

UNIVERZA V LJUBLJANI
Fakulteta za elektrotehniko

Matej Lenarčič

BREZZIČNO SENZORSKO OMREŽJE
GASILSKIH PREGRINJAL

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: doc. dr. Boštjan Murovec

Ljubljana, januar 2014

Zahvala

Zahvala gre mentorju doc. dr. Boštjanu Murovcu za vse nasvete pri pisanju diplomske naloge, za hitre odgovore na moja vprašanja in za strokovno pomoč. Zahvaljujem se tudi podjetju INEA d.o.o. za možnost sodelovanja pri izdelavi brezžičnega senzorskega omrežja, ter še posebej mag. Igorju Steinerju za usmerjanje in praktične nasvete.

Za podporo med študijem se posebno zahvaljujem družini in zaročenki Tini.

Povzetek

Diplomska naloga opisuje razvoj brezžičnega senzorskega omrežja za meritev temperature in vlage v zaščitnih pregrinjalih za gasilce. Opisan je tudi razvoj računalniške aplikacije, ki omogoča spremljanje prejetih podatkov in opozarjanje uporabnika aplikacije na nevarne razmere, ki jim je oblačilo izpostavljeno. Namen celotnega sistema je povečati varnost gasilcev v situacijah, kjer je ogroženo njihovo življenje.

Na podlagi podanih funkcionalnih in tehničnih zahtev, katerim mora zasnovani sistem ustrezati, se lotimo razvoja funkcionalnega in energijsko učinkovitega sistema. Sprva sta predstavljeni zgradba in zasnova brezžičnih senzorskih omrežij s poudarkom na konceptih, ki bodo uporabni pri zasnovi sistema. Temu sledi načrtovanje koncepta celotnega sistema, ki nam bo kasneje omogočal lažje sprejemanje odločitev. Razvojni proces vključuje izbiro strojne opreme in razvoj programske opreme ter se zaključi s testiranjem in izboljšavo sistema.

Rezultat razvoja je brezžično senzorsko omrežje z računalniško aplikacijo, izdelano za operacijski sistem Windows. Omrežje obsega brezžična vozlišča, ki merijo vrednosti preko štirih senzorjev temperature in vlage. Vozlišča opozorijo uporabnika na nevarne razmere z uporabo svetlobne in vibracijske pozivne naprave. Vsi izmerjeni podatki se brezžično pošljejo na USB modul, ki je priključen na osebni računalnik, kjer uporabnik aplikacije spremlja izmerjene vrednosti iz pregrinjal.

Ključne besede: brezžično senzorsko omrežje, zaščitno pregrinjala za gasilce, merjenje temperature in vlage, računalniška aplikacija

Abstract

This thesis describes the development of a wireless sensor network that measures temperature and humidity in protective clothing for firefighters. Also described is the development of a computer application that enables monitoring of received data and alerting the user to exposed dangerous conditions. The purpose of the entire system is to increase the safety of firefighters in life-threatening situations.

On the basis of given functional and technical requirements, which the designed system must meet, we begin the development of a functional and energy efficient system. Firstly we examine the structure and design of wireless sensor networks, with an emphasis on the concepts that will be useful when designing the system. This is followed by planning a concept for the entire system, which will later enable easier decision making. The design process includes the selection of hardware and the development of software and is concluded by testing and improving the system.

The result of the development is a wireless sensor network with a computer application, designed for the Windows operating system. The network is comprised of wireless nodes that measure values with four sensors of temperature and humidity. The nodes alert the firefighter to unsafe conditions with the use of a light and vibrational alerting device. All measured data is wirelessly sent to the USB module, which is plugged into the personal computer, where the user of the application monitors the measured values in the clothing.

Key words: wireless sensor network, protective clothing for firefighters, measurement of temperature and humidity, computer application

Kazalo vsebine

1. Uvod v brezžična senzorska omrežja.....	1
1.1 Zgradba dela	2
1.2 Brezžično senzorsko omrežje	2
1.3 Funkcionalne zahteve.....	3
1.4 Tehnične zahteve	3
2. Zgradba brezžičnih senzorskih omrežij.....	5
2.1 Različne aplikacije uporabe BSO.....	5
2.1.1 Primeri aplikacij	6
2.1.2 Napovedi in izzivi komercializacije.....	8
2.2 Sestavni deli BSO	8
2.3 Strojna oprema senzorskega vozlišča.....	10
2.3.1 Krmilnik.....	11
2.3.2 Napajalni vir	12
2.3.3 Senzorji in aktuatorji	12
2.3.4 Komunikacijski podsistem.....	13
2.3.5 Pomnilnik	13
2.4 Komunikacija med vozlišči.....	14
2.4.1 Radijska komunikacija	14
2.4.2 Protokoli za dostop do medija	15
2.4.3 Višje komunikacijske plasti	16
2.5 Energijski pretvorniki	17
2.5.1 Fizikalni principi delovanja.....	17
2.5.2 Senzorji in aktuatorji	19
2.5.3 Merjenje temperature s p-n spojem.....	20

2.5.4	Merjenje vlažnosti s kondenzatorjem	21
3.	Zasnova brezžičnih senzorskih omrežij	23
3.1	Faktorji zasnove.....	23
3.1.1	Upravljanje s sistemom.....	24
3.1.2	Zgradba sistema	24
3.1.3	Delovanje sistema.....	25
3.2	Osnovni principi zasnove.....	26
3.2.1	Splošni principi.....	26
3.2.2	Omrežno procesiranje	26
3.2.3	Podatkovna centričnost	27
3.3	Energijska učinkovitost vozlišča	28
3.3.1	Načini obratovanja mikrokrmilnika.....	28
3.3.2	Radijska komunikacija.....	28
3.3.3	Procesiranje	29
3.4	Zasnova brezžične komunikacije	30
3.4.1	Vrednotenje z decibeli	30
3.4.2	Doseg radijske komunikacije	31
3.4.3	Kakovost brezžične komunikacije	32
4.	Izdelava koncepta brezžičnega omrežja	33
4.1	Komunikacijska topologija	33
4.2	Inteligentno zaščitno pregrinjalo.....	34
4.3	Računalnik z USB modulom in aplikacijo.....	35
4.4	Programska oprema sistema	36
4.4.1	Programska oprema vozlišča IZP	36
4.4.2	Programska oprema centralnega vozlišča.....	37
4.4.3	Računalniška aplikacija IZP Nadzor	37

5. Izbira strojne opreme	39
5.1 Brežžična vozlišča JeeLabs	39
5.2 Primerjava senzorjev.....	41
5.3 Kombinirani senzor SHT-11	44
5.3.1 Pretvorba merilnih podatkov	45
5.4 Svetlobni in vibracijski pozivnik.....	46
5.5 Napajalni vir in polnilnik	47
6. Izdelava programske opreme	49
6.1 Arduino okolje.....	49
6.2 Programska oprema vozlišča IZP	51
6.2.1 Funkcija setup()	51
6.2.2 Funkcija loop()	52
6.2.3 Funkcija doMeasure()	52
6.2.4 Funkcija doReport().....	53
6.2.5 Zgradba komunikacijskih paketov	54
6.3 Programska oprema centralnega vozlišča	55
6.3.1 Funkcija setup()	55
6.3.2 Funkcija loop()	55
7. Izdelava računalniške aplikacije	57
7.1 Scenariji uporabe	57
7.1.1 Nastavitev.....	58
7.1.2 Testiranje.....	59
7.1.3 Aktivna in pasivna uporaba.....	60
7.2 Razvoj aplikacije.....	61
7.2.1 Niti aplikacije	61
7.2.2 Shranjevanje zgodovine.....	62

7.2.3 Alarmiranje	63
7.3 Uporabniški vmesnik	63
7.4 Zgradba programske kode	65
8. Testiranje in izboljšava sistema	67
8.1 Meritev dosega brezžične komunikacije	67
8.2 Zamenjava radijskega modula	69
8.3 Izračun življenjske dobe baterije	70
8.4 Testiranje sistema	70
9. Zaključek	75
10. Priloge	77
10.1 Električni shemi vezij pozivnikov LilyPad	77
10.2 Električni shemi vezij brezžičnih vozlišč JeeLabs	78
10.3 Električna shema vezja izboljšanega vozlišča	80
10.4 Seznam slik	81
10.5 Seznam tabel	82
11. Viri in literatura	83

Seznam uporabljenih simbolov

ADC	Analog-to-digital converter	Analogno digitalni pretvornik
ASIC	Application-specific integrated circuit	Integrirano vezje po meri
BER	Bit error rate	Razmerje napačnih bitov
CLR	Common Language Runtime	Izvajalno okolje s skupnim jezikom
CRC	Cyclic redundancy check	Preverjanje s ciklično redundanco
CSV	Comma-separated values	Vrednosti ločene z vejico
dBi	Decibel isotropic	Izotropni decibel
dBm	Decibel milliwatt	Decibel milivat
DSP	Digital signal processor	Digitalni signalni procesor
EEPROM	Electrically erasable programmable read-only memory	Električno izbrisljiv in programirljiv bralni pomnilnik
ETSI	European Telecommunications Standards Institute	Evropski institut za telekomunikacijske standarde
FCC	Federal Communications Commission	Zvezna komisija za komunikacije
FEC	Forward error correction	Vnaprejšnje popravljanje napak
FPGA	Field-programmable gate array	Poljsko programirljiva matrika vrat
FSK	Frequency-shift keying	Diskretna frekvenčna modulacija
I ² C	Inter-Integrated Circuit	Vodilo med integriranimi vezji
IDE	Integrated development environment	Integrirano razvojno okolje
ISM	Industrial, scientific and medical	Industrijski, znanstveni in medicinski
LED	Light-emitting diode	Svetilna dioda
MAC	Media access control	Nadzor dostopa do medija
OSI	Open Systems Interconnection	Medsebojna povezava odprtih sistemov
OTP	One-time programmable	Enkratno programirljiv
PER	Packet error rate	Razmerje napačnih paketov
QoS	Quality of service	Kvaliteta storitve
RAM	Random-access memory	Bralno-pisalni pomnilnik
RISC	Reduced instruction set computing	Računanje z zmanjšanim naborom ukazov
ROM	Read-only memory	Bralni pomnilnik

VI

RTS/CTS	Request to send/clear to send	Zahteva za pošiljanje/dovoljenje za pošiljanje
SMD	Surface-mount device	Naprava za površinsko namestitev
SNR	Signal-to-noise ratio	Razmerje signal-šum
SQL	Structured Query Language	Strukturiran poizvedbeni jezik
UART	Universal Asynchronous Receiver/Transmitter	Univerzalni asinhroni sprejemnik/oddajnik
UML	Unified Modeling Language	Poenoten jezik za modeliranje
USART	Universal Synchronous/Asynchronous Receiver/Transmitter	Univerzalni sinhroni/asinhroni sprejemnik/oddajnik
USB	Universal Serial Bus	Univerzalno serijsko vodilo
UWB	Ultra-wideband	Ultra širokopasovno
VHF	Very high frequency	Zelo visoka frekvenca
VLF	Very low frequency	Zelo nizka frekvenca
WSN	Wireless sensor network	Brezžično senzorsko omrežje

1. POGLAVJE

Uvod v brezžična senzorska omrežja

The problems of the world cannot possibly be solved by skeptics or cynics whose horizons are limited by the obvious realities. We need men who can dream of things that never were.

- John F. Kennedy

Za brezžična senzorska omrežja (angl. Wireless Sensor Networks – WSN), krajše BSO, v literaturi najdemo vrsto različnih definicij. Pogosto posamezne definicije izključujejo določeno podmnožico sistemov, ki bi jih prav tako lahko uvrstili pod BSO. Zaradi tega BSO preprosto definiramo kot omrežje prostorsko porazdeljenih vozlišč (angl. node, mote), ki s senzorji merijo fizikalne veličine in med seboj brezžično komunicirajo.

Področje BSO je zanimivo z akademskega vidika, saj prinaša vrsto izzivov, katerih rešitev zahteva poglobljene raziskave, tako teoretične, kot eksperimentalne. Vedno bolj postaja zanimivo tudi z gospodarskega vidika, ko padajoče cene senzorjev, procesiranja in radijske komunikacije omogočajo ekonomsko upravičenost brezžičnih merilnih sistemov. Področje je zelo interdisciplinarno, saj vsebuje znanja elektrotehnike in računalništva, pogosto pa je še pomembnejše znanje, ki se dotika posamezne aplikacije v kateri se BSO uporablja.

Namen te diplomske naloge je predstavitev razvoja BSO, ki meri temperaturo in vlago v temperaturno odpornih pregrinjalih za gasilce. Predstavljen je tudi razvoj računalniške aplikacije, ki omogoča prejemanje, shranjevanje in alarmiranje izmerjenih vrednosti. BSO je bil razvit v podjetju INEA d.o.o. iz Ljubljane, ki se ukvarja z avtomatizacijo, energetiko in proizvodno informatiko.

1.1 Zgradba dela

Namen uvoda je vpeljava bralca v tematiko BSO ter opredelitev namena in ciljev dela. Na kratko je predstavljen BSO za gasilce in v nadaljevanju so podane funkcionalne in tehnične zahteve, ki jih moramo doseči.

V začetku drugega poglavja je predstavljenih nekaj zanimivih primerov aplikacij BSO iz različnih področij nato pa so opisani najpogostejši sestavni deli tipične BSO ter njihove značilnosti. V naslednjih dveh podpoglavjih sta podrobneje predstavljena komunikacija in senzorji, s poudarkom na merjenju temperature in vlažnosti. Celotno poglavje bralca vpelje v tehnični del BSO.

V tretjem poglavju je predstavljena problematika zasnove BSO. Opisani so faktorji ter osnovni principi zasnove. Predstavljeni so teoretični pristopi k večanju energijske učinkovitosti in dosega brezžične komunikacije vozlišča.

V naslednjih poglavjih je opisan razvoj BSO za meritev temperature in vlage v oblačilu za gasilce ter računalniške aplikacije, ki omogoča odjemanje in prikaz podatkov. Na podlagi zahtev, ki so podane v uvodu, je predstavljen koncept BSO, prilagojen naši aplikaciji. Zasnove se lotimo z izbiro strojne opreme, čemur sledi programiranje vozlišč in nato izdelava računalniške aplikacije. Na koncu je opisano še testiranje sistema in izboljšanje dosega brezžične komunikacije.

1.2 Brezžično senzorsko omrežje

Glavna naloga celotnega sistema je spremljanje temperature in vlage na uporabnikih t. i. inteligentnega zaščitnega pregrinjala (IZP) preko aplikacije, ki je nameščena na osebni računalnik. IZP je pregrinjalo, izdelano iz temperaturno odpornih materialov, ki ščitijo uporabnika (gasilca) pred nevarnostjo izpostavitve visokim temperaturam. Pregrinjalo ima vsite komponente brezžičnega vozlišča. Izmerjene vrednosti senzorjev se brezžično pošiljajo centralnemu vozlišču, ki je priključeno na osebni računalnik. Računalniška aplikacija komunicira s centralnim vozliščem in prejema izmerjene vrednosti posameznih IZP-jev ter skrbi za nadaljnjo obdelavo informacij.

Takšen sistem omogoča daljinski nadzor nad uporabniki IZP-jev v situacijah, kjer je potencialno ogroženo njihovo življenje. Pri gašenju požarov se trenutno pogosto

zanašamo na to, da gasilci sami presodijo, kdaj se je potrebno umakniti zaradi previsoke temperature. S tem sistemom uporabnik aplikacije (nadzornik) na varni razdalji spremlja situacijo in gasilce obvesti, da naj se umaknejo, če so izmerjene vrednosti previsoke.

Preden se lotimo zasnove sistema, želimo natančno razjasniti, kakšne so funkcionalne in tehnične zahteve, ki jim mora zasnovani sistem ustrezati. Zahteve je v veliki meri podal naročnik, nekatere pa smo dodali tudi sami.

1.3 Funkcionalne zahteve

Na vsakem pregrinjalu želimo odjemati vrednosti temperature in vlage na štirih lokacijah (dve spredaj in dve zadaj) na notranji strani oblačila, kar pomeni, da potrebujemo štiri senzorje temperature in vlage. Senzorji naj zagotavljajo podatke o temperaturnem stanju in o temperaturnih gradientih pod oblačilom. Žično naj bodo povezani s centralno enoto IZP, ki naj bo na pregrinjalu nameščena na najbolj primerni lokaciji. Centralna enota naj minimalno obsega mikrokrmilnik in radijsko-frekvenčni oddajnik. Njena naloga je periodično odjemanje vrednosti s senzorjev, obdelava podatkov in brezžično pošiljanje vrednosti centralnemu vozlišču. Centralna enota potrebuje vir napajanja, ki je na IZP nameščen na drugi, bolj primerni lokaciji. Omogočeno mora biti polnjenje vira napajanja. S centralno enoto naj bo žično povezan svetlobni pozivnik, ki naj bo na IZP nameščen na takšni lokaciji, da je v vidnem polju uporabnika IZP. Uporabi se lahko tudi druga vrsta pozivnika, pomembno je le, da je učinkovit pri opozarjanju uporabnika na nevarnost. Ob previsoki izmerjeni temperaturi se pozivnik sproži.

Centralno vozlišče, ki skrbi za prejemanje izmerjenih podatkov od IZP vozlišč, naj bo priklopljeno na osebni računalnik preko USB priključka. Naloga aplikacije je poleg odjemanja tudi prikaz podatkov na uporabniškem vmesniku, opozarjanje uporabnika aplikacije na presežene mejne vrednosti temperature in vlage na IZP-jih (alarmiranje) ter shranjevanje prejetih vrednosti skupaj s časom in datumom (logiranje) v poljubno vrsto datoteke na trdem disku računalnika, kar omogoča kasnejši pregled dogodkov in izmerjenih vrednosti.

1.4 Tehnične zahteve

Vse elektronske komponente naj imajo čim nižjo porabo električne energije, naj bodo dolgoročno zanesljive ter odporne na tresljaje, prah, vlago ter nizke in visoke

temperature. Zaželena je tudi majhna fizična velikost in čim boljša cenovna ugodnost. Priporoča se uporabo komponent v SMD tehnologiji, kar prispeva k majhnosti sistema. Za centralno enoto se zahteva zadostna hitrost izvajanja (procesiranja) in čim manjša poraba električne energije. Vir napajanja naj sestavlja baterija, ki omogoča ponovno polnjenje in naj omogoča samostojno delovanje vozlišča najmanj 24 ur. Za brezžično komunikacijo naj se uporabi nelicenčni pas ISM, predvidoma v frekvenčnem razponu 866 MHz – 868 MHz. Predviden doseg brezžične komunikacije je 300 m na prostem in do 30 m v zgradbah.

Osebni računalnik ima nameščen operacijski sistem Microsoft Windows XP ali Microsoft Windows 7, zato mora biti računalniška aplikacija, ki odjema podatke s centralnega vozlišča, kompatibilna s tema dvema operacijskima sistemoma. Aplikacija naj ima tudi možnost delovanja v ozadju, tako da je dostopna preko orodne vrstice. Celoten sistem mora delovati nemoteno tudi v primeru različnih napak oz. izjem.

Elektronske komponente IZP-ja med uporabo ne smejo biti izpostavljene previsokim ali prenizkim temperaturam, ki presegajo meje delovanja elektronike. Ne pričakuje se, da bi komponente bile odporne na neposreden stik z vodo, zato morajo biti primerno zaščitene.

2. POGLAVJE

Zgradba brezžičnih senzorskih omrežij

Fools ignore complexity. Pragmatists suffer it. Some can avoid it. Geniuses remove it.

- Alan Perlis

Princip redukcionizma v znanosti pravi, da je vsak sistem le vsota njegovih delov. Delitev na podсистeme ima mnogo prednosti, vprašanje pa je kdaj je še smiselna. Pri kompleksnih sistemih so deli medsebojno močno povezani z velikim številom korelacij, ki nam pogosto niso v celoti razumljive, kaj šele da bi jih matematično modelirali. Takrat reduciramo sistem na sestavne dele in upoštevamo le najpomembnejše korelacije med njimi, ali pa pristopimo k problemu celovito (t. j. sistemsko) in se naravnost soočimo z dejstvom, da gre za kompleksen sistem. Za celovit pristop k problemu potrebujemo predvsem veliko znanja in izkušenj z danega področja.

BSO-ji so kompleksni sistemi in jih zato pogosto obravnavamo celovito. Vseeno v tem poglavju pristopimo k zgradbi BSO z nekaj redukcionizma, ki nam omogoča lažje razumevanje ter pridobitev potrebnega znanja za celovit pristop.

2.1 Različne aplikacije uporabe BSO

Da dobimo vpogled v fleksibilnost BSO in da raziščemo različne robne pogoje pri njihovi zasnovi, so v nadaljevanju predstavljene raznovrstne uporabne aplikacije. Iz različnih področij si bomo pogledali primere aplikacij, pri čemer nas bodo zanimale predvsem takšne, ki najbolj odstopajo. Morda se zdi, da je kakšna predstavljena aplikacija v tem času neekonomična, nepraktična ali morda celo nesmiselna, zato je predstavljenih tudi nekaj napovedi in izzivov glede komercializacije sistemov BSO.

2.1.1 Primeri aplikacij

Na področju **kmetijstva** najdemo veliko število aplikacij, ki močno prispevajo k večji kakovosti produkta ali k boljšemu gospodarjenju z viri. Te aplikacije se v literaturi pogosto pojavljajo v sklopu sistemov natančnega kmetovanja (angl. precision agriculture), ki za pravilno delovanje potrebujejo natančne in ažurne informacije o stanju prsti. Primer takšne aplikacije je sistem brezžičnih podzemnih senzorjev, ki merijo vlažnost prsti in te podatke sporočajo sistemu za irigacijo, ta pa se nato inteligentnejše odloča o lokaciji in stopnji irigacije [1, str. 1-2]. Takšna omrežja se imenujejo brezžična podzemna senzorska omrežja (angl. wireless underground sensor networks – WUSN). Eden največjih izzivov pri zasnovi takšnih omrežij je izdelava učinkovitejših anten, ki kljubujejo velikim izgubam pri brezžični komunikaciji skozi zemljo. Na komunikacijo močno vpliva tudi globina senzorjev, sestava in vlažnost tal ter višina rastlinja.

Veliko je govora o inteligentni **avtomatizaciji poslovnih in stanovanjskih zgradb**, kar v literaturi zasledimo tudi pod izrazi inteligentni ali pametni domovi oz. zgradbe. Sledeči primer pa se nanaša na dejavnost spremljanja strukturnega stanja zgradb (angl. structural health monitoring – SHM). Gre za obširno tehnično področje, ki omogoča spremljanje sprememb določenih lastnosti zgradb in z uporabo statističnih ter drugih prijemov ugotavlja njeno stanje. Uporaba takšnih tehnologij je še posebej pomembna pri zahtevnejših zgradbah, kot so nebotičniki, mostovi in tuneli, ter na lokacijah, ki so izpostavljene naravnim katastrofam (npr. potresna območja). Pogosto so lokacije, kjer želimo postaviti brezžična vozlišča, težko dostopne in nimajo možnosti priključitve na električno omrežje. Zaradi tega želimo izdelati takšna vozlišča, ki se samodejno napajajo z uporabo tehnologij zbiranja energije iz okolice (angl. energy harvesting/scavenging). Primer takšnega sistema je brezžično senzorsko omrežje za ugotavljanje strukturnega zdravja podzemnih tunelov za mestno podzemno železnico [2]. Brezžična vozlišča so opremljena s senzorji, ki merijo natezno napetost sten tunelov in so nameščena na primerne lokacije. Zaradi pomanjkanja zunanega električnega napajanja se napajajo z majhnimi vetrovnimi turbinami, ki proizvedejo električno energijo ob vsaki vožnji vlaka mimo vozlišča. Zbrana električna energija se shrani v superkondenzatorju in omogoča izvedbo meritev ter pošiljanje podatkov po omrežju do centralnega vozlišča.

Na področju **avtomobilizma** najdemo različne predloge, kako avtomobile v bližini medsebojno povezati, pogosto z namenom preprečevanja avtomobilskih nesreč (npr. z opozarjanjem na zastoje). Za takšne aplikacije se pogosto uporablja izraz avtomobilska *ad hoc* omrežja (angl. vehicular ad hoc network – VANET). Značilno za takšna omrežja je, da omogočajo povezavo avtomobilov do nekaj sto metrov ter da so topološko zelo dinamična (avtomobili se neprestano vklapljajo in izklapljajo iz omrežij). Omeniti velja, da se posamezna vozlišča napajajo iz avtomobilskega akumulatorja ali baterij (v primeru električnih vozil), zaradi česar optimizacija porabe energije ni prioriteta.

Med **industrijske aplikacije** uvrstimo BSO, katerih cilj je ugotavljanje lokacije podvodnih robotov, ki imajo običajno omejene energijske zaloge, zaradi česar želimo izvesti čim bolj energijsko učinkovito lokalizacijo [3, str. 27-29]. Podvodna akustična senzorska omrežja (angl. underwater acoustic sensor networks – UASN) se uporabljajo tudi za lokalizacijo avtonomnih podvodnih vozil (angl. autonomous underwater vehicles). Za spremljanje bazenov nuklearnih odpadkov se robot, opremljen z akustičnimi senzorji samodejno lokalizira s pomočjo stacionarnih akustičnih oddajnikov, kar omogoča učinkovit nadzor bazena. Oddajniki so v drugih aplikacijah nameščeni tudi na plavajoče boje ali na dno oceana.

Zanimive primere uporabe najdemo na področju **varnosti** na lokacijah, kjer je ogroženo človeško življenje. Veliko razvojnih sredstev se namenja razvoju letelih plovil brez posadke (angl. unmanned aerial vehicle – UAV), ki bi jih sicer same po sebi težko uvrstili v področje BSO. V danem primeru pa se uporabljajo kot mobilna brezžična senzorska vozlišča, skupaj s statičnimi senzorskimi vozlišči, nameščenimi na tleh, katerih naloga je nadzor in varovanje izbranega področja [4, str. 10-12]. Za UAV se uporabi multikopter (npr. kvadkopter), ki je opremljen z naprednimi senzorji (npr. kamera za vidno ali infrardečo svetlobo), medtem ko senzorji na tleh služijo zaznavanju dogodkov, ki ogrožajo varnostne razmere na področju. Mobilna vozlišča preletavajo območje glede na vgrajene algoritme in v primeru zaznanega dogodka talnih senzorjev obiščejo to lokacijo. Delovati morajo usklajeno in se medsebojno dopolnjevati tako, da pokrivajo čim večji del nadzorovanega področja, kar dosežejo z brezžično komunikacijo med vozlišči. BSO, ki ga sestavljajo izključno ali pretežno mobilna brezžična vozlišča, imenujemo mobilno brezžično senzorsko omrežje (angl. mobile WSN – MWSN). V tem primeru gre za hibridno omrežje statičnih in mobilnih brezžičnih senzorskih vozlišč.

Nove tehnologije se pogosto najprej uporabijo v **vojaških aplikacijah**. Še posebej pereča težava za vojsko je lokalizacija ostrostrelcev, saj ti predstavljajo veliko nevarnost pehoti. Za ta namen obstaja več komercialnih sistemov, ki uporabljajo različne tehnologije (npr. infrardeča kamera). Dan BSO uporablja akustične senzorje in omogoča natančnost izračuna lokacije do 1 m in zakasnitev lokalizacije manjšo od 2 sekund [5]. Pred uporabo se vozlišča časovno sinhronizirajo, vendar večjo težavo predstavlja samodejna lokalizacija vozlišč, ki je nujna za izračun lokacije strelca. Del izračuna se opravi na vozliščih, zato da se zmanjša količina poslanih podatkov. Končni izračun se izvede na računalniku.

2.1.2 Napovedi in izzivi komercializacije

Podjetje IDTechEx, ki izvaja tržne raziskave, je opravilo raziskavo o prihodnosti tržnega potenciala BSO [6]. Po izsledkih raziskave bo celotna tržna vrednost BSO v desetih letih zrasla z 0,45 milijarde dolarjev v letu 2012 na 2 milijardi dolarjev v letu 2022. Druga raziskava postavlja velikost trga le za industrijske BSO na 4 milijarde dolarjev do leta 2017 [7]. Najpomembnejši izzivi, ki zavirajo realizacijo takšne rasti, se dotikajo področja komunikacije in energijskih virov. V prihodnosti se pričakuje pospešena standardizacija komunikacijskih tehnologij, novi napredki v tehnologijah virov energije in posledično daljše življenjske dobe BSO. Najpomembneje pa naj bi k razvoju BSO pripomogli nižji stroški procesiranja in odjemanja, večje energijske gostote virov energije ter zavedanje podjetij in posameznikov o potencialu večje povezanosti, sledljivosti in merljivosti.

2.2 Sestavni deli BSO

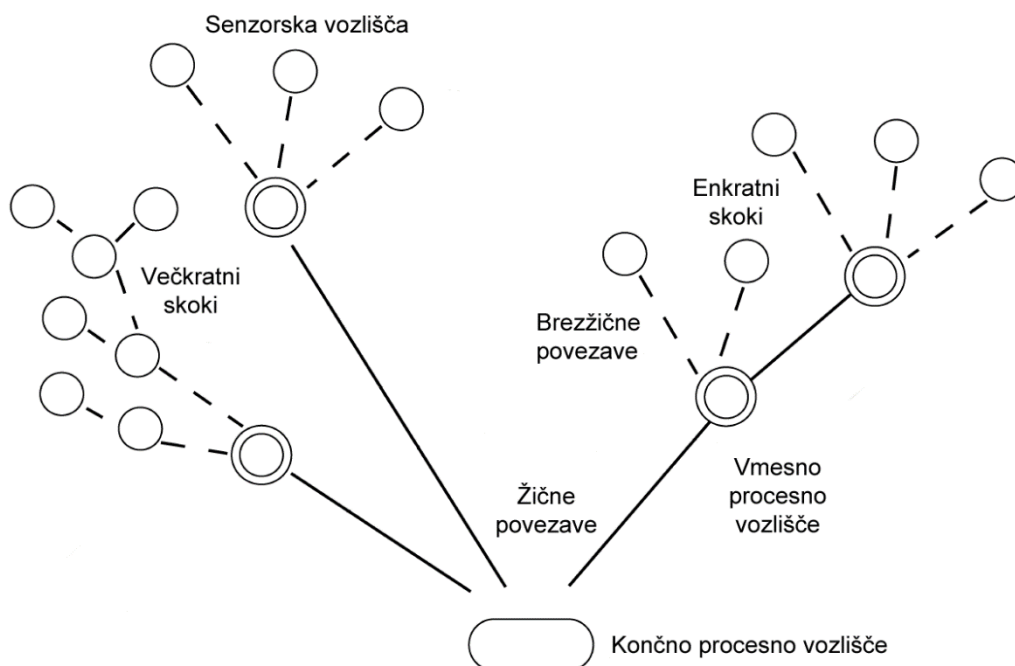
Večino BSO lahko po posameznih funkcijah razdelimo na štiri glavne sestavne dele: (1) sklop senzorjev in aktuatorjev, (2) brezžično povezovalno omrežje, (3) centralna točka izmenjave informacij in (4) množica računskih zmogljivosti. S temi funkcionalnostmi zgradimo BSO, katerega naloga je

- s senzorji odjemati in preko aktuatorjev vplivati na fizikalne veličine, ter
- pošiljati in prejemati informacije preko brezžičnega povezovalnega omrežja,
- pošiljati in prejemati informacije skozi centralno točko izmenjave informacij in
- z uporabo računskih zmogljivosti na vseh nivojih učinkovito upravljati s podatki.

BSO-je v grobem razdelimo v dve kategoriji [8, str. 7]. BSO prve kategorije so skoraj vedno prepletena (angl. mesh) omrežja s komunikacijo z večkratnimi skoki (angl. multi-

hop) med posameznimi vozlišči, ki povečini uporabljajo dinamično usmerjanje (komunikacijska pot od izvirnega vozlišča do centralnega vozlišča se lahko spreminja). BSO druge kategorije so običajno sistemi, kjer vozlišča komunicirajo od točke to točke (angl. point-to-point) ali od večih točk do ene točke (angl. multipoint-to-point) z uporabo komunikacije z enkratnimi skoki (angl. single-hop), ki običajno uporabljajo statično usmerjanje (za komunikacijo med izvirnim vozliščem in centralnim vozliščem se uporablja le ena pot). BSO prve kategorije običajno sestavlja veliko število vozlišč na večji površini za razliko od BSO druge kategorije, kjer je število vozlišč običajno manjše in obratuje tudi na manjši površini. Dana kategorizacija služi lažji razčlenitvi zahtev pri zasnovi BSO, saj ostre meje med kategorijama ni mogoče načrtati.

Na sliki 2.1 je prikazan primer značilne BSO, ki jo uvrstimo v drugo kategorijo, kljub temu da na skrajnem levem delu omrežja uporablja komunikacijo z večkratnimi skoki. Iz slike je razvidno, da uporablja statično usmerjanje, saj ima vsako vozlišče le eno komunikacijsko pot do centralnega vozlišča. Topologija omrežja (t. j. konfiguracija komunikacije) je drevesna, kar ponovno ne ustreza opisu druge kategorije. Pri takšni topologiji je priporočeno, da so vozlišča, skozi katere se vrši komunikacija zelo zanesljiva, da ne pride do izpada celotne veje omrežja v primeru nepravilnega delovanja enega vozlišča. Razvidno je tudi, da omrežje uporablja tako žični kot brezžični medij komunikacije. Zaradi visoke zanesljivosti je zaželeno, da se, kjer je to mogoče in smiselno, uporabljajo žične povezave med vozlišči. Nazadnje omenimo še komunikacijo z enkratnimi skoki, ki se uporablja v desni in sredinski veji ter komunikacijo z večkratnimi skoki, ki jo izkorišča skrajna leva veja omrežja. Pri komunikaciji z večkratnimi skoki, ki se običajno uporablja v prepletenih omrežjih, posamezna brezžična vozlišča delujejo kot komunikacijski posredniki, običajno zato, da se zmanjša poraba energije za prenos informacij, na račun večje zakasnitve prenosa podatkov. Na drugi strani pri komunikaciji z enkratnimi skoki brezžična vozlišča delujejo kot končni komunikacijski elementi v omrežju in ne omogočajo pošiljanja prejetih informacij drugim vozliščem.

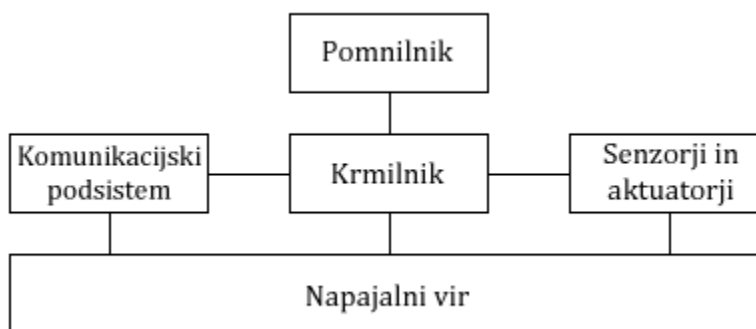


Slika 2.1: Primer značilnega brezžičnega senzorskega omrežja.

Omeniti velja, da je glede na vrsto BSO lahko nepogrešljiv del sistema tudi informacijski sistem, ki skrbi za nadaljnjo obdelavo prejetih podatkov, arhiviranje in predstavitev podatkov končnim uporabnikom. Tak sistem lahko dodaja tudi možnost ročnega posega in nadzora nad delovanjem BSO ter nenazadnje omogoča integracijo BSO z drugimi informacijskimi sistemi. Zasnova podpornih informacijskih sistemov spada med ključne dejavnike uspešnega delovanja BSO.

2.3 Strojna oprema senzorskega vozlišča

Strojno opremo posameznega senzorskega vozlišča ponazorimo s petimi glavnimi komponentami [9, str. 18-19], ki so prikazane na sliki 2.2. Pogosto imajo vozlišča tudi druge komponente, težko pa si jih predstavljamo brez katerekoli izmed podanih komponent. Obstajajo tudi izjeme, kot na primer vozlišče brez senzorjev in aktuatorjev, ki s komunikacijsko napravo ugotavlja le bližino ostalih vozlišč preko moči prejetega signala in ne meri nobene druge fizikalne veličine. V takšnem primeru prevzame naprava za komunikacijo tudi nalogo senzorja.



Slika 2.2: Glavne komponente preprostega senzorskega vozlišča.

2.3.1 Krmilnik

Osrednji del brezžičnega vozlišča je krmilnik. Najširše ga opišemo kot diskreten podsistem, ki ob poljubnih trenutkih prejema podatke iz ostalih komponent, nad njimi izvaja različne aritmetične in logične operacije ter rezultate pošilja ostalim komponentam. Najpogostejše naloge krmilnika so [9, str. 19]:

- branje vrednosti s senzorjev,
- procesiranje prebranih podatkov,
- pošiljanje podatkov ob izbranih trenutkih,
- prejemanje podatkov iz ostalih vozlišč ter
- upravljanje z aktuatorji.

Za opravljanje naloge krmilnika je v večini BSO najprimernejši mikrokrmilnik, ki je v osnovi računalnik na integriranem vezju s procesorjem, pomnilnikom in programirljivo vhodno-izhodno periferijo. Obstajajo tudi druge rešitve: DSP (angl. digital signal processor) je primeren za procesiranje velike količine vektorskih podatkov, klasičen FPGA (angl. field-programmable gate array) omogoča fleksibilnost pri programiranju, ASIC (angl. application-specific integrated circuit) pa pride v poštev pri količinski proizvodnji in daje boljšo učinkovitost. Omeniti velja tudi moderna FPGA integrirana vezja, ki pogosto vsebujejo več procesorjev in enot za procesiranje digitalnih signalov, ter kot taki konkurirajo rešitvam DSP. Uporabni so v aplikacijah, ki zahtevajo veliko vektorskega procesiranja, visoko energijsko učinkovitost, ter možnost kasnejše nadgradnje programske opreme.

Glede na vrsto aplikacije naloge krmilnika segajo od preprostega branja senzorskih podatkov in pošiljanja le-teh drugim vozliščem, do zapletenih izračunov z uporabo algoritmov umetne inteligence, ki omogočajo inteligentno odločanje o različnih nalogah.

2.3.2 Napajalni vir

Vozlišče brez napajanja ne more opravljati svojih nalog, zato je potrebno posebno pozornost posvetiti različnim napajalnim virom. Za napajanje vozlišč potrebujemo električno energijo, zato se osredotočimo na vire, ki so zmožni sami po sebi ali skupaj z dodatnimi komponentami dovajati električno energijo. V grobem jih razdelimo glede na količino energije, ki jo dovajajo: obilni, pogojno-obilni in redki. Med obilne vire spadajo viri, kjer ni bojazni, da bi ob normalnem delovanju zmanjkalo energije za napajanje vozlišča (npr. električno omrežje, avtomobilski akumulator). Pogojno-obilni viri nudijo dovolj velike količine energije ob določenih izpolnjenih pogojih (npr. sončne celice podnevi ob sončnem vremenu). Redke vire energije pa zaznamuje predvsem omejena količina energije, ki jo dovajajo, na krajši rok (npr. vibracijski generator električne energije) ali na daljši rok (npr. baterija).

Najpogostejše uporabljeni energijsko redek vir za napajanje vozlišč je elektrokemična baterija, ki jo delimo na primarno in sekundarno vrsto [10]. Primarne baterije ne omogočajo polnjenja in imajo v splošnem višje gostote energije kot sekundarne baterije, ki omogočajo obraten proces shranjevanja energije. Pri zasnovi BSO nas zanimajo predvsem kapaciteta baterije, zmožnost dovajanja želenih tokov, lastnosti samodejnega praznjenja ter v primeru sekundarnih baterij učinkovitost polnjenja. Pomembno je tudi dobro poznavanje pojava relaksacije, ki povzroča navidezno polnjenje baterije v odsotnosti porabnika, zaradi učinka kemične difuzije v celicah [9, str. 33].

2.3.3 Senzorji in aktuatorji

Senzorje v splošnem sestavlja neke vrste energijski pretvornik in spremljajoči merilni, ojačevalni in procesni elementi. Krmilnik je s senzorji povezan posredno ali neposredno in skrbi za odjemanje vrednosti ter pogosto tudi za nadaljnje procesiranje pri izračunu izmerjene fizikalne veličine. Senzorjem je povečini potrebno zagotoviti električno napajanje. Aktuatorje uvrstimo skupaj s senzorji, ker si delijo nekatere skupne

lastnosti. Oboje sestavlja energijski pretvornik in spremljajoči elementi, zagotoviti jim je potrebno električno napajanje in povezani so s krmilnikom, ki jih upravlja.

Senzorji in aktuatorji zaokrožajo različne stopnje kompleksnosti. Lahko jih sestavlja le pretvornik, ki mu je potrebno zagotoviti še ojačenje in procesiranje, ali pa so samostojen senzor ali aktuator s svojim lastnim procesnim elementom in pomnilnikom, ki s krmilnikom komunicira po komunikacijskem protokolu. Večja kompleksnost običajno prinaša večjo porabo energije in lažje programiranje krmilnika.

2.3.4 Komunikacijski podsistem

Za komunikacijo se najpogosteje uporablja radijsko-frekvenčna komunikacija, lahko pa se uporabijo tudi drugi frekvenčni razponi elektromagnetnega spektra, ali celo komunikacija preko drugih medijev. Vsekakor komunikacijska naprava potrebuje vir napajanja ter povezavo s krmilnikom, od katerega prejema ukaze. Pogosto ima vgrajen svoj pomnilnik, v katerega shranjuje nastavitve komunikacije in komunikacijske podatke.

Komunikacijski podsistem običajno vsebuje tako sprejemnik kot oddajnik, vendar to ni zahtevano za vse vrste aplikacij, saj si lahko zamislimo senzorsko vozlišče, ki le pošilja podatke senzorjev in ne prejema nikakršnih povratnih informacij. Po drugi strani lahko izdelamo senzorsko omrežje z vozlišči, ki le prejemajo ukaze in nimajo možnosti pošiljanja podatkov. Pri gradnji takšnega omrežja upoštevamo lastno nezanesljivost brezžičnih medijev in naredimo sistem odporen na izgubo informacij.

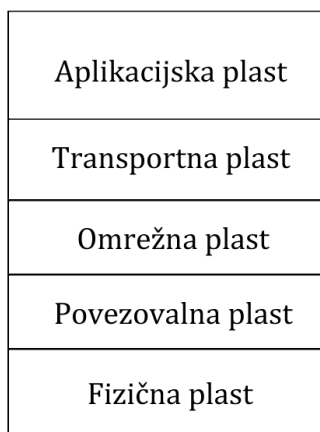
2.3.5 Pomnilnik

Vsako vozlišče s procesorskim krmilnikom uporablja vsaj RAM pomnilnik, ki je potreben za delovanje in je že sestavni del mikrokrmilnika. Ostale vrste pomnilnikov se uporabljajo glede na zahteve aplikacije. Programska koda je shranjena v EEPROM ali flash pomnilniku, včasih pa tudi v ROM pomnilniku. Flash pomnilnik omogoča shranjevanje podatkov, ki jih ne želimo izgubiti ob prekinitvi napajanja, in služi kot začasni pomnilnik, kadar ima RAM premajhno kapaciteto ali med izklopom RAM pomnilnika zaradi varčevanja z energijo [9, str. 21].

2.4 Komunikacija med vozlišči

Komunikacija med vozlišči ni omejena na radijsko komunikacijo ampak obsega tudi druge načine komunikacije, kot so optična komunikacija, infrardeča komunikacija, ultrazvočna komunikacija in komunikacija preko magnetne indukcije. Vsekakor za večino aplikacij najbolj ustreza komunikacija v radijskem delu elektromagnetnega spektra, na kar se osredotočimo.

Slika 2.3 prikazuje tipično zgradbo protokolnega sklada BSO, ki se od ISO/OSI referenčnega modela razlikuje v odsotnosti predstavitvene in sejne plasti. Na višjih plasteh sklada so protokoli običajno razviti glede na zahteve dane aplikacije, medtem ko na nižjih plasteh že obstajajo protokoli namenjeni BSO-jem.



Slika 2.3: Tipičen protokolni sklad BSO.

2.4.1 Radijska komunikacija

Na najnižji fizični plasti komunikacije se ukvarjamo predvsem z modulacijo in demodulacijo digitalnih podatkov, t. j. njihovo pretvorbo v in iz analognega signala. Za razliko od žične komunikacije, brezžični kanali niso vodeni, kar pomeni, da propagacija signala ni omejena na točno določene lokacije [9, str. 86]. Pomembna odločitev za radijsko komunikacijo je izbira frekvenčnega pasu, ki lahko sega od zelo nizkih frekvenc (angl. very low frequency – VLF) 3 kHz do zelo visokih (angl. very high frequency – VHF) 300 GHz, ki že prehaja v infrardeči del elektromagnetnega spektra. Pogosto uporabljen frekvenčni pas je nelicenčni ISM (angl. Industrial, Scientific, and Medical) pas, ki obsega več razponov frekvenc (v Evropi od 866 MHz do 868 MHz in od 2,4 GHz do 2,4835 GHz).

Modulacija je definirana kot proces spreminjanja ene ali več lastnosti visokofrekvenčnega nosilnega signala. Ločimo analogno in digitalno modulacijo, pri čemer je cilj analogne modulacije prenesti analogni signal preko analognega kanala na višje frekvence (npr. AM in FM modulacija). Za nas je zanimiva predvsem digitalna modulacija, kjer digitalno nastavljamo parametre analognega signala, pri čemer glede na tok bitov spreminjamo amplitudo (angl. amplitude shift keying – ASK), frekvenco (angl. frequency shift keying – FSK) ali fazo (angl. phase shift keying – PSK). Poznamo tudi zahtevnejše metode, kot npr. QAM (angl. quadrature amplitude modulation), kjer spreminjamo tako amplitudo kot fazo. Drugačen princip modulacije uporabljajo metode s porazdeljenim spektrom (angl. spread spectrum), ki signal omejene pasovne širine porazdelijo po veliko širšem pasovnem območju. To omogoča večjo varnost in manjšo podvrženost šumu. V to skupino metod spadajo: FHSS (angl. frequency-hopping spread spectrum), DSSS (angl. direct-sequence spread spectrum), THSS (angl. time-hopping spread spectrum), CSS (angl. chirp spread spectrum) in še mnoge variacije.

Omeniti velja tudi UWB (angl. ultra wide band) komunikacijo, ki se v senzorskih omrežjih zaenkrat ne pojavlja pogosto, predstavlja pa odstopanje od običajnega koncepta brezžične komunikacije. Namesto modulacije digitalnega signala na nosilno frekvenco, se digitalna sekvenca neposredno prenese na zelo širok frekvenčni pas, kot serija zelo kratkih pulzov [9, str. 27]. Cilj je zagotoviti čim bolj pravokoten signal, kar pomeni, da ti pulzi zasedajo frekvence od nekaj Hz do več GHz. Medtem, ko je UWB oddajnik dokaj preprosto izdelati, zahtevajo sprejemniki izvedbo natančne časovne sinhronizacije. Ta vrsta komunikacije omogoča izredno natančno meritev razdalj med sprejemnikom in oddajnikom.

2.4.2 Protokoli za dostop do medija

Protokoli za dostop do medija ali MAC protokoli (angl. medium access control), ki spadajo v povezovalno plast sklada, se ukvarjajo s problematiko učinkovitega deljenja prenosnega medija med vozlišči. Protokoli se močno razlikujejo glede na to ali komunikacija poteka po vodenem ali nevodenem mediju. Ker se osredotočamo na brezžično radijsko komunikacijo, bomo omenili tiste MAC protokole, ki so primerni za komunikacijo po nevodenem mediju. Brezžična komunikacija prinaša dodatne težave pri zasnovi MAC protokolov, kot je npr. problem skrite postaje (angl. hidden-terminal

problem), kjer dve postaji, ki medsebojno nista v dosegu, oddajata hkrati in s tem povzročita trk pri prejemniku.

V grobem razdelimo MAC protokole na FA (angl. fixed assignment), DA (angl. demand assignment) in RA (angl. random access). Pri FA protokolih so dani komunikacijski viri dolgoročno fiksno porazdeljeni med vozlišča, tako da načeloma ne more priti do trkov. Tipični protokoli te vrste so TDMA (angl. time division multiple access), FDMA (angl. frequency division multiple access), CDMA (angl. code division multiple access) in SDMA (angl. space division multiple access), ki razdeljujejo komunikacijo glede na čas, frekvenco in prostor [11, str. 325-326].

DA protokoli razdelijo vire glede na kratkotrajne potrebe posameznih vozlišč, običajno le za trajanje enega podatkovnega prenosa. Razdelimo jih na centralizirane (angl. centralized) in porazdeljene (angl. distributed), glede na prisotnost ali odsotnost centralne avtoritete. V centraliziranih protokolih vozlišča sporočijo svoje zahteve centralnemu vozlišču, ki nato vozliščem dodeli vire. Tak sistem vsebuje enojno točko odpovedi (angl. single point of failure), ki je v splošnem nezaželena. Druga rešitev je uporaba žetona (angl. token), ki si ga vozlišča podajajo med seboj. Le trenutni lastnik žetona ima dovoljenje oddajati na določenem kanalu.

Tretja vrsta MAC protokolov so RA protokoli, kjer so vozlišča nekoordinirana in komunicirajo popolnoma porazdeljeno. V to skupino spadajo CSMA (angl. carrier sense multiple access) protokoli, kjer vozlišče pred oddajanjem na mediju posluša, ali prenos že poteka in oddaja le v primeru nezasedenosti. V nasprotnem primeru vozlišče odloži prenos za izbran čas, ki je običajno naključen. Ti protokoli uporabljajo tudi različne prijeme, ki razrešijo problem skrite postaje (npr. RTS/CTS protokol).

2.4.3 Višje komunikacijske plasti

Protokoli omrežne plasti sklada se ukvarjajo z učinkovitim usmerjanjem, pri čemer z usmerjanjem glede na vire (angl. resource-aware routing) običajno želimo čim bolj zmanjšati porabo električne energije. Pri podatkovno-centričnem usmerjanju (angl. data-centric routing) protokol daje večji poudarek podatkom kot posameznim meritvam na vozliščih. S tem se na vozliščih izvaja agregacija podatkov in se posledično zmanjša poraba energije za komunikacijo.

Nadzor topologije (angl. topology control) je področje, ki se ukvarja s čim učinkovitejšo rabo virov v BSO. Običajno gre za namensko omejevanje komunikacije med vozlišči, z manjšanjem oddajne moči, vpeljavo hierarhije, opravljanjem koordinacije na vozliščih ter celo z izklapljanjem vozlišč [9, str. 265-288]. Gre za aktivno raziskovalno področje, katerega cilj je zasnova algoritmov, ki za dan problem čim bolj učinkovito uporabljajo razpoložljive energijske vire.

Protokoli v transportni plasti se v BSO redkeje uporabljajo, uporabni pa so v primeru, da želimo dostopati do sistema preko interneta ali preko drugih zunanjih omrežij [12, str. 106-407]. Tudi aplikacijski protokoli za senzorska omrežja se ne uporabljajo pogosto. Ena vrsta aplikacijskega protokola je protokol za upravljanje s senzorji (angl. sensor management protocol), ki uporabniku preko interneta omogoča spreminjanje nastavitev vozlišč, postavljanje pravil agregacije podatkov, konfiguracijo sistema, ipd.

2.5 Energijski pretvorniki

Pretvornik (angl. transducer) je naprava, ki pretvarja eno obliko energije v drugo [13, str. 9-12]. V sklopu BSO s senzorji običajno pretvarjamo iz neelektrične v električno energijo, z aktuatorji pa iz električne energije v druge oblike energije. Kombinirani pretvorniki (npr. ultrazvočni merilnik) obenem delujejo kot senzorji in kot aktuatorji, saj odjemajo vrednosti iz okolja in nanj tudi aktivno fizikalno vplivajo. V nadaljevanju so predstavljeni principi delovanja različnih pretvornikov, ki se pogosteje uporabljajo v BSO. Temu sledi seznam pogosteje uporabljenih senzorjev in aktuatorjev ter podrobnejša razlaga delovanja senzorja temperature in senzorja vlage.

2.5.1 Fizikalni principi delovanja

V nadaljevanju so predstavljeni nekateri fizikalni principi, ki omogočajo meritev raznovrstnih fizikalnih veličin. Piezoupornostni pojav (angl. piezoresistive effect) omogoča meritev natega nekaterih kovin in polprevodnikov preko upornosti. Piezoelektrični pojav (angl. piezoelectric effect) pretvarja silo na piezoelektrični material v potencialno razliko. Pojav lahko uporabimo tudi za aktuator, saj sprememba v napetosti ustvarja silo in posledično spremembo v debelini materiala. Piroelektrični pojav (angl. pyroelectric effect) ustvarja električni naboj glede na toplotni tok. Za meritev odmika se

uporabljajo tako kapacitivni kot induktivni senzorji. Pojav tuneliranja uspešno koristimo za meritev majhnih nanometerskih premikov. Hallov pojav (angl. Hall effect) proizvaja električno napetost na prevodniku prečno glede na smer električnega toku zaradi zunanjega magnetnega polja. Uporablja se za meritev magnetnega polja. Podoben je magnetoupornostni pojav (angl. magnetoresistive effect), ki povzroči, da se upornost materiala spremeni zaradi zunanjega magnetnega polja.

Termomehanska (angl. thermo-mechanical) pretvorba povzroča raztezanje in krčenje materiala ter posledično meritev temperature. Termoupornostni pojav (angl. thermoresistive effect) spreminja upornost materialu glede na temperaturo. Termospoji (angl. thermocouples) temeljijo na termoelektričnem Seebeck pojavu, ki povzroča potencialno razliko med dvema različnima spojenima materialoma z različno temperaturo. Resonančni temperaturni senzorji izkoriščajo pojav, pri katerem se s spremembo temperature spremeni tudi resonančna frekvenca enojnega kristala silicijevega dioksida (SiO_2). Uporablja se predvsem za zaznavo zelo majhnih temperaturnih sprememb.

Optični pretvorniki pretvarjajo svetlobo v različne oblike energije. Pojav fotoelektričnosti (angl. photoelectric effect) povzroča emisijo elektronov ob izpostavitvi materiala svetlobi. Fotokonduktivnosti (angl. photoconductivity) poveča električno prevodnost materiala, fotovoltaičnost (angl. photovoltaic effect) pa na njem ustvarja električni potencial.

Kemični in biološki pretvorniki uporabljajo različne pristope za zaznavanje prisotnosti ali odsotnosti snovi vseh vrst. Uporovni kemični senzorji (angl. chemiresistors) imajo premazane elektrode, ki se jim spremeni upornost pri izpostavitvi določenim kemičnim agentom. Elektrokemični pretvorniki merijo tokove, ki jih povzročata kemična oksidacija in redukcija na površini elektrode. Biosenzorji imajo biokemično aktivno tanko plast, ki pretvori spremembe lastnosti v merljive električne ali optične signale.

Akustični senzorji uporabljajo zvok kot medij zaznavanja, pri čemer Dopplerjev pojav omogoča meritev hitrosti. Ultrazvok daje informacije o različnih lastnosti mehanske opreme (npr. vibracije). Sonar uporablja čas potovanja zvoku za meritev razdalje in je še posebej uporaben pod vodno gladino.

Elektromagnetni spekter se poleg komunikacije uporablja tudi za merjenje različnih veličin. Dopplerjev pojav omogoča meritev hitrosti. Infrardeča svetloba se lahko uporabi za nočni vid (angl. night vision) in zaznavanje toplote. Meri se lahko tudi razdalja preko časa, ki ga signal potrebuje za potovanje. Med drugim tudi GPS sistem uporablja radijske frekvence za lokalizacijo.

2.5.2 Senzorji in aktuatorji

Tabela 2.1 vsebuje seznam pogostejših fizikalnih veličin, ki jih odjemajo senzorji, ter principov odjemanja, ki jih uporabljajo. Senzorje v splošnem delimo na neposredne in kompleksne [14, str. 56], pri čemer neposredni senzorji pretvarjajo neelektrične dražljaje v električni signal brez vmesnih pretvorb. Veliko dražljajev pa potrebuje več korakov pretvorbe, za kar poskrbijo kompleksni senzorji.

Fizikalna veličina	Fizikalni princip ali tehnologija odjemanja
Pritisk	Piezoupornost, kapacitivnost
Temperatura	Termistor, termomehanika, termospoj
Vlažnost	Upornost, kapacitivnost
Pretok	Sprememba pritiska, termistor
Položaj	EM, GPS, kontaktni senzor
Hitrost	Dopplerjev pojav, Hallov pojav, optoelektričnost
Kotna hitrost	Optični enkoder
Pospešek	Piezoupornost, piezoelektričnost, optična vlakna
Nateg	Piezoupornost
Sila	Piezoelektričnost, piezoupornost
Navor	Piezoupornost, optoelektričnost
Vibracije	Piezoupornost, piezoelektričnost, optična vlakna, zvok, UZ
Kontakt	Kontaktno stikalo, kapacitivno
Bližina	Hallov pojav, kapacitivnost, magnetno, akustično, RF
Razdalja	EM, magnetno, tuneliranje
Gibanje	EM, IR, akustično

Tabela 2.1: Seznam nekaterih fizikalnih veličin in principov odjemanja za senzorje.

V sklopu BSO pojem aktuator zajema vse naprave, ki na podlagi električnega signala fizikalno vplivajo na okolje in so, tako kot senzorji, le pretvorniki energije. Fizikalni principi, ki se izkoriščajo za delovanje aktuatorjev, so pogosto manj eksotični, kot tisti za senzorje. Za mehansko gibanje (npr. motor) in za elektroakustiko (npr. zvočnik) se

običajno izkoriščajo principi elektromagnetike. Med elektrooptične aktuatorje uvrstimo različne svetlobne vire, ki izkoriščajo fotoelektrične pojave (npr. LED, laserska dioda). Za termoelektrično hlajenje se izkorišča Peltierjev pojav, ki ustvarja toplotni tok med spojem dveh različnih materialov.

Zaradi pojava opazovalca (angl. observer effect) vsak senzor, ki iz okolja odjema neko fizikalno veličino, na okolje tudi nehoti vpliva. Kombinirani pretvorniki za razliko od senzorjev na okolje aktivno vplivajo z namenom, da se izmeri povratna informacija njihovega učinkovanja na okolje. Običajno ne ločujemo kombiniranih pretvornikov od senzorjev, saj je končni namen kombiniranega pretvornika enak kot pri senzorju, t. j. izmeriti neko fizikalno veličino. Zaradi njihove narave delovanja imajo pogosto večjo porabo energije kot senzorji.

2.5.3 Merjenje temperature s p-n spojem

Temperatura je v termodinamiki definirana kot merilo naključnega gibanja in vibriranja sestavnih delcev materije. Termodinamična temperatura se meri v Kelvinih (K), nas pa običajno zanima temperatura v stopinjah Celzija (°C).

$$T_{\text{°C}} = T_{\text{K}} - 273,16 \quad (2.1)$$

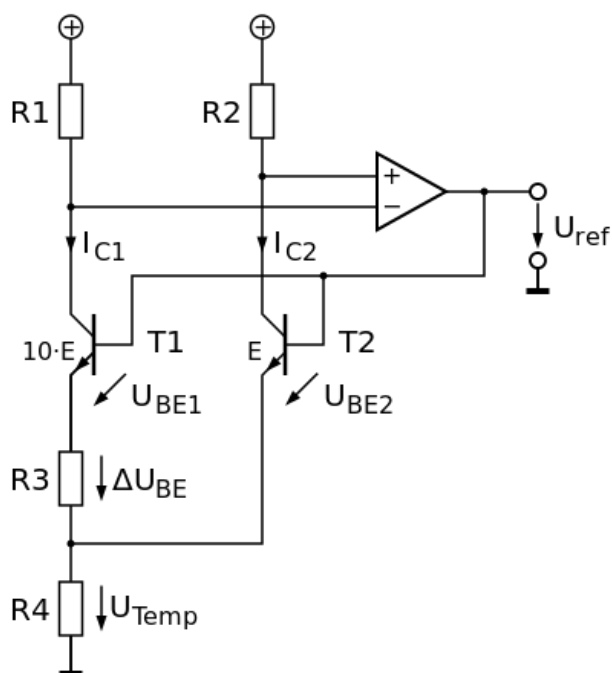
Meritev temperature se pogosto izvaja s silicijevim bandgap temperaturnim senzorjem. Senzor izkorišča lastnost silicijeve diode ali bipolarnega tranzistorja, da je prevodna napetost p-n spoja temperaturno odvisna. Z uporabo Brokaw bandgap referenčnega vezja, prikazanega na sliki 2.4, primerjamo napetosti pri dveh različnih tokovih I_{C1} in I_{C2} [15]. Sledeča enačba omogoča izračun temperature okolice:

$$\Delta V = \frac{KT}{q} \ln \left(\frac{I_{C1}}{I_{C2}} \right) \quad (2.2)$$

Kjer je

- ΔV – napetostna razlika med tranzistorjema v Voltih,
- K – Boltzmannova konstanta ($1,38 \cdot 10^{-23} \frac{\text{J}}{\text{K}}$),
- T – absolutna temperatura v Kelvinih,
- q – naboj elektrona ($-1,6 \cdot 10^{-19} \text{ C}$) in
- I_{C1}, I_{C2} – tokova, ki tečeta skozi tranzistorja v Amperih.

Iz enačbe je razvidno, da je napetostna razlika proporcionalna absolutni temperaturi T .



Slika 2.4: Brokaw bandgap referenčno vezje.

2.5.4 Merjenje vlažnosti s kondenzatorjem

Za meritev vlažnosti se uporabljata tako upornost kot kapacitivnost. Vlažnost je vsebnost vodne pare v zraku ali drugem plinu. Absolutna vlažnost je definirana kot masa vodne pare m_v na enoto prostornine mešanice zraka in vodne pare V .

$$a = \frac{m_v}{V} \quad \left[\frac{\text{kg}}{\text{m}^3} \right] \quad (2.3)$$

Relativna vlažnost, katero senzorji običajno merijo, je razmerje delnega pritiska vodne pare v mešanici zraka in vodne pare e_v ter nasičenega parnega tlaku vode e_v^* pri dani temperaturi.

$$r = \frac{e_v}{e_v^*} \cdot 100 \% \quad [\% \text{ RV}] \quad (2.4)$$

Enota relativne vlažnosti je % RV, pri čemer 0 % RV pomeni odsotnost vodne pare v zraku, 100 % RV pa največjo možno količino vodne pare v zraku pri dani temperaturi.

Kapacitivnemu senzorju vlažnosti se spreminja merjena kapacitivnost glede na relativno vlažnost okoliškega zraka. Kapacitivnost kondenzatorja je odvisna od relativne dielektričnosti, ki za kondenzator z zračnim dielektrikom znaša [14, str. 396]:

$$\varepsilon_r = 1 + \frac{1,58}{T} \cdot \left(P + 48 \frac{P_v}{T} H \right) \cdot 10^{-6} \quad (2.5)$$

Kjer je:

- T – temperatura zraka v Kelvinih,
- P – tlak zraka z vodno paro v Pascalih,
- P_v – tlak nasičene vodne pare pri temperaturi T v Pascalih in
- H – relativna vlažnost v % RV.

Iz enačbe je razvidno, da je dielektričnost in posledično kapacitivnost zračnega kondenzatorja premo sorazmerna z relativno vlažnostjo zraka. Namesto zraka se običajno uporabljajo drugi izolatorji (npr. steklo, keramika, silicij), ki hitro vsrkajo in izpustijo vlago glede na vlažnost okoliškega zraka. Kapacitivnost je običajno približno proporcionalna relativni vlažnosti:

$$C \approx C_0(1 + \alpha_h H) \quad (2.6)$$

Kjer je

- C_0 – kapacitivnost kondenzatorja pri $H = 0$ % RV in
- α_h – konstanta kondenzatorja.

3. POGLAVJE

Zasnova brezžičnih senzorskih omrežij

A good scientist is a person with original ideas. A good engineer is a person who makes a design that works with as few original ideas as possible.

- Freeman Dyson

Cilj tega poglavja je raziskati nekatera področja, ki omogočajo učinkovitejše odločanje pri sprejemanju inženirskih odločitev oz. kompromisov. Ugotovimo namreč, da je na področju BSO sprejeta odločitev skoraj vedno tudi kompromis, kadar pa temu ni tako, se pogosto odločamo med večimi podobnimi možnostmi, ki so vse več ali manj enako dobre, ali pa imamo le eno dobro možnost.

Zasnova BSO je iterativen proces, katerega cilj je zasnovati sistem z lastnostmi, ki pri danih omejitvah čim bolj ustrezajo zahtevam. Pogosto so zahteve podane preveč ohlapno, želimo namreč, da so te natančno podane in na zasnovanem sistemu merljive. Zato so v prvem podpoglavju predstavljene različne zahteve, omejitve in lastnosti BSO, ki jih poimenujemo faktorji zasnove.

3.1 Faktorji zasnove

Za vsakega izmed faktorjev se lahko pri zasnovi sistema pojavi določena zahteva, ki jo moramo doseči, hkrati pa nekatere faktorje smatramo kot spremenljivke, ki jih želimo optimizirati. Od posamezne aplikacije je odvisno, ali je posamezen faktor

- vnaprej fiksno določen,
- ima predpisane vrednosti, ki jih je potrebno doseči, ali
- nima nobenih posebnih pogojev.

Faktorje zasnove razdelimo glede na upravljanje, zgradbo in delovanje sistema.

3.1.1 Upravljanje s sistemom

Prvi faktor zasnove je **namestitev** (angl. deployment) sistema in vozlišč. Načini namestitve segajo od fiksne namestitve do izpusta iz letala, kjer se vozlišča naključno razporedijo po večjem geografskem področju. Namestitev se izvede enkrat, večkrat ali pa se izvaja neprenehoma, kadar se omrežju dodajajo nova vozlišča, npr. ob izpadih starih. Zaradi enostavnejše namestitve se običajno pričakuje, da se vozlišča samodejno povežejo in organizirajo. Zahtevana **enostavnost namestitve** je odvisna predvsem od tega, kdo bo namestitev izvajal.

Omenimo tudi faktorja **vzdrževanja** in **uporabe** sistema, ki sta za končnega uporabnika zelo pomembna. Lastnosti teh dveh faktorjev so močno odvisne od aplikacije in ostalih dejavnikov. Tudi tukaj se pričakujeta čim večja enostavnost in čim nižji stroški.

3.1.2 Zgradba sistema

Faktor oblike (angl. form factor) vozlišč BSO je običajno omejen na določeno velikost. Posledično se uporabijo manjše elektronske komponente in energijski viri, kar včasih predstavlja težave pri doseganju ostalih ciljev (npr. življenjska doba vozlišča).

Eden izmed najpomembnejših faktorjev je **cena**, kjer končno stranko pogosto zanimajo **celotni stroški sistema** (angl. total cost of ownership). Pri večjem številu vozlišč se običajno trudimo zmanjšati **ceno vozlišča**, saj to predstavlja glavni del stroškov.

V primeru **mobilnih vozlišč** in komunikacije z večimi skoki se mora omrežje pogosto reorganizirati za doseg najučinkovitejšega prenosa podatkov. Poseben primer je **mobilnost dogodkov**, kjer imamo opravka z naborom mobilnih dogodkov oz. objektov, ki jim sledimo. Ko se vir dogodka giblje skozi prostor, ga spremlja področje aktivnosti senzorjev, pri čemer so senzorji izven tega področja neaktivni [9, str. 62-63].

Heterogenost vozlišč je faktor, ki pove ali sta strojna in programska oprema vozlišč enaki, ali se razlikujeta med posameznimi vozlišči. Običajno težimo k temu, da so vozlišča homogena, saj to pomeni manjše stroške razvoja, testiranja in proizvodnje. Nekateri aplikacije zaradi specifičnih zahtev potrebujejo različna vozlišča.

Velikost omrežja v splošnem obsega najmanj dve vozlišči in navzgor ni omejena. Od števila vozlišč je odvisnih mnogo faktorjev, predvsem pa tisti, ki se nanašajo na **omrežno topologijo**. Pri komunikaciji z večimi skoki na primer pogosto želimo, da poljubni dve vozlišči komunicirata z določenim največjim številom skokov.

Omenimo tudi faktor **fleksibilnosti** sistema, ki ga običajno interpretiramo kot prilagodljivost arhitekture BSO na različne aplikacije.

3.1.3 Delovanje sistema

Življenjska doba je eden izmed najpomembnejših faktorjev in je običajno vnaprej predpisan. Življenjsko dobo sistema definiramo kot čas do prvega izpada vozlišča, čas do 50 % izpadlih vozlišč, ali čas ko neka točka v prostoru ni več pokrita niti z enim senzorjem, ipd. [9, str. 8, 65]. Izbrana definicija je odvisna od aplikacije in jo dosegamo z ustrezno **energijsko učinkovitostjo** vozlišča. Energijsko učinkovitost merimo z veličinami kot sta poraba energije na pravilno prejet bit informacije in poraba energije za poročanje enega unikatnega dogodka.

Pokritost je povezana z velikostjo omrežja in gostoto vozlišč. Pokritost običajno pomeni prostorsko pokritost s senzorji in je tako odvisna od dosega senzorjev. Pomemben faktor je tudi doseg brezžične komunikacije. Pri odvečni pokritosti več vozlišč meri veličine na isti lokaciji. Običajno je zaželena gosta pokritost, saj je sistem robustnejši in ima daljšo življenjsko dobo zaradi odvečnih vozlišč.

Pod faktor **kakovost storitve** (angl. quality of service – QoS) uvrstimo tiste faktorje, ki se nanašajo na komunikacijo in poročanje o dogodkih. Prva zahteva pri komunikaciji je **zanesljivost** prejetih podatkov, ki so natančni z določeno mero verjetnosti. Pri poročanju o dogodkih oz. pošiljanju podatkov pričakujemo neko **zakasnitev**, ki jo običajno želimo zmanjšati. Pri nekaterih aplikacijah je še posebej pomembna **časovna natančnost** (angl. temporal accuracy).

Skalabilnost je lastnost BSO, da nemoteno in s podobno kakovostjo storitve deluje pri povečevanju števila vozlišč. **Prilagodljivost** se nanaša predvsem na spreminjajočo sestavo BSO. Pri zasnovi želimo doseči čim večjo **robustnost**, kar pomeni, da je sistem odporen na izpade posameznih vozlišč.

Varnost sistema zahteva uporabo kriptografskih algoritmov, kar pomeni večjo kompleksnost programske opreme vozlišč. Od aplikacije je odvisno, kakšna stopnja varnosti se zahteva, tudi glede na to, kakšne so posledice morebitnega vdora v sistem.

3.2 Osnovni principi zasnove

Zaradi obširnosti področja BSO težko podamo splošna pravila, ki veljajo za vse aplikacije, zato le naštejmo nekatere osnovne principe, ki pomagajo pri zasnovi in omogočajo doseganje ciljev. Najprej so podani nekateri splošni principi, podrobneje pa si pogledamo omrežno procesiranje in podatkovno centričnost.

3.2.1 Splošni principi

Kadar sta skalabilnost in robustnost pomembna faktorja zasnove, je običajno potrebno izdelati **porazdeljeno omrežje** vozlišč, ki je sposobno samoorganizacije. Tudi centralizirano omrežje ima prednosti, predvsem na področju energijske učinkovitosti.

Dodatno učinkovitost BSO dosežemo z **zniževanjem natančnosti**. Kadar omrežje ve, kakšna je zahtevana natančnost, uporabi le toliko virov (senzorji, procesna moč, energija), da je ta natančnost dosežena.

V omrežjih kjer imajo vozlišča **podatke o lokaciji**, se te informacije lahko izkoristi za večjo učinkovitost komunikacije. Komunikacijski protokol mora biti tudi primerno zgrajen, glede na pričakovane **vzorke aktivnosti** (angl. activity patterns) [9, str. 73]. Zgodi se lahko, da v omrežjih s pričakovano majhno pogostostjo dogodkov pride do večjega dogodka, kar ob neprimerni zasnovi povzroči prepolnost komunikacijskih kanalov.

Heterogena vozlišča imajo različne komponente, npr. različne kapacitete baterij. Tudi pri zbiranju energije iz okolice so nekatera vozlišča boljša od drugih. Primerno **izkoriščanje heterogenosti** dosežemo z dobrim poznavanjem zgradbe omrežja in dane aplikacije.

Pri komunikaciji se **protokolni sklad, temelječ na komponentah**, razlikuje od tipičnega protokolnega sklada, prikazanega na sliki 2.3, v tem, da se ne uporabljajo vedno vse komponente komunikacije [9, str. 74]. Protokolni sklad je zgrajen iz zbirke komponent, na podlagi katerih se gradijo protokoli in algoritmi. Takšen sklad je fleksibilen in s tem primeren za uporabo v različnih aplikacijah.

3.2.2 Omrežno procesiranje

V kompleksnejših omrežjih vozlišča ne delujejo le kot posredniki ampak aktivno sodelujejo pri odločanju o delovanju omrežja in pri procesiranju podatkov. **Omrežno procesiranje** omogoča večjo učinkovitost in dodatno porazdeljenost omrežja.

Najpreprostejši način omrežnega procesiranja je **agregacija** (angl. aggregation), kjer vozlišča združujejo večje količine podatkov v manjšo količino. To omogoča večjo učinkovitost omrežja, saj se porabi manj energije za brezžično komunikacijo. Učinkovitost agregacije merimo z veličinami: natančnost rezultata, polnost (odstotek meritev uporabljenih za izračun rezultata), zakasnitev in zmanjšanje količine poslanih podatkov [9, str. 341-342].

Z uporabo **kompresije** zmanjšamo količino poslanih podatkov. Za vozlišča so pogosto kompresijski algoritmi prezahtevni, zato se v nekaterih primerih uporabi **prostorska korelacija**, ki izkorišča dejstvo, da imajo senzorji, ki so si blizu, pogosto podobne meritve. Tako se za posamezno meritev pošilja le tista informacija, ki pove za koliko se vrednost meritve razlikuje od ostalih vozlišč oz. od povprečja.

Tudi kadar je potrebno izvajati zahtevnejše izračune, se včasih še vedno izplača računati na posameznih vozliščih, kljub omejeni procesni moči, če je s tem zmanjšano število poslanih podatkov. Primer takšnega **porazdeljenega procesiranja** je porazdeljen FFT (angl. fast Fourier transform) algoritem, ki se uporablja za procesiranje signalov [9, str 69].

3.2.3 Podatkovna centričnost

V podatkovno-centričnih BSO ni tako pomembna identiteta vozlišča kot podatki, ki jih vozlišča s pomočjo senzorjev proizvajajo. Zato se pri nekaterih aplikacijah izplača zgraditi komunikacijsko arhitekturo okoli podatkov namesto vozlišč. **Podatkovna centričnost** omogoča uporabo drugačnih omrežnih arhitektur, kot se uporabljajo v tradicionalnih omrežjih (npr. **zahtevaj/pošlji**, angl. request/send). Primer tega je komunikacija vrste **objavi/naroči** (angl. publish/subscribe), kjer vozlišča objavljajo podatke in se na njih naročajo [9, str. 72]. Po objavi podatkov so vsi naročniki na tovrstne podatke obveščeni. Če naročnik ne želi več prejemati podatkov, se od njih odjavi.

Drugačen pristop k podatkovni centričnosti je tak, ki smatra BSO kot podatkovno bazo. Za poizvedovanje podatkov se uporabljajo podobne poizvedbe kot v računalniških podatkovnih bazah, npr. z jezikom SQL. Pri takem pristopu se senzorji smatrajo kot navidezne tabele, na katerih se uporablja relacijske operatorje.

3.3 Energijska učinkovitost vozlišča

Poraba energije je eden najbolj perečih izzivov BSO. Za največjo učinkovitost je pri vseh elementih vozlišča potrebno čim bolj zmanjšati porabo energije. Posebno pozornost je potrebno nameniti radijskemu oddajniku in sprejemniku, saj je ta običajno največji porabnik.

3.3.1 Načini obratovanja mikrokrmilnika

Mikrokrmilnik ima običajno več obratovalnih načinov z različno porabo energije, med katera spadajo aktivni način (angl. active mode), način pripravljenosti (angl. idle mode), način spanja (angl. sleep mode) itd. [9, str. 36-38]. Menjava načina delovanja običajno zahteva čas in dodatno energijo, zato nas zanima, kdaj je še smiselno preklopiti mikrokrmilnik v način spanja pred naslednjim dogodkom. Preklop v način spanja je smiseln le kadar je dodatna potrošena energija za vrnitev iz načina spanja manjša od privarčevane energije, oz. kadar je čas do naslednjega dogodka dovolj velik. S sledečo neenačbo si pomagamo pri tej odločitvi:

$$(t_d - t_1) > \frac{1}{2} \left(\tau_{dol} + \frac{P_a + P_s}{P_a - P_s} \tau_{gor} \right) \quad (3.1)$$

Kjer je:

- t_1 – čas odločanja o preklopu v način spanja,
- t_d – čas naslednjega dogodka,
- τ_{dol} – čas preklopa v način spanja,
- τ_{gor} – čas preklopa v aktiven način,
- P_a – poraba moči v aktivnem načinu,
- P_s – poraba moči v načinu spanja.

Pri tem modeliramo porabo moči med preklopoma kot aritmetično sredino obeh porab moči. Če neenačba velja, potem se ob času t_1 priporoča preklop v način spanja.

3.3.2 Radijska komunikacija

Za izračun porabe energije radijskega oddajnika uporabimo model, povzet po Min in Chandrakasan [16, str. 2-3]. Potrebno energijo za pošiljanje enega paketa podatkov s predhodno izklopljenega radijskega oddajnika modeliramo z enačbo:

$$E_{\text{tx}}(N, R_C, P_{\text{amp}}) = P_{\text{start}}T_{\text{start}} + \frac{N}{R_C R} (P_{\text{txElec}} + P_{\text{amp}}) \quad (3.2)$$

kjer je:

- P_{start} – moč vklopa oddajnika,
- T_{start} – čas vklopa oddajnika,
- P_{txElec} – moč aktivnega oddajanja,
- P_{amp} – izgubna moč ojačevalnika,
- N – število bitov pred FEC (angl. forward error correction),
- R – bitna hitrost radijskega oddajnika in
- R_C – stopnja konvolucijskega kodiranja.

FEC je orodje za iskanje in odpravljanje napak pri prenosu podatkov po nezanesljivih medijih. Eden izmed možnih FEC je konvolucijska koda, ki vsak m -bitni niz zakodira v daljši n -biten niz, pri čemer je stopnja konvolucijskega kodiranja $R_C = m/n$.

Podobno izračunamo porabo energije prejema enega paketa paketov, pri uporabi Viterbi dekodiranja konvolucijske kode, z enačbo:

$$E_{\text{rx}}(N, R_C, E_{\text{decbit}}) = P_{\text{start}}T_{\text{start}} + P_{\text{rxElec}} \frac{N}{R_C R} + E_{\text{decbit}}N \quad (3.3)$$

Kjer je:

- P_{start} – moč vklopa sprejemnika,
- T_{start} – čas vklopa sprejemnika,
- P_{rxElec} – moč sprejemanja,
- N – število bitov pred FEC,
- R – bitna hitrost radijskega sprejemnika,
- R_C – stopnja uporabljenega konvolucijskega kodiranja in
- E_{decbit} – energija potrebna za dekodiranje enega bita informacije.

3.3.3 Procesiranje

Poraba električne energije zaradi procesiranja mikrokrmilnika je običajno veliko manjša kot poraba zaradi brezžične komunikacije. Zmanjšanje porabe energije se doseže z nižanjem napajalne napetosti in frekvence obratovanja mikrokrmilnika, kot je to razvidno iz enačbe, ki opisuje porabo električne energije zaradi procesiranja [12, str. 404]:

$$P_p = CV_{dd}^2 f + V_{dd} I_0 e^{V_{dd}/n'V_T} \quad (3.4)$$

Kjer je:

- C – skupna kapacitivnost preklapljanja v Faradih,
- V_{dd} – napajalna napetost v Voltih in
- f – frekvenca preklapljanja v Hertzih,
- I_0 – tok puščanja v Amperih in
- $n'V_T$ – pragovna napetost v Voltih.

Iz osnovnega člena enačbe 3.4 je razvidno, da se s padanjem napajalne napetosti kvadratno manjša poraba energije. Vendar se z manjšanjem pragovne napetosti večja poraba energije zaradi izgubnih tokov.

3.4 Zasnova brezžične komunikacije

Za dobro zasnovo BSO je ključno poznavanje načinov vrednotenja in računanja dosega ter kakovosti brezžične komunikacije. Osnovni principi so predstavljeni v nadaljevanju.

3.4.1 Vrednotenje z decibeli

Za vrednotenje moči se v radijski tehnologiji običajno uporablja enota decibel-milivat (dBm), ki predstavlja razmerje moči glede na 1 mW. Pri pretvorbi si pomagamo z enačbama:

$$x = 10 \log \frac{P}{1 \text{ mW}} \quad (3.5)$$

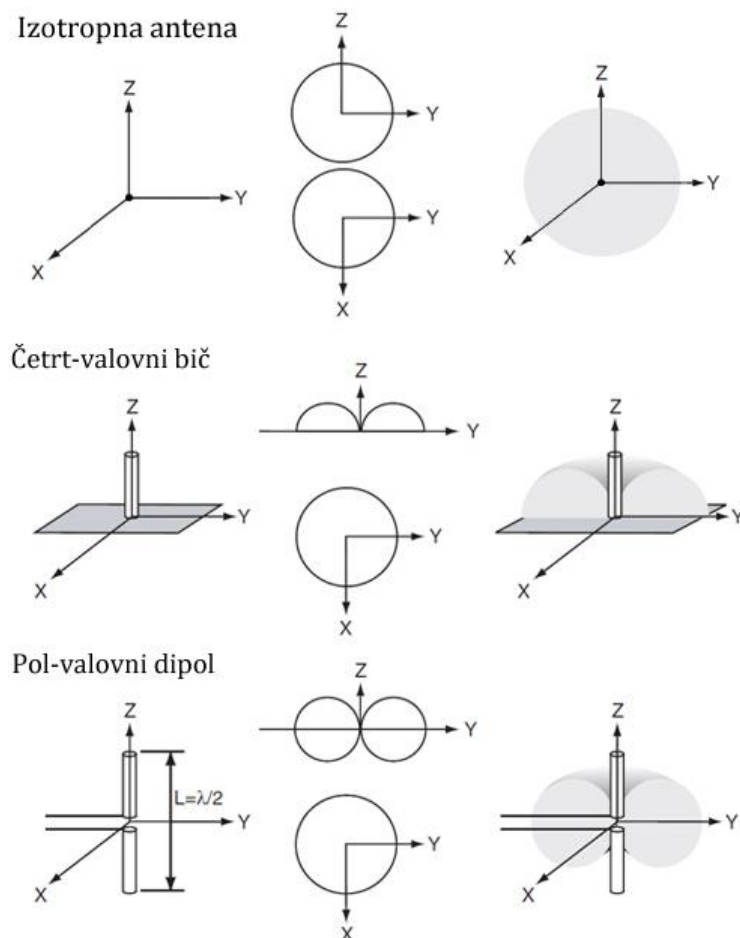
$$P = 1 \text{ mW} \cdot 10^{\frac{x}{10}} \quad (3.6)$$

Pri primerjavi signala z nosilnim signalom, ko na primer želimo ovrednotiti moč emisij na neželenih frekvencah (angl. spurious emissions), uporabimo enoto dBc (angl. decibels relative to carrier).

Slika 3.1 prikazuje tri preproste antene in njihove sevalne diagrame [17]. Prva izotropna antena je teoretična in se uporablja za primerjanje z drugimi antenami. Za vrednotenje ojačenja anten se uporabljajo sledeče enote:

- dBi – ojačenje antene glede na izotropno anteno (angl. isotropic antenna),
- dBd – ojačenje antene glede na pol-valovni dipol (angl. half wave dipole) in

- dBq – ojačenje antene glede na četrto-valovni bič (angl. quarter wave whip).



Slika 3.1: Nekatere antene in njihovi sevalni diagrami.

3.4.2 Doseg radijske komunikacije

Največji teoretično moč radijskega signala pri sprejemni anteni izračunamo z Friisovo enačbo (angl. Friis transmission equation) [18]:

$$P_s = P_o + G_s + G_o + 20 \log_{10} \left(\frac{\lambda}{4\pi R} \right) \quad (3.7)$$

Kjer je:

- P_s – moč signala pri sprejemni anteni v dBm,
- P_o – izhodna moč oddajalne antene v dBm,
- G_s – ojačenje sprejemne antene glede na izotropno anteno v dBi,
- G_o – ojačenje oddajalne antene glede na izotropno anteno v dBi,
- λ – valovna dolžina radijskega signala v metrih in
- R – razdalja med antenama v metrih.

S preurejeno enačbo izračunamo največji teoretični doseg, pri znani najmanjši možni moči signala pri sprejemni anteni.

$$R = \frac{\lambda}{4\pi} 10^{\frac{G_s + G_o + P_o - P_s}{20}} \quad (3.8)$$

Enačba ne upošteva motečih pojavov, kot so odbijanje, polarizacija in absorpcija, ki bistveno skrajšajo dejanski doseg komunikacije. Friisovi enačbi lahko dodamo koeficiente, ki upoštevajo te dejavnike, vendar jih je težko natančno določiti.

Doseg komunikacije računamo tudi s statističnimi modeli. Rayleigh bledenje (angl. Rayleigh fading) je model, ki se pogosto uporablja za ugotavljanje propagacije radijskega signala v urbanih okoljih, kjer anteni nista medsebojno vidni. Ricianovo bledenje (angl. Rician fading) se uporablja za izračun propagacije signala, ko je neposreden (angl. line-of-sight) signal močnejši od ostalih, vendar prihaja do delnega izničenja signala zaradi večih signalnih poti.

3.4.3 Kakovost brezžične komunikacije

Kakovost brezžične komunikacije lahko merimo s številom nepravilno prejetih podatkov. BER (angl. bit error rate) je razmerje napačno prejetih bitov glede na celotno število poslanih bitov. Verjetnost napačnega bita (angl. bit error probability) p_e je pričakovana vrednost (matematično upanje) BER. PER (angl. packet error rate) je razmerje napačno prejetih paketov podatkov glede na celotno število poslanih paketov. Paket je napačno prejet, če je vsaj en bit v paketu napačen. Verjetnost napačno prejetega paketa podatkov dolžine N bitov izračunamo z enačbo:

$$p_p = 1 - (1 - p_e)^N \quad (3.9)$$

Za majhne verjetnosti napačnih bitov uporabimo splošeno enačbo:

$$p_p \approx p_e N \quad (3.10)$$

Za vrednotenje kakovosti brezžične komunikacije se uporablja tudi razmerje signal-šum (angl. signal-to-noise ratio – SNR), ki se običajno podaja v decibelih [9, str.95]:

$$SNR_{dB} = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{sum}}} \quad (3.11)$$

Za analizo radijskega sprejemnika se uporabljajo BER krivulje (angl. BER curves), ki prikazujejo pričakovan BER pri dani moči vhodnega signala ali vrednosti SNR-ja.

4. POGLAVJE

Izdelava koncepta brezžičnega omrežja

Plans are worthless, but planning is everything.

- Dwight D. Eisenhower

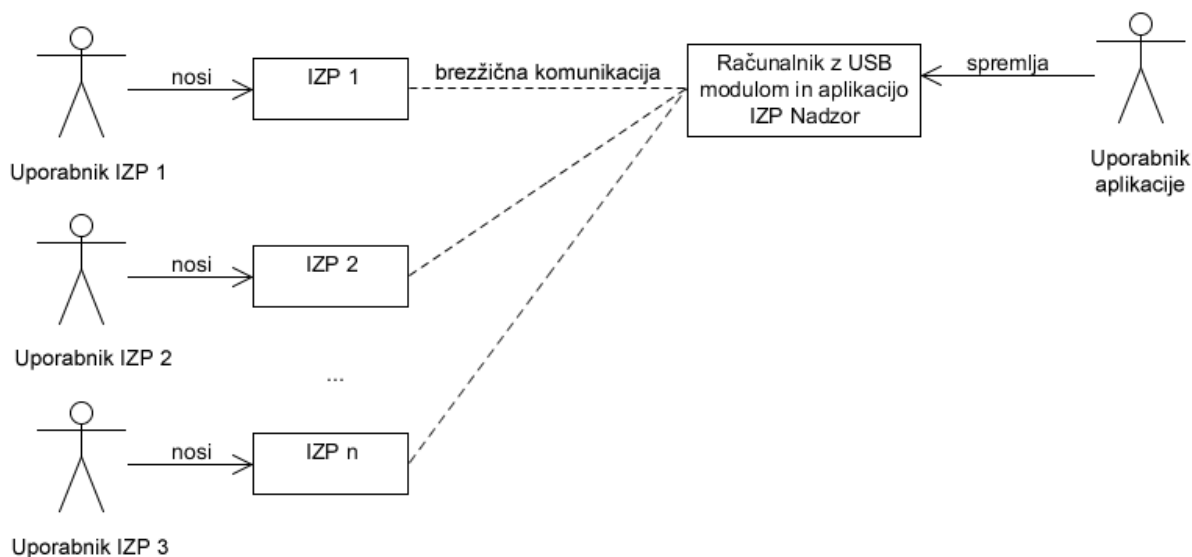
Glede na podane zahteve se lotimo izdelave koncepta BSO, ki nam bo omogočal lažje sprejemanje tehničnih odločitev in nam bo dal vpogled v delovanje, sestavo in uporabo sistema. Pri izgradnji koncepta si bomo pomagali z UML diagrami (angl. Unified Modeling Language), ki se običajno uporabljajo pri razvoju programske opreme.

4.1 Komunikacijska topologija

Konceptualno sistem razdelimo v dva dela. Prvi del sestavljajo IZP-ji in njihovi uporabniki, drugi del pa je osebni računalnik z aplikacijo in USB modulom ter njegov uporabnik. Topološko sestavljajo prvi del brezžična senzorska vozlišča, medtem ko se v drugem delu sistema nahaja centralno vozlišče. Iz slike 4.1 je razvidno, da poteka komunikacija med IZP-ji in centralnim vozliščem v zvezdasti strukturi in z enkratnimi skoki, saj komunikacija med vozlišči IZP ni predvidena.

Zaradi čim manjše porabe električne energije na IZP vozliščih ter tudi zaradi manjše kompleksnosti se odločimo za uporabo enosmerne komunikacije, kar pomeni, da vozlišča IZP delujejo le kot oddajniki in centralno vozlišče le kot sprejemnik. Zaradi te odločitve nimamo druge možnosti, kot da uporabimo zvezdasto topologijo z enkratnimi skoki. Posledica te odločitve je, da vozlišče IZP ne more vedeti ali je centralno vozlišče prejelo poslane podatke. Zaradi narave aplikacije to ne predstavlja težave. Vozlišča IZP namreč relativno pogosto pošiljajo izmerjene vrednosti (na vsakih 5 sekund), tako da v primeru izgubljenega paketa centralno vozlišče počaka na naslednjega. Vsak paket

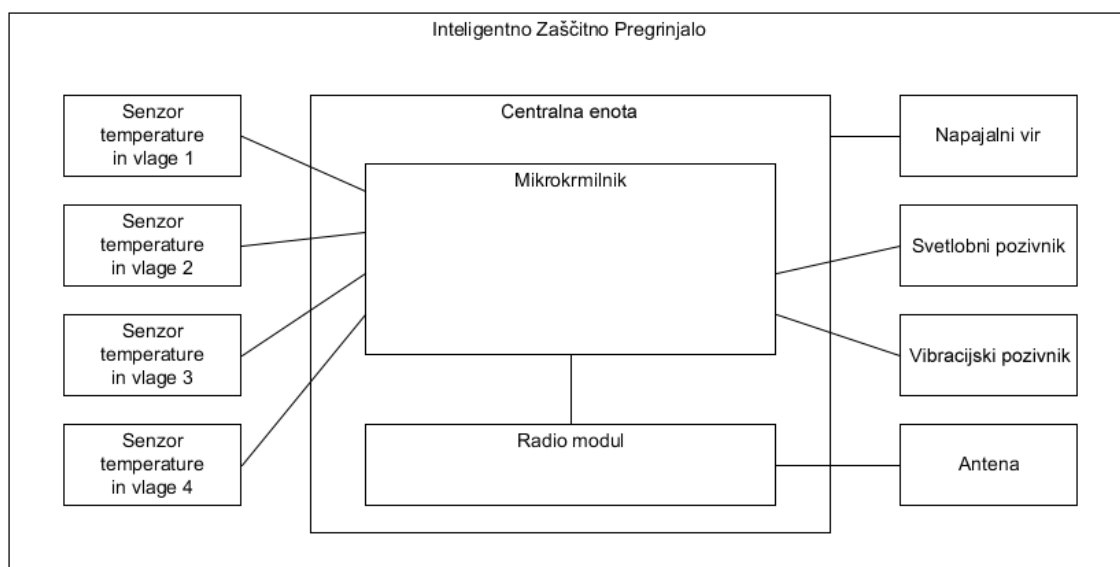
vsebuje izmerjene vrednosti temperature in vlage na vseh štirih senzorjih ter morebitne dodatne informacije o stanju IZP vozlišča (npr. stanje baterije).



Slika 4.1: Konceptualna zgradba sistema ter struktura komunikacije.

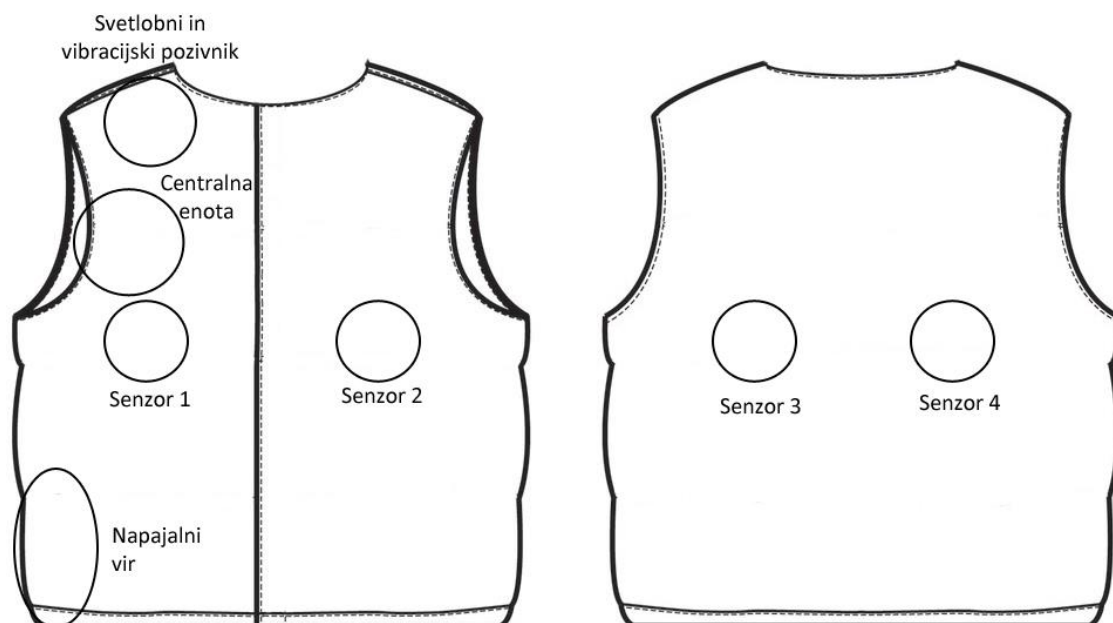
4.2 Inteligentno zaščitno pregrinjalo

V zahtevah je podano kakšni so sestavni elementi strojne opreme IZP-ja. Poleg svetlobnega pozivnika uporabimo še vibracijski pozivnik. Na sliki 4.2 so prikazane pomembnejše elektronske komponente IZP-ja ter žične povezave med njimi.



Slika 4.2: Elementi strojne opreme IZP-ja in povezave med njimi.

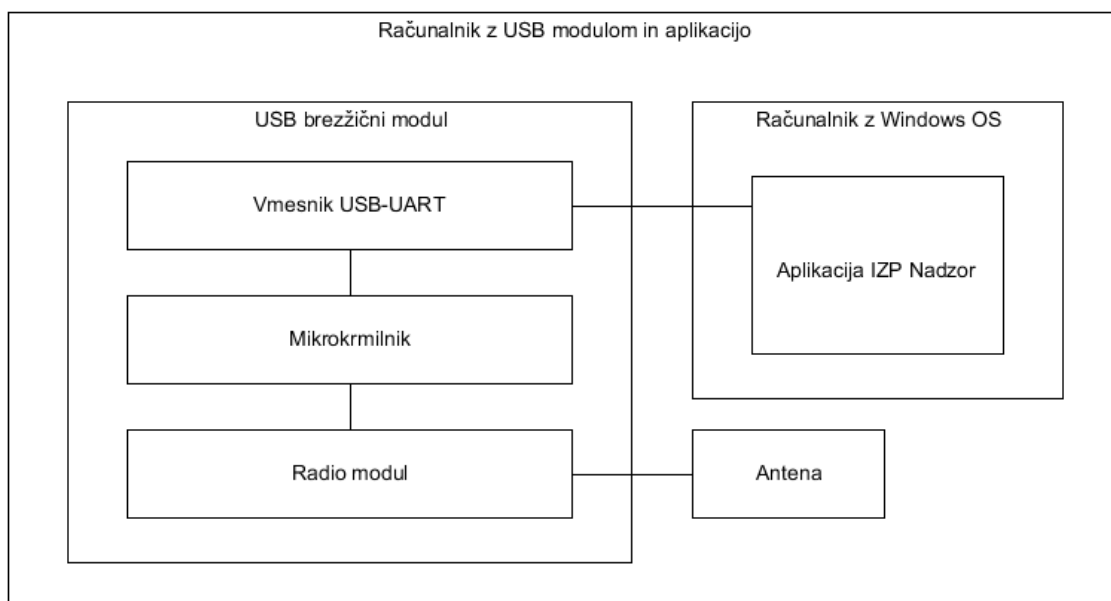
Slika 4.3 prikazuje lokacije elementov na IZP. Dva senzorja se nahajata spredaj in dva zadaj, napajalni vir bo vшит na bočni strani, vozlišče na prsnem delu in pozivnika na ramenskem delu pregrinjala.



Slika 4.3: Lokacije elementov na IZP.

4.3 Računalnik z USB modulom in aplikacijo

Na sliki 4.4 je prikazan drugi del sistema, ki ga sestavlja osebni računalnik z operacijskim sistemom Windows in aplikacijo, ki jo poimenujemo IZP Nadzor. Na USB priključek računalnika je priključen brezžični USB modul oz. centralno vozlišče BSO. USB modul sestavljajo mikrokontroler, radijski oddajnik/sprejemnik ter vmesnik med USB in UART komunikacijo, ki omogoča komunikacijo mikrokontrolerka z računalniško aplikacijo. USB modul ne potrebuje dodatnega vira napajanja, saj se napaja preko USB povezave z osebnim računalnikom.

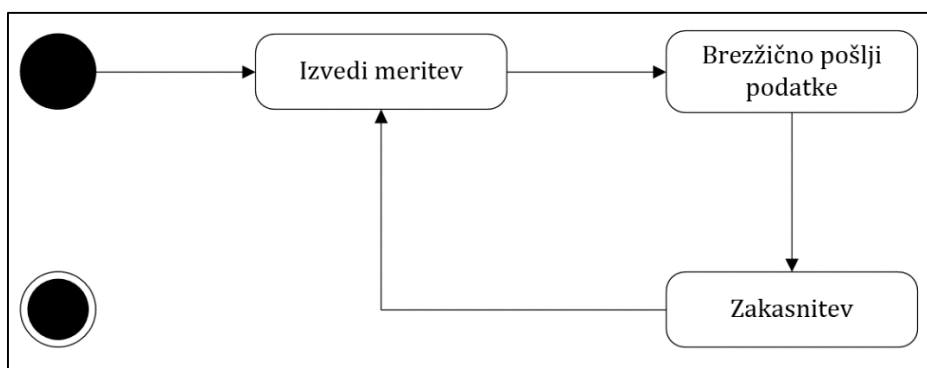


Slika 4.4: Glavni elementi osebne računalnika s centralnim vozliščem in aplikacijo.

4.4 Programska oprema sistema

4.4.1 Programska oprema vozlišča IZP

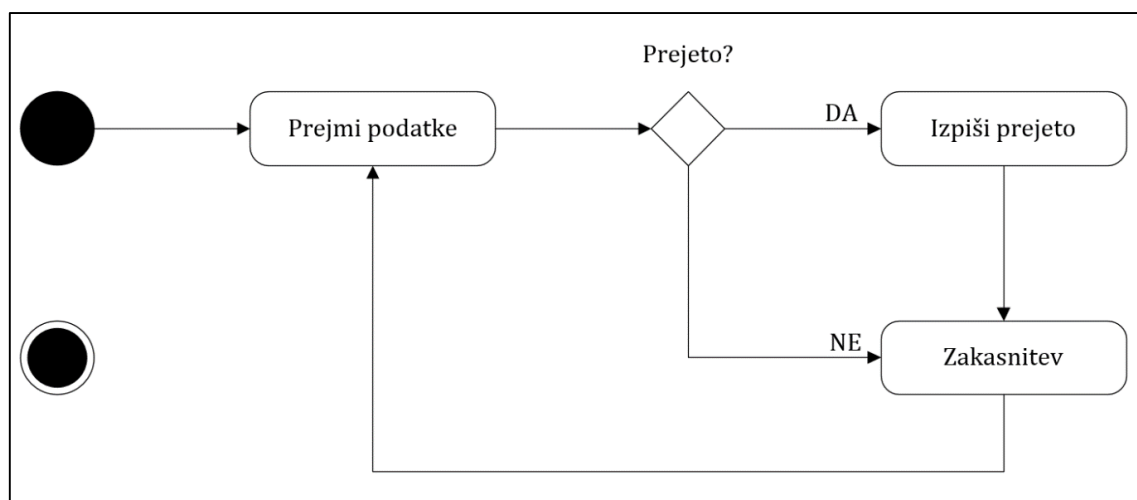
Za prikaz delovanja programske opreme vozlišča IZP izdelamo UML diagram aktivnosti (angl. activity diagram), ki je prikazan na sliki 4.5. Diagram prikazuje potek delovanja delovne zanke programa na visokem nivoju. Najprej se izvede meritev s senzorji, potem se podatke brezžično pošlje centralnemu vozlišču in po vnaprej določeni zakasnitvi se zanka ponovi. Zakasnitev je vnaprej določena na 5 sekund in je potrebna zato, da vozlišče varčuje pri porabi električne energije, ki je omejena.



Slika 4.5: Diagram aktivnosti delovne zanke vozlišča IZP.

4.4.2 Programska oprema centralnega vozlišča

Podobno izdelamo tudi diagram aktivnosti za centralno vozlišče, kot je prikazan na sliki 4.6. Prejemu podatkov iz IZP-jev sledi izpis prejetih vrednosti na UART vrata mikrokrmilnika. Ponovno dodamo določeno zakasnitev, s katero preprečimo preveliko obremenjenost mikrokrmilnika.



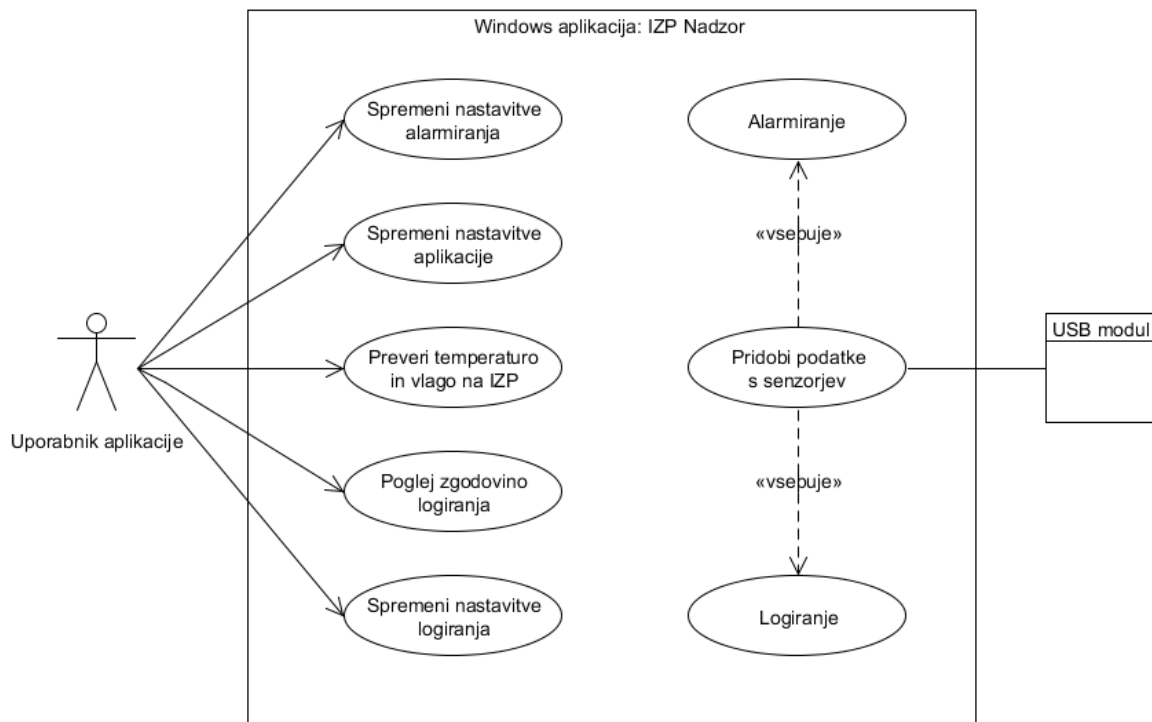
Slika 4.6: Diagram aktivnosti delovne zanke centralnega vozlišča.

4.4.3 Računalniška aplikacija IZP Nadzor

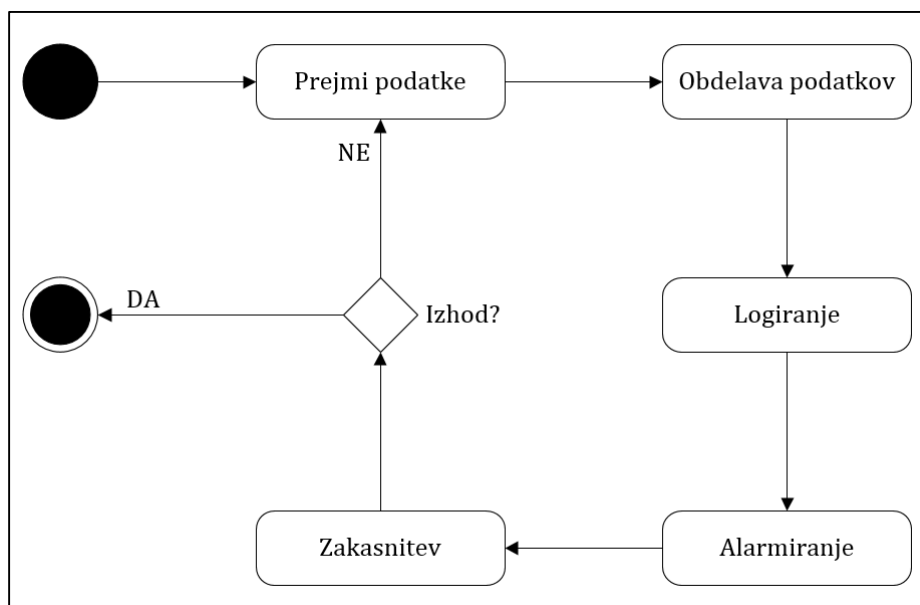
Slika 4.7 prikazuje UML diagram uporabe (angl. use case diagram) računalniške aplikacije IZP Nadzor. Diagram vsebuje tiste aktivnosti, ki so za končnega uporabnika razumljive in pomembne, ne prikazuje pa notranjega delovanja aplikacije.

Predvidimo pet glavnih aktivnosti, med katere spadajo: nadzor nad prejetimi vrednostmi iz senzorjev, pregled zgodovine teh vrednosti ter spreminjanje različnih nastavitev. Od uporabnika pričakujemo tudi, da razume, da aplikacija prejema podatke preko USB modula ter jih shranjuje (logiranje) ter preverja, če so vrednosti izven določenih meja (alarmiranje).

Slika 4.8 prikazuje diagram aktivnosti glavne zanke aplikacije, ki vsebuje najpomembnejše aktivnosti aplikacije. To so obdelava podatkov, logiranje in alarmiranje. Tudi za to zanko predvidimo zakasnitev, ki bo nastavljena na 1 sekundo.



Slika 4.7: Diagram uporabe aplikacije IZP Nadzor.



Slika 4.8: Diagram aktivnosti glavne delovne zanke aplikacije.

5. POGLAVJE

Izbira strojne opreme

The cheapest, fastest, and most reliable components are those that aren't there.

- Gordon Bell

Za sistem želimo izbrati strojno opremo, ki ustreza zahtevam, je zanesljiva in jo je možno dobiti po relativno nizki ceni. Pod strojno opremo sistema spadajo komponente, ki so prikazane na slikah 4.2 in 4.4, to so:

- centralna enota in USB brezžični modul,
- senzorji temperature in vlažnosti,
- zvočni in svetlobni pozivnik ter
- vir napajanja.

5.1 Brezžična vozlišča JeeLabs

Za strojno opremo brezžičnega vozlišča želimo rešitev, ki je že razvita in preizkušena, saj s tem bistveno zmanjšamo čas razvoja sistema. Po primerjavi brezžičnih vozlišč na trgu se odločimo za uporabo odprtokodnih rešitev proizvajalca JeeLabs, ki temeljijo na Arduino strojni in programski opremi.

Uporabimo dve vrsti brezžičnih vozlišč. Prvo je splošno senzorsko vozlišče JeeNode SMD (ver. 1), prikazano na sliki 5.1. Vozlišče ima Atmel mikrokontroler ATmega328 in radijski oddajnik/sprejemnik (angl. transceiver) RFM12B proizvajalca HopeRF. JeeNode SMD ima pet zunanjih priključkov, kar omogoča priključitev senzorjev in komunikacijskih modulov (npr. za komunikacijo z računalnikom).



Slika 5.1: Brezžično senzorsko vozlišče JeeNode SMD.

Druga vrsta vozlišča je USB modul JeeLink na sliki 5.2, ki služi kot centralno vozlišče. Vozlišče uporablja mikrokontroler Atmel ATmega328p ter oddajnik/sprejemnik RFM12B. JeeLink nima zunanjih priklonov, ima pa flash pomnilnik s kapaciteto 16 Mbit, ki je namenjen shranjevanju prejetih podatkov. Električni shemi vezij JeeNode SMD in JeeLink sta v prilogi 10.2.



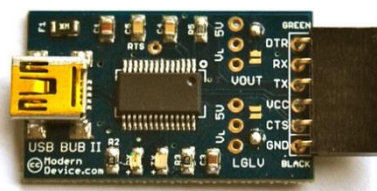
Slika 5.2: Brezžični USB modul JeeLink.

Obe vozlišči imata žično anteno (četrtnski bič), ki naj bi bila dolga eno četrtno valovne dolžine izbrane frekvence komunikacije, kar znaša:

$$L = \frac{1}{4} \cdot \frac{c}{f} = \frac{1}{4} \cdot \frac{299.792.458 \frac{\text{m}}{\text{s}}}{868 \text{ MHz}} = 8,63 \text{ cm} \quad (5.1)$$

Kjer je c hitrost svetlobe in f frekvenca brezžične komunikacije. Dejanska dolžina antene znaša 8 cm. Antena mora biti za dobro delovanje čim bolj ravna.

Za komunikacijo JeeNode SMD z računalnikom uporabimo modul USB BUB-II, prikazan na sliki 5.3. Modul vsebuje čip FT232R, ki deluje kot USB-UART vmesnik in omogoča komunikacijo med mikrokontrolerom in osebnim računalnikom.



Slika 5.3: Modul za USB komunikacijo USB BUB-II.

Mikrokrmilnik, ki ga uporablja JeeNode je Atmel ATmega328. Gre za 8-bitni mikrokrmilnik z AVR RISC arhitekturo, 32 KB flash pomnilnika, 1 KB EEPROM pomnilnika in 2 KB SRAM pomnilnika [19]. Po specifikacijah deluje v industrijskem temperaturnem razponu (od $-40\text{ }^{\circ}\text{C}$ do $85\text{ }^{\circ}\text{C}$). Poraba energije pri frekvenci delovanja 1 MHz, napajalni napetosti 1,8 V in temperaturi okolice $25\text{ }^{\circ}\text{C}$ znaša:

- 0,2 mA v aktivnem načinu,
- 0,1 μA v načinu »power-down«,
- 0,75 μA v načinu »power-save«.

Ohranitev podatkov znaša do 20 let pri $85\text{ }^{\circ}\text{C}$ in do 100 let pri $25\text{ }^{\circ}\text{C}$, kar zadošča za našo aplikacijo.

Druga pomembna komponenta, ki močno vpliva na delovanje vozlišča, je radijski modul RFM12B, proizvajalca HopeRF. Je večkanalni FSK oddajnik/sprejemnik na enem čipu z nizko porabo energije, skladen z FCC in ETSI predpisi za uporabo v ISM frekvenčnih pasovih 433 MHz, 868 MHz in 915 MHz. Za delovanje čipa sta potrebna le zunanji kvarčni kristal in filtriranje napajanja, vse ostale funkcije so integrirane. Modul zagotavlja urin signal tudi mikrokrmilniku, zato da ni potrebe po uporabi dveh kristalov.

5.2 Primerjava senzorjev

Glede na zahteve sistema potrebujemo štiri kombinirane senzorje temperature in relativne vlažnosti. Skupaj z izbrano strojno opremo vozlišča se uporabljata dva različna senzorja, za katera je že spisana programska oprema za komunikacijo:

- HYT-131 proizvajalca B+B Thermo-Technic GmbH (Nemčija) in
- SHT11 proizvajalca Sensirion (Švica).

Ugotoviti želimo kateri izmed senzorjev bolj ustreza našim tehničnim zahtevam in kateri je cenovno ugodnejši.

Oba senzorja sta fizično majhna integrirana senzorja, za katera proizvajalca trdita, da imata majhno porabo električne energije in dobro dolgoročno stabilnost [20, 21]. Vsebujeta kapacitivno-napetostni pretvornik za merjenje relativne vlažnosti preko kapacitivnosti ter silicijev bandgap temperaturni senzor (silicijeva dioda). Silicijev bandgap temperaturni senzor izkorišča dejstvo, da je prevodna napetost silicijeve diode odvisna od temperature. S primernim vezjem dosežemo, da je korelacija med napetostjo in temperaturo linearna. Kapacitivni senzor relativne vlažnosti uporablja princip po katerem je dielektrična konstanta odvisna od temperature in vlage dielektrika.

Oba merilna elementa sta vezana na 14-bitni ADC. Vsak senzor je posamično kalibriran. Koeficienti kalibracije so shranjeni v OTP pomnilniku (pomnilnik, ki omogoča le enkratno pisanje). Nad meritvami se izvede digitalno procesiranje z danimi koeficienti in nato se meritve začasno shranijo v električno izbrisljiv EEPROM pomnilnik, iz katerega jih kasneje mikrokrmilnik prebere.

Senzorja uporabljata dve žici za komunikacijo z mikrokrmilnikom ter dve za napajanje. Razlikujeta se po načinu komunikacije. Senzor HYT-131 ima vmesnik, ki je kompatibilen z I²C vodilom, senzor SHT11 pa ima natančno podana navodila za komunikacijo, ki ne temeljijo na nobenem protokolu. Razporeditev pinov je enaka za oba senzorja, tako da sprememba tiskanega vezja ni potrebna.

V tabeli 5.1 so prikazani različni tehnični podatki senzorjev iz dokumentov njunih proizvajalcev. Ugotovimo, da sta si senzorja v večini lastnosti podobna, bistveno se razlikujeta le pri temperaturnem območju shrambe, za senzor HYT-131 pa nekateri podatki niso podani.

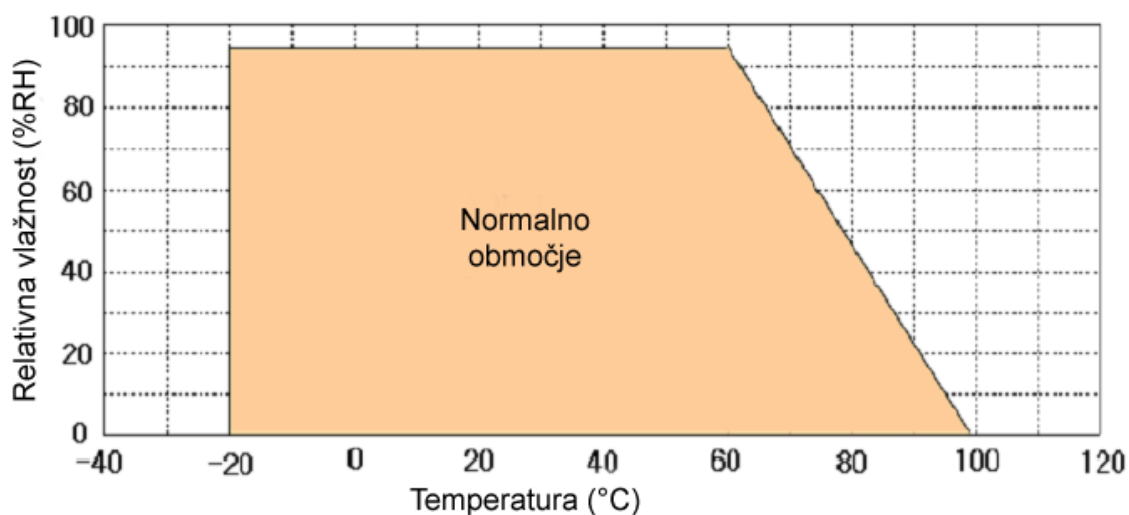
Oba senzorja delujeta stabilno znotraj normalnega območja, ki je prikazano na sliki 5.4. Dolgoročna izpostavljenost razmeram izven normalnega območja (še posebej pri vlažnosti večji od 80 % RV) povzroči začasen odmik meritev vlažnosti. Po povratku v normalno območje se odmik počasi izniči.

Zanima nas tudi največja natančnost meritev pri različnih temperaturah, kar prikazuje slika 5.5. Ugotovimo, da ima senzor HYT-131 boljšo natančnost meritev, pri čemer se zavedamo, da natančnost meritve za našo aplikacijo ni ključnega pomena.

Primerjamo tudi največjo natančnost meritve vlažnosti, prikazano na sliki 5.6, kjer ugotovimo, da ima senzor HYT-131 ponovno boljšo natančnost meritev, vendar tokrat za manjši faktor.

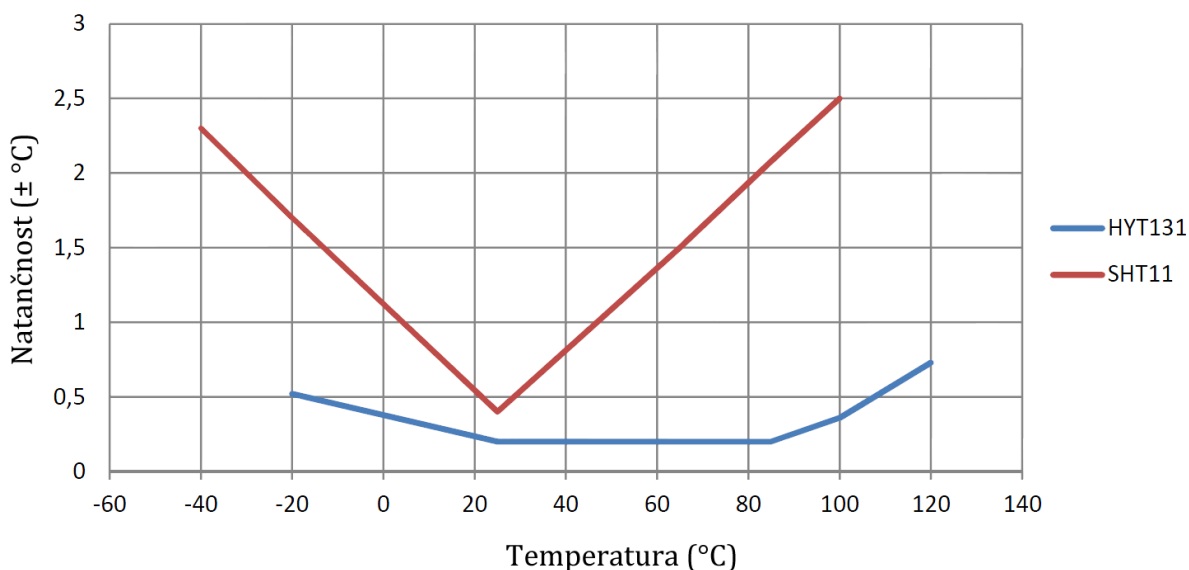
Obratovalni pogoji	HYT-131	SHT11
Napajalna napetost	2,3 V – 5,5 V	2,4 V – 5,5 V (tip. 3,3 V)
Temperaturno območje shrambe	-55 °C – 150 °C	10 °C – 50 °C
Območje vlažnosti shrambe	?	20 % RV – 60 % RV
Senzor temperature		
Ločljivost	12 bitov	12-14 bitov
Temperaturni razpon obratovanja	-40 °C – 120 °C	-40 – 123,8 °C
Natančnost	$\pm 0,3$ °C	$\pm 0,4$ °C
Odzivni čas	15 – 25 s	5 – 30 s
Dolgoročno lezenje	?	$< 0,04$ °C / leto
Ponovljivost	?	$\pm 0,1$ °C
Senzor vlažnosti		
Ločljivost	12 bitov	8-12 bitov
Vlažnostni razpon obratovanja	0 % RV – 100 % RV	0 % RV – 100 % RV
Natančnost	± 2 % RV	± 2 % RV
Odzivni čas	10 s	8 s
Ponovljivost	?	$\pm 0,1$ % RV
Histereza	$< \pm 1,5$ % RV	tip. $\pm 1,0$ % RV
Dolgoročno lezenje	?	$< 0,5$ % RV / leto

Tabela 5.1: Primerjava parametrov obeh senzorjev.

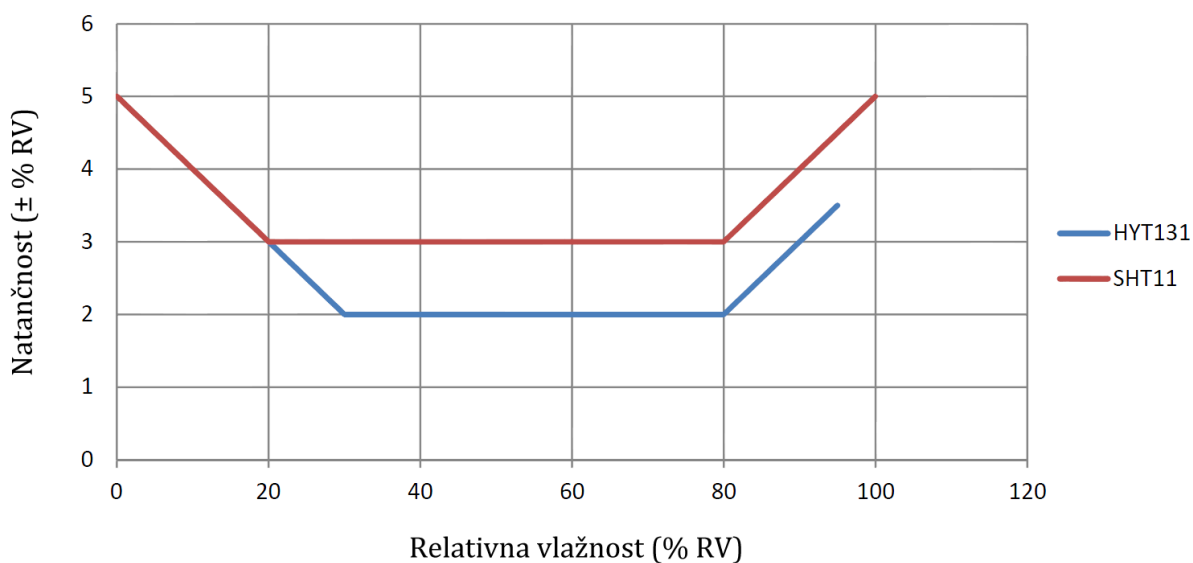


Slika 5.4: Normalno območje delovanja obeh senzorjev.

Senzorja primerjamo še cenovno in ugotovimo, da je senzor SHT11 povprečno cenejši za 20 %. Glede na to, da ni občutnejših razlik pri zmogljivostih senzorjev in da nam natančnost ni ključnega pomena, se odločimo za uporabo cenejšega senzorja SHT11.



Slika 5.5: Največja natančnost meritev temperature pri različnih temperaturah.



Slika 5.6: Največja natančnost meritev vlage pri različnih vlažnostih (pri 25 °C).

5.3 Kombinirani senzor SHT-11

Komunikacija z izbranim senzorjem SHT11 se prične s sekvenco za začetek prenosa, čemur sledi eden izmed ukazov. Možni ukazi so: meritev temperature, meritev relativne vlažnosti, branje statusnega registra, pisanje statusnega registra ter mehki reset, ki resetira vmesnik in pobriše statusni register. Statusni register omogoča branje in pisanje nekaterih naprednejših funkcij senzorja, kot so izbira merilne ločljivosti, vklop in

izklop vgrajenega grelca ter ugotavljanje ali je napetost na senzorju prenizka. Omogočena nam je tudi izključitev nalaganja kalibracijskih podatkov pred vsako meritvijo, s čimer zmanjšamo čas meritve za 10 ms.

Ekstremne razmere in izpostavljenost hlapom topila odmaknejo vrednosti senzorja toliko, da se te ne povrnejo v normalno stanje. Senzor pripeljemo nazaj v stanje kalibriranosti s postopkom obnovitve. Postopek obnovitve za senzor SHT11 je sledeč: peka pri temperaturi 100 °C – 105 °C pri < 5 % RV za 10 ur ter zatem rehidracija pri temperaturi 20 °C – 30 °C pri približno 75 % RV za 12 ur. Poraba električne energije za senzor SHT11 znaša:

- Spanje: 2 µW – 5 µW
- Meritev: 3 mW
- Povprečno: 90 µW

5.3.1 Pretvorba merilnih podatkov

Po pridobitvi merilnih podatkov iz senzorja mora mikrokrmilnik nad njimi izvesti določene računske operacije, da pridobi točen rezultat meritev. Temperaturni senzor je zelo linearen, zato dobljene vrednosti preprosto pretvorimo z enačbo:

$$T = d_1 + d_2 \cdot S_T \quad (5.2)$$

Kjer je S_T dobljena vrednost s senzorja, T prava izmerjena vrednost temperature, koeficienta d_1 in d_2 pa sta podana v tabelah 5.2 in 5.3.

VDD	$d_1[^\circ\text{C}]$
5 V	- 40,1
4 V	- 39,8
3,5 V	- 39,7
3 V	- 39,6
2,5 V	- 39,4

Tabela 5.2: Koeficient za izračun temperature pri različnih napajalnih napetostih.

S_T	$d_2[^\circ\text{C}]$
14 bit	0,01
12 bit	0,04

Tabela 5.3: Koeficient za izračun temperature pri različnih natančnostih meritve.

Pri relativni vlažnosti je potrebno kompenzirati nelinearnost senzorja z enačbo:

$$RV_{lin} = c_1 + c_2 \cdot S_{RV} + c_3 \cdot S_{RV}^2 \quad [\% RV] \quad (5.3)$$

Kjer je S_{RV} vrednost relativne vlažnosti s senzorja, koeficienti c_1 , c_2 in c_3 pa so podani v tabeli 5.4 za dve različni natančnosti meritev.

S_{RV}	c_1	c_2	c_3
12 bit	- 2,0468	0,0367	$- 1.5955 \cdot 10^{-6}$
8 bit	- 2,0468	- 0,5872	$- 4.0845 \cdot 10^{-6}$

Tabela 5.4: Koeficienti za nelinearno kompenzacijo relativne vlažnosti.

Pri temperaturah, ki bistveno odstopajo od 25 °C je potrebno meritve relativne vlažnosti tudi temperaturno kompenzirati.

$$RV = (T_{°C} - 25) \cdot (t_1 + t_2 \cdot S_{RV}) + RV_{lin} \quad [\% RV] \quad (5.4)$$

Kjer je $T_{°C}$ izmerjena temperatura v °C, koeficienta t_1 in t_2 pa sta podana v tabeli 5.5 za dve različni natančnosti meritev.

S_{RV}	t_1	t_2
12 bit	0,01	0,00008
8 bit	0,01	0,00128

Tabela 5.5: Koeficienta za temperaturno kompenzacijo relativne vlažnosti.

5.4 Svetlobni in vibracijski pozivnik

Za pozivnike uporabimo LilyPad plošče proizvajalca SparkFun Electronics, ki so namenjene vstiju v oblačila. Svetlobni pozivnik LilyPad Tri-Color LED, prikazan na sliki 5.7, omogoča osvetlitev poljubne barve s tremi barvnimi kanali (rdeč, zelen, moder). Premer pozivnika je 20 mm. Tiskano vezje ima debelino 0,8 mm.



Slika 5.7: Svetlobni pozivnik LilyPad Tri-Color LED.

Drugi pozivnik je vibracijski pozivnik LilyPad Vibe Board, prikazan na sliki 5.8. Tiskano vezje ima nameščen SMD vibracijski motor Precision Microdrives 310-101, ki se vrti s hitrostjo 12.000 obratov na minuto. Dimenzije tiskanega vezja pozivnika so enake:

20 mm premera pri 0,8 mm debeline. Električni shemi vezij obeh pozivnikov se nahajata v prilogi 10.1 na slikah 10.1 in 10.2.



Slika 5.8: Vibracijski pozivnik LilyPad Vibe.

5.5 Napajalni vir in polnilnik

Za vir napajanja želimo elektrokemično baterijo sekundarne vrste z napajalno napetostjo vsaj 3,3 V. Odločimo se za baterijo z litijevim polimerom kapacitete 1500 mAh in tremi izhodnimi napetostmi: 4 V, 8 V in 12 V. Zmožnost praznjenja baterije je označena kot 25C, kar pomeni, da se lahko teoretično izprazni v 144 s, pri čemer vzdržuje največji tok 37,5 A. Praktično žice takšnega toku ne bi prenesle. Vsekakor je zmožnost praznjenja baterije zadostna za našo aplikacijo.

Za baterijo potrebujemo tudi polnilnik, ki jo je zmožen v doglednem času napolniti. Odločimo se za polnilnik proizvajalca Sparkfun s 5 V vhodne napetosti in zmožnostjo polnjenja pri toku 1 A. Polnilnik temelji na polnilnem krmilniku za baterije z litijevim polimerom MCP73843-410 proizvajalca Microchip.

6. POGLAVJE

Izdelava programske opreme

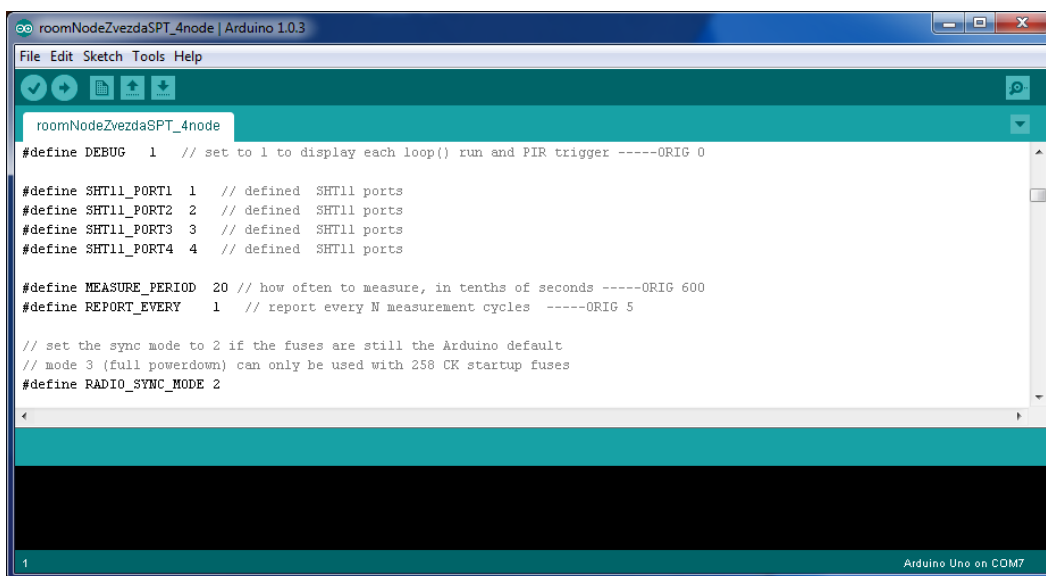
A good programmer is someone who always looks both ways before crossing a one-way street.

- Doug Linder

Za razvoj programske opreme na JeeNode vozlišču in JeeLink USB modulu uporabimo Arduino integrirano razvojno okolje (angl. integrated development environment – IDE). Z okoljem izdelamo dva programa, prvega za brezžično vozlišče IZP (JeeNode) in drugega za centralno vozlišče (brezžični USB modul).

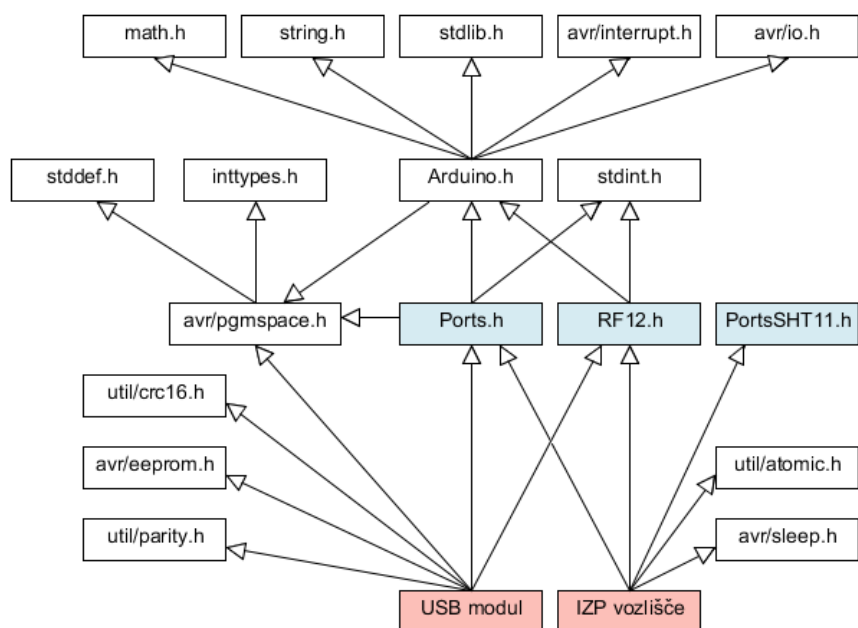
6.1 Arduino okolje

Arduino platforma sestoji iz odprtokodne strojne opreme (mikrokrmilnik AVR in ostale komponente na tiskanem vezju) ter odprtokodne programske opreme, ki jo sestavljata prevajalnik in zagonski nalagalnik. Za programiranje strojne opreme uporabimo več-platformno Arduino IDE razvojno okolje, prikazano na sliki 6.1. Okolje vsebuje urejevalnik besedila za pisanje in urejanje izvorne kode, področje za sporočila ter tekstovno konzolo. Okolje omogoča preverjanje, prevajanje in nalaganje programske kode na mikrokrmilnik preko navideznega serijskega porta. Ena datoteka izvorne kode se imenuje skica (angl. sketch) in je na datotečnem sistemu shranjena s končnico ».ino«. Področje za sporočila in konzola prikazujeta raznovrstna sporočila, opozorila in napake v zvezi z delovanjem programa. Aplikacija omogoča izbiro vrste Arduino plošče, ki je trenutno v uporabi.



Slika 6.1: Arduino razvojno okolje v operacijskem sistemu Windows 7.

Izvorno kodo pišemo v programskem jeziku C++, v katerem so spisane tudi knjižnice z različnimi razredi, funkcijami in definicijami. Uporaba knjižnic močno pospeši izdelavo programske opreme, saj vsebujejo že izdelane funkcije za upravljanje s strojno opremo in za manipulacijo podatkov. Poleg standardnih knjižnic lahko uporabljamo knjižnice drugih proizvajalcev in lastne knjižnice. Slika 6.2 prikazuje uporabo knjižnic v obeh programih. Za oba uporabimo standardne knjižnice Arduino ter knjižnice proizvajalca JeeLabs, ki so na sliki obarvane z modro barvo.



Slika 6.2: Struktura knjižnic v programski opremi.

Vsaka skica naj bi vsebovala funkciji `setup()` in `loop()`, ki sta definirani v knjižnici `Arduino.h`. Funkcija `setup()` se pokliče ob zagonu programa in se uporablja za začetno nastavitvev (inicializacija spremenljivk, nastavitvev načinov delovanja pinov, ipd.). Funkcija se izvrši samo enkrat po vklopu in po ponovnem zagonu vezja. Za tem se začne neprekinjeno izvajati funkcija `loop()`, ki predstavlja glavno zanko programa.

6.2 Programska oprema vozlišča IZP

Program na brezžičnem senzorskem vozlišču IZP sestavlja ena Arduino skica, ki uporablja knjižnice, kot je prikazano na sliki 6.2. Izhajali smo iz obstoječega primera in ga nadgradili glede na naše zahteve. Funkcije v skici so:

- `setup()` – postavitve naprave,
- `loop()` – delovna zanka naprave,
- `shtDelay()` – zakasnitev za izvedbo meritev,
- `doMeasure()` – izvedba meritve,
- `doReport()` – izvedba poročila.

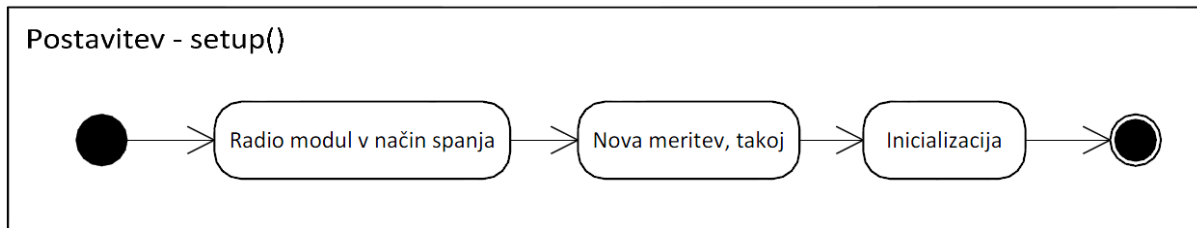
Program vsebuje tudi nekatere definicije (parametre), ki se fiksno nastavijo v kodi:

- `MEASURE_PERIOD` – časovna perioda izvajanja meritev (nastavljeno na 5 s),
- `REPORT_EVERY` – izvedba poročila na vsakih toliko meritev (nastavljeno na 1),
- `ALARM_TEMP` – temperatura pri kateri se sproži alarm (nastavljeno na 50 °C).

Izvajanje programa temelji na vgrajenem planerju (angl. scheduler), ki omogoča izvajanje dogodkov z zakasnitvijo.

6.2.1 Funkcija `setup()`

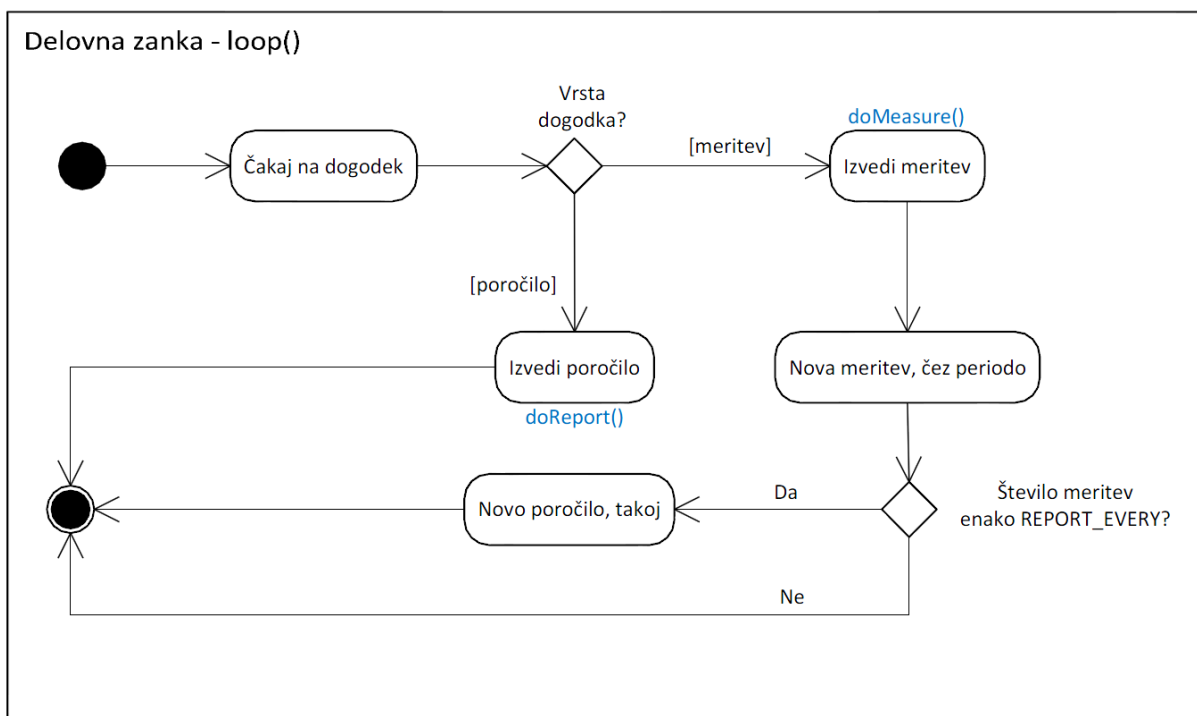
Delovanje posameznih funkcij ponazorimo z UML diagrami aktivnosti. Na sliki 6.3 je prikazan diagram aktivnosti za funkcijo `setup()`. Ta postavi radijski modul v način spanja in na planerju doda nov dogodek za izvedbo meritve, takoj ko je možno. V funkciji se postavijo tudi nekatere začetne nastavitve pinov ter brezžične komunikacije. Funkcija `setup()` se nato zaključi in se ne kliče več do naslednjega zagona vozlišča.



Slika 6.3: Diagram aktivnosti funkcije za postavitev naprave – setup().

6.2.2 Funkcija loop()

Slika 6.4 prikazuje diagram aktivnosti funkcije loop(), ki se izvaja neprekinjeno. Funkcija v začetku izvajanja čaka na dogodek. Med čakanjem je mikrokrmilnik v načinu varčevanja z energijo. Možna dogodka sta dva: izvedba meritve in izvedba poročila. Pri izvedbi meritve se kliče funkcija doMeasure() ter na planerju se nastavi nova meritev čez izbrano periodo. Če je število izvedenih meritev od zadnjega pošiljanja poročila enako nastavljenemu parametru REPORT_EVERY, se na planer doda novo poročilo, ki se izvede takoj. V primeru dogodka za izvedbo poročila se pokliče funkcija doReport().

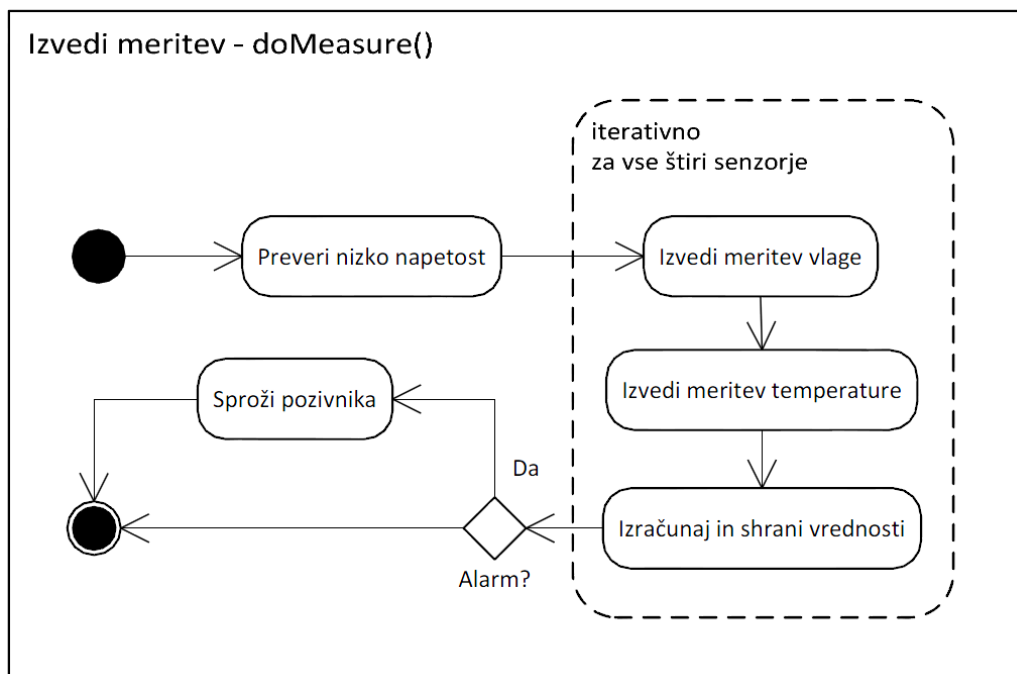


Slika 6.4: Diagram aktivnosti funkcije za neprekinjeno izvajanje – loop().

6.2.3 Funkcija doMeasure()

Slika 6.5 prikazuje diagram aktivnosti funkcije doMeasure(), ki najprej preveri morebitno nizko napetost na radijskem modulu, ter shrani stanje v paket. Za tem se

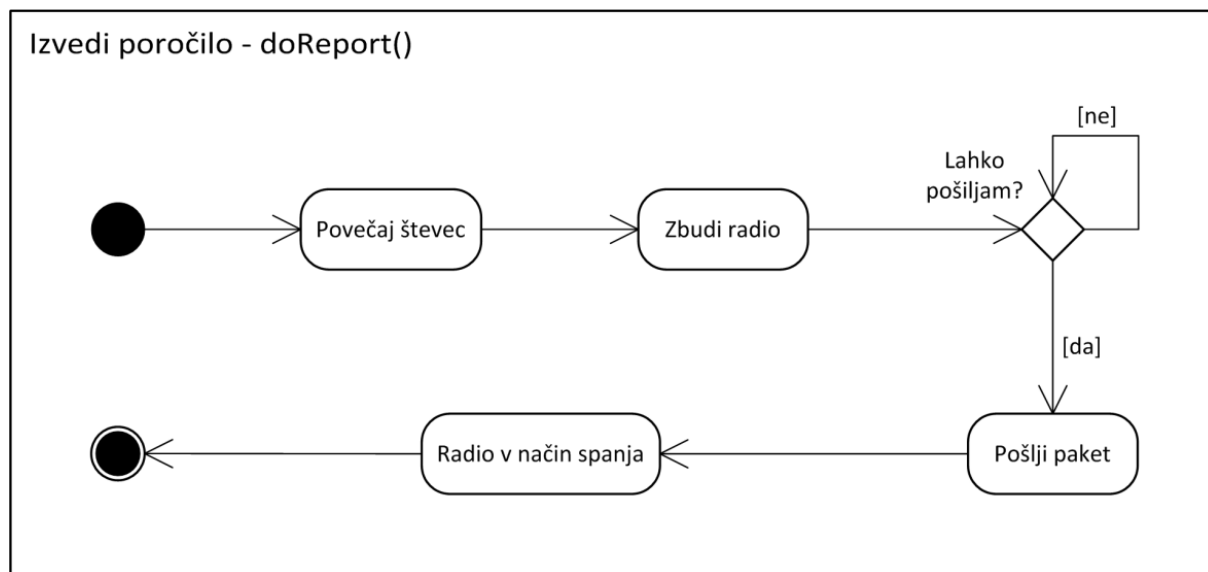
iterativno izvedejo meritve in izračuni temperature ter vlage na vseh štirih kombiniranih senzorjih. Za izvedbo meritev se kliče funkcija `measure()`, za izračun pa funkcija `calculate()` razreda `SHT11`. Vrednosti se shranijo v paket, če pa je katera izmed temperatur previsoka, se sproži še pozivnika.



Slika 6.5: Diagram aktivnosti funkcije za izvedbo meritev – `doMeasure()`.

6.2.4 Funkcija `doReport()`

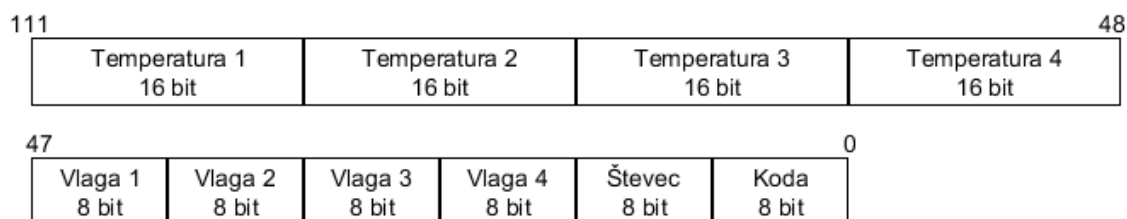
Slika 6.6 prikazuje diagram aktivnosti funkcije `doReport()`, ki skrbi za brezžično pošiljanje izmerjenih podatkov. Najprej se poveča 8-bitni ciklični števec, ki služi diagnostičnim namenom in nato se iz varčnega načina prebudi radijski modul. Paket se pošlje, ko je to možno, na koncu pa se radijski modul ponovno postavi v način spanja.



Slika 6.6: Diagram aktivnosti funkcije za izvedbo poročila – doReport().

6.2.5 Zgradba komunikacijskih paketov

Zgradba paketov, ki jih vozlišče brezžično pošilja je prikazana na sliki 6.7. Zgradba paketa je v C++ jeziku definirana kot struktura, sestavljena iz štirih 16-bitnih spremenljivk tipa »integer« in šestih 8-bitnih spremenljivk tipa »byte«. Števec je 8-bitni ciklični števec poslanih paketov, ki se uporablja za diagnostične namene. Koda se uporablja za opozarjanje na določene dogodke ali stanja in trenutno služi le opozarjanju na nizko napetost napajalnega vira, kar zavzame 1 bit. Tako ostaja 7 prostih bitov za kasnejše razširitve. Velikost paketa je majhna, le 112 bitov informacije, saj želimo čim manjšo porabo električne energije. Poleg tega krajši paketi pomenijo manjšo možnost prekrivanja z ostalimi signali in posledično večjo verjetnost uspešne komunikacije. Perioda pošiljanja mora biti dovolj majhna, da ima aplikacija čim bolj ažurne podatke in hkrati ne sme biti premajhna, zaradi prevelike porabe energije. Odločimo se za periodo izvajanja meritev na vsakih 5 sekund ter za pošiljanje poročila po vsaki izvedeni meritvi.



Slika 6.7: Zgradba paketov, ki jih pošilja IZP brezžično vozlišče.

6.3 Programska oprema centralnega vozlišča

Tudi na USB modulu programska oprema sestoji iz enega izvirnega dokumenta, ki uporablja knjižnice tako, kot so prikazane na sliki 6.2. Podobno smo tudi tukaj izhajali iz obstoječega primera, vendar ga za razliko od IZP vozlišča nismo veliko spreminjali. Spremenili smo le nastavitve za brezžično komunikacijo, saj ostalo že ustreza našim zahtevam.

Program za komunikacijo z računalnikom uporablja serijski port UART oz. USART, ki je prisoten na vseh Arduino ploščah. S portom upravljamo z uporabo objekta `Serial`, ki je definiran v standardnih knjižnicah. Uporabljena serijska komunikacija ima sledeče lastnosti:

- hitrost 57.600 bitov na sekundo,
- 8 podatkovnih bitov,
- brez paritete in
- 0 stop bitov.

Najpomembnejše funkcije v izvorni datoteki so:

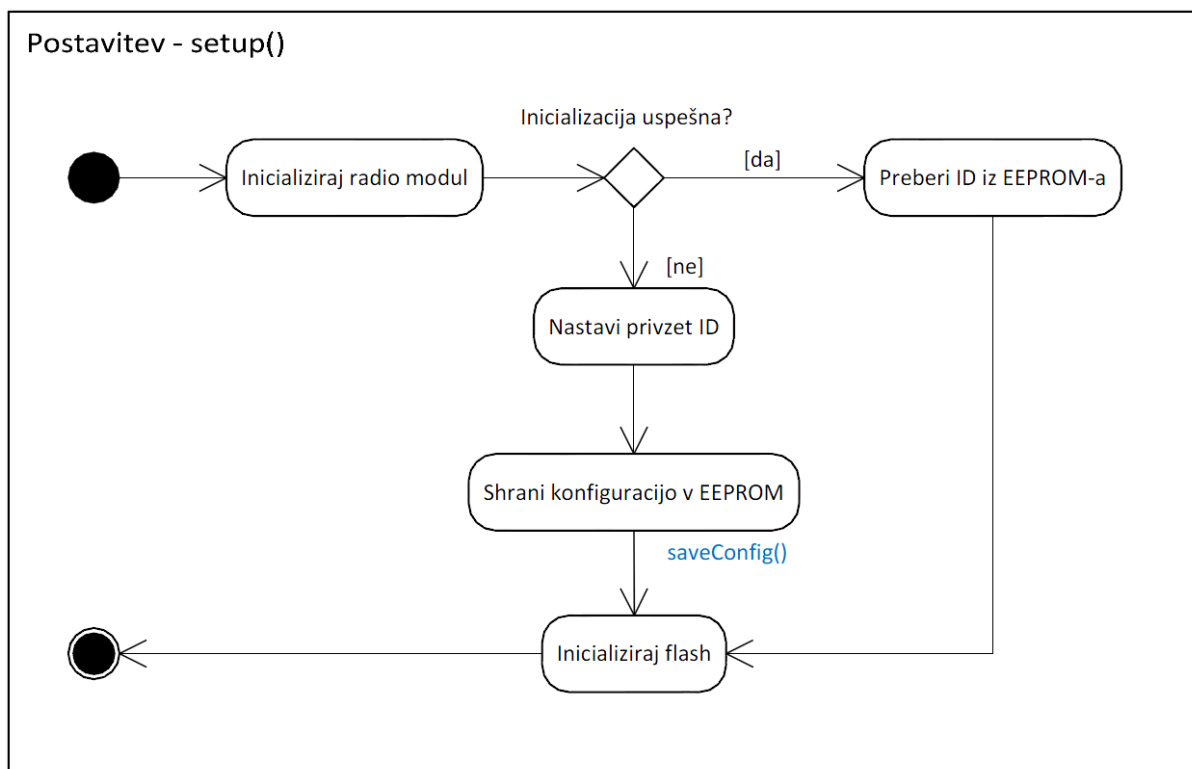
- `setup()` – postavitve naprave,
- `loop()` – delovna zanka naprave,
- `saveConfig()` – shrani trenutno konfiguracijo v EEPROM,
- `handleInput()` – izvrši prejet ukaz.

6.3.1 Funkcija `setup()`

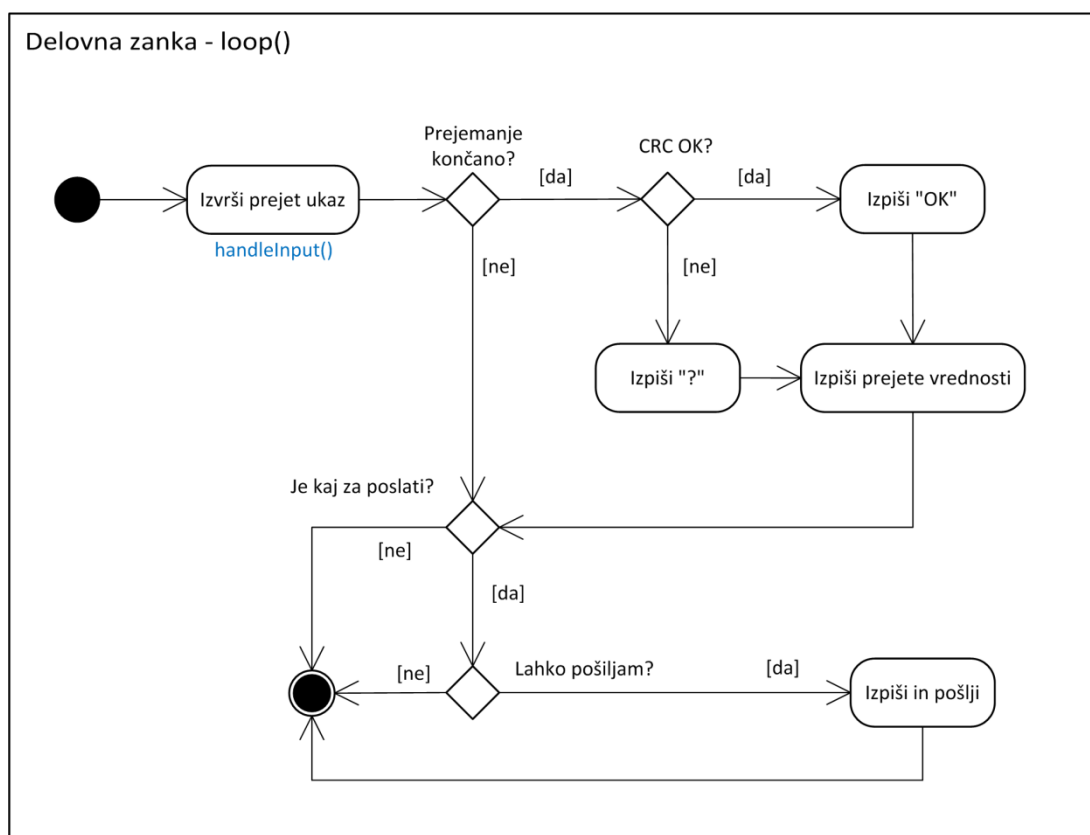
Diagram aktivnosti funkcije `setup()` je prikazan na sliki 6.8. Funkcija najprej izvede inicializacijo radijskega modula iz podatkov zapisanih v EEPROM pomnilniku. Če je uspešna, se prebere vrednost ID vozlišča, v nasprotnem primeru se nastavi privzeta vrednost. Za konec se izvede še inicializacija flash pomnilnika.

6.3.2 Funkcija `loop()`

Diagram aktivnosti funkcije `loop()` je prikazan na sliki 6.9. Najprej se izvede funkcija `handleInput()`, ki preveri če je bil preko serijske komunikacije prejet kakšen ukaz in se izvrši. Po končanem prejemanju podatkov se izračuna CRC ter prejete vrednosti, čemur sledi še pošiljanje podatkov, če je trenutno to možno.



Slika 6.8: Diagram aktivnosti funkcije za postavitev USB modula – setup().



Slika 6.9: Diagram aktivnosti delovne zanke USB modula – loop().

7. POGLAVJE

Izdelava računalniške aplikacije

As far as the customer is concerned, the interface is the product.

- Jef Raskin

V poglavju 4.6 so opisane vse glavne funkcionalnosti, ki jih pričakujemo od računalniške aplikacije IZP Nadzor. Za razvoj aplikacije uporabimo splošno agilno metodo, ki temelji na osnovnih idejah ponavljajočega (iterativnega) in postopnega (inkrementalnega) razvoja. Takšna metodologija nam omogoča dober nadzor nad razvojem in večjo fleksibilnost ter hkrati prinaša nekatere obveznosti. Programska koda mora biti pregledna in razumljiva ter mora omogočati razširitve. Spisana mora biti na tak način, da je možno uvesti večje konceptualne spremembe brez prevelikega napora. Za razliko od nekaterih drugih metod (npr. waterfall metoda) so spremembe zahtev med razvojem zaželeni, saj na tak način lahko izdelamo tehnično dovršen produkt, ki je zadovoljiv končnemu uporabniku.

7.1 Scenariji uporabe

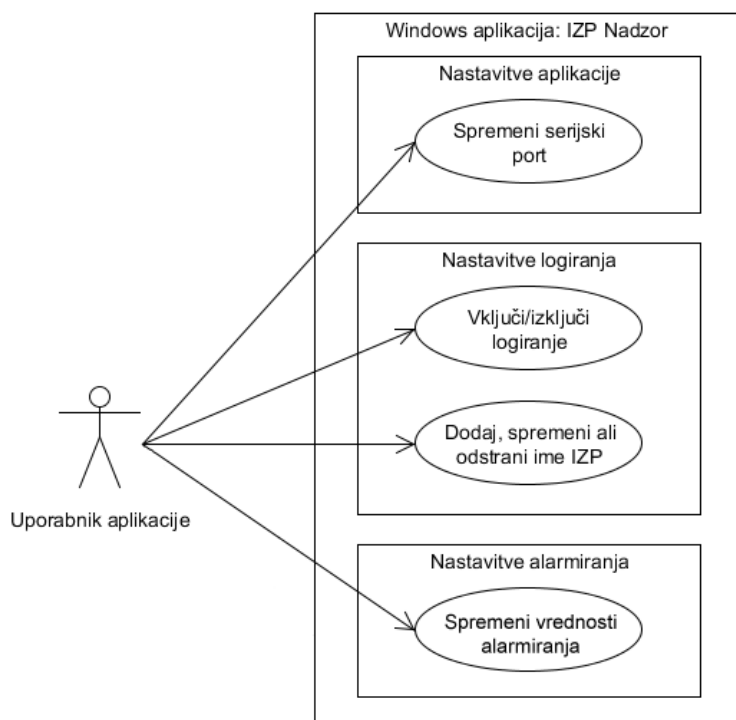
Preden začnemo z razvojem računalniške aplikacije, želimo predvideti, kako se bo ta uporabljala z vidika končnega uporabnika. To nam omogoča lažje sprejemanje odločitev pri zasnovi aplikacije in njenega uporabniškega vmesnika. Različne scenarije uporabe aplikacije prikažemo z UML diagrami uporabe, ki prikazujejo interakcijo uporabnika s sistemom. Potrebno se je zavedati, kateri scenariji so bolj in kateri manj pogosti, zato da pravilno prioriziramo razvoj ter izdelamo učinkovit uporabniški vmesnik. Druga zadeva, ki nas zanima glede scenarijev, je njihova pomembnost. Pomembnejšim scenarijem namenimo več časa preprečevanju, iskanju in odpravljanju

napak v programski kodi. Glede na situacijo in način uporabe aplikacije njeno uporabo razdelimo na štiri glavne scenarije:

- nastavitev,
- testiranje,
- pasivna uporaba in
- aktivna uporaba.

7.1.1 Nastavitev

Na sliki 7.1 je prikazan diagram uporabe za scenarij spreminjanja nastavitev aplikacije. Vrste nastavitev razdelimo na tri dele: nastavitve aplikacije, logiranja in alarmiranja.



Slika 7.1: Diagram uporabe pri scenariju spreminjanja nastavitev aplikacije.

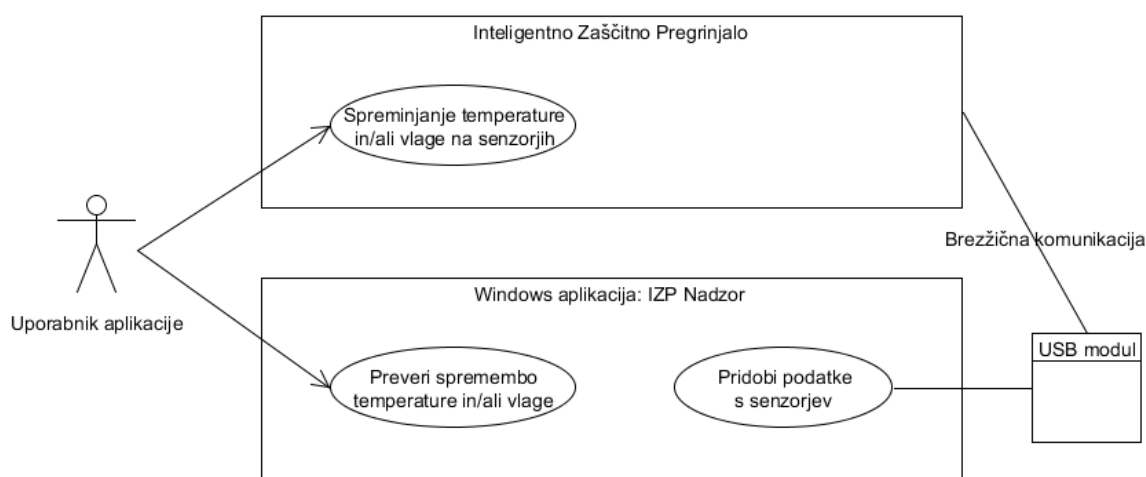
Najpomembnejša nastavev je zagotovo izbira serijskega porta za komunikacijo z USB modulom. Zaenkrat je to edina nastavev, ki spada v prvo skupino. Vseeno želimo pustiti odprto možnost kasnejšega dodajanja novih nastavitev. Za logiranje želimo imeti možnost izključitve, če uporabnik te funkcije ne potrebuje. Druga pomembna nastavev je možnost prireditve imena vsakemu vozlišču IZP. Vsako vozlišče ima namreč edinstveno

identifikacijsko (ID) številko in to nastavitve povežemo s poljubnim imenom (npr. ime uporabnika IZP), kar omogoča lažji nadzor. Zadnja skupina nastavitve omogoča spreminjanje vrednosti temperature in vlage, pri katerih se vključi opozorilo oz. alarm.

Za shranjevanje nastavitve v konfiguracijsko datoteko uporabimo značkovni jezik XML. Sklepamo, da je XML povprečnemu uporabniku razumljiv do te mere, da v primeru težav lahko ročno popravi kakšno nastavitve po navodilih ali samostojno. V vsakem primeru mora biti aplikacija zgrajena tako, da uporabniku ni potrebno ročno spreminjati nastavitve. Če vseeno pride do neustreznih sprememb konfiguracijske datoteke, mora aplikacija težavo razrešiti na primeren način ali ponovno ustvariti konfiguracijsko datoteko s privzetimi nastavitvami.

7.1.2 Testiranje

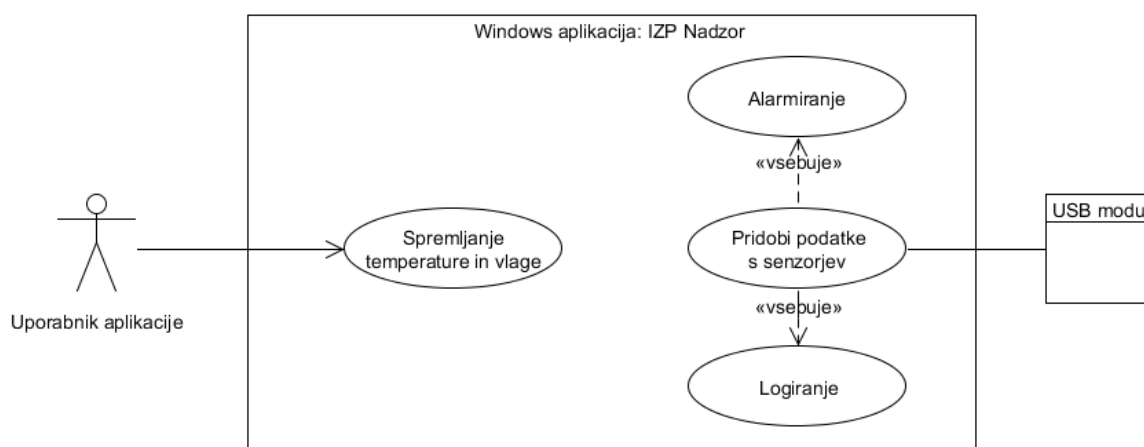
Pri scenariju testiranja uporabnik preverja, če aplikacija pravilno pridobiva vrednosti s senzorjev. Diagram uporabe, ki ponazarja testiranja sistema, je prikazan na sliki 7.2. Uporabnik aplikacije ima na razpolago enega ali več IZP-jev na katerih v okolici senzorjev spreminja temperaturo in vlago. Način spreminjanja veličin ni predpisan pričakuje pa se, da uporabnik ne uporablja metod, ki bi poškodovale IZP. Po začetni nastavitvi aplikacije želi uporabnik najprej ugotoviti ali prejema vrednosti od IZP-ja in nato prične s spreminjanjem veličin v okolici senzorjev. Takšno testiranje je zaželeno zato, da se pred uporabo sistema ugotovijo morebitne težave.



Slika 7.2: Diagram uporabe pri scenariju testiranja aplikacije.

7.1.3 Aktivna in pasivna uporaba

Slika 7.3 prikazuje diagram uporabe pri scenariju **aktivne in pasivne uporabe**. Konceptualno se scenarija malo razlikujeta, praktično gledano pa je razlika bistvena. V obeh primerih je aplikacija zagnana in IZP-ji so v uporabi, vendar pri pasivni uporabi ni nujno da uporabnik spremlja aplikacijo, saj je ta lahko v pomanjšanem načinu oz. v orodni vrstici. Pri aktivni uporabi so uporabniki IZP-jev potencialno izpostavljeni visokim temperaturam, zato uporabnik aplikacije aktivno spremlja vrednosti na aplikaciji.



Slika 7.3: Diagram uporabe pri scenariju aktivne/pasivne uporabe.

Pri uporabi sistema uporabniki IZP-jev pogosto ne bodo izpostavljeni življenjsko nevarnim situacijam in še redkeje bodo v nevarnosti izpostavitvi previsokim temperaturam. Zaradi tega razloga predvidimo scenarij pasivne uporabe, kjer se ne pričakuje, da uporabnik aktivno spremlja aplikacijo. Če na enem od IZP-jev vseeno pride do prekoračitve mej, se mora aplikacija prikazati na zaslonu pred ostalimi aplikacijami in s tem opozoriti uporabnika na morebitno nevarnost.

Pri aktivni uporabi sistema gre za situacijo, kjer so uporabniki IZP-jev izpostavljeni nevarnosti, zato se predvideva, da uporabnik aplikacije aktivno spremlja odčitke s senzorjev ter v primeru prekoračitve na to opozori uporabnike IZP-ja. Celotna aplikacija mora biti zasnovana okoli tega najpomembnejšega scenarija, saj je ključno, da je aplikacija odzivna in enostavna za uporabo takrat, ko jo najbolj potrebujemo.

Poleg vseh omenjenih scenarijev, ne smemo pozabiti tudi na nekatere druge aktivnosti, ki so prav tako pomembne za končnega uporabnika. Namestitev aplikacije na računalniku mora biti enostavna za povprečnega uporabnika in mora vsebovati vsa

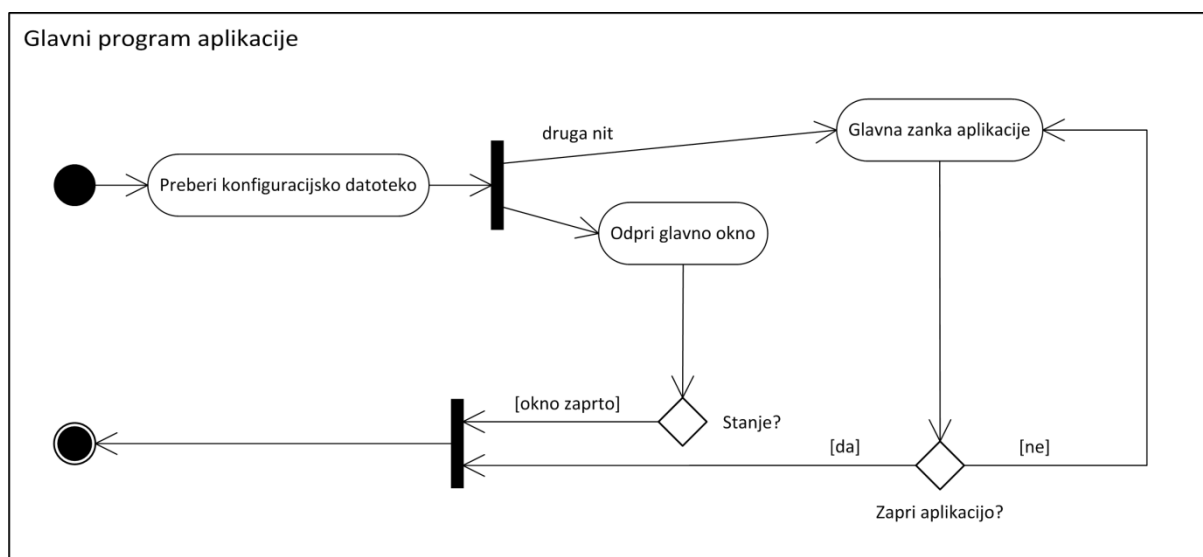
potrebna navodila. Podobno velja tudi za izbris aplikacije. Uporabniku mora biti omogočen tudi pregled zgodovine prejetih podatkov.

7.2 Razvoj aplikacije

Za razvoj aplikacije se odločimo za programsko ogrodje .NET, okolje Microsoft Visual Studio 2012 ter jezik C#. Ta kombinacija nam omogoča hiter razvoj aplikacij za Windows in hkrati daje dovolj svobode pri zasnovi. Aplikacije, razvite v .NET ogrodju, se izvajajo v aplikacijskem navideznem stroju (angl. application virtual machine), ki se imenuje CLR (angl. Common Language Runtime). Knjižnica razredov ter CLR skupaj sestavljata .NET ogrodje. Uporabimo različico .NET 3.5, ker je ta že v osnovi nameščena na operacijskem sistemu Windows 7.

7.2.1 Niti aplikacije

Aplikacija uporablja en proces in teče na dveh nitih (angl. thread), prikazanih na sliki 7.4. Prva nit skrbi za nemoteno delovanje uporabniškega vmesnika, druga pa za odjemanje podatkov, logiranje in alarmiranje. Če pride do zaustavitve ali napake v prvi niti se pomembne funkcije še naprej izvajajo. Če se zaustavi druga nit, jo prva nit samodejno ponovno zažene.

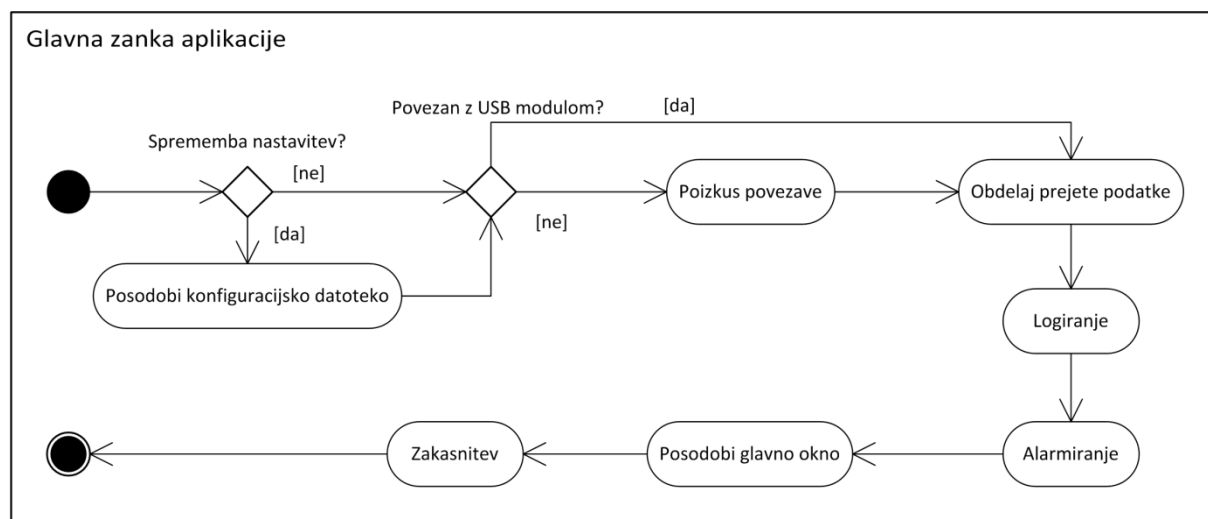


Slika 7.4: Diagram aktivnosti programa aplikacije.

Ob zagonu aplikacije prva nit prebere konfiguracijsko datoteko, zažene drugo nit in odpre glavno okno aplikacije ter izvaja aplikacijsko sporočilno zanko (angl. application

message loop). Ta zanka nato čaka na posamezne dogodke iz uporabniškega vmesnika, kot je na primer klik uporabnika na gumb. Ob vsakem dogodku se zažene ustrezen upravljalca dogodka (angl. event handler), kjer se izvede sekvenca ukazov. Ko uporabnik zapre glavno okno, se izvede funkcija, ki glavni zanki sporoči, da naj zaključi s svojimi aktivnostmi. Ko sta obe niti zaključili z delom, se ponovno združita in program se preneha izvajati.

Druga nit poganja glavno zanko aplikacije, katere diagram aktivnosti je prikazan na sliki 7.5. Zanka vedno najprej preveri če so se nastavitve spremenile in če je potrebno posodobi konfiguracijsko datoteko. Če aplikacija še ni povezana z USB modulom preko serijskega porta, se izvede poizkus povezave. Temu sledi branje in obdelava prejetih podatkov, logiranje in alarmiranje. Ob vsakem prejemu novih podatkov se izvede tudi posodobitev vrednosti na glavnem oknu. Temu sledi zakasnitev, ki preprečuje prepogosto izvajanje glavne zanke.



Slika 7.5: Diagram aktivnosti glavne zanke aplikacije.

7.2.2 Shranjevanje zgodovine

Za shranjevanje zgodovine prejetih vrednosti (logiranje) se odločimo za datoteke formata CSV (angl. Comma-Separated Values). CSV datoteke so priročen način shranjevanja vrstic podatkov z enakim številom elementov v vsaki vrstici, saj so elementi preprosto ločeni z vejico ali podpičjem. Ta format se pogosto uporablja za uvoz in izvoz podatkov iz aplikacije. S tem omogočimo uporabniku, da lažje uvozi te podatke v druge aplikacije. Podatki se shranjujejo v datoteke z imenom »log_X.csv«, kjer je X datum v obliki zaporedja vrednosti: leto, mesec, dan. To omogoča preprosto razvrščanje datotek po

datumu v direktoriju, saj se za vsak dan ustvari nova datoteka. Zgradba enega zapisa v datoteki je sledeča:

Čas; ID; Ime; T1; T2; T3; T4; H1; H2; H3; H4; C; LB

Elementi v zapisu so:

- Čas – Čas zapisa,
- ID – Identifikacijska številka IZP-ja,
- Ime – Ime IZP-ja, ki je trenutno dodeljeno tej ID številki v nastavitvah aplikacije,
- T1,...,T4 – Izmerjene vrednosti temperature,
- H1,...,H4 – Izmerjene vrednosti vlage,
- C – Vgrajeni 8 bitni (0..255) ciklični števec poslanih paketov za diagnostiko ter
- LB – Zastavica za sporočanje nizke napetosti na bateriji.

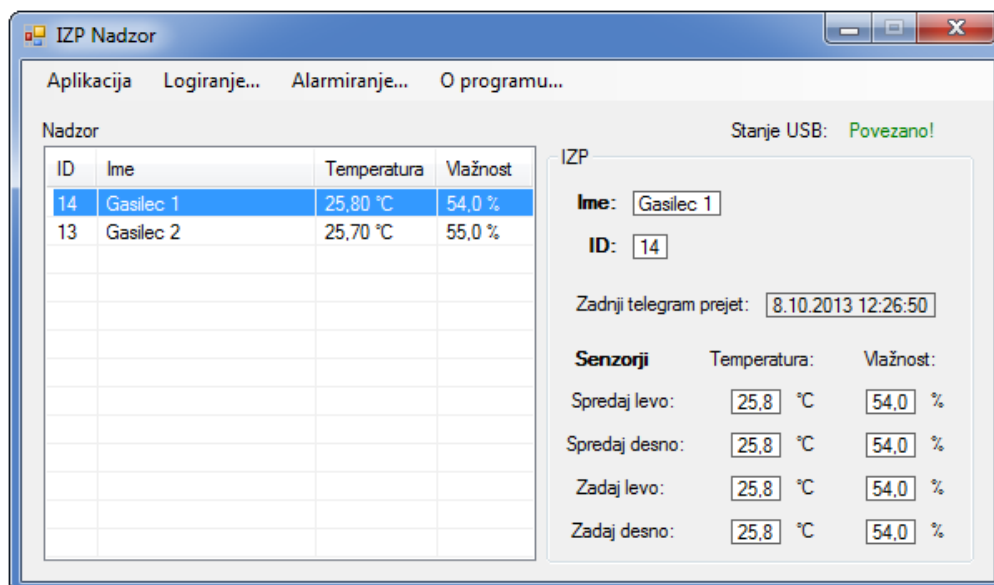
7.2.3 Alarmiranje

Za funkcijo alarmiranja sprogramiramo mejo opozorila in mejo alarma. Ko neka vrednost doseže mejo opozorila, se uporabnika aplikacije opozori, da se dana vrednost nevarno bliža meji alarma. Če med aktivno uporabo vrednost preseže mejo alarma, je uporabnik IZP-ja v nevarnosti in je potrebno uporabnika aplikacije na to ustrezno opozoriti.

7.3 Uporabniški vmesnik

Želimo, da je aplikacija zgrajena tako, da uporabnik ne potrebuje priročnika za uporabo. Njen uporabniški vmesnik mora biti preprost in intuitiven za uporabo. Hkrati želimo, da je uporabniku oteženo, da bi pri uporabi aplikacije storil hujšo napako.

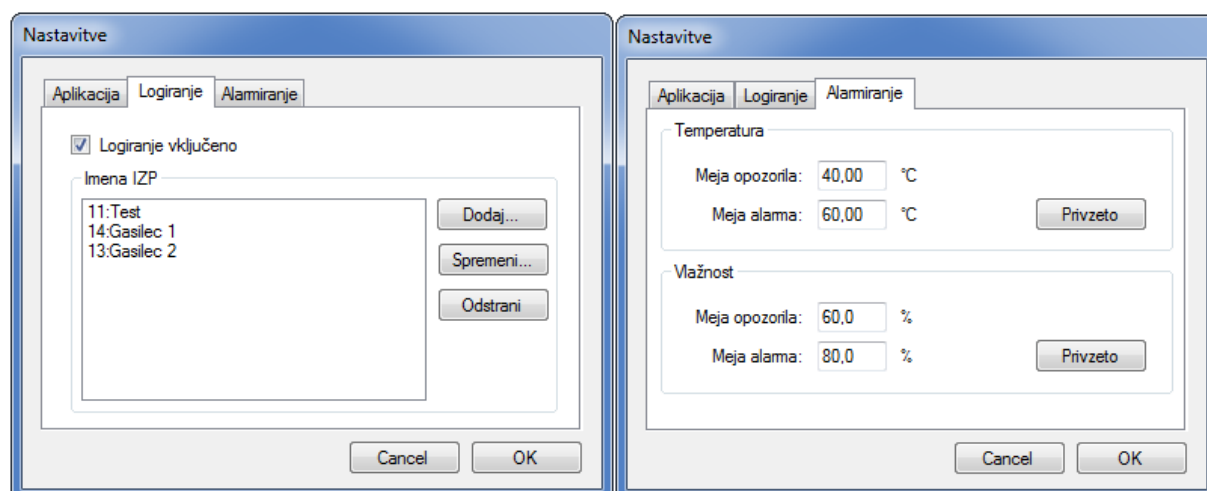
Na sliki 7.6 je prikazano glavno okno aplikacije med obratovanjem sistema. Na levi strani vmesnika se nahaja tabela, ki vsebuje vsa vozlišča IZP ter podatke o trenutni povprečni temperaturi in vlažnosti vozlišča. Ko izberemo eno izmed vozlišč, se na desni strani prikažejo bolj natančni podatki o vrednosti s senzorjev ter čas prejetja zadnjega paketa podatkov od izbranega vozlišča. Desno zgoraj je prikazano stanje komunikacije z USB modulom, ki uporabniku sporoči ali je sistem pravilno nastavljen.



Slika 7.6: Glavno okno aplikacije IZP Nadzor.

Kadar posamezna vrednost na senzorju prekorači mejo opozorila ali alarma, se okvirček obarva oranžno oz. rdeče. Če povprečna vrednost preseže mejo, se obarva celotna vrstica v tabeli.

Okno za nastavitve vsebuje tri strani: aplikacija, logiranje in alarmiranje. Prva stran okna vsebuje le eno nastavitvev za spremembo serijskega porta komunikacije z USB modulom. Druga in tretja stran okna sta prikazana na sliki 7.7 in vsebujeta tiste nastavitve, ki so bile opisane pri izdelavi koncepta. Stran za nastavitve logiranja omogoča izklop logiranja ter dodelitev imena posameznim IZP-jem. Na strani za alarmiranje nastavimo meje opozorila in alarma in jih vrnemo na privzete vrednosti.



Slika 7.7: Druga in tretja stran okna za nastavitve.

7.4 Zgradba programske kode

Visual C# je splošno-namenski objektno naravnan programski jezik, ki ga je razvilo podjetje Microsoft v sklopu .NET ogrodja. Programska koda je organizirana v izvirne dokumente s končnico .cs, pri čemer uporabljamo dve vrsti dokumentov: razred (angl. class) in Windows okno (angl. Windows form). Razredi aplikacije IZP Nadzor so:

- Program.cs – vsebuje vstopno točko in glavno zanko aplikacije,
- Logging.cs – vsebuje glavne aktivnosti logiranja,
- Log.cs – razred za posamezen vpis pri logiranju,
- SettingsData.cs – vsebuje nastavitve aplikacije ter funkcije za branje in pisanje nastavitvev v XML datoteko,
- IdNamePair.cs – preprost razred, ki povezuje ID številko in ime vozlišča.

Za uporabniški vmesnik uporabimo Windows Forms, kljub temu, da je Windows Presentation Foundation naprednejši sistem za grafične vmesnike Windows aplikacij. Windows Forms je starejši in enostavnejši sistem oken, ki je zaradi enostavnosti aplikacije bolj primeren. Aplikacija vsebuje sledeča Windows okna:

- MainForm.cs – glavno okno aplikacije, prikazano na sliki 7.6,
- SettingsForm.cs – okno za nastavitve aplikacije, prikazano na sliki 7.7,
- IdNamePairChange.cs – okno za spremembo imena IZP in
- AboutBox.cs – okno z informacijami o aplikaciji.

Aplikacija vsebuje tudi InstallShield namestitveno datoteko, ki jo prav tako izdelamo v okolju Visual Studio 2012.

V nadaljevanju so na kratko opisane funkcije posameznih razredov. Razred Program vsebuje definicije sledečih funkcij:

- Main() – vstopna točka in prva nit aplikacije,
- ThreadMain() – druga nit aplikacije (glavna zanka),
- serialPort_ErrorReceived() – upravljalca dogodka za napako na serijskem portu,
- serialPort_DataReceive() – upravljalca dogodka za prejem podatkov.

Razred Logging vsebuje čakalno vrsto za prejete pakete, saj se shranjevanje v datoteko ne izvede takoj ob prejetju posameznega paketa. Razred vsebuje funkciji:

- SubmitLog() – doda en paket podatkov na konec čakalne vrste in
- ProcessLogs() – shrani podatke v čakalni vrsti v datoteko.

Razred `SettingsData` vsebuje spremenljivke za nastavitve programa ter funkciji:

- `ReadSettingsFromFile()` – prebere nastavitve iz XML datoteke in
- `WriteSettingsToFile()` – shrani nastavitve v XML datoteko.

Branje in shranjevanje nastavitev v datoteko izvedemo z razredom `XmlSerializer` iz .NET knjižnice `System.Xml.Serialization`, ki omogoča hitro pretvorbo objektov v in iz XML oblike. Glavna prednost uporabe takšnega orodja je fleksibilnost, saj pri spremembi nastavitev (npr. dodajanje nove nastavitve) ni potrebno spreminjati kode omenjenih dveh funkcij.

Tudi okno `MainForm` je razred, ki vsebuje sledeče pomembnejše funkcije:

- `MainForm()` – konstruktor, ki izvede inicializacijo,
- `UpdateForm()` – osvežuje prikazane podatke na glavnem oknu in
- `OpenSettingsForm()` – odpre okno za nastavitve.

Razred vsebuje še mnogo funkcij, ki so upravljalci dogodkov aktivnosti uporabnika (npr. klik na gumb).

Pri pisanju programske kode posebno pozornost namenjamo sinhronizaciji niti (angl. `thread synchronization`), s čemer preprečujemo napake v delovanju aplikacije zaradi hkratnega pisanja in branja spremenljivk.

8. POGLAVJE

Testiranje in izboljšava sistema

Quality in a service or product is not what you put into it.

It is what the client or customer gets out of it.

- Peter F. Drucker

Že med izdelavo sistema izvajamo testiranje, da sproti preverjamo pravilno delovanje programske kode. Po zaključenem razvoju izvedemo celovito testiranje sistema, z glavnim ciljem odpravljanja napak in preverjanja pravilnega delovanja.

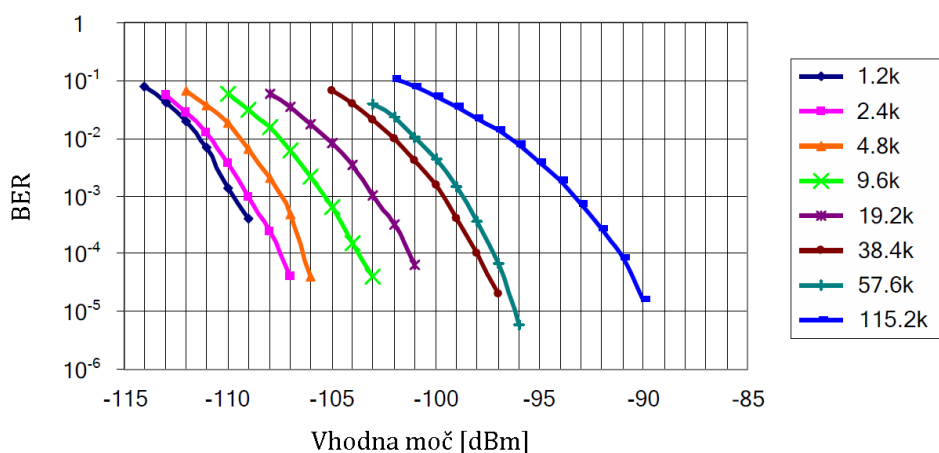
8.1 Meritev dosega brezžične komunikacije

Zaradi značilnosti sistema je doseg brezžične komunikacije vozlišča eden izmed ključnih faktorjev. Z uporabo Friisove enačbe izračunamo največji teoretični doseg izbrane radijske komunikacije:

$$R = \frac{\lambda}{4\pi} 10^{\frac{G_s + G_o + P_o - P_s}{20}} = \frac{0,3454 \text{ m}}{4\pi} 10^{\frac{5,19 \text{ dBi} + 5,19 \text{ dBi} + 5 \text{ dBm} + 99 \text{ dBm}}{20}} = 14.391 \text{ m} \quad (8.1)$$

Pri čemer za G_s in G_o vzamemo vrednost 5,19 dBi, kar je največje teoretično možno ojačenje četrt-valovnega biča napram izotropni anteni, ki velja le pri idealni neskončno veliki ozemljitveni plošči (angl. ground plane). Vrednosti za moč oddajanja P_o in moč pri sprejemniku P_s dobimo v specifikacijah radijskega modula [22, str. 10-11]. Za moč oddajanja vzamemo največjo moč oddajanja pri izbrani frekvenci, kar znaša $P_o = 5 \text{ dBm}$. Za moč pri sprejemniku iz slike 8.1 razberemo občutljivost sprejemnika pri želenem $BER = 10^{-3}$ in izbrani hitrosti komunikacije $BR = 49,26 \text{ kbit/s}$ ter dobimo vrednost $P_s = -99 \text{ dBm}$. Dobljena vrednost $R = 14.391 \text{ m}$ opisuje razdaljo, pri kateri bo poslan bit

informacije uspešno prejet pri sprejemniku z verjetnostjo $p = 1 - 10^{-3}$, pri čemer sta oddajna in sprejemna antena idealni, motilni pojavi zanemarjeni in predpostavljena je medsebojna vidnost v idealnem mediju.



Slika 8.1: BER krivulje radijskega modula RFM12B.

Izkaže se, da enačba dovolj natančno opisuje radijsko komunikacijo le v določenih aplikacijah, kot npr. pri satelitski komunikaciji in v gluhih komorah, ki preprečujejo odboje. Dejanski doseg komunikacije je veliko manjši in ga najlažje ter najhitreje ugotovimo s praktičnimi meritvami.

Prvo serijo meritev izvedemo pri medsebojni vidnosti oddajnika in sprejemnika, pri čemer postopoma večamo razdaljo med njima. Rezultate meritev prikazuje tabela 8.1. Merimo število poslanih paketov, število uspešno prejetih paketov, število napačno prejetih paketov ter število izgubljenih paketov. Izračunamo tudi skupni odstotek uspešno prejetih paketov.

Razdalja	Vseh	Uspešnih	Napačnih	Izgubljenih	Uspešnost
20 m	42	42	0	0	100 %
70 m	40	38	1	1	95 %
160 m	40	36	1	0	90 %
300 m	46	26	11	9	56,5 %
440 m	50	9	15	26	18 %

Tabela 8.1: Rezultati meritev pri medsebojni vidnosti.

Rezultati meritev pri medsebojni vidnosti sami po sebi malo povedo o obnašanju komunikacije znotraj in v okolici zgradb, zato izvedemo še drugo serijo meritev, pri

katerih nas bolj kot razdalja zanima število ovir. Meritve so izvedene v armirano-betonski zgradbi z opečnatimi stenami in so navedene v tabeli 8.2.

Št. sten	Razdalja	Vseh	Uspešnih	Napačnih	Izgubljenih	Uspešnost
1	5 m	45	45	0	0	100 %
2	7 m	55	55	0	0	100 %
3	10 m	69	64	3	2	93 %
4	15 m	61	28	10	23	45,9 %

Tabela 8.2: Rezultati meritev znotraj in v okolici zgradb.

Iz rezultatov meritev ugotovimo, da komunikacija ne ustreza tehničnim zahtevam za doseg brezžične komunikacije iz poglavja 1.4, ki predvidevajo komunikacijo na prostem do 300 m in v zgradbah do 30 m. Kakovost komunikacije pri 300 m je nezadostna že pri medsebojni vidnosti, medtem ko se na prostem pričakujejo dodatne ovire. V zgradbah je komunikacija neustrezna že pri štirih stenah, medtem ko se na razdalji 30 m pričakuje tudi 8 ali več sten ter pri različnih višinah oddajnika in sprejemnika tudi armirano-betonske plošče.

8.2 Zamenjava radijskega modula

Zaradi nezadostnega dosega brezžične komunikacije zamenjamo radijski modul z močnejšo različico RFM12BP, ki zmore do 500 mW izhodne moči. Novo vezje vozlišča IZP je prikazano v prilogi 10.3 na sliki 10.5. Dodali smo tudi analogno stikalo MAX323CPA, ki se uporablja za vklop in izklop ojačenega oddajanja preko izhodnega pina mikrokrmilnika [23].

Naredimo podoben izračun kot za radijski modul RFM12B, pri čemer spremenimo le moč oddajanja na $P_o = 27$ dBm, kar dobimo iz specifikacij modula [24, str. 5].

$$R = \frac{\lambda}{4\pi} 10^{\frac{G_s + G_o + P_o - P_s}{20}} = \frac{0,3454 \text{ m}}{4\pi} 10^{\frac{5,19 \text{ dBi} + 5,19 \text{ dBi} + 27 \text{ dBm} + 99 \text{ dBm}}{20}} = 181 \text{ km} \quad (8.2)$$

Dobljen teoretični doseg je večji za faktor 12,6. Ponovno opravimo meritve in ugotovimo, da znaša uspešnost prejema paketov 100 % pri vseh meritvenih pogojih iz tabel 8.1 in 8.2. S tem vozlišče ustreza zahtevam sistema.

8.3 Izračun življenjske dobe baterije

Za teoretični izračun življenjske dobe brezžičnega vozlišča uporabimo enačbo 3.2, s katero izračunamo porabo energije brezžičnega pošiljanja enega paketa podatkov.

$$E_{\text{tx}} = P_{\text{start}} T_{\text{start}} + \frac{N}{R_C R} (P_{\text{txElec}} + P_{\text{amp}}) = \frac{156}{49\,260} 2,76 \text{ W} = 8,74 \text{ mJ} \quad (8.3)$$

Porabo energije zaradi vklopa radijskega oddajnika ($P_{\text{start}} T_{\text{start}}$) zanemarimo, saj znaša le 155 nJ. Velikost paketa podatkov N znaša 156 bitov (12 bitov – preamble, 16 bitov – sinhronska beseda, 112 bitov – podatki in 16 bitov – CRC). Radijski oddajnik ne uporablja konvolucijskega kodiranja, zato za R_C vzamemo vrednost 1. Skupno moč porabe energije za oddajanje izračunamo iz podatka o vhodnem toku v oddajnik pri dani napajalni napetosti.

$$P_{\text{txElec}} + P_{\text{amp}} = U_{PA} \cdot I_{\text{tx}} = 12 \text{ V} \cdot 230 \text{ mA} = 2,76 \text{ W} \quad (8.4)$$

Kapaciteta baterije znaša 1500 mAh, kar je pri napetosti 12 V enakovredno 64,8 kJ energije. Če zanemarimo izgube zaradi procesiranja in ostalih elementov na IZP vozlišču, lahko dobimo življenjsko dobo vozlišča pri pošiljanju podatkov na vsakih 5 sekund.

$$t_{\text{bat}} = \frac{64,8 \text{ kJ}}{8,74 \text{ mJ}} \cdot 5 \text{ s} = 37 \cdot 10^6 \text{ s} = 429 \text{ dni} \quad (8.5)$$

Realno takšne življenjske dobe ne moremo pričakovati, saj smo zanemarili veliko pomembnih elementov, katerim težje določimo porabo (npr. mikrokrmilnik). Prav tako ima prototipno vezje nameščeno tudi LED diodo, ki sveti skoraj neprenehoma in s tem bistveno skrajša življenjsko dobo vozlišča.

8.4 Testiranje sistema

S testiranjem želimo ugotoviti ali sistem ustreza funkcionalnim in tehničnim zahtevam ter deluje kot pričakovano. Poiskati želimo morebitne napake v programski opremi in preveriti ustreznost izbrane strojne opreme. Testiranje aplikacije IZP Nadzor in programske opreme na IZP vozliščih ter USB modulu smo izvajali že med razvojem. Odkritih je bilo več manjših pomanjkljivosti in napak, ki smo jih ustrezno odpravili.

Na sliki 8.2 je prikazan prototip IZP-ja, ki smo ga izdelali v namen testiranja. Vezje je zaščiteno v plastičnem valju. Na ramenskem delu je razviden svetlobni pozivnik, pod njim pa se nahaja vibracijski pozivnik. Štirje senzorji temperature in vlage so razporejeni

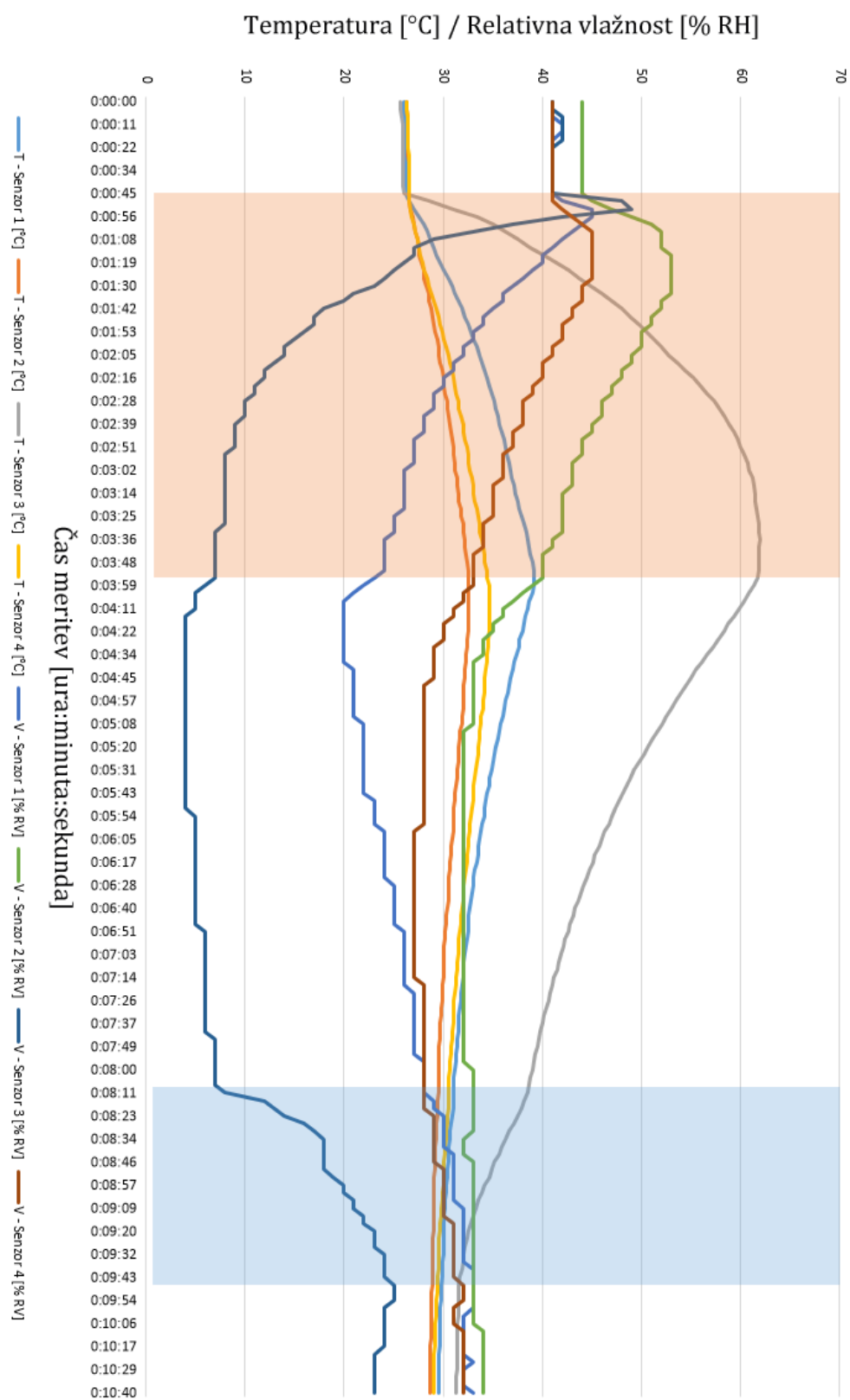
po pregrinjalu. Prototipno pregrinjalo še ne vsebuje temperaturno odpornih zaščitnih materialov, saj je prvotni namen testiranja ugotoviti pravilno delovanje BSO.

Testiranje celotnega sistema izvajamo s prototipom IZP iz slike 8.2 in osebnim računalnikom, ki ima priključen USB modul. Zanima nas pravilno delovanje sistema pod različnimi pogoji, zato testiramo sistem v tako v sončnem kot tudi v deževnem vremenu, kjer je opazna večja vrednost izmerjene relativne vlažnosti. Pravilno delovanje aplikacije IZP Nadzor preverimo na treh računalnikih z različnimi operacijskimi sistemi: Windows XP, Windows 7 in Windows 8.1. Sistem uspešno prestane vse teste brez napak v delovanju. S prototipom ponovimo tudi test dosega brezžične komunikacije, pri čemer ponovno dobimo rezultate, ki ustrezajo zahtevam.



Slika 8.2: Prototip Inteligentnega Zaščitnega Pregrinjala.

Opravimo tudi test segrevanja enega senzorja s sušilnikom za lase. Zanimata nas predvsem odzivnost senzorja ter pravilen vklop svetlobnega in vibracijskega pozivnika. Sušilnik za lase postavimo 15 cm nad 3. senzor na prototipu IZP. Slika 8.3 prikazuje meritve temperature in vlage na vseh štirih senzorjih. V času od 0:00:45 do 0:03:55 je bil sušilnik prižgan z največjo stopnjo gretja, kar je na sliki označeno s svetlo rdečo barvo. Po dobrih 4 minutah ohlajanja vklopimo sušilnik v načinu brez gretja (samo pihanje), kar pospeši ohlajanje proti sobni temperaturi in poveča vlažnost materiala in zraka v okolici senzorja. Pihanje je na grafu označeno s svetlo modro barvo.



Slika 8.3: Meritve senzorjev pri testiranju s sušilnikom za lase.

Iz grafa na sliki 8.3 je razvidno, da se pri segrevanju s sušilnikom merjena temperatura ustali pri 62 °C. Začeten porast relativne vlažnosti pripišemo temu, da ima toplejši zrak višjo vsebnost vodne pare. Kmalu pa začne segret material v okolici senzorja navzemati vodno paro iz zraka, kar osuši zrak in zmanjša izmerjeno relativno vlažnost. Ob pihanju zraka iz okolice se vlažnost ponovno poveča. Opazen je tudi vpliv segrevanja na druge senzorje, ki imajo podobne vendar manj intenzivne odzive. Pri koncu testiranja je razvidno, da se je zaradi segrevanja temperatura pregrinjala povečala na približno 30 °C, relativna vlažnost znotraj pregrinjala pa je padla na približno 30 % RV. Pri izmerjeni temperaturi nad 50 °C na 3. senzorju sta se pravilno vklopila svetlobni in vibracijski pozivnik. Iz grafa je razvidna tudi odzivnost senzorjev na spremembe, ki je dovolj hitra in ustreza zahtevam. Test segrevanja ponovimo še na ostalih senzorjih in dobimo podobne rezultate.

Nazadnje izvedemo še test življenjske dobe vozlišča. Baterijo napolnimo, vklopimo vozlišče in spremljamo delovanje preko aplikacije IZP Nadzor. Iz zapisov v CSV datotekah ugotovimo, da je baterija zdržala 184 ur, kar ustreza zahtevam. Zaradi dolge življenjske dobe vozlišč bo potrebno redkeje polniti baterije, manjša pa je tudi verjetnost, da bi se baterija izpraznila, medtem ko je pregrinjalo v uporabi.

9. POGLAVJE

Zaključek

Na začetku dela so bila širše predstavljena brezžična senzorska omrežja, s poudarkom na tistih konceptih, ki so uporabni na praktični aplikaciji. Drugo poglavje je služilo kot uvod v področje BSO, pri čemer smo predstavili najpomembnejše skupne imenovalce različnih vrst BSO. V tretjem poglavju so bili predstavljeni nekateri splošni pristopi k zasnovi BSO. Natančneje smo si pogledali energijsko učinkovitost in doseg brezžične komunikacije, ki sta za naš primer bila najpomembnejša faktorja.

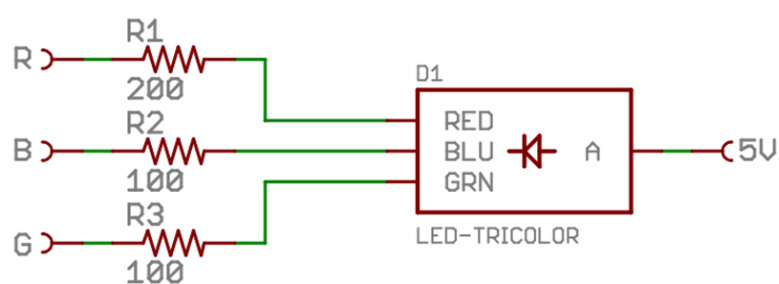
V nadaljevanju je bila opisana izdelava BSO, kjer smo pri razvoju uporabili nekatere ideje iz prejšnjih poglavij. Po predstavljenih zahtevah sistema smo izdelali koncept in na tej podlagi izbrali primerno strojno opremo. Temu sta sledila pisanje programske opreme in izdelava računalniške aplikacije. Pri testiranju smo ugotovili, da je doseg brezžične komunikacije nezadosten. Sistem smo izboljšali z močnejšimi radijskimi moduli in ponovno opravili teste. Ugotovili smo, da sistem deluje po zahtevah.

Če bi želeli dodatno izboljšati BSO, bi uporabili dvosmerno komunikacijo in uvedli temu primeren komunikacijski protokol. Dodali bi možnost nastavljanja določenih parametrov iz računalniške aplikacije, kot npr. temperaturna meja alarma na IZP in perioda pošiljanja meritev. Za dodatno optimizacijo porabe energije bi sprogramirali vozlišča tako, da redkeje pošiljajo podatke, kadar je temperatura na senzorjih v normalnem razponu.

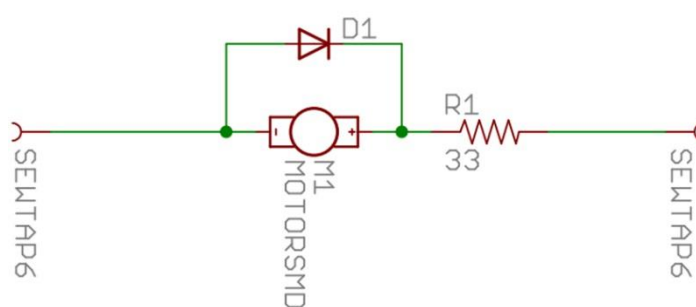
10. POGLAVJE

Priloge

10.1 Električni shemi vezij pozivnikov LilyPad

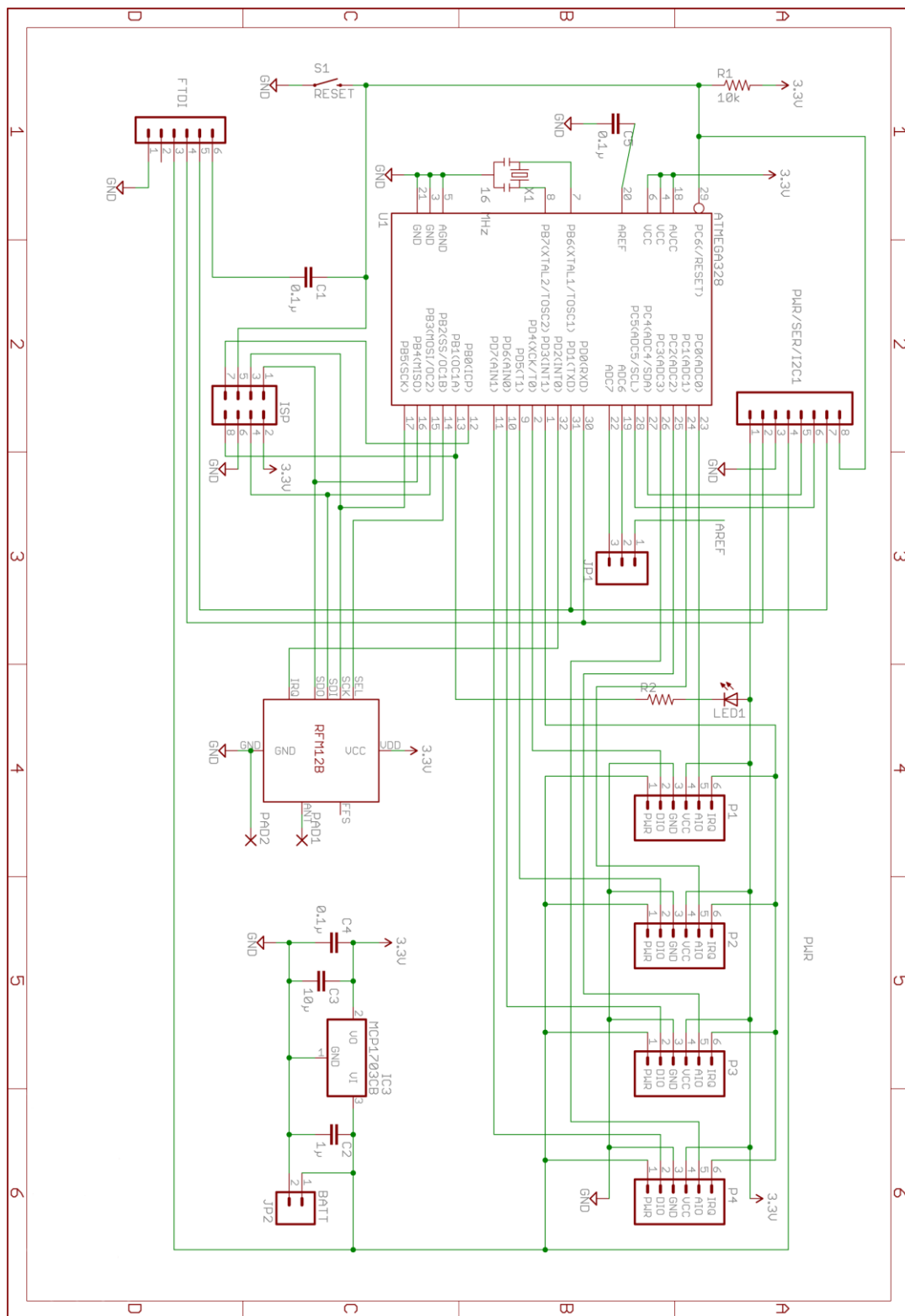


Slika 10.1: Električna shema vezja pozivnika LilyPad Tri-Color LED.

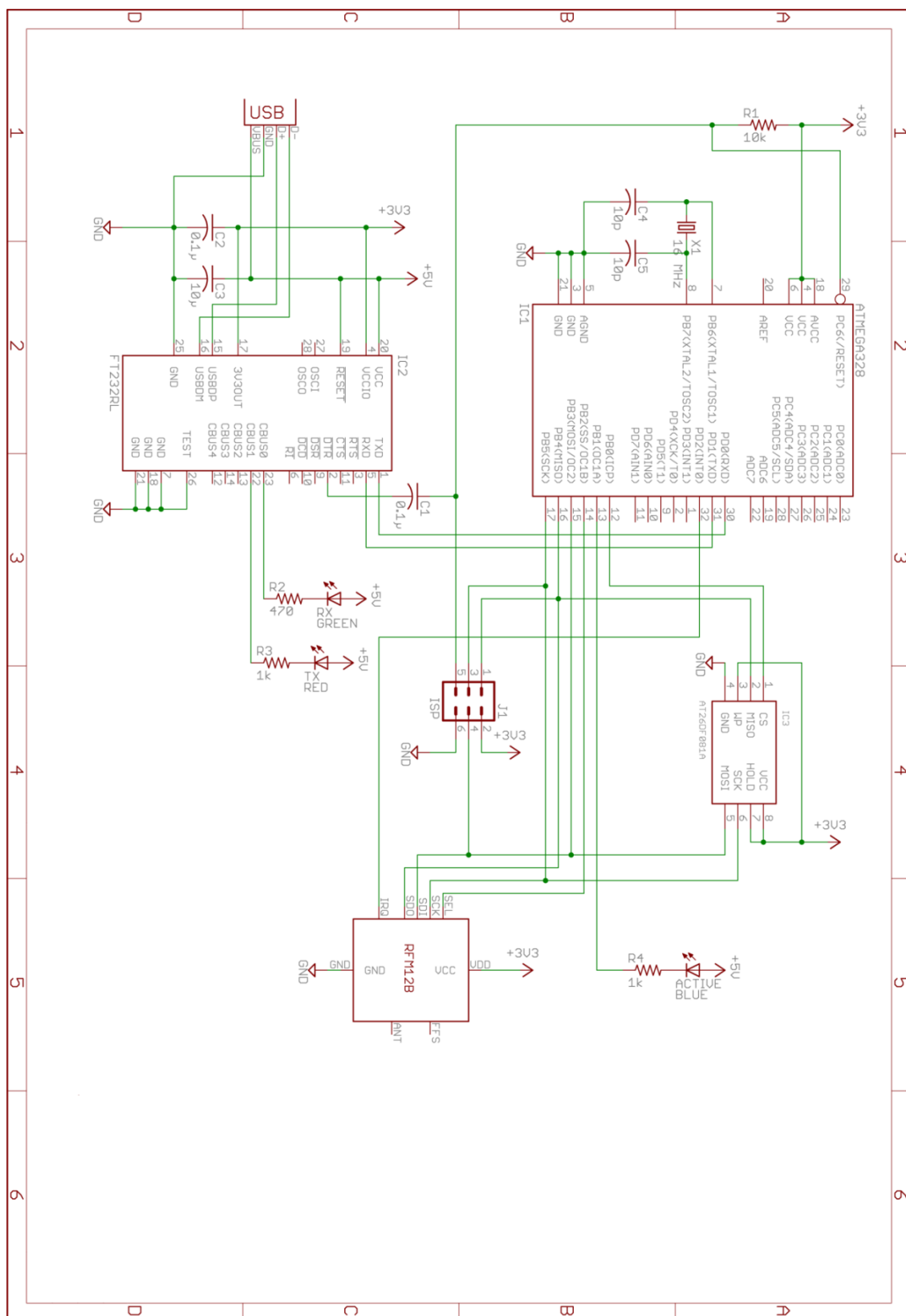


Slika 10.2: Električna shema vezja pozivnika LilyPad Vibe Board.

10.2 Električni shemi vezij brezžičnih vozlišč JeeLabs

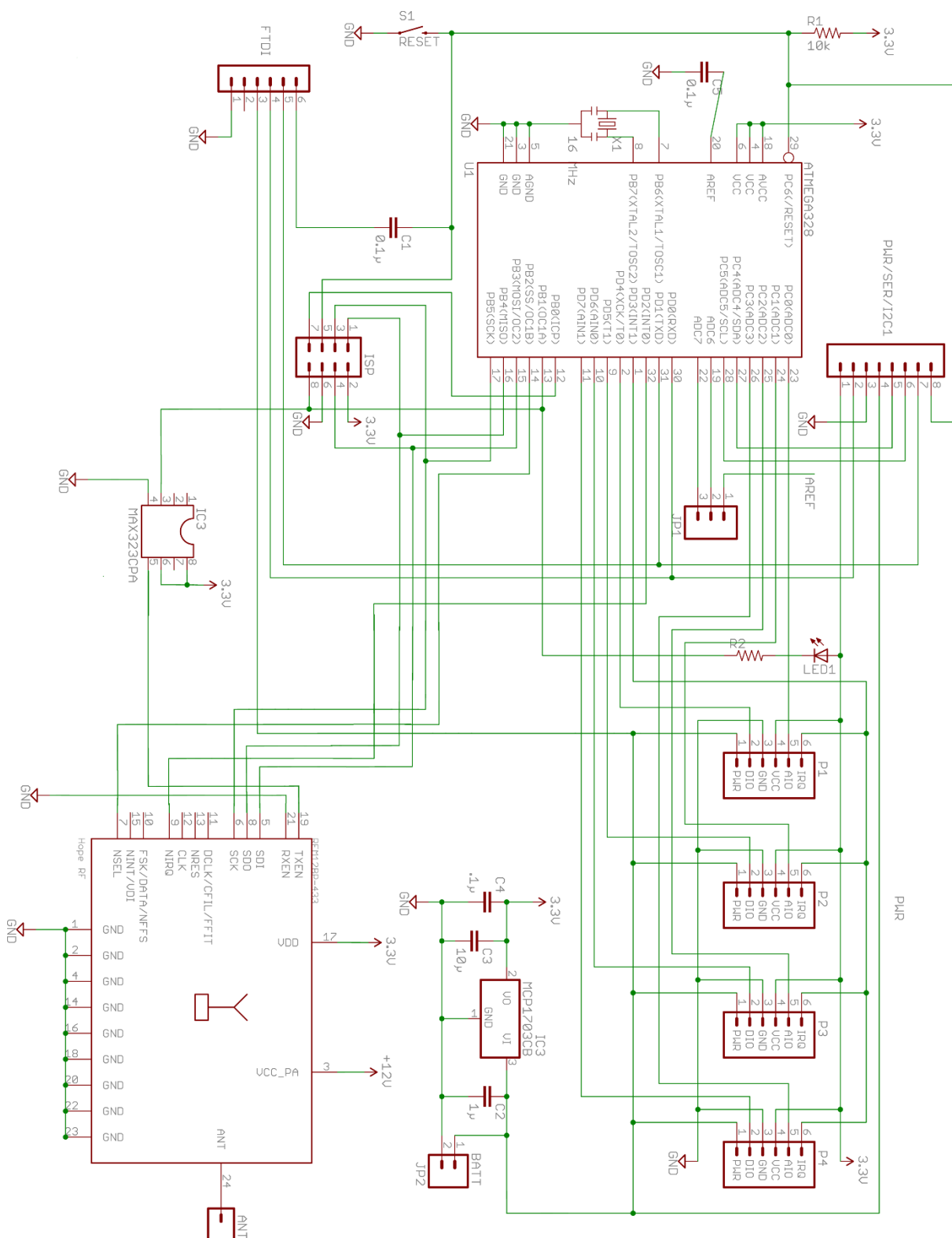


Slika 10.3: Električna shema vezja brezžičnega vozlišča JeeNode SMD.



Slika 10.4: Električna shema vezja USB brezžičnega modula JeeLink.

10.3 Električna shema vezja izboljšanega vozlišča



Slika 10.5: Električna shema izboljšanega vezja vozlišča IZP.

10.4 Seznam slik

Slika 2.1: Primer značilnega brezžičnega senzorskega omrežja.	10
Slika 2.2: Glavne komponente preprostega senzorskega vozlišča.	11
Slika 2.3: Tipičen protokolni sklad BSO.	14
Slika 2.4: Brokaw bandgap referenčno vezje.	21
Slika 3.1: Nekatere antene in njihovi sevalni diagrami.	31
Slika 4.1: Konceptualna zgradba sistema ter struktura komunikacije.	34
Slika 4.2: Elementi strojne opreme IZP-ja in povezave med njimi.	34
Slika 4.3: Lokacije elementov na IZP.	35
Slika 4.4: Glavni elementi osebnega računalnika s centralnim vozliščem in aplikacijo.	36
Slika 4.5: Diagram aktivnosti delovne zanke vozlišča IZP.	36
Slika 4.6: Diagram aktivnosti delovne zanke centralnega vozlišča.	37
Slika 4.7: Diagram uporabe aplikacije IZP Nadzor.	38
Slika 4.8: Diagram aktivnosti glavne delovne zanke aplikacije.	38
Slika 5.1: Brezžično senzorsko vozlišče JeeNode SMD.	40
Slika 5.2: Brezžični USB modul JeeLink.	40
Slika 5.3: Modul za USB komunikacijo USB BUB-II.	41
Slika 5.4: Normalno območje delovanja obeh senzorjev.	43
Slika 5.5: Največja natančnost meritev temperature pri različnih temperaturah.	44
Slika 5.6: Največja natančnost meritev vlage pri različnih vlažnostih (pri 25 °C).	44
Slika 5.7: Svetlobni pozivnik LilyPad Tri-Color LED.	46
Slika 5.8: Vibracijski pozivnik LilyPad Vibe.	47
Slika 6.1: Arduino razvojno okolje v operacijskem sistemu Windows 7.	50
Slika 6.2: Struktura knjižnic v programski opremi.	50
Slika 6.3: Diagram aktivnosti funkcije za postavitve naprave – setup().	52
Slika 6.4: Diagram aktivnosti funkcije za neprekinjeno izvajanje – loop().	52
Slika 6.5: Diagram aktivnosti funkcije za izvedbo meritev – doMeasure().	53
Slika 6.6: Diagram aktivnosti funkcije za izvedbo poročila – doReport().	54
Slika 6.7: Zgradba paketov, ki jih pošilja IZP brezžično vozlišče.	54
Slika 6.8: Diagram aktivnosti funkcije za postavitve USB modula – setup().	56
Slika 6.9: Diagram aktivnosti delovne zanke USB modula – loop().	56

Slika 7.1: Diagram uporabe pri scenariju spreminjanja nastavitev aplikacije.	58
Slika 7.2: Diagram uporabe pri scenariju testiranja aplikacije.....	59
Slika 7.3: Diagram uporabe pri scenariju aktivne/pasivne uporabe.	60
Slika 7.4: Diagram aktivnosti programa aplikacije.	61
Slika 7.5: Diagram aktivnosti glavne zanke aplikacije.....	62
Slika 7.6: Glavno okno aplikacije IZP Nadzor.	64
Slika 7.7: Druga in tretja stran okna za nastavitve.....	64
Slika 8.1: BER krivulje radijskega modula RFM12B.	68
Slika 8.2: Prototip Inteligentnega Zaščitnega Pregrinjala.	71
Slika 8.3: Meritve senzorjev pri testiranju s sušilnikom za lase.	72
Slika 10.1: Električna shema vezja pozivnika LilyPad Tri-Color LED.	77
Slika 10.2: Električna shema vezja pozivnika LilyPad Vibe Board.	77
Slika 10.3: Električna shema vezja brezžičnega vozlišča JeeNode SMD.	78
Slika 10.4: Električna shema vezja USB brezžičnega modula JeeLink.....	79
Slika 10.5: Električna shema izboljšanega vezja vozlišča IZP.	80

10.5 Seznam tabel

Tabela 2.1: Seznam nekaterih fizikalnih veličin in principov odjemanja za senzorje.	19
Tabela 5.1: Primerjava parametrov obeh senzorjev.	43
Tabela 5.2: Koeficient za izračun temperature pri različnih napajalnih napetostih.....	45
Tabela 5.3: Koeficient za izračun temperature pri različnih natančnostih meritve.	45
Tabela 5.4: Koeficienti za nelinearno kompenzacijo relativne vlažnosti.	46
Tabela 5.5: Koeficienta za temperaturno kompenzacijo relativne vlažnosti.....	46
Tabela 8.1: Rezultati meritev pri medsebojni vidnosti.	68
Tabela 8.2: Rezultati meritev znotraj in v okolici zgradb.....	69

11. POGLAVJE

Viri in literatura

- [1] X. Dong et al, Autonomous Precision Agriculture Through Integration of Wireless Underground Sensor Networks with Center Pivot Irrigation Systems, University of Nebraska-Lincoln, Lincoln, 2012
- [2] A. Cammarano et al, Energy-harvesting WSNs for structural health monitoring of underground train tunnels, Sapienza University of Rome, 2013
- [3] S. Wang et al, Wireless Sensor Networks for Underwater Localization: A Survey, University of Essex, Colchester, UK, 2012
- [4] E. P. de Freitas, Cooperative Context-Aware Setup and Performance of Surveillance Missions Using Static and Mobile Wireless Sensor Networks, Halmstad University, Halmstad, 2011
- [5] G. Simon et al, Sensor Network-Based Countersniper System, Vanderbilt University, Nashville, Tennessee, 2004
- [6] Wireless Sensor Networks (WSN) 2012-2022: Forecasts, Technologies, Players, December 2013 (<http://www.idtechex.com/research/reports/wireless-sensor-networks-wsn-2012-2022-forecasts-technologies-players-000314.asp>)
- [7] Industrial Wireless Sensor Networks (IWSN) Market - Global Forecast And Analysis (2012 - 2017): By Technology, Components, Applications, Geography (<http://www.transparencymarketresearch.com/industrial-wireless-sensor-networks-iwsn-market.html>)
- [8] K. Sohraby et al, Wireless sensor networks: technology, protocols, and applications, John Wiley & Sons, Inc., 2007
- [9] H. Karl et al, Protocols and architectures for wireless sensor networks, John Wiley & Sons, Inc., 2005
- [10] Classification of Cells and Batteries (<http://depts.washington.edu/matseed/batteries/MSE/classification.html>)

- [11] M. Ilyas et al, Handbook of sensor networks: compact wireless and wired sensing systems, CRC Press LLC, 2005
- [12] I.F. Akyildiz et al, Wireless sensor networks: a survey, Georgia Institute of Technology, Atlanta, Georgia, 2001
- [13] F. L. Lewis, Wireless Sensor Networks, The University of Texas at Arlington, Ft.Worth, Texas, 2004
- [14] J. Fraden, Handbook of Modern Sensors, Physics, Designs and Applications, Third Edition, Springer-Verlag New York Inc., 2004
- [15] A. P. Brokaw, A Simple Three-Terminal IC Bandgap Reference, IEEE Journal of Solid-State Circuits, Vol. SC-9, No.6, December 1974
- [16] R. Min, A. Chandrakasan, A Framework for Energy-Scalable Communication in High-Density Wireless Networks, Massachusetts Institute of Technology, Cambridge, Massachussets, 2002
- [17] J. Donovan, Selecting Antennas for Embedded Designs, Low-Power Design for Convergence Promotions LLC, 2012 (<http://www.digikey.com/us/en/techzone/wireless/resources/articles/selecting-antennas-for-embedded-designs.html>)
- [18] Friis Equation – (aka Friis Transmission Formula) (<http://www.antenna-theory.com/basics/friis.php>)
- [19] Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash, Atmel, 2012 (<http://www.atmel.com/Images/doc8161.pdf>)
- [20] Digital Humidity Sensor HYT-131, B+B Thermo-Technik GmbH (<http://www.hygrosens.com>)
- [21] Datasheet SHT1x (SHT10, SHT11, SHT15), Humidity and Temperature Sensor IC Sensirion AG (<http://www.sensirion.com>)
- [22] RFM12B Universal ISM Band FSK Transceiver, HOPE MICROELECTRONICS CO., LTD, Guangdong, China, 2006 (<http://www.hoperf.com/upload/rf/rfm12b.pdf>)
- [23] Maxim Integrated Products, Maxim Precision, Single-Supply SPST Analog Switches, California, 1997 (<http://datasheets.maximintegrated.com/en/ds/MAX323-MAX325.pdf>)
- [24] Universal ISM Band FSK Transceiver Module With 500 mW Output Power RFM12BP, HOPE MICROELECTRONICS CO., LTD, Guangdong, China, 2006 (<http://www.hoperf.com/upload/rf/rfm12bp.pdf>)

IZJAVA

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

Ljubljana, januar 2014

Matej Lenarčič