

UNIVERZA V LJUBLJANI
Fakulteta za elektrotehniko

Peter Vitez

REGULACIJA SOBNE TEMPERATURE Z INTERNETNIM
DOSTOPOM

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: doc. dr. Boštjan Murovec

Ljubljana, junij 2007

Zahvala

Na prvem mestu se bi rad zahvalil doc. dr. Boštjanu Murovcu za vso pomoč, nasvete, pregled tega diplomskega dela in moralno podporo. Zahvaljujem se tudi vsem sodelavcem v podjetju Iskra Avtoelektrika d.d., ki so mi na kakršen koli način pomagali pri izgradnji modula in programske opreme.

Kazalo vsebine:

Kazalo vsebine:	I
Kazalo slik:	II
Kazalo tabel:	III
Kazalo grafov:	III
Povzetek	V
Abstract	VII
1 UVOD	1
1.1 Kratek zgodovinski pregled	1
2 TCP/IP ARHITEKTURNI MODEL	5
2.1 Povezovanje med omrežji	5
2.2 Referenčni model ISO OSI	6
2.3 Sloji TCP/IP protokola	9
2.3.1 Enkapsulacija podatkov	12
2.4 IP Naslavljanje	13
2.4.1 Naslovni razred A	14
2.4.2 Naslovni razred B	15
2.4.3 Naslovni razred C	15
2.4.4 Naslovni razred za pošiljanje več sistemom (D)	15
2.4.5 Eksperimentalni naslovni razred (E)	16
2.5 Protokol za pretvorbo naslovov (ARP)	16
2.5.1 Protokol za obratno pretvorbo naslovov (RARP)	19
2.6 Internetni protokol za krmilna sporočila (ICMP)	19
2.7 Podomreževanje	20
2.7.1 IP maska	21
2.7.2 Biti podomrežne maske	22
2.8 Transportni sloj TCP	23
2.8.1 TCP	23
2.8.2 Vrata in vtičnice	24
2.8.3 Vzpostavljanje povezave	25
2.8.4 Mehanizmi potrjevanja	27
2.8.5 Kontrola pretoka z drsečim oknom	28
2.9 Protokol za prenos datotek FTP	29
3 MICROCHIP TCP/IP SKLAD	31
3.1 Sloji sklada	31
3.2 Konfiguracija sklada	32
3.3 Uporaba sklada	33
3.4 Microchipov HTTP Strežnik	34
3.4.1 Generiranje dinamičnih HTTP strani	35
3.4.2 HTTPExecCmd	36
3.5 Prevajalnik MPFS	37
3.6 Microchipov FTP strežnik	38
4 MERJENJE TEMPERATURE	41
4.1 Merjenje temperature s senzorjem AD592CN	41
4.1.1 Postopek merjenja	42
4.2 Merjenje Temperature z uporabo termistorja	48
4.2.1 Postopek merjenja	51

4.2.2	<i>Napaka merilnega rezultata</i>	56
5	REGULACIJA TEMPERATURE	59
5.1	Dvostopenjska regulacija	59
5.2	Programska izvedba dvopoložajne regulacije	66
5.2.1	Delovanje v primeru napake	69
6	MODUL	71
6.1	SPI komunikacija	72
6.2	Močnostni del	74
6.3	Grafični vmestnik	76
7	VERIFIKACIJA REZULTATOV	81
8	ZAKLJUČEK	83
9	LITERATURA	85
	Izjava:	89
	Priloga:	91

Kazalo slik:

Slika 2.1:	ISO OSI referenčni model	7
Slika 2.2:	Nivoji TCP/IP protokola	10
Slika 2.3:	Enkapsulacija podatkov	12
Slika 2.4:	Prikaz naslovnih razredov	14
Slika 2.5:	Pregled tabele ARP	17
Slika 2.6:	Razpršeno pošiljanje paketa	18
Slika 2.7:	Primerjanje IP naslova	18
Slika 2.8:	Pošiljanje strojnega naslova povpraševalcu	19
Slika 2.9:	Podomreževanje	21
Slika 2.10:	TCP paket	24
Slika 2.11:	Delitev vrat	24
Slika 2.12 :	Vzpostavitev povezave s TCP	26
Slika 2.13:	Kontrola pretoka z drsečim oknom	28
Slika 2.14:	FTP protokol	29
Slika 2.15:	Prikaz komunikacije s FTP strežnikom	30
Slika 3.1:	Microchipov TCP/IP sklad	31
Slika 3.2:	Prenos datotek na strežnik	38
Slika 4.1:	Izhodna karakteristika senzorja	42
Slika 4.2:	Osnovna vezava senzorja (levo) in kalibracija pri določeni temperaturi (desno) ...	42
Slika 4.3:	Merjenje temperature s temperaturnim senzorjem AD592	44
Slika 4.5:	Vezava termistorja v napetostni delilnik	52
Slika 5.1:	Prehod iz P regulatorja v dvopoložajni ali ON/OFF regulator	59
Slika 5.2:	Statična karakteristika dvopoložajnega regulatorja	60
Slika 5.3:	Potek temperature	61
Slika 5.4:	Bločna shema regulacijskega sistema z dvopoložajnim regulatorjem	62
Slika 5.5:	Potek regulirane in regulirne veličine, če ne upoštevamo mrtvega časa	63
Slika 5.6:	Potek regulirane in regulirne veličine, če ne upoštevamo histereze	64
Slika 5.7:	Potek regulirane in regulirne veličine, če upoštevamo histerezo in mrtvi čas	65
Slika 5.8:	Grafični prikaz regulacijskega algoritma	69
Slika 6.1:	Razpored priključnih sponk in slika mikrokrmilnika PIC18F4620	71
Slika 6.2:	Osnovna vezalna shema mikrokrmilnika in ethernet krmilnika v omrežje	72
Slika 6.3:	SPI komunikacija	73

Slika 6.4: Potek SPI komunikacije	74
Slika 6.5: Močnostni del	75
Slika 6.6: Internetni modul	76
Slika 6.7: Grafični vmestnik.....	77
Slika 6.8: Prenos datoteke na strežnik	78
Slika 6.9: Izgled pogovornega okna	79

Kazalo tabel:

Tabela 1: Parametri NTK upora	50
Tabela 2: Karakteristika NTK upora in merilnega člena.....	57
Tabela 3: Rezultati meritev	81

Kazalo grafov:

Graf 1: Karakteristika NTK upora.....	48
Graf 2: Odvisnost napetosti od temperature	53
Graf 3: Odvisnost temperature od napetosti	54
Graf 4: Napaka zaradi kvantizacije	55
Graf 5: Napake merilnega člena	58
Graf 6: Temperaturno odstopanje.....	82

Povzetek

V okviru diplomske naloge smo raziskali možnost regulacije sobne temperature na daljavo preko internetne povezave. Izdelali smo neodvisen internetni modul, ki ga vključimo v internetno omrežje in preko njega reguliramo temperaturo. Glavna sestavna dela modula sta Microchipov mikrokrmilnik PIC18F4620 in Microchipov mrežni krmilnik ENC28J60. Za pisanje programa smo uporabljali funkcije iz Microchipovega TCP/IP sklada.

Veliko pozornosti smo posvetili samemu merjenju temperature, zato smo izdelali dva termometra. Prvega smo realizirali s pomočjo termistorja, drugega pa z uporabo temperaturnega senzorja AD592. Verifikacija rezultatov z referenčnim termometrom je pokazala, da je za merjenje temperature primernejši temperaturni senzor AD592, saj je natančnejši. Regulacijski postopek smo izvedli s pomočjo dvostopenjske (ON-OFF) regulacije.

Upravljanje z modulom poteka preko uporabniškega vmesnika, s pomočjo katerega enostavno spreminjamo in nastavljamo določene parametre modula. Modul je zasnovan tako, da ga lahko z različnimi konfiguracijami uporabljamo za najrazličnejše aplikacije: najrazličnejše regulacije, zajemanje podatkov, nadzor, alarmiranje, spletno krmiljenje kamere,...

Ključne besede:

PIC18F4620, ENC28J60, TCP/IP sklad, termistor, AD592, dvostopenjska (ON - OFF) regulacija, modul.

Abstract

The thesis discusses regulation of indoor temperature. The end user communicates with the system by means of Internet connection. The outcome of the work is an independent module, which is suitable for regulation of temperature. The main components are Microchip's microcontroller PIC18F4620 and Microchip's Ethernet controller ENC28J60. The software part of internet subsystem is founded on Microchip's TCP/IP stack, which supports cooperative multitasking execution of various applications' threads.

A great deal of attention has been focused on the temperature measurements, for which two different temperature sensors are developed. The first one is made of thermistor whereas the second one is made of temperature sensor AD592 from Analog Devices. Verification of results with reference thermometer reveals that the sensor AD592 is more suitable for the task, due to its higher initial accuracy. The regulation of temperature is based on classical switching (ON – OFF) regulation.

A dedicated user interface is developed for setting and monitoring the parameters of module. The module is suitable for implementation of many different applications, like online regulation, data acquisition, control, alarming systems, WebCam control, to name a few.

Key words:

PIC18F4620, ENC28J60, TCP/IP stack, thermistor, AD592, switching (ON – OFF) regulation, module.

1 UVOD

Vse skupaj se je začelo pod okriljem ameriške vojske projekta ARPAnet. Dandanes je nepogrešljiv pripomoček skoraj vsakega posameznika. Razvija se čez vse meje in nam narekuje ritem življenja. Seveda, govorimo o Internetu.

Internet je od svojih začetkov do danes presegel vsa pričakovanja. Sčasoma je internetno omrežje preraslo v akademsko in raziskovalno omrežje, kasneje pa je postalo zanimivo tudi za komercialne uporabnike. V tem času je Internet podvojil svojo velikost približno vsakih trinajst mesecev, kar kaže na izredno hitro rast. S širjenjem omrežja se je povečalo tudi število različnih vrst aplikacij, ki tečejo na njem. Vedno več ljudi vsakodnevno uporablja storitve medmrežja pri svojem delu, veliko je podjetij, ki preko Interneta opravljajo del svojega poslovanja.

V zadnjem času se vse več govori o spletnem krmiljenju in upravljanju. Pojavile so se inteligentne hiše, ki uporabnikom omogočajo najrazličnejše vrste udobja. Seveda skoraj vsi algoritmi tečejo na zmogljivih računalnikih, tako da so aplikacije praktično brez omejitev. Seveda je naloga veliko težavnejša, če sistem poganja mikrokrmilnik z omejenimi sredstvi. Cilj, ki smo si ga zadali v diplomski nalogi, je izdelati internetni modul, s katerim bomo lahko oddaljeno regulirali temperaturo in vse skupaj spremljali na svetovnem spletu. Modul, ki smo ga izdelali, ni namenjen le regulaciji temperature, ampak ga je možno z ustreznimi prilagoditvami uporabljati tudi v druge namene. Da bomo lažje razumeli, kako poteka komunikacija med mikrokrmilnikom in oddaljenim računalnikom, si bomo za začetek ogledali ključne protokole, ki jih vsebuje internet. V samem nadaljevanju bomo opisali postopek konfiguracije modula ter postopek merjenja in regulacije temperature. V zaključku smo opisali postopek verifikacije in dobljene rezultate.

1.1 Kratek zgodovinski pregled

Internet je nastal iz razvojno-raziskovalnih potreb ameriškega industrijsko-vojaškega kompleksa. S povečevanjem priljubljenosti Interneta so poleg raziskovalno-akademiških ustanov dostop do njega začele ponujati komercialne organizacije, kot so telefonske družbe AT&T, MCI, British Telecom in podobne. Tako so se na Internet začeli povezovati tudi tisti, ki se ne ukvarjajo z raziskovalno dejavnostjo. Rezultat tega je, da je Internet postal omrežje z več kot 60.000 omrežji [13 str. 10].

Ena znanih hrbtenic (hrbtenice ali glavna oporna omrežja zagotavljajo visoko hitrost in velik obseg podatkovnih zvez med regionalnimi omrežji) je bila vsekakor NSFNET, ki jo je financiralo ameriško obrambno ministrstvo v okviru projekta ARPAnet.. Pri nas je najbolj znana hrbtenica, ki jo vzdržuje Center vlade za informatiko ARNES. Vloga ameriške zvezne vlade v razvoju in financiranju Interneta je ohranjala Internet nekomercialen do pred zadnjih nekaj let. Zaradi komercializacije se Internet razvija čedalje hitreje in prodira v tradicionalno zaprta okolja, kot so Kitajska, Vietnam in Kuba [13 str. 11].

Mejniki v zgodovini Interneta so sledeči:

- 1968-1973 začetek projekta ARPANET, ki ga financira ameriško obrambno ministrstvo,
- 1973-1981 raziskovalni projekt ARPANET teče na univerzah ZDA,
- 1981 nastanek Bitneta (Bitnet je akademsko omrežje, ki ga je financirala korporacija IBM; evropska različica, znana kot EARN, za povezovanje med računalniki uporablja protokole korporacije IBM),
- 1983 sprejeti standardni protokoli Interneta (TCP/IP) za povezovanje in komunikacijo med računalniki in začetek delovanja Interneta,
- 1984 vzpostavitev sistema imenskega prostora – DNS (Domain Name System), ki je osnova za preprosto delovanje vseh storitev Interneta; ameriško obrambno ministrstvo prepusti Internet raziskovalcem [13 str. 12],
- 1986 National Science Foundation (ameriška zvezna ustanova za financiranje raziskovalne dejavnosti ZDA) vzpostavi močno hrbtenico omrežja, znano kot NFSnet, in financira vzdrževanje hrbtenice, dodeljevanje imen (storitve imenskega prostora) in omrežnega informacijskega centra za Internet (NIC),
- 1989 vzpostavitev prvih komercialnih omrežij Internet, ki za medsebojno povezovanje uporabljajo NFSnet,
- ARPAnet preneha delovati,
- 1991 v ZDA sprejet zakon o računalniški dejavnosti na pobudo Al Gora, ki je zagotovil 2,9 milijarde ameriških dolarjev vladnega denarja za gradnje močne in hitre hrbtenice, ki je zagotovila dostop tudi komercialnim firmam,
- 1991 ustanovitev prvega CIX (Commercial Internet Exchange) v ZDA, nekakšno komercialno hrbtenico, podobno NFSnetu, ki zagotavlja enakopraven dostop vsem komercialnim ponudnikom; pozneje CIX-i nastanejo tudi v Evropi,

- 1992 ustanovitev Europanet – hrbtnice za evropska raziskovalna omrežja in ARNES (slovensko akademsko omrežje),
- 1992 ustanovitev združenja ISOC – Internet Society,
- 1995 NFSnet opustijo, storitve pa prevzame telefonska družba AT&T,
- 1995 ustanovitev slovenskega združenja za Internet (ISOC – Slovenian chapter of ISOC) [13 str. 13].

2 TCP/IP ARHITEKTURNI MODEL

Termin »TCP/IP« protokol je prevzel ime po dveh najpomembnejših internetnih protokolih: Transmission Control Protocol (TCP) in Internet Protocol (IP). Drugo ime, ki je manj v uporabi je Skupek internetnih protokolov (ang. Internet Protocol Suite) in se uporablja predvsem v uradnih dokumentacijah internetnih standardov. Od tu naprej bomo zaradi razumljivosti in kratkosti uporabljal termin TCP/IP za celoten protokol [12 str. 30].

2.1 Povezovanje med omrežji

Omrežja različnih vrst lahko povežemo med seboj (ang. Internetworking). Povezovanje med omrežji je usmerjanje med omrežji oziroma povezljivost različnih omrežij med seboj. Glavni cilj načrtovanja TCP/IP protokola je bil zagotoviti prav povezljivost med omrežji. Prednost, ki se tu pokaže, je omogočanje komuniciranja med gostitelji različnih omrežij, tudi tistih, ki jih ločujejo ogromne razdalje.

Besedi »Internetwork« in »internet« sta okrajšava fraze internetno povezanega omrežja. Kadar besedo internet pišemo z veliko začetnico se ta navezuje na svetovni splet internetnih omrežij [12 str. 30].

Internet sestoji iz sledečih omrežnih skupin:

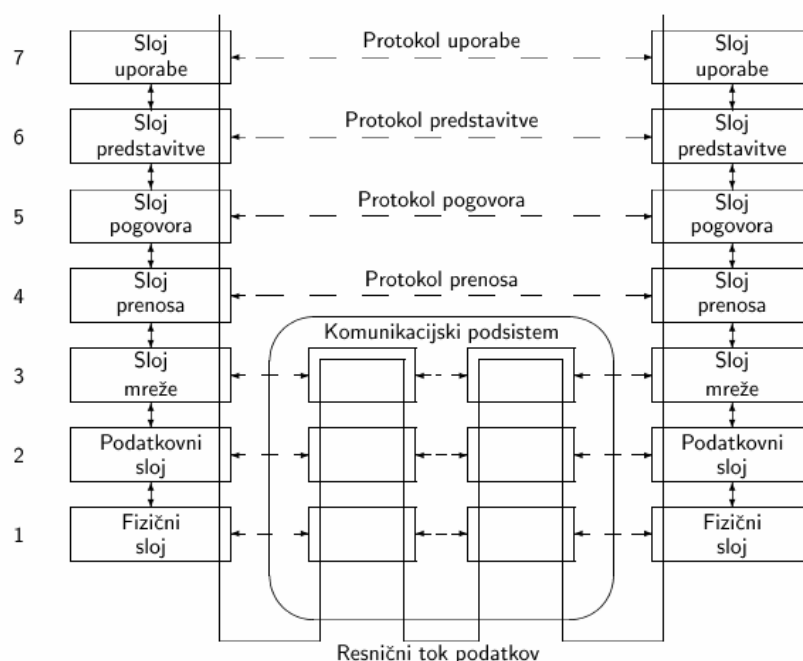
- hrbtenice: velika omrežja, ki so prvotno povezovala druga omrežja, poznana kot omrežno dostopne točke (ang. network access points, NAPs), ali internetno izmenljiva mesta (ang. internet exchange points, IXPs),
- regijske omrežne povezave (univerze, inštituti, ...),
- komercialna omrežja zagotavljajo uporabnikom in lastnikom zakupljenih omrežij dostop do hrbtenice,
- lokalno omrežje ima omejeno geografsko območje in je običajno omejeno na neko zgradbo ali organizacijo.

2.2 Referenčni model ISO OSI

Da bi zagotovili združljivost sistemov različnih proizvajalcev ter omejili število različnih izvedb komunikacijskih postopkov in opreme, je mednarodna organizacija za standardizacijo ISO [5 str. 187] v začetku osemdesetih let predlagala model, po katerem naj bi gradili računalniška omrežja. Referenčni model OSI [5 str. 358] (ang. Open Systems Interconnection, se nanaša na odprte sisteme, to je sisteme, ki so odprti za komunikacijo z drugimi odprtimi sistemi [2 str. 16]. Model OSI obsega sedem slojev, do česar so prišli ob upoštevanju naslednjih načel:

- sloj mora opravljati natančno določeno nalogo,
- nov sloj je potreben tam, kjer je potreben nov nivo abstrakcije,
- upoštevati je treba možnosti standardizacije,
- sloji morajo biti izbrani tako, da se čimbolj zmanjša pretok informacije med njimi (na vmesnikih),
- število slojev naj se zadrži v razumnih mejah, da ne bi s prevelikim številom slojev po nepotrebnem večali obsežnosti, vendar naj bo slojev dovolj, tako da ne bi bili prisiljeni združiti dveh načelno različnih aktivnosti v isti sloj.

Referenčni model ISO OSI je prikazan na sliki 2.1, kjer je razviden odnos med dvema končnima vozliščema, ki ju povezuje komunikacijski podsistem [2 str. 17]. Model OSI sam po sebi ni arhitektura mreže, saj ne določa niti storitev niti protokolov posameznih slojev.



Slika 2.1: ISO OSI referenčni model [2 str. 17]

Določa le funkcije posameznih slojev. Res je ISO izdelal ali privzel tudi standarde za posamezne sloje, vendar ti standardi niso sestavni del modela. Za posamezne sloje se lahko izbere poljubno kombinacijo protokolov. Iz tega sledi, da dve postaji, ki sicer komunicirata po referenčnem modelu OSI, a uporabljata različen protokol vsaj na enem sloju, med seboj ne moreta komunicirati. Postaje lahko med seboj komunicirajo samo, če uporabljajo enak sklad protokolov ali enak protokolovni 'profil' (ang. Protocol Stack, Protocol Suite, Protocol Profil), enake protokole na vseh nivojih. V praksi se nekatere bolj uveljavljene protokolovne sklade imenuje na kratko kar protokoli.

Fizični sloj (ang. Physical Layer) skrbi za prenos informacijskih signalov po komunikacijskem kanalu. Osnovna informacijska enota tega sloja je bit. Sloj določa mehanske, električne in postopkovne lastnosti naprav in tokokrogov [2 str. 18].

Podatkovni sloj (ang. Data Link Layer) uporablja storitve fizičnega sloja (prenos signalov) in zagotavlja naslednjemu višjemu sloju (mrežnemu sloju) visoko zanesljivost prenosa podatkov v obe smeri med sosednjimi vozlišči. V ta namen deli daljša zaporedja bitov na manjše okvirje (ang. Frames) ter skrbi za pravilen prenos posameznih okvirjev od vozlišča do vozlišča s preverjanjem pravilnosti prenesenega okvirja in s potrjevanjem pravilnosti sprejema.

Mrežni sloj (ang. Network Layer) skrbi za delovanje komunikacijskega podsistema. Na sliki 2.1 vidimo, da vmesna vozlišča, ki tvorijo komunikacijski podsystem, opravljajo naloge samo prvih treh (spodnjih) slojev, do vključno mrežnega sloja. Osnovna podatkovna enota tega sloja je paket (ang. Packet). Ključna naloga tega sloja je zagotavljanje poti prenosa od oddajnega do sprejemnega vozlišča. Pot od oddajnika do sprejemnika je lahko določena čisto statično (vnaprej z izvedbo omrežja in se nikoli ne spremeni), lahko pa je drugačna za vsako novo zvezo, za vsako novo sporočilo ali celo za vsak posamezen paket (dinamično).

Prenosni sloj (ang. Transport Layer) streže sosednjemu višjemu sloju (sloju pogovora). Od njega prevzema podatke, jih po potrebi deli na manjše enote - podatkovne enote prenosnega sloja (ang. Transport Protokol Data Unit - TPDU), zatem pa predaja naslednjemu nižjemu (mrežnemu) sloju ter skrbi, da podatki pogovornega nivoja pridejo pravilno na drugo stran [2 str. 19]. Torej skrbi za prenos informacije na drugo stran, kot pove tudi njegovo ime. Prenosni sloj je prvi 'pravi' sloj 'od-konca-do-konca' (ang. End-to-End). Z drugimi besedami, proces na eni končni postaji komunicira neposredno s sorodnim procesom na drugi končni postaji, brez posrednikov.

Pogovorni sloj (ang. Session Layer) omogoča 'pogovor' med uporabniškimi procesi na različnih postajah in s tem skrbi za organizacijo in strukturiranje dialoga. Podobno kot prenosni sloj omogoča prenos podatkov, poleg tega pa nudi tudi nekatere specifične storitve za določene aplikacije. Če bi za prenosni sloj rekli, da skrbi za prenos informacij med enim in drugim končnim vozliščem, skrbi pogovorni sloj za pogovor med istorodnima procesoma enega in drugega vozlišča. Večina omrežij tega sloja sploh nima. Na tem sloju je običajno (a ni nujno) realiziran sistem daljinskih klicev (ang. Remote Procedure Call), kot eden izmed možnih načinov komunikacije

med krajevno porazdeljenimi procesi.

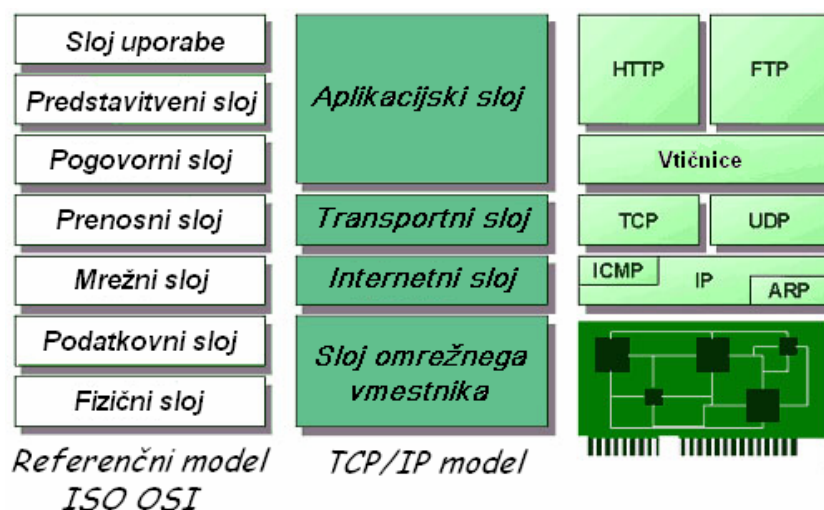
Predstavitveni sloj (ang. Presentation Layer) opravlja pogoste storitve, ki zahtevajo splošno rešitev za večino uporabnikov teh storitev. Tipične storitve tega sloja so: kodiranje in prekodiranje (npr. ASCII v EBCDIC), šifriranje, zgoščevanje podatkov in podobno.

Sloj uporabe (ang. Application Layer) predstavlja vmesnik med končnim uporabnikom in komunikacijskim sistemom. Vsebuje morda največ storitev in protokolov.

2.3 Sloji TCP/IP protokola

Kot večino omrežne programske opreme je tudi TCP/IP protokol razvrščen v sloje. Taka slojna predstavitev nakazuje uporabo termina protokolovnega sklada (ang. protocol stack), kateri se navezuje na sklad slojev v protokolu. Uporabljen je lahko za pozicioniranje in za funkcionalno primerjavo) TCP/IP protokola v primerjavi z drugimi, kot so SNA (Systems Network Architecture) in OSI (Open System Interconnection) model. Slika 2.2 prikazuje funkcionalno primerljivost med referenčnim ISO OSI modelom ter TCP/IP modelom.

Z delitvijo komunikacijskih programov v sloje dopušča protokolovni sklad delitev opravil, enostavnost implementacije in testiranje kode ter zmožnost razvijanja alternativnih slojnih implantov. Sloj n nudi določeno storitev samo sloju $n+1$ in za opravljanje svoje naloge izkorišča storitve sloja $n-1$. Na primer, IP nivo poskrbi za prenos podatkov od enega gostitelja k drugemu brez kakršne koli garancije o zanesljivem prenosu ali podvajanju. Transportni protokol TCP poskrbi, da se prenos izvede zanesljivo in v pravilnem vrstnem redu [12 str. 6].



Slika 2.2: Nivoji TCP/IP protokola [15]

Protokolovni sklad TCP/ IP se popolnoma ne ujema z referenčnim ISO OSI modelom. Namesto sedmih slojev uporablja le štiri. Pogosto je imenovan tudi kot internetni protokolovni sklad in je razdeljen na naslednje štiri sloje [14]:

- sloj omrežnega vmesnika (ang. network interface layer),
- internetni sloj (ang. Internet layer),
- transportni (prenosni) sloj (ang. transport layer),
- aplikacijski sloj (ang. application layer).

Sloj omrežnega vmesnika: ustreza fizičnemu sloju in sloju podatkovne povezave referenčnega modela OSI, komunicira neposredno z omrežjem. Je vmesnik med omrežno arhitekturo (kot je na primer vodilo z žetonom (ang. token bus), obroč z žetonom (ang. token ring) ali ethernet) in internetnim slojem. Sam protokolni sklad TCP/IP ne standardizira teh slojev in prav to je njegova velika prednost. Na sloju omrežnega vmesnika je veliko protokolov lokalnih omrežij (npr. Ethernet, token ring in FDDI), ki jih je TCP/IP zmožen povezati v prostrano omrežje s svojimi protokoli na višjih slojih

Internetni sloj: na internetnem (medmrežnem) sloju protokoli enkapsulirajo okvirje protokolov lokalnih omrežij v internetne datagrame in s pomočjo usmerjevalnih algoritmov paket dostavijo na predpisani cilj. Štirje protokoli so najpomembnejši na internetnem sloju: Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP) in Internet Group Management Protocol (IGMP).

IP je predvsem odgovoren za naslavljanje in usmerjanje paketkov med računalniki in omrežji. ARP služi za povezavo med strojnimi (npr. Ethernet) in IP naslovi strojne opreme računalnikov in drugih sistemov v istem fizičnem omrežju. ICMP pošilja sporočila in sporoča napake glede pošiljanja paketa. IGMP je uporabljen v komunikaciji med gostitelji in njihovimi agenti, ki so odgovorni za naslavljanje več računalnikom. Zagotavlja oblikovanje začasne skupine naslovov, dodajanje in brisanje članov iz skupine. Pošiljanje datagramov več gostiteljem (angl. multicast) je določen kot prenos datagramov k skupini gostiteljev. Vsak datagram, ki je namenjen več gostiteljem je dostavljen vsem članom skupine z zanesljivostjo, ki velja za običajne datagrame .

Transportni sloj: zagotavlja komunikacijsko sejo med računalniki. Dva prenosna protokola sta Transmission Control Protocol (TCP) in User Datagram Protocol (UDP). Izbira prenosnega protokola je odvisna od privzete metode pošiljanja podatkov. TCP skrbi za povezavne in zanesljive komunikacije. Uporablja se za potrditev pošiljanja podatkov.

UDP skrbi za nepovezavno komunikacijo in tako ne zagotavlja, da bodo poslani podatki prispeli. Aplikacije, ki ponavadi uporabljajo UDP, pošiljajo majhne količine podatkov naenkrat. Za zanesljivo pošiljanje podatkov (če je potrebno) je odgovorna aplikacija.

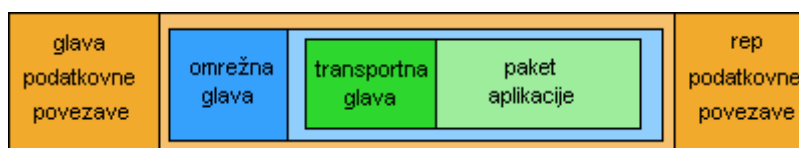
Aplikacijski sloj: je na vrhu TCP/IP sklada. To je sloj, kjer imajo aplikacije dostop do omrežja. Obstaja veliko standardnih storitev v aplikacijskem sloju, kot so splet, e-pošta, FTP, Telnet, SNMP, DNS.

2.3.1 Enkapsulacija podatkov

Povezava med protokoli, ki delujejo na različnih slojih ISO OSI modela povzroči, da različni sloji prevzamejo pakete od višjih slojev kot podatke in jim dodajo svojo glavo (v primeru Ethernet sloja tudi rep). Na primer, ko aplikacija sproži zahtevo za omrežni prenos, se prenašajo podatki od zgoraj navzdol skozi protokolni sklad. Aplikacijski sloj pošlje paket na prenosni sloj. Ko paket doseže prenosni sloj, mu protokol prenosnega sloja doda svojo glavo. Glava je sestavljena iz polj, ki vsebujejo informacije, ki so značilne za ta protokol. Originalni paket postane podatkovno polje v paketu protokola transportnega sloja [16].

Preden so podatki preneseni iz enega sloja na drugega, so najprej razgrajeni v paketke ali enote informacij, ki so prenesene kot celota iz enega računalnika na drugega v omrežju. Omrežje prenese paket iz enega programskega sloja na drugega v enakem vrstnem redu kot so sloji. Na vsakem sloju se programsko dodajo informacije ali naslov, da se paket lahko uspešno prenese po omrežju.

Na sprejemni strani gre paket skozi sloje v obratnem vrstnem redu. Služnostni programi na vsakem sloju preberejo k paketu dodane informacije, jih odstranijo in pošljejo paket naprej do naslednjega sloja. Ko paket končno prispe do aplikacijskega sloja, so dodane informacije odstranjene in tako je paket spet v svoji originalni obliki, ki je razumljiva sprejemniku. Sliki 2.3 prikazuje primer enkapsulacije podatkov.



Slika 2.3: Enkapsulacija podatkov [16]

Proces, s katerim protokoli dodajajo svojo glavo in rep, se imenuje podatkovna enkapsulacija (ang. data encapsulation). Proces je podoben tistemu pri pošiljanju pisma. Podatki se dajo v ovojnico, doda se ji naslove, znamko in se jih pošlje.

Z izjemo najnižjega sloja v omrežnem modelu ISO OSI noben sloj ne more poslati podatkov neposredno na drug računalnik. Podatke morajo na računalniku, ki

podatke pošilja, biti poslane po vsakem sloju navzdol dokler ne prispejo do fizičnega sloja. Nato se informacije pošljejo po prenosnem mediju do računalnika, ki podatke sprejema in nato potujejo po računalniških slojih, dokler ne prispejo do sprejemnega sloja.

Interakcija med istoležnimi sloji se dogaja preko vmesnih slojev. Vmesni sloj definira ponujene storitve najnižjih omrežnih slojev do zgornjih in naprej določa kakšen bo vstop do teh slojev. Vsak sloj na računalniku navidezno komunicira neposredno z enakoležečim slojem na drugem računalniku.

2.4 IP Naslavljanje

IP naslov je predstavljen z 32-bitno nepredznačeno binarno vrednostjo, ki jo pišemo v grupah po tri številke, ločene s pikami. Na primer: 216.239.39.99 je veljaven internetni naslov. Preslikava internetnega IP naslova v nam razumlivejšo obliko (www.google.com), opravi Sistem za domenska imena (ang. Domain Name System) DNS.

Da lahko vsakega gostitelja identificiramo, je potrebno, da ima le ta svoj unikaten IP naslov. Vsak IP naslov se sestoji iz dveh delov:

- omrežna številka ali omrežno identifikacijo (*ang. net ID*),
- številko gostitelja ali gostiteljevo identifikacijo (*ang. host ID*).

IP naslov = <omrežna številka><številka gostitelja>

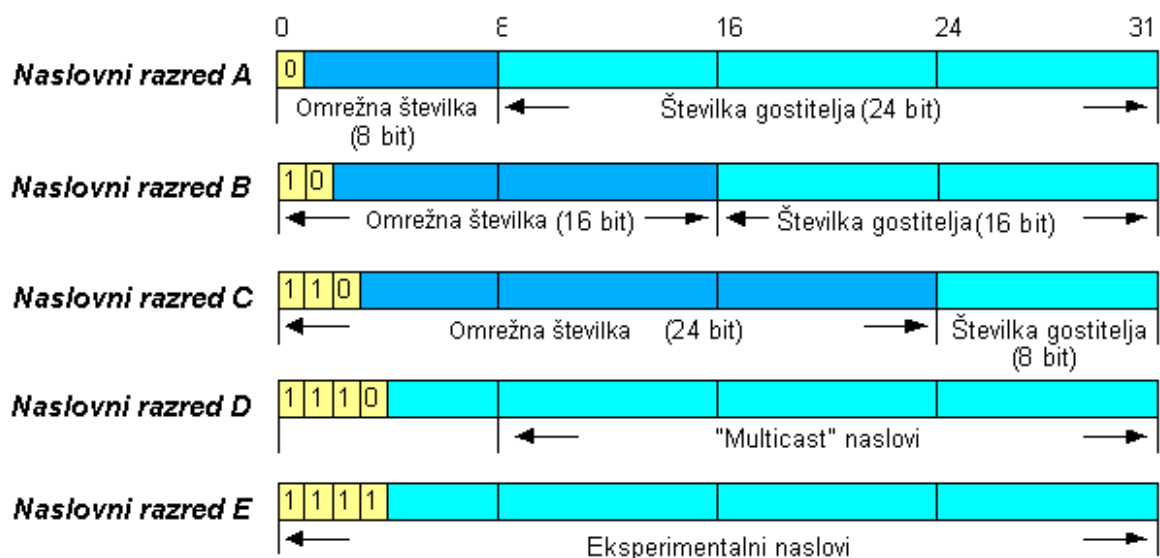
Omrežna številka je priskrbljena s strani Omrežnega Internetnega Informacijskega Centra (InterNIC) in je unikatna v celotnem Internetnem omrežju. Na primer: 128.2.7.9 je predstavitev IP naslova z omrežno številko 128.2 in številko gostitelja 7.9. V nadaljevanju bomo opisali pravila za delitev IP naslova [12 str. 68].

Binarna vrednost internetnega naslova 128.2.7.9 je:

1000000 00000010 00000111 00001001

IP naslov uporablja IP protokol za unikatno identifikacijo gostiteljev omrežja. Za pošiljanje datagramov določenemu IP naslovu mora biti ciljni IP naslov preveden v fizični naslov, za kar poskrbi ARP (Address Resolution Protocol) .

Naslovno področje Interneta je razdeljeno v razrede (slika 2.4). Vsak razred opredeli del naslova IP, ki prepozna omrežno predpono in del, ki prepozna gostiteljevo predpono. naslovni razredi IP so A, B in C.



Slika 2.4: Prikaz naslovnih razredov [17]

Internetna skupnost določa pet IP razredov, ki so prilagojeni omrežjem različnih velikosti in posebnim namenom. Naslovni razred definira, kateri biti se uporabljajo za omrežni del in kateri biti se uporabljajo za gostiteljev del. Razredi tudi določajo največje možno število omrežij in največje možno število gostiteljev v omrežju. Naslovni razred lahko določimo s pomočjo števil v prvem oktetu. 32-bitni način naslavljanja IP podpira do 3.720.314.628 gostiteljev [17].

2.4.1 Naslovni razred A

Naslovi razreda A so dodeljeni omrežjem z zelo velikim številom gostiteljev. Bit najvišjega reda v naslovnem razredu A je vedno 0. Naslednjih sedem bitov, ki zapolnijo prvi oktet, dopolnijo omrežni ID. Preostalih 24 bitov (zadnji trije okteti)

predstavljajo gostiteljev ID. To omogoča največ 126 omrežij in skoraj 17 milijonov gostiteljev v omrežju (natančno število je 16.777.214). Za omrežja razreda A se je uporabila kar polovico celotnega prostora, čeprav so redka omrežja s tolikšnim številom gostiteljev. Omrežni ID ne more biti 127, ta predpona je rezervirana za diagnostične funkcije (localhost).

2.4.2 Naslovni razred B

Naslovi razreda B so dodeljeni srednje velikim in velikim omrežjem. Dva najvišja bita v razredu B sta vedno dvojiško 10. Naslednjih 14 bitov (dopolni prva dva okteta) sestavlja omrežni ID. Preostalih 16 bitov (zadnja dva okteta) predstavljata gostiteljev ID. To omogoča največ 16384 omrežij in približno 65.000 gostiteljev na omrežje (natančno število je 65.534).

2.4.3 Naslovni razred C

Naslovni razred C uporabljamo za manjša lokalna omrežja (ang. local area network). Trije biti najvišjega razreda v naslovnem razredu C so vedno dvojiško 110. Naslednjih 21 bitov (dopolni prve tri oktete) sestavlja omrežni ID. Preostalih 8 bitov (zadnji oktet) predstavlja gostiteljev ID. To dovoljuje nekaj več kot 2 milijona omrežij in 254 gostiteljev v omrežju. Omrežij tega razreda je največ.

2.4.4 Naslovni razred za pošiljanje več sistemom (D)

Naslovi razreda D se uporabljajo v skupinah za oddajanje več sistemom (ang. multicast). Te skupine lahko vsebujejo enega ali več gostiteljev ali pa sploh nobenega. Štirje najvišji biti v naslovnem razredu D so vedno dvojiško 1110. Naslednji biti določajo specifično skupino, v kateri odjemalec sodeluje. Tu ni omrežnih ali gostiteljevih bitov. Paketi so podani izbranim skupinam gostiteljev na omrežju. Samo gostitelji, ki so registrirani pri določenem naslovu, prejmejo paket. IP naslov je lahko stalno določen (ang. well-known address) ali pa je ustvarjen začasno (ang. transient multicast group). Uporablja se za videokonference in spletno televizijo.

2.4.5 Eksperimentalni naslovni razred (E)

To so eksperimentalni naslovi, ki niso na voljo splošni uporabi. So uporabljeni pri preskušanju interneta in so bili včasih rezervirani za rabo v prihodnosti. Biti z najvišjimi utežmi v naslovnem razredu E so 1111.

2.4.5.1 Posebni naslovi IP

Naslov 255.255.255.255 se lahko uporablja za razpršeno (ang. broadcast) naslavljanje v lokalnem omrežju. Usmerjevalniki preprečujejo uporabo razpršenega naslavljanja.

Naslov 127.0.0.1 ali localhost je edini v celotnem omrežju 127.0.0.0. Uporablja se za testiranje lokalnega sistema, ali za pošiljanje podatkov med različnimi protokoli (storitvami), ki tečejo na istem računalniku.

RFC (ang. Request For Comments) 1597 rezervira določene IP naslove za zasebna omrežja. Ker ti naslovi niso dostopni v internetu, ni potrebno, da so unikatni. Ti naslovi so:

10.0.0.0 — 10.255.255.255 (v naslovnem razredu A),

172.16.0.0 — 172.31.255.255 (v naslovnem razredu B),

192.168.0.0 — 192.168.255.255 (v naslovnem razredu C).

2.5 Protokol za pretvorbo naslovov (ARP)

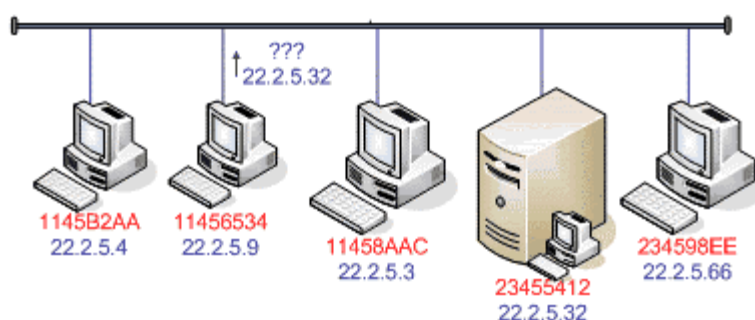
Vsak računalnik v internetu mora imeti vsaj en IP naslov, vendar je ta naslov na sloju podatkovne povezave nepomemben podatek, saj protokoli na tem sloju ne razumejo IP naslovov. Lokalna omrežja razumejo samo naslove lokalnih omrežij (MAC ali strojne naslove). Postavlja se vprašanje, kako naj se IP paket zapakira v okvir ethernet ali token ring. Uporabiti moramo protokol za prevedbo naslovov [18].

ARP je odgovoren za pridobivanje strojnih naslovov gostiteljev IP v lokalnem omrežju, ki temeljijo na razpršenem oddajanju (ang. broadcast). ARP uporablja razpršeno oddajanje ciljnega naslova IP, da bi pridobil strojni naslov ciljnega računalnika ali usmerjevalnika. Preden lahko IP paket pošljemo do drugega računalnika, mora biti poznan strojni naslov naprave ali računalnika, ki mu je paket namenjen (oziroma lokalnega usmerjevalnika, če ciljni računalnik ni na istem

lokalnem omrežju). ARP določa strojne naslove (naslove MAC), ki so povezani z IP naslovom. Če ARP ne vsebuje naslova v svojem lastnem pomnilniku, odda zahtevo za pridobitev naslova. Vsi gostitelji v omrežju procesirajo zahtevo, in tistemu, kateremu je dodeljen ciljni IP naslov, vrne odgovor, ki vsebuje lastni strojni naslov.

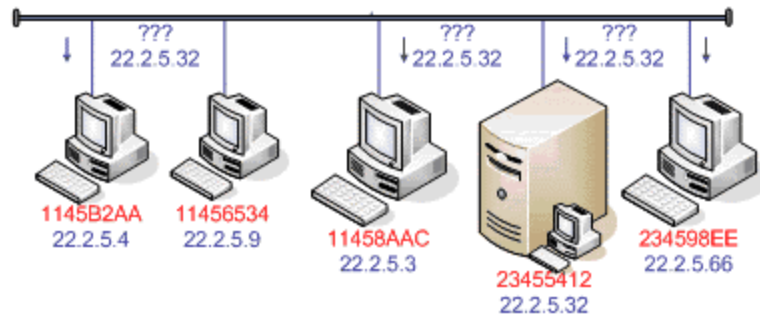
Preden se lahko komunikacija med dvema računalnikoma oziroma gostiteljema prične, se mora IP naslov vsakega računalnika pretvoriti v strojni naslov. Ta postopek vsebuje ARP zahtevo in ARP odgovor, kot prikazuje naslednji primer.

ARP zahteva se izvede vedno, ko želi nek računalnik komunicirati z drugim računalnikom. Ko IP sloj določi, da je IP naslov v lokalnem omrežju, izvorni gostitelj pregleda svoj ARP pomnilnik (slika 2.5) za strojni naslovnega gostitelja oziroma računalnika.



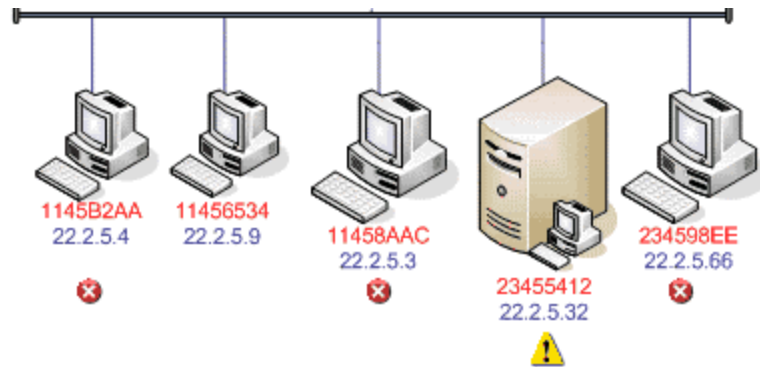
Slika 2.5: Pregled tabele ARP [18]

Če ne najde naslova, ARP sestavi zahtevo z vprašanjem "Kdo ima ta IP naslov in kakšen je njegov strojni naslov?" IP in strojni naslov izvirnega računalnika sta vsebovana v zahtevi ARP. Zahteva ARP je poslana z razpršenim oddajanjem (ang. broadcast) tako, da jo lahko sprejmejo in obdelajo vsi gostitelji (slika 2.6).



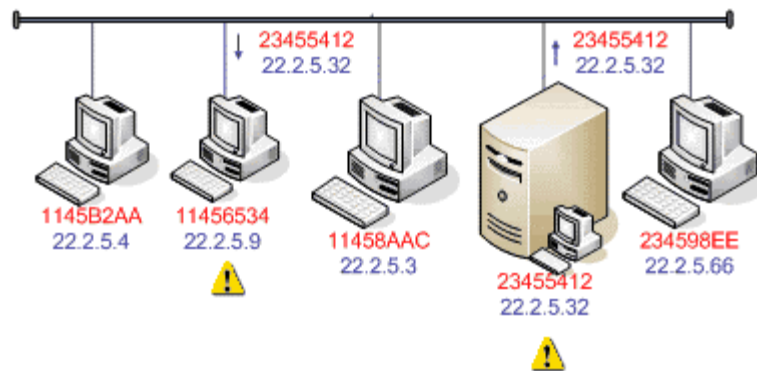
Slika 2.6: Razpršeno pošiljanje paketa [18]

Vsak gostitelj v omrežju prejme ARP zahtevo in preveri svoj IP naslov. Če gostitelj ne najde enakosti prezre zahtevo, kot prikazuje slika 2.7.



Slika 2.7: Primerjanje IP naslova [18]

Naslovni gostitelj ugotovi, da se zahtevani IP naslov ujema z njegovim lastnim, zato pošlje odgovor ARP neposredno do izvirnega gostitelja (oziroma računalnika, vozlišča) skupaj z svojim strojnim naslovom. Nato posodobi svoj ARP pomnilnik z IP naslov in strojnim naslovom izvirnega računalnika. Komunikacija se zaključi, ko izvorni računalnik prejme odgovor (slika 2.8).



Slika 2.8: Pošiljanje strojnega naslova povpraševalcu [18]

2.5.1 Protokol za obratno pretvorbo naslovov (RARP)

Protokol za obratno prepoznavanje naslovov (ang. reverse address resolution protocol). Računalniki, ki nimajo trdih diskov in se zaganjajo iz pomnilnika na omrežni kartici, nimajo vnaprej dodeljenega IP naslova. Imajo pa strojni naslov, saj je le-ta enolično določen z mrežno kartico. Tak računalnik svoj IP naslov pridobi iz RARP strežnika. RARP strežnik vsebuje zbirko podatkov v obliki tabele ARP, ki jo ustvari sistemski administrator. V nasprotju z ARP, RARP protokol ponuja IP številko za napravo, ki sporoča strojni naslov. Ko RARP strežnik sprejme zahtevo po IP številki od vozlišča v omrežju, odgovori tako, da pošlje ustrezno IP številko iz usmerjevalne tabele nazaj. Danes se v te namene večinoma uporablja protokol DHCP (Dynamic Host Configuration Protocol). DHCP dinamično dodeljuje IP naslove iz določenega obsega, ki jih določi administrator. Ko je IP naslov dodeljen postaji, DHCP strežnik ne dovoli več dodeljevanja tega IP naslova drugi postaji. To preprečuje podvajanje IP naslovov [18].

2.6 Internetni protokol za krmilna sporočila (ICMP)

Internet control message protocol (ICMP) je eden izmed osnovnih protokolov v protokolovnem skladu TCP/IP. Uporabljajo ga omrežni računalniki za pošiljanje sporočil o napakah, na primer da storitev ni dosegljiva ali računalnik ni dosegljiv.

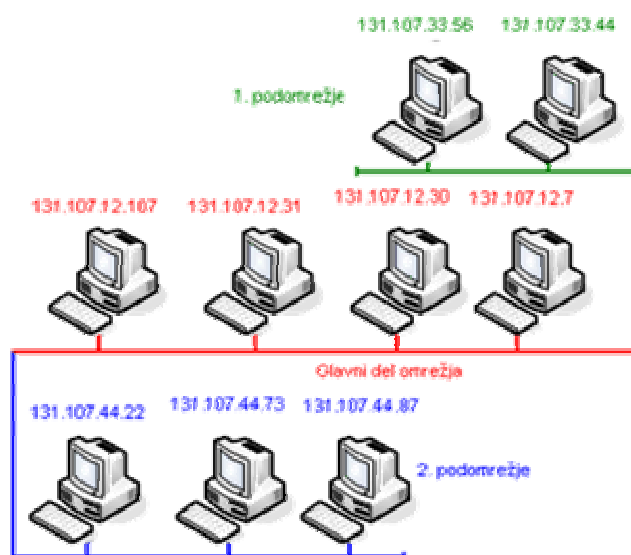
ICMP se razlikuje od TCP in UDP v tem, da običajno ni na voljo aplikacijskemu sloju. Ena izmed izjem je pripomoček *ping*, ki tvori sporočila ICMP za ugotavljanje dosegljivosti nekega gostitelja in časa dostopa do gostitelja [21].

ICMP sporočila so tipičen odgovor na napake v datagramih IP ali kot diagnostično orodje za usmerjevalne namene. Različica ICMP v internet protokolu verzije 4 (Ipv4), se imenuje ICMPv4. ICMP sporočila so generirana na internetnem sloju. IP enkapsulira sporočila ICMP, jim doda glavo ter jih pošlje na zahtevani cilj. Na primer, vsaka naprava (kot so usmerjevalniki), ki posreduje IP datagrame, zmanjša polje TTL (ang. time to live) v glavi protokola IP za ena. Če TTL doseže vrednost nič, določena naprava pošlje sporočilo ICMP izvirnemu sistemu, da je TTL potekel. Paketi s sporočili ICMP so enkapsulirani neposredno v en sam IP datagram, kar pomeni, da dostava sporočil ni zagotovljena.

2.7 Podomreževanje

Vsi gostitelji (*ang. hosts*) v omrežju morajo imeti isto omrežno identifikacijo (številko). Ta lastnost omrežja povzroča težave, ko omrežja rastejo. Na primer: šola ima omrežje razreda C in ima 200 gostiteljev. Kasneje želi dodati 100 gostiteljev, vendar je največje možno število gostiteljev 254. V osnovni izvedbi internetnih omrežij sta obstajali dve možni rešitvi te situacije.

Šola doda še eno omrežje razreda C in ima tako dve omrežji, ki lahko skupaj imata do 508 gostiteljev. V tem primeru komunikacija med omrežjema poteka preko usmerjevalnika. Druga možnost je razširitev omrežja na razred B, ki dovoljuje 65.534 gostiteljev. Izkoriščenost števila gostiteljev je majhna. Slika 2.9 prikazuje podomreževanje [20].



Slika 2.9: Podomreževanje [20]

Zaradi premajhnega števila omrežij v B razredu in omenjene slabe izkoriščenosti naslovov, se je v teku razvoja interneta razvila alternativna rešitev uporabe podomrežij. Deljenje IP omrežja na podomrežja zahteva, da vsak segment uporablja različen omrežni ID ali podomrežni ID. Edinstveni podomrežni ID je ustvarjen za vsak segment z delitvijo bitov v gostiteljevem ID v dva dela. En del je uporabljen za identificiranje segmenta kot edinstvenega omrežja, drugi del pa identificira gostitelje.

Za podomreževanje obstaja več razlogov. Organizacije podomrežujejo, da lahko naredijo eno logično omrežje z uporabo več fizičnih segmentov. S tem lahko:

- mešamo različne tehnologije, kot sta ethernet in token ring,
- premagamo omejitve obstoječe tehnologije, kot npr. prekoračitev najvišjega števila gostiteljev na segment,
- omejimo omrežno prezasedenost s preusmerjanjem prometa in omejitvijo razpršenega oddajanja

2.7.1 IP maska

Z razširjanjem interneta po svetu se je povečala potreba po IP naslovih. Internet se je razširil tudi v države, ki so bile prej računalniško skoraj »nepismene«. IP4 naslovi so danes praktično razdeljeni. Ker je internet pod ameriškim nadzorom, ima tako

univerza v Standfordu več IP naslovov kot celotna Kitajska. Rešitev, ki jo ponuja podomreževanje za povečanje razpoložljivih IP naslovov je IP maska. Uporaba omrežne maske je danes obvezna [20].

Le malo omrežij na svetu je razreda A, vendar je temu razredu namenjenega polovico naslovnega prostora. Z uporabo (pod)omrežne maske razdelitev omrežij v naslovne razrede ni več potrebna.

Uporaba podomrežjevanja

Pred razdelitvijo na podomrežja, je potrebno razmisliti, kaj se bo v omrežju dogajalo v prihodnje:

- koliko bo fizičnih segmentov v omrežju in
- koliko bo postaj v vsakem segmentu; Vsak gostitelj TCP/IP potrebuje vsaj en IP naslov.

Določiti je potrebno:

- podomrežno masko za celotno omrežje,
- unikatno podomrežno masko za vsak segment,
- številke gostiteljev za vsako podomrežje.

2.7.2 Biti podomrežne maske

Podomrežna maska je 32-bitno število. Kombinacija bitov v podomrežni maski določa največje število postaj v določenem podomrežju. Pri določanju podomrežne maske je potrebno misliti na prihodnji razvoj omrežja. Več bitov se uporabi za omrežni del, na več podomrežij je mogoče razdeliti omrežje, toda manj postaj je lahko v podomrežju

Spodnji izračun prikazuje izračun postaj v podomrežjih naslovnega razreda B:

Porabimo 3 bite za omrežni del = 6 podomrežj; 8190 postaj na podomrežje,

Porabimo 8 bite za omrežni del = 254 podomrežj; 254 postaj na podomrežje.

Če se uporabi več bitov za podomrežno masko, je mogoče v prihodnje povečati število podomrežij, toda omejeno je število postaj v podomrežju. Če se uporabi

manj bitov, to dovoljuje povečanje števila postaj v podomrežju, toda omejeno je število podomrežij.

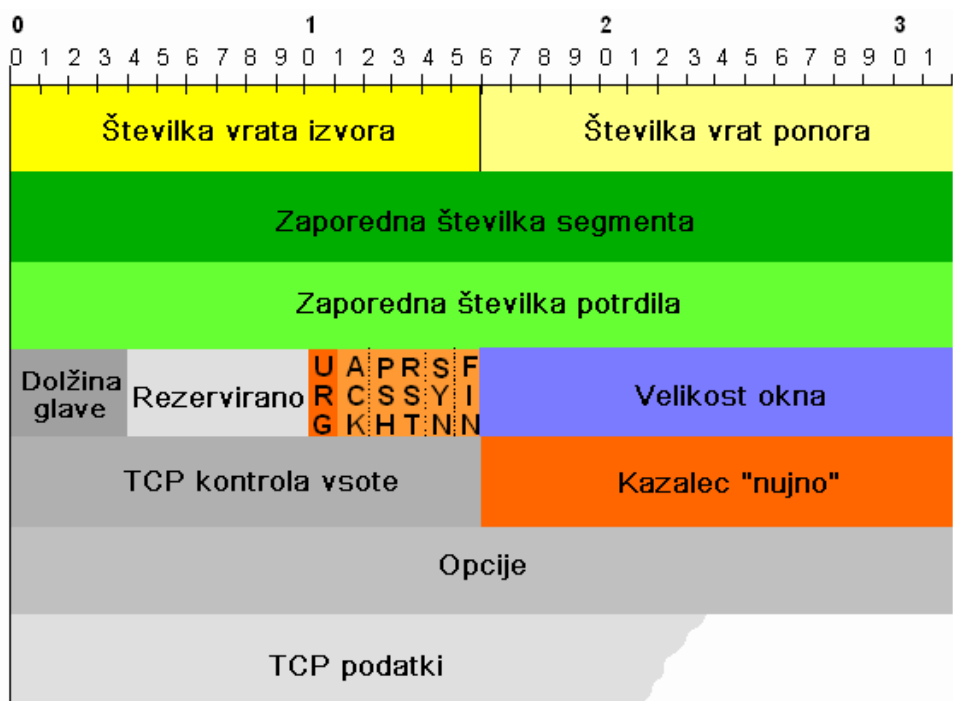
2.8 Transportni sloj TCP

V skladu s pravili transportnega sloja v modelu ISO OSI so protokoli transportnega sloja odgovorni za vzpostavljanje in vzdrževanje povezave med dvema končnima gostiteljema. Transportni cilj poskrbi za potrditve uspešnega prenosa, kontrolo pretoka in pravilni vrstni red paketov. Upravlja tudi ponovno pošiljanje paketov. Transportni sloj v TCP/IP lahko uporablja protokol TCP ali UDP, odvisno od zahtev in potreb pošiljanja [22].

Protokolovni sklad TCP/IP je dobil ime po kombinaciji protokolov TCP in IP, ki skupaj zagotavljata storitve, ki upravljajo z večino prometa v omrežju TCP/IP. Internetne aplikacije, kot so spletni brskalniki, odjemalci FTP in pregledovalniki elektronske pošte so vsi odvisni od protokola TCP, ki je sposoben prenašati velike količine podatkov brez napak. TCP je definiran v dokumentu RFC 793, ki ga je objavila IETF (ang. Internet Engineering Task Force) leta 1981.

2.8.1 TCP

Transmission Control Protocol (TCP) je protokol transportnega sloja in se uporablja za zanesljiv prenos podatkov med gostitelji in omrežji. Je povezovalni protokol, ki vzpostavlja povezave (imenovane tudi seja ali navidezni kanal) med dvema računalnikoma, preden so podatki poslani. Po potrebi TCP sloj podatke razdeli na obvladljivo majhne dele, in jim doda TCP glavo. Za vzpostavitev zaneslive povezave TCP uporablja metodo trosmernega usklajevanja (ang. three-way handshake). Ker je TCP povezavni protokol, se najprej vzpostavi povezava med odjemalcem in pošiljateljem. Pri povezavi je določen odjemalčev IP naslov in vrata (ang. port), ter pošiljateljev IP naslov in vrata, na katerih storitev posluša strežnik. IP Naslov, povezan z določenimi vrati, tvori vtičnico (angl. socket). Par odjemalčeve in pošiljateljeve vtičnice tvori TCP povezavo, ki je edinstveno določena. Slika 2.10 prikazuje TCP paket.



Slika 2.10: TCP paket [19]

2.8.2 Vrata in vtičnice

Tako TCP kot UDP s specifikacijo številke vrat identificirata protokol ali proces, ki generira ali sprejema podatke. Standardno dodeljene številke vrat so objavljene v RFC 1700 skupaj s seznamom dobro poznanih vrat v TCP/IP. Aplikacije in storitve lahko uporabljajo do 65.536 vrat. Da bi vtičnico specificirali v URL moramo najprej napisati IP naslov, nato pa mu sledimo s številko vrat. Tako na primer vtičnica 192.168.2.10:21 naslavlja vrata 21 na sistemu s tem IP naslovom. Ker ima FTP številko vrat 21, ta vtičnica naslavlja FTP strežnik, ki teče na tem računalniku.

Tipične aplikacije in storitve TCP/IP običajno uporabljajo prvih 1023 vrat (slika spodaj). To so tako imenovana dobro poznana vrata (ang. well-defined ports) [22].

	F T P	te in et		w w w	aplikacije	T F T P		ntp	
	21	25	vrata TCP	80		69	vrata UDP	123	
TCP					UDP				
IP									

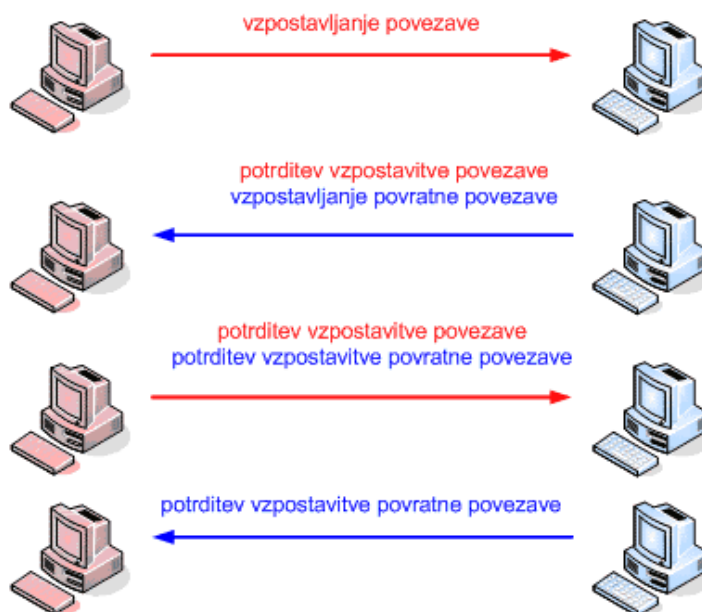
Slika 2.11: Delitev vrat [22]

Običajno ni potrebno določiti številko vrat, ko vpisujemo URL, ker program ki ga uporabljamo sklepa, da se želimo povezati na dobro znana vrata. Na primer, spletni brskalnik naslovi vse URL, ki jih vpišemo, na vrata 80, ki so vrata HTTP protokola spletnega strežnika, razen če administrator ne določi drugače. IANA številke vrat so priporočila, ne pravila. Spletni strežnik lahko konfiguriramo tako, da uporablja druga vrata in veliko administratorjev dodeli alternativna vrata zato, da jih lahko dosežejo le uporabniki, ki poznajo pravo številko vrata. Npr. ustvarimo srednje skrito spletišče tako, da številko vrat nastavimo na 81; za dostop moramo napisati na primer: `www.mojnaslov.com:81`.

Storitve in aplikacije uporabljajo vtičnice, da vzpostavijo povezave z drugimi gostitelji. Če morajo aplikacije zagotoviti dostavo podatkov, potem vtičnica uporabi povezavno orientirano storitev (TCP). Če ne potrebuje potrditve dostave podatkov, se uporabi nepovezavno storitev (UDP).

2.8.3 Vzpostavljanje povezave

TCP je povezavno usmerjen protokol, kar pomeni, da se mora vzpostaviti povezava med dvema sistemoma, preden lahko izmenjujeta podatke. Ta povezava poskrbi, da sta oba računalnika prisotna in delujeta pravilno ter sta pripravljena na sprejemanje podatkov. TCP povezava ostane aktivna skozi ves proces izmenjave podatkov. Slika 2.12 prikazuje postopek vzpostavitve povezave.



Slika 2.12 : Vzpostavitev povezave s TCP [22]

V večini primerov TCP povezava obstaja za čas izmenjave datoteke. Na primer, ko spletni brskalnik vzpostavi povezavo s strežnikom v Internetu, najprej vzpostavi povezavo, nato sprejme podatke, ki so zahtevani v URL. Ko je datoteka enkrat prenesena, sistem poruši povezavo. Ko brskalnik procesira preneseno datoteko, lahko zasledi povezave do slik, zvočnih izsekov ali drugih datotek, potrebnih za prikaz spletne strani. Brskalnik nato vzpostavi povezavo do strežnika za vsako datoteko, ki je del spletne strani. Zato lahko ena sama spletna stran potrebuje ducate TCP povezav do strežnika za prenos datotek [22].

Proces vzpostavljanje povezave sestavlja izmenjava treh sporočil. Nobeno od sporočil ne vsebuje podatkov aplikacijskega sloja. Namen teh sporočil je, razen ugotavljanja prisotnosti drugega računalnika, predvsem v izmenjavi začetnih števil, s katerimi bodo računalniki označevali poslane podatke.

Vsak sistem, ki sprejme podatke, pošlje svoj odgovor. Ko enkrat strežnik dobi odjemalčevo potrditev, je povezava vzpostavljena in sistemi so pripravljeni, da izmenjujejo sporočila, ki vsebujejo aplikacijske podatke. Zato je povezava TCP v bistvu sestavljena iz dveh ločenih enosmernih povezav, ki tečeta v nasprotni smeri, s čimer je TCP polno dvosmeren protokol.

Pomembno je imeti v mislih, da je povezava, vzpostavljena med TCP sistemoma, le logična povezava. Posamezna TCP sporočila so še vedno vsebovana v datagramih IP in uporabljajo nepovezavne storitve IP. Sporočila lahko uporabijo različne poti do cilja in lahko celo prispejo v drugačnem vrstnem redu kot so bila poslana. TCP je narejen tako, da računa na vse te možnosti in segmente datoteke spravi v pravilen vrstni red.

2.8.4 Mehanizmi potrjevanja

Potrjevanje je namenjeno odpravljanju napak. Glede na funkcionalno razmejenost sloja ni težko razumeti, da so tipi napak odvisni od protokolnega sloja, na kateri deluje določen protokol. Poglejmo si najbolj tipične napake, ki jih srečujemo na posameznih slojih [6 str. 152]:

Na fizičnem sloju lahko opazujemo stanje konektorja (vključen ali izključen) in o statusu fizične povezave obveščamo sistem in seveda tudi končnega uporabnika. Napake so največkrat posledica okvar ali motenj na strojni opremi.

Na podatkovnem sloju odkrivamo napake, ki se zgodijo pri prenosu okvirja po neidealnem prenosnem mediju. Večina bralcev je že slišala za kontrolo parnosti (ang. parity check) ali preverjanje CRC (ang. cyclic redundancy check), ki omogočata sprejemniku identificirati paket, na katerem se je na prenosnem mediju pojavila sprememba - napaka. Seveda bo sprejemnik v takem primeru ustrezno reagiral - zahteval bo na primer, da se določena paket pošlje ponovno.

Na omrežnem in transportnem sloju se odkrivanje napak omeji predvsem na kontrolo zaporedja sprejetih paketov. To pomeni, da se ugotavlja, ali kak paket manjka, ali gre za večkrat sprejet isti paket, ali pa se je spremenil vrstni red paketov pri sprejemniku.

Na sejnemu, predstavitvenemu in aplikacijskemu sloju gre predvsem za odkrivanje vsebinskih, semantičnih nepravilnosti uporabniških podatkov.

Le nekaj vrst napak je takih, da jih pod določenimi pogoji sprejemnik lahko odpravi avtonomno: večkratni sprejem istih paketov lahko sprejemnik reši tako, da take pakete identificira s pomočjo zaporedne številke in nato kopije zavrže, drug tipičen

primer avtonomnega odpravljanja napak pa je urejanje vrstnega reda sprejetih paketov na osnovi zaporednih števil - seveda za to potrebuje vgrajeno sposobnost za sortiranje.

Na nižjih slojih gre po pravilu za opravljanje napak na sloju posameznega paketa, medtem ko na višjih slojih, kjer je poleg sintakse pomembna predvsem vsebina aplikacijsko-uporabniških podatkov, napake odpravljamo na sloju sporočila.

Ne glede na sloj obravnavanja napak govorimo o pozitivni (ACK) in negativni potrditvi (NACK). Ti dve sporočili sta namenjeni kontroli pravilnosti/nepravilnosti izvajana določene povezane storitve. Sprejemnik pošlje sporočilo ACK, ko spozna sprejeti paket/sporočilo kot neoporečen, v nasprotnem primeru pa odgovori s sporočilom NACK. Na osnovi teh dveh sporočil (ACK, NACK) lahko oddajnik ustrezno ukrepa oziroma v sodelovanju s sprejemnikom odpravi napako.

Seveda je smiselno tudi to, da mehanizmi potrjevanja zagotavljajo avtomatsko in za vsak sloj neodvisno odpravljanje napak. To pomeni, da je odpravljanje napak za preostale sloje, predvsem pa za uporabnika, transparentno [6 str. 153-154].

2.8.5 Kontrola pretoka z drsečim oknom

Kontrola pretoka je mehanizem, s katerim ciljni sistem v TCP povezavi varuje čakalne vrste podatkov pred prekomerno napolnitvijo, tako da nadzira število oddanih podatkov, ki jih sprejemnik še ni sprejel. Spodnja slika prikazuje koncept drsečega okna [6 str. 178-179].



Slika 2.13: Kontrola pretoka z drsečim oknom [22]

Vsak sistem ima omejeno količino medpomnilniškega prostora (ang. buffer), v katerega shranjuje prihajajoče podatke. Podatki tam ostanejo dokler sistem ne generira potrditvenih sporočil. Delovanje protokola z drsečim oknom je sledeče.

- Sprejemnik potrjuje sprejete zloge po enem od potrditvenih protokolov, ko pa se sprejemnikova čakalna vrsta dovolj napolni, preneha potrjevati sprejete pakete.

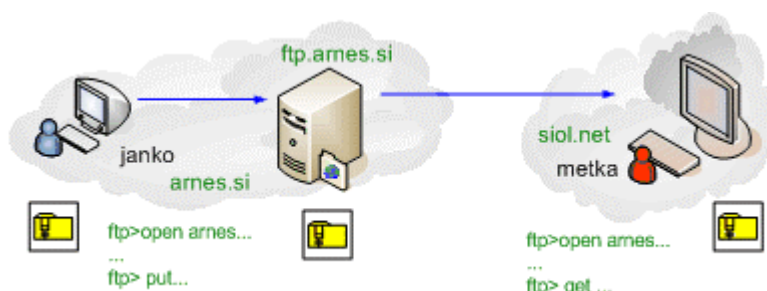
- Posledica tega je, da bo oddajnik poslal le še toliko zlogov, da bo zapolnil kvoto, ki mu je dovoljena brez potrditve, nato pa se bo ustavil. Z drugimi besedami, oddajnik ugotovi, da je maksimalno število nepotrjenih poslanih paketov doseglo maksimum, in neha oddajati.
- Ko se sprejemnikova čakalna vrsta dovolj izprazni, sprejemnik začne vnovič potrjevati sprejete zloge, ob sprejemanju potrditev pa oddajnik začne vnovič oddajati.

Širino okna imenujemo maksimalno dovoljeno število nepotrjenih paketov, ki jih oddajnik lahko pošlje sprejemniku.

2.9 Protokol za prenos datotek FTP

Protokolovni sklad TCP/IP ima dvoje protokolov za prenos datotek: FTP (ang. File Transfer Protocol) in TFTP (ang. Trivial FTP). Prvi uporablja zanesljiv prenos s protokolom TCP, drugi pa poteka preko protokola UDP.

FTP Protokol izvira iz operacijskega sistema Unix, kjer se je prav tako uporabljal za prenose datotek med računalniki. Z razvojem interneta je ta protokol, ki je eden od starejših standardov interneta, postal vsestransko uporabljen protokol za prenos datotek. Konceptualni potek FTP protokola prikazuje slika spodaj .

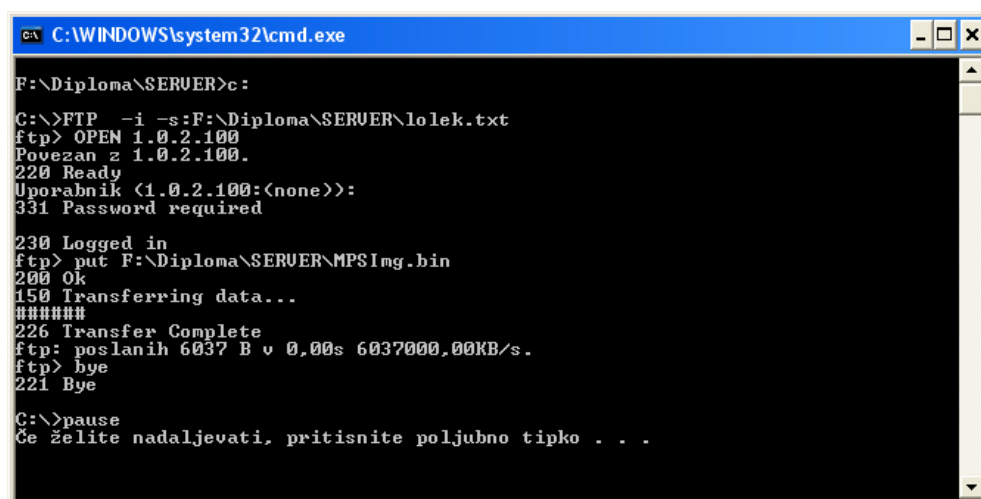


Slika 2.14: FTP protokol [23]

FTP protokol uporablja dvoje vrat, 20 in 21. Vrata 20 so uporabljena za vzpostavitev povezave in za nadzor povezave, vrata 21 pa so uporabljena za

uporabnikove datoteke (datoteke, ki bodo prenesene). V zadnjem času je FTP protokol vse manj v rabi, ker ga izpodriva HTTP (ang. Hypertext Transfer Protocol) protokol. Razlog je v tem, da HTTP poleg polne funkcionalnosti FTP protokola podpira še druge funkcije [3 str. 693].

FTP klienta zaženemo iz ukazne vrstice z ukazom FTP. Povezavo s strežnikom vzpostavimo z ukazom OPEN in imenom strežnika. Ko je zveza med odjemalcem in strežnikom vzpostavljena, strežnik čaka na ukaze odjemalca. Z ukazom GET odjemalec dostopa do datotek na strežniku, z ukazom PUT pa jih pošilja strežniku. Ko je prenos podatkov zaključen, se podatkovna vrata zaprejo. Za prikaz vseh ukazov napišimo v ukazni vrstici FTP **HELP** in pritisnemo Enter. Za pomoč pri določenem ukazu, napišemo **HELP**, presledek in ime ukaza. Spodnja slika prikazuje uporabo FTP protokola iz ukazne vrstice.



```
C:\WINDOWS\system32\cmd.exe
F:\Diploma\SERVER>c:
C:\>FTP -i -s:F:\Diploma\SERVER\lolek.txt
ftp> OPEN 1.0.2.100
Povezan z 1.0.2.100.
220 Ready
331 Password required

230 Logged in
ftp> put F:\Diploma\SERVER\MPSImg.bin
200 Ok
150 Transferring data...
#####
226 Transfer Complete
ftp: poslanih 6037 B v 0,00s 6037000,00KB/s.
ftp> bye
221 Bye

C:\>pause
Če želite nadaljevati, pritisnite poljubno tipko . . .
```

Slika 2.15: Prikaz komunikacije s FTP strežnikom

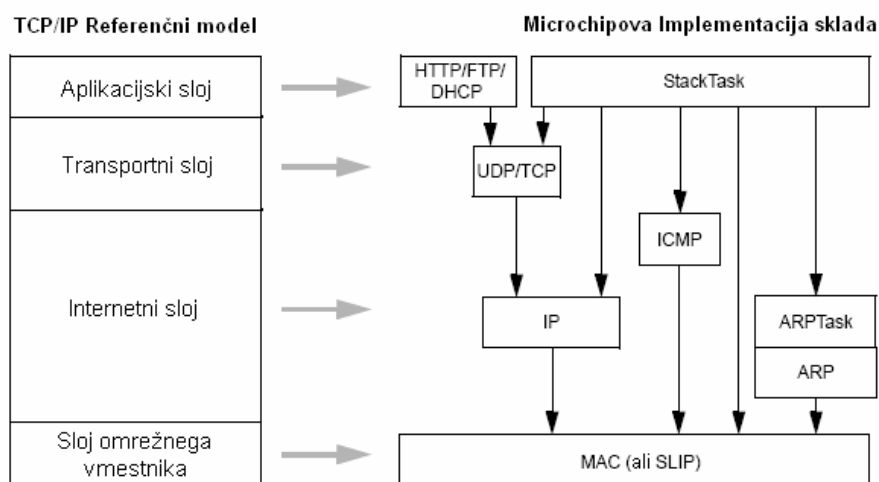
3 MICROCHIP TCP/IP SKLAD

Microchipov TCP/IP sklad je pripomoček, ki uporabnikom omogoča gradnjo najrazličnejših aplikacij, ki temeljijo na TCP/IP protokolu. Najzanimivejše za programerja je modularna organizacija sklada, ki omogoča enostavno programiranje najzahtevnejših aplikacij. Zanimivo je, da potencialnim uporabnikom ni potrebno poznavanje celotne specifikacije TCP/IP protokola. Vse kar potrebujemo za izdelavo aplikacije je dobra ideja, ostale podatke dobimo v specifikaciji Microchipovega TCP/IP sklada, kjer so razložene funkcije in primeri z njimi [11 str. 1].

Programski sklad je napisan v programskem jeziku C, namenjen obema Microchipovima prevajalnikoma (C 18 in Hi-Tech PICC18). Odvisno od uporabe prevajalnika se izvirne datoteke avtomatično prilagodijo. Microchipov TCP/IP sklad je namenjen izključno za uporabo na Microchipovi družini PIC18 mikrokontrolerov.

3.1 Sloji sklada

Kot TCP/IP referenčni model je tudi Microchipov TCP/IP sklad sestavljen iz več slojev (slika spodaj). Programske kode, ki implementirajo določene sloje, so vsebovane v ločenih izvornih datotekah [11 str. 2].



Slika 3.1: Microchipov TCP/IP sklad [11 str. 2]

Z razliko od TCP/IP referenčnega modela, mnogo slojev v Microchipovem TCP/IP skladu neposredno dostopa do slojev, ki niso neposredno pod njimi.

Glavna razlika med Microchipovim in referenčnim TCP/IP skladom sta dodatna modula: »StackTask« in »ARPTask«. StackTask upravlja operacije v skladu in vse njihove module, medtem ko ARPTask upravlja opravila Address Resolution Protokola (ARP sloja).

Microchipov TCP/IP sklad uporablja tehniko imenovano kooperativna večopravilnost (ang. cooperative multitasking). Zanj je značilno, da morajo biti procesi med seboj dosledni in morajo sami prepuščati CPU¹ ostalim procesom. V večopravilnem sistemu se izvaja več nalog, ki so med seboj neodvisne. Slabost takega sistema je v tem, da če en proces predolgo zaseda procesor, se izvajanje ostalih procesov ustrezno upočasni ali celo zablokira. V vsakem primeru mora aplikacija, ki uporablja Microchipov TCP/IP sklad upoštevati pravila kooperativne večopravilnosti. Pripomnimo, da Microchipov TCP/IP sklad ne vsebuje vseh modulov (protokolov), ki jih običajno vsebuje TCP/IP sklad. Čeprav ti moduli niso prisotni, se jih lahko implementira kot ločene module.

3.2 Konfiguracija sklada

Večopravilnost omogoča uporabnikovi glavni aplikaciji izvrševati svoja opravila brez upravljanja TCP/IP sklada. Za doseganje te funkcionalnosti je potrebno, da vse aplikacije, ki uporabljajo TCP/IP sklad, upoštevajo pravila, ki jih določa večopravilnost. Poleg tega mora uporabnik razumeti nekaj osnov konfiguracije sklada [11 str. 3]. Za poenostavitev konfiguracije sklada uporabljamo C-jevske definicije. Za omogočanje, onemogočanje ali postavitve določenega parametra uporabnik spremeni enega ali več teh definicij (*#define xxx*). Večina jih vsebuje knjižnica »StackTask.h«.

¹ CPU Centralno procesna enota (ang. Central processing unit)

3.3 Uporaba sklada

Sam Microchipov TCP/IP sklad vsebuje nekaj že napisanih projektov za različne konfiguracije. Na proizvajalčevem dokumentu [11 str. 5] so opisani vsi projekti, ki jih sklad vsebuje ter katere module uporablja.

Microchipovega TCP/IP sklad uporabimo v aplikaciji, kot je prikazano spodaj. Pozorni moramo biti na to, da aplikacija ne vsebuje opravil, ki bi predolgo zasedala procesor, saj bi se lahko izvajanje programa upočasnilo ali pa celo podrla.

```
#define THIS_IS_STACK_APPLICATION
#include "StackTsk.h"
#include "Tick.h"
#include "dhcp.h" // Samo če uporabimo DHCP.
#include "http.h" // Samo če uporabimo HTTP.
#include "ftp.h" // Samo če uporabimo FTP.
// Ostale knjižnice
...
// Vstop v glavni program
void main(void)
{
    // Inicializacija uporabnikovih komponent
    ...

    // Inicializacija komponent sklada

    TickInit();
    StackInit();
    HTTPInit(); // Samo če uporabimo HTTP.
    FTPInit(); // Samo če uporabimo FTP.
    // Vstop v neskončno zanko
    while(1)
    {
        // Posodobitev števca tick.
        TickUpdate();
        // Dovolimo koordinatorju sklada izvrševanje naloge.
        StackTask();
        // Dovolimo modulom sklada izvršitev naloge.
        HTTPServer(); // Samo če uporabimo HTTP.
        FTPServer(); // Samo če uporabimo FTP.
        // Tukaj vnesemo svojo kodo
        DoAppSpecificTask();
    }
}
```

3.4 Microchipov HTTP Strežnik

Kar naredi sklad zanimiv in privlačen za uporabnike pa je gotovo prikazovanje in spreminjanje vhodov mikrokrmilnika prek svetovnega spleta v realnem času (ang. real-time). To omogoča Microchipov HTTP strežnik [11 str. 77], ki ni polno funkcionalen glede na HTTP specifikacije, vendar je prilagojen z delo na vgradnih sistemih. V naslednjih nekaj vrsticah bomo nakazali postopek, ki je potreben, da projekt pripravimo za delo z Microchipovim HTTP strežnikom:

Potrebno je:

1. definirati `STACK_USE_HTTP_SERVER` v knjižnici »StackTask.h«,
2. izbrati želeno `MAX_HTTP_CONNECTIONS` (maksimalno število HTTP povezav); prav tako v knjižnici »StackTask.h«,
3. vključiti izvorni datoteki »http.c« in »mpfs.c« v projekt,
4. odvisno od lokacije shranitve internetne strani je potrebno definirati `MPFS_USE_PGRM` (če je vsebina shranjena v notranjem pomnilniku) oz. `MPFS_USE_EEPROM` (še je vsebina shranjena v zunanjem EEPROM-u);
če uporabljamo zunanji EEPROM za shranjevanje vsebine je potrebno vključiti tudi »xeprom.c«,
5. dopolniti funkcijo `main()` tako, da bo vključevala HTTP strežnik.

Preden lahko prikazujemo internetne strani, jih je potrebno spremeniti v format, ki ga sklad podpira [11 str. 77]. Strežnik ima nastavljeno za privzeto internetno stran datoteko z imenom »index.html«. To pomeni, da morajo vse aplikacije vsebovati datoteko s tem imenom, če to ni drugače definirano v izvorni datoteki »http.c«; `HTTP_DEFAULT_FILE_STRING`. Za pravilno delovanje je pomembno, da strani ne vsebujejo določenih ne-alfanumeričnih znakov, kot so: (',<,>,#,%,[,],{,},|,\,^,~)! Kot smo že povedali, je HTTP strežnik omejen na delo z vgradnimi sistemi, zato ne podpira vseh datotek, ampak le (privzete): ».txt«, ».htm«, ».gif«, ».cgi«, ».jpg«, ».cla« in ».wav«. Če želimo uporabljati tudi datoteke drugih tipov, je potrebno spremeniti tabelo »httpFiles« v datoteki »http.c«

3.4.1 Generiranje dinamičnih HTTP strani

HTTP strežnik lahko dinamično spreminja vsebino internetne strani in prikazuje informacije, kot so vhodi/izhodi mikrokrmilnika. Za doseg tega cilja uporablja prilagojen CGI (ang. common gateway interface) standard. To je standard, za povezovanje zunanjih aplikacij z informacijskimi strežniki. Za prikazovanje dinamičnih sprememb je potrebno generirati potrebne CGI datoteke, to so datoteke s končnico (*.cgi). CGI datoteke generiramo tako, da vsebujejo znakovni niz '%xx', kjer simbol '%' služi kot kontrola, 'xx' pa predstavlja dvoštevlično identifikacijsko številko spremenljivke. Spremenljivke lahko zavzamejo vrednost med 00 - 99. Ko HTTP strežnik obdeluje tak znakovni niz, odstrani simbol '%' in kliče funkcijo HTTPGetVar. Če je potrebno, da prikazujemo simbol '%', moramo generirati CGI datoteko, ki vsebuje niz 'xx%%' [11 str. 78].

Primer:

Predpostavimo, da stran Status.cgi vsebuje sledečo HTML kodo:

```
...  
<td>S3=%04</td><td>D6=%01</td><td>D5=%00</td>  
...
```

Med izvajanjem datoteke, HTML dobi v obdelavo niz '%04', kar pomeni zahtevo za klic funkcije HTTPGetVar(4, HTTP_START_OF_VAR, &value). Uporabnik implementira funkcijo HTTPGetVar na sledeči način [11 str. 79]:

```
WORD HTTPGetVar(BYTE var, WORD ref, BYTE *val)  
{  
    // Identifikacija spremenljivke  
    // Je morda RB5 ?  
    if ( var == 4 )  
    {  
        // Če je RB5 v visokem stanju vrnemo '1'  
        // oz. '0' če je nizka  
        if ( PORTBbits.RB5 )  
            *val = '1';  
        else  
            *val = '0';  
        // Povemo http-ju da je to zadnji zlog  
        // spremenljivke  
        return HTTP_END_OF_VAR;  
    }  
    else  
        // Preverimo ostale spremenljivke  
        ...  
}
```

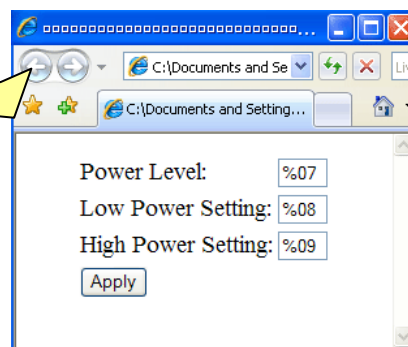
Funkcija preveri stanje vrat RB5 in v primeru visokega nivoja se na HTML strani prikaže vrednost '1' oziroma '0'.

3.4.2 HTTPExecCmd

V prejšnjem primeru smo spoznali funkcijo HTTPGetVar, s pomočjo katere preberemo vrednost določene spremenljivke za prikaz na strani [11 str. 81]. Sedaj pa bi radi neko zahtevo poslali v mikrokrmilnik in jo procesirali. Spodaj je prikazan postopek pisanja HTML kode, ki se shrani v tekstovno datoteko s končnico '.html'. Na spodnji sliki je prikazan dokument, ki ga odpremo s pomočjo internetnega brskalnika [11 str. 82].

Primer HTML
kode za izgradno
dinamične strani

```
<html>
<body><center>
<FORM METHOD=GET action=Power.cgi>
<table>
<tr><td>Power Level:</td>
<td><input type=text size=2 maxlength=1 name=P value=%07></td></tr>
<tr><td>Low Power Setting:</td>
<td><input type=text size=2 maxlength=1 name=L value=%08></td></tr>
<tr><td>High Power Setting:</td>
<td><input type=text size=2 maxlength=1 name=H value=%09></td></tr>
<tr><td><input type=submit name=B value=Apply></td></tr>
</table>
</form>
</body></html>
```



Predpostavimo, da uporabnik v okenska Power Level, Power Settings in High Power Settings vnese vrednosti '5', Low '1' in '9'. Ob kliku na gumb 'Apply' brskalnik generira zahtevo v obliki niza "Power.cgi?P=5&L=1&H=9" in jo pošlje HTML strežniku. Strežnik kliče funkcijo HTTPExecCmd s sledečimi parametri:

```
argv[0] = "Power.cgi", argv[1] = "P", argv[2] = "5", argv[3] = "L", argv[4] = "1", argv[5] = "H", argv[6] = "9"
argc = 7
```


Funkcijo HTTPExecCmd implementiramo sledeče[11 str. 82]:

```
void HTTPExecCmd(BYTE *argv, BYTE argc)
{
    BYTE i;
    // Preverimo vsak argument funkcije
    // Preskočimo ime datoteke argv[0] zato začnemo z indeksom 1
    for (i = 1; i < argc; i++)
    {
        // identficiramo parameter.
        if ( argv[i][0] == 'P' ) // Je to power level?
        {
            // Naslednji parameter vsebuje vrednost spremenljivke Power level.
            PowerLevel = atoi(argv[++i]);
        }
        else if ( argv[i][0] == 'L' ) // Je to Low Power ?
            LowPowerSetting = atoi(argv[++i]);
        else if ( argv[i][0] == 'H' ) // Je to High Power ?
            HighPowerSetting = atoi(argv[++i]);
    }
}
```

Funkcija je napisana tako, da v spremenljivke PowerLevel, LowPowerSetting in HighPowerSetting shrani vrednosti ki smo jih vpisali v brskalniku.

Tako generirana stran ob zahtevi za branje prekorači maksimalno število argumentov, ki je določena s MAX_HTTP_ARGS (v »http.c«); privzeta vrednost je nastavljena na 5, vendar v našem primeru potrebujemo 7.

3.5 Prevajalnik MPFS

Ko enkrat generiramo vse potrebne datoteke, ki bi jih radi prikazovali na internetu, jih je potrebno prevesti v obliko, ki bo razumljiva HTTP strežniku. To storimo s pomočjo priloženega MPFS prevajalnika. MPFS je prevajalnik, ki ga poženemo iz ukazne vrstice s pripadajočimi argument [11 str. 84]i:

```
mpfs [/?] [/c] [/b] [/r<Block>] <InputDir> <OutputFile>
```

Kjer posamezni argumenti pomenijo:

/? prikaže pomoč,

/c generira datoteko tipa 'C' ,

/b generira binarno datoteko (privzeta nastavitve),

/r nastavimo blok rezerviranih zlogov (privzeto 32),

<InputDir> pod do mape ki vsebuje datoteke, ki jih je potrebno prevesti,

<OutputFile> izhodna datoteka, ki jo dobimo.

3.6 Microchipov FTP strežnik

Microchipov FTP strežnik poskrbi za prenos datotek, ki bi jih radi prikazovali, na strežnik. Prav tako kot HTTP strežnik tudi FTP nima polne funkcionalnosti, ki jo narekujejo standardi, saj je ta namenjen vgradnim sistemom. Da bi implementirali FTP strežnik v aplikacijo je potrebno storiti sledeče [11 str. 85]:

Potrebno je:

1. definirati `STACK_USE_FTP_SERVER` v knjižnici »StackTsk.h« ,
2. po potrebi povečati število vtičnic, ki je nastavljeno na 2,
3. vključiti izvorni datoteki »http.c« in »mpfs.c« v projekt,
4. odvisno od lokacije shranitve internetne strani je potrebno definirati `MPFS_USE_PGRM` (če je vsebina shranjena v notranjem pomnilniku) oz. `MPFS_USE_EEPROM` (še je vsebina shranjena v zunanjem EEPROM-u);

če uporabljamo zunanji EEPROM za shranjevanje vsebine je potrebno vključiti tudi »xeprom.c«,

5. dopolniti funkcijo `main()` tako, da bo vključevala FTP strežnik.

Spodnja slika prikazuje potek prenašanja datoteke na strežnik:

```
c:\ ftp 10.10.5.15
220 ready
User (10.10.5.15: (none)): ftp
331 Password required
Password: microchip
230 Logged in
ftp> put mpfsimg.bin
200 ok
150 Transferring data...
226 Transfer Complete
ftp> 16212 bytes transferred in
0.018seconds 16212000.00Kbytes/sec.
ftp> quit
221 Bye
```

Slika 3.2: Prenos datotek na strežnik

1. Z ukazom *ftp 10.10.5.15* se povežemo na zelen strežnik (kjer je 10.10.5.15 IP naslov strežnika),
2. vnesemo ustrezno uporabniško ime in geslo. Če je prijava uspešna, dobimo odgovor Logged in,
3. ukaz *put mpfsimg.bin* prenesemo želeno datoteko na strežnik. Pomembno je opozoriti, da ob prijavi na strežnik mora pot v ukazni vrstici kazati na mesto kjer imamo shranjeno želeno datoteko.

4 MERJENJE TEMPERATURE

Tržišče nam ponuja širok spekter najrazličnejših temperaturnih senzorjev, od preprostih analognih do naprednejših, ki imajo digitalni izhod. Kateri senzor je ustrezen za dani primer uporabe, je odvisno od konkretnih zahtev. S tem mislimo na občutljivost senzorja, izhodno karakteristiko, uporabljeno opremo, ... ter seveda ceno. Naša aplikacija temelji na uporabi mikrokrmilnika z vgrajenim A/D pretvornikom in smo zato imeli proste roke pri izbiri senzorja v smislu senzorjevega analognega ali digitalnega izhoda. Cilj, ki smo si ga zadali, je izdelati termometer, s katerim bi lahko merili temperaturo na čim širšem temperaturnem območju. Ob enem si želimo, da bi bil dovolj natančen, da bi ga lahko uporabljali za regulacijo sobne temperature. Zahteve so sledeče:

- čim širše območje uporabe, vsaj med $+13\text{ }^{\circ}\text{C}$ in $+33\text{ }^{\circ}\text{C}$, kar ustreza sobni temperaturi,
- želena natančnost $\pm 1\text{ }^{\circ}\text{C}$.

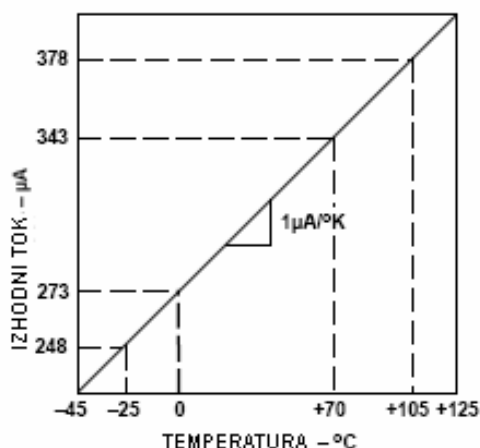
Zaradi primerjave rezultatov smo izdelali dva termometra. Prvi temelji na uporabi NTK upora, drugega pa smo izdelali s pomočjo temperaturnega senzorja AD592CN.

4.1 Merjenje temperature s senzorjem AD592CN

AD592 je analogni senzor proizvajalca ANALOG DEVICES. Izhod senzorja je tok, ki je proporcionalen absolutni temperaturi, odlikuje ga izjemno linearna izhodna karakteristika. Navedimo nekaj njegovih parametrov [9 str. 1]:

- široko območje uporabe: $-25\text{ }^{\circ}\text{C}$ do $+105\text{ }^{\circ}\text{C}$,
- širok razpon napajalne napetost: $+4\text{ V}$ do $+30\text{ V}$,
- izhodni tok $1\text{ }\mu\text{A/K}$,
- zagotovljena začetna natančnost $\pm 0,5\text{ }^{\circ}\text{C}$.

Slika 4.1 prikazuje izhodno karakteristiko senzorja.

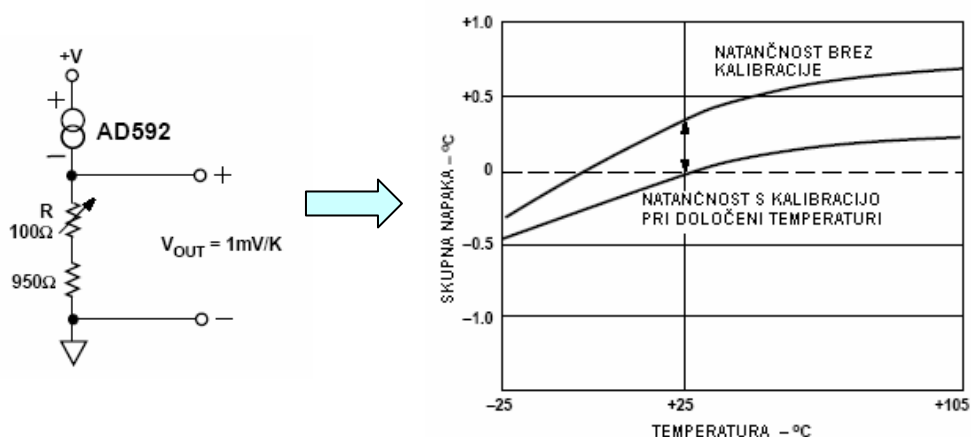


Slika 4.1: Izhodna karakteristika senzorja [9 str. 2]

4.1.1 Postopek merjenja

V proizvajalčevem katalogu [9 str. 6] dobimo vse potrebne informacije in vezalno shemo za izdelavo ustreznega merilnega člena. Osnovna vezalna shema je prikazana na spodnji sliki. S tako izdelanim vezjem lahko izničimo napako pri določeni temperaturi, ki je običajno postavljena v sredino želenega temperaturnega območja. Postopek kalibracije je sledeč: potenciometer R spreminjamo toliko časa, dokler ne dosežemo izhodne napetosti $1\text{ mV}/^{\circ}\text{C}$ (to se zgodi približno takrat, ko ima upor vrednost $R=50\ \Omega$ saj je takrat skupna upornost trimerja in upora $1\text{ k}\Omega$).

Ob uporabi takega vezja je obvezna kalibracija z referenčnim termometrom.



Slika 4.2: Osnovna vezava senzorja (levo) in kalibracija pri določeni temperaturi (desno) [9 str. 5]

Najprej smo izračunali, ali bi lahko s takim vezjem merili temperaturo preko celotnega merilnega območja senzorja s predpisano natančnostjo. V našem primeru imamo mikrokrmilnik z vgrajenim 10 bitnim A/D pretvornikom ter napetostnim območjem vhodne napetosti med 0 in +5 V, iz česar izračunamo nominalno ločljivost AD pretvorbe (LSB²).

$$\text{LSB} = 5 \text{ V} / 1024 = 0,0048 \text{ V}$$

Iz tega izračunamo, kolikšna je najmanjša sprememba temperature, ki jo še zaznamo:

$$\Delta T = \frac{4,88 \text{ mV}}{1 \text{ mV} / ^\circ\text{C}} = 4,88 ^\circ\text{C}$$

Prav tako se je potrebno zavedati, da ja nedoločenost A/D pretvorbe ± 1 LSB, kar doprinese k končnemu rezultatu dodatno napako $\pm 4,88 ^\circ\text{C}$. Iz do sedaj povedanega sklepamo, da je tako vezje neprimerno za merjenje temperature preko celotnega merilnega območja. Poraja se vprašanje, ali se razmere izboljšajo pri omejitvi na ožje temperaturno območje z ustrezno prilagoditvijo območja vhodnih napetosti A/D pretvornika. Če bi tako vezje uporabili za merjenje samo sobne temperature, bi bilo dovolj izbrati temperaturno območje med $+13 ^\circ\text{C}$ in $+33 ^\circ\text{C}$. Iz gornjega grafa odčitamo območje napetosti, ki jih lahko pričakujemo na vhodu A/D pretvornika ob upoštevanju gornjih predpostavk.

Pri temperaturi $13 ^\circ\text{C}$ dobimo na vhod A/D pretvornika 286 mV

Pri temperaturi $33 ^\circ\text{C}$ dobimo na vhod A/D pretvornika 306 mV

Na mikrokrmilnik lahko tako priklopimo $-V_{\text{ref}} = 286 \text{ mV}$ in $+V_{\text{ref}} = 306 \text{ mV}$, iz tega sledi vrednost LSB:

$$\text{LSB} = (306 - 286) \cdot 10^{-3} \text{ V} / 1024 = 19,5 \text{ } \mu\text{V}$$

Iz tega sledi najmanjša sprememba temperature, ki jo še lahko zaznamo:

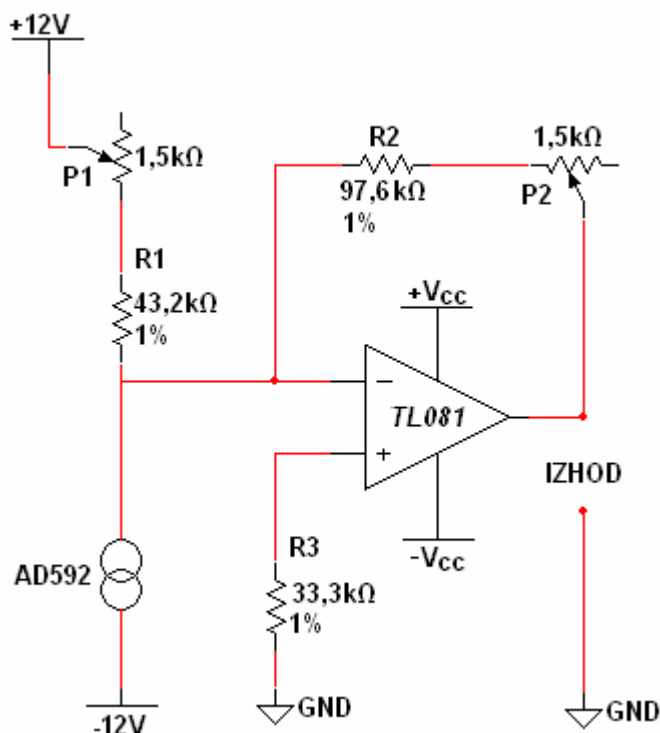
$$\Delta T = \frac{19,5 \text{ } \mu\text{V}}{1 \text{ mV} / ^\circ\text{C}} = 0,02 ^\circ\text{C}$$

S takim vezjem izpolnjujemo zastavljene cilje, vendar je to le teoretični izračun. V praksi bi nas omejevalo vsaj dvoje:

² LSB: bit z najmanjšo utežjo (ang. least significant bit) [5 str. 245]

- območje referenčne napetosti A/D pretvornika ni poljubno; npr. minimalna napetostna diferenca med V_{ref+} in V_{ref-} mora biti vsaj 3 V [7 str. 359],
- težko je zagotoviti natančni referenci napetosti 286 mV in 306 mV.

Da se izognemo navedenim težavam, uporabimo vezje ki je prikazano na spodnji sliki.



Slika 4.3: Merjenje temperature s temperaturnim senzorjem AD592

Operacijski ojačevalnik [32 str. 2] uporabimo, da z ustreznim ojačanjem dobimo drugo (širše) območje napetosti in s tem večjo ločljivost vezja. Enačba, ki opisuje izhodno napetost vezja je sledeča [4 str. 8-19]:

$$U_{izh} = -(R_2 + P_2) \cdot \left(\frac{U_{ref}}{R_1 + P_1} - I_G \right) \quad (4.1)$$

Termometer konfiguriramo tako, da v temperaturnem območju med 0 °C in 50 °C dobimo izhodno napetost med 0 V in 5 V, kar ustreza občutljivosti 100 mV/ °C.

Zagotovljena začetna natančnost je $\pm 0,5\text{ }^{\circ}\text{C}$ [9 str. 1], kar je pretvorjeno v tok $\pm 0,5\text{ }\mu\text{A}$. Kalibracijo vezja izvedemo v dveh korakih:

Kompenzacija pri $0\text{ }^{\circ}\text{C}$

Kompenzacijo pri $0\text{ }^{\circ}\text{C}$ smo izvedli z nastavljanjem potenciometra P_1 . Potenciometer P_1 spreminjamo toliko časa, dokler na izhodu operacijskega ojačevalnika ne dobimo 0 V . Pri idealnem operacijskem ojačevalniku se to zgodi, ko velja $I_G = I_{R1}$, sicer z nastavitvijo izničimo enosmerne napake, ki jih neidealnosti prinesejo.

Iz tega izračunamo upor R'_1 (skupna upornost potenciometra in upora):

$$R'_1 = R_1 + P_1 = U_{\text{ref}} / I_G = 12\text{ V} / 273\text{ }\mu\text{A} = 43,96\text{ k}\Omega \rightarrow (R_1 = 43,2\text{ k}\Omega \pm 1\%)$$

Izbrali bi lahko upor s poljubno toleranco s tem, da moramo z uporom in potenciometrom skupaj pokriti celoten interval upornosti, ki je posledica kalibracijskega pogreška.

Ob upoštevanju kalibracijskega pogreška ($\pm 0,5\text{ }^{\circ}\text{C}$ oz. $\pm 0,5\text{ }\mu\text{A}$) dobimo:

$$I_G = I_G \pm 0,5\text{ }\mu\text{A}$$

$$R'_1 = R_1 + P_1 = U_{\text{ref}} / I_{G1} = 12\text{ V} / 272,5\text{ }\mu\text{A} = 44,04\text{ k}\Omega$$

$$R'_1 = R_1 + P_1 = U_{\text{ref}} / I_{G2} = 12\text{ V} / 273,5\text{ }\mu\text{A} = 43,87\text{ k}\Omega$$

Zaradi kalibracijskega pogreška dobimo območje upornosti, ki ga moramo pokrivati z uporom in potenciometrom.

$$43,87\text{ k}\Omega \leq R'_1 \leq 44,04\text{ k}\Omega, R'_1 \text{ dobimo v razponu } 170\text{ }\Omega.$$

Potenciometer izberemo $P_1 > 1,3\text{ k}\Omega = 1,5\text{ k}\Omega$ (s tako izbranim potenciometrom lahko pokrijemo vse vrednosti med $43,87\text{ k}\Omega$ in $44,04\text{ k}\Omega$).

Kompenzacija pri $50\text{ }^{\circ}\text{C}$

Pri $50\text{ }^{\circ}\text{C}$ želimo dobiti izhodno napetost $U_{\text{izh}} = 5\text{ V}$:

pri $50\text{ }^{\circ}\text{C} = 323\text{ K}$ dobimo tok senzorja $I_G = 323\text{ }\mu\text{A}$,

$I_- = I_{R1} - I_G = 273\text{ }\mu\text{A} - 323\text{ }\mu\text{A} = -50\text{ }\mu\text{A}$, torej teče tok $50\text{ }\mu\text{A}$ v senzor (čez potenciometer P_2 in upor R_2).

Ker poznamo tok in napetost, lahko določimo upornost:

$$R'_2 = R_2 + P_2 = U_{\text{izh}} / I_- = 5\text{ V} / 50\text{ }\mu\text{A} = 100\text{ k}\Omega \rightarrow (R_2 = 97,6\text{ k}\Omega \pm 1\%).$$

Izbrali bi lahko upor s poljubno toleranco s tem, da moramo z uporom in potenciometrom pokriti celoten interval upornosti, ki je posledica nelinearnosti senzorja.

Ob upoštevanju nelinearnosti ($\pm 0,15 \text{ }^{\circ}\text{C}$ oz. $\pm 0,15 \text{ }\mu\text{A}$), dobimo:

$$R'_2 = R_2 + P_2 = U_{\text{izh}}/I_- = 5 \text{ V} / 50,15 \text{ }\mu\text{A} = 99,7 \text{ k}\Omega.$$

$$R'_2 = R_2 + P_2 = U_{\text{izh}}/I_- = 5 \text{ V} / 49,85 \text{ }\mu\text{A} = 100,3 \text{ k}\Omega.$$

$99,7 \text{ k}\Omega \leq R'_2 \leq 100,3 \text{ k}\Omega$; R'_2 dobimo v razponu $600 \text{ }\Omega$.

Iz tega sledi $P_2 > 3,68 \text{ k}\Omega = 4,7 \text{ k}\Omega$ (s tako izbranim potenciometrom lahko pokrijemo vse vrednosti med $99,7 \text{ k}\Omega$ in $100,3 \text{ k}\Omega$).

Upornost R_3 izberemo na sledeč način:

$$R_3 = R'_1 || R'_2 = 29,9 \text{ k}\Omega ; R_3 = 30 \text{ k}\Omega$$

Ob tako izbranih uporih preverimo, kakšno napako vnesemo v rezultat zaradi tolerance uporov, temperaturnega lezenja, spremembe napajalne napetosti in napake senzorja. Izračunamo skupno standardno deviacijo (M_{Uout}) izhodne napetosti po sklopu enačbi 4.2.

Izračun je sledeč [28 str. 65]:

$$M_{\text{Uout}} = \sqrt{\left[\left(\frac{\delta U_{\text{out}}}{\delta R_1} \cdot M_{R1} \right)^2 + \left(\frac{\delta U_{\text{out}}}{\delta P_1} \cdot M_{P1} \right)^2 + \left(\frac{\delta U_{\text{out}}}{\delta R_2} \cdot M_{R2} \right)^2 + \left(\frac{\delta U_{\text{out}}}{\delta P_2} \cdot M_{P2} \right)^2 + \left(\frac{\delta U_{\text{out}}}{\delta U_{\text{ref}}} \cdot M_{U_{\text{ref}}} \right)^2 + \left(\frac{\delta U_{\text{out}}}{\delta T} \cdot M_T \right)^2 \right]}$$

$$\frac{\delta U_{\text{out}}}{\delta R_1} = \frac{\delta U_{\text{out}}}{\delta P_1} = (R_2 + P_2) \cdot \frac{U_{\text{ref}}}{(R_1 + P_1)^2} = 614,4 \cdot 10^{-6} \text{ }\mu\text{A},$$

$$\frac{\delta U_{\text{out}}}{\delta R_2} = \frac{\delta U_{\text{out}}}{\delta P_2} = -\left(\frac{U_{\text{ref}}}{R_1 + P_1} - I_G \right) = -4,54 \cdot 10^{-6} \text{ }\mu\text{A}, \quad (4.2)$$

$$\frac{\delta U_{\text{out}}}{\delta U_{\text{ref}}} = -\frac{R_2 + P_2}{R_1 + P_1} = -2,29,$$

$$\frac{\delta U_{\text{out}}}{\delta T} = (R_2 + P_2) \cdot \frac{\partial I_G}{\partial T} = 0,1 \text{ V/K}$$

Spodaj izračunamo mejne vrednosti posameznih elementov, ki so posledica temperaturnega lezenja. Temperaturno lezenje je podano v delih na milijon (ang. part per million; ppm). V našem primeru je termometer dimenzioniran tako, da lahko meri temperaturo med $0 \text{ }^{\circ}\text{C}$ in $+50 \text{ }^{\circ}\text{C}$. Ker ga z našo aplikacijo uporabljamo za merjenje sobne temperature med $+13 \text{ }^{\circ}\text{C}$ in $+33 \text{ }^{\circ}\text{C}$ smo zato upoštevali napako temperaturnega lezenja uporov samo na tem območju. Povprečna sobna temperature je nekje $+23 \text{ }^{\circ}\text{C}$, tako se temperatura lahko spremeni za $\pm 10 \text{ }^{\circ}\text{C}$.

V spodnji enačbi smo zato napako zaradi temperaturnega lezenja uporov množimo z 10 K.

Mejna vrednost napajalne izhaja iz nihanja napajalne napetosti, ki smo jo približno ocenili z osciloskopom. Mejna vrednost M_T je nelinearnost senzorja.

Izračun je sledeč:

$$\begin{aligned} M_{R1} &= R_1 \cdot 50 \text{ ppm/K} \cdot 10 \text{ K} = 108 \, \Omega \quad , \quad M_{R2} = P_2 \cdot 50 \text{ ppm/K} \cdot 10 \text{ K} = 244 \, \Omega \\ M_{P1} &= P_1 \cdot 50 \text{ ppm/K} \cdot 10 \text{ K} = 9,4 \, \Omega \quad , \quad M_{P2} = P_2 \cdot 250 \text{ ppm/K} \cdot 10 \text{ K} = 49,4 \, \Omega \quad , \quad M_{U_{\text{ref}}} < 0,01 \text{ V} \quad , \quad M_T = \pm 0,15^\circ \text{C} \\ \Rightarrow M_{U_{\text{out1}}} &= \underline{0,072 \text{ V}} \end{aligned}$$

Iz gornjih enačb lahko sklepamo, da je za večjo natančnost meritve potrebno izbirati preciznejše elemente tiste, katerih odvod izhodne napetosti po določeni spremenljivki zavzame največjo vrednost. S kalibracijo vezja pri spodnji in zgornji meji želenega merilnega območja se znebimo vpliva toleranc uporov in enosmernih napak operacijskega ojačevalnika. Vplivov zaradi temperaturnega lezenja ne moremo odpraviti, zato izberemo upore s čim manjšim faktorjem temperaturnega lezenja (ppm/K).

Skupna napaka pri tako izbranih elementih znaša 72 mV oziroma $\pm 0,72^\circ \text{C}$. Če bi izbrali precizijski upor R_1 s faktorjem lezenja $\pm 25 \text{ ppm/K}$, bi se napaka znižala na $\pm 0,32^\circ \text{C}$. Tu moramo upoštevati še napako, ki jo vneseta operacijski ojačevalnik zaradi lezenja napetostnega premika in napaka A/D pretvornika:

$$M_{U_{\text{out}}} = M_{U_{\text{out1}}} + \left(1 + \frac{R_2'}{R_1} \right) \cdot u_{\text{off}} + \text{LSB} = 72 \text{ mV} + 42,9 \text{ mV} + 4,9 \text{ mV} = \underline{\underline{119,8 \text{ mV} \quad \text{oz.} \quad \mp 1,20^\circ \text{C}}}$$

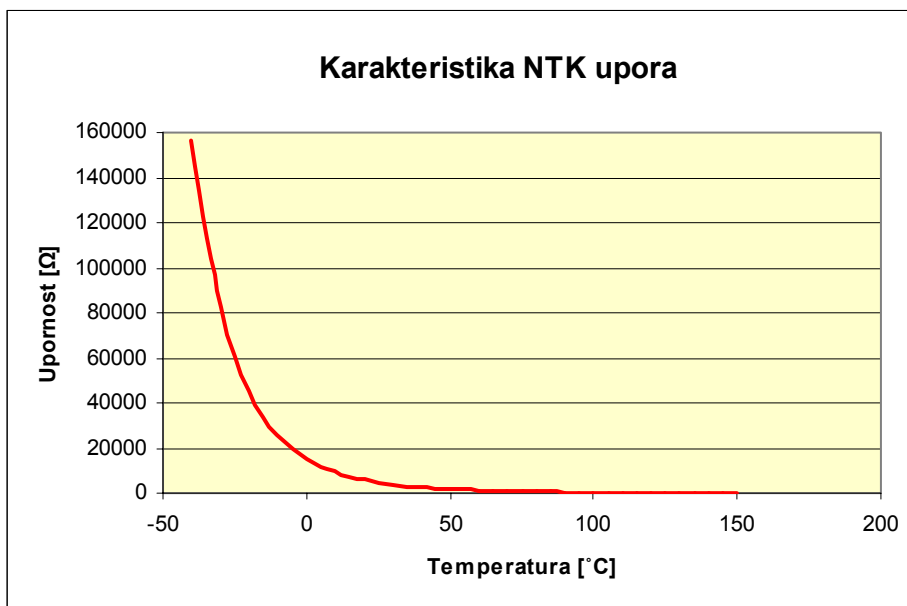
Iz rezultata je razvidno, da lezenje napetostnega premika operacijskega ojačevalnika doprinese znaten pogrešek meritve. Za doseganje večje natančnosti in za izpolnitev zahtev bi morali uporabiti operacijski ojačevalnik, ki bi imel lezenje napetostnega premika reda μV .

4.2 Merjenje Temperature z uporabo termistorja

Termistorje (nelinearne upore, katerih upornost se spreminja v odvisnosti od temperature) delimo glede na njihov temperaturni koeficient upornosti (TK_r) na:

- NTK upore (z negativnim TK_r),
- PTK upore (s pozitivnim TK_r).

Na spodnji sliki je prikazana karakteristika NTK upora vrednosti 4,7 k Ω . Iz slike je razvidno, da je sprememba upornosti pri porastu temperature izrazito nelinearna [24].



Graf 1: Karakteristika NTK upora

Tipična vrednost TK_r znaša od -0,01/K do -0,07/K, kar je približno 10-krat več od TK_r kovin. Materiali, ki se uporabljajo za izdelavo NTK uporov, so zmesi kovinskih oksidov, torej dobri izolatorji, ki jih z dodatkom ustreznih atomov primesi pretvorimo v polprevodniške keramike. Uporabili smo upor proizvajalca BCcomponents serije 2322 640 6... z upornostjo 4,7 k Ω in 5 % toleranco.

V nadaljevanju nakazujemo, kako smo izpeljali enačbe, ki jih uporabljamo za izračun temperature. Prvi pomemben podatek NTK upora je enačba, ki opisuje spreminjanje upornosti v odvisnosti od temperature, in spremembo temperature zaradi spremembe upornosti. Ti dve enačbi sta sledeči [10 str. 6]:

$$R(T) = R_{\text{ref}} \times e^{A+B/T+C/T^2+D/T^3} \quad (4.3)$$

in

$$T(R) = \left(A_1 + B_1 \ln \frac{R}{R_{\text{ref}}} + C_1 \ln^2 \frac{R}{R_{\text{ref}}} + D_1 \ln^3 \frac{R}{R_{\text{ref}}} \right)^{-1} \quad (4.4)$$

kjer je R_{ref} upornost pri referenčni temperaturi (v tem primeru 25 °C), medtem ko so A , B , C , D , A_1 , B_1 , C_1 in D_1 konstante, ki jih dobimo iz tabele 1, glede na vrednost B_{25}/B_{85} , ki je podana v proizvajalčevem katalogu. Razmerje B_{25}/B_{85} pomeni materialno konstanto NTK upora, ki jo proizvajalci pogosto podajajo preko neposredno izmerjenih upornosti NTK upora pri dveh temperaturah (običajno pri $T_1 = 25$ °C in $T_2 = 85$ °C). Enačba, ki opisuje izračun konstante je sledeča:

$$B_{25}/B_{85} = \frac{\ln \frac{R_1}{R_2}}{\frac{1}{T_1} - \frac{1}{T_2}} = \frac{\ln \frac{4700 \Omega}{503 \Omega}}{\frac{1}{298 \text{ K}} - \frac{1}{358 \text{ K}}} = 3977 \text{ K}$$

kjer oznake pomenijo:

- $R_1(T_1 = 25$ °C) – upornost termistorja pri temperaturi 25 °C,
- $R_1(T_1 = 85$ °C) – upornost termistorja pri temperaturi 85 °C,
- temperaturi T_1 in T_2 sta absolutni temperaturi.

S tako izračunano konstanto in podatki iz proizvajalčevega kataloga (tabela 1) najdemo vse konstante, ki jih rabimo za izračun upornosti in temperature iz zgornjih dveh enačb.

B _{25/85} -VALUE (K)	A	B (K)	C (10 ⁵ K ²)	D (10 ⁶ K ³)	A ₁ (10 ⁻³)	B ₁ (10 ⁻⁴ K ⁻¹)	C ₁ (10 ⁻⁶ K ⁻²)	D ₁ (10 ⁻⁷ K ⁻³)
2880	-9.094	2251.74	229098	-27.4482	3.354016	3.495020	2.095959	4.260615
2990	-10.2296	2887.62	132336	-25.0251	3.354016	3.415560	4.955455	4.364236
3041	-11.1334	3658.73	-102895	0.516652	3.354016	3.349290	3.683843	7.050455
3136	-12.4493	4702.74	-402687	31.96830	3.354016	3.243880	2.658012	-2.70156
3390	-12.6814	4391.97	-232807	15.09643	3.354016	2.993410	2.135133	-8.05672
3528	-12.060	3.688	-0.076	-5.915	3.354016	2.909670	1.632136	0.719220
3528	-21.095	11.930	-25.139	248.120	3.354016	2.933908	3.494314	-7.71269
3560	-13.072	4.191	-0.472	-11.993	3.354016	2.884193	4.118032	1.786790
3740	-13.897	4.558	-0.983	-7.522	3.354016	2.744032	3.666944	1.375492
3977	-14.634	4.792	-1.153	-3.731	3.354016	2.569355	2.626311	0.675278
4090	-15.532	5.230	-1.605	-5.414	3.354016	2.519107	3.510939	1.105179
4190	-16.035	5.459	-1.911	-3.328	3.354016	2.460382	3.405377	1.034240
4370	-16.872	5.759	-1.943	-6.869	3.354016	2.367720	3.585140	1.255349
4570	-17.644	6.023	-2.032	-7.184	3.354016	2.264097	3.278184	1.097628

Tabela 1: Parametri NTK upora [10 str. 7]

Gornji enačbi (4.3 in 4.4) za izračun upornosti in temperature vneseta maksimalno napako 0,005 °C v temperaturnem obsegu med 25 °C in 125 °C in maksimalno napako 0,015 °C v temperaturnem obsegu med -40 °C in +25 °C. Napake so posledica interpolacije podatkov.

V nadaljevanju je prikazano, kako izračunamo odstopanje temperature od nominalne deklarirane vrednosti. Enačba 4.5, ki opisuje to relacijo, je sledeča [10 str. 6]:

$$Z = \left[\left(1 + \frac{X}{100} \right) \times \left(1 + \frac{Y}{100} \right) - 1 \right] \times 100 \% \quad (4.5)$$

kjer oznake pomenijo:

Z – skupno odstopanje,

X – odstopanje zaradi tolerance samega NTK upora (5%),

Y – odstopanje zaradi tolerance materialne konstante termistorja B.

Iz tega sledi temperaturno odstopanje (ΔT) po enačbi 4.6:

TC = temperaturni koeficient (iz proizvajalčevega kataloga),

ΔT = temperaturno odstopanje

$$\Delta T = \frac{Z}{TC} \quad (4.6)$$

Primer:

izračun odstopanja pri temperaturi 0 °C, X=5 %, Y=0,89 % in TC=5,08 %/K. Podatka Y in TC sta navedena v katalogu, ki je priložen.

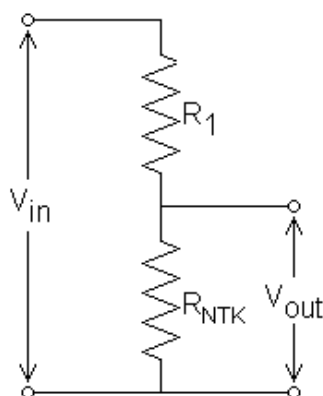
$$\begin{aligned} Z &= \left[\left(1 + \frac{5}{100} \right) \times \left(1 + \frac{0,89}{100} \right) - 1 \right] \times 100 \% \\ &= \{ 1,05 \times 1,0089 \} \times 100 \% = 5,9345 \% \approx (5,93 \%) \\ \Delta T &= \frac{Z}{TC} = \frac{5,93}{5,08} = 1,167^{\circ}\text{C} (\approx 1,17^{\circ}\text{C}) \end{aligned}$$

To nam pove, da napaka zaradi tolerance termistorja pri temperaturi 0 °C ni večja od $\pm 1,17^{\circ}\text{C}$.

4.2.1 Postopek merjenja

Merjenje temperature smo izvedli tako, da smo termistor vezali v napetostni delilnik in merili padec napetosti na njem. Napetostni delilnik smo dimenzionirali tako, da imamo na polovici temperaturnega območja termistorja polovični padec napetosti (oz. delilno razmerje 1/2). Za merjenje napetosti smo uporabili 10-bitni A/D pretvornik z območjem vhodnih napetosti med 0 in 5 V. Najprej smo določili upor v napetostnem delilniku, po sledečem postopku:

- ker je 50 °C približno na polovici merilnega območja NTK upora, smo se odločili, da bo pri tej temperaturi padec napetosti na NTK uporu približno 2,5 V (polovica območja vhodne napetosti A/D pretvornika). Upor izračunamo po enačbi 4.6.



$$\begin{aligned}
 V_{\text{out}} &= \frac{R_{\text{NTK}}}{R_{\text{NTK}} + R_1} \cdot V_{\text{in}} \\
 \frac{V_{\text{out}}}{V_{\text{in}}} &= \frac{R_{\text{NTK}}}{R_{\text{NTK}} + R_1} \quad |^{-1} \\
 \frac{V_{\text{in}}}{V_{\text{out}}} - 1 &= \frac{R_1}{R_{\text{NTK}}} \quad (4.6)
 \end{aligned}$$

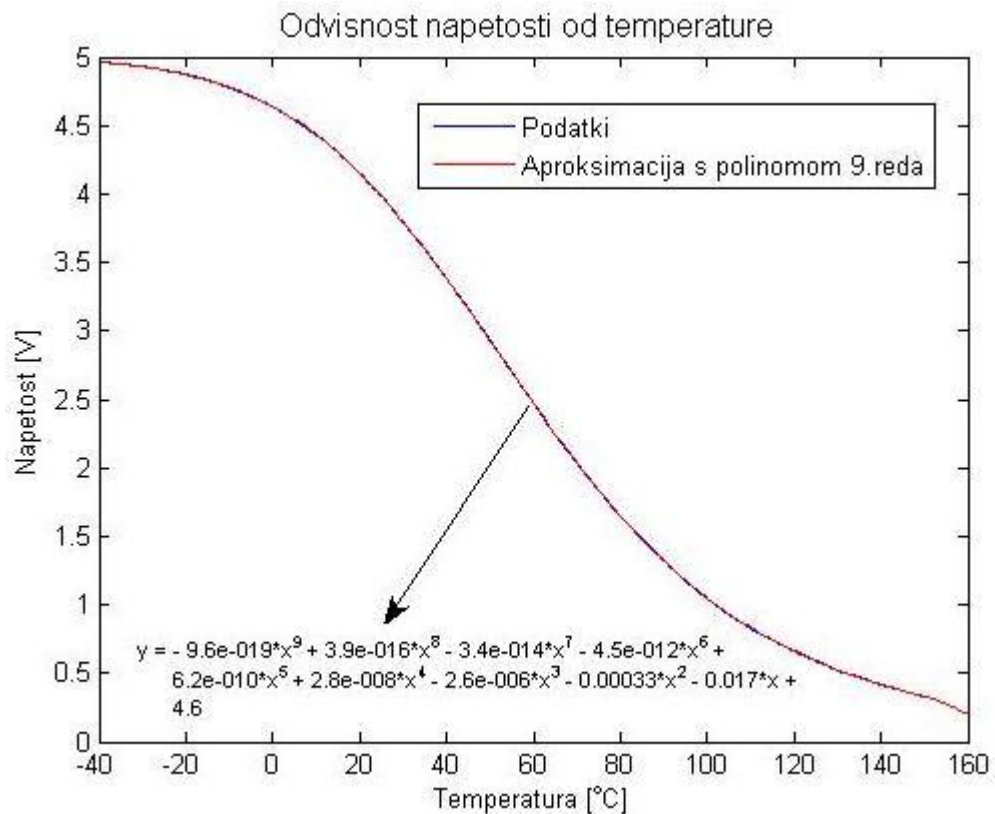
Slika 4.5: Vezava termistorja v napetostni delilnik

Razmerje, ki ga izračunamo je ($R_1 / R_{\text{NTK}} = 1$). To pomeni, da mora imeti upor na sredini temperaturnega območja, ki ga želimo meriti, enako upornost kot jo ima termistor pri tej temperaturi. Upornost dodatnega upora je tako:

$$R_1 = R_{\text{NTK}} = 1190 \, \Omega \approx \underline{1200 \, \Omega}$$

Izbrali smo si $1200 \, \Omega$ upor s 1% toleranco. V spodnjih izračunih bomo prikazali kakšno napako dobimo zaradi tako izbranega upora. S tako dimenzioniranim delilnikom (slika 4.5) merimo temperaturni obseg med $-0 \, ^\circ\text{C}$ in $+150 \, ^\circ\text{C}$, padec napetosti pri temperaturi $+150 \, ^\circ\text{C}$ je $0,33 \, \text{V}$ pri $-40 \, ^\circ\text{C}$ pa približno $4,96 \, \text{V}$.

Na grafu 2 je prikazan potek napetosti na NTK uporu v odvisnosti od temperature, medtem ko dvisnost temperature od padca napetosti prikazuje graf 3. Na vsakem grafu je narisana tudi polinomska interpolacija podatkov ter pripadajoča enačba, ki jih uporabljamo za izračun trenutne in želene temperature. Za aproksimacijo karakteristike termistorja smo uporabili enačbo devetega reda. Red enačbe smo spreminjali in ugotovili, da je deveti red optimalen. Ob nadaljnjem zviševanju reda se napaka interpolacije skoraj ne spreminja. Problem predstavljajo tolerance uporov. Če bi si za aproksimacijo izbrali polinom nižjega reda, bi tako v rezultat vnesli večjo napako in s tem večjo netočnost meritve.



Graf 2: Odkvisnost napetosti od temperature

Spodnja enačba opisuje polinomsko interpolacijo napetosti v odkvisnosti od temperature. Pomen simbolov v spodnji enačbi je sledeč:

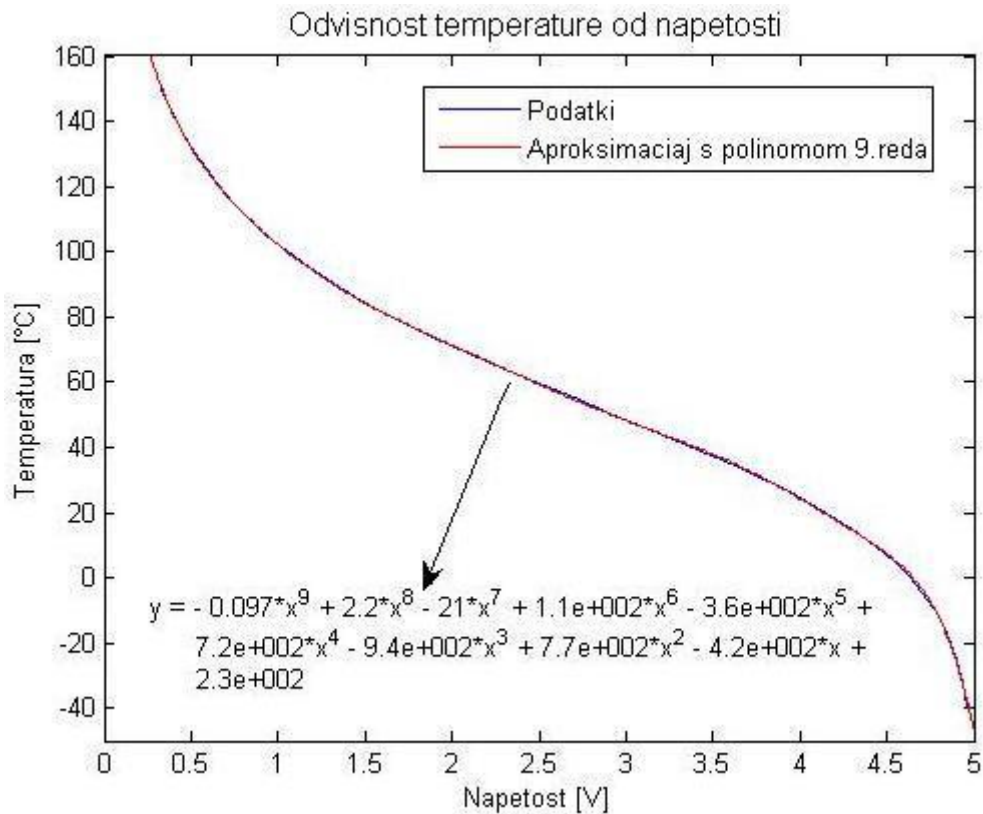
y = napetost

x = temperatura

p_1 do p_{10} = koeficjenti polinoma

$$y = p_1 \cdot x^9 + p_2 \cdot x^8 + p_3 \cdot x^7 + p_4 \cdot x^6 + p_5 \cdot x^5 + p_6 \cdot x^4 + p_7 \cdot x^3 + p_8 \cdot x^2 + p_9 \cdot x + p_{10}$$

$p_1 = -9,5915e-19$
 $p_2 = 3,8926e-16$
 $p_3 = -3,3921e-14$
 $p_4 = -4,5108e-12$
 $p_5 = 6,1542e-10$
 $p_6 = 2,7621e-8$
 $p_7 = -2,5798e-6$
 $p_8 = -325,05e-6$
 $p_9 = -17,037e-3$
 $p_{10} = 4,6362$



Graf 3: Odkvisnost temperature od napetosti

Spodnja enačba opisuje polinomsko interpolacijo temperaturev odkvisnosti od napetosti. Pomen simbolov v spodnji enačbi je sledeč:

y = temperatura

x = napetost

p_1 do p_{10} = koeficjenti polinoma

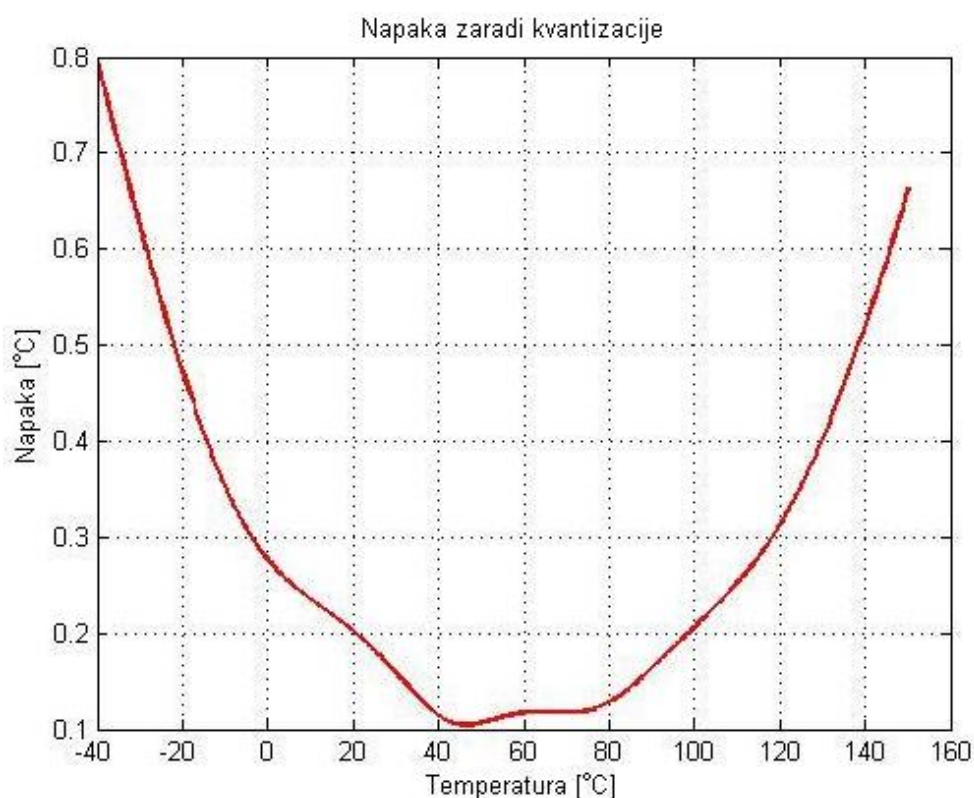
$$y = p_1 \cdot x^9 + p_2 \cdot x^8 + p_3 \cdot x^7 + p_4 \cdot x^6 + p_5 \cdot x^5 + p_6 \cdot x^4 + p_7 \cdot x^3 + p_8 \cdot x^2 + p_9 \cdot x + p_{10}$$

$p_1 = -0.097$
 $p_2 = 2.17$
 $p_3 = -20.79$
 $p_4 = 110.28$
 $p_5 = -356.04$
 $p_6 = 724.84$
 $p_7 = -938.30$
 $p_8 = 771.14$
 $p_9 = -423.56$
 $p_{10} = 232.46$

Ker je karakteristika termistorja nelinearna, se tudi napaka merilnega sistema spreminja v odvisnosti od merjene temperature. Iz grafov 2 in 3 je razvidno, da je v nekaterih območjih naklon krivulje strmejši, zato bo tam napaka zaradi kvantizacije večja oz. v strmejšem delu je potrebna manjša sprememba temperature, da se to odraža v spremeni napetost za 1 LSB. Ker imamo območje vhodne napetosti A/D pretvornika med 0 in 5 V se LSB izračuna:

$$\text{LSB} = 5 \text{ V} / 1024 = 0,0048 \text{ V}$$

Na grafu 4 je prikazana napaka kvantizacije ($\pm 1 \text{ LSB}$) zaradi nelinearne karakteristike NTK upora (graf 3). Graf smo dobili tako, da smo celotno merilno območje razdelil na 1024 delcev (oz. povečevali korak napetosti med 0 in 5 V za $4,88 \text{ mV} = 1 \text{ LSB}$). Nato smo s pomočjo transformacijske krivulje izračunal pripadajoče temperature in primerjali vrednost temperature n z $(n-1)$. Tako smo dobili temperaturne razlike, ki pomenijo napako kvantizacije pri določeni temperaturi. Spodnji graf potrди naša pričakovanja, da bo najmanjša napaka tam, kjer je transformacijska krivulja najpoložnejša.



Graf 4: Napaka zaradi kvantizacije

4.2.2 Napaka merilnega rezultata

Na točnost meritve poleg napake termistorja (toleranca 5 %) vpliva tudi napaka dodanega upora (toleranca 1 %), ki skupaj s termistorjem sestavlja napetostni delilnik. Iz tabele 2 je razvidno, da smo s tako vezavo v rezultat vnesli znatno napako.

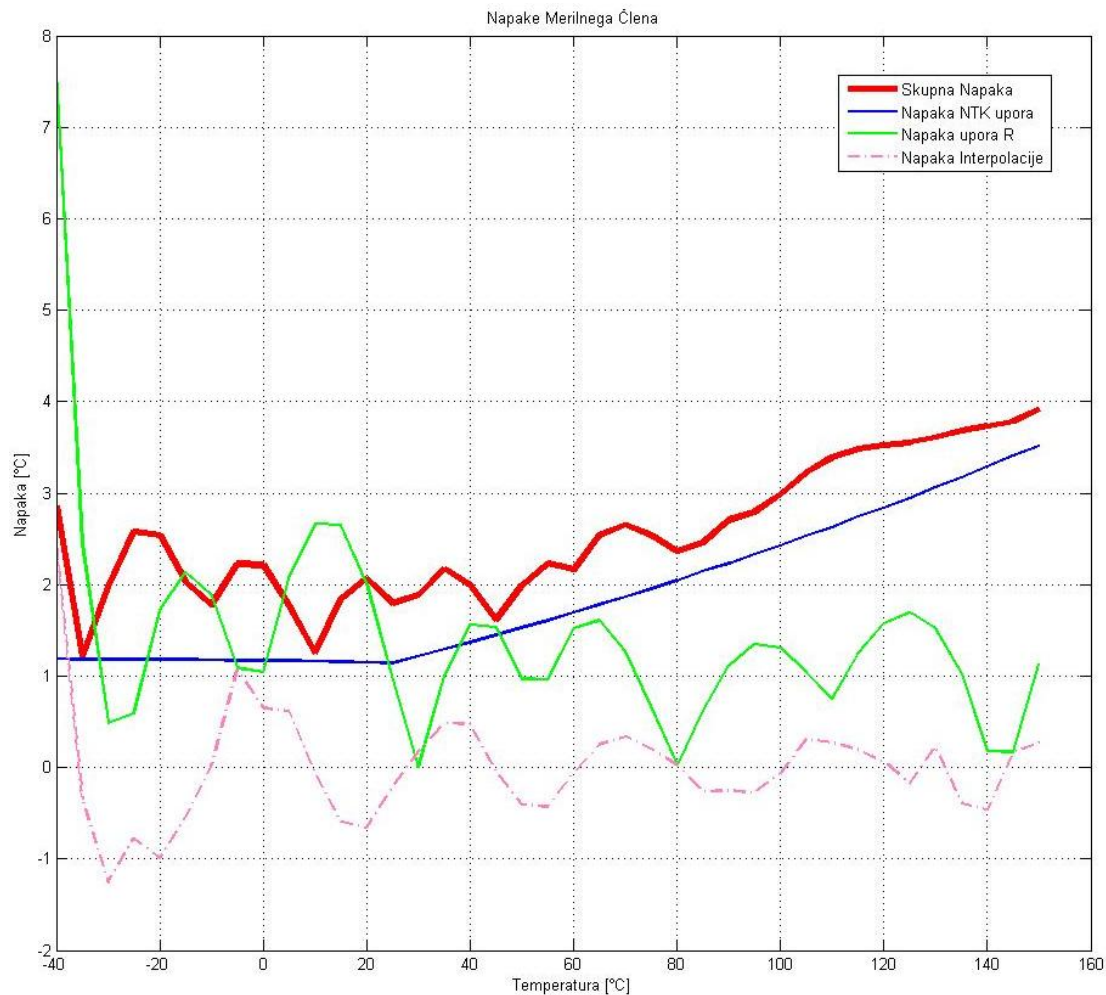
Pomen posameznih spremenljivk v spodnji tabeli:

- T [°C]: temperatura,
- R_{NTC} : nominalna upornost termistorja pri določeni temperaturi,
- ΔR [%]: deviacija upornosti zaradi tolerance parametra B,
- TC [% K]: temperaturni koeficient pri določeni temperaturi,
- Tol_{NTC} [%]: toleranca termistorja pri določeni temperaturi,
- Napak NTC [$\pm$ °C]: napaka samega termistorja pri določeni temperaturi,
- Napaka apro. [°C]: je napaka zaradi aproksimacije podatkov s polinomom,
- Maksimalna Napaka [$\pm$ °C]: je maksimalna napaka merilnega člena.

$T [^{\circ}\text{C}]$	$R_{\text{NTC}} [\Omega]$	$\Delta R [\%]$	$TC [\%K]$	$ToI_{\text{NTC}} [\%]$	Napaka NTC $[\pm^{\circ}\text{C}]$	Napaka apro. $[\pm^{\circ}\text{C}]$	Maksimalna Napaka $[\pm^{\circ}\text{C}]$
-40	156100	6,57	2,66	7,79	1,19	2,47	2,86
-35	112800	6,36	2,41	7,53	1,18	-0,35	1,22
-30	82350	6,15	2,17	7,28	1,18	-1,26	1,98
-25	60770	5,95	1,94	7,04	1,18	-0,78	2,58
-20	45300	5,76	1,71	6,80	1,18	-0,99	2,54
-15	34080	5,58	1,5	6,58	1,18	-0,55	2,03
-10	25870	5,4	1,29	6,35	1,18	0,03	1,78
-5	19810	5,24	1,08	6,13	1,17	1,07	2,22
0	15300	5,08	0,89	5,93	1,17	0,65	2,21
5	11910	4,92	0,7	5,74	1,17	0,62	1,76
10	9340	4,78	0,52	5,55	1,16	-0,07	1,26
15	7378	4,64	0,34	5,36	1,15	-0,59	1,84
20	5869	4,5	0,17	5,18	1,15	-0,66	2,07
25	4700	4,37	0	5,00	1,14	-0,22	1,79
30	3788	4,25	0,16	5,17	1,22	0,18	1,89
35	3072	4,13	0,32	5,34	1,29	0,48	2,17
40	2505	4,02	0,47	5,49	1,37	0,48	1,99
45	2055	3,91	0,62	5,65	1,45	-0,05	1,61
50	1694	3,8	0,77	5,81	1,53	-0,40	1,98
55	1405	3,7	0,91	5,96	1,61	-0,43	2,22
60	1170	3,6	1,05	6,10	1,70	-0,07	2,16
65	979,7	3,51	1,18	6,24	1,78	0,25	2,54
70	823,9	3,42	1,31	6,38	1,86	0,34	2,65
75	696	3,33	1,44	6,51	1,96	0,21	2,54
80	590,5	3,25	1,57	6,65	2,05	0,02	2,37
85	503,1	3,16	1,69	6,77	2,14	-0,26	2,46
90	430,3	3,09	1,81	6,90	2,23	-0,26	2,70
95	369,4	3,01	1,93	7,03	2,33	-0,28	2,79
100	318,3	2,94	2,04	7,14	2,43	-0,07	2,99
105	275,3	2,87	2,15	7,26	2,53	0,31	3,23
110	238,9	2,8	2,26	7,37	2,63	0,27	3,39
115	208	2,73	2,37	7,49	2,74	0,20	3,48
120	181,7	2,67	2,47	7,59	2,84	0,06	3,52
125	159,2	2,61	2,57	7,70	2,95	-0,17	3,56
130	139,9	2,55	2,67	7,80	3,06	0,23	3,61
135	123,3	2,49	2,77	7,91	3,18	-0,40	3,67
140	109	2,43	2,86	8,00	3,29	-0,46	3,73
145	96,6	2,38	2,96	8,11	3,41	0,16	3,78
150	85,8	2,33	3,05	8,20	3,52	0,28	3,91

Tabela 2: Karakteristika NTK upora in merilnega člena

V spodnjem grafu so prikazane napake, merilnega člena. Največjo napako dobimo, zaradi tolerance uporov. Boljši rezultat bi dobili le v primeru če bi upor tolerance 1% zamenjali s precizijskim uporom tolerance 0,01%.



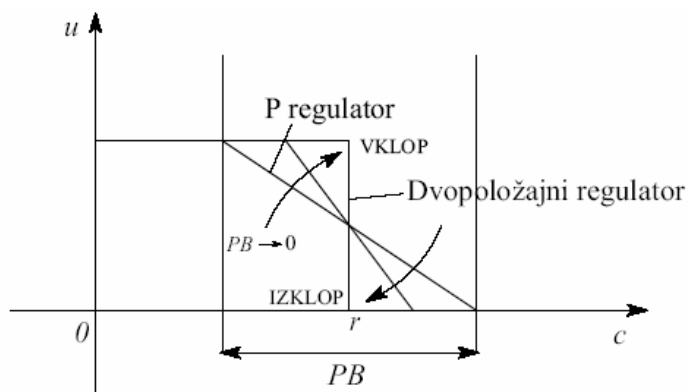
Graf 5: Napake merilnega člena

5 REGULACIJA TEMPERATURE

V tem poglavju bomo opisali potek regulacije želene temperature. Ker smo modul izdelali za regulacijo sobne temperature ni bilo potrebno po načrtovanju kompleksnih regulacijskih postopkov ampak smo izdelali dvostopenjsko (ON-OFF) regulacijo. V primeru, da bi želena temperatura bistveno vplivala na sam proces bi moral izbrati zahtevnejše regulacijske postopke.

5.1 Dvostopenjska regulacija

Zvezno delujoči regulatorji generirajo regulirno veličino, ki lahko zavzame vse vrednosti med minimalno in maksimalno vrednostjo [1 str. 231]. Vendar pa je možno dobiti hitrejši odziv (krajši čas vzpona) na ta način, da že ob relativno majhnih pogreških izvršni sistem pride v nasičenje. Zato si lahko predstavljamo, da nastane stopenjski regulator iz zveznega proporcionalnega³ regulatorja tako, da mu zmanjšamo proporcionalno območje oz. povečamo ojačenje. V primeru, da je $PB = 0$ oz. $K_P = \infty$, preide statična karakteristika proporcionalnega (P) zveznega regulatorja v statično karakteristiko dvopoložajnega ali (ON-OFF) regulatorja. Prehod iz P regulatorja v dvopoložajni prikazuje slika 5.1.

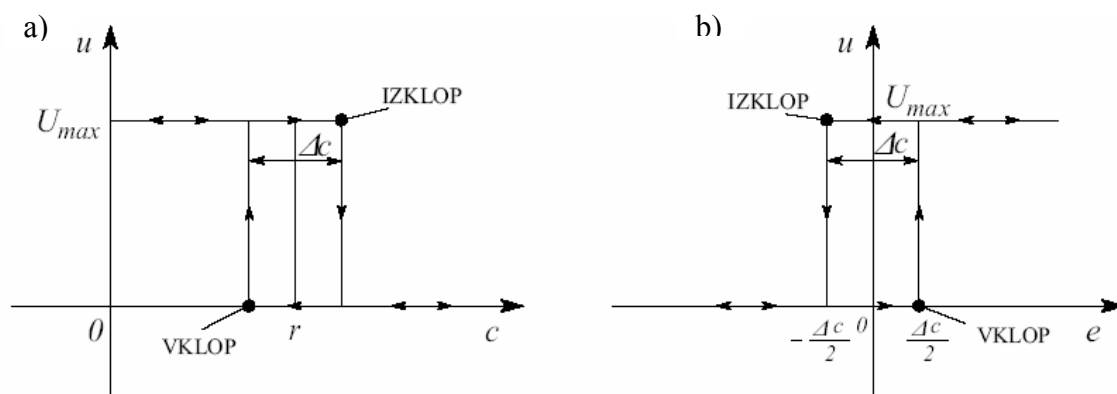


Slika 5.1: Prehod iz P regulatorja v dvopoložajni ali ON/OFF regulator [1 str. 231]

Dvopoložajni regulator obratuje torej le v dveh preklopnih stanjih - običajno vklop in izklop. Če je regulirana veličina pod vrednostjo reference ali želene vrednosti r , daje regulator polni regulirni signal $U_{\max}$ kar predstavlja vklop izvršnega sistema. Če pa je

³ Za proporcionalni regulator je značilno, da je regulirna veličina premo sorazmerna pogrešku.

regulirana veličina večja od reference, regulator daje vrednost 0 (ali tudi kakšno drugo vrednost $U_{\min}$) kar običajno pomeni, da se končni izvršni člen izključi. Pri reverznem delovanju je ravno obratno. Tak idealni dvopoložajni regulator sicer dobro regulira proces, če le-ta nima večjih zakasnitev, vendar pa pri tem izredno hitro preklaplja. Tehnične izvedbe dvopoložajnih regulatorjev izkazujejo vedno neko histerezo, kar pomeni, da vklop nastopi pri drugi vrednosti kot izklop. Vklop nastopi pri pozitivnem pogrešku ($c^4 < r$), izklop pa pri negativnem pogrešku ($c > r$). Slika 5.2a prikazuje statično odvisnost regulirne veličine od regulirane⁵ veličine, slika 5.2b pa od pogreška regulacijskega sistema. Z Δc je označena velikost histereze. Karakteristiki veljata za direktno delovanje, v primeru reverznega delovanja pa se ustrezno spremenita.



Slika 5.2: Statična karakteristika dvopoložajnega regulatorja [1 str. 232]

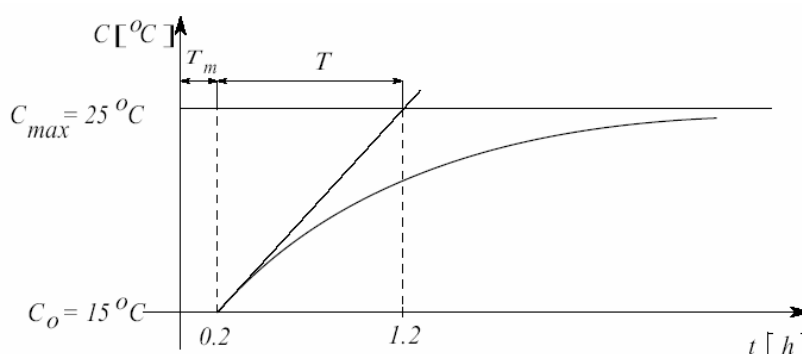
Dvopoložajni regulatorji so lahko električni, mehanski, hidravlični, pnevmatski ali kombinirani [1 str. 232]. Uporabljajo se za regulacijo temperature, tlaka, nivoja, itd. Primerni so tudi za računalniško izvedbo, saj v tem primeru ne potrebujemo D/A pretvornika, ampak vhod izvršnega sistema povežemo na digitalno izhodno linijo računalnika .

Prednost dvopoložajne regulacije je predvsem v ceni izvedbe, medtem ko ima z-regulacijskega stališča več slabosti: nihanje regulirane veličine, udari v mreži zaradi vklapljanja in izklapljanja, obraba kontaktov itd.

⁴ c – regulirana veličina (v našem primeru je to temperatura)

Dvopoložajno reguliranje proporcionalnih procesov

Za boljše razumevanje dvopoložajne regulacije si zamislimo naslednji termični proces: če ne deluje grelnik [1 str. 233], je temperatura v prostoru enaka temperaturi okolice (npr. 15 °C). Če pri tej temperaturi vključimo grelnik, ki daje regulirno veličino $u = U_{\max}$ (npr. 5 kW), se temperatura v prostoru dvigne po prehodnem pojavu na temperaturo $C_{\max}$ (npr. 25 °C). Če sedaj izključimo grelnik, se bo temperatura vrnila na začetno temperaturo C_0 zaradi izgub (odvajanje toplote skozi stene, okna, ...). Proces 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 5.3.



Slika 5.3: Potek temperature [1 str. 233]

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

$$G_p(s) = \frac{K}{Ts + 1} e^{-T_m s}$$
$$K = \frac{C_{\max} - C_0}{U_{\max}} = \frac{10^\circ\text{C}}{5\text{kW}} = 2^\circ\text{C/kW}$$
$$T = 1\text{h} \quad T_m = 0.2\text{h} \text{ (mrtvi čas)}$$

Pomen posameznih simbolov v gornji enačbi:

K – ojačenje sistema,

T – časovna konstanta sistema,

T_m – mrtvi čas sistema,

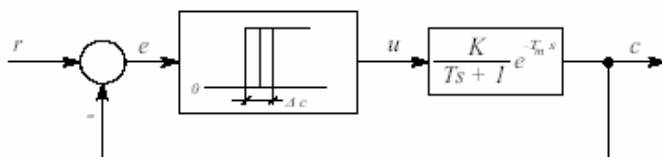
$C_{\max}$ – maksimalna temperatura (ki jo lahko dosežemo),

C_0 – temperatura okolice,

$U_{\max}$ – maksimalna regulirna veličina (vkloplejno grelo).

Merjeni odziv sicer ne izkazuje čistega mrtvega časa, vendar lahko nadkritično dušenje sisteme višjega reda zadovoljivo aproksimiramo s sistemom 1. reda in mrtvim časom. Prav tako predpostavljamo, da je časovna konstanta pri gretju in ohlajevanju enaka, kar običajno povsem ne drži [1 str. 234].

Z vklapljanjem in izklapljanjem grelnika lahko vzdržujemo temperaturo med C_0 in $C_{\max}$. Obravnavani regulacijski sistem prikazuje slika 5.4.



Slika 5.4: Bločna shema regulacijskega sistema z dvopoložajnim regulatorjem
[1 str. 234]

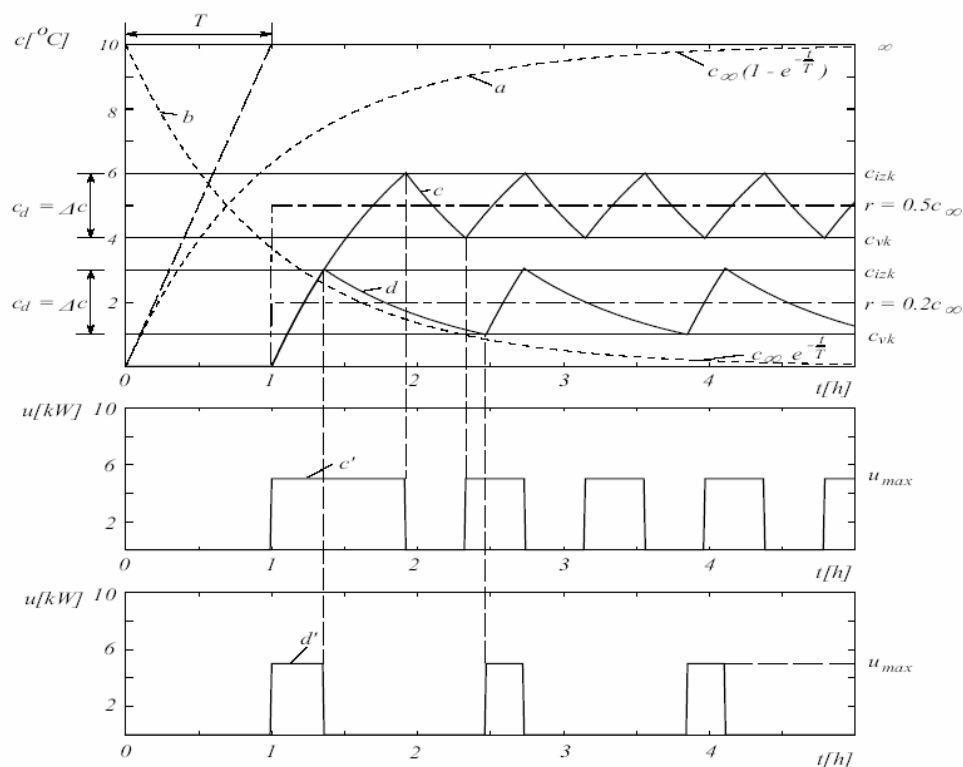
Slika 5.4 kaže delovanje v delovni točki, zato se pri vklopljenem grelniku $u = U_{\max} = 5 \text{ kW}$ regulirana veličina giblje od 0 do C_{∞} (0 do $10 \text{ }^{\circ}\text{C}$). Zato lahko izbiramo referenco v področju od 0 do C_{∞} (0 do $10 \text{ }^{\circ}\text{C}$ kar v absolutnih vrednostih pomeni med $15 \text{ }^{\circ}\text{C}$ in $25 \text{ }^{\circ}\text{C}$).

Analitična obravnava takšnega povratnozančnega sistema je zelo zahtevna, zato je potrebno v časovnem prostoru ločeno analizirati fazo vklopa in izklopa. V vsaki od obeh faz se sistem vede linearno.

Zaradi enostavnosti najprej analizirajmo regulacijo v primeru, če proces nima mrtvega časa ($T_m = 0$). Regulirani in regulirni veličini za različne vrednosti reference prikazuje slika 5.5. Pri vklopu regulatorja je $u = U_{\max}$ in regulirana veličina bi v primeru razklenjene povratne znake eksponencialno naraščala do vrednosti $c = C_{\infty} = K \cdot U_{\max}$ (krivulja na sliki 5.5). Če bi pri doseženi temperaturi $c = C_{\infty}$ regulator izklopil, bi regulirana veličina v primeru razklenjene povratne zanke upadala po krivulji b.

Postavimo sedaj referenco na 50% reguliranega območja. Regulirana veličina se v tem primeru spreminja po krivulji c. V fazi ogrevanja opazimo, da regulator zaradi histereze ne izklopi pri $c = r$ ampak nekoliko kasneje pri vrednosti C_{izk} . Pri ohlajanju

pa regulator ne vklopi pri $c = r$ ampak spet nekoliko kasneje, ko doseže temperatura vrednost C_{vkl} .



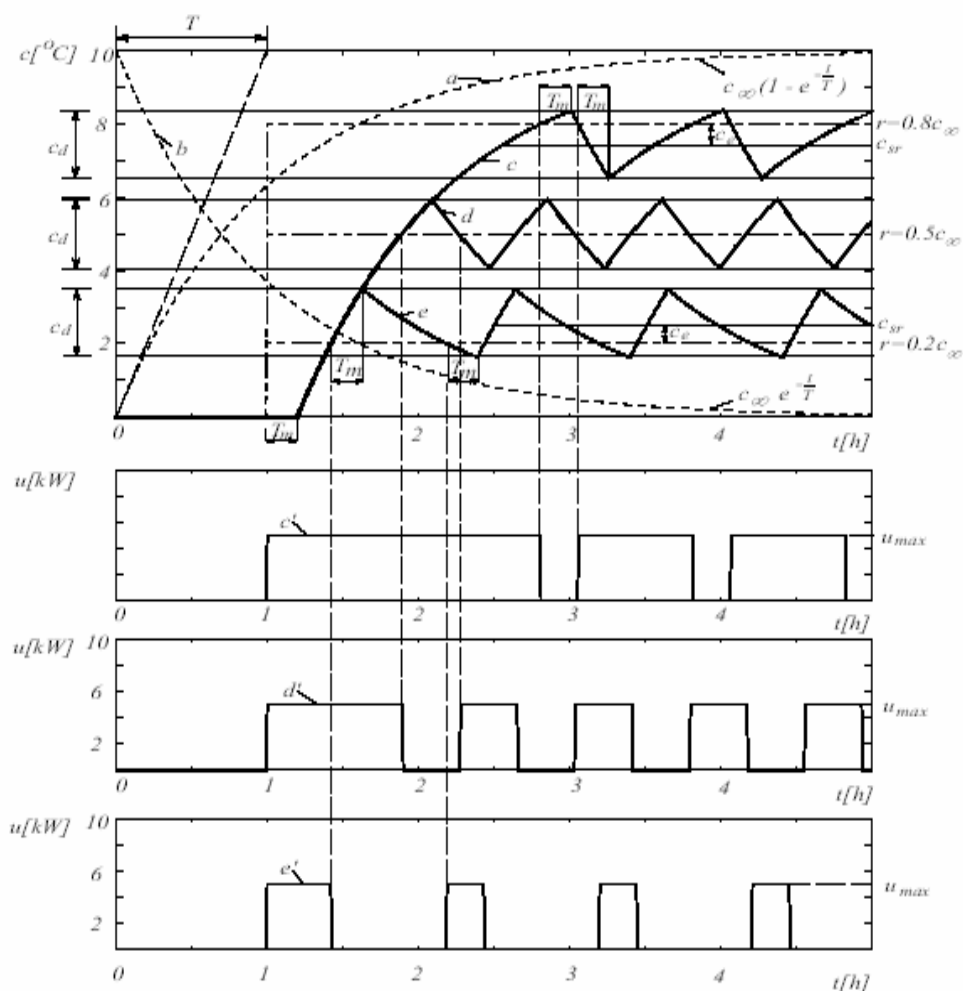
Slika 5.5: Potek regulirane in regulirne veličine, če ne upoštevamo mrtvega časa
[1 str. 235]

Na ta način se zaradi histereze regulatorja igra vklopov in izklopov okoli vrednosti, ki je podana z referenco, ponavlja. Če imamo možnost zmanjšanja širine histereze, se zmanjša razmik nihanja C_d , poveča pa se frekvenca preklopov. Frekvenca preklopov zavisi še od drugih parametrov [1 str. 235]. To jasno vidimo iz krivulje d , ki prikazuje potek pri nastavljeni referenci $r = 0,2C_{\infty}$. Frekvenca preklopov je manjša, pa tudi razmerje časov vklopa in izklopa, ki je bilo pri $r = 0,5 C_{\infty}$ 1:1, se v tem primeru zmanjša.

Podoben postopek določanja prehodnega pojava lahko uporabimo tudi pri procesih, ki imajo razen časovne konstante tudi mrtvi čas. Slika 5.6 prikazuje potek reguliranja

na procesu prvega reda z mrtvim časom, pri čemer smo zaradi enostavnosti predpostavili, da regulator nima histereze [1 str. 237].

Pri polno vključeni regulirni veličini se bliža regulirana veličina željeni vrednosti. Ker regulator nima histereze, se izklopi regulirna veličina takoj, ko c doseže vrednost r ($u = 0$). Vendar se pozna izklop na regulirani veličini šele po preteku mrtvega časa T_m . Zato c v tem času se narašča. Nato začne c upadati in ko doseže vrednost r , regulator zopet vklopi ($u = U_{\max}$). Podaljšano upadanje veličine c je ponovno posledica mrtvega časa procesa. Iz slike 5.6 spoznamo, da je frekvenca preklapov največja pri nastavitvi $r = 0,5 C_{\infty}$. Periodično nihanje poteka drugače kot na sliki 5.4. Če je $r = 0,5 C_{\infty}$, je srednja vrednost nihanja enaka referenčnemu signalu. Če $r \neq 0,5 C_{\infty}$, dobimo ustaljeni odmik c_e ($c_e = r - c_{sr}$). Za $r < 0,5 C_{\infty}$ je ustaljeni odmik negativen, za $r > 0,5 C_{\infty}$, pa pozitiven.



Slika 5.6: Potek regulirane in regulirne veličine, če ne upoštevamo histereze

[1 str. 237]

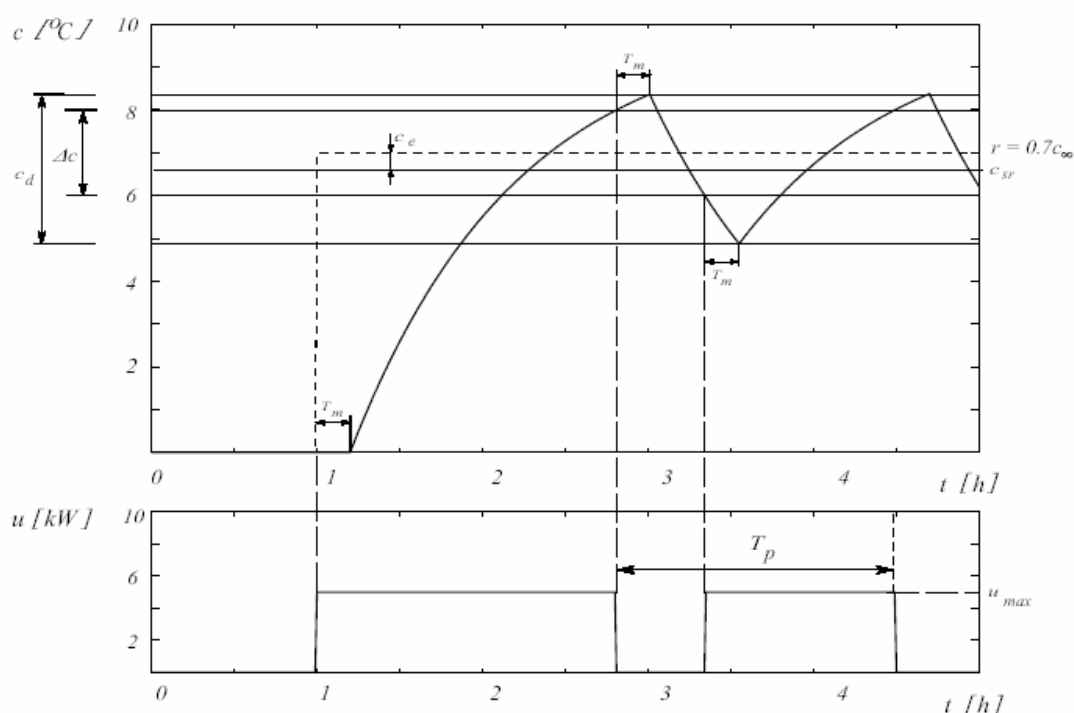
Pri regulaciji procesa z mrtvim časom in regulatorjem s histerezo se oba opisana poteka (sliki 5.5, 5.6) superponirata, kot kaže slika 5.7.

Na osnovi grafičnih prikazov lahko določimo karakteristične parametre - razmik nihanja C_d , ustaljeni odmik C_e in frekvenco prekopov $f_p=1/T_p$. Za praktične potrebe običajno zadostuje le približna določitev teh vrednosti, s čimer se obravnava precej poenostavi.

Razmik nihanja podaja enačba 5.1:

$$c_d = \Delta c + \frac{T_m}{T} c_\infty \quad (5.1)$$

Razmik nihanja je tem večji, čim večja je širina histereze, čim večji je mrtvi čas T_m , čim manjša je časovna konstanta T in čim večja je ustaljena vrednost C_∞ . Pri manjši časovni konstanti T se namreč regulirna veličina v času T_m bolj spremeni. Podobno vpliva tudi večji C_∞ . Ker je $C_\infty = K \cdot U_{\max}$, je razmik nihanja tem večji, čim večje je ojačenje procesa K ali čim večja je maksimalna regulirna veličina. V tem primeru je eksponentna krivulja strmejša in v času T_m se regulirana veličina bolj spremeni.



Slika 5.7: Potek regulirane in regulirne veličine, če upoštevamo histerezo in mrtvi čas [1 str. 238]

Periodo nihanja podaja enačba 5.2:

$$T_p = \frac{T_m + T \frac{\Delta c}{c_\infty}}{\frac{r}{c_\infty} \left(1 - \frac{r}{c_\infty} \right)} \quad (5.2)$$

Perioda nihanja je tem daljša, čim daljši sta T_m in T ter čim večja je širina histereze Δc . Minimalni T_p dosežemo pri $r = 0,5C_\infty$, torej kadar je referenca pri 50% reguliranega področja [1 str. 238].

Srednjo vrednost opisuje enačba 5.3:

$$c_{sr} = r + \frac{T_m}{T} \cdot \left(\frac{c_\infty}{2} - r \right) \quad (5.3)$$

Ustaljeno stanje opisuje enačba 5.4:

$$c_e = r - c_{sr} = \frac{T_m}{T} \left(r - \frac{c_\infty}{2} \right) \quad (5.4)$$

Le-ta je nič, če je $r = C_\infty$, pozitiven, če $r > C_\infty$ in negativen, če $r < C_\infty$. Če ni mrtvega časa, tudi ni ustaljenega odmika.

5.2 Programska izvedba dvopoložajne regulacije

Temperaturo smo regulirali okrog želene temperature s histerezo, ki jo poda uporabnik. V spodnjih nekaj vrsticah si bomo ogledali kako smo napisali algoritem za dvostopenjsko regulacijo.

Najprej se izvede inicializacija grela. Inicializacija grela smo poimenovali postopek s katerim preverimo, če smo dobili nova parametra⁶ regulacije. Pomagamo si s spremenljivko »prvic«. V primeru, da je spremenljivka prvic postavljena na vrednost 1, smo prejeli nove parametre regulacije. Zatem preverimo ali je trenutna temperatura večja od maksimalne⁷ temperature in če je, postavimo spremenljivko »kontrola« na vrednost 1. Spremenljivko kontrola uporabljamo zato, da nas obvešča,

⁶ Parametra regulacije sta: želena temperatura in histereza

⁷ Maksimalna temperatura = želena temperatura + histereza

ali smo že dosegli maksimalno temperaturo. To uporabljamo zato, da se grelo ne vklaplja vsakič, ko temperatura pade pod maksimalno temperaturo. Vrednost spremenljivke postavimo na 0, kadar dosežemo minimalno⁸ temperaturo.

```
If(prvic == 1){  
  If(trenutna < tempmax){  
    kontrola = 1;  
  }  
  else{  
    kontrola = 0;  
  }  
}
```

Inicializacija grela

Ko je inicializacija grela končana, vklapljamo oziroma izklapljamo grelo po sledečem algoritmu.

```
If(trenutna > tempmax && kontrola == 0){  
  GRELO_IO = 1;  
}
```

1

1. Prvi pogojni stavek (zgoraj) preveri ali je trenutna temperatura manjša od maksimalne in ali je spremenljivka kontrola postavljena na vrednost 0. Če so vsi pogoji izpolnjeni lahko sklepamo, da temperatura raste in še ni dosegla maksimalne temperature. V tem primeru postavimo spremenljivko GRELO_IO na vrednost 1, kar pomeni vključitev grela. V vsakem pogojnem stavku je logika obrnjena tako, da manjše pomeni večje in obratno. Vzrok temu je, da predstavlja manjša temperatura večji padec napetosti na upor in s tem večjo vrednost A/D pretvornika.

```
if(trenutna > tempmax && kontrola == 1{  
  GRELO_IO = 0;  
}
```

2

2. Drugi pogojni stavek preveri ali je trenutna temperatura manjša od maksimalne temperature in če je spremenljivka kontrola postavljena na 1. To pomeni, da smo že dosegli maksimalno temperaturo in čakamo, da temperatura ponovno pade na minimalno vrednost.

⁸ Minimalna temperatura = zelena temperatura – histereza

```
if(trenutna < tempmax && kontrola == 0){
  GRELO_IO = 0;
  kontrola = 1;
}
```



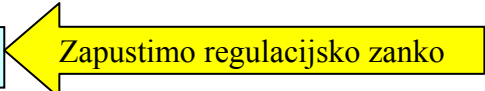
3. V zgornjem primeru smo dosegli maksimalno temperaturo zato izključimo grelo in postavimo spremenljivko kontrola na 1.

```
if(trenutna >= tempmin){
  GRELO_IO = 1;
  kontrola = 0;
}
```



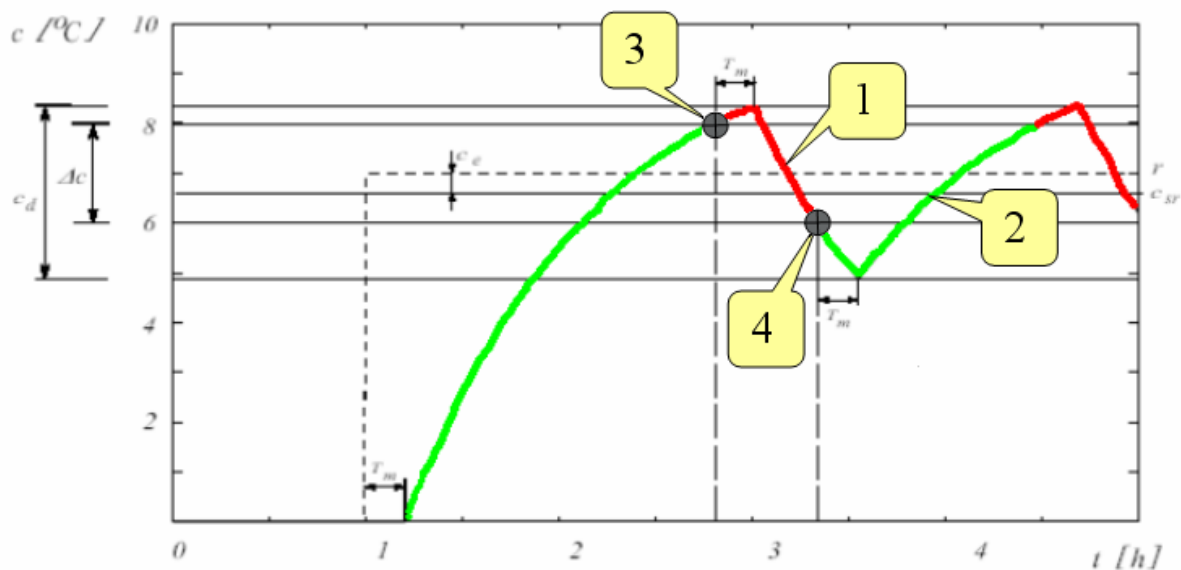
4. V zgornjem primeru smo dosegli minimalno temperaturo, zato vključimo grelo in postavimo spremenljivko kontrola na vrednost 0. Od te točke dalje čakamo, da dosežemo maksimalno temperaturo.

```
prvic = 0;
```



Preden zapustimo glavno regulacijsko zanko postavimo spremenljivko prvic na vrednost 0 zato, da v naslednjem koraku izpustimo inicializacijo grela. Spremenljivka se postavi na vrednost 1, ko uporabnik ponovno spremeni parametre regulacije.

Slika 5.8 prikazuje vpliv posameznih delov regulacijskega algoritma na regulirano veličino. Kjer pomeni zeleno obarvana krivulja vključeno grelo, rdeča pa izključeno grelo. Sivi krogi nakazujejo kje se spremeni vrednost spremenljivke kontrola.



Slika 5.8: Grafični prikaz regulacijskega algoritma [1 str. 238]

5.2.1 Delovanje v primeru napake

Med delovanjem regulacije lahko pride do napake v povezavi (prekine se povezava z internetom). V tem primeru smo poskrbeli tako, da uporabnik lahko izbere med tem ali se izvaja regulacija v temperaturnem območju, ki ga je uporabnik nazadnje določil ali pa modul preide v stanje, ki smo ga poimenovali (režim varčevanje).

Režim varčevanje regulira temperaturo v predhodno določenem temperaturnem območju. Režim delovanja izberemo tako, da sklenemo oziroma razklenemo namensko stikalo na modulu. V primeru sklenitve stikal bo regulator deloval v režimu varčevanje in obratno. Za indikacijo pojave napake skrbi led dioda, ki v primeru napake utripa 4-krat na sekundo. Iz režima varčevanje se vrnemo ob ponovni vzpostavitvi povezave, s ponovno nastavitvijo parametrov.

6 MODUL

Srce modula predstavlja Microchipov mikrokrmilnik PIC18F4620 na sliki spodaj. Večina Microchipovih mikrokrmilnikov ima skupen priključek za SPI (ang. Serial Peripheral Interface) in RS232 komunikacijo, zato skupna uporaba obeh vodil zahteva skrbno načrtovanje sistema. Mikrokrmilnik, ki smo ga izbrali, ima ti dve komunikaciji ločeni, kar je bil jeziček na tehnicni za našo odločitev. SPI komunikacijo uporabljamo za komunikacijo med mikrokrmilnikom in Ethernet krmilnikom, in mikrokrmilnikom ter EEPROM-om. RS232 komunikacijo bi lahko uporabljali za enostavno spreminjanje parametrov modula. Za generiranje mikrokrmilnikovega takta smo uporabili 10 MHz kristal in uporabili funkcijo 4xPLL (ang. phase locked loop), s tem smo zagotovili delovanje mikrokrmilnika na 40 MHz.

Poleg omenjenih prednosti, mikrokrmilnik izpolnjuje tudi vse ostale zahteve glede velikosti pomnilnika, A/D pretvornika... . Naj omenimo nekaj značilnosti [7 str. 1]:

pomnilnik FLASH: 64 KB,

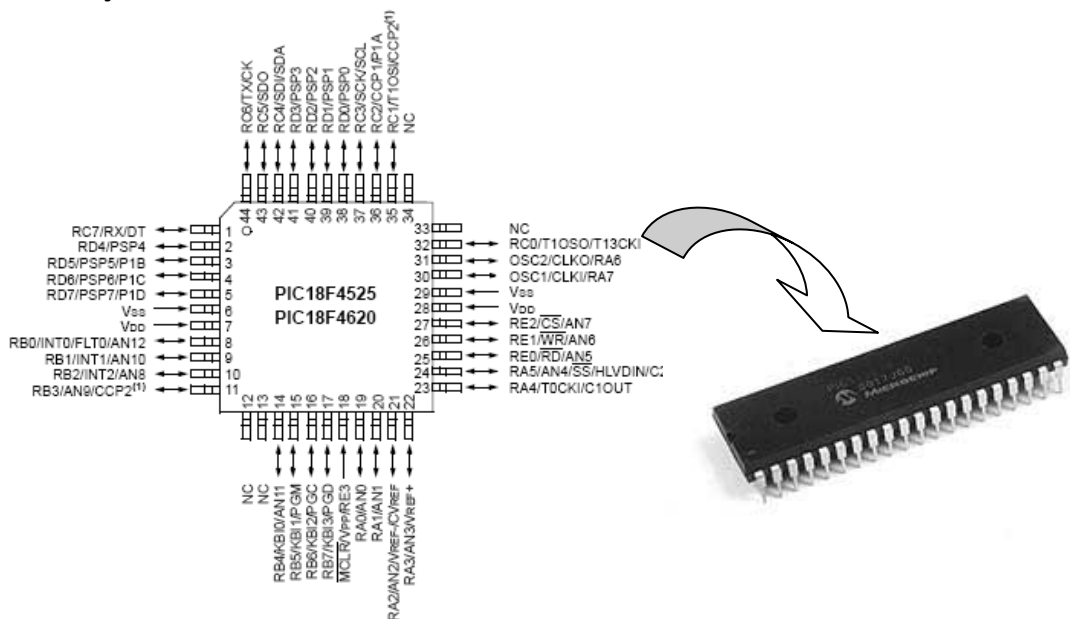
pomnilnik SRAM: 3986B,

EEPROM: 1024B,

10-bit A/D s 13 časovno prepletenih vhodov,

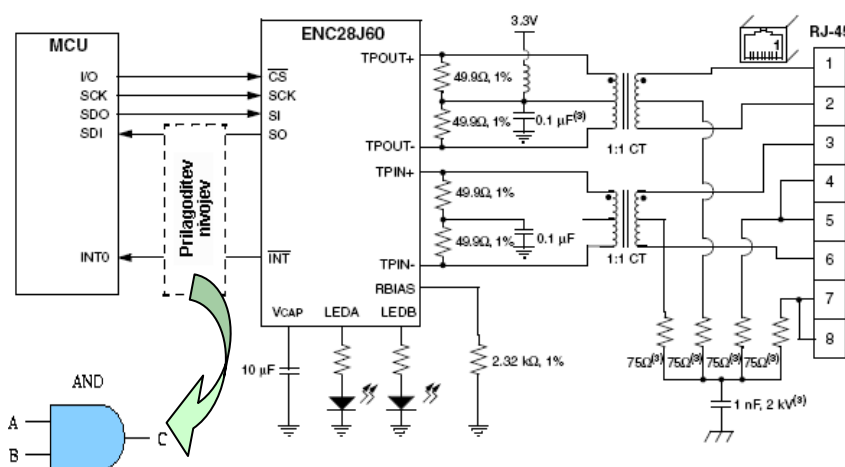
I/O vrat: 36,

komunikacija: I²C, SPI, UART.



Slika 6.1: Razpored priključnih sponk in slika mikrokrmilnika PIC18F4620

Za internetno povezavo poskrbi Microchipov ethernet krmilnik ENC28J60 [8]. Komunikacija poteka po SPI vodilu s frekvenco 10 MHz. Odlikuje ga predvsem majhno število vhodnih priključkov in majhno število uporabljenih elementov. Spodnja slika prikazuje ethernet krmilnik v tipični vezavi z mikrokrmilnikom. Tu ima pomembno vlogo primeren konektor (RJ45), ki že vsebuje ločilne transformatorje. Iz spodnje slike je razvidno, da je poleg signalnih diod in prilagoditvenih uporov $49,9\ \Omega$ potrebno dodati le še po en kondenzator in upor. Ker krmilnik deluje na 3,3 V potrebujemo prilagoditveno vezje za SPI komunikacijo v smeri ENC28J60 → mikrokrmilnik. V obratni smeri ni težav, ker so vhodi v vezje ENC28J60 tolerantni za napetosti do 5 V. Neprijetnost rešimo tako, da v vezje dodamo potrebno število IN⁹ vrat.

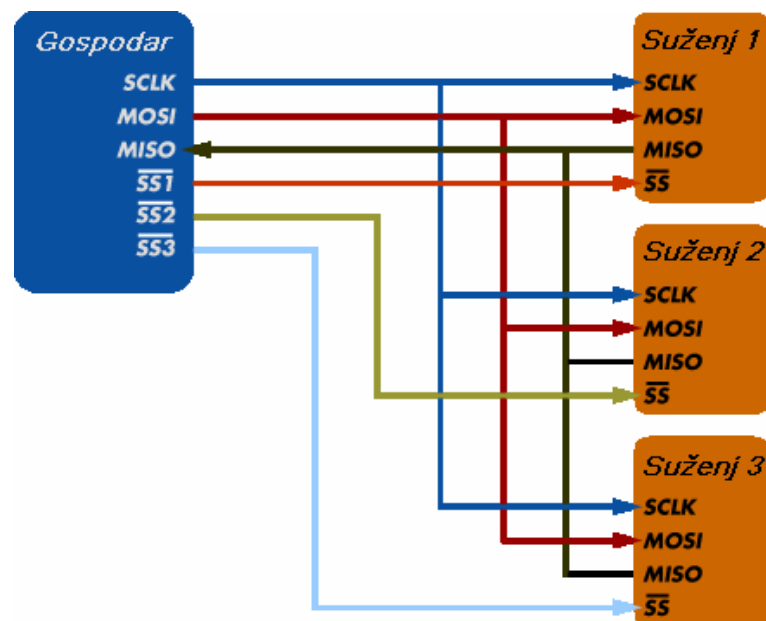


Slika 6.2: Osnovna vezalna shema mikrokrmilnika in ethernet krmilnika v omrežje
[8 str. 7]

6.1 SPI komunikacija

SPI komunikacija je serijska komunikacija, ki deluje po načelu gospodar-suženj (ang. master-slave). Komunikacijo odlikuje visoka hitrost in majhno število uporabljenih priključkov. Na spodnji sliki je prikazana tipična aplikacija, ki uporablja SPI komunikacijo. Uporabljena so tri skupna vodila, dodatno linijo pa doprinese vsak suženj, ki ga želimo priključiti [26].

⁹ Logična IN vrata: Elektronsko vezje, ki glede na vrednost vhodnih signalov izračuna vrednost logične operacije (konjunkcije). [5 str. 242]



Slika 6.3: SPI komunikacija [25]

Pomen posameznih signalov v SPI komunikaciji je naslednji:

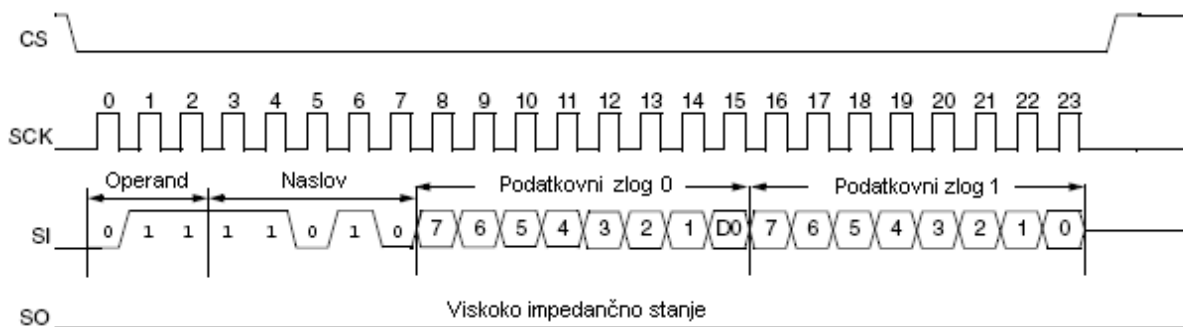
MOSI (ang. Master Output Slave Input). Kadar mikrokrmilnik deluje kot gospodar, je ta linija namenjena pošiljanju podatkov, kadar deluje kot suženj pa je linija namenjena sprejemanju podatkov.

MISO (ang. Master Input Slave Output). Kadar mikrokrmilnik deluje kot gospodar, je ta linija namenjena sprejemanju podatkov, kadar deluje kot suženj pa je linija namenjena pošiljanju podatkov. V primeru, da je komunikacija izključno enosmerna, se lahko komunikacija izvede s samo tremi linijami (odpade MOSI ali MISO).

SCK (ang. SPI Clock). Ta linija je namenjena določanju takta prenosa posameznih bitov. Takt daje gospodar, medtem ko pomikalni register sužnja sprejema podatke glede na takt.

CS (ang. Chip Select). S to linijo izberemo želenega sužnja tako, da jo postavimo na nizek nivo. To pomeni da kadar mikrokrmilnik deluje kot suženj in je ta priključek na visokem nivoju se ignorira takt na liniji SCK in podatke na MOSI liniji.

Slika 6.4 prikazuje potek SPI komunikacije med mikrokrmilnikom in ethernet krmilnikom ENC28J60.



Slika 6.4: Potek SPI komunikacije [8 str. 27]

Potek komunikacije je sledeč:

- za začetek komunikacije postavi gospodar linijo CS na nizek nivo krmilniku, s katerim želi komunicirati. Če je potrebno počakati določeno zakasnitev (npr. A/D pretvorbo), mora gospodar počakati vsaj toliko časa, da se le ta konča. Nato gospodar generira SPI takt, ki mora biti manjša ali kvečjemu enaka maksimalni frekvenci, ki jo suženj še podpira,
- gospodar pošlje podatke po MOSI liniji; suženj bere podatke na MOSI liniji,
- suženj pošlje podatke po MISO liniji; gospodar bere podatke na MISO liniji.

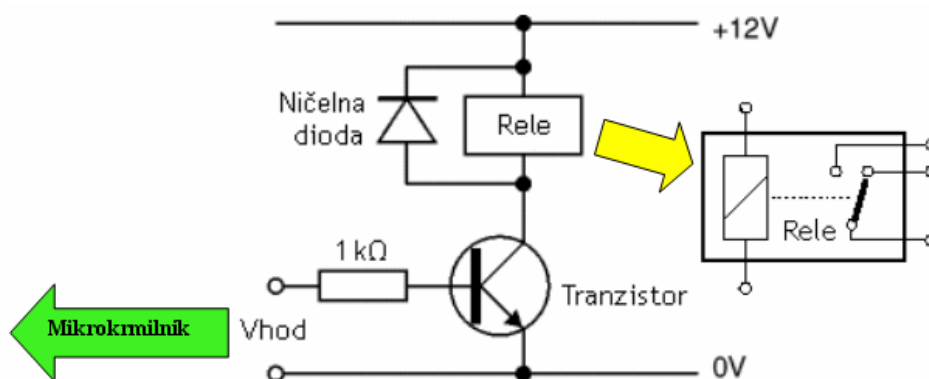
Medtem ko poteka komunikacija med gospodarjem in izbranim sužnjem, ostali sužnji, ki nimajo linije CS na nizkem nivoju, ignorirajo dogajanje na SPI vodilu.

6.2 Močnostni del

Ker moramo z mikrokrmilnikom vklaplјati/izklaplјati grelo v električnem omrežju, smo izhod mikrokrmilnika vezali na rele s podatki:

- Napetost preklopa releja 12 V,
- nazivna napetost močnostne veje releja 250 V,
- maksimalen tok močnostne veje releja 10 A.

Slika 6.5 prikazuje vezavo releja na vhod mikrokrmilnika. Vezje sestavlja tranzistor kot električno stikalo, ničelna dioda in upor, s katerim nastavimo bazni tok tranzistorja.



Slika 6.5: Močnostni del [27]

Minimalni tok, ki ga rele porabi za preklon, je 100 mA, zato izberemo tranzistor tako, da ima tok $I_C(\text{max})$ večji ali enak minimalnemu toku preklopa releja. Sami smo izbrali tranzistor 2N2904 ($I_C(\text{max})=500$ mA): lahko bi uporabil tudi katerega drugega, vendar smo tega imeli na razpolago. S takim vezjem lahko v našem primeru vklapljam porabnike do moči 2,5 kW.

Ničelna dioda:

Ob preklopu releja se skozi tuljavo spremeni tok, ki povzroči spremembo magnetnega pretoka [33, str. 83]. Zaradi tega se inducira napetost, ki temu toku nasprotuje. Najbolj neugodna situacija nastopi ob izklopu, ko tok skozi tranzistor in tuljavo hitro upade. V tuljavi se inducira napetost, ki ima tako polariteto, da se prišteva k baterijski napetosti in lahko prekorači maksimalno dopustno napetost $U_{ce}(\text{max})$. Posledica je uničenje tranzistorja. Da to preprečimo vežemo vzporedno k tuljavi diodo, ki je tako polarizirana, da dovoljuje nadaljnji tok skozi relejevo tuljavo, ko se tranzistor zapre.

Slika 6.6 prikazuje internetni modul, ki smo ga izdelali. Nastal je kot testni projekt podjetja Iskre Avtoelektrike d.d.. Namenjen je za vgradnjo v kontrolne naprave za

zajemanje podatkov in upravljanje naprav po lokalnem omrežju. Vezje smo v celoti načrtali in sestavili sami, tiskarina je bila izdelana v podjetju Lingva d.o.o. V prilogi je priložena vsa potrebna dokumentacija za izgradnjo modula (vezalna shema, seznam uporabljenih elementov, filmi za izdelavo vezja).



Slika 6.6: Internetni modul

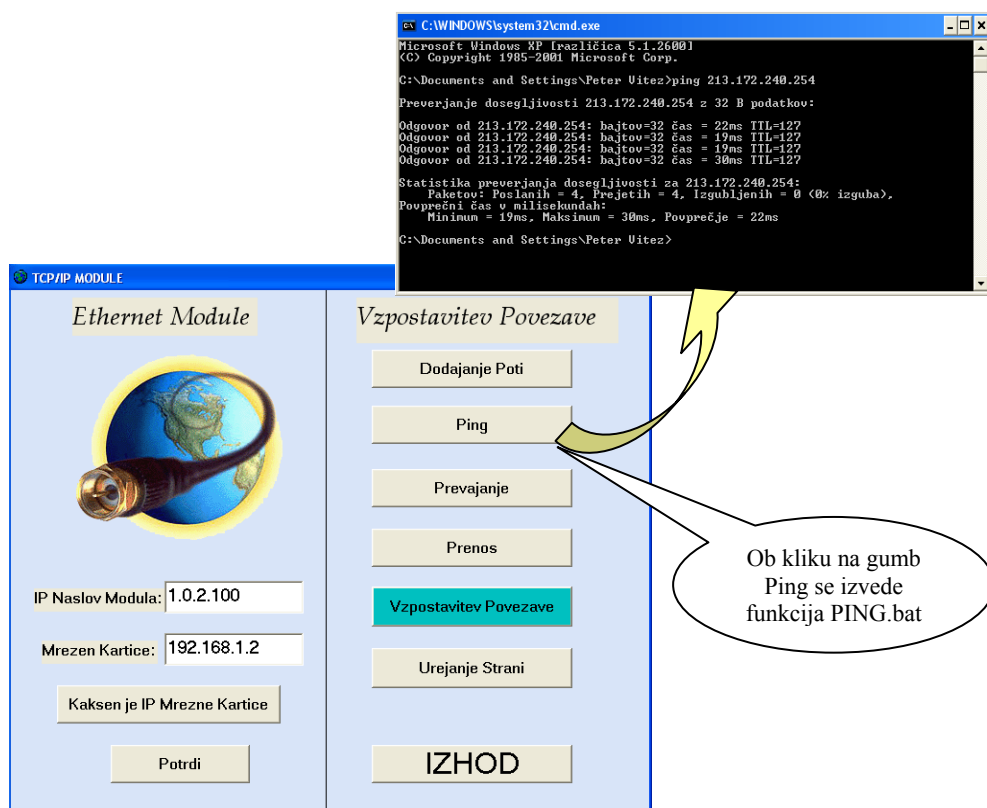
6.3 Grafični vmestnik

Za lažje izvajanje inicializacijskih postopkov in nastavitev modula smo izdelali grafični vmestnik (sliki 6.7) v programskem jeziku Visual Basic. Aplikacija je zgrajena tako, da vsak gumb kliče svoj skriptno datoteko(batch file) in jo izvede. Skriptne datoteke so datoteke, ki vsebujejo množico ukazov, ki jih lahko ročno vpisujemo preko konzolnega okna operacijskega sistema Windows, in se ob klicu izvršijo. V večini se jih uporablja kjer bi moral uporabnik večkrat vpisovati iste serije ukazov. Prepoznamo jih po končnici *.bat.

Postopek vzpostavitve povezave [29, 30, 31]:

- Najprej vnesemo IP naslov modula,

- drugo okence je namenjeno vnosu IP naslova mrežne kartice na katero je modul priključen. V primeru, da IP naslov kartice ni poznan je spodaj gumb (Kakšen je IP Mrežne Kartice), ki vrne IP naslov mrežne kartice,



Slika 6.7: Grafični vmestnik

- ob kliku na gumb (Potrdi) se ustvarijo nekatere *.bat datoteke, ki jih potrebujemo v nadaljevanju za vzpostavitev povezave,
- da lahko komuniciramo z modulom oziroma, da računalnik modul sploh zazna, je potrebno dodati nekatere nastavitve v računalnikovo usmerjevalno tabelo. Ob kliku na gumb (Dodajanje Poti) se izvede bat datoteka z vsebino:

```
route add 1.0.2.100 mask 255.255.255.255 192.168.1.2
```

S tem ukazom dodamo nov »cilj« z ustrezno masko in preходом (mrežno kartico). To je potrebno le v primeru, ko modul priključimo neposredno na računalnik,

- ob kliku na gumb Ping vidimo ali je bil prejšnji korak uspešen. Z ukazom ping opazujemo odmev modula, da ugotovimo, ali je le ta dosegljiv. Na tem mestu prvič komuniciramo z modulom,
- gumb (Prevajanje) je namenjen za prevajanje izvornih internetnih strani, ki jih naložimo v modul v obliki ene same datoteke, ki jo podpira Microchipov TCP/IP sklad. Gumb izvrši datoteko (Compile.bat), ki vsebuje program MPFS in ustrezne parametre.

MPFS -b WebPages Server.bin

MPFS je Microchipov namenski program za prevajanje internetnih strani v ustrezno obliko. Zaženemo ga s pomočjo ukazne vrstice z ustreznimi parametri. V našem primeru pomeni zastavica `-b`, da bomo ustvarili binarno datoteko s končnico (*.bin). Parameter WebPages je mapa, ki vsebuje vse datoteke, slike..., ki bi jih radi prikazali. Datoteka Server.bin je tako ustvarjena datoteka, ki je primerna za prenos na strežnik.

- Sedaj imamo vse pripravljeno za prenos datoteke na strežnik, od tega nas loči le klik na gumb (Prenos), ki vzpostavi povezavo s strežnikom in prenese željeno datoteko na strežnik.

```

C:\WINDOWS\system32\cmd.exe
F:\Diploma\SERVER>c:
C:\>FTP -i -s:F:\Diploma\SERVER\lolek.txt
ftp> OPEN 1.0.2.100
Povezan z 1.0.2.100.
220 Ready
Uporabnik (1.0.2.100:(none)):
331 Password required

230 Logged in
ftp> put F:\Diploma\SERVER\MP$Ing.bin
200 Ok
150 Transferring data...
#####
226 Transfer Complete
ftp: poslanih 6037 B v 0.00s 6037000.00KB/s.
ftp> bye
221 Bye

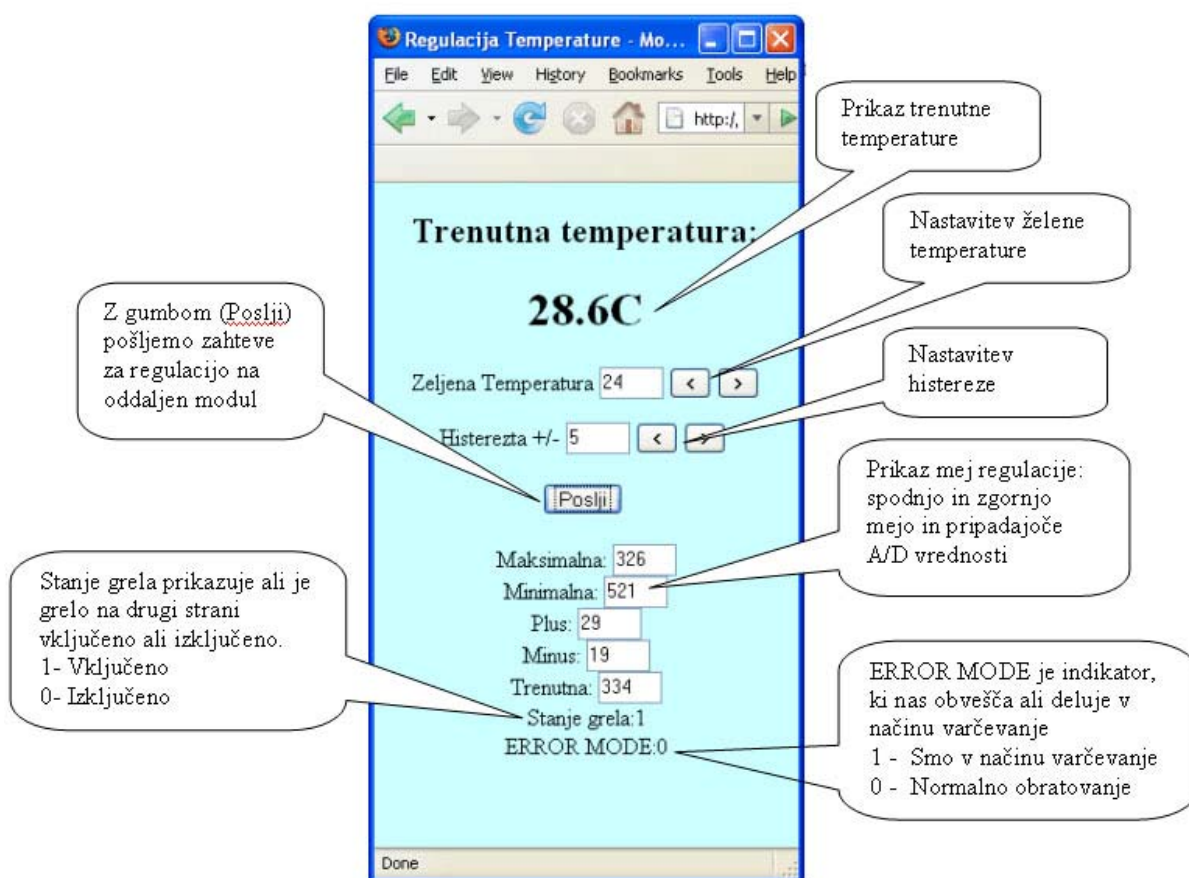
C:\>pause
Če želite nadaljevati, pritisnite poljubno tipko . . .

```

Slika 6.8: Prenos datoteke na strežnik

Tudi za ta korak smo napisali skriptno datoteko, ki najprej vzpostavi povezavo s strežnikom, se predstavi z ustreznim uporabniškim imenom in geslom in z ukazom PUT pošlje željeno datoteko na strežnik (slika zgoraj).

- Sedaj je potrebna samo še vzpostavitev povezave s strežnikom (klik na gumb Vzpostavitev Povezave)...Slika 6.9 prikazuje izgled pogovornega okna internetnega brskalnika.



Slika 6.9: Izgled pogovornega okna

Program se nahaja v mapi SERVER pod imenom Server.exe, vsebina ki jo prikažemo na strežniku pa v podmapi WebServer. Če želimo spremeniti vsebino internetne strani, je potrebno spremeniti vsebino mape WebServer, glavno stran pa poimenovati Index.html. S tako modifikacijo je program mogoče uporabiti za poljubno aplikacijo.

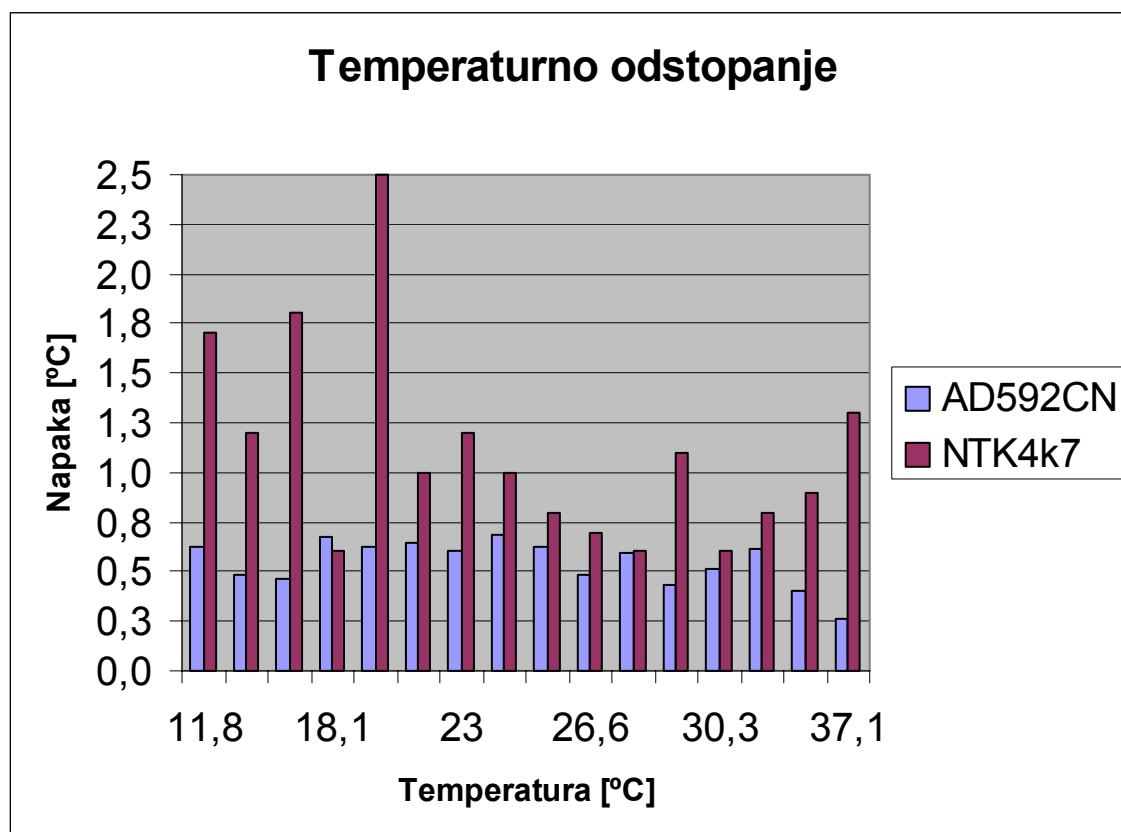
7 VERIFIKACIJA REZULTATOV

Za verifikacijo dobljenih rezultatov smo opravili nekaj primerjalnih meritev z uporabo referenčnega termometra z natančnostjo $\pm 0,1$ °C. V spodnji tabeli so prikazane vrednosti, ki smo jih dobili pri določenih temperaturah. V prvem stolpcu je temperatura, izmerjena z referenčnim termometrom, v drugem stolpcu je temperatura izmerjena s pomočjo NTK upora, ter v tretjem stolpcu pripadajoča absolutna napaka; temperatura, izmerjena s pomočjo senzorja AD592, in pripadajoča napaka sta podani v četrtem in petem stolpcu.

T [°C]	T _{NTK} [°C]	ΔT_{NTK} [°C]	T _{AD592} [°C]	ΔT_{AD592} [°C]
11,8	13,5	1,7	12,4	0,62
12,5	11,3	1,2	13,0	0,48
13,8	15,6	1,8	14,3	0,46
18,1	19,3	1,2	18,8	0,68
18,2	20,7	2,5	18,8	0,63
21,7	22,7	1	22,4	0,65
23	24,2	1,2	23,6	0,6
23,1	24,1	1	23,8	0,69
25	25,8	0,8	25,6	0,63
26,6	27,3	0,7	27,1	0,48
27,5	28,1	0,6	28,1	0,59
29	30,1	1,1	29,4	0,43
30,3	30,9	0,6	30,8	0,51
32,8	33,6	0,8	33,4	0,61
35,2	36,1	0,9	35,6	0,4
37,1	38,4	1,3	37,4	0,26

Tabela 3: Rezultati meritev

Spodnji graf prikazuje temperaturno odstopanje obeh metod.



Graf 6: Temperaturno odstopanje

Vidimo, da je za natančnejše merjenje temperature ustrežnejši temperaturni senzor AD592, saj lahko ob smotrni izbiri elementov dosežemo večjo natančnost. Merjenje temperature z uporabo NTK upora je primerno le za približno indikacijo temperature oz. za sisteme, ki ne zahtevajo visoke natančnosti. Iz grafa je tudi razvidno, da smo z uporabo NTK upora dobili precej manjše napake kot smo jih predvideli. Vendar to še ni razlog za trditev, da je tako izdelani termometer natančnejši, kot smo predvideli. V istih okoliščinah bi termometer, izdelan z enakimi elementi, lahko kazal večja odstopanja (posledice tolerance) in potrdil teoretične izračune.

8 ZAKLJUČEK

Cilj diplomske naloge je bil izdelati internetni modul, s katerim lahko prek interneta reguliramo in spremljamo temperaturo. Z verifikacijo natančnosti termometra smo ugotovili, da je le ta dovolj natančen, da ga lahko uporabimo za merjenje in regulacijo sobne temperature. Z nadaljnjimi prilagoditvami termometra bi lahko le tega uporabili tudi za natančnejše merjenje temperature. Izbrati bi bilo potrebno upore z manjšim temperaturnim koeficientom in operacijski ojačevalnik z dovolj majhnim napetostnim premikom. Kot regulacijski postopek smo izbrali dvostopenjsko (ON-OFF) regulacijo, ki je za sistem kot smo ga imeli dovolj dobra. Za zahtevnejše (kritične) sisteme bi si morali omisliti kakšen zahtevnejši regulacijski postopek.

Regulacija sobne temperature ni kritična v smislu ogrožanja človeških življenj v primeru neuspešne regulacije, vendar smo vseeno poskrbeli tudi za varnost v primeru napak. Tu imamo v mislih predvsem prekinitev povezave z internetom za daljši čas. V tem primeru ima uporabnik možnost določiti, ali se temperatura regulira v temperaturnem območju, ki je bilo nazadnje določeno ali pa v prej določenih mejah. Tak varnostni ukrep je dobrodošel predvsem s stališča varčevanja z energijo.

Z vedno hitrejšo rastjo interneta in s cenovno dostopnimi ponudbami postajajo tovrstne aplikacije vse zanimivejše. Ob neprestani rasti internetnega omrežja počasi zmanjkuje naslovnega prostora zato je v pripravi IPv6. Če pomislimo, da bo morda v prihodnje vse več proizvajalcev gospodinjskih aparatov in ostale opreme svoje naprave opremljevali s podobnimi moduli in nudili spletno krmiljenje in upravljanje je uporaba IPv6 skoraj obvezna.

9 LITERATURA

1. Borut Zupančič. *Zvezni regulacijski sistemi 1. del*. Založba FE in FRI, Ljubljana, 1992.
2. Stanislav Kovačič. *Komunikacije v avtomatiki (študijsko gradivo)*. http://vision.fe.uni-lj.si/classes/KA/Ka2001_Vse.pdf, Ljubljana 19. oktober 2001.
3. Andrew S. Tanenbaum. *Computer Networks THIRD EDITION*. Prentice-Hall International, Inc., USA, 1996.
4. Peter Šuhel. *Operacijski ojačevalniki v sistemih*. D design, Mengeš, 1995.
5. (avtor ni podan). *Leksikon Računalništva in Informatike*. Pasadena, Ljubljana, 2002.
6. Tone Vidmar. *Informacijsko komunikacijski sistemi*. Pasadena, Ljubljana, 2002.
7. (avtor ni podan). *Datasheet: 28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology*. Microchip Technology Inc., <http://ww1.microchip.com/downloads/en/DeviceDoc/...39626C.pdf>, dostopnost preverjena junija 2007.
8. (avtor ni podan). *Datasheet: ENC28J60, Stand-Alone Ethernet Controller with SPI Interface*. Microchip Technology Inc., <http://ww1.microchip.com/...downloads/en/DeviceDoc/39662b.pdf>, dostopnost preverjena junija 2007.
9. (avtor ni podan). *Datasheet: AD592 Low Cost, Precision IC Temperature Transducer*. Analog Devices, http://www.analog.com/UploadedFiles/...Data_Sheets/AD592.pdf, dostopnost preverjena junija 2007.
10. (avtor ni podan). *Datasheet: NTC thermistors, accuracy line 2322 640... 3/4/6*. BCcomponents, <http://eecs.oregonstate.edu/education/classes/...engr201/files/NTCThermistors.pdf>, dostopnost preverjena junija 2007.
11. Nilesh Rajbharti, *Datasheet: The Microchip TCP/IP Stack*. Microchip Technology Inc., <http://ww1.microchip.com/downloads/en/DeviceDoc/...39626C.pdf>, dostopnost preverjena junija 2007.

12. Lydia Parziale, David T. Britt, Chuck Davis, Jason Forrester, Wei Liu, Carolyn Matthews, Nicolas Rosselot. *TCP/IP Tutorial and Technical Overview*. IBM, <http://www.redbooks.ibm.com/redbooks/pdfs/...gg243376.pdf>, dostopnost preverjena junija 2007.
13. Borka Jerman-Blažič. *Internet. Založba Aleksander Štavs, Ljubljana, 1996*
14. Borut Breščak. *Sloji v TCP/IP*. http://www.s-sers.mb.edus.si/gradiva/w3/...omrezja/06_skladi/tcpip2.html, dostopnost preverjena junija 2007.
15. (avtor ni podan). *Slika ISO OSI*. <http://whatis.techtarget.com/digitalguide/...images/Misc/fsimage2a.jpg>, dostopnost preverjena junija 2007.
16. Borut Breščak. *Enkapsulacija podatkov*. http://www.s-sers.mb.edus.si/...gradiva/w3/omrezja/06_skladi/enkapsulacija.html, dostopnost preverjena junija 2007.
17. (avtor ni podan). *Slika IP naslavljanje*. http://www2.rad.com/networks/...1994/ip_addr/pic2.gif, dostopnost preverjena junija 2007.
18. Borut Breščak. *Protokol za prevedbo naslova*. http://www.s-sers.mb...edus.si/gradiva/w3/omrezja/36_osi3/arp.html, dostopnost preverjena junija 2007.
19. Gerhard M. Glaser. *Slika TCP – Header*. <http://www.tcp-ip-info.de/tcp...-ip-schulung/img106.gif>, dostopnost preverjena junija 2007.
20. Borut Breščak. *Podomreževanje*. http://www.s-sers.mb.edus.si/...gradiva/w3/omrezja/36_osi3/podomrezja.html, dostopnost preverjena junija 2007.
21. Borut Breščak. *Internet Contrl Message Protocol*. http://www.s-sers...mb.edus.si/gradiva/w3/omrezja/36_osi3/icmp.html, dostopnost preverjena junija 2007.
22. Borut Breščak. *Transmision Control Protocol*. http://www.s-sers.mb...edus.si/gradiva/w3/omrezja/40_transportni/tcp.html, dostopnost preverjena junija 2007.
23. Borut Breščak. *Slika FTP*. http://www.s-sers.mb.edus.si/gradiva/w3.../omrezja/60_storitve/ftp/ftp.gif, dostopnost preverjena junija 2007.
24. (avtor ni podan). *Nelinearni upori- Termistorji*. http://www.student-info.net/fe/fgg/baza-student/vss2/21202/datoteke/ntk-ptk_2004.pdf, dostopnost preverjena junija 2007.

25. (avtor ni podan). *Slika SPI komunikacija*. <http://www.totalphase.com/...support/articles/article03/spi-diagram.png>, dostopnost preverjena junija 2007.
26. Ivica Gjorgjiev. *Komunikacije – SPI*. http://www.mikroelektronika.co.yu/sr.../magazine/8broj/8br_11.htm, dostopnost preverjena junija 2007.
27. (avtor ni podan). *Slika releja*. http://www.mikroe.com/en/books/picbook/...7_chapter/15.gif, dostopnost preverjena junija 2007.
28. Franc Bergelj. *Meritve 1. del*. založba FE in FRI, Ljubljana, 1997
29. Mrhar Peter. *Uvod v HTML*. Flamingo, Nova Gorica, 1999
30. Wagner Richard. *JavaScript unleashed*. Sam sNet, Indianapolis, 1997
31. Aitken, Peter G. *Visual Basic 6 Programming*. Scottsdale, Coriolis 1998
32. (avtor ni podan). *Datasheet: TL081CN General purpose JFET single operational amplifiers*. ST, <http://www.ortodoxism.ro/datasheets/...stmicroelectronics/2299.pdf>, dostopnost preverjena junija 2007.
33. Peter šuhel, Boštjan Murovec. *Knjiga v elektronski obliki: Industrijska elektronika Elementi*. http://lie.fe.uni-lj.si/literatura/esd2005.../Knjiga_ESD.pdf, Ljubljana, 2005, dostopnost preverjena junija 2007.

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.

Vrhovlje, 12. junij 2007

Peter Vitez

Priloga:

Seznam uporabljenih elementov za izgradnjo modula:

<i>Količina</i>	<i>Proizvajalec</i>	<i>Oznaka v shemi</i>	<i>Vrednost</i>
1	Microchip	U2	PIC18F4620
1	Microchip	U3	ENC28J60
1	Microchip	U5	25LC256
1	Bourns	CON2	PT44201D
1		CON1	Konektor
1		U1	LM2594HVM-5,0
1		Q1	LM1117DT-3,3
5		R1,R4,R5,R11,R12	470 Ω
1		R2	1,2 k Ω
1		R3	NTK 4,7 k Ω
4		R6,R7,R8,R9	10 k Ω
1		R10	2,7 k Ω
4		R13,R14,R15,R16	49,9 Ω , 1%
1		C1	47 uF/25 V
17		C2,C3,C5,C7,C10,C11,C25, C26,27,C28,C18,C19,C20, C21,C22,C23,C24	100 nF/100 V
1		C4	100 uF/25 V
1		C6	47 uF/25 V
2		C8, C9	15 pF
2		C12, C13	18 pF
1		C15	10 uF
1		L1	220 uH/0,8 A
1		L2	Ferit
1		D1	BYD37
3		D2,D3,D4	LED DIODA
1		ZD1	BZV55
3		T1,T2,T3	Tipka mini 6x6
1		X1	Kristal 10 MHz
1		X2	Kristal 25 MHz
2		AND1, AND2	74AHC1G08
3		JP1, JP2, JP3	Jumper
1		JP4	Letvica 1x2
1		JP5	Letvica 1x6
1		JP6	Letvica 1x5
1		JP7	Letvica 1x8
1		JP8	Letvica 1x3

Opomba:

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