

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

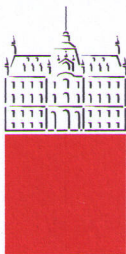
Luka Šepetavc

**Sistem za brezžičen zajem podatkov
z možnostjo nadzora preko spleta**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2009



Št. naloge: 01547/2009

Datum: 15.03.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **LUKA ŠEPETAVC**

Naslov: **SISTEM ZA BREŽIČEN ZAJEM PODATKOV Z ZMOŽNOSTJO
NADZORA PREKO SPLETA**


**SYSTEM FOR WIRELESS DATA ACQUISITION WITH WEB
SUPERVISION CAPABILITY**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Analizirajte možne rešitve za izgradnjo sistema, ki je sposoben brezžično zajemati podatke z oddaljenih senzorjev tako, da so dostopni tudi preko spleta. Sistem naj vsebuje tudi senzorski del, ki naj bo zasnovan tako, da lahko dalj časa deluje z baterijskim napajanjem. Razvijte in izdelajte strojno in programsko opremo vseh delov sistema. Kot osnovo uporabite mikrokontroler serije PIC18F, kot spletni strežnik pa brezplačen programski paket Microchip TCP/IP Stack. Sistem preizkusite in preverite pravilnost njegovega delovanja.

Mentor:


prof. dr. Dušan Kodek



Dekan:


prof. dr. Franc Solina

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Luka Šepetavc,

z vpisno številko 63040155,

sem avtor diplomskega dela z naslovom:

Sistem za brezžičen zajem podatkov z zmožnostjo nadzora preko spleta

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Dušana Kodeka
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 26.05.2009

Podpis avtorja:

Zahvala

Za pomoč pri izdelavi naloge se zahvaljujem mentorju, prof. dr. Dušanu Kodeku.

Kazalo

Povzetek	1
Abstract	2
1 Uvod	3
2 Zasnova in načrtovanje	4
2.1 Mrežni modul	5
2.1.1 Mikrokrmilniški del	5
2.1.2 Mrežni del	7
2.2 Sprejemni modul	9
2.3 Oddajni modul	10
3 Programska oprema	13
3.1 Zajem podatkov	13
3.2 Brežžična povezava	16
3.2.1 Kodiranje in dekodiranje	16
3.2.2 Protokol	20
3.3 Spletni strežnik	23
3.3.1 Prikaz dinamičnih podatkov	24
3.3.2 Zgradba spletne strani	25
3.4 Druge naloge	26
3.4.1 Sprejem in pretvorba podatkov	27
3.4.2 Dnevnik dogodkov	29
4 Primeri uporabe	31
4.1 Oddajni modul	31
4.2 Mrežni modul	32
4.3 Spletna stran	33

5	Sklepne ugotovitve	35
	Dodatek A Električne sheme modulov	37
	Dodatek B Fotografiji sistema	41
	Dodatek C Zaslonska slika spletne strani	43
	Literatura	45

Seznam uporabljenih kratic in simbolov

A/D — Analog-to-Digital

ADC — Analog-to-Digital Converter

AGC — Automatic Gain Control

ASK — Amplitude Shift Keying

CSS — Cascading Style Sheets

DHCP — Dynamic Host Configuration Protocol

EEPROM — Electrically Erasable Programmable Read-Only Memory

HTML — Hypertext Markup Language

HTTP — Hypertext Transfer Protocol

IP — Internet Protocol

LCD — Liquid Crystal Display

LED — Light-Emitting Diode

LSB — Least Significant Byte

MAC — Media Access Control

MSB — Most Significant Byte

NTP — Network Time Protocol

PDIP — Plastic Dual-In-line Package

PHY — Physical Layer

SMD — Surface Mount Device

SNTP — Simple Network Time Protocol

SPI — Serial Peripheral Interface Bus

TFTP — Trivial File Transfer Protocol

XML — Extensible Markup Language

Povzetek

Diplomsko delo govori o načrtovanju in izdelavi sistema, ki je sposoben zajemati različne podatke o okolju, jih brezžično prenesti na drugo lokacijo ter jih predstaviti na zaslonu in na spletni strani. V delu sta najprej opisana načrtovanje in zgradba električnih vezij. Predstavljeni so razlogi za izbiro posameznih elementov in njihova vloga v sistemu. Celoten sistem je razdeljen na več fizičnih vezij, oziroma modulov, ki opravljajo posamezne funkcije sistema. V splošnem je razdeljen na oddajni, sprejemni in mrežni modul, vsak izmed njih pa vsebuje lasten mikrokontroler serije PIC18F.

V drugi polovici dela je predstavljena zasnova programske opreme. Oddajni in sprejemni modul vsebujeta povsem lastno napisan program, ki med drugim skrbi za meritve temperature, relativne vlage, napetosti, ugotavljanje stanja stikala in kodiranje ter dekodiranje podatkov pri brezžičnem prenosu. Osnova programa mrežnega modula temelji na brezplačnem programskem paketu Microchip TCP/IP Stack, ki vsebuje sklad TCP/IP in celoten spletni strežnik. Dodane pa so funkcije za sprejem, obdelavo, hranjenje in prikaz podatkov, ki jih je možno spremljati na zaslonu in prek spletne strani.

Zadnje poglavje opisuje delovanje sistema in primere uporabe. V zaključku je sistem še ovrednoten glede na podane zahteve, naštetih pa je tudi nekaj možnihboljšav in nadgradenj.

Ključne besede:

zajem podatkov, brezžični prenos, PIC18F, kodiranje, spletni strežnik

Abstract

The thesis describes a design and realization of a system that is capable of acquiring data about environment, wirelessly transferring the data to another location, and presenting it on a screen and on a website. The description begins with a design and structure of electrical circuits. The reasons that lead to the choice of different elements and their role in the system are presented next. The system is divided into different electrical circuits or modules. They are transmitting, receiving, and network module. Each of them contains a PIC18F series microcontroller.

Design of software is described in the second half of the thesis. Custom programs were written for the transmitting and receiving modules. Among other tasks, the software is responsible for measuring the temperature, relative humidity, voltage, reading the state of the switch, and encoding and decoding of data which are transmitted over wireless link. Program of the network module is based on Microchip's TCP/IP Stack which also contains an implementation of a complete web server. On top of that, several other functions were added. They serve for reception, handling, storing and presentation of data that can be monitored on a screen or through a website.

Last chapter describes how the system works and presents different use case examples. At the end, system is assessed against given requirements. Also, some possible improvements and upgrades are given.

Key words:

data acquisition, wireless transfer, PIC18F, encoding, web server

Poglavje 1

Uvod

Sistemi za brezžično spremljanje informacij z oddaljenih lokacij so v zadnjem času precej aktualni. Največkrat se pojavljajo v povezavi s pametnimi hišami, vremenskimi postajami in različnimi industrijskimi aplikacijami. Privlačnost takšnih sistemov je predvsem v tem, da omogočajo spremljanje informacij z več različnih lokacij, pri tem pa ni potrebe po fizični navzočnosti uporabnika na teh mestih. Zmožnost brezžičnega prenosa podatkov pa še poveča njihovo fleksibilnost in uporabnost. Praktični primeri njihove uporabe so na primer spremljanje podatkov o vremenu, različne informacije o varnosti (npr. odprta vrata, okna), prenos slike s kamere ipd.

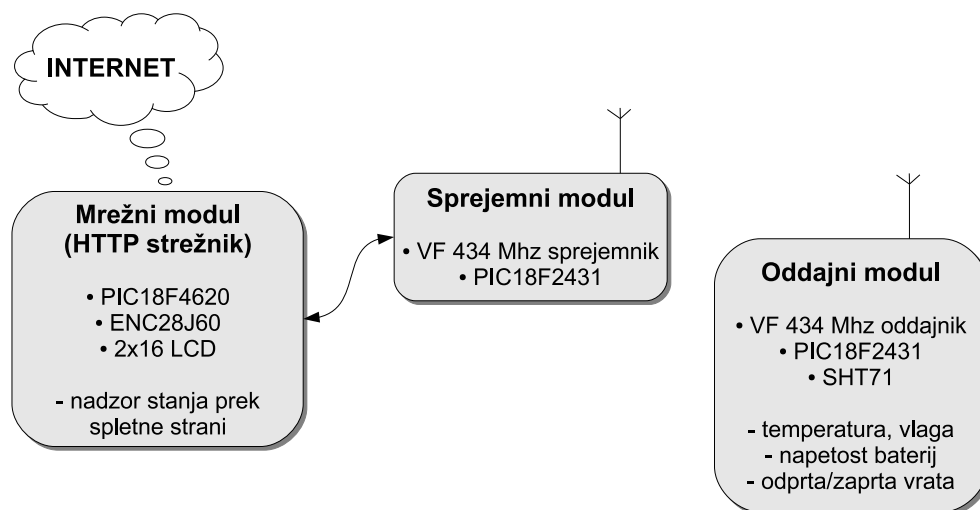
Takšni sistemi so v grobem sestavljeni iz dveh delov, iz oddajnega in sprejemnega. Oddajni del skrbi za zajem informacij iz okolice (največkrat z različnimi senzorji) in prenos teh informacij do sprejemnika. Sprejemni del pa poleg sprejema informacij poskrbi za njihovo obdelavo in/ali predstavitev. Bolj podrobne naloge so opredeljene z uporabnikovimi zahtevami. Zahteve pa hkrati določajo tudi druge lastnosti sistema, kot so zanesljivost delovanja in prenosa podatkov, zmogljivost, odzivnost in nenazadnje tudi ceno.

Za diplomsko nalogo sem se odločil načrtovati in izdelati sistem, ki omogoča zajem nekaj osnovnih podatkov o okolju (temperatura, relativna vlaga, odprta/zaprta vrata) ter jih brezžično posredovati sprejemniku. Ta jih mora predstaviti na vgrajenem zaslonu in prek spletne strani. Oddajnik naj bi se napajal z baterijami, zato mora biti energijsko učinkovit, njegov doseg pa mora zadoščati razdaljam v okviru enega stanovanja ali nekaj deset metrov na prostem. Sprejemni del mora vsebovati samostojen spletni strežnik ter zmožnost shranjevanja informacij o pomembnih dogodkih. Zanesljivost delovanja sistema ni kritična, pomembna pa je njegova cenovna dostopnost in relativna enostavnost izdelave v samogradnji.

Poglavje 2

Zasnova in načrtovanje

Sistem kot celota je relativno kompleksen, zato je bila pri načrtovanju izbrana modularna zasnova. Oddajnik je sicer že zaradi tega, ker je brezžičen, fizično ločen od sprejemnika, in je zato obravnavan kot samostojen modul. Tudi njegova zasnova je relativno enostavna, zato ni deljen na manjše enote. Drugače je pri sprejemnem delu, ki je zaradi kompleksnosti sestavljen iz več modulov oz. fizičnih vezij. V osnovi je razdeljen na sprejemni in mrežni modul. Podrobneje pa se mrežni modul deli še na dva dela, mikrokrmilniški in mrežni. Takšna razdelitev je olajšala predvsem izdelavo električnih vezij, obenem pa se je poenostavil tudi razvoj sistema. Slika 2.1 prikazuje konceptualno shemo celotnega sistema.

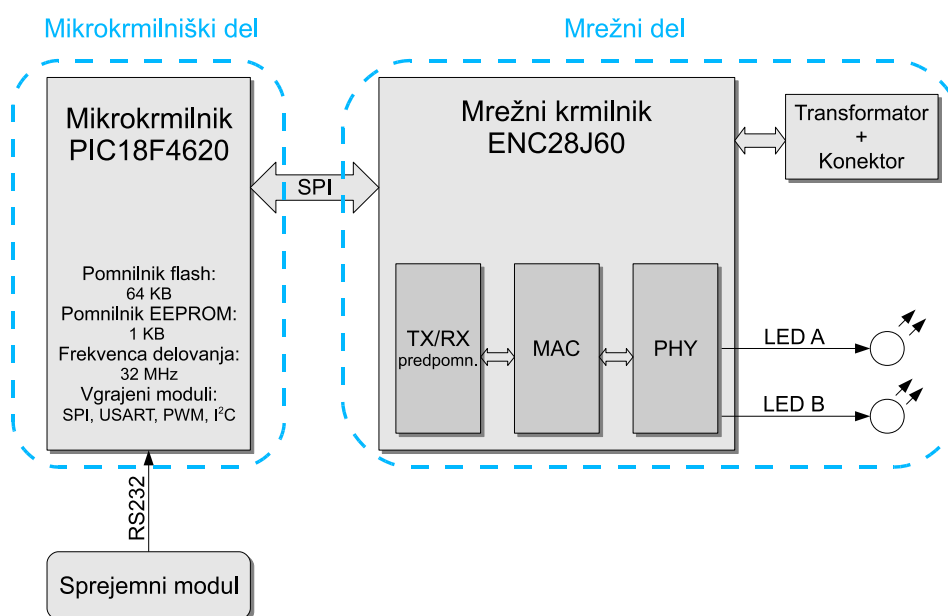


Slika 2.1: Konceptualna shema celotnega sistema, razdeljenega na module.

V tem poglavju sta opisana zasnova in načrtovanje posameznih modulov tako, kot sta si kronološko sledila. Predstavljene so uporabljene komponente in sheme posameznih električnih vezij. Opis se začne z mrežnim, nadaljuje pa s sprejemnim in oddajnim modulom.

2.1 Mrežni modul

Mrežni modul je razdeljen na dva dela, na mikrokrmilniški in mrežni. Podrobnejša shema te delitve je prikazana na Sliki 2.2. V nadaljevanju sledi opis.



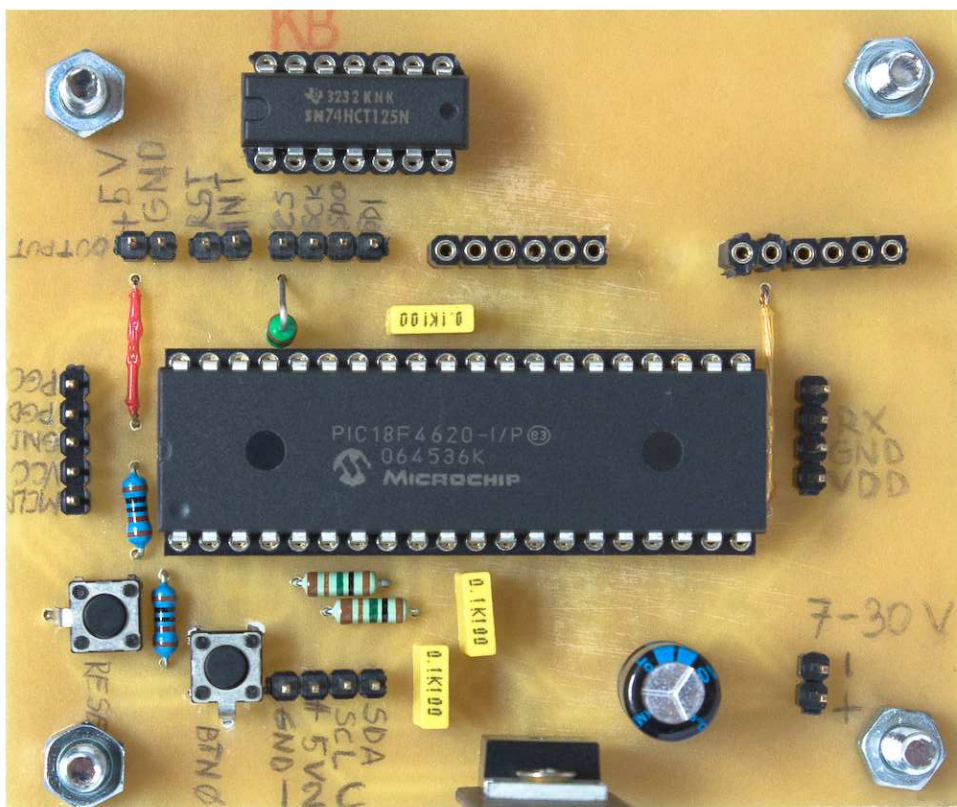
Slika 2.2: Konceptualna shema podrobnejše delitve mrežnega modula na mikrokrmilniški in mrežni del.

2.1.1 Mikrokrmilniški del

Mikrokrmilniški del mrežnega modula je osnovan na 8-bitnem mikrokrmilniku PIC18F4620 [10] podjetja Microchip Technology. Eden glavnih vzrokov za njegovo izbiro je zadostna količina pomnilnika flash, ki znaša 64 KB. Ta še ravno zadošča za uporabo Microchipove implementacije sklada TCP/IP, ki je uporabljena pri razvoju sistema. Pomembno lastnost predstavlja tudi vgrajen strojni

modul za serijsko komunikacijo SPI (Serial Peripheral Interface Bus), saj je ta vrsta komunikacije nujno potrebna pri povezovanju z izbranim mrežnim krmilnikom ENC28J60. Prednost tega mikrokrmilnika je tudi v njegovi dostopnosti v različnih oblikah ohišij. Izbrani čip je v ohišju tipa PDIP (Plastic Dual-In-line Package), katero ima v primerjavi z drugimi oblikami večje razdalje med nožicami in tako poenostavi izdelavo vezij ter ročno spajkanje elementa. K lažjemu razvoju programske opreme pa pripomore tudi to, da proizvajalec za omenjeno serijo mikrokrmilnikov ponuja prevajalnik za programski jezik C.

Električna shema vezja je predstavljena na Sliki A.1, fotografija izdelanega vezja pa na Sliki 2.3. Za stabilno napajanje mikrokrmilnika skrbi linearni



Slika 2.3: Fotografija mikrokrmilniškega dela z mikrokrmilnikom PIC18F4620.

napetostni regulator L7805C, ki po priporočeni shemi [6], skupaj z dvema kondenzatorjema, uravnava izhodno napetost na 5 V. Ker se mikrokrmilniški del povezuje z mrežnim delom, ta pa se napaja z napetostjo 3,3 V, je med njima treba poskrbeti za prilagajanje logičnih nivojev. Problematična je smer med

izhodi mrežnega krmilnika in vhodi mikrokrmilnika. Vhodi slednjega namreč zahtevajo nivoje CMOS, torej je za predstavitev logične 1 treba zagotoviti napetost vsaj 4,0 V [10] str. 335. Mrežni krmilnik pa lahko na izhodu zagotovi največ 3,3 V [5], kar je premalo. Problem je rešljiv na več načinov. Proizvajalec mrežnega krmilnika predlaga uporabo vrat AND ali 3-state, v praksi pa se uporabljajo tudi rešitve s tranzistorji FET, upori pull-up ipd. [3]. Izbrana je bila rešitev s čipom SN74HCT125N [13], ki vsebuje štiri neodvisna vrata 3-state. Ta namreč delujejo v skladu z nivoji TTL in na vhodu za logično 1 zahtevajo napetost vsaj 2 V, na izhodu pa zagotovijo napetost najmanj 4,4 V. Takšni napetostni nivoji pa so povsem v skladu z določenimi zahtevami. Nivojev v nasprotni smeri ni treba prilagajati, saj so vhodi mrežnega krmilnika tolerantni na napetost 5 V.

Za neposreden prikaz podatkov je na modul priklopljen zaslon LCD (Liquid Crystal Display). Uporabljen je precej razširjen tip zaslona, kompatibilen s standardom HD44780, s paralelnim vmesnikom in zmožnostjo prikaza dveh vrstic po šestnajst znakov. Konkreten model ima oznako DEM 16216 SYH-PY in ima tudi funkcijo osvetlitve zaslona. Sicer pa je na vezju še nekaj drugih komponent, kot na primer konektorji za priklop napajanja, mrežnega dela, sprejemnega modula ter programatorja. Vgrajeni sta tudi dve tipki, z eno mikrokrmilnik ponastavimo (tipka reset), drugi pa lahko dodelimo poljubno programsko funkcijo.

2.1.2 Mrežni del

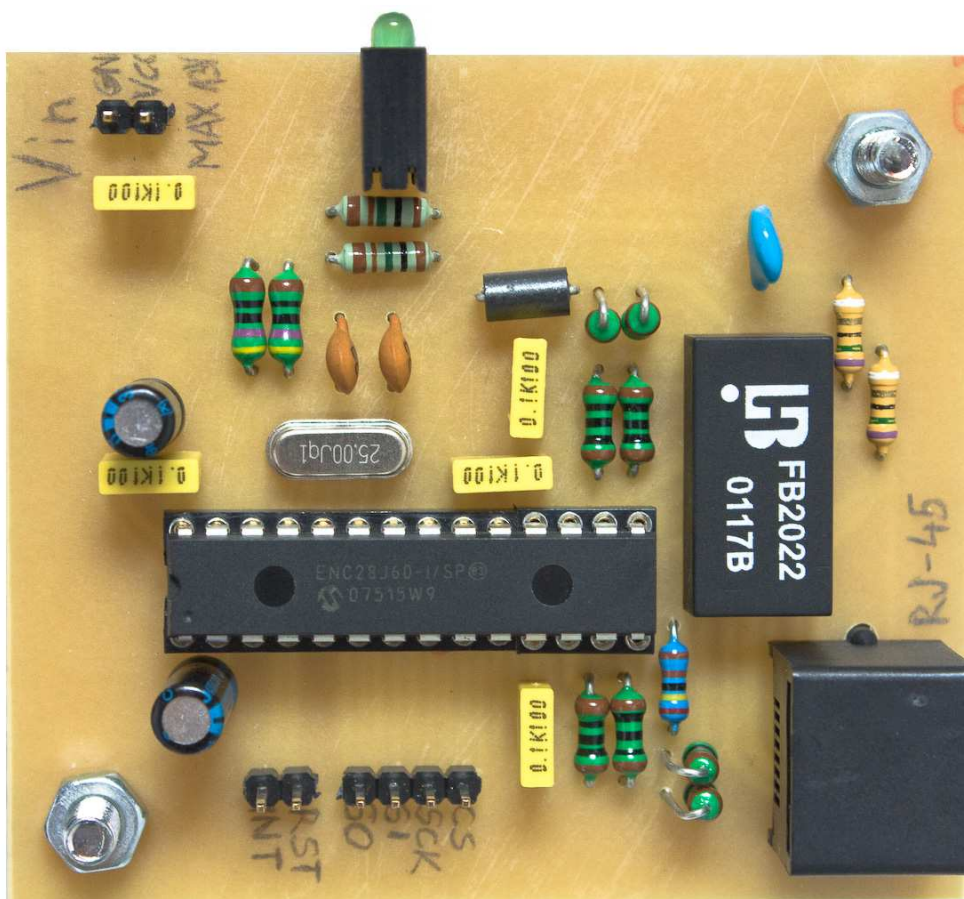
Osnova tega dela je Microchipov mrežni krmilnik ENC28J60, ki implementira 2 najnižji plasti po Ethernet standardu IEEE 802.3. To sta PHY (Physical Layer, sl. fizična plast) in MAC (Media Access Control, slov. krmiljenje dostopa do medija). Vsebuje tudi 8 KB predpomnilnika za vhodne in izhodne podatkovne pakete ter vmesnik SPI za komunikacijo z gostiteljem. Gostitelj je lahko poljuben mikrokrmilnik, ki pa mora za uspešno povezovanje v omrežja ethernet zagotoviti lastno implementacijo sklada TCP/IP.

Shema na Sliki A.2 v veliki meri temelji na priporočeni shemi proizvajalca [5] str. 7. Večina elementov, kot so oscilator, transformator in pripadajoči upori, je nujno potrebna za pravilno delovanje mrežne komunikacije. Elementi, na primer dušilka s feritnim jedrom, upora in kondenzator na strani mrežnega konektorja, pa so namenjeni odpravljanju elektromagnetnih motenj. Za prikazovanje stanja mrežnega vmesnika sta na vezju diodi LED (Light-Emitting Diode). Z eno krmilnik naznanja prisotnost mrežne povezave, z drugo pa prenos podatkov. Ker krmilnik za svoje delovanje zahteva napetost 3,3 V, po-

trebuje ustrezen napajalni del. Uporabljen je linearni napetostni regulator z zelo nizkim padcem LD29080DT33R [7], in lahko kot vhodno napetost sprejme tisto iz mikrokrmilniškega dela, ki znaša 5 V.

Nekatere elemente za izdelavo mrežnega dela, ki so v prosti prodaji relativno težko dostopni, je možno najti na odsluženi mrežni kartici ali modemu. Na ta način je bil izbran in uporabljen ustrezen transformator, visokonapetostni kondenzator ter dušilka s feritnim jedrom. Vezje pa se lahko še nekoliko poenostavi z uporabo ustreznega konektorja RJ45 z vgrajenim transformatorjem.

Fotografija mrežnega dela je na Sliki 2.4



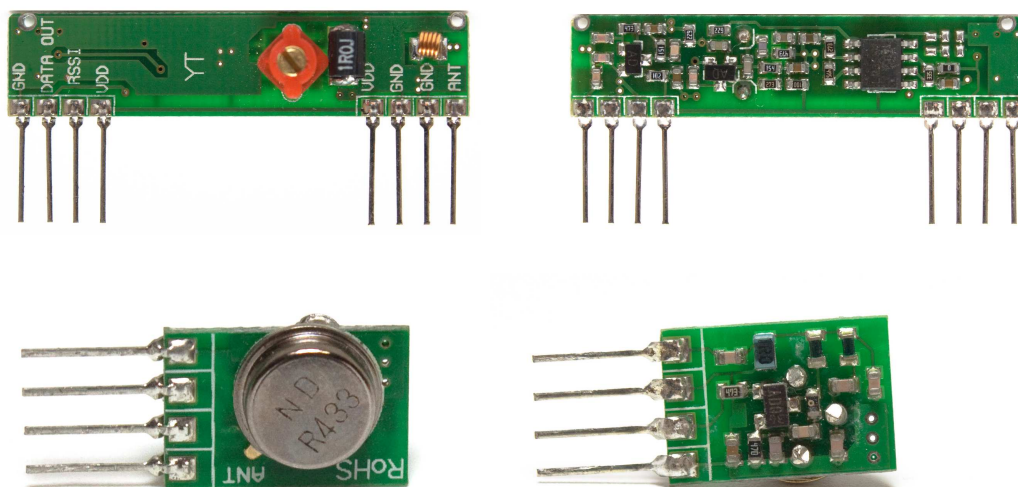
Slika 2.4: Fotografija mrežnega dela z mrežnim krmilnikom ENC28J60.

2.2 Sprejemni modul

Tako sprejemni kot oddajni modul uporabljata isti model mikrokrmilnika, PIC18F2431. Odločitev zanj temelji na dveh razlogih. Prvi je omenjen že pri opisu mrežnega modula, in sicer da je za serijo PIC18F na voljo prevajalnik za programski jezik C. Drugi razlog pa je njegova zmogljivost oziroma številčnost vgrajenih modulov ter vhodno/izhodnih nožic (teh je 28), ki omogočajo priklop številnih senzorjev, zaslona in drugih perifernih naprav.

Razlog, da tudi sprejemni modul vsebuje lasten mikrokrmilnik, je na eni strani v izbiri tipa sprejemnika, na drugi pa v fleksibilnosti takšnega modula. Na ta način je namreč neodvisen in ga je možno povezati s poljubnim drugim modulom ali sistemom. Omogoča pa tudi nadgradnje z dodajanjem novih funkcij.

Bistven del sprejemnega modula je brezžični sprejemnik. Glavno vodilo pri njegovi izbiri je bila cena. Tako je bila sprejeta odločitev za ločen sprejemnik in pripadajoč oddajnik podjetja Laipac, z oznakama RLP434 in TLP434A [12, 14]. Njuni fotografiji sta na Sliki 2.5.

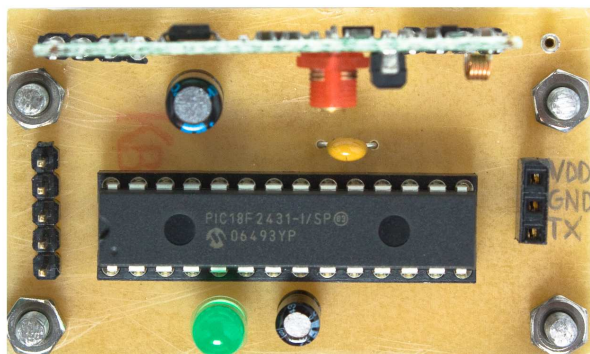


Slika 2.5: Fotografiji sprejemnika RLP434 (zgoraj) in oddajnika RLP434A (spodaj).

Gre za brezžična modula nizke moči, ki delujeta na frekvenci 433,92 MHz in uporabljata amplitudno modulacijo. Dovoljujeta prenos podatkov do največje hitrosti 4800 b/s in imata doomet na prostem do približno 100 m. Zanju je značilno, da ne vsebujeta kodirnika/dekodirnika ali kakšne druge logike, ki bi

skrbela za zanesljivost prenosa podatkov. To je treba zagotoviti z ustrezno programsko rešitvijo ali strojno opremo. O tem je več napisanega v poglavju o programski opremi.

Električna shema sprejemnega modula je prikazana na Sliki A.3, fotografija vezja pa na Sliki 2.6. Svojega napajalnega dela ne potrebuje, ker dobiva tok s



Slika 2.6: Fotografija sprejemnega modula z mikrokrmilnikom PIC18F2431 in sprejemnikom RLP434.

stabilizirano električno napetostjo iz mrežnega modula, vsi elementi pa delujejo na napetosti 5 V. Zaradi lažjega razvoja ter možnosti nadgrajevanja programa je na vezju tudi konektor za programiranje mikrokrmilnika. Dioda LED prikazuje sprejetje veljavnega paketa podatkov. Poleg ležišča za sprejemnik je tudi priključek za anteno. To je možno izdelati iz izolirane žice, dolge eno četrtino valovne dolžine ($\lambda/4$). Točna dolžina takšne antene, z upoštevanjem frekvence 433,92 MHz in korekcijskega faktorja 0,95, znaša 16,4 cm [1].

Fotografija celotnega strežniškega sistema, v katerem sta sprejemni in mrežni modul ter zaslon LCD, je na Sliki B.1.

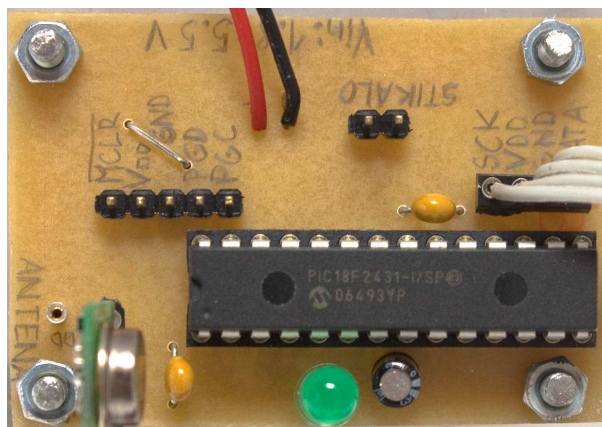
2.3 Oddajni modul

Ena izmed postavljenih zahtev za oddajni modul je ta, da mora biti energijsko učinkovit, saj se napaja z baterijami. Zahteva je v prvi vrsti izpolnjena tako, da je uravnavanje napetosti izvedeno s tako imenovanim *charge-pump* regulatorjem, ki je bistveno bolj učinkovit od običajnega linearnega napetostnega regulatorja. Gre za tip DC/DC-pretvornika, ki deluje na podlagi kapacitivnosti, torej z uporabo kondenzatorjev. Izbran je čip podjetja Maxim, z oznako MAX1595EUA50, ki pri vhodni napetosti od 1,8 V do 5,5 V zagotavlja tok do

125 mA in stabilizirano izhodno napetost 5 V. Njegova učinkovitost je odvisna od vhodne napetosti in porabe bremena. V primeru oddajnega modula, kjer vhodna napetost znaša 2,5 V, izhodni tok pa 10 mA, je učinkovitost regulatorja 80 %. Pri tem izračunu so upoštevani podatki iz dokumentacije elementa [11] str. 3, graf *toc05*. Izbira vrednosti kondenzatorjev, ki so potrebni za delovanje regulatorja, ter njihova vezava, je izvedena po priporočeni shemi [11] str. 5, Figure 4.

K večji učinkovitosti porabe energije pripomoreta tudi mikrokrmilnik in oddajnik. Prvi omogoča funkcijo spanja, v kateri se izvajanje programa in delovanje modulov ustavi, poraba energije pa pade na nekaj mikroamperov. Oddajnik pa ima to lastnost, da deluje samo takrat, ko je na podatkovnem vhodu logična 1 oziroma napetost v intervalu od $V_{cc}-0,5$ V do $V_{cc}+0,5$ V. Pri napajalni napetosti 5 V je to med 4,5 V in 5,5 V. Če na vhod postavimo logično 0 oziroma napetost, nižjo od 0,3 V, oddajnik ne deluje, poraba pa je takrat zanemarljiva.

Celotna električna shema oddajnega modula je prikazana na Sliki A.4 in se ne razlikuje veliko od tiste od sprejemnega modula. Tudi tu je na voljo konektor za programiranje, dioda LED za indikacijo poslanega paketa, ležišče oddajnika s priključkom za anteno ter mikrokrmilnik. Vsi ti elementi so vidni tudi na fotografiji modula, ki je predstavljena na Sliki 2.7. V nasprotju s



Slika 2.7: Fotografija oddajnega modula z mikrokrmilnikom PIC18F2431 in oddajnikom TLP434A.

sprejemnim modulom je tu dodan senzor za merjenje temperature in relativne zračne vlage, z oznako SHT71. To je digitalni senzor, ki izmerjene podatke pošilja gostitelju prek serijskega vodila, podobnega I²C. Temperaturo meri v razponu od -40 °C do +123,8 °C s točnostjo $\pm 0,4$ °C, relativno vlažnost zraka

pa v območju od 0 % do 100 % s točnostjo $\pm 3,0$ %. Ločljivost izmerjenih količin je 14 bitov za temperaturo in 12 bitov za relativno vlago.

Merjenje napetosti baterije je izvedeno z analogno-digitalnim pretvornikom, ki je vgrajen v mikrokrmilnik. Referenčno napetost v tem primeru predstavlja izhodna napetost regulatorja. Na modul je možno priključiti še zunanje stikalo s poljubno funkcija, na primer spremljanje stanja vrat (odprta/zaprta).

Na Sliki B.2 je fotografija celotnega oddajnega modula z napajalnimi baterijami.

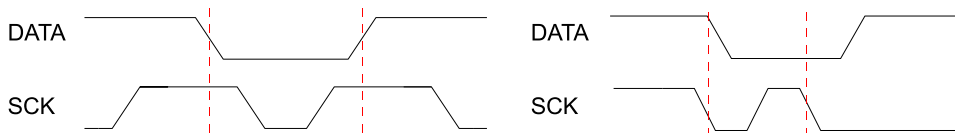
Poglavje 3

Programska oprema

Vsi trije moduli imajo lasten mikrokrmilnik, na vsakem od njih pa teče lasten program. V tem poglavju sta opisana razvoj in zgradba programske opreme vsakega izmed teh modulov. Opisi niso strogo ločeni po fizičnih enotah, temveč po smiselnih programskih enotah. Tako je najprej opisan zajem podatkov pri oddajnem modulu, nato realizacija brezžičnega prenosa podatkov med njim in sprejemnim modulom, nato sledi komunikacija med sprejemnim in mrežnim modulom, na koncu pa še opis spletnega strežnika in drugih nalog mrežnega modula. Tak vrstni red opisa sledi tudi naravni poti, po kateri tečejo podatki od zajema do njihovega prikaza.

3.1 Zajem podatkov

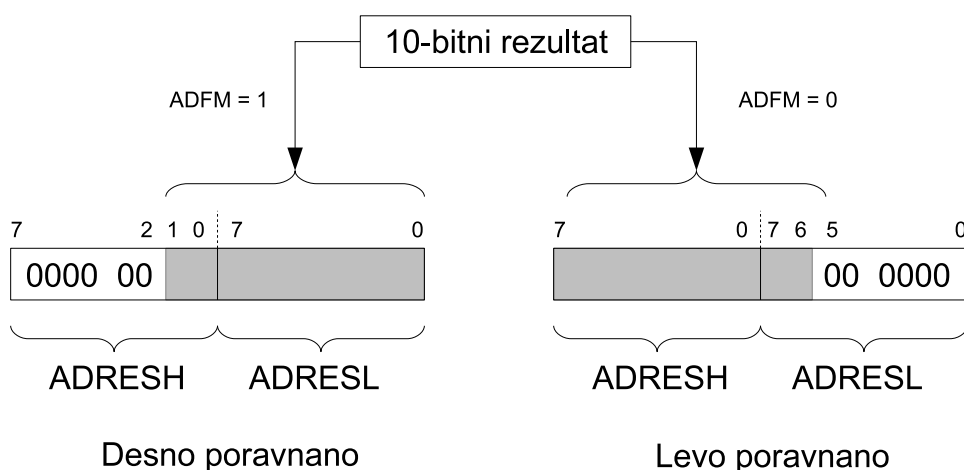
Zajem podatkov se začne z meritvijo temperature in relativne vlage z uporabo senzorja SHT71. Ta senzor uporablja serijski protokol, ki je precej podoben I²C. Od njega se razlikuje samo v ukazu START in bitu ACK, ki sta prikazana na Sliki 3.1. Za komunikacijo se uporabljata 2 signala: SCK, ki predstavlja uro, in DATA, ki predstavlja podatkovni signal. Senzor je v tem protokolu suženj (angl. slave), mikrokrmilnik pa gospodar (angl. master). Vsak prenos podatkov začne gospodar, in sicer tako, da najprej izvede ukaz START, nato pa pošlje naslovno-ukazni bajt. Zgornji trije biti naslovno-ukaznega bajta vsebujejo naslov naprave, spodnjih pet pa ukaz, ki naj ga izvede. Senzor po prejetju ukaza izvede naloženo opravilo, nato pa sporoči stanje pripravljenosti s potrditvenim bitom ACK. Če je na primer senzor moral opraviti neko meritev, to pomeni, da lahko gospodar po prejetju bita ACK začne sprejemati podatke. Ta poteka tako, da gospodar upravlja s signalom SCK, suženj pa s signalom DATA. Ob vsaki visoki urini periodi gospodar prebere stanje signala DATA in



Slika 3.1: Potek ukaza START (levo) in potrditvenega bita ACK (desno) pri serijskem protokolu senzorja SHT71.

tako sestavlja podatkovni bajt od najpomembnejšega bita naprej. Gospodar vsak uspešen sprejem bajta zaključi s potrditvenim bitom ACK. V primerih, ko je vrednost rezultata zapisana z več kot osmimi bitmi, mora senzor poslati dva bajta, rezultat v njiju pa je desno poravnan. Če gospodar prejetje zadnjega podatkovnega bajta potrdi z ACK, senzor pošlje še bajt *checksum*, s katerim se preveri pravilnost prenesenih podatkov. Postopka za merjenje temperature in relativne vlage sta skoraj enaka. Naslov je v obeh primerih isti in ima v dvojiškem zapisu vrednost 000. Razlikujeta se le v ukazih, ki sta dvojiško predstavljena kot 00011 za temperaturo in 00101 za relativno vlago.

Napetost baterij, s katerimi se napaja oddajni modul, se meri z analogno-digitalnim pretvornikom (angl. ADC – Analog-to-digital Converter), vgrajenim v mikrokrmilnik. Vrednost meritve je predstavljena z 10 bitmi, torej v

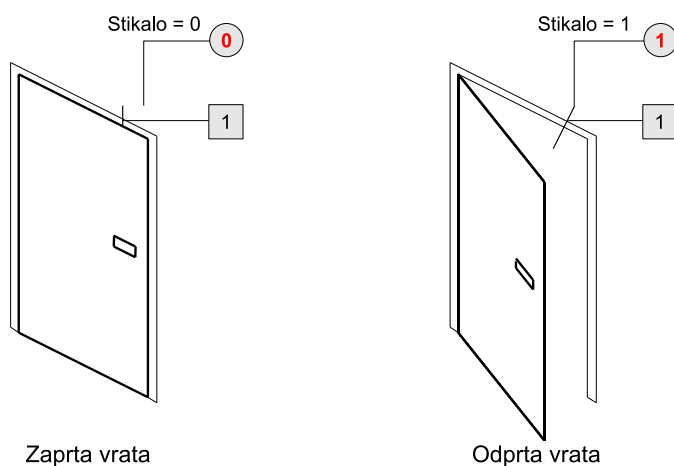


Slika 3.2: Leva in desna poravnanos 10-bitnega rezultata v dveh 8-bitnih registrih. Povzeto po [9].

1024 korakih. Sama pretvorba je relativno enostavna. Največ dela zahteva inicializacija modula ADC, ki jo je treba opraviti na začetku izvajanja pro-

grama. Tu je treba določiti analogno vhodno nožico, referenčno napetost, način pretvorbe (enkratni ali ponavljajoči se), določiti vir ure za pretvornik in na koncu omogočiti sam pretvornik. Po končani inicializaciji meritev izvajamo tako, da nastavimo bit za začetek pretvorbe, rezultat pa preberemo v registrih ADRESH in ADRESL. Poravnanosť rezultata, vidna na Sliki 3.2, je odvisna od nastavitve bita ADFM v registru ADCON2 in je lahko leva ali desna, pri čemer pa je najpomembnejši bit vedno na levi. Referenčno napetost predstavlja izhod iz regulatorja napetosti, in je 5 V. To tudi pomeni, da lahko pretvornik A/D (Analog-to-Digital) meri napetosti v območju od 0 V do 5 V. S takšnim območjem lahko zadovoljimo skoraj vse možne vhodne napetosti, razen tistih v intervalu od 5,0 V do 5,5 V, kolikor jih še dovoljuje napetostni regulator. V tem primeru bi bilo za točno meritev treba uporabiti napetostni delilnik.

Zadnji izmed podatkov, ki jih zajema oddajni modul, je stanje stikala. Z njim se določa, ali so vrata odprta ali zaprta in je torej razklenjeno ali sklenjeno. Ko je sklenjeno, je stanje vhodne nožice 1, v nasprotnem primeru pa 0. Shemo stanj stikala prikazuje Slika 3.3. Ker s stikalom nadziramo pomembne dogodke,



Slika 3.3: Stanje stikala pri odprtih in zaprtih vratih.

se mora sistem hitro odzivati na spremembe stanja. Zaradi tega je na vhodni nožici omogočena strojna prekinitve ob spremembi stanja. S tem dosežemo, da program ob vsaki sklenitvi ali razklenitvi stikala takoj skoči v prekinitveno funkcijo. V njej se nato ugotovi točna vrsta dogodka in ustrezno ukrepa. Tudi v primeru, ko je mikrokrmilnik v stanju spanja, ga ta vrsta prekinitve prebudi in preusmeri v izvrševanje omenjene funkcije.

Na trajanje zajema podatkov najbolj vplivata meritvi temperature in vlage.

Dolžina njune meritve je odvisna od izbrane ločljivosti. Pri vrednosti 14 bitov za temperaturo in 12 bitov za relativno vlago porabita v najslabšem primeru skupaj dobre 0,3 s časa. Meritev napetosti in preverjanje stanja stikala trajata zanemarljivo kratek čas. Prvi zahteva največ $240 \mu\text{s}$, drugi pa se opravi v času le enega ukaza, torej v $0,5 \mu\text{s}$.

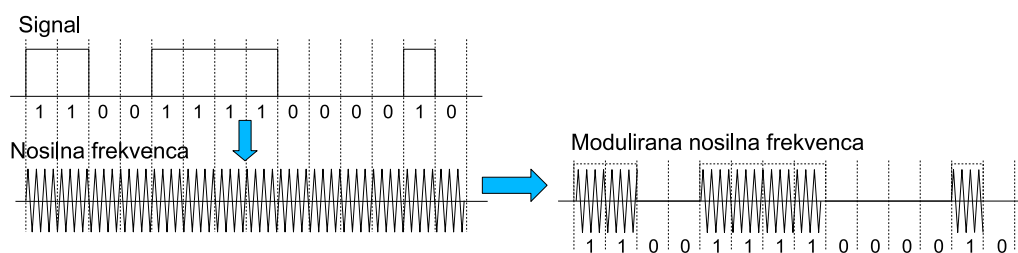
3.2 Brezžična povezava

Da bi lahko prek brezžične povezave zagotovili zanesljiv prenos podatkov, je treba upoštevati nekatere lastnosti oddajnika in sprejemnika ter vplive iz okolice. Pri uporabljenih modelih TLP434A in RLP434 je treba poskrbeti tako za kodiranje/dekodiranje podatkov kot tudi za implementacijo ustreznega protokola. Na kratke razdalje je enostavne prenose podatkov sicer možno opravljati tudi brez kodiranja in uporabe protokola, vendar je njihova zanesljivost zelo slaba.

3.2.1 Kodiranje in dekodiranje

Gre za postopek, ki na nizki ravni, tj. na ravni signala, pomaga povečati zanesljivost prenosa podatkov. Za uporabo tega postopka obstajajo naslednji razlogi.

Modula pri brezžičnem prenosu podatkov uporabljata digitalno amplitudno modulacijo (angl. ASK – Amplitude Shift Keying) z dvema stanjema. Pri njej se signal modulira tako, da je pri vrednosti 1 nosilna frekvenca prisotna, pri vrednosti 0 pa ne. Na Sliki 3.4 je prikazan postopek pretvorbe vhodnega signala z vrednostjo 11001111000010 v modulirano nosilno frekvenco. Postopek

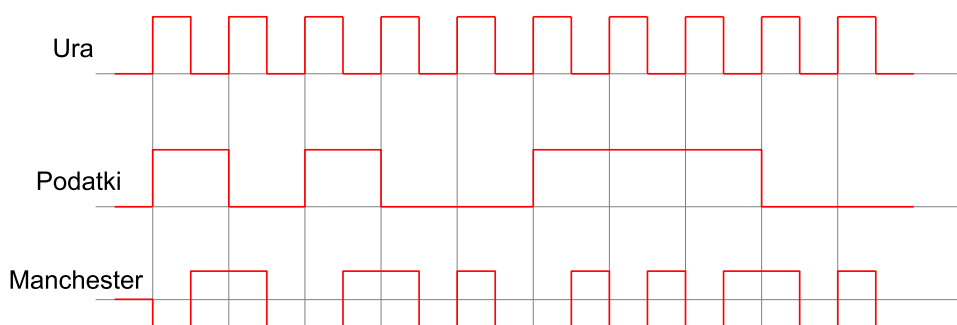


Slika 3.4: Prikaz amplitudne modulacije signala. Povzeto po [4].

pretvorbe je pri sprejemniku ravno nasproten. Če se zazna prisotnost nosilne frekvence, je vrednost sprejetega signala 1, če pa te frekvence ni zaznati, je vrednost signala 0.

Pomembna lastnost sprejemnika je samodejno ojačenje sprejema (angl. AGC – Automatic Gain Control). Jakost signala lahko namreč precej niha, na kar vplivata moč in oddaljenost oddajnika. Sprejemnik mora zato šibkejše signale bolj ojačiti, močnejše pa manj. S tem doseže optimalno jakost signala, ki je primerna za nadaljnjo obdelavo, tj. vzorčenje in kvantizacijo. Moč trenutnega ojačenja je odvisna od povprečne jakosti preteklih signalov. Na tem mestu pa lahko pride do težav, če je na sprejemniku dalj časa prisoten signal iste vrste oziroma jakosti. Za primer vzemimo signal, v katerem je *daljše* zaporedje ničel. Ta je po modulaciji predstavljen kot odsotnost nosilne frekvence, se pravi, da oddajnik takrat sploh ne oddaja. Sprejemnik bo v tem času sprejemal samo šum iz okolice, ki je načeloma precej nižje jakosti kakor signal našega oddajnika. Sprejemnik se temu prilagodi s povečano stopnjo ojačenja sprejema. Tu pa se pojavita dve težavi. Prva je ta, da so v šumu lahko prisotne tudi frekvence, ki ustrezajo naši nosilni frekvenci. Te bo sprejemnik interpretiral kot običajen signal in demodulirano vrednost poslal na izhod. Druga težava pa se pojavi v trenutku, ko po *daljšem* zaporedju ničel nastopi ena ali več enic. Takrat oddajnik začne oddajati, do sprejemnika pa v trenutku pride relativno močan signal. Sprejemnik ga zaradi trenutno visoke nastavitve ojačenja ne more pravilno obdelati (ne more določiti, ali je v njem prava nosilna frekvenca). Odzivnost, s katero zmanjša stopnjo ojačenja, pa je v takih primerih prepočasna, zato se prvi del *močnejšega* signala izgubi.

V izogib tem težavam je treba uporabiti takšno vrsto kodiranja, ki v kratkem časovnem intervalu čim bolj uravnovesi razmerje med ničlami in enicami. Samo na ta način lahko sprejemnik prilagodi ojačenje na najprimernejši nivo, ki ni niti previsok niti prenizek. Ena izmed enostavnejših in pogosto uporabljenih vrst je Manchester kodiranje [17] (Slika 3.5). Pri njem se ničla ali

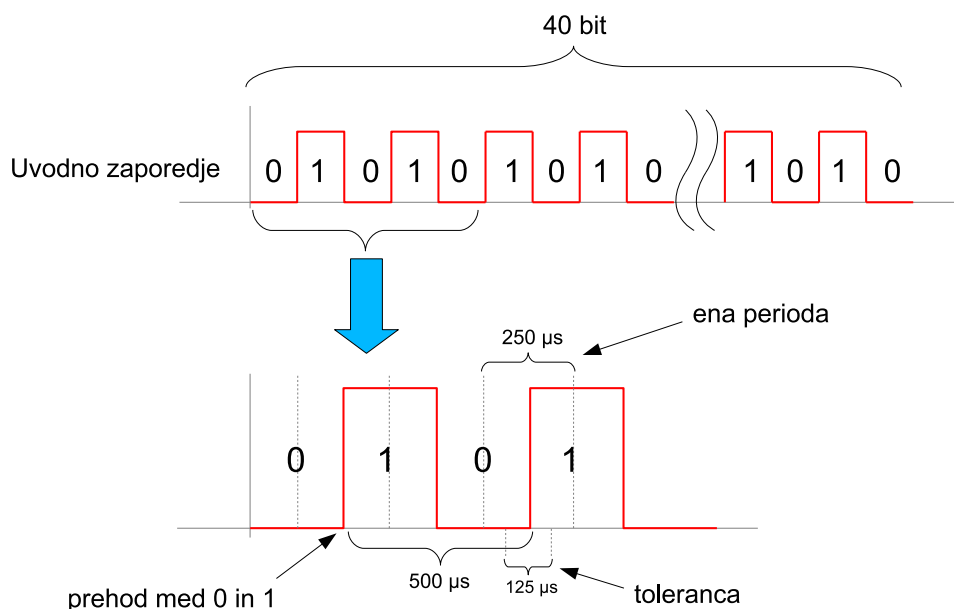


Slika 3.5: Delovanje Manchester kodiranja.

enica ponovi največ dvakrat, zato je signal zelo uravnotežen. Obstajata dva

dogovora, po katerih se kodira podatke, med njima pa je le trivialna razlika. Izvirni, po G. E. Thomasu, logično ničlo predstavi kot prehod signala iz nizkega v visoko stanje, logično enico pa s prehodom signala iz visokega v nizko stanje. V implementaciji pa je bil uporabljen drug dogovor, in sicer po standardu IEEE 802.3, ki zahteva ravno nasprotno prehode. Iz načina delovanja je mogoče ugotoviti, da sta za kodiranje vsakega podatkovnega bita potrebna 2 kodirna bita. V praksi to pomeni, da se pasovna širina, ki je na voljo za prenos podatkov, prepolovi. Sprejemnik RLP434 lahko po specifikacijah deluje z najvišjo hitrostjo 4800 b/s, iz tega sledi, da se najvišja hitrost z uporabo Manchester kodiranja zmanjša na 2400 b/s. Takšno zmanjšanje pasovne širine v našem primeru ne predstavlja ovire, saj oddajamo samo na vsakih trideset sekund ali redkeje, podatkovni paketi pa so veliki le 170 bitov.

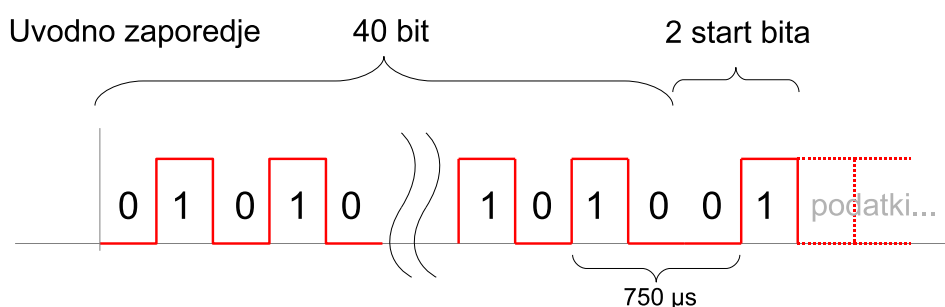
Kodiranje in dekodiranje sta implementirana programsko. Pri tem je zelo pomembna natančna sinhronizacija med oddajnikom in sprejemnikom. Na oddajni strani so časovne periode signala generirane z običajnimi programskimi zakasnitvami (angl. delay). Njihove vrednosti so bolj natančno nastavljene eksperimentalno, in sicer z uporabo simulatorja MPLAB SIM. Simulator nam pri tem pomaga tako, da izračuna točen čas izvajanja določenega dela programa, na primer med delom, kjer se signal postavi v visoko stanje, in delom, kjer se vrne v nizko.



Slika 3.6: Detekcija prehodov v uvodnem zaporedju enic in ničel.

Nekaj več dela je pri sprejemniku, kjer je treba prejeti signal programsko dekodirati. Začne se z detekcijo uvodnega zaporedja enic in ničel, ki je na čelu vsakega poslanega paketa (Slika 3.6). To zaporedje, dolžine 40 bitov, je sestavljeno iz alternirajočih vrednosti 0 in 1, ki na ta način ustvarijo značilen vzorec za lažjo prepoznavo začetka paketa. Obenem tak uravnotežen vzorec služi tudi za prilagoditev ojačitvene stopnje sprejemnika. Zaznavanje prehodov iz nizkega v visoko stanje in nasprotno se dela z uporabo strojne prekinitve, ki ob vsakem takem prehodu preusmeri izvajanje programa v prekinitveno funkcijo. V njej se z uporabo časovnika preverja, ali so se na sprejemniku pojavili ustrezni prehodi stanj in to ob zahtevanih časovnih intervalih. Konkretno se preverja, ali se na *približno* vsakih $500 \mu\text{s}$ pojavi prehod iz nizkega v visoko stanje. Interval $500 \mu\text{s}$ sledi iz izbrane hitrosti prenosa, ki znaša 4000 b/s . Pri tej hitrosti je namreč dolžina ene periode $250 \mu\text{s}$, rob, ki predstavlja začetek enice, pa se pojavi na vsakih $500 \mu\text{s}$. Interval, oziroma toleranca, v kateri se mora pojaviti tak rob, da je še vedno veljaven, znaša $125 \mu\text{s}$.

Uvodnemu zaporedju sledita tako imenovana začetna ali start bita (Slika 3.7). Gre za zaporedje z vrednostjo 01, s katerim je mogoče identificirati



Slika 3.7: Prikaz dveh start bitov, ki sledita uvodnemu zaporedju.

začetek pravih podatkovnih bitov. Njegova značilnost je ta, da koncu uvodnega zaporedja, ki se konča z ničlo, doda še eno ničlo, nato pa enico. V prekinitveni funkciji se to ugotovi tako, da se prehod iz 0 v 1 zazna šele po $750 \mu\text{s}$ in ne $500 \mu\text{s}$, kot je to običajno. Da se uvod v podatkovni paket lahko razglasi za veljavnega, je treba sprejeti vsaj enajst od skupno dvajsetih ustreznih prehodov. Takšen pogoj je postavljen povsem eksperimentalno. Na eni strani predstavlja dovolj visok prag, da šuma ni mogoče prepogosto zamenjati za signal, na drugi strani pa dovolj nizek prag, da lahko sprejemnik prvih nekaj prehodov izpusti, ker nivo ojačitve takrat še ni prilagojen. Pri vsakem prehodu, ki se pojavi ob nedovoljenem časovnem intervalu, se števec

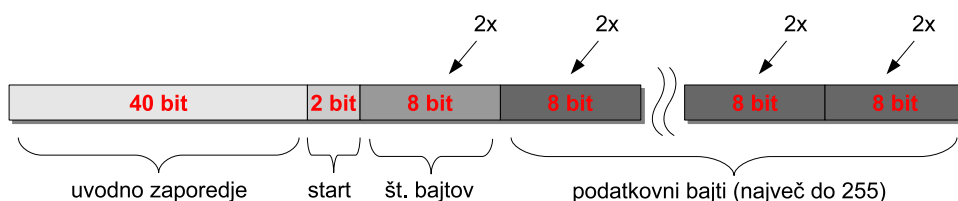
pravilno prejetih prehodov ponastavi in začne šteti znova. Na ta način se že na začetku izognemo nepotrebnim obdelavi celotnega paketa, katerega vir je po vsej verjetnosti šum.

Na koncu tega postopka je naloga prekinitvene funkcije opravljena. Strojne prekinitve se izklopijo, program pa nadaljuje izvajanje v funkciji za sprejem podatkov. Podatki so po protokolu sestavljeni iz več bajtov — več o tem v naslednjem podpoglavju. Sprejemna funkcija deluje tako, da na začetku vsakega bajta, v zanki *while*, počaka na sprejem njegovega prvega bita (prehod iz 0 v 1 ali nasprotno), od tam naprej pa s programskimi časovnimi zakasnitvami vzorči ostalih sedem bitov. S tem postopkom se doseže dovolj natančna sinhronizacija na nivoju bajta, ki v praksi zadostuje za zanesljiv sprejem.

3.2.2 Protokol

Kodiranje poskrbi le za zanesljivejši prenos podatkov na nivoju signala, ne more pa preprečiti ali odkriti vseh napak. Težavne so predvsem tiste, zaradi katerih je sprejeti signal popačen do te mere, da vsebuje napačne vrednosti podatkov, časovne periode signala pa kljub temu ustrezajo zahtevam običajnega signala. Največkrat jih povzroči premočan šum iz okolice ali prešibek signal oddajnika. Rešitev tega problema je v uporabi protokola in v njem ustreznega sistema za odkrivanje napak.

Protokol sam po sebi ni namenjen le odkrivanju in potencialnemu odpravljanju napak. Namenjen je predvsem strukturiranju podatkov v celoto, ki se najpogosteje imenuje (podatkovni) paket. V tem sistemu je uporabljen lastno zasnovan protokol, predstavljen na Sliki 3.8. Njegova zgradba je zelo eno-



Slika 3.8: Zgradba protokola za brezžičen prenos podatkov.

stavna. Znano je že, da se začne z uvodnim zaporedjem in start bitoma. S tem si sprejemnik pomaga pri nastavljanju ojačitve in ugotavljanju dejanskega začetka podatkovnih bitov. Sledijo zaporedja bajtov z naslednjim pomenom. Prvi pove, koliko podatkovnih bajtov vsebuje paket. Ker gre za 8-bitno spremenljivko, je največje možno število 255. Paket se nato zaključi s točno toliko

podatkovnimi bajti, kolikor jih je predvideno. Zgradba dejanskega podatkovnega paketa je prikazana v Tabeli 3.1.

Tabela 3.1: Zgradba podatkovnega paketa.

uvodno zaporedje	40 bitov
start bita	2 bita
2x št. podatkovnih bajtov	16 bitov
2x temperatura MSB	16 bitov
2x temperatura LSB	16 bitov
2x relativna vlaga MSB	16 bitov
2x relativna vlaga LSB	16 bitov
2x napetost baterije MSB	16 bitov
2x napetost baterije LSB	16 bitov
2x stanje vrat	16 bitov
skupaj	170 bitov

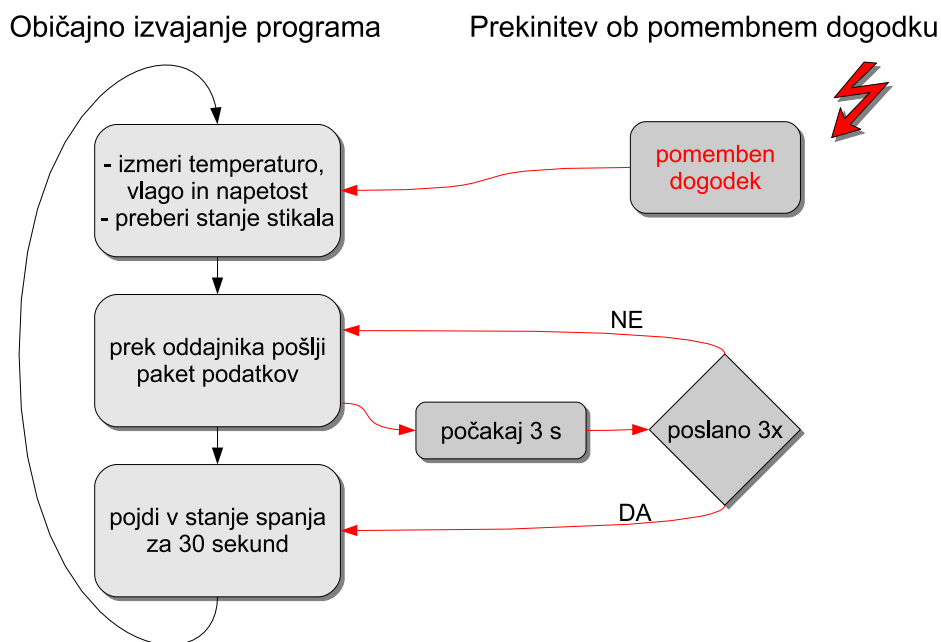
Podatkovni bajti se zaradi uporabljenega načina odkrivanja napak pošiljajo dvakrat. S kratico MSB (Most Significant Byte) so označeni najpomembnejši bajti, z LSB (Least Significant Byte) pa najmanj pomembni. Celoten paket vsebuje 170 bitov, zato prenos pri hitrosti 4000 b/s in ob upoštevanju Manchester kodiranja, traja 42,5 ms. Čas trajanja prenosa ustreza tudi zakonskim predpisom [15], ki v frekvenčnem spektru 433,05 Mhz—434,79 MHz zahtevajo obratovalni cikel do 10 %. Gre za časovno razmerje v vsakem enournem obdobju aktivnega oddajanja.

Na tem mestu lahko izračunamo tudi delež časa, v katerem je aktiven celoten oddajni modul. Ta znaša približno 0,35 s in se ponavlja v intervalu po 30 s. Na vsakih trideset sekund se torej mikrokrmilnik zbudi iz spanja ter v manj kot pol sekunde opravi potrebne meritve in pošiljanje podatkov prek oddajnika.

Odkrivanje napak pri prenosu podatkov je možno na več načinov. Pogosto uporabljen je tako imenovan algoritem *checksum* [16]. Deluje tako, da nad vhodnimi podatki izvede določeno operacijo (npr. XOR ali modulo) in rezultat te operacije pošlje skupaj s podatki. Isto operacijo izvede tudi sprejemnik, nato pa svoj rezultat primerja s prejetim. Uspešnost detekcije napak je odvisna tako od vrste operacije nad podatki kot tudi od dolžine rezultata. Vrsta operacije posledično vpliva tudi na procesorski čas, ki se porabi za njeno izvajanje. Dolžina rezultata pa na količino podatkov, ki jih treba prenesti. V našem

primeru ni strogih zahtev glede velikosti paketa, je pa zaželeno, da operacija ni zahtevna. Zaradi tega je bil zasnovan eden izmed najenostavnejših načinov odkrivanja napak. Vsak bajt, tako tisti, ki določa število podatkovnih bajtov, kot tudi dejanski podatkovni bajti, se pošlje dvakrat zapored. Algoritem ima le to nalogo, da vsak prvi sprejeti bajt primerja z drugim in ugotovi, ali sta enaka. Če so vsi pari bajtov enaki, je paket veljaven, če kateri izmed njih ni enak, se celoten paket zavrže. S tem postopkom je mogoče odkrivanje napak, ne pa tudi odpravljanje. Očitno je tudi, da se količina prenesenih podatkov z njim dvakrat poveča, operacija primerjanja bajtov pa je trivialna.

Paketi se lahko pošiljajo v poljubnih časovnih intervalih. Smiselno pa je, da niso pregosti, saj se s tem poveča poraba energije oddajnega modula. Primeren interval je med 30 sekundami in 1 minuto. Kljub temu pa so nekateri podatki ali vsaj dogodki pomembni in bi želeli biti o njih obveščeni s krajšo odzivnostjo oziroma takoj. Tako so na primer obravnavani dogodki ob spremembah stanja



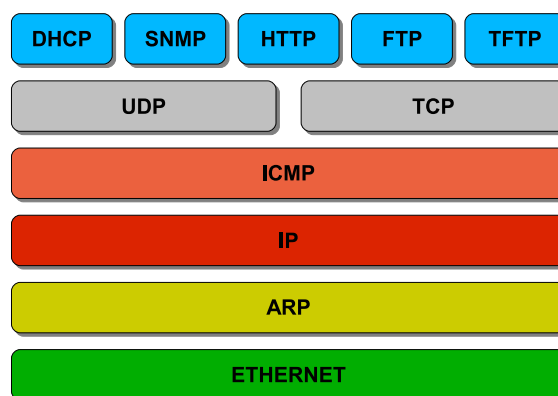
Slika 3.9: Diagram običajnega poteka programa ter ob pomembnem dogodku.

stikala. Ob vsaki spremembi se zato sproži strojna prekinitev, nato pa se nemudoma, v presledkih po tri sekunde, pošlje tri pakete z najaktualnejšo vsebino podatkov. Isto vsebino paketa se večkrat pošlje zato, da se s tem poveča zanesljivost njegovega prenosa. Diagram poteka programa ob takšnem

dogodku je prikazan na Sliki 3.9. S črnimi puščicami je označen običajen potek programa, ko oddajnik obratuje v 30-sekundnih intervalih. Z rdečimi puščicami pa je označen potek programa ob pomembnem dogodku. Takrat se v trenutku izvedejo vse meritve in preverjanje sklenjenosti oz. razklenjenosti stikala, nato pa se trikrat zaporedoma, s premori po tri sekunde, pošlje paket z najnovejšimi podatki.

3.3 Spletni strežnik

Najobsežnejši izmed programov je spletni strežnik, ki teče v mrežnem modulu. Za njegovo delovanje je treba implementirati preostali del sklada TCP/IP, ki še ni vsebovan v mrežnem krmilniku, ter aplikacijo za strežnik HTTP (Hypertext Transfer Protocol). Večino tega dela si lahko prihranimo z uporabo ene izmed obstoječih implementacije sklada. V našem primeru gre za Microchip TCP/IP Stack v4.55, katerega zgradba je prikazana na Sliki 3.10.



Slika 3.10: Plasti sklada TCP/IP v Microchipovi implementaciji v4.55. Vir: [8].

Prednost uporabe tega programskega paketa je tudi v tem, da že vsebuje številne module, kot je spletni strežnik, odjemalec za elektronsko pošto, strežnik telnet, odjemalec TFTP (Trivial File Transfer Protocol), odjemalec in strežnik DHCP (Dynamic Host Configuration Protocol) itd. Tako se nam sploh ni treba ukvarjati s podrobnostmi razvoja in implementacije spletnega strežnika in se lahko osredotočimo samo na razvoj dejanske aplikacije. Ta mora zagotavljati sprejem, obdelavo in prikaz podatkov ter vodenje dnevnika pomembnih dogodkov.

V sklopu tega sklada je že vsebovana naslednja zgradba programa — po začetni inicializaciji sledi glavna zanka, v kateri se krožno obdelujejo posamezne naloge strežnika:

```
while(1)
{
    StackTask();           // obdelaj naloge sklada (delo s paketi)

    StackApplications(); // obdelaj naloge strežnika

    ProcessIO();         // preglej stanje vhodno-izhodnih naprav

    AnnounceIP();       // na zaslonu prikaži trenutni naslov IP
}
```

Najprej se izvršijo naloge sklada, kjer se preverijo dohodni paketi in kličejo ustrezne komponente za njihovo obdelavo. Naslednje so na vrsti aplikacije, med katere spadata strežnik HTTP in odjemalec SNTP (Simple Network Time Protocol) za sinhronizacijo časa. Prvi se mora odzivati na povezave, ki vsebujejo zahteve, drugi pa osveževati datum in uro v 10-minutnih intervalih. Med preostale naloge spadata še pregledovanje vhodno-izhodnih naprav, kot na primer tipke, ter prikazovanje naslova IP (Internet Protocol), ko se ta spremeni. Ti deli programa so že implementirani, deli, ki jih je bilo treba dodati, pa so opisani v nadaljevanju.

3.3.1 Prikaz dinamičnih podatkov

Mikrokrmilniški sistemi so navadno namenjeni nadzoru podatkov, katerih vrednost se s časom spreminja. In če jih želimo prikazovati na spletni strani, mora biti ta dinamična. Pri straneh, ki so v osnovi napisane v HTML-ju (Hypertext Markup Language), lahko spremenljive podatke predstavimo z dinamičnimi spremenljivkami, ki jih na obeh straneh označimo s tilde, na primer `~spremenljivka1~`. Ko program spletnega strežnika naleti na takšno spremenljivko, pokliče tako imenovano *callback* funkcijo. To funkcijo mora v kodo dodati razvijalec sam, v imenu pa mora vsebovati ime dinamične spremenljivke. V našem primeru je ime take funkcije `HTTPPrint_spremenljivka1`. V njej se nato postori vse, kar je potrebno za obdelavo in prikaz vsebine spremenljivke. Preprost primer kode HTML, ki na spletni strani prikaže točen čas:

```
<html>
```

```
Točen čas je: ~ura~
```

</html>

Spremenljivka je tu označena s tilde zato, da lahko njeno vsebino dinamično spreminjamo.

Koda v jeziku C, ki na mestu dinamične spremenljivke `~ura` izpiše točen čas:

```
void HTTPPrint_ura(void)
{
    BYTE string[11]; // rezerviraj prostor za niz dolžine 11 znakov
    ultoa(GetCurrentTime(), string); // čas (število) pretvori v niz

    TCPPutString(sktHTTP, string); // izpiši niz
}
```

Najprej opazimo, da je v imenu funkcije vsebovano ime dinamične spremenljivke. V funkciji pa je tik pred izpisom treba vse podatke pretvoriti v znakovni niz. Sam izpis se izvrši s klicem funkcije `TCPPutString`, ki kot parameter sprejme trenutni *socket*, označen z globalno spremenljivko `sktHTTP`, in naslov niza. Večina izpisov, katerih dolžina znaša največ 16 bajtov, poteka po opisanem postopku. Za daljše nize pa je treba opraviti več korakov. Pri tem si pomagamo s klicem funkcije `TCPIsPutReady(sktHTTP)`. Ta vrne število bajtov, ki jih lahko v trenutnem koraku izpišemo. Število že poslanih bajtov pa lahko shranjujemo v spremenljivko `curHTTP.callbackPos`, katere vrednost je dosegljiva v vseh nadaljnjih korakih.

3.3.2 Zgradba spletne strani

V splošnih zahtevah je zapisano, da naj bi prek spletnega vmesnika bilo mogoče spremljanje vseh izmerjenih podatkov in zgodovine dogodkov. Ker se te vsebine v rednih intervalih spreminjajo, bi bilo praktično, da se tudi samodejno osvežujejo in uporabniku prikazujejo v realnem času. Takšno funkcionalnost omogoča tehnologija AJAX, ki združuje skriptni jezik JavaScript in opisni jezik XML (Extensible Markup Language). Zahtevki lahko v ozadju samodejno pridobivajo aktualne podatke s strežnika in jih prikazujejo, ne da bi bilo pri tem treba prenašati celotno stran.

Osnova strani je zgrajena iz HTML-ja, oblika je določena s predlogo CSS (Cascading Style Sheets), vsa logika za prenos, obdelavo in prikaz podatkov pa je napisana v JavaScriptu. V nadaljevanju je prikazan poenostavljen primer, ki samodejno osvežuje prikaz temperature.

Spremenljivke, ki jih osvežuje JavaScript, so najprej zapisane v datoteki XML:

```
<spremenljivke>
<temp>temp</temp>
</spremenljivke>
```

Opazimo, da je spremenljivka označena s tilde ter tako predstavlja dinamično spremenljivko, ki kliče svojo funkcijo *callback*.

Mesto v kodi HTML, kjer želimo prikazati temperaturo, lahko označimo s poljubno spremenljivko. Pomembno je le, da jo z istim imenom naslovimo v vključeni kodi JavaScript. V njej se namreč prebere vrednost spremenljivke iz datoteke XML ter se jo prikaže na spletni strani:

```
<html>
Trenutna temperatura: <span id="temp"?</span>
// del kode JavaScript je vključen kar v datoteko HTML
<script type="text/javascript">
function updateStatus(xmlData)
{
  // preberi vrednost spremenljivke 'temp'
  var temperature = getXMLValue(xmlData, 'temp');
  // njeno vrednost zapiši v HTML
  document.getElementById('temp').innerHTML = temperature;
}
</script>
</html>
```

Izbran interval osveževanja podatkov je nastavljen na 2 sekundi, kar predstavlja dovolj kratek čas, v katerem se lahko hitro opozori tudi na pomembne dogodke.

3.4 Druge naloge

Poleg večjih, zgoraj opisanih nalog, mrežni modul opravlja še številne druge, ki so predstavljene v tem podpoglavju. Gre za pretvarjanje podatkov, vzdrževanje dnevnika dogodkov, sinhronizacijo točnega časa ter prikaz informacij na zaslonu LCD.

3.4.1 Sprejem in pretvorba podatkov

Temperatura, relativna vlaga, napetost in stanje stikala so podatki, ki jih sprejemni modul dobi v osnovni, neobdelani obliki. Nato jih pošlje mrežnemu modulu, ta pa jih mora pred prikazom še primerno obdelati.

Povezava med sprejemnim in mrežnim modulom ima naslednje lastnosti. Med seboj sta povezana s serijskim vmesnikom RS-232, in sicer le z enim signalom, prek katerega sprejemni modul enosmerno pošilja podatke mrežnemu modulu. Oba mikrokrmilnika vsebujeta strojni modul s podporo vmesniku RS-232, zato je uporaba preprosta. Po začetni inicializaciji, v kateri se določi hitrost prenosa, vrsta (asinhroni/sinhroni) in še nekaj drugih parametrov, se lahko začne prenašanje podatkov. Posamezen bajt se pošlje tako, da se ga zapiše v register TXREG. Mikrokrmilnik mrežnega modula ima omogočeno strojno prekinitev ob sprejemu bajta. Ko do nje pride, se vrednost bajta prebere iz sprejemnega registra RCREG.

Ker gre za žično povezavo, je zanesljivost prenosa visoka, in ni potrebe po kompleksnem protokolu. Zaradi tega je uporabljen primitiven postopek, ki na začetek in konec celotnega paketa podatkov doda bajt z vrednostjo 255. Dolžina celotnega paketa pa je omejena na 9 bajtov. Torej, ko do sprejemnika prispe devet bajtov, se preveri le še to, ali imata prvi in zadnji vrednost 255. Če pogoji ustrezajo vsem zahtevam, je paket veljaven, sicer ni.

Prispeli podatki so še v neobdelani obliki. Zato jih je treba pretvoriti v format, prijazen do uporabnika. Pretvorbe na prvi pogled sicer niso videti težavne, a se izkaže, da je za izračun temperature in vlage treba uporabiti števila s plavajočo vejico ter operacijo množenja. To pa 8-bitnemu krmilniku, z omejenim pomnilnikom in relativno nizko procesorsko zmogljivostjo, predstavlja težave. Velika večina pomnilnika se namreč porabi za delovanje strežnika in nalog, povezanih z njim. Zato ostane premalo prostega prostora, ki bi ga prevajalnik želel rezervirati ob uporabi operacij s plavajočo vejico. Zaradi tega se nekatere pretvorbe v mrežnem modulu, ki so namenjene prikazu na zaslonu LCD, izvedejo po enostavnejšem postopku. Podatki, ki se morajo prikazati na spletni strani, pa se lahko bolj natančno izračunajo kar na strani odjemalca, in sicer v spletnem brskalniku, z uporabo jezika JavaScript.

Pretvorba temperature po formuli 3.1 [2] poteka na strežniku, rezultat pa se uporabi za prikaz na zaslonu in spletni strani.

$$\text{Temperatura} = d_1 + d_2 \cdot SO_T \quad (3.1)$$

d_1 in d_2 sta konstanti, SO_T pa je sestavljena združena vrednost obeh bajtov, ki ju vrne senzor. Prva konstanta je odvisna od napajalne napetosti in znaša

-40,1 pri 5 V. Druga konstanta je odvisna od ločljivosti meritve in ima pri 14-bitni ločljivosti vrednost 0,01.

Relativno vlažnost lahko izračunamo na dva načina. Formula 3.2 je manj natančna in enostavnejša, zato se uporablja za prikaz na zaslonu mrežnega modula.

$$RH_{linearna} = c_1 + c_2 \cdot SO_{RH} + c_3 \cdot SO_{RH}^2 \quad (3.2)$$

Natančnejši način (3.3) upošteva temperaturno kompenzacijo, zato je bolj zapleten in se računa le na strani odjemalca.

$$RH_{prava} = (T_{\circ C} - 25) \cdot (t_1 + t_2 \cdot SO_{RH}) + RH_{linearna} \quad (3.3)$$

Podobno kot prej so c_1 , c_2 in c_3 konstante in imajo v našem primeru vrednosti -4, 0,0405 in $-2,8 \cdot 10^{-6}$, konstanti t_1 in t_2 pa vrednosti 0,01 in 0,00008. SO_{RH} je surova vrednost, ki jo pošlje senzor, $T_{\circ C}$ je izračunana temperatura v stopinjah Celzija, $RH_{linearna}$ pa izračunana relativna vlažnost po prvem postopku.

S poznavanjem vrednosti temperature in relativne zračne vlage se lahko izračuna še rosišče. Gre za mersko enoto, ki nam pove, pri kateri temperaturi se začne iz zraka izločati voda. Njen izračun (3.4) je še bolj zahteven od tistega za zračno vlago, zato se uporablja le za prikaz na spletni strani.

$$\text{Rosišče} = T_n \cdot \frac{\ln\left(\frac{RH}{100\%}\right) + \frac{m \cdot T}{T_n + T}}{m - \ln\left(\frac{RH}{100\%}\right) - \frac{m \cdot T}{T_n + T}} \quad (3.4)$$

Vrednost konstant T_n in m je odvisna od temperature. V območju nad lediščem se uporabljata vrednosti 243,12 in 17,62, pod lediščem pa 272,62 in 22,46.

Pri merjenju napetosti baterije ADC modul vrača vrednosti v intervalu od 0 do 1023. V uveljavljen prikaz napetosti jih pretvorimo po enačbi 3.5.

$$\text{Napetost} = ADC_{vrednost} \cdot \frac{5 \text{ V}}{1023} \quad (3.5)$$

V enačbi je privzeto, da je referenčna napetost 5 V, prav toliko pa znaša tudi največja napetost, ki jo želimo izmeriti.

Stanje stikala ponazarjata vrednosti 0 in 1, zato je pri izpisu uporabljen stavek *if*. Primer v psevdokodi:

```
if(stikalo == 0)
    print("Vrata_zaprta.");
else if(stikalo == 1)
    print("Vrata_ODPRTA!");
```

3.4.2 Dnevnik dogodkov

Ideja beleženja zgodovine dogodkov ja ta, da se ob neki spremembi stanja shrani datum in uro ter omogoči poznejši vpogled v take zapise. V našem primeru predstavlja pomemben dogodek odpiranje vrat, saj to potencialno pomeni vlom v neki prostor. Uporabnik lahko ob pregledu zapisov dogodkov ugotovi dejanski vzrok za tak dogodek (bodisi jih je odprl sam ali pa druga oseba).

Vsa logika, ki skrbi za vzdrževanje dnevnika, se nahaja v programu mrežnega modula. Prvi pogoj, ki mora biti izpolnjen, da se ustvari zapis o dogodku, je sprememba stanja vrat iz »zaprt« v »odprta«. Takrat se v tabelo dogodkov shrani trenutni čas, nato pa sledi prepisovanje vsebine tabele v pomnilnik EEPROM (Electrically Erasable Programmable Read-Only Memory). Slednje je nujno, da se dnevnik ohrani tudi ob izklopu napajanja. Pisanje v EEPROM poteka po en bajt naenkrat in zahteva skupno šest korakov. V prvem se na naslovu 0x00 (šestnajstiško) shrani število trenutno zapolnjenih mest v pomnilniku. Nato sledi prepisovanje ure in datuma, ki sta v formatu Unix. To je zapis, ki predstavi trenutni čas kot število sekund, ki so pretekle od 1. januarja 1970, in je dolžine 32 bitov. Zaradi tega se za shranjevanje časa porabi nadaljnje štiri korake. Nazadnje se še na naslovu 0x01 (šestnajstiško) poveča vrednost kazalca na pomnilnik, ki določa mesto, kamor se bo shranil naslednji zapis. Pomnilnik je krožne oblike in je programsko omejen na kapaciteto največ 20 zapisov. Najnovejši zapisi zato prepišejo najstarejše. Uporabnik ima na mrežnem modulu na voljo tudi tipko, s katero lahko izbriše celotno vsebino dnevnika.

Pregledovanje vsebine dnevnika je zaradi obsežnosti na voljo le prek spletne strani. Tam je viden seznam datumov in točnih ur, ob katerih je prišlo do pomembnega dogodka. S pomočjo JavaScripta je oblikovan padajoč prikaz po času, prav tako pa tudi pretvorba iz formata Unix v format »DD. mesec LLLL, HH:MM:SS« po lokalnem času odjemalca. Uporaba vgrajenih funkcij za pretvorbo formata datuma in ure je videti tako:

```
// ustvarimo spremenljivko tipa Date
var datum = new Date(unixTime);
// v HTML spremenljivko cas izpišemo datum in uro po lokalnem času
document.getElementById('cas').innerHTML =
    datum.toLocaleDateString() + ", " + datum.toLocaleTimeString();
```

Točen čas predstavlja pomemben del pri shranjevanju dogodkov. Vgrajen oscilator mikrokrmilnika ni dovolj natančen, da bi zadovoljil to potrebo. Druga rešitev je uporaba posebnega čipa, ki vzdržuje natančen čas in datum. Vendar

tudi tu ostane slabost, da je čas treba ročno nastavljanje, v najboljšem primeru dvakrat na leto, ob spremembi poletnega in zimskega časa. Poleg tega obe rešitvi zahtevata neprestano napajanje, tudi ob izklopu iz električnega omrežja, kar še poveča kompleksnost in stroške izdelave. Najboljša rešitev v tem primeru je zato sinhronizacija ure prek časovnega strežnika NTP (Network Time Protocol). Modul je tako ali tako ves čas povezan z internetom, potrebe po neprestanem napajanju pa ni več. Programski modul, ki se zna povezati na tak strežnik in redno vzdrževati točen čas, je že vgrajen v Microchipov sklad. Imenuje se odjemalec SNTP, ki je poenostavljena različica odjemalca NTP in je namenjen prav vgrajenim sistemom. S časovnim strežnikom se poveže na vsakih 10 minut in pridobi točno vrednost ure. Med posameznimi sinhronizacijami uro vzdržuje program v mikrokrmilniku. Ta jo vsako sekundo osveži in jo kljub ne zelo natančnemu oscilatorju ohranja dovolj točno za naše potrebe.

Poglavje 4

Primeri uporabe

V tem poglavju sledi prikaz uporabe sistema. Predstavljene so vse tri enote, s katerimi ima uporabnik stik. Najprej je opisan oddajni modul, nato mrežni modul z zaslonom, na koncu pa še vmesnik spletne strani.

4.1 Oddajni modul

Za električno napajanje je priporočljiva uporaba dveh akumulatorskih baterij tipa AA, z nazivno napetostjo 1,2 V. Ob priklopu na napajanje začne modul takoj delovati.

Pri merjenju temperature in vlage je pomembna postavitev senzorja. Za čim bolj točne meritve mora biti senzor nameščen v senci ter ne neposredno pri tleh ali v bližini sten. Prav tako ne sme biti zaprt v neprodušnem ohišju, saj je s tem onemogočeno merjenje realne relativne vlage ozračja.



Slika 4.1: Fotografija mikrostikala s tremi priključki.

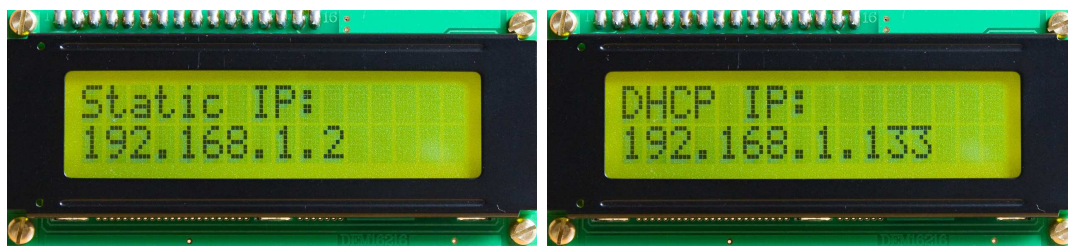
Na koncu ostane še priklop stikala. V primeru, ko želimo zaznavati odpiranje in zapiranje vrat, je najprimernejše mikrostikalo (Slika 4.1). Navadno ima

3 priključke. Prvi par priključkov prevaja električni tok takrat, ko je stikalo razklenjeno, in ga ne prevaja, ko je stikalo sklenjeno. Drugi par deluje ravno nasprotno. Predvideva se, da je stikalo nameščeno tako, da je pri zaprtih vratih sklenjeno, pri odprtih pa razklenjeno. Zaradi tega je za pravilno določanje stanja treba na modul priključiti prvi par priključkov.

4.2 Mrežni modul

Vir napajanja je omrežni napajalnik, ki mora zagotoviti usmerjeno napetost v območju od 7,5 do 35 V. Pri tem velja, da se z višjo vhodno napetostjo manjša učinkovitost napetostnega regulatorja. Vsa napetostna razlika, pomnožena z izhodnim tokom, se namreč sprošča kot odvečna toplota.

Modul se lahko v omrežje povezuje na dva načina, bodisi s statičnim naslovom IP ali pa z dodeljenim dinamičnim. V prvem primeru se na zaslonu prikaže vnaprej določen naslov, prikazan na Sliki 4.2 (levo). Če pa se mrežni



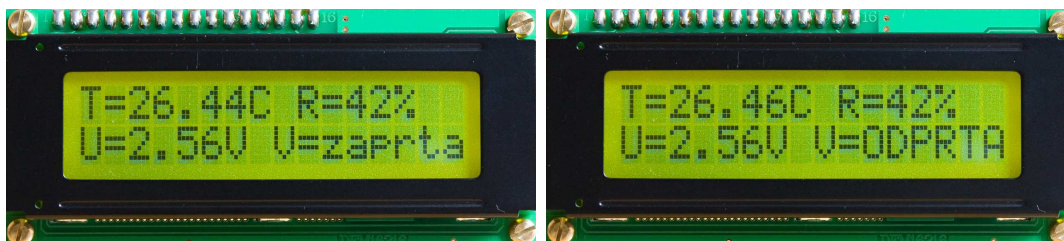
Slika 4.2: Prikaz statičnega naslova IP (levo) in dinamičnega naslova IP (desno).

modul poveže z usmerjevalnikom (angl. router), mu ta po protokolu DHCP dodeli lasten naslov IP, ki je viden na Sliki 4.2 (desno). Podatki o naslovu se prikažejo vsakič, ko se ta spremeni in so prikazani toliko časa, dokler jih ne prepisejo novi podatki z oddajnega modula.

Aktualne informacije so na zaslonu predstavljene z naslednjimi simboli po vrstnem redu:

- T — temperatura zraka v stopinjah Celzija
- R — relativna vlažnost zraka v odstotkih
- U — napetost napajalnih baterij v voltih
- V — stanje vrat z besedo zaprta/ODPRTA

Na Sliki 4.3 sta fotografiji zaslona z zgoraj opisanimi podatki. Na levi je primer z zaprtimi, na desni pa z odprtimi vrati.



Slika 4.3: Prikaz temperature, relativne vlage, napetosti in stanja vrat: zaprta (levo), odprta (desno).

4.3 Spletna stran

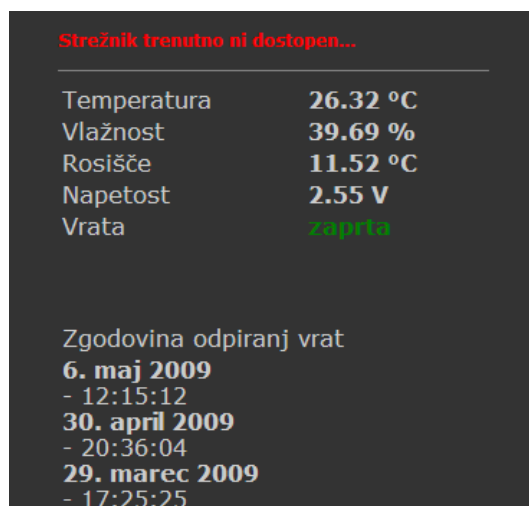
Do spletne strani je mogoče dostopati na naslovu IP, ki ga strežnik uporablja v priključenem omrežju. Treba pa je paziti na razliko med lokalnim in javnim naslovom. Lokalni je tisti, ki mu ga dodeli usmerjevalnik oziroma tisti, ki je že predhodno določen, tj. statičen. Na zaslonu mrežnega modula se prikaže le ta vrsta naslovov. Javni pa je običajno naslov usmerjevalnika ali prehoda (angl. gateway). Razlika med njima je v tem, da imajo do prvega dostop samo odjemalci, ki so priklopljeni v isto omrežje, do drugega pa vsi, tudi odjemalci prek interneta. Če želimo spletno stran gostiti na zunanjem naslovu, je treba na usmerjevalniku nastaviti ustrezno posredovanje vrat (angl. port forwarding). Uporabljajo se vrata 80, ki so običajna za strežnike HTTP.

Spletna stran je prikazana na Sliki 4.4. Na levi je izpis stanja, ko so vrata zaprta, na desni pa je vidno stanje odprtih vrat in nov vnos v dnevnik. Opaziti je mogoče, da je nov zapis v dnevnik izrinil najstarejšega. Zaradi preglednosti je zgodovina dogodkov omejena na tri zapise. Primer strani s celotnim dnevnikom je na Sliki C.1.

Na strani odjemalca se izvaja tudi del kode, ki preverja dostopnost strežnika. Če v roku desetih sekund strežnik ne odgovori na zahtevek, se na vrhu strani izpiše sporočilo o nedostopnosti. Primer je prikazan na Sliki 4.5. Takoj, ko je strežnik ponovno dostopen, sporočilo izgine, nadomestita pa ga trenutni datum in ura.



Slika 4.4: Zaslonski sliki spletne strani. Vrata zaprta (levo), vrata odprta in nov vnos v dnevnik (desno).



Slika 4.5: Zaslonska slika spletne strani, ko strežnik ni dostopen.

Poglavje 5

Sklepne ugotovitve

Končni izdelek ustreza vsem uvodoma podanim zahtevam. Torej, oddajnik zajema podatke o temperaturi, relativni vlagi in stanju vrat. Brezžično jih pošlje sprejemniku, ta pa jih prikaže prek spletne strani in vgrajenega zaslona. Oddajni modul je zaradi uporabe stikalnega napajalnika energetsko učinkovit, njegov doseg pa v praksi povsem zadovoljuje razdaljam v okviru stanovanja, na prostem pa do 100 metrov. Poleg tega, da mrežni del vsebuje spletni strežnik, je zmožen tudi trajnega hranjenja zgodovine dogodkov. Celoten sistem se je v okviru testiranja izkazal za zanesljivega, tako s stališča prenosa podatkov kot tudi zanesljivosti delovanja. Izbira komponent je omogočila izdelavo v samogradnji, nekoliko več komentarja pa zahteva njegova cena.

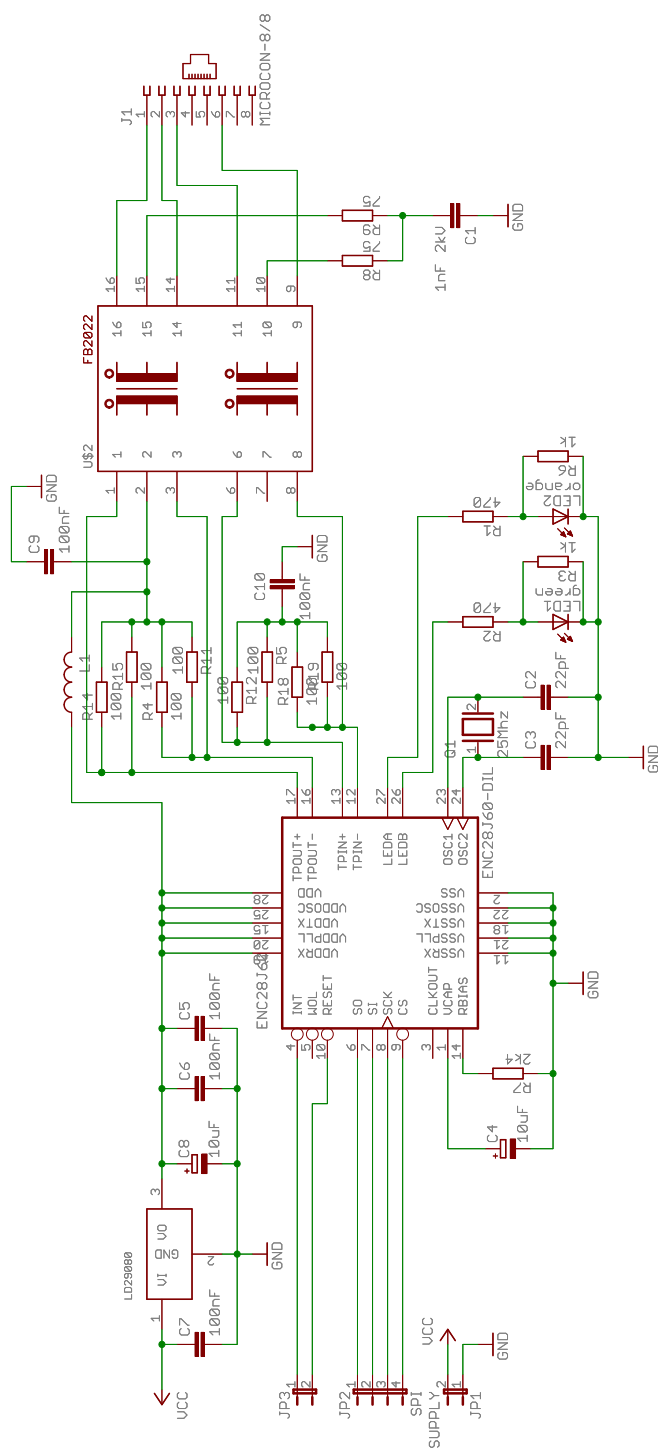
Cena končnega prototipa znaša približno 40 €, s tem da so bili nekateri dražji elementi pridobljeni kot brezplačni vzorci (mikrokrmilniki, mrežni krmilnik, temperaturni senzor in napetostna regulatorja), nekaj cenejših, a težje dostopnih elementov, pa je vzetih iz odsluženih naprav. Če bi kupili tudi te elemente, bi prototip stal dobrih 85 €, kar je že precej visoka cena. Nanjo najbolj vpliva vrednost temperaturnega senzorja SHT71, ki predstavlja 25 % celotne cene. Zanimivo je tudi, da izbrani tipi komponent, ki na eni strani poenostavijo izdelavo prototipa, na drugi strani povišajo njegovo ceno, saj so čipi v večjih ohišjih dražji od tistih v manjših.

Prva izmed izboljšav je zato ravno izbira komponent v manjših ohišjih. Na ta način bi bilo možno celo nadomestiti oba mikrokrmilnika in mrežni krmilnik v sprejemnem in mrežnem modulu z enim samim mikrokrmilnikom, ki združuje vse njihove funkcionalnosti. Primer takega mikrokrmilnika je PIC18F67J60 v ohišju SMD (Surface Mount Device), njegova cena pa je 60 % nižja od vseh treh čipov, katere bi nadomestil. S to izbiro bi se zmanjšala tudi poraba energije mrežnega modula in velikost vezja. Nasploh bi izbira elementov v ohišjih tipa

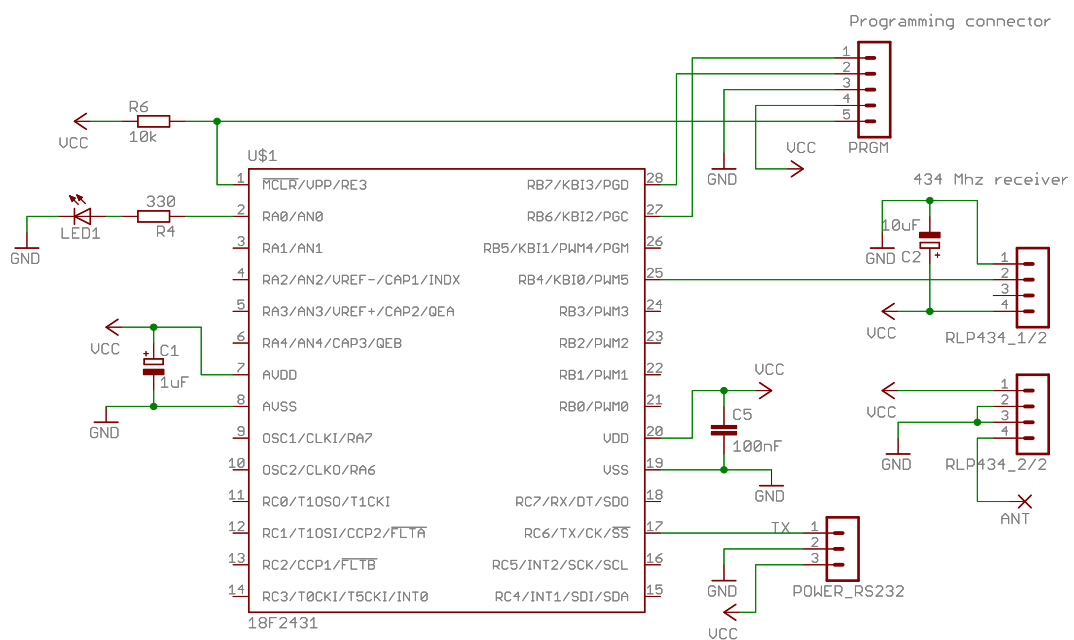
SMD in uporaba dvoplastnih vezij občutno zmanjšala velikost končnega izdelka in stroške materiala.

Prostor za izboljšave je tudi pri izbiri oddajnika in sprejemnika. Na trgu so na voljo sprejemno/oddajni moduli, ki imajo enako ceno kot ločen oddajnik in sprejemnik. Uporabljajo zanesljivejšo frekvenčno modulacijo in že vsebujejo kodirnik ter dekodirnik. Ker vsebujejo sprejemnik in oddajnik, se ponuja še možnost dvosmerne komunikacije.

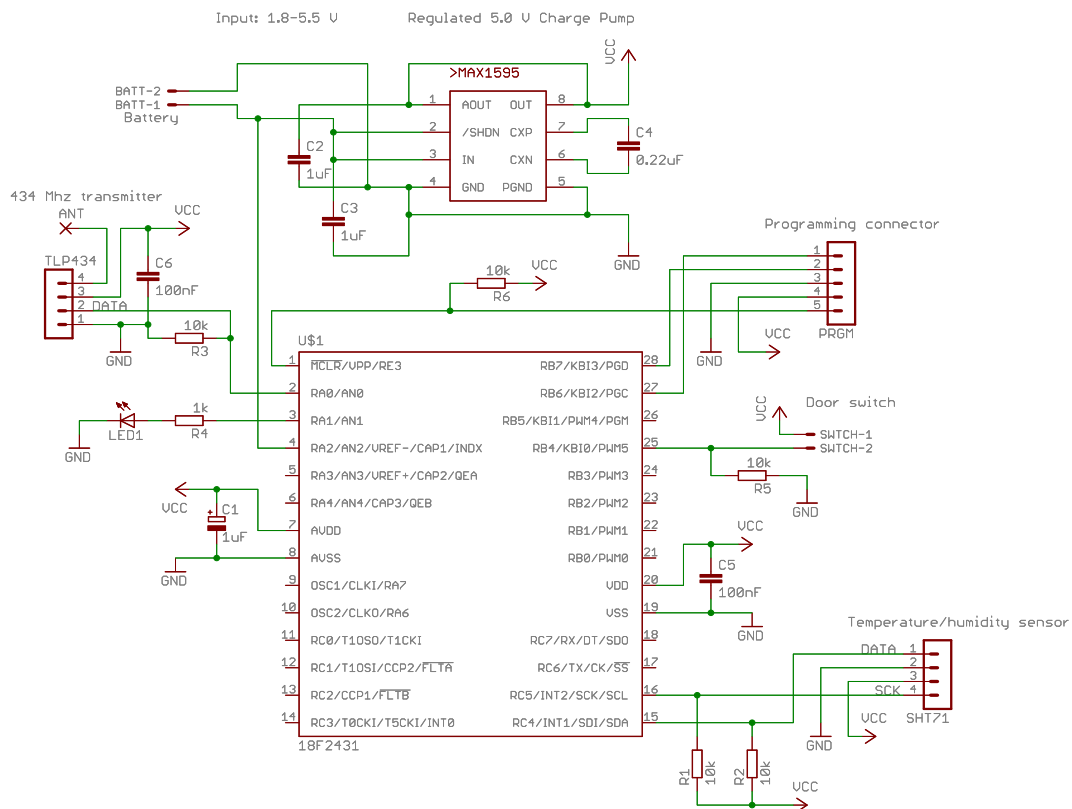
Zanimiva nadgradnja oddajnega modula bi bila njegova popolna energetska avtonomnost, ki bi jo bilo mogoče doseči z uporabo fotovoltaičnih celic. Tudi učinkovitost mrežnega modula bi bila višja, če bi stabilizirano napetost namesto linearnega regulatorja zagotavljal stikalni napajalnik. Veliko potenciala pa ponuja tudi mikrokrmilnik oddajnega modula, ki ima prostih še petnajst nožic, na katere je možno priklopiti še številne senzorje in podobne periferne naprave.



Slika A.2: Električna shema mrežnega dela z mrežnim krmilnikom ENC28J60.



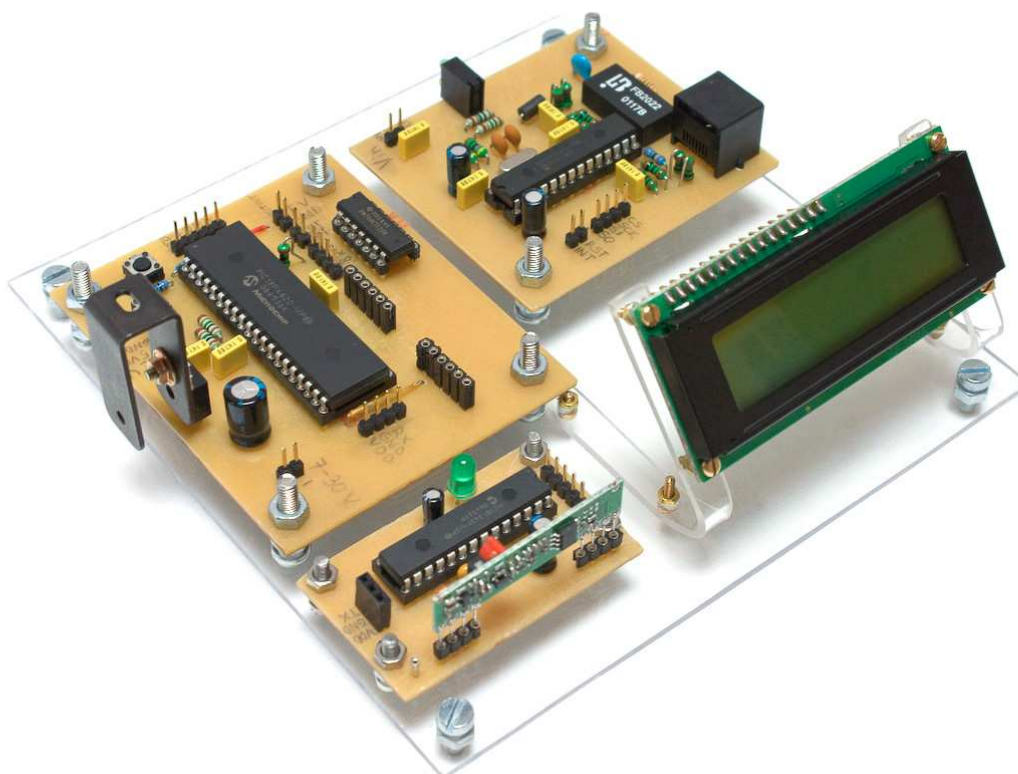
Slika A.3: Električna shema sprejemnega modula z mikrokrmilnikom PIC18F2431 in sprejemnikom RLP434.



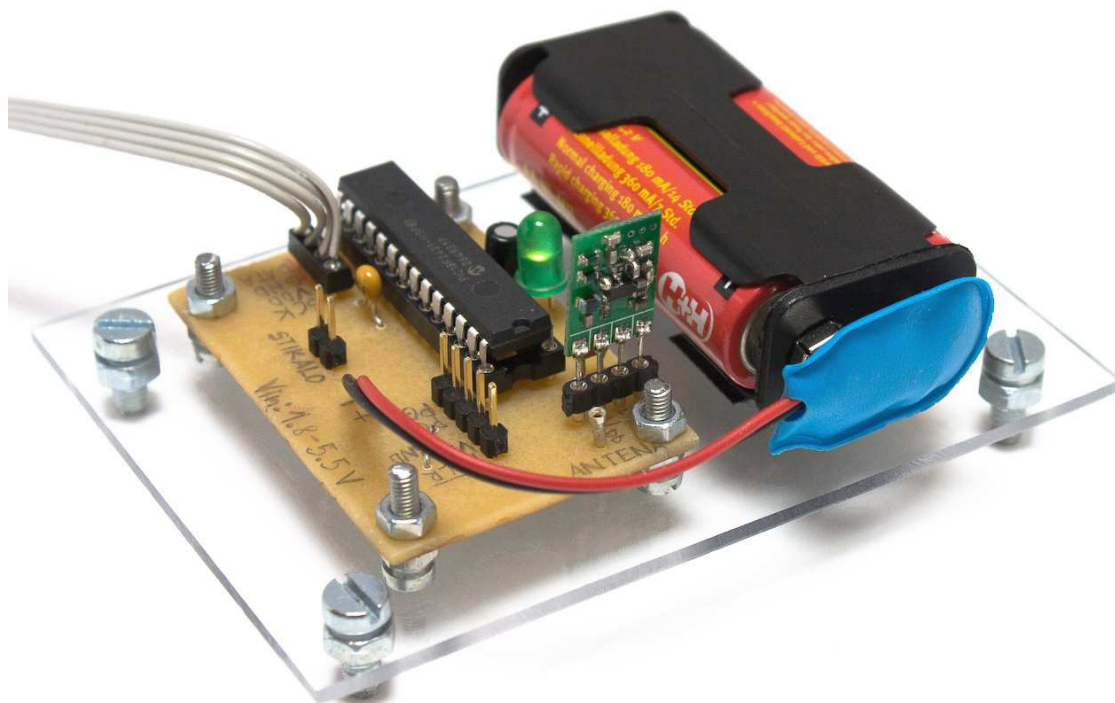
Slika A.4: Električna shema oddajnega modula z mikrokrmilnikom PIC18F2431, oddajnikom TLP434A ter senzorjem SHT71.

Dodatek B

Fotografiji sistema



Slika B.1: Fotografija sprejemnega modula, mrežnega modula (sestavljena iz mikrokrmilniškega in mrežnega dela) ter zaslona LCD.



Slika B.2: Fotografija oddajnega modula z napajalnimi baterijami.

Dodatek C

Zaslonska slika spletne strani



Slika C.1: Zaslonska slika spletne strani.

Literatura

- [1] (2000) Application Note 01 V / UHF Antenna Design, str. 2, 8. Dostopno na:
http://www.numatechnologies.com/pdf/an_antenna.pdf
- [2] (2009) Datasheet SHT7x (SHT71, SHT75) Humidity and Temperature Sensor. Dostopno na:
http://www.sensirion.com/en/pdf/product_information/DatasheethumiditysensorSHT7x.pdf
- [3] (2004) P. Dhond, Selecting the Right Level-Translation Solution. Dostopno na:
<http://focus.tij.co.jp/jp/lit/an/scea035a/scea035a.pdf>
- [4] (2009) Digital modulation, ASK, FSK and PSK. Dostopno na:
<http://www.brightfuture-ic.com/03.htm>
- [5] (2008) ENC28J60 Data Sheet. Dostopno na:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>
- [6] (2008) L78xx Positive voltage regulators, str. 7, Figure 4. Dostopno na:
<http://www.st.com/stonline/products/literature/ds/2143/l7805c.pdf>
- [7] (2007) LD29080 series 800mA Fixed and adjustable output very low drop voltage regulator. Dostopno na:
<http://www.st.com/stonline/products/literature/ds/10918/ld29080xx33.pdf>
- [8] (2009) Microchip: TCP/IP Stack for PIC18, PIC24, dsPIC & PIC32, Figure 2. Dostopno na:
http://microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537041
- [9] (2007) PIC18F2331/2431/4331/4431 Data Sheet, str. 260, Figure 20-5. Dostopno na:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39616C.pdf>

- [10] (2006) PIC18F2455/2550/4455/4550 Data Sheet. Dostopno na:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39626e.pdf>
- [11] (2002) Regulated 3.3V/5.0V Step-Up/Step-Down Charge Pump. Dostopno na:
<http://datasheets.maxim-ic.com/en/ds/MAX1595.pdf>
- [12] RLP434 - RF ASK Hybrid Modules for Radio Control. Dostopno na:
<http://www.laipac.com/Downloads/Easy/rlp434.pdf>
- [13] (2003) SN54HCT125, SN74HCT125 Quadruple bus buffer gates with 3-state outputs. Dostopno na:
<http://focus.ti.com/lit/ds/symlink/sn74hct125.pdf>
- [14] TLP434 - RF ASK Hybrid Modules for Radio Control. Dostopno na:
<http://www.laipac.com/Downloads/Easy/tlp434a.pdf>
- [15] (2006) Uradni list Evropske unije: Odločba komisije z dne 9. november 2006 o uskladitvi radijskega spektra za uporabo naprav kratkega dosega. str. 4-5. Dostopno na:
<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:312:0066:0070:SL:PDF>
- [16] (2009) Wikipedia: Checksum. Dostopno na:
<http://en.wikipedia.org/wiki/Checksum>
- [17] (2009) Wikipedia: Manchester code. Dostopno na:
http://en.wikipedia.org/wiki/Manchester_code