanja strojnega naslova

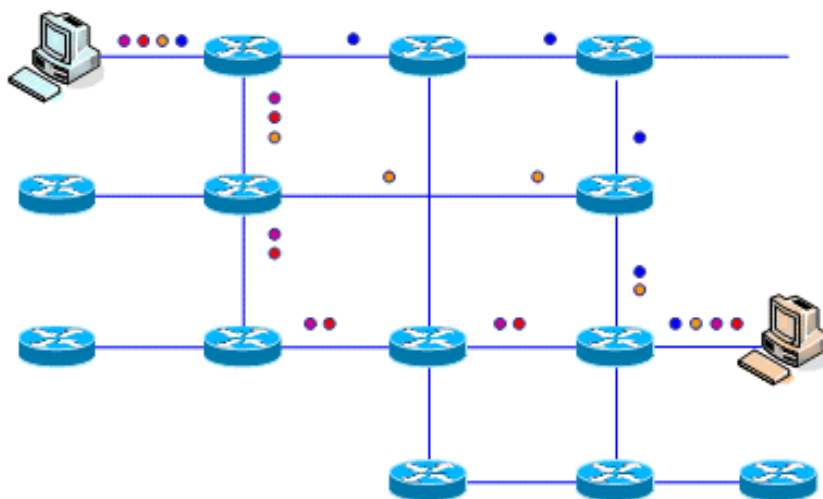
Naprava, ki bi želela poslati paket, najprej vsem gostiteljem pošlje ARP povpraševanje (ang. ARP Request). ARP povpraševanje vsebuje naslovni IP naslov s sporočilom »Ali v omrežju obstaja naprava s tem IP naslovom? Prosim, sporoči mi svoj strojni naslov«. Ko naslovna naprava razpozna svoj IP naslov, pošiljatelju preko ARP odgovora (ang. ARP Reply) sporoči svoj strojni naslov. Pošiljatelj le-tega nato uporabi za pošiljanje podatkov naslovniku.

Za uspešno delovanje ARP protokola je ključno, da gostitelji vzdržujejo ARP predpomnilnik (ang. ARP Cache). Le-ti hranijo povezave med IP naslovi in strojnimi naslovi, ki so bile vzpostavljene v roku zadnjih 20 minut.

2.7 UDP protokol

Nepovezovalni protokol (ang. User Datagram Protocol) [1, poglavje 12, 4, poglavje 11] se uporablja za pošiljanje datagramov gostiteljem v omrežju brez potrebe po predhodnem komuniciranju, ki se uporablja za vzpostavitev posebnih prenosnih kanalov. Za delovanje uporablja vrata in vtičnice (ang. Socket) o katerih bomo govorili v naslednjem razdelku.

UDP protokol uporablja enostaven transportni model, ki ne vključuje protokola rokovanja (ang. Handshake Protocol). To pomeni, da le-ta vzpostavi nezanesljivo povezavo, pri kateri se datagrami lahko izgubijo, pridejo na cilj v napačnem vrstnem redu ali so podvojeni, kar prikazuje Slika 9 [10].



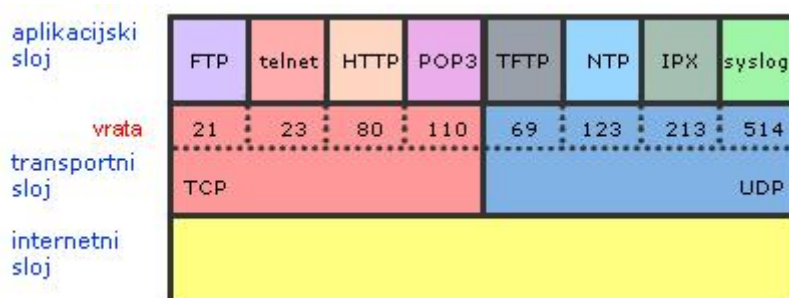
Slika 9 – Pošiljanje podatkov pri uporabi UDP protokola

UDP predpostavlja, da preverjanje in odpravljanje napak nista potrebna, kar opazno zmanjša kompleksnost in poveča hitrost delovanja. Zaradi teh dveh pozitivnih lastnosti ga uporabljamo v časovno-kritičnih aplikacijah. Uporabljajo ga številne internetne aplikacije, pri katerih morajo biti sporočila kratka ter sestavljena iz enega

paketa za poizvedbo in enega paketa za odgovor. Med te aplikacije spada DHCP (Dynamic Host Configuration Protocol) o katerem bomo povedali več v nadaljevanju.

2.7.1 Vrata in vtičnice

Vrata [4, poglavje 1.9] so programska struktura, ki določa končno točko v komunikaciji. Uporabljata jo tako UDP kot tudi TCP protokol. Vrata so identificirana s 16-bitno številko vrat. Prvih 1023 vrat, ki jih imenujemo dobro poznana vrata (ang. well-known ports) je rezerviranih za pogosto uporabljene aplikacije (POP3, HTTP, FTP). Ostala se uporabljajo za poljubne namene komuniciranja. Slika 10 [11] prikazuje skrčen seznam vrat, ki jih uporabljata UDP in TCP protokola.



Slika 10 – Vrata, ki jih uporabljata TCP in UDP protokola

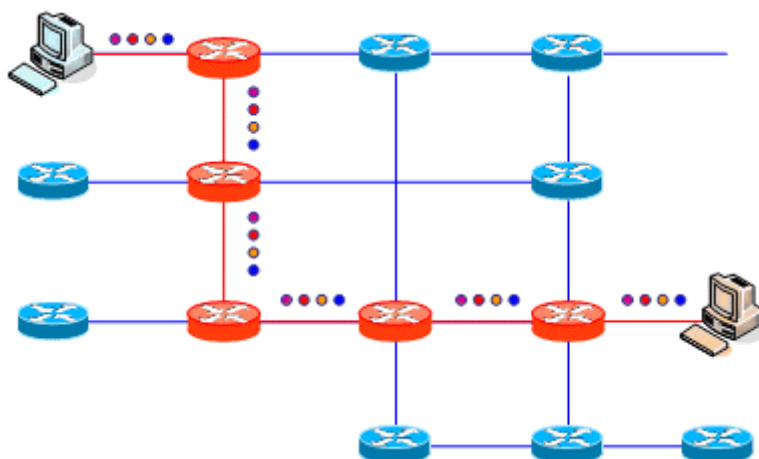
Oba protokola imata vsak svoj seznam vrat. Če neka aplikacija uporablja za svoje delovanje samo TCP protokol (FTP z vrati 21), lahko ta vrata istočasno koristi druga aplikacija, ki uporablja za delovanje samo UDP protokol.

Vtičnica je kombinacija IP naslova in številke vrat, ki se uporablja za vzpostavitev povezave z določenim gostiteljem. Vtičnica 192.168.2.5:80 naslavlja vrata 80, ki določajo HTTP strežnik, kateri teče na napravi z IP naslovom 192.168.2.5.

2.8 TCP protokol

TCP [1, poglavje 13, 4, poglavja 17-24] je povezavno orientiran protokol (ang. connection-oriented), ki zagotavlja zanesljiv pretok podatkov od aplikacije na eni napravi do aplikacije na drugi napravi.

Tipična analogija, ki se uporablja za lažje razumevanje povezavno orientiranega protokola, je telefonski klic, kjer je potrebno najprej poklicati naslovnika in počakati, da se javi, šele nato lahko povemo, kdo kliče in se začnemo pogovarjati. Pri TCP povezavi sodelujeta natanko dva gostitelja, kar pomeni, da protokol ne podpira povezave s skupino gostiteljev ali vsemi gostitelji na lokalnem omrežju. O vzpostavitvi povezave bomo podrobneje govorili v naslednjem razdelku. Slika 11 [12] prikazuje pretok podatkov pri povezavno orientiranem protokolu.

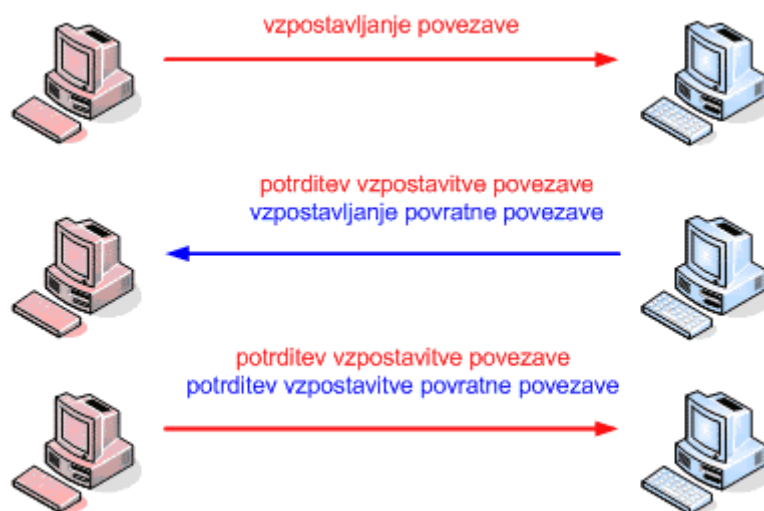


Slika 11 – Pošiljanje podatkov pri uporabi TCP protokola

TCP protokol zagotavlja zanesljivo povezavo na različne načine. Aplikacijske podatke razdeli na TCP segmente, katerih dolžina je najprimernejša za pošiljanje. Naslovna naprava potrdi uspešno sprejet segment s pravo vsebino s sistemom potrjevanja. V nasprotnem primeru bo pošiljatelj še enkrat poslal enak segment. Poleg tega sprejemna stran prepozna in zavrže podvojene prejete pakete. TCP sloj postavi prejete segmente v pravi vrstni red in jih preda aplikacijskemu sloju. Uporablja nadzor pretoka, o katerem bomo govorili v enem od naslednjih razdelkov.

2.8.1 Vzpostavljanje in prekinjanje povezave

TCP protokol uporablja za vzpostavitev povezave metodo trosmernega rokovanja (ang. three-way handshake) [4, poglavje 18]. Vzpostavitev seje bomo razložili s pomočjo Slike 12 [13], ki prikazuje potek trosmernega rokovanja.



Slika 12 – Trosmerno rokovanje za vzpostavitev seje

V prvi fazi pošlje odjemalec (rdeči računalnik) strežniku (modri računalnik) zahtevo za vzpostavitev povezave. V njej je podana številka vrat, kamor se želi povezati, in naključna številka, ki bo v nadaljevanju služila razvrščanju prejetih segmentov na strani strežnika. Poleg potrdila o uspešno sprejetem segmentu pošlje strežnik odjemalcu zahtevo za vzpostavitev povratne povezave, kamor doda naključno številko za razvrščanje prejetih paketov na strani odjemalca. Ta v zadnji fazi potrdi uspešno sprejet segment in seja je vzpostavljena od te točke naprej.

Prekinitev povezave poteka v štirih korakih, ker je seja dvosmerna. To pomeni, da je sestavljena iz dveh ločenih enosmernih povezav, ki tečeta v nasprotnih smereh. Vsaka od enosmernih povezav se prekine v dveh korakih. Pravilo je, da lahko katerakoli stran, ki želi prenehati pošiljati podatke, prekine povezavo.

2.8.2 Nadzor pretoka

Protokol nadzora pretoka se uporablja za izogibanje pošiljanja segmentov s hitrostjo, pri kateri jih sprejemna naprava ni sposobna zanesljivo prejeti in obdelati. To pride do izraza pri komunikaciji med dvema napravama, ki do omrežja dostopata z različnima hitrostima (osebni računalnik in pametni telefon). Za nadzor pretoka uporablja TCP protokol sistem drsečega okna (ang. sliding window), ki ga prikazuje Slika 13 [14].



Slika 13 – Drseče okno

Sprejemna naprava doda vsakemu potrditvenemu sporočilu informacijo o velikosti naslednjega segmenta (v zlogih), ki ga je pripravljena sprejeti. Pošiljatelj sme poslati v naslednjem segmentu največ toliko podatkov, kot je sprejemna stran sporočila v potrditvenem sporočilu. V nadaljevanju pošiljatelj čaka s pošiljanjem, dokler sprejemna naprava ne pošlje potrditvenega sporočila in z njim novega podatka o dovoljeni velikosti naslednjega segmenta.

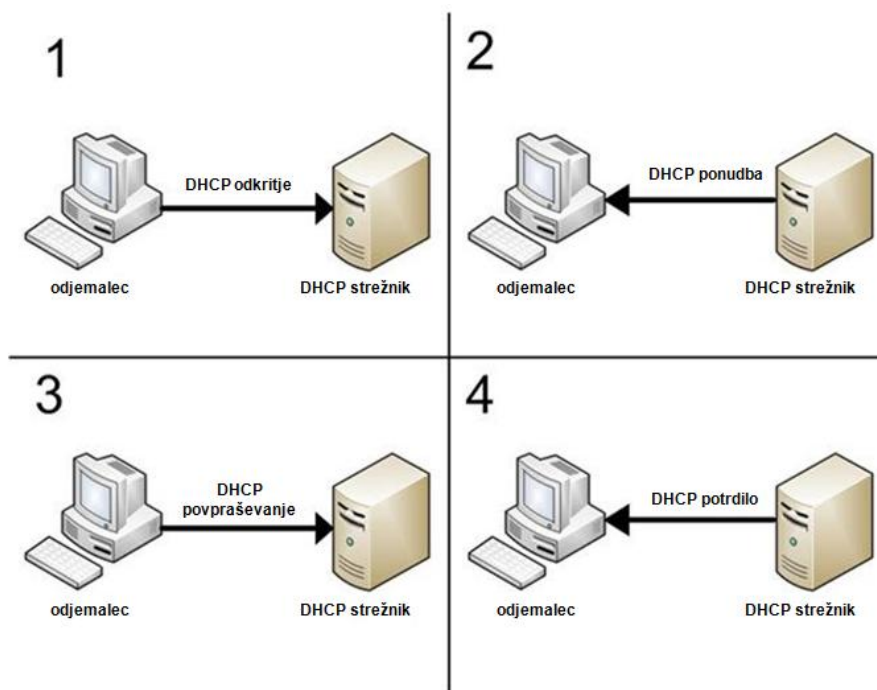
2.9 DHCP protokol

DHCP (Dynamic Host Configuration Protocol) [1, poglavja 23.8-23.18] je protokol za samodejno konfiguriranje gostiteljskih naprav s pomočjo enega ali več DHCP strežnikov. Za delovanje uporablja UDP protokol. DHCP omogoča avtomatično izdajanje IP naslovov, kar pomeni, da omrežnemu administratorju ni potrebno posredovati vsakič, ko se v omrežje priključi nova naprava. Protokol vzdržuje bazo podatkov o napravah, ki so priključene v omrežje in s tem preprečuje, da bi dve napravi dobili enak IP naslov.

DHCP strežnik omogoča tri načine dodeljevanja IP naslovov. Pri *dinamičnem načinu* je napravi ob povezavi v omrežje dodeljen IP naslov, ki ga po končani uporabi vrne.

V *samodejnem načinu* DHCP strežnik hrani seznam preteklih dodeljenih IP naslovov, da dodeli napravi enak IP naslov, kot ga je že imela. Administrator pri *statičnem načinu* ročno vpiše relacije med strojnimi naslovi naprav in IP naslovi.

Delovanje DHCP protokola si bomo najlažje predstavljali s pomočjo Slike 14 [15].



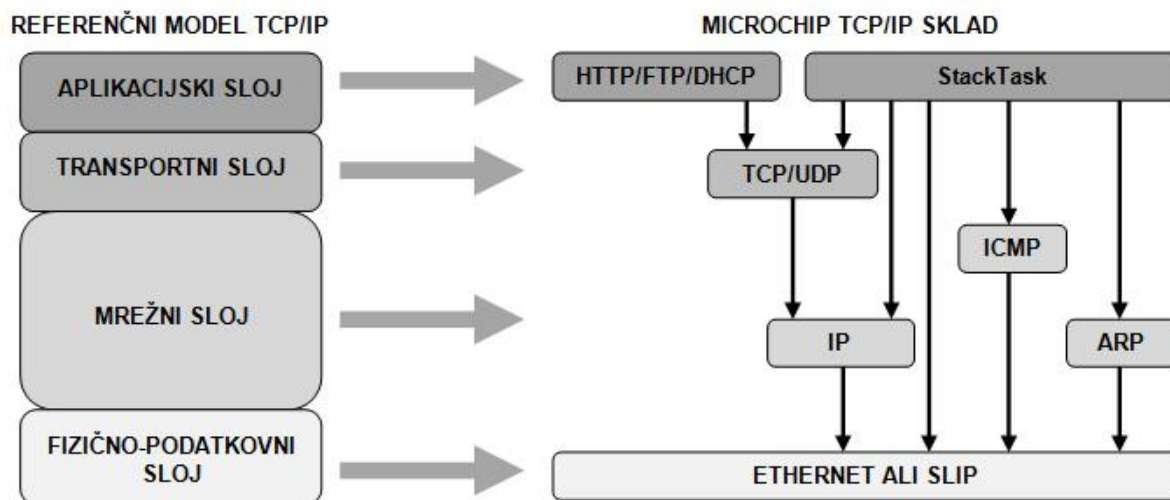
Slika 14 – DHCP samodejno dodeljevanje IP naslova

Naprava z vključitvijo v omrežje odda sporočilo, ki se imenuje DHCP odkritje (ang. DHCP discovery). S tem sporočilom išče razpoložljive DHCP strežnike. Ko DHCP strežnik prejme sporočilo nove naprave, rezervira prost IP naslov in ga z dodatnimi podatki o omrežju pošlje odjemalcu. Tako sporočilo se imenuje DHCP ponudba (ang. DHCP offer). Odjemalec sporoči izbiro DHCP strežnika z DHCP povpraševanjem (ang. DHCP request). DHCP strežnik v zadnjem koraku pošlje odjemalcu DHCP potrdilo (ang. DHCP acknowledgement) z dodeljenim IP naslovom, časom njegove uporabe in ostalimi informacijami, ki jih je odjemalec zahteval.

3 Delovanje in uporaba Microchip TCP/IP sklada

Microchip TCP/IP sklad [16], [17] je odprtokodni skupek mrežnih protokolov, ki z obravnavo skoraj vseh interakcij med omrežnim vmesnikom in aplikacijo zagotavljajo uporabniku osnovo za razvoj najrazličnejših vgrajenih aplikacij, ki temeljijo na modelu TCP/IP. Vključuje module pogosto uporabljenih protokolov v aplikacijskem sloju, kot so HTTP, SMTP za pošiljanje e-pošte, Telnet, FTP in druge. Sklad vključuje poleg omenjenih tudi visoko zmogljivi implementaciji TCP in UDP protokolov, ki na uporabljenem krmilniku ne zasedeta veliko prostora in ostale pogosto uporabljene protokole, kot so IP, ICMP, DHCP, ARP in DNS.

Microchip TCP/IP sklad je zasnovan na modularni zgradbi, zato uporabnikom ni potrebno poznati podrobnega delovanja omrežnih protokolov. Poleg modularne zgradbe je skladu dodanih veliko primerov najrazličnejših aplikacij, med drugimi HTTP strežnik, Ping strežnik in odjemalec, aplikacija za pošiljanje e-pošte in druge. Le-te lahko uporabnik z malo truda prilagodi svojim željam in zahtevam, saj je celoten sklad napisan v programskem jeziku C [18]. Slika 15 [16, str. 2] prikazuje primerjavo modularne zgradbe sklada in referenčnega modela TCP/IP.



Slika 15 – Modularna zgradba TCP/IP sklada

TCP/IP sklad je tako kot referenčni model TCP/IP razdeljen na več slojev. Programski moduli, ki opisujejo določene sloje, so vsebovani v ločenih izvornih

datotekah. Potrebno je omeniti, da se Microchipov sklad razlikuje od referenčnega modela v dejstvu, da ima mnogo slojev neposreden dostop do slojev, ki niso neposredno pod njimi, kar lahko opazimo tudi na zgornji sliki. Naslednje odstopanje od referenčnega modela je v dveh dodatnih modulih, ki se imenujeta StackTask in StackApplications, o katerih bomo govorili kasneje.

3.1 Delovanje sklada

V vsaki aplikaciji, ki uporablja TCP/IP sklad, moramo obvezno vključiti datoteke, ki so navedene v Tabeli 2 [17, poglavje Potrebne datoteke].

Tabela 2 – Datoteke sklada, potrebne za pravilno delovanje aplikacije

Ime datoteke	Uporaba
<i>Main.c, Main.h</i>	Pisanje programa
<i>ARP.c, ARP.h</i>	Določanje strojnih naslovov iz IP naslovov
<i>Delay.c, Delay.h</i>	Zagotavljanje zakasnitev določenim modulom sklada
<i>Datoteke, ki določajo uporabljen fizično-podatkovni sloj</i>	
<i>Helpers.c, Helpers.h</i>	Pomoč nekaterim modulom sklada
<i>IP.c, IP.h</i>	Storitve mrežnega sloja
<i>StackTsk.c, StackTsk.h</i>	Inicializacija in tekoče delovanje sklada
<i>Tick.c, Tick.h</i>	Zagotavljanje časovnika
<i>HardwareProfile.h</i>	Določanje zunanje strojne opreme (Ethernet vmesnik, ...)
<i>TCPIPConfig.h</i>	Nastavitev programskih lastnosti sklada
<i>MAC.h</i>	Zagotavljanje programskih struktur fizično-podatkovnemu sloju
<i>TCPIP.h</i>	Glavna konfiguracijska datoteka

Ostale datoteke, ki jih vključimo v našo aplikacijo, so poljubne in odvisne od zahtev programerja.

Poleg obvezne uporabe nekaterih datotek, je potrebno v aplikacijo vključiti programsko strukturo [19, poglavje 3.1.16] tipa APP_CONFIG, ki jo ponavadi

imenujemo AppConfig. Ta struktura definira uporabljene omrežne naslove, zastavice in omrežno ime naprave. Podatki se lahko vpišejo v strukturo AppConfig na dva načina:

- naložijo se iz statično določenih spremenljivk ali
- nastavijo se preko DHCP protokola, če omrežje to omogoča.

3.1.1 Datoteka *main*

Programer naj bi pisanje aplikacije vedno začel z inicializacijo uporabljene strojne opreme, kot so LED diode, LCD prikazovalniki, serijski vmesnik, oscilator in druge. V naslednjem koraku je potrebno inicializirati strojno opremo, ki jo uporablja sklad. Najprej je potrebno klicati *TickInit()*, ki pripravi časovnik, ki je kritičnega pomena za delovanje sklada. V nadaljevanju sledi inicializacija ostalih strojnih vmesnikov, ki jih sklad nudi programerju, kot je na primer serijska komunikacija z zunanjim EEPROM pomnilnikom.

Po nastavitvi strojnih vmesnikov sledi nastavitev sklada. Ker je večina spremenljivk, ki so nujne za delovanje sklada, shranjenih v strukturi AppConfig, zagotovi programer na tem mestu njihovo inicializacijo.

Ko smo zadostili zgornjim korakom, lahko inicializiramo sklad s klicem funkcije *StackInit()*, ki samodejno izvede inicializacije vseh protokolov, ki jih nameravamo uporabiti in smo jih omogočili v datoteki *TCPIPConfig.h*.

Po končani inicializaciji stopimo v neskončno zanko v kateri se izvajajo vse naloge aplikacije. V tej zanki moramo obvezno redno klicati funkciji *StackTask()* in *StackApplications()*.

Prva izvaja vse časovne operacije, ki jih sklad potrebuje, ter nadzira sprejemanje in pošiljanje paketov. Ta funkcija poleg omenjenega usmerja in predaja sprejete pakete zahtevanim slojem sklada. Druga funkcija se uporablja za klicanje uporabljenih modulov sklada. Ko naprava sprejme ICMP sporočilo, bo ta funkcija samodejno klicala ICMP modul, ki bo obdelal podatke v čakanju.

Vse operacije teh dveh funkcij delujejo po principu stroja stanj (ang. State Machine), ki zagotavlja, da si nobena operacija ne prilasti procesorja za daljše obdobje. Ker je sklad zgrajen po principu kooperativne večopravilnosti (ang. Cooperative Multitasking), je to zelo pomembno za tekoče delovanje vseh ostalih operacij, o čemer bomo več govorili v naslednjem razdelku.

Za vsak klic funkcije *StackApplications()* je potrebno klicati funkcijo *StackTask()*, pri čemer ni zahtevano, da se ti dve funkciji kličeta s predpisano frekvenco. Pri prenizki frekvenci klica funkcije *StackTask()* se zmanjša prepustnost, saj vzame vsak klic iz sprejemnega pomnilnika največ en paket.

Čas, potreben za obhod zanke, ni fiksno določen, saj je odvisen od večih faktorjev. Pri sprejemanju ali pošiljanju paketa bo čas obhoda daljši, kot pri mirovanju.

3.1.2 Kooperativna večopravilnost

Kooperativna večopravilnost [17, poglavje Kooperativna večopravilnost] je metoda uporabljanja večih aplikacij na eni napravi, pri kateri le-ta po končani uporabi prepusti procesor drugi aplikaciji. To nam omogoča navidezno sočasno izvajanje programov.

Če nameravamo v aplikaciji uporabljati načelo kooperativne večopravilnosti, je potrebno vsa opravila zgraditi na način, ki med izvajanjem dovoljuje ostalim aplikacijam dostop do procesorja. Zelo uporaben način gradnje takih operacij je uporaba stroja stanj, ki daljše operacije razdeli na več delov, katere aplikacija izvaja postopoma. Pri tem se naše opravilo kliče periodično, kjer za sledenje stanj uporabljamo zunanjo spremenljivko.

Na tem mestu se nam lahko pojavi vprašanje, kako izvajati procese, ki so časovno odvisni. Predpostavimo, da želimo na vsakih 10 sekund za 5 sekund prižgati LED diodo, medtem ko moramo biti vmes zmožni odgovarjati na ICMP sporočila (Ping). Če bi bili ti dve časovni enoti realizirali z zakasnilnimi zankami, ostala časovno kritična opravila v tem času ne bi imela dostopa do procesorja. Pri takih nalogah uporabljamo za zagotovitev pravilnega delovanja sklada vgrajen časovnik. Pri

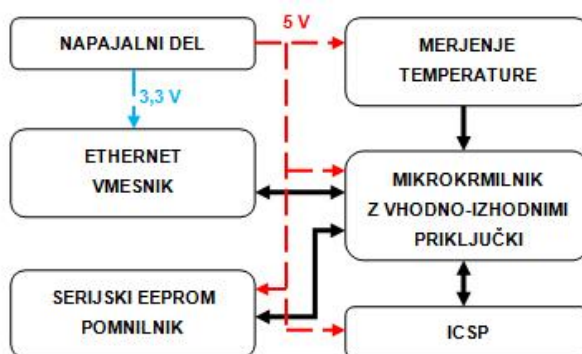
snovanju zakasnitev so nam v veliko pomoč funkcije sklada, ki uporabljajo vgrajen časovnik in makri, ki pretvorijo ljudem razumljive časovne enote v pulze časovnika.

4 Zgradba in delovanje modula

Do sedaj smo v poglavju 2 spoznali teoretične osnove povezovanja naprave v internetno omrežje ter v poglavju 3 govorili o programerski implementaciji modela TCP/IP, katero predstavlja odprtokodni TCP/IP sklad podjetja Microchip. Znanje, pridobljeno skozi ti dve poglavji, lahko s pridom uporabimo za izgradnjo aplikacije, kakršne smo si zamislili v uvodu. V tem poglavju si bomo najprej ogledali modul kot celoto, ki jo bomo v nadaljevanju razčlenili na manjše funkcionalno zaokrožene enote. Zaradi velikosti je električna vezalna shema predstavljena v Prilogi A, medtem ko je spisek uporabljenih električnih elementov predstavljen v Prilogi B.

Začetna zasnova vezja je temeljila na minimalizmu, kar pomeni, da je vezje sestavljeno iz čim manj sestavnih delov. Cilj je bil zgraditi vezje, ki se bo uporabljalo kot pametni vmesnik za regulacijo temperature v prostoru. V začetni fazi načrtovanja je bila naša glavna naloga izbrati primerno kombinacijo mikrokrmilnika in Ethernet vmesnika, ki bo zagotavljala interakcijo z internetnim omrežjem. Odločili smo se za uporabo Microchipovega Ethernet krmilnika ENC28J60 v povezavi z Microchipovim mikrokrmilnikom iz družine 8-bitnih naprav. Ena izmed pomembnejših nalog modula je zadosti natančno zaznavanje temperature ambienta, zato je tej temi namenjeno celotno naslednje poglavje.

Zgoraj opisana zasnova vezja se je tekom gradnje in programiranja razvila v modul, ki poleg regulacije temperature omogoča uporabo za razvojne namene. Vezje je shematsko prikazano na Sliki 16.

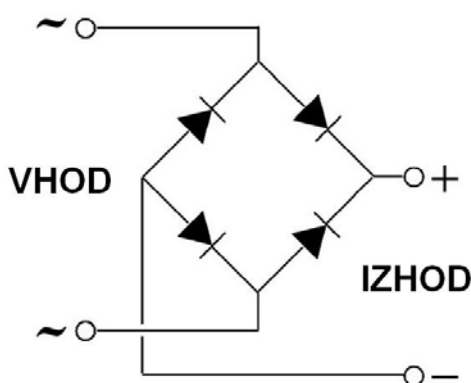


Slika 16 – Shematski prikaz zgradbe modula

To pomeni, da uporabnik, večč programiranja, uporabi modul kot razvojno okolje za lastne aplikacije, ki uporabljajo dostop do internetnega omrežja. Za omogočanje dostopa do mikrokrmilnika smo modulu dodali vezje ICSP (ang. In **C**ircuit **S**erial **P**rograming), ki se uporablja za serijsko programiranje že vgrajenih mikrokrmilnikov. Vezju smo dodali LED diode, ki služijo tako indikaciji določenih dogodkov med primarnim obratovanjem modula, kot tudi testiranju delovanja aplikacije med razvojem. Poleg omenjenih razširitev smo vezju dodali 1 Mbit serijski EEPROM pomnilnik proizvajalca Microchip.

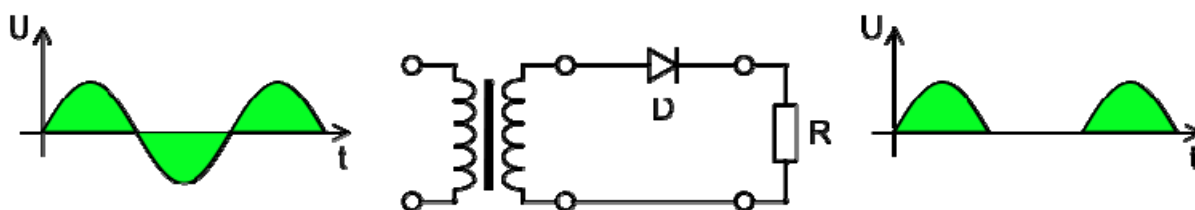
4.1 Napajalni del

Modul zahteva za pravilno delovanje dve napajalni napetosti. Napajalna napetost 3,3 V je potrebna za delovanje Ethernet krmilnika ENC28J60. Napajalna napetost 5 V se uporablja za napajanje mikrokrmilnika ter ostalih integriranih vezij. Ker je napajanje zagotovljeno preko zunanjega napajalnika, obstaja bojazen, da zaradi napačne polaritete priključka uničimo vezje. To nevarnost smo odpravili z uporabo Graetzovega mostiča [20], ki je sestavljen iz štirih diod, kot prikazuje Slika 17 [21].



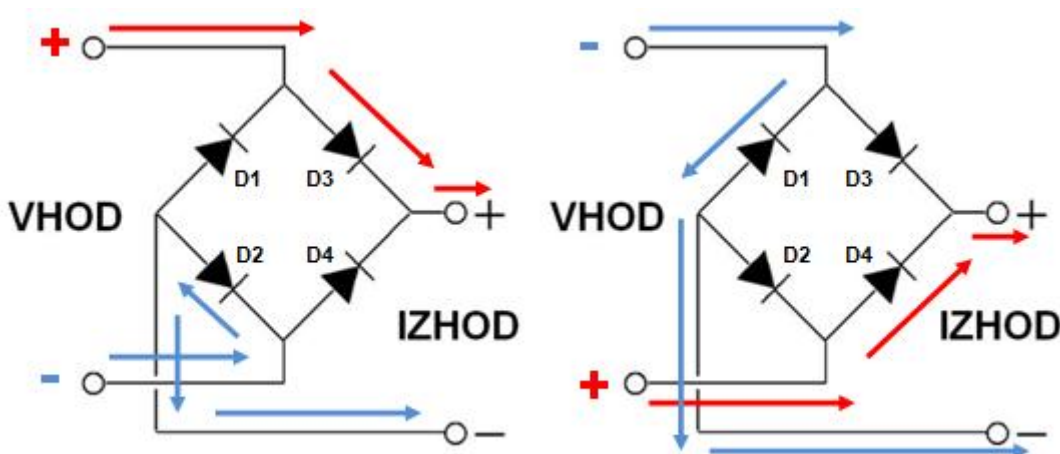
Slika 17 – Graetzov mostič

Greatzov spoj se primarno uporablja za usmerjanje izmenične napetosti v enosmerno. Delovanje diodnega mostiča bomo lažje razumeli, če s pomočjo Slike 18 [20] preučimo vpliv ene diode na izmenično napetost.



Slika 18 – Delovanje diode

Dioda je električni element, ki prevaja električni tok samo v eni smeri, v našem primeru od levega proti desnemu priključku. To pomeni, da bomo zaznali padec napetosti na upor R le v primeru pozitivnih polaritet vhodne izmenične napetosti. Sedaj, ko razumemo, kako deluje dioda, lahko naše razmišljanje razširimo na diodni mostič, katerega delovanje prikazuje Slika 19 [22].



Slika 19 – Delovanje Graetzovega mostiča

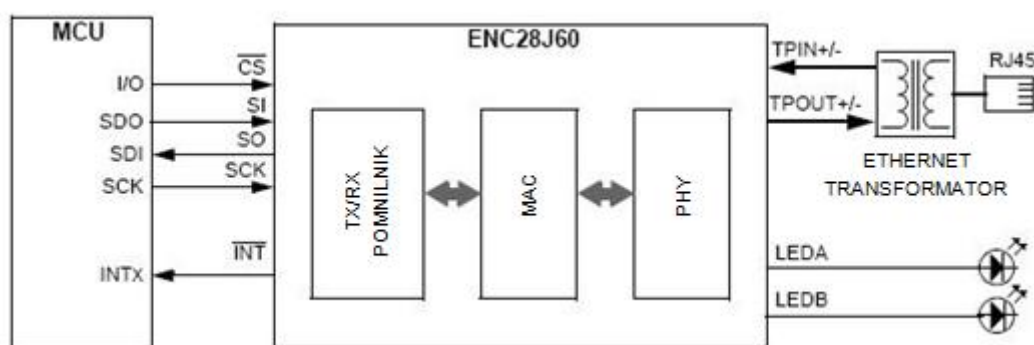
Na levi strani slike je vrh diodnega mostiča na pozitivnem potencialu. Električni tok teče le preko diode $D3$, kar pomeni, da je desni izhodni priključek na pozitivnem potencialu, medtem ko je levi izhodni priključek na negativnem. Desni del slike prikazuje inverzno stanje vhodne napetosti, kjer je na pozitivnem potencialu spodnji del mostiča. Tok teče le preko diode $D4$, kar pomeni, da bo na pozitivnem potencialu desni izhodni priključek in na negativnem levi. Izhodni sponki imata v obeh primerih enako polariteto napetosti. S tem elementom je možnost poškodovanja vezja zaradi napačne polaritete vhodne napetosti odpravljena.

Za Graetzov mostič vežemo napetostna regulatorja LM7805 in LM3940. Prvi se uporablja za zagotavljanje 5 V napetosti in ima na vhodni in izhodni sponki vezana

gladilna kondenzatorja, ki filtrirata visoko in nizkofrekvenčni šum. Izhodna napetost prvega regulatorja služi kot vhodna napetost drugega, ki zagotavlja napajalno napetost Ethernet krmilniku. Za pravilno delovanje je potrebno na izhodno sponko proti masi vezati kondenzator z vrednostjo vsaj 47 μF .

4.2 Ethernet krmilnik ENC28J60

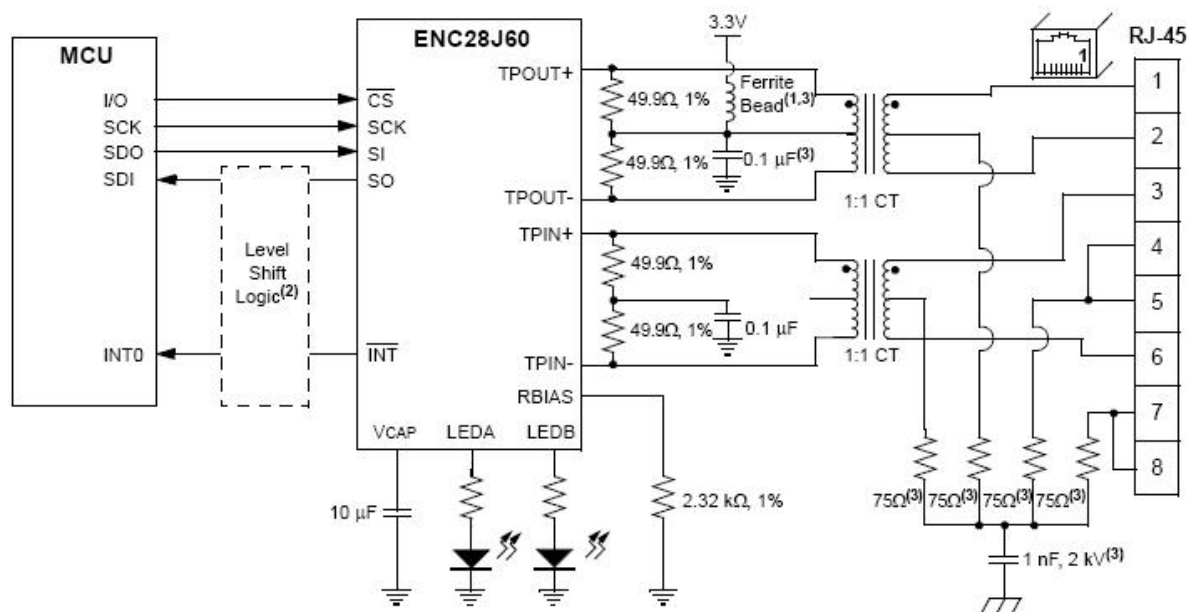
ENC28J60 [23] je samostojen Ethernet krmilnik proizvajalca Microchip, ki uporablja SPI vmesnik za komunikacijo z mikrokrmilnikom. Vključuje vrsto načinov filtriranja prispelih paketov in vgrajen modul za računanje kontrolnih vsot IP datagramov. Shematsko vezavo krmilnika prikazuje Slika 20 [23, str. 4]. Komunikacija z mikrokrmilnikom poteka preko prekinitvene priključne sponke in štirih priključnih sponk SPI, medtem ko za pošiljanje in sprejemanje paketov skrbijo štiri priključne sponke. Za prikazovanje aktivnosti na omrežju povežemo na izhodni priključni sponki dve LED diodi.



Slika 20 – Tipična komunikacijska vezava ENC28J60

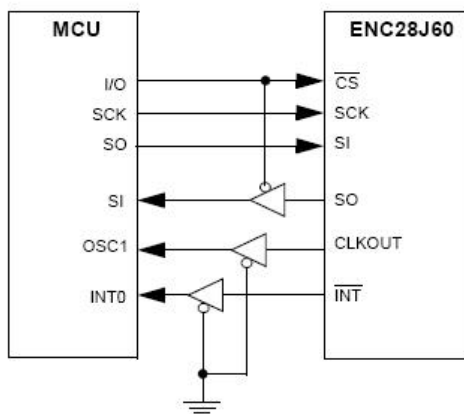
Krmilnik uporablja lasten 25 MHz oscilator, katerega signal je dostopen mikrokrmilniku preko priključne sponke CLKOUT. Frekvenca izhodnega signala se nastavlja preko delilnika, ki omogoča delilna razmerja 1:1, 1:2, 1:3, 1:4 in 1:8. Ob zagonu vsiljuje krmilnik na priključno sponko CLKOUT frekvenco 6,25 MHz, kar je privzeta nastavitvev. Za delovanje uporablja mikrokrmilnik signal oscilatorja iz Ethernet krmilnika, zato želimo uporabljati delilno razmerje 1:1.

Za pravilno delovanje krmilnika je kritično uporabiti pravilen Ethernet konektor, ki vsebuje primerne ločilne transformatorje in nekaj dodatnih električnih elementov, kar prikazuje Slika 21 [23, str. 7].



Slika 21 – Zunanje povezave Ethernet krmilnika ENC28J60

Čeprav uporablja krmilnik 3,3 V napajanje, je bil načrtovan za čim lažjo integracijo v sistem, ki uporablja 5 V napajanje. To pomeni, da vsi vhodni priključki podpirajo 5 V signale brez negativnih posledic. Na strani mikrokrmilnika je potrebno za pravilno zaznavanje napetostnih nivojev zagotoviti ustrezno ojačenje vhodnih signalov. V našem primeru smo to dosegli z uporabo integriranega vezja 74ACT125, katerega vezavo prikazuje Slika 22 [23, str. 8].



Slika 22 – Ojačevanje signalov med ENC28J60 in PIC18F4620

Priključka LEDA in LEDB podpirata samodejno zaznavanje polaritete pri ponovnem zagonu krmilnika. Priključek LEDB se poleg omenjenega uporablja za samodejno nastavljanje tipa povezave krmilnika:

- če je priključek uporabljen kot izvor električnega toka, se nastavi izmenično dvosmerna povezava ali
- če je priključek uporabljen kot ponor električnega toka, se nastavi dvosmerna povezava.

Naša aplikacija uporablja izmenično dvosmerno povezavo, zato smo LED diodi vezali, kot je navedeno v prvi alineji.

4.3 Vhodno-izhodni del z mikrokrmilnikom

Del vezja, ki nadzoruje delovanje modula, je sestavljen iz mikrokrmilnika PIC18F4620. Glavni vzrok, ki je pripeljal do izbire omenjenega mikrokrmilnika, je uporaba Microchipovega TCP/IP sklada, katerega implementacija zahteva relativno velik programski FLASH pomnilnik. V Tabeli 3 [24, str. 1] najdemo glavne karakteristike uporabljenega mikrokrmilnika.

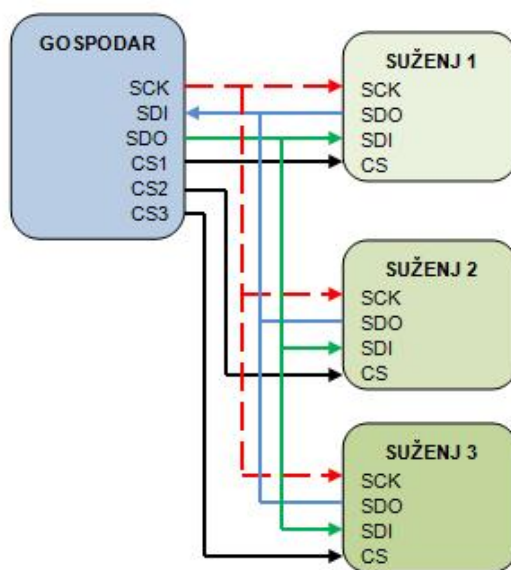
Tabela 3 – Glavne karakteristike mikrokrmilnika PIC18F4620

PIC18F4620	
FLASH pomnilnik	64 kB
SRAM pomnilnik	3986 B
EEPROM pomnilnik	1024 B
Vhodno-izhodni priključki	36
Št. kanalov 10-bitnega AD pretvornika	13
Podpora SPI	DA
Podpora I²C	DA
Priključki EUSART	1
Število primerjalnikov	2
Število časovnikov 8bit/16bit	1/3

Mikrokrmilnik uporablja zunanji 25 MHz urin takt, ki ga generira Ethernet krmilnik ENC28J60. Komunikacija z ostalimi deli modula poteka preko serijskega SPI protokola, ki ga bomo podrobneje obdelali v naslednjem razdelku. Večina neuporabljenih vhodno-izhodnih priključkov je vezanih na moške letvice, kar omogoča uporabniku enostavno razširitev namembnosti modula. Celotno 8-bitno polje registra TRISD se uporablja kot izhod, na katerega je vezanih osem LED diod. Pri primarni namembnosti modula se uporablja samo ena LED dioda, ki prikazuje stanje povezave Ethernet vmesnika v omrežje. Priklop grela je možen preko priključnih sponk 1 in 2 letvice SV1, kjer je priključna sponka 1 uporabljena kot masa.

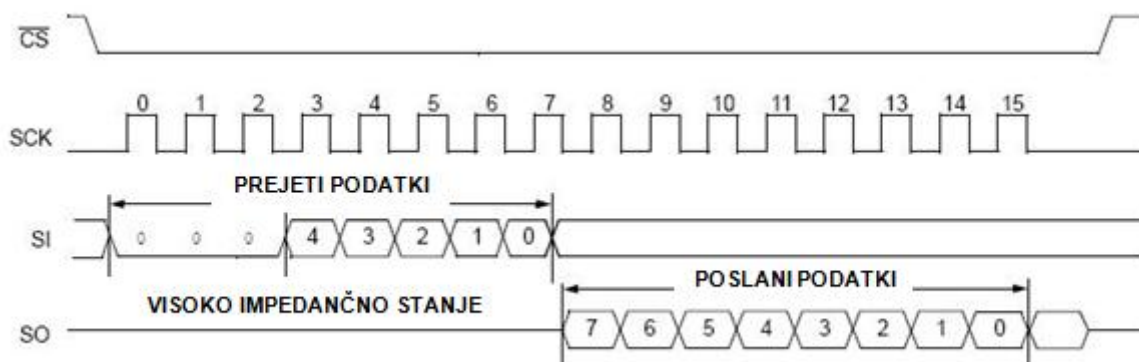
4.3.1 SPI protokol

SPI (Serial Peripheral Interface) [19, poglavje 7.1.1] protokol je predstavnik sinhrona serijske komunikacije, ki uporablja dvosmerno komunikacijo in deluje v načinu gospodar/suženj. Prenos podatkov vedno začne gospodar, medtem ko ostale naprave, ki so konfigurirane kot sužnji, samo odgovarjajo na zahteve gospodarja. Zaradi dvosmerne komunikacije se podatki med napravami prenašajo z veliko hitrostjo. SPI protokol dovoljuje enega gospodarja in enega ali več sužnjev. Pri večjem številu sužnjev si naprave delijo komunikacijske linije, kot prikazuje Slika 23 [19, str. 375].



Slika 23 – SPI komunikacija med gospodarjem in večimi sužnji

Tej komunikaciji se reče tudi štiri-žilna (ang. four-wire) serijska komunikacija, saj vidimo, da se za komunikacijo med gospodarjem in sužnjem uporabljajo štiri povezave. Primer SPI komunikacije med gospodarjem in sužnjem prikazuje Slika 24 [23, str. 27].

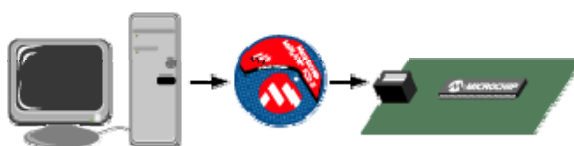


Slika 24 – Primer SPI komunikacije gledano s strani sužnja

Gospodar izbere želenega sužnja s postavitvijo CS linije na nizek nivo. Sužnji, ki ne sodelujejo pri komunikaciji, imajo CS linije na visokem nivoju, kar pomeni, da podatkov ne sprejemajo. Gospodar nato generira ustrezen signal takta, katerega podpira tudi suženj. V naslednjem koraku pošlje sužnju preko SDO linije podatke in ukaz, ki mu pove, kaj mora narediti s temi podatki. Suženj sprejme podatke preko priključne sponke SDI in nato glede na prejeti ukaz ustrezno odgovori gospodarju preko priključne sponke SDO. Le-ta podatke sprejme preko priključne sponke SDI. Po končani komunikaciji gospodar preneha zagotavljati signal takta in na koncu dvigne CS linijo.

4.4 Vmesnik za programiranje modula

Vmesnik za programiranje modula je sestavljen iz konektorja RJ11, ki omogoča direktno priključitev Microchipovega programatorja ICD, kot prikazuje Slika 25 [25].



Slika 25 – Priključitev programatorja ICD preko konektorja RJ11

Razhroščevanje že vgrajenih mikrokrmilnikov omogočata dve namenski liniji PGC in PGD. Na mikrokrmilnik se poleg teh dveh veže linijo za ponovni zagon in dve napajalni liniji. Vezju je dodana tipka, ki je pri samostojnem obratovanja modula namenjena ročnemu ponovnemu zagonu. Le-ta je na MCLR linijo vezana preko diode. Ob sklenitvi tipke se MCLR linija postavi na nizek nivo, kar rezultira v ponovnem zagonu mikrokrmilnika. Pri razklenjeni tipke pull-up upor preprečuje prevajanje diode in ne moti signala na MCLR liniji.

4.5 Serijski EEPROM pomnilnik

Modulu je za shranjevanje podatkov dodan zunanji EEPROM pomnilnik 25LC1024 [26] proizvajalca Microchip, ki z mikrokrmilnikom komunicira preko SPI vodila. Poleg štirih standardnih priključkov za SPI komunikacijo ima čip še dve dodatni krmilni liniji, in sicer HOLD, ki se uporablja za začasno zaustavitev komunikacije ter WP, ki služi preprečevanju pisanja v nadzorni register EEPROM-a.

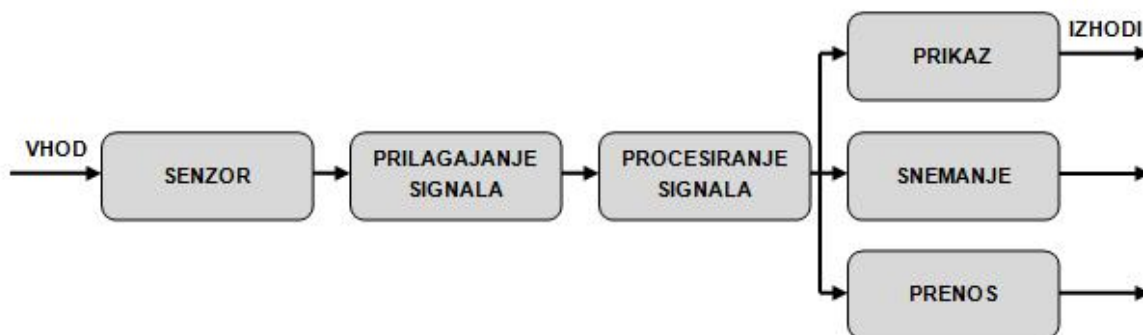
5 Merjenje in regulacija temperature

V prejšnjem poglavju smo spoznali sestavne dele modula, ki uporabniku omogočajo komuniciranje preko internetnega omrežja, programiranje modula ter shranjevanje podatkov. Za doseg zastavljene naloge je potrebno modulu poleg vseh omenjenih funkcionalnosti dodati zmožnost čim bolj natančnega zajemanja temperature prostora in njene regulacije. Ti dve nalogi sta kritični za pravilno delovanje aplikacije, zato si ju bomo v tem poglavju ogledali bolj podrobno.

Za boljšo preglednost bomo nadaljevali s tematiko prejšnjega poglavja, zato bomo najprej razložili zgradbo in delovanje vezja za zajem temperature.

5.1 Merjenje temperature

Merjenje je proces objektivnega prirejanja števil lastnostim merjenega objekta ali dogodka, ki ga lahko predstavimo s splošnim merilnim sistemom, katerega prikazuje Slika 26 [27, str. 6].



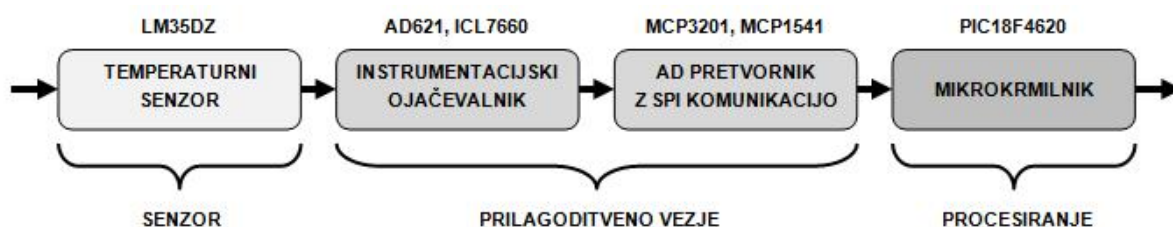
Slika 26 – Splošna zgradba merilnega sistema

Merilni sistem kot celota je razdeljen na več ločenih sklopov, kjer vsak sklop opravlja točno določeno funkcijo. Na modulu so realizirani vsi sklopi merilnega sistema razen prikaza, ki je implementiran na strani strežnika. Poleg tega se merilni sistem od splošnega razlikuje v dejstvu, da sta prenos in prikaz vezana zaporedno in ne

vzporedno, kar je posledica uporabe dveh ločenih sistemov za zajem in prikaz podatkov.

5.1.1 Zgradba merilnega vezja

Načrtovanje vezja je potekalo v skladu s Sliko 25, kjer je bilo potrebno v začetni fazi izbrati primeren senzor za zajem temperature. Zaradi velikega števila različnih izvedb temperaturnih senzorjev na trgu smo že na samem začetku določili nekaj omejitev, ki so pripomogle k hitrejši in lažji izbiri ustreznega senzorja. Prvi izbirni kriterij je bil podatek, da se bo modul uporabljal za regulacijo sobne temperature, kar pomeni, da bi želeli zajemati temperaturo v območju od 10 °C do 40 °C z natančnostjo vsaj ± 1 °C. Druga omejitev je bila določena na podlagi lažje nadaljnje obdelave signala, zato je zaželeno, da je izhod senzorja kalibriran na Celzijevo temperaturno skalo in linearno odvisen od vhodne temperature. Poleg omenjenih strojnih zahtev želimo uporabiti senzor, ki je široko dostopen in ima nizko ceno. Povezavo med sklopi merilnega sistema in zgradbo vezja si bomo lažje predstavljali, če si bomo ogledali Sliko 27, ki shematsko prikazuje zgradbo merilnega vezja.



Slika 27 – Shematska zgradba vezja

S pomočjo teh omejitev smo izbrali precizni napetostni temperaturni senzor LM35DZ [28] proizvajalca National Semiconductor z naslednjimi karakteristikami:

- Linearen izhod +10 mV/°C
- Zagotovljena natančnost $\pm 0,5$ °C pri 25 °C
- Območje uporabe od -55 °C do 150 °C
- Območje napajanja od 4 V do 30 V
- Majhno samosegrevanje (0,08 °C v mirnem ozračju)
- Tipična nelinearnost $\pm 0,25$ °C

V proizvajalčevem katalogu [28, str. 1 in 7] najdemo podatek, da za pravilno delovanje senzorja ne potrebujemo veliko dodatnih elementov.

Zajeto informacijo je potrebno pretvoriti v obliko, ki je primerna za procesiranje. Potrebno se je zavedati, da se operacije na mikrokrmilniku izvajajo na diskretnem nivoju, zato z AD pretvornikom pretvorimo analogen izhod senzorja v digitalno obliko. Za čim bolj natančno pretvorbo je priporočljivo uporabljati celotno vhodno napetostno območje AD pretvornika, zato šibek izhodni signal senzorja najprej ojačimo. Poleg ojačitve je potrebno območje napetosti kalibrirati glede na zahtevano temperaturno območje. Pri spodnji meji temperaturnega območja ($10\text{ }^{\circ}\text{C}$) želimo napetost 0 V in pri zgornji meji temperaturnega območja ($40\text{ }^{\circ}\text{C}$) maksimalno napetost.

Izhodni signal iz senzorja ojačimo in zamaknemo z instrumentacijskim ojačevalnikom [29, poglavje 3], ki zaradi svoje zgradbe ponuja visok rejekcijski faktor (ang. Common Mode Rejection Ratio) [30, str. 5]. Le-ta je definiran kot razmerje med diferenčnim in sofaznim ojačenjem, pri čemer večja vrednost rejekcijskega faktorja pomeni kvalitetnejši ojačevalnik. Ker se naš merilni sistem uporablja za zajem temperature, je bilo potrebno izbrati instrumentacijski ojačevalnik, katerega parametri so čim bolj temperaturno neodvisni. Izbrali smo instrumentacijski ojačevalnik AD621 [31] proizvajalca Analog Devices, ki nam ponuja naslednje karakteristike:

- Enostavno nastavljivo ojačenje 10 ali 100
- Široko območje napajanja med $\pm 2,3\text{ V}$ do $\pm 18\text{ V}$
- Največ $0,15\%$ skupna napaka ojačenja
- Skupno temperaturno lezenje ojačenja $\pm 15\text{ ppm}/^{\circ}\text{C}$
- Največ $125\text{ }\mu\text{V}$ skupnega napetostnega premika
- Skupno temperaturno lezenje napetostnega premika $\pm 1\text{ }\mu\text{V}/^{\circ}\text{C}$

Le-ta uporablja za pravilno delovanje dvojno napajanje, kar pomeni, da moramo poleg pozitivne napajalne napetosti generirati tudi enako veliko negativno napajalno napetost. To dosežemo z uporabo CMOS napetostnega pretvornika ICL7660 [32].

Izhod instrumentacijskega ojačevalnika povežemo direktno na AD pretvornik. Zaradi možnosti uporabe modula kot razvojno okolje smo se odločili za zunanjo izvedbo. Izbrali smo 12-bitni AD pretvornik MCP3201-B [33] proizvajalca Microchip, ki

uporablja SPI protokol za komunikacijo z mikrokrmilnikom. Poleg SPI komunikacije ga odlikujejo naslednje karakteristike:

- Psevdo-diferencialni vhod
- Integrirano vzorčno in zadrževalno vezje
- Napajalno območje od 2,7 V do 5,5 V
- Hitrost vzorčenja 100 ksps pri 1,6 MHz urinem taktu
- Napaka nelinearnosti ± 1 najmanj pomembnega bita (ang. LSB)

AD pretvornik uporablja za pravilno delovanje zunanjo referenčno napetost. V proizvajalčevem katalogu [33, str. 12] je specificirano, da napetostna referenca določa analogno vhodno napetostno območje ter velikost najmanj pomembnega bita. Z nižanjem napetostne reference se niža velikost najmanj pomembnega bita in s tem viša ločljivost AD pretvorbe. Izhodni digitalni signal je funkcija vhodnega analognega signala in napetostne reference.

Izbrali smo precizno referenco MCP1541 [34] proizvajalca Microchip, ki na svojem izhodu zagotavlja napetost 4,096 V. Omenimo lahko tudi:

- Začetna natančnost ± 1 %
- Temperaturno lezenje največ ± 50 ppm/ $^{\circ}\text{C}$

Izbrana referenčna napetost definira velikost najmanj pomembnega bita 1 mV, kar pomeni, da smo teoretično sposobni zajemati temperaturo prostora z ločljivostjo 0,1 $^{\circ}\text{C}$.

Procesiranje zajete temperature se izvaja na mikrokrmilniku, kar bomo podrobneje opisali v naslednjem podpoglavju.

5.1.2 Dimenzioniranje elementov vezja za zajem temperature

Za pravilno delovanje elementov, opisanih v prejšnjem razdelku, je potrebno uporabiti nekaj dodatnih pasivnih električnih elementov. Pri izbiri in dimenzioniranju le-teh si najbolj pomagamo s proizvajalčevimi katalogi, v katerih so specificirane priporočljive vezave. Poleg tega večina proizvajalcev v katalog vključi nekaj primerov tipičnih aplikacij uporabe s pripadajočimi vezalnimi shemami.

Proizvajalec temperaturnega senzorja LM35DZ v svojem katalogu [28, str. 1 in 7] priporoča uporabo gladilnega kondenzatorja vrednosti 0,1 μF , ki ga vežemo med napajalni liniji čim bližje priključitvenima nogicama. Iz izhodne linije smo proti masi priključili upor, ki omogoča delovanje senzorja v celotnem temperaturnem območju. Vrednost upora je določena z (1).

$$R_1 = \frac{V_{dd}}{50 \mu\text{A}} = \frac{5 \text{ V}}{50 \mu\text{A}} = 100 \text{ k}\Omega \quad (1)$$

Izhodna napetost instrumentacijskega ojačevalnika je določena z (2),

$$U_{izh} = G \cdot (V^+ - V^-) \quad (2)$$

kjer sta V^+ in V^- neinvertirajoči in invertirajoči vhod instrumentacijskega ojačevalnika ter G vrednost ojačenja. Brez uporabe ojačevalne stopnje dobimo

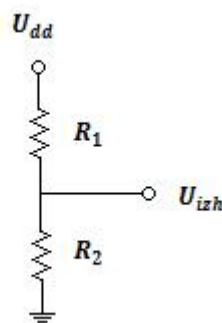
- pri $T = 10^\circ\text{C}$ izhodno napetost 100 mV in
- pri $T = 40^\circ\text{C}$ izhodno napetost 400 mV.

Pri izbiri primerne ojačenja vhodnega signala je potrebno upoštevati vhodno napetostno območje AD pretvornika, ki je določeno z uporabljenimi referenčnimi napetostjo ter napajalno napetost instrumentacijskega ojačevalnika, ki znaša $\pm 5 \text{ V}$. Proizvajalčev katalog instrumentacijskega ojačevalnika [31, str. 2] podaja, da je le-ta zmožen generirati izhodno napetost v območju od -3,9 V do 3,8 V, kar pomeni, da za ojačitev signala uporabimo vrednost ojačenja 10. Pri predpostavki, da je invertirajoči vhod instrumentacijskega ojačevalnika vezan na maso, dobimo

- pri $T = 10^\circ\text{C}$ izhodno napetost 1 V in
- pri $T = 40^\circ\text{C}$ izhodno napetost 4 V.

S tako izbranim ojačenjem kršimo zgornjo omejitev izhodne napetosti, ki jo je instrumentacijski ojačevalnik sposoben generirati. To pomeni, da nismo sposobni zajemati celotnega zahtevanega temperaturnega območja.

Rešitev tega problema najdemo v (2), kjer vidimo, da lahko z uporabo invertirajočega vhoda znižamo izhodno napetost. Zaradi lažje nadaljnje obdelave smo se odločili, da na invertirajoči vhod pripeljemo napetost 100 mV, kar rezultira v 1 V nižji izhodni napetosti. Napetost 100 mV realiziramo z uporovnim delilnikom, ki ga prikazuje Slika 28.



Slika 28 – Uporovni delilnik

Izhodna napetost uporovnega delilnika je definirana z (3),

$$U_{izh} = U_{dd} \frac{R_2}{R_1 + R_2} \quad (3)$$

kjer znaša $U_{izh} = 0,1 \text{ V}$, $U_{dd} = 5 \text{ V}$ in $R_1 = 100 \text{ k}\Omega$. Iz (3) izrazimo iskano upornost

$$R_2 = \frac{U_{izh} R_1}{U_{izh} - U_{dd}} = 2,04 \text{ k}\Omega \quad (4)$$

Vrednost upora $R_2 = 2,04 \text{ k}\Omega$ ne obstaja, zato izberemo najbližjo vrednost $R_2 = 2,05 \text{ k}\Omega$. S tem povečamo izhodno napetost delilnika na $100,44 \text{ mV}$, kar je zanemarljiva napaka.

Proizvajalec instrumentacijskega ojačevalnika v katalogu [31, str. 14] priporoča uporabo dveh gladilnih kondenzatorjev vrednosti $0,1 \mu\text{F}$, ki ju vežemo med napajalni liniji in maso.

V proizvajalčevem katalogu napetostnega pretvornika [32, str. 7] je določeno, da le-ta za delovanje uporablja dva polarizirana kondenzatorja vrednosti $10 \mu\text{F}$.

S temi dodatnimi pasivnimi elementi je vezje za zajem temperature v celoti definirano.

5.1.3 Izračun pogreška meritve temperature

Uporabljeni elementi nimajo idealnih karakteristik, kar pomeni, da z dodajanjem elementov vnašamo dodatne napake v meritev. Najprej zapišemo enačbo, ki definira funkcijsko odvisnost izhodne napetosti instrumentacijskega ojačevalnika od

uporabljenih elementov. V (2) namesto V^+ vstavimo izhodno napetost temperaturnega senzorja U_{temp} . Nato izenačimo V^- z izhodno napetostjo napetostnega delilnika U_{izh} in v (2) namesto V^- vstavimo (3). Tako dobimo (5),

$$U_{izh} = G \cdot (U_{temp} - U_{dd} \frac{R_2}{R_1 + R_2}) \quad (5)$$

katero bomo v nadaljevanju uporabili za izračun pogreška [35, poglavje 3.4]. Iz proizvajalčevih katalogov prepišemo podatke, ki jih bomo potrebovali v nadaljevanju:

$$R_1 = 100 \text{ k}\Omega, 0,1 \%, \pm 15 \text{ ppm}/^\circ\text{C}$$

$$R_2 = 2,05 \text{ k}\Omega, 0,1 \%, \pm 15 \text{ ppm}/^\circ\text{C}$$

$$G_{error} = 0,15 \%, \pm 5 \text{ ppm}/^\circ\text{C}$$

$$U_{offset-vhod} = 400 \text{ }\mu\text{V}$$

Za izračun skupne standardne deviacije izhodne napetosti instrumentacijskega ojačevalnika uporabimo (6):

$$M_{U_{izh}} = \sqrt{(\frac{\partial U_{izh}}{\partial G} \cdot M_G)^2 + (\frac{\partial U_{izh}}{\partial T} \cdot M_{U_{temp}})^2 + (\frac{\partial U_{izh}}{\partial U_{dd}} \cdot M_{U_{dd}})^2 + (\frac{\partial U_{izh}}{\partial R_1} \cdot M_{R_1})^2 + (\frac{\partial U_{izh}}{\partial R_2} \cdot M_{R_2})^2} \quad (6)$$

Najprej izračunamo vse parcialne odvode, ki se pojavljajo v (6):

$$\frac{\partial U_{izh}}{\partial G} = U_{temp} - U_{dd} \frac{R_2}{R_1 + R_2} = -0,10044096 \text{ V} \quad (7)$$

$$\frac{\partial U_{izh}}{\partial T} = G \frac{\partial U_{temp}}{\partial T} = 0,1 \frac{\text{V}}{^\circ\text{C}} \quad (8)$$

$$\frac{\partial U_{izh}}{\partial U_{dd}} = -G \frac{R_2}{R_1 + R_2} = 0,200881921 \quad (9)$$

$$\frac{\partial U_{izh}}{\partial R_1} = G \cdot U_{dd} \frac{R_2}{(R_1 + R_2)^2} = 9,842328 \cdot 10^{-6} \frac{\text{V}}{\Omega} \quad (10)$$

$$\frac{\partial U_{izh}}{\partial R_2} = -G \cdot U_{dd} \frac{R_1}{(R_1 + R_2)^2} = -0,48011 \cdot 10^{-3} \frac{\text{V}}{\Omega} \quad (11)$$

Pri izračunu mejnih vrednosti upoštevamo, da bomo modul uporabljali v temperaturnem območju med 10°C in 40°C , kjer predpostavimo, da je povprečna sobna temperatura 25°C . Iz tega sledi, da se lahko temperatura spreminja v območju $\pm 15^\circ\text{C}$. Za mejno vrednost temperaturnega senzorja vzamemo njegovo nelinearnost, medtem ko mejna vrednost napajalne napetosti izhaja iz njenega nihanja, ki je ocenjeno s pomočjo preciznega voltmetra.

$$M_G = G \cdot 5 \frac{\text{ppm}}{^\circ\text{C}} \cdot 15^\circ\text{C} = 750 \cdot 10^{-6} \quad (12)$$

$$M_{U_{temp}} = 0,3^\circ\text{C} \quad (13)$$

$$M_{U_{dd}} = 0,01 \text{ V} \quad (14)$$

$$M_{R_1} = R_1 \cdot 15 \frac{\text{ppm}}{^\circ\text{C}} \cdot 15 ^\circ\text{C} = 22,5 \Omega \quad (15)$$

$$M_{R_2} = R_2 \cdot 15 \frac{\text{ppm}}{^\circ\text{C}} \cdot 15 ^\circ\text{C} = 0,46125 \Omega \quad (16)$$

V (6) vstavimo izračunane vrednosti iz (7) do (16):

$$M_{U_{izh}} = \sqrt{5,6747 \cdot 10^{-9} \text{ V} + 0,9 \cdot 10^{-3} \text{ V} + 4,03535 \cdot 10^{-6} \text{ V} + 49,04 \cdot 10^{-9} \text{ V} + 49,04 \cdot 10^{-9} \text{ V}}$$

$$M_{U_{izh}} = 0,03 \text{ V}$$

Poleg izračunane napake je potrebno upoštevati napako zaradi napetostnega premika instrumentacijskega ojačevalnika in napako AD pretvorbe. Prva je podana glede na vhod instrumentacijskega ojačevalnika, zato jo je potrebno preračunati na izhodno stran. Pri drugi upoštevamo vrednost najmanj pomembnega bita.

$$Pogrešek_{skupni} = M_{U_{izh}} + U_{offset-izhod} + LSB \quad (17)$$

$$U_{offset-izhod} = G \cdot U_{offset-vhod} = 4 \text{ mV} \quad (18)$$

$$LSB = \frac{U_{ref}}{4096} = 1 \text{ mV} \quad (19)$$

$$Pogrešek_{skupni} = 0,03 \text{ V} + 4 \text{ mV} + 1 \text{ mV} = \pm 35 \text{ mV}$$

Celotna napaka meritve znaša **$\pm 35 \text{ mV}$** ali prevedeno v temperaturo **$\pm 0,35 ^\circ\text{C}$** .

Poglaviten razlog za majhen pogrešek je precizijski instrumentacijski ojačevalnik, ki ima skoraj zanemarljiv napetostni premik. Poleg instrumentacijskega ojačevalnika je pomemben majhen koeficient temperaturnega lezenja izbranih elementov. Iz dobljenega rezultata lahko zaključimo, da smo dosegli cilj zajemanja temperature prostora z natančnostjo vsaj $\pm 1 ^\circ\text{C}$.

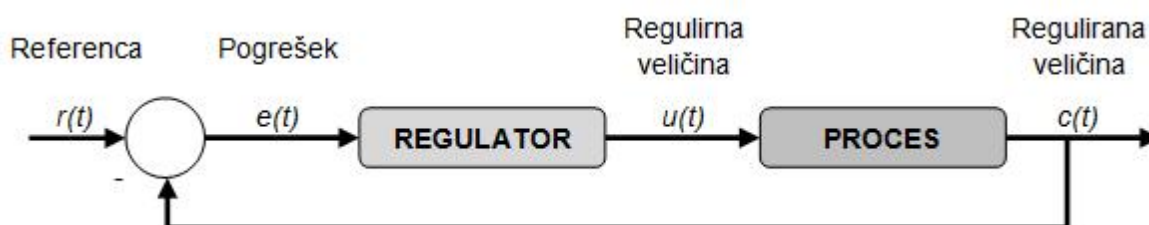
5.2 Regulacija temperature

V tem podpoglavju si bomo ogledali na kakšen način uravnavamo temperaturo v prostoru. Najprej bomo na kratko obdelali osnove zaprtozančnega vodenja z namenom spoznavanja uporabljenih pojmov in diagramov. V nadaljevanju bomo razložili delovanje našega sistema vodenja. Slednje bo zajemalo razlago programske implementacije dvostopenjskega regulatorja in algoritmov, ki se uporabljajo za pridobivanje oziroma generiranje referenčne veličine.

5.2.1 Splošno o sistemih vodenja

Vodenje sistemov [36, str. 1-5] je odprtozančno ali zaprtozančno vplivanje na (realni) objekt z namenom, da dosežemo želene cilje oziroma želeno vedenje objekta, ki ga v teoriji imenujemo proces. V nadaljevanju se bomo osredotočili le na zaprtozančni sistem vodenja, ki ga bomo uporabili v aplikaciji.

Zaprtozančni sistem (regulacijski sistem) se od odprtozančnega (krmilni sistem) razlikuje po povratni zanki, ki poteka od izhodne proti vhodni ali referenčni veličini. Splošni regulacijski sistem prikazuje Slika 29 [36, str. 3].

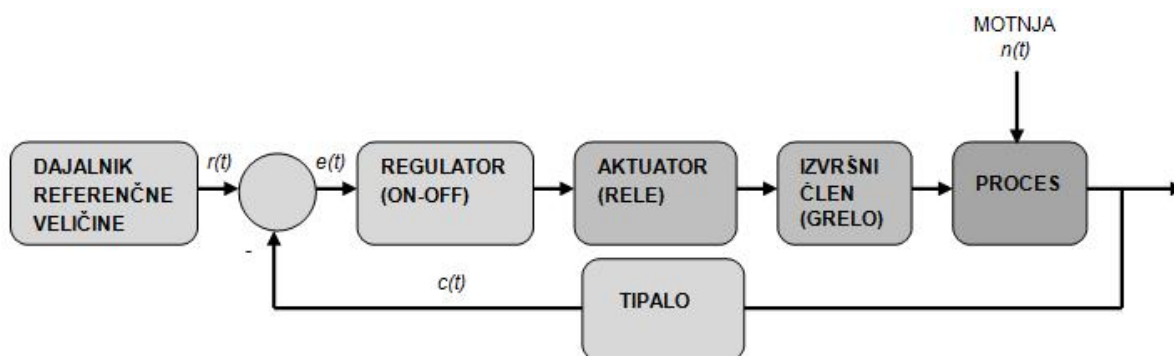


Slika 29 – Bločni diagram zaprtozančnega regulacijskega sistema

Da dobimo točnejšo regulirano veličino $c(t)$, jo moramo primerjati z referenco $r(t)$, regulator pa definira ustrezno regulirno veličino $u(t)$ na osnovi pogreška $e(t)$ tako, da le-ta zmanjšuje pogrešek. Človek je prav gotovo najkompleksnejši tovrstni regulacijski sistem, saj vsebuje nešteto povratnozančnih struktur, ki omogočajo kompleksne operacije in koordinirano delovanje.

Regulacija sobne temperature s pomočjo termostata predstavlja najenostavnejši primer zaprtozančne regulacije. Za razumevanje delovanja predpostavimo, da sta zunanja in notranja temperatura precej nižji od referenčne temperature. V tem primeru je termostat vključen, kar pomeni, da je grelo vključeno in temperatura v sobi narašča. Sistem mora biti načrtan tako, da je toplota, ki jo ustvarja grelo Q_{vh} , bistveno večja od izgub Q_{izh} (odvajanje toplote skozi stene). S tem temperatura v sobi hitro narašča, dokler ni nekoliko višja od želene temperature. Takrat se grelo izključi in temperatura začne počasi upadati, saj na proces deluje le $-Q_{izh}$. Ko upade nekoliko pod želeno temperaturo, se grelo ponovno vključi in celoten cikel se ponovi.

Bločni diagram takega sistema je predstavljen na Sliki 30 [36, str. 4].

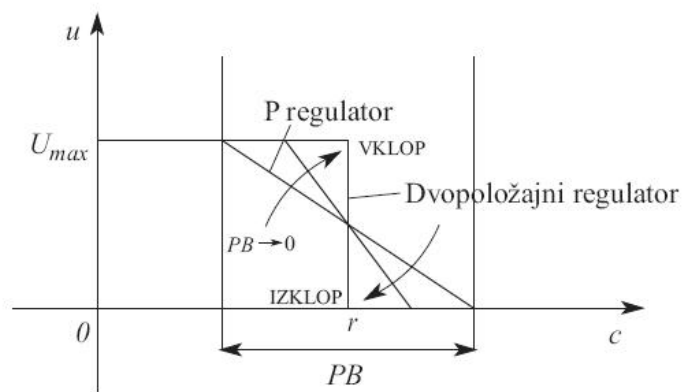


Slika 30 – Bločni diagram zaprtozančnega sistema ogrevanja

Proces je ogrevanje sobe, regulirana veličina je temperatura prostora, medtem ko odvajanje toplote skozi stene, sprememba zunanje temperature, odprtje okna in druge predstavljajo motnjo. Aktuator je naprava, ki signal regulatorja pretvori v signal, ki krmili končni izvršni člen (rele za vklopjanje in izklopjanje električnega grela). Pri pozitivnem pogrešku se grelo vključi ter pri negativnem izključi (če zanemarimo histerezo v regulatorju, ki je potrebna, da zmanjšamo število preklapov). Termostat smo razdelili na štiri bloke, ki so na zgornji sliki označeni z enako barvo: v dajalnik referenčne temperature, v tipalo regulirane veličine, v sumacijsko točko (primerjalnik) ter v regulator.

5.2.2 Delovanje dvopoložajnega regulatorja

Stopenjsko delujoči regulatorji se od zveznih razlikujejo v regulirni veličini, ki lahko zavzame samo določena stanja. V našem primeru dvostopenjskega regulatorja [36, str. 254-267] lahko regulirna veličina zavzame dve stanji, in sicer vklopljeno ter izklopljeno. S tako izvedbo regulatorja je možno dobiti hitrejši odziv ali krajši čas vzpona, saj že relativno majhna vrednost pogreška pripelje izvršni sistem v nasičenje. S povečevanjem ojačenja proti neskončnosti, kar je enako zmanjševanju proporcionalnega območja (PB) proti nič, zvezni proporcionalni (P) regulator spremenimo v dvopoložajni stopenjski regulator, kar prikazuje Slika 31 [36, str. 255].



Slika 31 – Nastanek dvopoložanega regulatorja iz zveznega P regulatorja

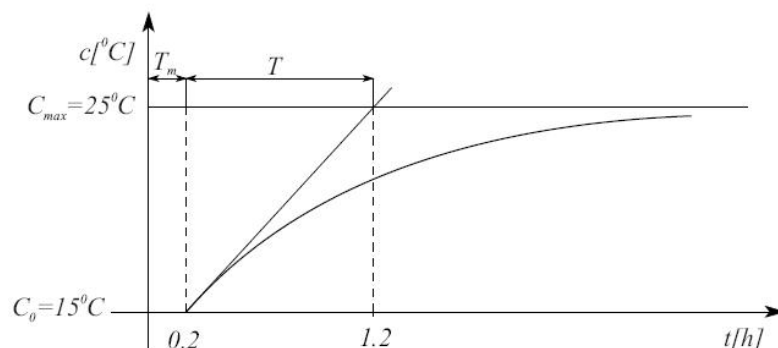
Dvopoložajni regulator generira pri pozitivnem pogrešku, kjer je referenčna veličina večja od regulirane, polni regulirni signal U_{max} , kar smatramo kot vklop izvršnega sistema. Pri negativnem pogrešku je referenčna veličina manjša od regulirane, zato dvopoložajni regulator generira vrednost 0, kar smatramo kot izklop izvršnega sistema. Tak regulator dobro regulira procese, ki nimajo večjih zakasnitev, vendar pri tem izredno hitro preklaplja, kar lahko pripelje do poškodbe ali v skrajnem primeru, do uničenja izvršnega sistema. Realni dvopoložajni regulatorji uporabljajo pri svojem delovanju histerezo, ki zagotavlja različni vrednosti za vklop in izklop izvršnega člena. Vklop izvršnega sistema nastopi pri nekoliko pozitivnem pogrešku ($c < r$), izklop pa pri nekoliko negativnem pogrešku ($c > r$). Pri tem z Δc označujemo velikost histereze.

Dvopoložajno regulacijo odlikuje nizka cena izvedbe, medtem ko ima z vidika regulacije kar nekaj pomanjkljivosti, kot so nihanje regulirane veličine, obraba kontaktov ter v primeru močnostnih izvršnih sistemov, udari v omrežju zaradi vklapljanja in izklapljanja.

5.2.3 Dvostopenjska regulacija temperature

Za boljše razumevanje dvopoložajne regulacije si zamislimo naslednji termični proces: če grelnik ne deluje, je temperatura v prostoru enaka temperaturi okolice C_0 (15 °C). Če pri tej temperaturi vključimo grelnik, ki daje regulirno veličino $u = U_{max}$ (5 kW), se temperatura v prostoru dvigne po prehodnem pojavu na temperaturo C_{max} (25 °C). Le-ta bo ob izklopu grelnika zaradi izgub (odvajanje toplote skozi stene,

okna, ...) padla na začetno vrednost C_o . Proces lahko za majhne spremembe v delovni točki (temperatura okolice) izrazimo s sistemom 1. reda z mrtvim časom. Odziv, s pomočjo katerega pridemo do modela, prikazuje Slika 32 [36, str. 258].



Slika 32 – Dvig temperature zaradi delovanja grela

Če za delovno točko izberemo temperaturo okolice, model dobro opisuje prenosna funkcija

$$G_p(s) = \frac{K}{Ts+1} e^{-T_m s} \quad (20)$$

$$K = \frac{C_{max} - C_o}{U_{max}} = \frac{10^\circ\text{C}}{5\text{kW}} = 2 \frac{^\circ\text{C}}{\text{kW}} \quad (21)$$

$$T = 1 \text{ h} \quad T_m = 0,2 \text{ h}$$

V (20) pomenijo

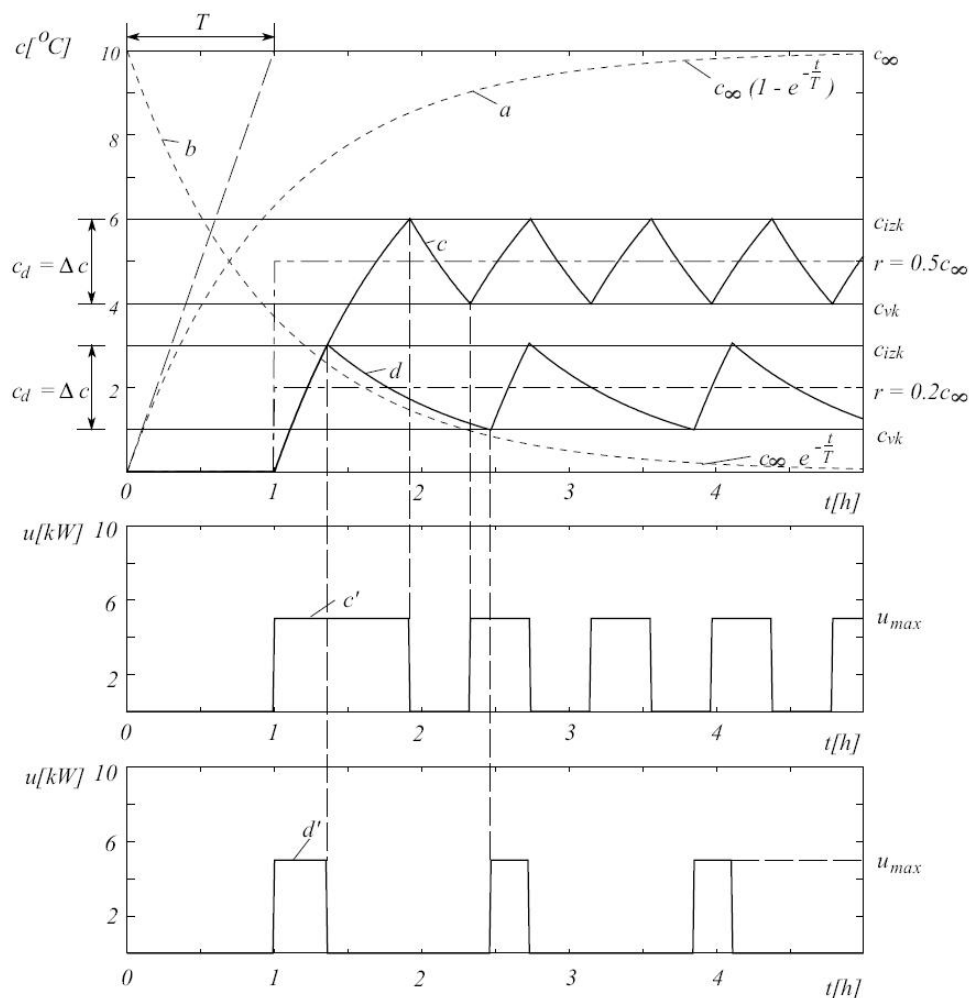
K ojačenje sistema,

T časovna konstanta sistema in

T_m mrtvi čas sistema.

Z vklapljanjem in izklapljanjem grelnika je možno vzdrževati temperaturo med C_o in C_{max} , zlasti pa nekje v sredini tega območja.

Zaradi nelinearnosti sistema je potrebno analizo delovanja razdeliti na dva dela, in sicer na analizo regulacije brez upoštevanja mrtvega časa ter na analizo regulacije brez histereze z dodanim mrtvim časom. Prvi primer prikazuje Slika 33 [36, str. 259].

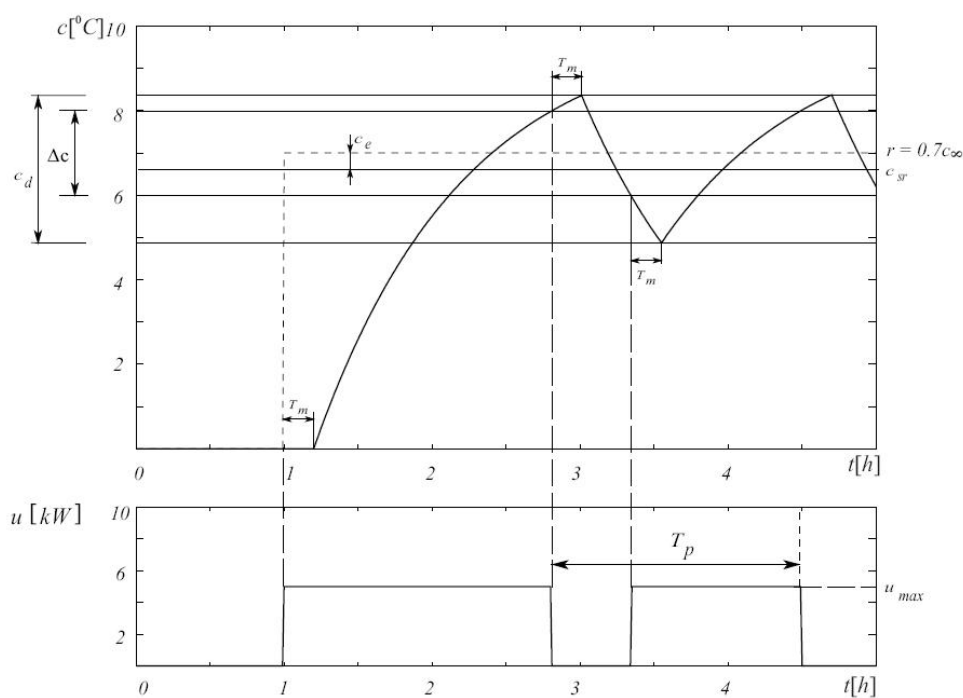


Slika 33 – Potek veličin sistema brez upoštevanja mrtvega časa

Pri vključenem grelu bo temperatura v prostoru brez povratne zanke eksponentialno naraščala po krivulji *a* do vrednosti $c = c_{\infty} = C_{max}$. Če bi pri tej temperaturi gredo izklopili, bi temperatura upadala po krivulji *b* do vrednosti $c = C_0$. S postavitvijo referenčne veličine na vrednost $r = 0,5c_{\infty}$ se bo temperatura spreminjala po krivulji *c*. Regulator zaradi uporabljene histereze grela ne izklopi pri $c = r$, ampak pri vrednosti $c = c_{izk}$. Za vklop grela velja obratno, kjer se le-to vključi pri vrednosti $c = c_{vk}$. Z zmanjševanjem širine histereze se bo povečala frekvenca preklopov. Ta je odvisna tudi od vrednosti referenčne veličine, kar vidimo na primeru krivulje *d*, ki prikazuje spreminjanje temperature pri vrednosti reference $r = 0,2c_{\infty}$. Poleg frekvence preklopov se zmanjša tudi razmerje časov vklopa in izklopa. V tem primeru dobimo ustaljeno stanje, značilno za nelinearne sisteme, ki ga imenujemo *limitni cikel*.

Pri regulaciji temperature brez histereze in z dodanim mrtvim časom le-ta vpliva na sistem zelo podobno kot histereza. Pri pozitivnem pogrešku je gredo vključeno in regulirana veličina se bliža referenčni vrednosti. Gredo se izklopi takoj, ko je pogrešek enak 0, saj regulator ne uporablja histereze. Ker je sistemu dodan mrtvi čas, se izklop grela pozna na regulirani veličini šele po preteku T_m , kar pomeni, da bo v tem času temperatura v prostoru naraščala. Mrtvi čas ima na proces enak vpliv tudi pri ohlajanju sistema. Periodično nihanje poteka drugače, kot v primeru regulacije s histerezo, saj pri referenčni vrednosti $r \neq 0,5c_\infty$ dobimo ustaljeni odmik $c_e = r - c_{sr}$. Ustaljeni odmik pomeni, da vrednosti referenčne veličine in regulirane veličine ne bosta enaki.

Z združitvijo obeh analiz se opisana poteka superponirata in dobimo potek veličin, ki jih prikazuje Slika 34 [36, str. 262].



Slika 34 – Potek veličin sistema pri upoštevanju histereze in mrtvega časa

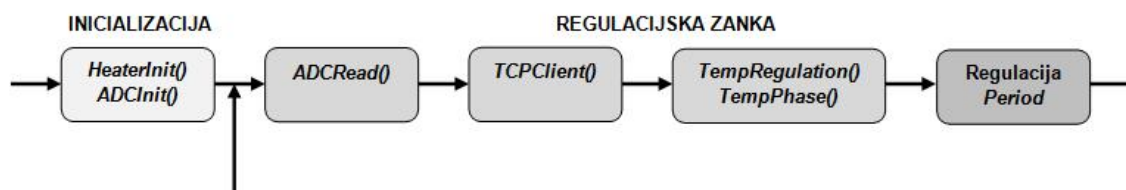
Razmik nihanja c_d se poveča, če povečamo širino histereze Δc , povečamo mrtvi čas T_m , zmanjšamo časovno konstanto T in povečamo ustaljeno vrednost temperature c_∞ . Ker velja $c_\infty = KU_{max}$, je razmik nihanja odvisen od ojačenja sistema K in maksimalne regulirne veličine U_{max} . Če povečamo mrtvi čas, časovno konstanto ter

histerezo, se poveča perioda nihanja T_p . Le-ta je poleg omenjenega odvisna od izbire vrednosti referenčne veličine in bo najkrajša pri $r = 0,5c_\infty$.

5.2.4 Programska realizacija dvostopenjskega regulatorja

Naša aplikacija je zasnovana tako, da je celoten sistem vodenja razdeljen na dva dela, ki sta realizirana na ločenih napravah, povezanih preko internetnega omrežja. Najprej si bomo ogledali regulator, ki je implementiran na modulu, medtem ko si bomo v naslednjem razdelku ogledali, kako je realizirana povezava na strežnik. Ta služi generiranju referenčne veličine, shranjevanju ter prikazu prejetih podatkov.

Delovanje regulacijske zanke bomo najlažje opisali s pomočjo bločnega diagrama, ki ga prikazuje Slika 35.

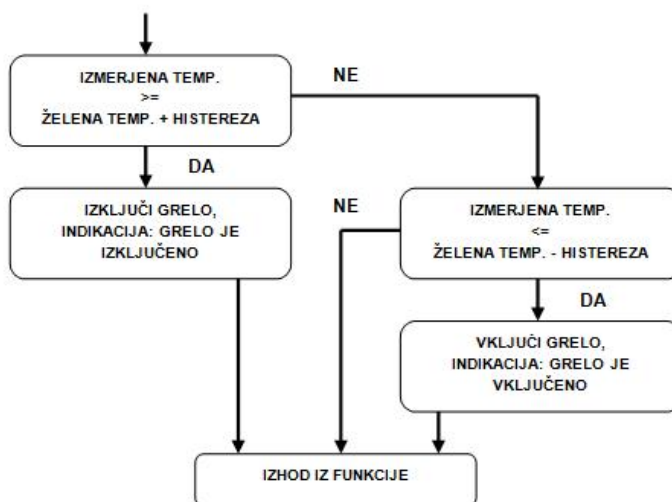


Slika 35 – Bločni diagram regulacijske zanke

Ob zagonu se najprej izvede inicializacija uporabljenih vhodno-izhodnih priključkov ter komunikacije z zunanjim AD pretvornikom, kar realiziramo s funkcijama *HeaterInit()* in *ADCInit()*. Poleg izmerjene temperature v prostoru (*AmbientTemp*) želimo na strežnik poslati informacijo o stanju grela (*Indication*, vključeno/izključeno) ter informacijo o poteku temperature (*TempPhase*, narašča/pada). Vse tri spremenljivke so zajete v programski strukturi *AmbientData*, do katerih dostopamo z ukazom *AmbientData.ImeSpremenljivke*. Funkcija *ADCRead()* dodeli vrednost prvi spremenljivki, funkciji *TempRegulation()* in *TempPhase()* pa drugima dvema. Podatke pošljemo na strežnik med klicem prve in drugih dveh funkcij, kar pomeni, da sta v spremenljivki *TempPhase* in *Indication* pri prvem obhodu regulacijske zanke vpisani naključni številki. Na samem začetku zanke zato poskrbimo, da se ti dve spremenljivki inicializirata na vrednost 0.

Z vstopom v regulacijsko zanko najprej preberemo temperaturo prostora in vrednost zapišemo v spremenljivko *AmbientTemp*. Za regulacijo potrebujemo poleg izmerjene vrednosti temperature še referenčno vrednost in histerezo, ki ju pridobimo s strežnika. Komunikacija s strežnikom je zajeta v funkciji *TCPClient()*, ki se poleg prejemanja informacij uporablja za pošiljanje podatkov na strežnik. Delovanje funkcije si bomo podrobneje ogledali v naslednjem razdelku.

Z uporabo funkcije *TCPClient()* s strežnika prejmemo vrednosti histereze (*Hysteresis*), želene temperature v prostoru (*DesiredTemp*) in frekvence pošiljanja podatkov na strežnik (*Period*). Te tri spremenljivke so združene v programski strukturi *TCPOutputs*. Do posameznih spremenljivk dostopamo na enak način, kot je opisano zgoraj. V funkciji *TempRegulation()* uporabimo poleg vrednosti izmerjene temperature tudi vrednosti histereze in želene temperature. Z uporabo IF pogojnega stavka [18, poglavje 12] preverimo ali je potrebno grelo vključiti ali izključiti. Delovanje funkcije prikazuje bločni diagram na Sliki 36.



Slika 36 – Potek funkcije *TempRegulation()*

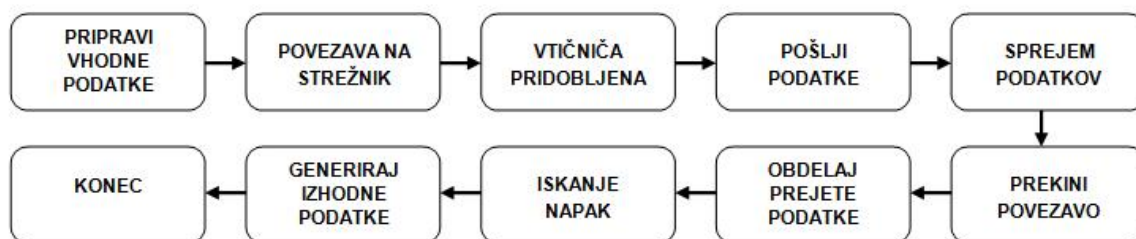
Funkcija se uporablja tudi za generiranje indikacije o stanju grela, ki se nato vpiše v spremenljivko *Indication*. Funkcija *TempPhase()* iz vhodnih podatkov o trenutni in zadnji vrednosti izmerjene temperature preko IF ELSE pogojnega stavka generira informacijo o poteku regulirane veličine. Če je trenutna izmerjena temperatura višja od zadnje izmerjene temperature, lahko zaključimo, da temperatura v prostoru narašča in obratno.

Zadnji blok na Sliki 34 nam pove, da se v nadaljevanju izvaja regulacija temperature. V časovnem okviru, ki je določen s spremenljivko *Period*, v vsakem obhodu zanke izpustimo komuniciranje s strežnikom. Modul v tem časovnem intervalu smatramo kot samostojno zaprto enoto. To si lahko dovolimo, ker imajo termični procesi počasno dinamiko in posledično v času brez komunikacije s strežnikom ne izgubimo veliko informacij.

5.2.5 Komunikacija s strežnikom

Komunikacija s strežnikom je eden najpomembnejših programskih segmentov, ki je potreben za želeno delovanje aplikacije. Na strani modula je celoten algoritem pošiljanja in sprejemanja podatkov zajet v funkciji *TCPClient()*. Na strani strežnika se za komuniciranje ter generiranje in shranjevanje podatkov uporablja skripta *TCPIPConnection.php*. Najprej si bomo ogledali delovanje funkcije na strani modula.

Le-ta je zgrajena iz desetih segmentov stroja stanj, ki jih shematsko prikazuje Slika 37.



Slika 37 – Bločni diagram funkcije *TCPClient()*

- Pripravi vhodne podatke: Uporablja se za pretvorbo številskih vrednosti vhodnih spremenljivk v besedni niz, ki ga bomo v nadaljevanju poslali strežniku. V tem segmentu se generira besedni niz, ki zajema dolžino prvega niza.
- Povezava na strežnik: Povezavo s strežnikom vzpostavimo z vzpostavitvijo vtičnice (pošljemo prvi del trosmernega rokovanja). Poleg tega nastavimo

časovnik, s katerim določimo časovni interval, v katerem mora biti povezava potrjena.

- Vtičnica pridobljena: Če povezava ni bila vzpostavljena v časovnem intervalu, ki je bil določen v prejšnjem segmentu, prekinemo vtičnico in se pomaknemo v prejšnji segment. Če je bila povezava uspešno vzpostavljena se pomaknemo v naslednji segment.
- Pošlji podatke: V odhodnem predpomnilniku zgradimo paket, ki ga želimo poslati. Sestavlja ga glava s podatki, ki jih za delovanje potrebuje TCP sloj in telo z besednim nizom podatkov, ki smo ga generirali v prvem segmentu. Preden izstopimo iz segmenta, pošljemo TCP paket.
- Sprejem podatkov: Iz prejetega TCP paketa, ki ga kot odgovor pošlje strežnik, izluščimo podatke, ki se nahajajo v njegovem telesu. Le-to je od glave ločeno z dvema praznima vrstama, ki ju z besednim nizom zapišemo kot "\r\n\r\n". Besedne nize iščemo s funkcijo *StringSearch()*.
- Prekini povezavo: Če se povezava ni samostojno prekinila, jo prekinemo v tem segmentu.
- Obdelaj prejete pakete: Strežniki, ki za pošiljanje podatkov uporabljajo HTTP verzijo 1.1, razdelijo podatke na več kosov, kar jim omogoča, da začnejo pošiljati podatke, še preden poznajo njihovo dolžino. Tak paket ne uporablja polja o dolžini podatkov, ampak je dolžina posameznih kosov specificirana v telesu paketa. Zato je potrebno, da je klient (modul) sposoben iz posameznih kosov sestaviti prvotno sporočilo, kar dosežemo z uporabo funkcije *ProcessChunkedData()*. Le-ta za delovanje uporablja pravila, ki določajo obliko takega paketa.
- Iskanje napak: Za dodatno preverjanje pravilnosti prejetega paketa je na samem začetku telesa paketa postavljen besedni niz "OK\r\n". Ta zagotavlja, da je strežnik uspešno sprejel in obdelal paket, ki smo ga poslali v segmentu Pošlji podatke.
- Generiraj izhodne podatke: Ker je zaporedje prejetih informacij v paketu točno določeno, s pomočjo funkcije *StringSearch()* iz telesa izluščimo posamezne vrednosti (želena temperatura, histereza in čas regulacije). Le-te pretvorimo v številske vrednosti in jih nato zapišemo v pripadajoče spremenljivke. Te so dostopne kot izhodni podatki funkcije *TCPCClient()*.

- Konec: V zadnjem segmentu določimo, da bo ob naslednjem klicu funkcije na vrsti prvi segment.

Če ni povezave s strežnikom, funkcija obstane med drugim in tretjim segmentom, zato je potrebno ročno spremljati stanje povezave. Komunikacijo prekinemo, če modul v petih poskusih ne vzpostavi povezave s strežnikom in nadaljujemo z regulacijo. Pri tem gre za napako, ki jo bomo obdelali v naslednjem razdelku.

Na strani strežnika se ob prejemu paketa aktivira skripta *TCPIPConnection.php*. Ta iz telesa prejetega paketa izlušči poslane podatke o izmerjeni temperaturi v prostoru, indikaciji grela in poteku temperature in jih shrani v spremenljivke. Nato iz dveh tekstovnih datotek prebere nastavitve ogrevanja in urnik prostora. Skripta iz teh podatkov z uporabo IF ELSE pogojne strukture generira ustrezne vrednosti želene temperature, histereze in časa regulacije. Prejeti podatki iz modula in podatki o zelenih vrednostih se shranijo v tekstovni datoteki za statistične namene. Kot zadnje sestavi skripta paket z zahtevanimi vrednostmi in ga pošlje modulu.

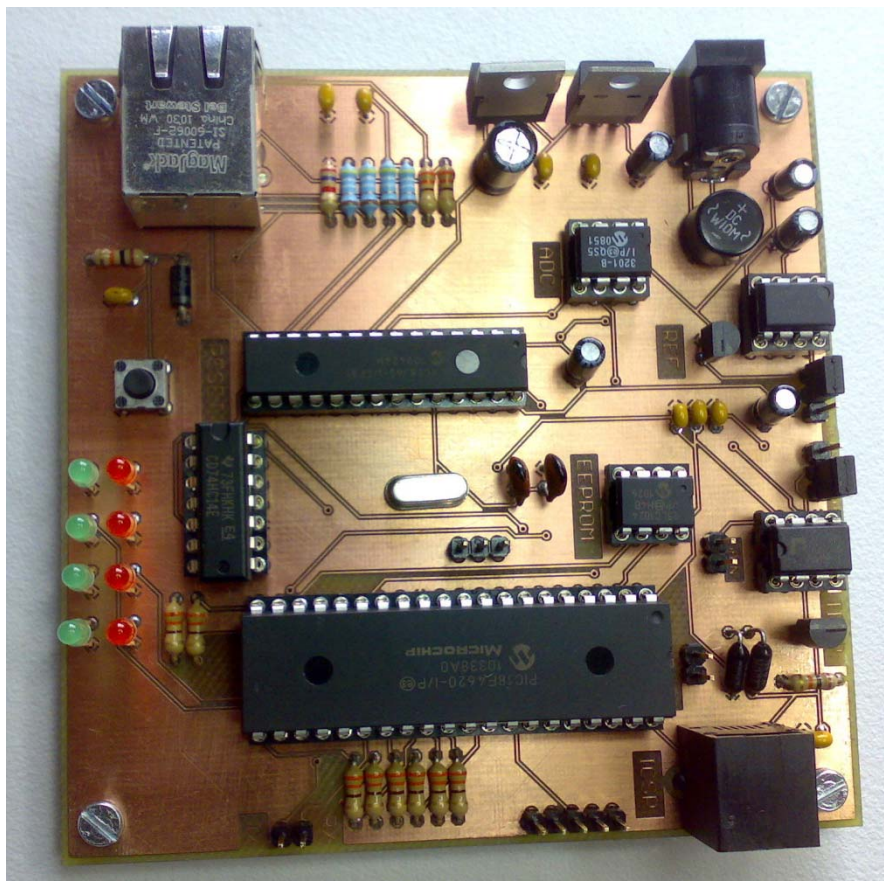
5.2.6 Delovanje modula v primeru napake

Lahko se zgodi, da modul zaradi določenih razlogov, kot so izpad elektrike na strani strežnika, napaka na omrežju, poškodovan kabel in drugih, ne more vzpostaviti povezave s strežnikom oziroma se le-ta prekine. Na strani modula je potrebno zagotoviti, da regulacija temperature v prostoru kljub napaki nemoteno deluje. Program na modulu v taki situaciji uporabi za regulacijo ali prednastavljene vrednosti, ki se uporabljajo samo v primeru napake, ali zadnje prejete vrednosti s strani strežnika. V aplikaciji uporabljamo zaradi varnosti prvo možnost, kjer so prednastavljene vrednosti nastavljene na

- želena temperatura: 21 °C,
- histereza: ± 1 °C in
- čas regulacije: 10 min.

6 Praktična izvedba aplikacije

Aplikacija je sestavljena iz strojnega dela, ki je implementiran na modulu in uporabniškega vmesnika, ki je implementiran na strežniku. Slika 38 prikazuje zgrajen modul. Delovanje le-tega je bilo predstavljeno v prejšnjih poglavjih, zato se bomo v tem poglavju omejili na možnosti, ki nam jih ponuja uporabniški vmesnik.

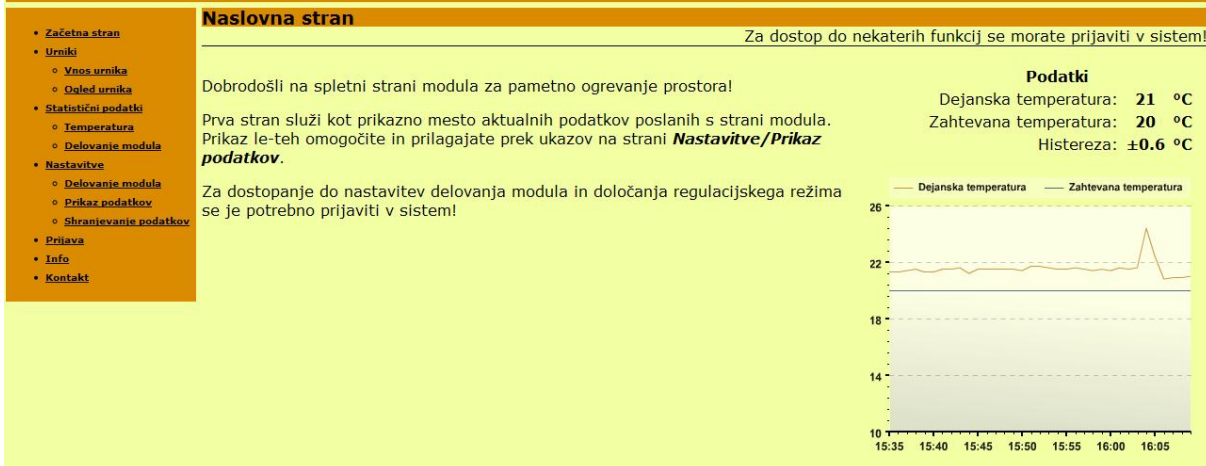


Slika 38 – Zgrajen modul

Pod besedno zvezo uporabniški vmesnik spada spletna stran, ki teče na strani strežnika. Ta je zgrajena z uporabo skriptnega jezika PHP [37], jezika HTML in predlog CSS [38]. Osnovno spletno stran smo zgradili z jezikom HTML in ji nato določili izgled s predlogami CSS, medtem ko se PHP ukazi uporabljajo za kreiranje dinamičnih spletnih strani. Prva stran spletne strani je predstavljena na Sliki 39.

Room Statistics & Regulation

Vse dostopne informacije o vašem prostoru zbrane na enem mestu



Slika 39 – Prva stran spletne strani

Sestavljena je iz treh delov, in sicer naslovnega predela na vrhu, vertikalnega menija na levi strani in glavnega prikaznega dela na sredini. Prva dva sta statično določena, medtem ko se na glavnem delu prikazuje vsebina izbrane strani.

Začetna stran ne služi samo uvodu, ampak se poleg tega uporablja za prikaz aktualnih podatkov, ki so bili poslani s strani modula. Uporabnik lahko omogoči prikaz podatkov na strani *Nastavitve/Prikaz podatkov*.

Uporabnik ima na strani *Urniki* možnost ogleda urnika prostora (*Urniki/Ogled urnika*). S prijavo v sistem nam je omogočen vnos urnika in pripadajočih nastavitve ogrevanja (*Urniki/Vnos urnika*), med katerimi so:

- dnevna temperatura,
- nočna temperatura,
- histereza in
- časovno obdobje dnevne temperature.

Stran *Statistični podatki* nam omogoča vpogled v shranjene podatke, in sicer temperaturo (*Statistični podatki/Temperatura*) ter delovanje modula (*Statistični podatki/Delovanje modula*). Prva nam ponuja izbiro izgleda prikazanega grafa shranjenih podatkov o temperaturi. Izbiramo lahko med tremi vrstami grafov in določamo časovno os. Shranjene podatke je možno s klikom na označeno povezavo prenesti na svoj računalnik v obliki tekstovne datoteke. Druga nam omogoča

preverjanje delovanja modula in grela. Opazujemo lahko zgodovino stanja povezave modula na stražnik, trend temperature v prostoru (naraščanje/padanje) ter stanje grela (vklopljeno/izklopljeno).

Uporabnik preko strani *Nastavitve* nastavlja delovanje modula (*Nastavitve/Delovanje modula*), shranjevanje podatkov (*Nastavitve/Shranjevanje podatkov*) in prikaz podatkov (*Nastavitve/Prikaz podatkov*). Za dostop do prvih dveh se je potrebno prijaviti v sistem. Prva možnost nam preko izbirnega menija omogoča nastavitve časa izvajanja regulacije na modulu. Uporabnik preko nastavitve na strani *Prikaz podatkov*, ki je prikazana na Sliki 40, vpliva na izgled naslovne strani. Iz izbirnega menija izberemo čas osveževanja podatkov, omogočimo prikaz grafa ter pri tem izberemo, katere podatke želimo spremljati in v kakšni obliki. Poleg omenjenega imamo možnost izbire prikaza tabele z aktualnimi podatki iz modula.

Na strani *Info* najde uporabnik nekaj podatkov o avtorju spletne strani, medtem ko stran *Kontakt* vsebuje e-poštni naslov, na katerega je možno poslati vprašanja ali pripombe glede delovanja spletne strani in aplikacije.

Room Statistics & Regulation

Vse dostopne informacije o vašem prostoru zbrane na enem mestu

Prikaz podatkov Za dostop do nekaterih funkcij se morate prijaviti v sistem!

• Začetna stran
• Urniki
 ◦ Vnos urnika
 ◦ Ogled urnika
• Statistični podatki
 ◦ Temperatura
 ◦ Delovanje modula
• **Nastavitve**
 ◦ Delovanje modula
 ◦ Prikaz podatkov
 ◦ Shranjevanje podatkov
• Prijava
• Info
• Kontakt

Nastavitev prikaza

Hitrost osveževanja podatkov: 5min

Nastavitve grafa

Prikaži graf: ☒

Prikaži želeno temperaturo: ☒

Prikaži dejansko temperaturo: ☒

Prikaži histerezo: ☐

Tip grafa: Linjski

Izpis podatkov

Prikaži želeno temperaturo: ☒

Prikaži dejansko temperaturo: ☒

Prikaži histerezo: ☒

Vnesi spremembe

Slika 40 – Nastavitve prikaza podatkov na naslovni strani

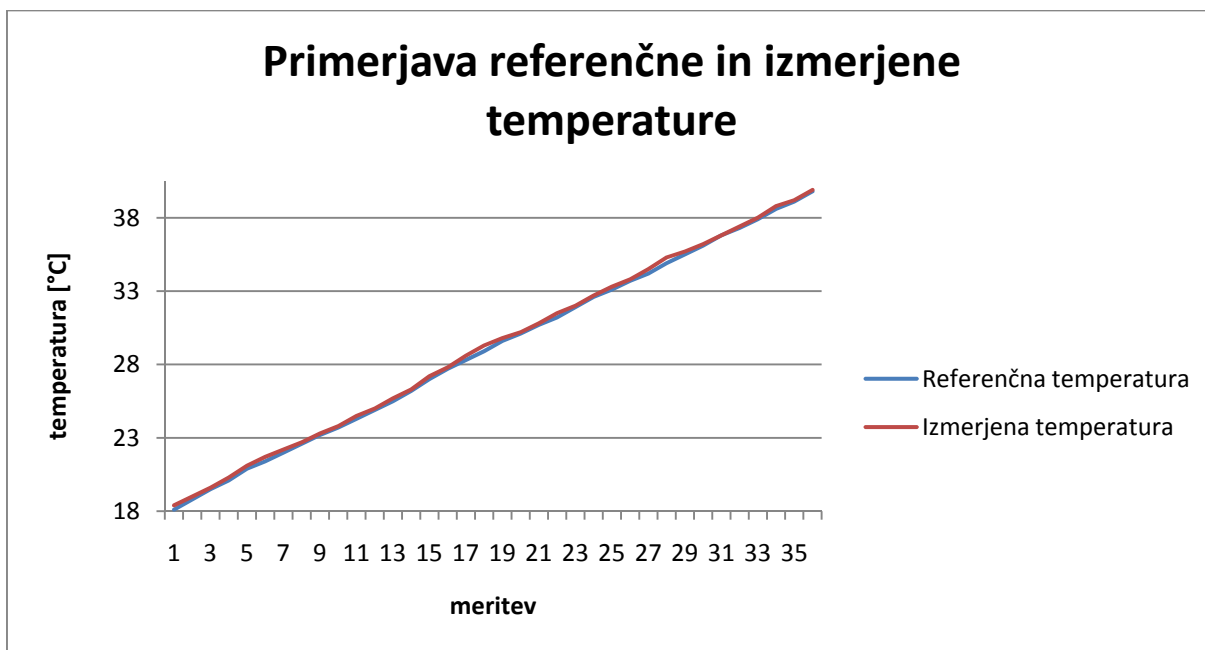
7 Verifikacija rezultatov

Za preverjanje izračunane teoretične vrednosti odstopanja izmerjene temperature smo opravili vzporedne meritve z referenčnim termometrom proizvajalca Burster, ki ima predpisano natančnost $\pm 0,1$ °C. Meritve so bile opravljene v zaprtem prostoru, kjer smo opazovali naraščanje temperature od 18,1 °C do 39,8 °C. Rezultati meritev so predstavljeni v Tabeli 4.

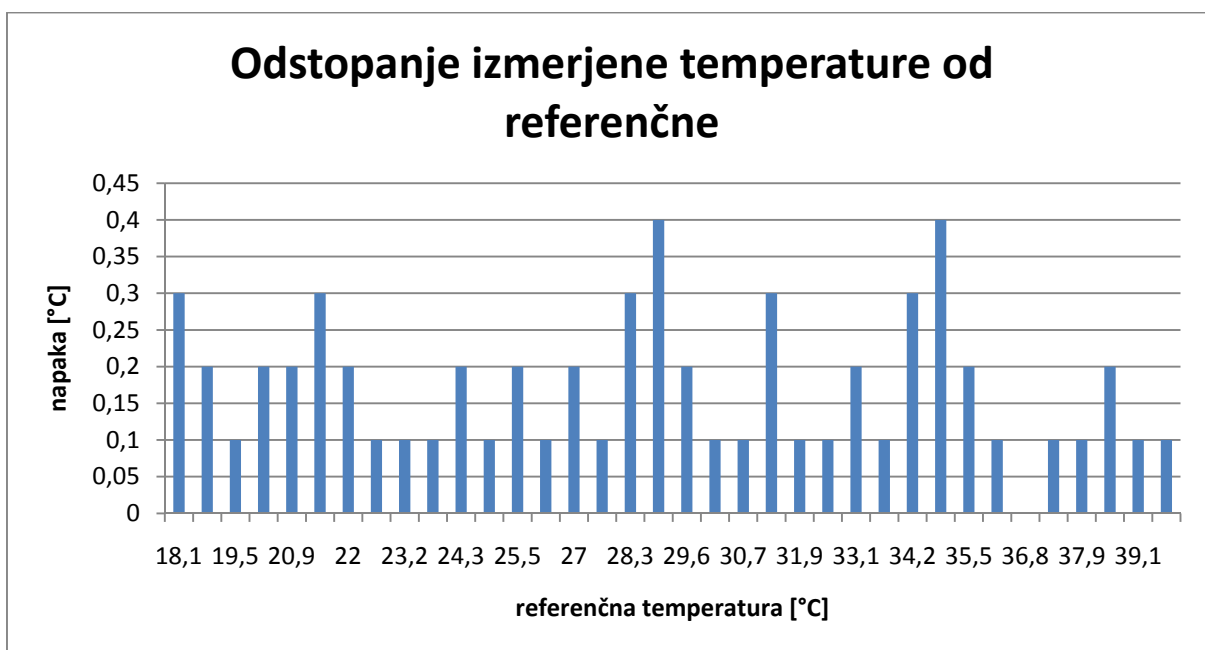
Tabela 4 – Rezultati primerjalnih meritev

T_{REF} [°C]	T_{MODUL} [°C]	ΔT_{MODUL} [°C]	T_{REF} [°C]	T_{MODUL} [°C]	ΔT_{MODUL} [°C]
18,1	18,4	0,3	29,6	29,8	0,2
18,8	19,0	0,2	30,1	30,2	0,1
19,5	19,6	0,1	30,7	30,8	0,1
20,1	20,3	0,2	31,2	31,5	0,3
20,9	21,1	0,2	31,9	32,0	0,1
21,4	21,7	0,3	32,6	32,7	0,1
22,0	22,2	0,2	33,1	33,3	0,2
22,6	22,7	0,1	33,7	33,8	0,1
23,2	23,3	0,1	34,2	34,5	0,3
23,7	23,8	0,1	34,9	35,3	0,4
24,3	24,5	0,2	35,5	35,7	0,2
24,9	25,0	0,1	36,1	36,2	0,1
25,5	25,7	0,2	36,8	36,8	0,0
26,2	26,3	0,1	37,3	37,4	0,1
27,0	27,2	0,2	37,9	38,0	0,1
27,7	27,8	0,1	38,6	38,8	0,2
28,3	28,6	0,3	39,1	39,2	0,1
28,9	29,3	0,4	39,8	39,9	0,1

Graf 1 prikazuje primerjavo referenčne temperature in temperature, izmerjene z modulom. Graf 2 prikazuje odstopanje izmerjene temperature od referenčne.



Graf 1 – Primerjava referenčne in izmerjene temperature



Graf 2 – Odstopanje izmerjene temperature od referenčne

Odstopanje izmerjene temperature od referenčne v celotnem merilnem območju ni večje od 0,4 °C, kar potrjuje izračunano maksimalno napako iz 5.1.3. V tej napaki je potrebno upoštevati začetno natančnost senzorja LM35DZ, ki znaša $\pm 0,5$ °C pri 25 °C.

8 Zaključek

Cilj te diplomske naloge, izdelati aplikacijo pametnega ogrevanja, je bil v celoti izpolnjen in nadgrajen z razširitvijo uporabe modula za razvojne namene. Pravilnost delovanja smo preverili z uporabo referenčnega termometra, ki je potrdil natančnost zajemanja temperature prostora preko modula. Natančnost bi se opazno izboljšala z uporabo kvalitetnejšega, a hkrati tudi dražjega senzorja za zajem temperature, kar za izbrano aplikacijo ni smiselno.

Za reguliranje sobne temperature je uporabljena dvostopenjska regulacija, ki je primerna za izbrano aplikacijo, saj je dinamika spreminjanja temperature zadosti počasna. Uporabnik ima možnost nastaviti vse parametre regulacije preko spletne strani, ki služi tako prikazu kot tudi vnosu podatkov. V aplikacijo je implementiran tudi varnostni mehanizem. Ta ob izgubi komunikacije med modulom in strežnikom zagotavlja regulacijskemu algoritmu prednastavljene parametre. Čeprav napaka v delovanju ne ogroža človeških življenj, varnostni mehanizem preprečuje brezpomensko ogrevanje prostora.

Nadgradnja v razvojni modul pomeni, da programer oziroma uporabnik začne razvoj svoje aplikacije na preverjenem okolju, kar zmanjša napake pri izdelavi in skrajša proces razhroščevanja. Naslednja pozitivna lastnost je modularnost, saj je modul zasnovan tako, da je možno večino neizkoriščenih priključnih sponk mikrokrmilnika uporabiti za razširitev delovanja. Modul poleg komunikacije preko Ethernet omrežja omogoča serijsko komuniciranje (EUSART) preko priključnih sponk RX in TX. To pomeni, da je možno modul z ustreznim vmesnim vezjem priključiti direktno na osebni računalnik ali na katero drugo napravo.

Po izdelavi modula in njegovem testiranju smo ugotovili, da ga je potrebno za pravilno delovanje ozemljiti. Brez ozemljitve je vezje na nedoločenem električnem potencialu, ki se zaradi parazitnih vplivov spreminja, kar v najhujšem primeru pomeni nedelovanje modula.

Literatura

RAČUNALNIŠKA OMREŽJA

1. Douglas E. Comer, Internetworking with TCP/IP Vol 1: Principles, Protocols, and Architecture, Fourth Edition, Prentice Hall, 2000
2. Vodilo, http://colos1.fri.uni-lj.si/ERI/RAC_SISTEMI_OMREZJA/html/...racunalniska_omrezaja/topologija_vodilo.gif, dostopnost preverjena junija 2011
3. CSMA/CD, http://gradiva.txt.si/m/racunalnistvo/24_pristopne/img/csmacd.gif, dostopnost preverjena junija 2011
4. W. Richard Stevens, TCP/IP Illustrated, Vol 1: The Protocols, Addison Wesley, Boston 1994
5. M. Simmons, Ethernet Theory of Operation, Microchip Technology Inc., 2008, <http://ww1.microchip.com/downloads/en/AppNotes/01120a.pdf>, dostopnost preverjena junija 2011
6. Encapsulation (networking), [http://en.wikipedia.org/wiki/...Encapsulation_\(networking\)](http://en.wikipedia.org/wiki/...Encapsulation_(networking)), dostopnost preverjena junija 2011
7. IP Address, http://en.wikipedia.org/wiki/IP_address, dostopnost preverjena junija 2011
8. Subnetwork, <http://en.wikipedia.org/wiki/Subnetwork>, dostopnost preverjena junija 2011
9. Prevedba lokalnega naslova IP, http://sers.s-sers.mb.edus.si/gradiva/rac/...moduli/upravljanje_ik/62_arp/datoteke/arp1.gif, http://sers.s-sers.mb.edus.si/gradiva/rac/moduli/upravljanje_ik/62_arp/datoteke/arp2.gif, http://sers.s-sers.mb.edus.si/gradiva/rac/moduli/upravljanje_ik/62_arp/...datoteke/arp3.gif, http://sers.s-sers.mb.edus.si/gradiva/rac/moduli/...upravljanje_ik/62_arp/datoteke/arp4.gif, dostopnost preverjena junija 2011
10. User Datagram Protocol, http://sers.s-sers.mb.edus.si/gradiva/rac/moduli/...upravljanje_ik/64_udp/datoteke/datagram.gif, dostopnost preverjena junija 2011
11. Vrata in vtičnice, http://www.s-sers.mb.edus.si/gradiva/w3/omrezja/...40_transportni/tcp/vrata.png, dostopnost preverjena junija 2011

12. Transportni sloj, http://www.s-sers.mb.edus.si/gradiva/w3/omrezja/...40_transportni/tcp/kanal.gif, dostopnost preverjena junija 2011
13. Transportni sloj, http://sers.s-sers.mb.edus.si/gradiva/rac/moduli/...upravljanje_ik/66_tcp/datoteke/tcp_vzpostavitev.gif, dostopnost preverjena junija 2011
14. Kontrola pretoka, http://sers.s-sers.mb.edus.si/gradiva/rac/moduli/...upravljanje_ik/66_tcp/datoteke/okno2.gif, dostopnost preverjena junija 2011
15. Everything you need to know about DHCP as a System Administrator, http://blogs.microsoft.co.il/blogs/netanelb/WindowsLiveWriter/...EverythingyouneedtoknowaboutDHCPasaSyste_10CE4/DHCP_DORA.jpg, dostopnost preverjena junija 2011

DELOVANJE IN UPORABA MICROCHIP TCP/IP SKLADA

16. The Microchip TCP/IP Stack, <http://ww1.microchip.com/downloads/en/...AppNotes/00833c.pdf>, dostopnost preverjena junija 2011
17. Microchip TCP/IP Stack v5.25 Help file, Microchip Technology, Inc., 2010
18. Dan Gookin, C for Dummies, 2nd Edition, Wiley Publishing, Inc., 2004
19. Dogan Ibrahim, Advanced PIC Microcontroller Projects in C, Elsevier Ltd., 2008

ZGRADBA IN DELOVANJE MODULA

20. Rectifier, <http://en.wikipedia.org/wiki/Rectifier>, dostopnost preverjena junija 2011
21. Diode Bridge Rectifier, <http://images.blog-u.net/wp-content/uploads/2010/09/...Diodes-bridge-Rectifier.jpg>, dostopnost preverjena junija 2011
22. Graetz, <http://www.sn.schule.de/~gyfloeha/rt/lex03/graetz05.gif>, dostopnost preverjena junija 2011
23. (avtor ni podan) Datasheet: ENC28J60, Stand-Alone Ethernet Controller with SPI Interface, Microchip Technology, Inc., 2008, <http://ww1.microchip.com/...downloads/en/DeviceDoc/39662c.pdf>, dostopnost preverjena junija 2011
24. (avtor ni podan) Datasheet: PIC18F4620, 28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nano Watt Technology, Microchip Technology, Inc., 2008, <http://ww1.microchip.com/downloads/en/...DeviceDoc/39626e.pdf>, dostopnost preverjena junija 2011

25. MPLAB ICD2, http://www.microchip.com/stellent/groups/devtools_sg/...documents/devtools/%7Eexport/en010046%7E26%7Een006096%...7EPRESSReleaseLayout/118902-3.png, dostopnost preverjena junija 2011
26. (avtor ni podan) Datasheet: 25LC1024, 1Mbit SPI Bus Serial EEPROM, Microchip Technology, Inc., 2010, <http://ww1.microchip.com/downloads/...en/DeviceDoc/22064D.pdf>, dostopnost preverjena junija 2011

MERJENJE IN REGULACIJA TEMPERATURE

27. J. Drnovšek, J. Bojkovski, Merilni sistemi 1 in 2, študijska skripta, Ljubljana 2005
28. (avtor ni podan) Datasheet: LM35, Precision Centigrade Temperature Sensors, National Semiconductor, 2000, <http://www.national.com/ds/...LM/LM35.pdf>, dostopnost preverjena junija 2011
29. Charles Kitchin, Low Counts, A Designer's Guide to Instrumentation Amplifiers, Analog Devices, Inc., 2006
30. Peter Šuhel, Operacijski ojačevalnik v sistemih, Mengeš 1995
31. (avtor ni podan) Datasheet: AD621, Low Drift, Low Power Instrumentation Amplifier, Analog Devices, Inc., 2001, http://www.analog.com/static/...imported-files/data_sheets/AD621.pdf, dostopnost preverjena junija 2011
32. (avtor ni podan) Datasheet: ICL7660, ICL7760A, CMOS Voltage Converter, Intersil, 2005, <http://datasheets.maxim-ic.com/en/ds/ICL7660-MAX1044.pdf>, dostopnost preverjena junija 2011
33. (avtor ni podan) Datasheet: MCP3201, 2,7 V 12-Bit AD Converter with SPI Serial Interface, Microchip Technology, Inc., 2007, <http://ww1.microchip.com/...downloads/en/DeviceDoc/21290e.pdf>, dostopnost preverjena junija 2011
34. (avtor ni podan) Datasheet: MCP1525/1541, 2,5 V and 4,096 V Voltage References, Microchip Technology, Inc., 2005, <http://ww1.microchip.com/...downloads/en/devicedoc/21653b.pdf>, dostopnost preverjena junija 2011
35. Franc Bergelj, Meritve 1. del, Fakulteta za elektrotehniko, Ljubljana, 2002
36. Borut Zupančič, Zvezni regulacijski sistemi, 1. del, Fakulteta za elektrotehniko in računalništvo, Ljubljana, 2010

PREDSTAVITEV APLIKACIJE

37. Janet Valada, PHP & MySQL For Dummies, 3rd Edition, Wiley Publishing, Inc., 2007
38. Ian Lloyd, Build Your Own Web Site the Right Way Using HTML & CSS, 2nd Edition, SitePoint Pty. Ltd., 2008

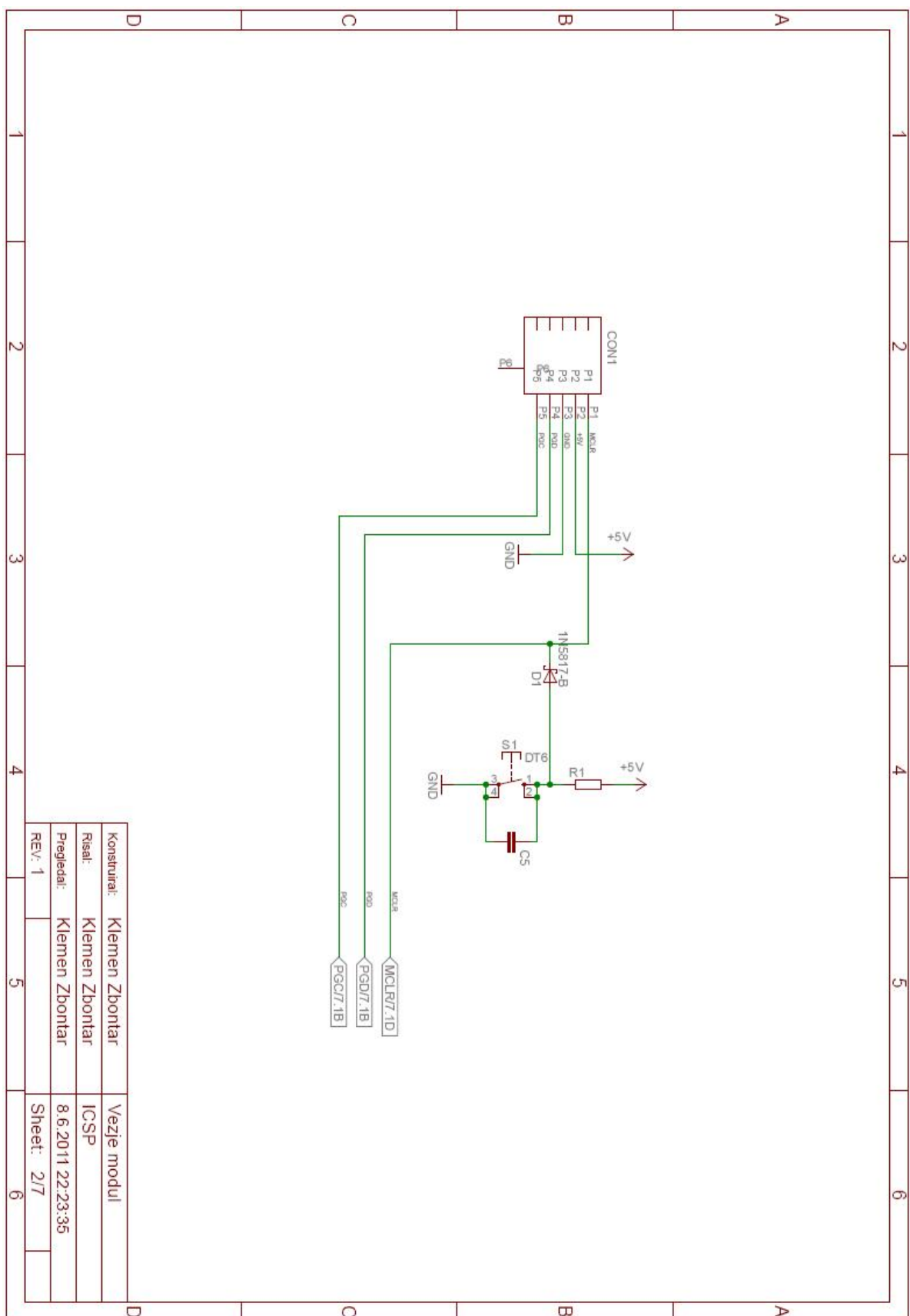
Izjava

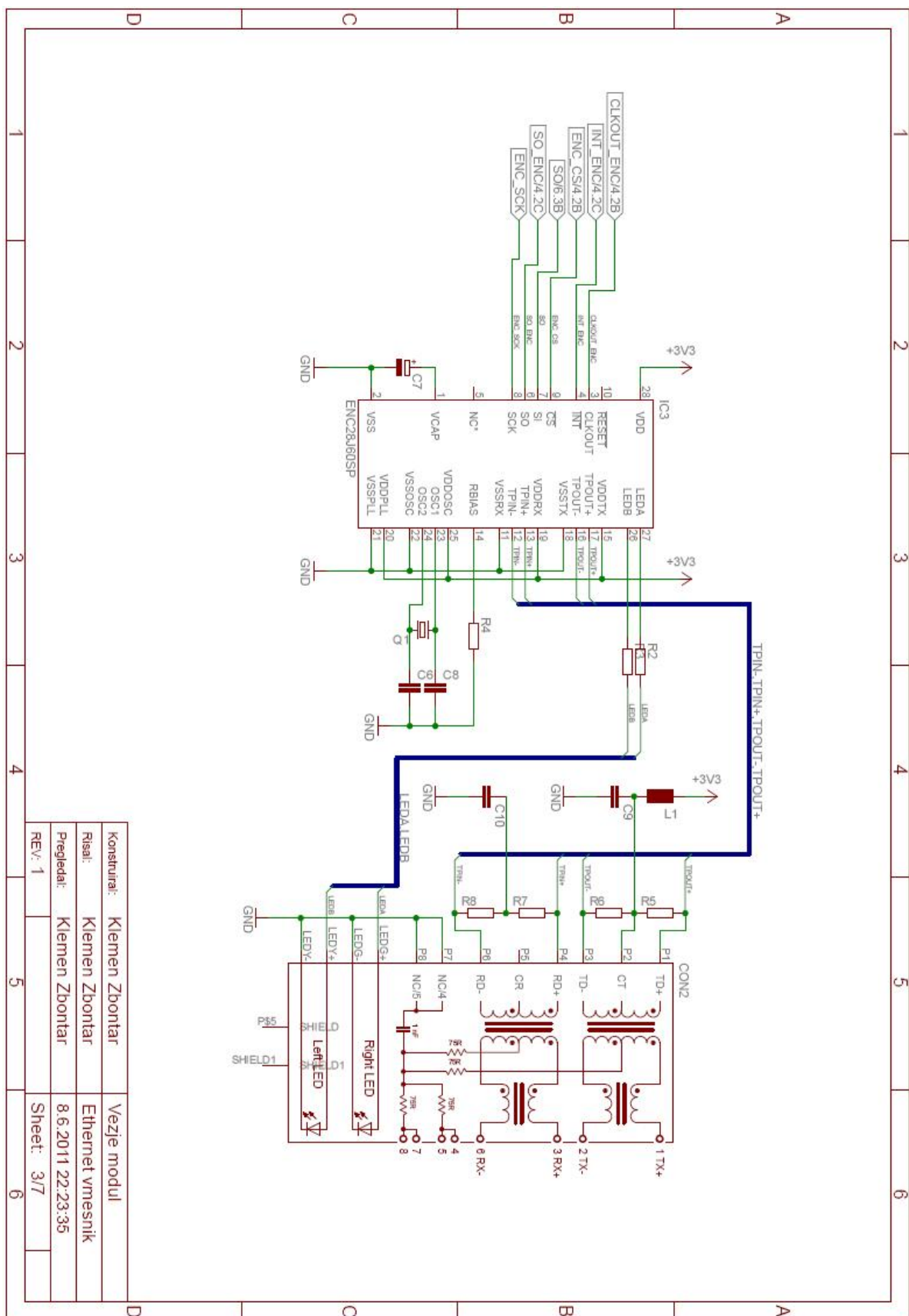
Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Boštjana Murovca. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

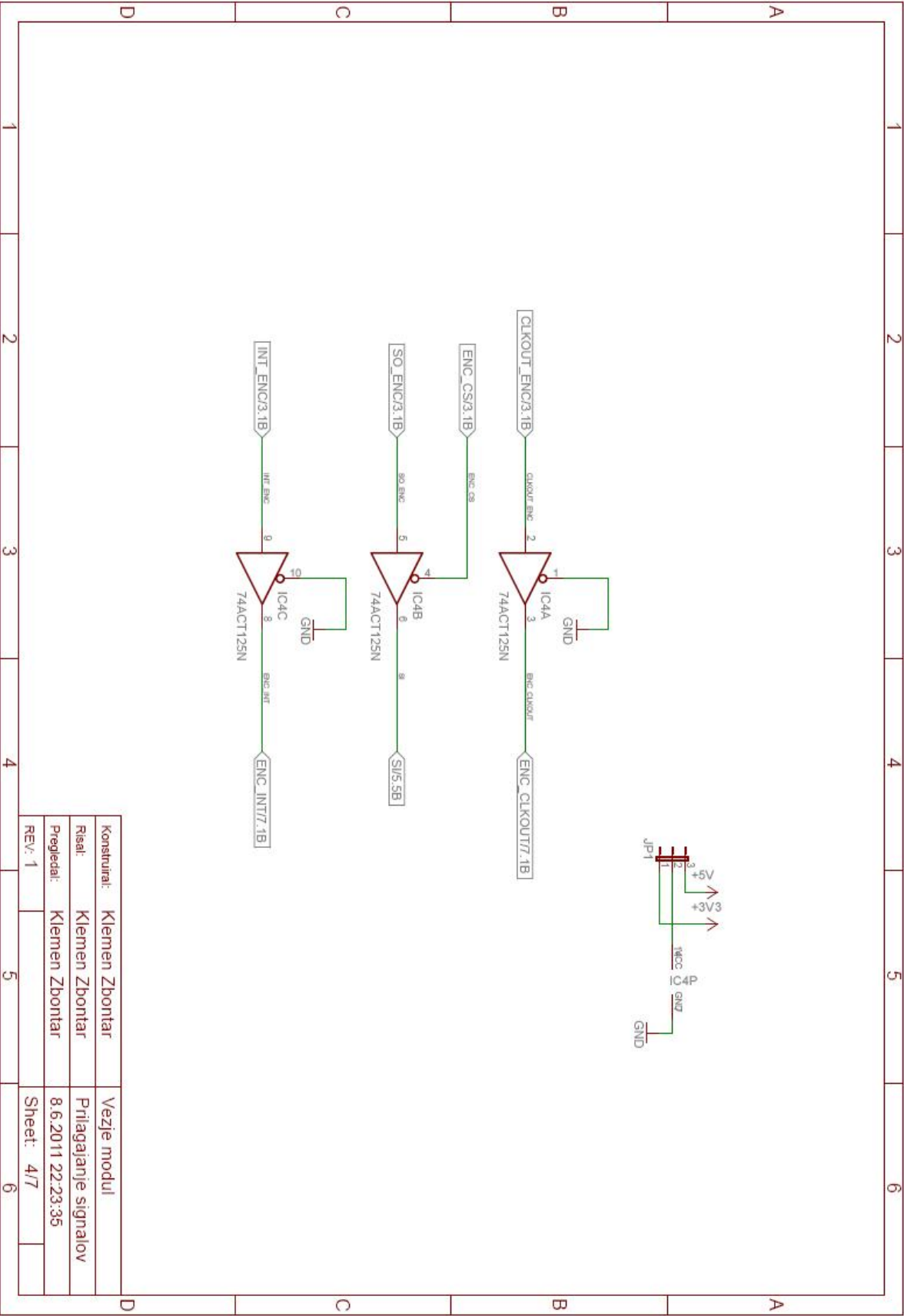
Jesenice, 11. junij 2011

Klemen Žbontar

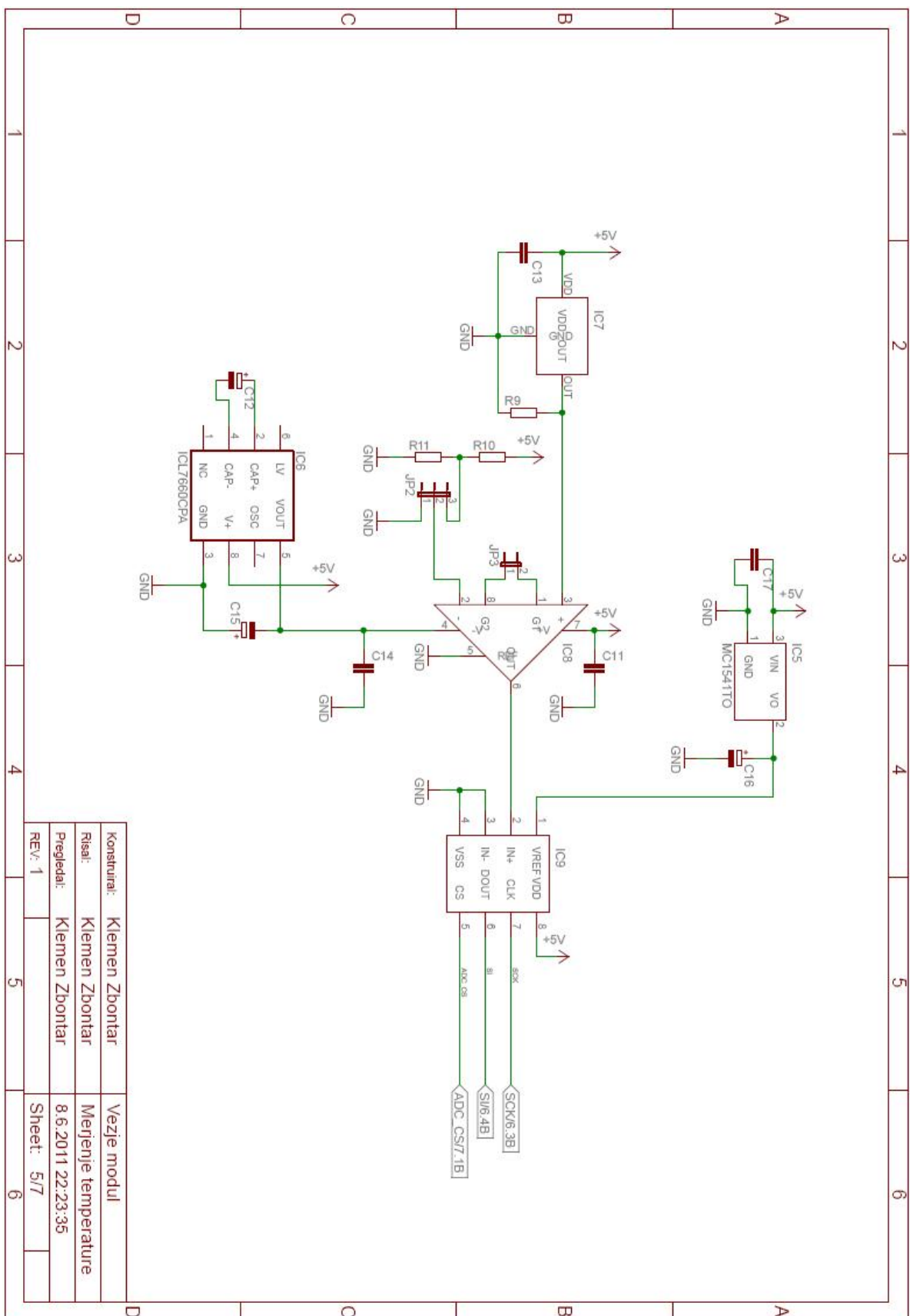
Konstruiral:	Kiemen Zbontar	Vezje modul
Risat:	Kiemen Zbontar	Napajanje +3,3V in +5V
Pregledat:	Kiemen Zbontar	8.6.2011 22.23.35
REV: 1		Sheet: 1/7

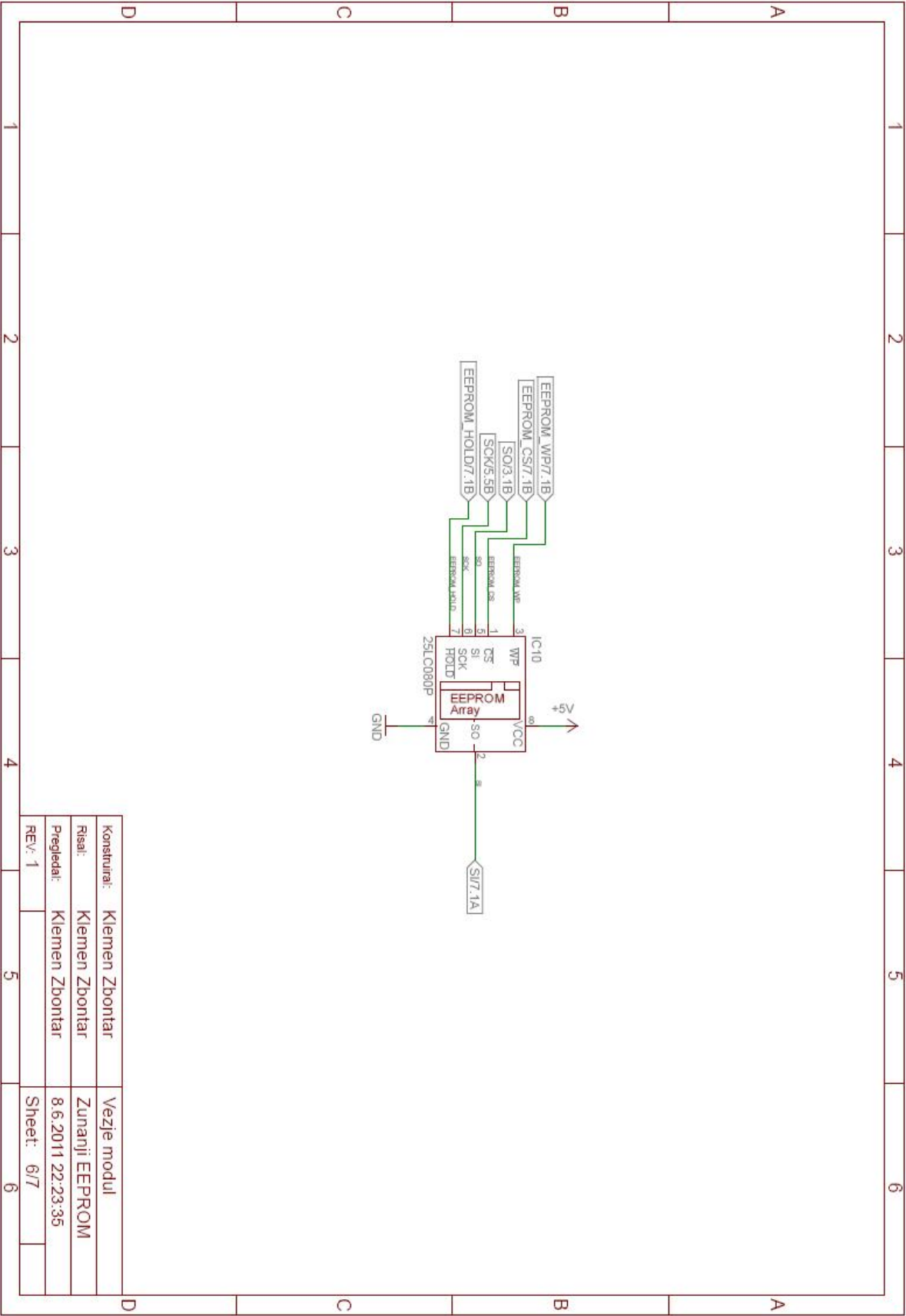


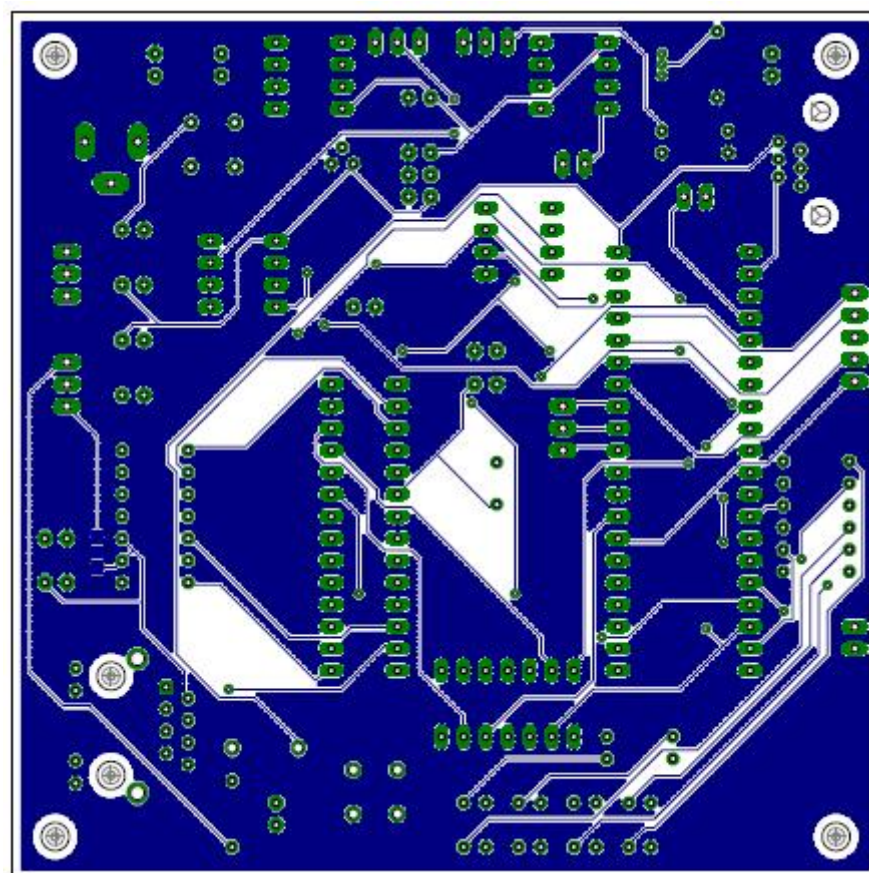
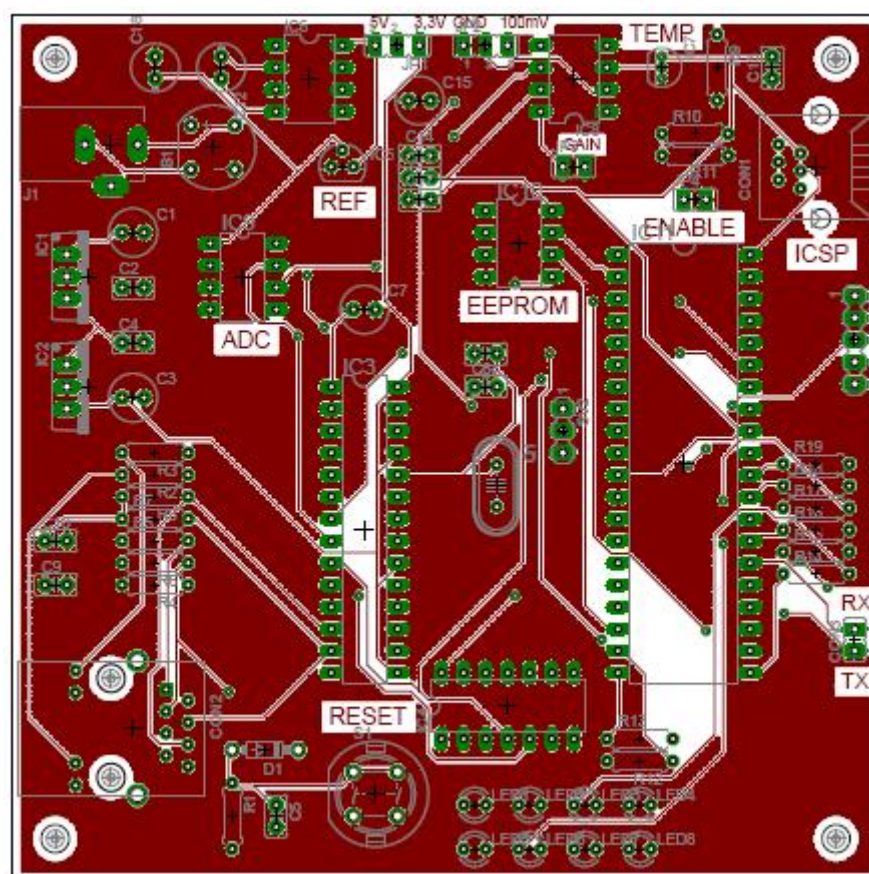


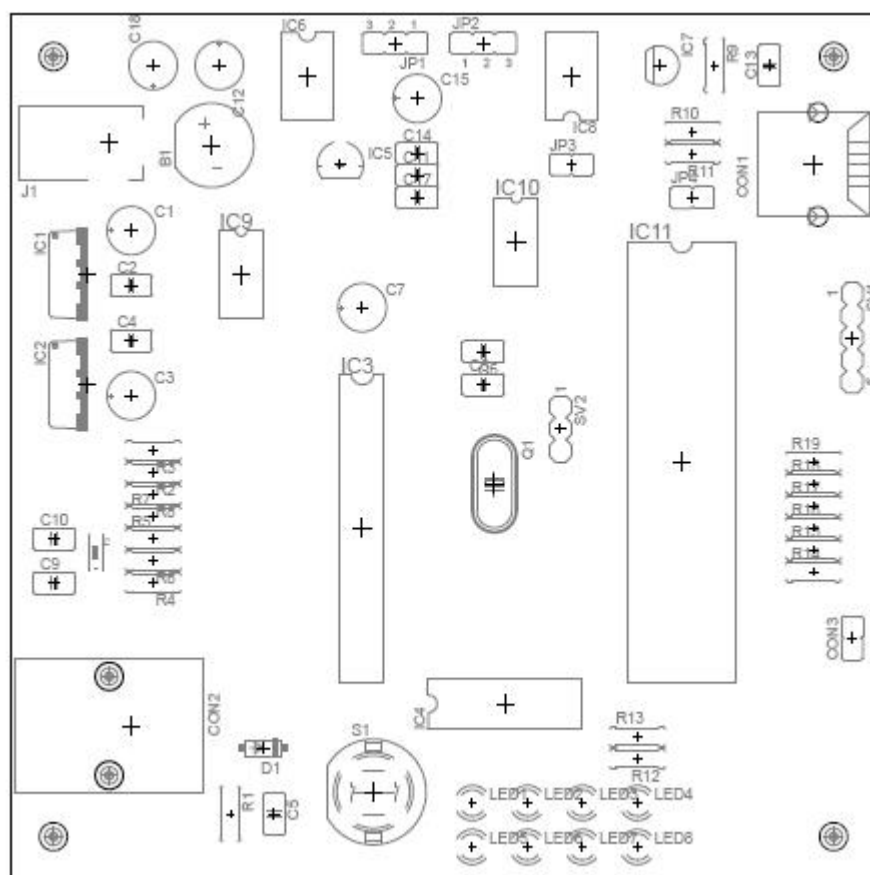


Konstruiral:	Klemen Zbontar	Vezje modul
Risai:	Klemen Zbontar	Prilagajanje signalov
Pregledal:	Klemen Zbontar	8.6.2011 22:23:35
REV: 1		Sheet: 4/7









B Priloga – spisek uporabljenih elementov

Seznam uporabljenih elementov za izgradnjo modula:

Tabela 5 – Spisek uporabljeni elementov

Oznaka v shemi	Proizvajalec	Vrednost	Količina
J1		Jack, 2,1 mm	1
IC1	Fairchild	LM7805	1
IC2	National Semiconductor	LM3940	1
IC3	Microchip	ENC28J60	1
IC4	National Semiconductor	74ACT125N	1
IC5	Microchip	MCP1541	1
IC6	Intersil	ICL7660	1
IC7	National Semiconductor	LM35DZ	1
IC8	Analog Devices	AD621	1
IC9	Microchip	MCP3201	1
IC10	Microchip	25LC1024	1
IC11	Microchip	PIC18F4620	1
C1, C7, C12, C15, C16		10 μ F/25 V	5
C2, C4, C5, C9, C10, C11, C13, C14, C17		100 nF	9
C3		100 μ F/25 V	1
C6, C8			2
R2, R3, R12, R13, R14, R15, R16, R17, R18, R19		330 Ω	10
R1		10 k Ω	1
R9		100 k Ω	1
R5, R6, R7, R8		49,9 k Ω , 1 %	4
R4		2,32 k Ω , 1 %	1
R10		100 k Ω , 0,1 %	1
R11		2,05 k Ω , 0,1 %	1
B1		1 A, 600 V	1
D1		1N5817	1
L1		Feritno jedro	1

Q1		25 MHz	1
S1		Tipka nizka	1
CON1		RJ11	1
CON2		RJ45	1
CON3		Letvica M 2P	1
SV1		Letvica M 5P	1
SV2		Letvica M 3P	1
JP1, JP2		Jumper 3P	2
JP3, JP4		Jumper 2P	2
LED1, LED2, LED3, LED4		LED rdeča, 3 mm	4
LED5, LED6, LED7, LED8		LED zelena, 3 mm	4

Če proizvajalec ni naveden, so lahko elementi poljubnega proizvajalca, če izpolnjujejo vse zahteve. Vsi upori, ki nimajo predpisanih toleranc so 5 %.