



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Владимир Моравчевић

**Решење *ZigBee* програмске подршке на
централном контролеру у складу са *НА*
1.2 спецификацијом**

МАСТЕР РАД

Нови Сад, 2016



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :	
Идентификациони број, ИБР :	
Тип документације, ТД :	Монографска документација
Тип записа, ТЗ :	Текстуални штампани материјал
Врста рада, ВР :	Дипломски – мастер рад
Аутор, АУ :	Владимир Моравчевић
Ментор, МН :	др Иштван Пап, доцент
Наслов рада, НР :	Решење ZigBee програмске подршке на централном контролеру у складу са НА 1.2 спецификацијом
Језик публикације, ЈП :	Српски / латиница
Језик извода, ЈИ :	Српски
Земља публикавања, ЗП :	Република Србија
Уже географско подручје, УГП :	Војводина
Година, ГО :	2016
Издавач, ИЗ :	Ауторски репринт
Место и адреса, МА :	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО : (поглавља/страна/ цитата/табела/слика/графика/прилога)	7 поглавља / 36 стране / 14 цитата / 1 табела / 12 слика / 3 графика
Научна област, НО :	Електротехника и рачунарство
Научна дисциплина, НД :	Рачунарска техника
Предметна одредница/Кључне речи, ПО :	Систем за кућну аутоматизацију, ZigBee, Кластер, Профил, Крајња тачка
УДК	
Чува се, ЧУ :	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН :	
Извод, ИЗ :	Програмску подршку на централном контролеру потребно је проширити са модулом за комуникацију преко ZigBee протокола. Имплементација треба да буде у складу са НА 1.2 спецификацијом. ZigBee програмска подршка треба да задовољи корисничке захтеве по питању одзива, проширивости и робусности механизма. Имплементација мора да прође тестове који потврђују да је у складу са НА 1.2 спецификацијом и да задовољава корисничка очекивања.
Датум прихватања теме, ДП :	
Датум одбране, ДО :	
Чланови комисије, КО :	Председник: др Иван Каштелан, доцент
	Члан: др Милан Лукић, доцент
	Члан, ментор: др Иштван Пап, доцент
	Потпис ментора



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	Monographic publication		
Type of record, TR :	Textual printed material		
Contents code, CC :	Master Thesis		
Author, AU :	Vladimir Moravčević		
Mentor, MN :	PhD Ištvan Pap, docent		
Title, TI :	An Solution for ZigBee software support on central controller device according HA 1.2 specification		
Language of text, LT :	Serbian		
Language of abstract, LA :	Serbian		
Country of publication, CP :	Republic of Serbia		
Locality of publication, LP :	Vojvodina		
Publication year, PY :	2016		
Publisher, PB :	Author's reprint		
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	7 chapters / 36 pages / 14 ref. / 1 tabela / 12 pictures / 3 graphs		
Scientific field, SF :	Electrical Engineering		
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW :	Home automation, ZigBee, Cluster, Profile, Endpoint		
UC			
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N :			
Abstract, AB :	Software support on central controller device should be extended with modules for communication via the ZigBee protocol. Implementation should be in accordance with the HA 1.2 specification. ZigBee software should maximize user experience. The implementation must pass tests that are defined by HA 1.2 specification and confirm customer expectations in terms of responsiveness, scalability and robustness.		
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	PhD Ivan Kaštelan	
	Member:	PhD Milan Lukić	Menthor's sign
	Member, Mentor:	PhD Ištvan Pap	

Zahvalnost

Zahvaljujem se mentoru doc. dr Ištvanu Papu i stručnom saradniku Romanu Pavloviću na svojim savetima tokom izrade master rada.

Svojim kolegama sa projekta i fakulteta se zahvaljujem na podršci i saradnji tokom izrade ovog rada. Zajedno smo izneli ovaj projekat i postavili ga na stabilne noge.

Posebnu zahvalnost dugujem svojoj porodici i prijateljima na razumevanju i podršci kako tokom studija tako i tokom ovog projekta i izrade master rada.

SADRŽAJ

1. Uvod.....	1
2. Teorijske osnove	3
2.1 ZigBee	3
2.1.1 ZigBee stek specifikacija	3
2.1.2 Klaster.....	6
2.1.3 Profil	6
2.2 Opis problema	7
3. Koncept rešenja.....	9
3.1 Arhitektura sistema za kućnu automatizaciju.....	9
3.1.1 Arhitektura programske podrške na centralnom kontoleru	10
3.1.2 Arhitektura upravljačke aplikacije za kontrolu pametne kuće.....	10
3.2 Programska podrška za ZigBee protokol	12
3.2.1 ZigBee magistrala	13
3.2.2 ZigBee uređaj.....	14
3.2.3 Komunikacija između magistale i uređaja.....	16
3.2.4 Klaster modul.....	16
3.3 Provera rešenja	18
4. Programsko rešenje.....	19
4.1 Implementacija programske podrške za ZigBee protokol.....	19
4.1.1 ZigBee magistrala	20
4.1.1.1 Proces za slanje API komandi na ZNP modul	20
4.1.1.2 Proces za prijem API komandi sa ZNP modula	21
4.1.1.3 Proces za obradu API komandi ZNP modula.....	22

4.1.1.4	Proces za inicijalizaciju modula i ponovno slanje komandi.....	22
4.1.1.5	ZigBee komanda.....	23
4.1.1.6	ZigBee deskriptor	23
4.1.2	Klaster.....	24
4.1.3	ZigBee uređaj.....	25
4.1.4	ZigBee generični uređaj.....	27
5.	Testiranje i rezultati	28
5.1	HA 1.2 testiranje	28
5.2	Testiranje performansi.....	30
6.	Zaključak	34
7.	Literatura.....	35

SPISAK SLIKA

Slika 1: Sistem za kućnu automatizaciju.....	2
Slika 2: ZigBee stek[7].....	4
Slika 3: Topologija mreže[7]	5
Slika 4: Klijent/Server model klastera	6
Slika 5: Hijerarhiska organizacija krajnje tačke.....	7
Slika 6: Arhitektura sistema za kućnu automatizaciju	9
Slika 7: Blok diagram organizacije upravljačke aplikacije.....	11
Slika 8: Mesto servisa u arhitekturi programseke podrške centralnog kontrolera	12
Slika 9: Arhitektura programske podrške za ZigBee protokol.....	13
Slika 10: Organizacija ZigBee maglstrale	14
Slika 11: Dijagram nasleđivanja za ZigBee uređaje	15
Slika 12: Organizacija ZigBee uređaja	17
Slika 13: Komunikacija između procesa za slanje i prijem komandi	21
Slika 14: Prikaz inicializacione procedure	22
Slika 15: Obrada korisničkog zahteva.....	26
Slika 16: Topologija testnog okruženja.....	31

SPISAK TABELA

Table 1: Lista urađenih testova	29
---------------------------------------	----

SKRAĆENICE

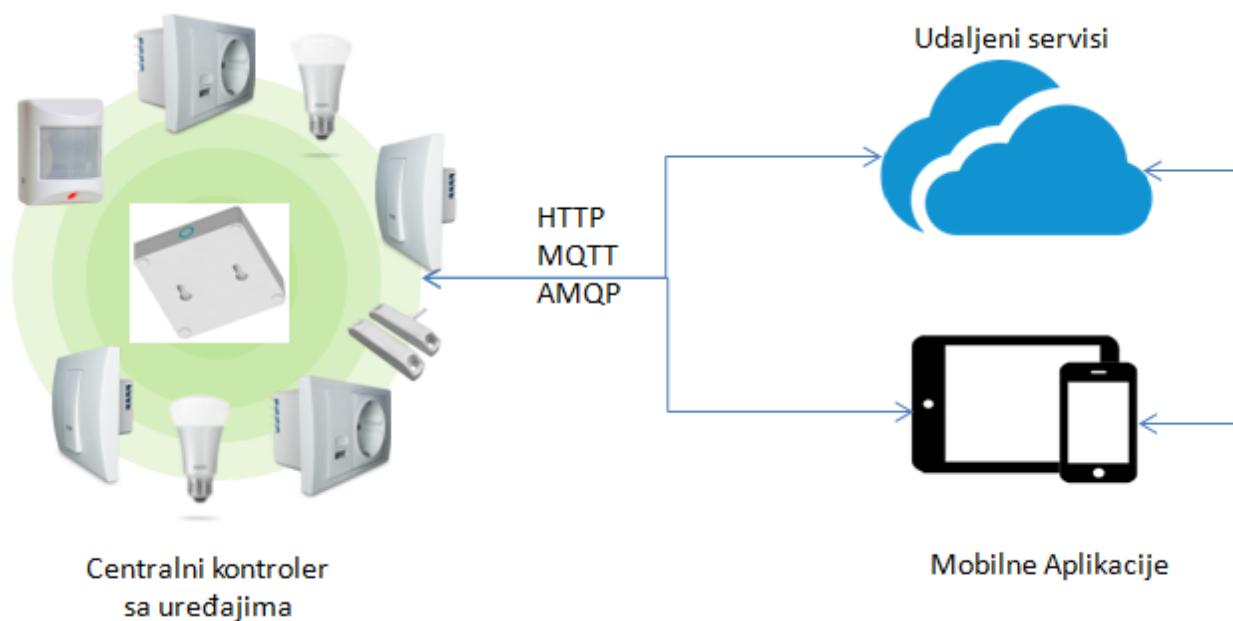
BLE	- <i>Bluetooth Low Energy</i> , Bluetooth male potrošnje
WPAN	- <i>Wireless Personal Area Network</i> , Bežična lična mreža
ISO	- <i>International Organization for Standardization</i> , Internacionalna organizacija za standardizaciju
OSI	- <i>Open Systems Interconnection</i> , Otvoren sistem povezivanja
TCP	- <i>Transmission Control Protocol</i> , Protokol za kontrolu prenosa
IP	- <i>Internet Protocol</i> , Internet protokol
HA	- <i>Home Automation Profile</i> , Profil za kućnu automatizaciju
HTTP	- <i>HyperText Transfer Protocol</i> , Protokol za prenos hiper teksta
MQTT	- <i>Message Queuing Telemetry Transport</i>
AMQP	- <i>Advanced Message Queuing Protocol</i> , Naperdni protokol za redove poruka
XML	- <i>Extensible Markup Language</i> , Proširiv tekst za označavanje
IOT	- <i>Internet Of Things</i> , Internet stvari
API	- <i>Application Programming Interface</i> , Interfejs za programiranje aplikacija
REST	- <i>Representational State Transfer</i> , Reprezentativni prenos stanja
KBPS	- <i>Kilo Byte Per Second</i> , Kilobajta u sekundi
CSMA/CA	- <i>Carrier Sense Multiple Access with Collision Avoidance</i> , Višestruki pristup sa osluškivanjem nosilaca i izbegavanjem kolizije
SOC	- <i>System On Chip</i> , Sistem na čipu
ZNP	- <i>ZigBee Network Processor</i> , ZigBee mrežni procesor

1. Uvod

Napretkom tehnologije smanjene su dimenzije uređaja uz povećanje sposobnosti i smanjenje cene. Ovo je omogućilo da sve više „pametnih“ uređaja bude prisutno u kućama olakšavajući svakodnevne poslove. Povezivanjem takvih uređaja u celinu tj. sistem omogućilo je da se kućni poslovi automatizuju. Oblast u okviru informacionih tehnologija koja se bavi razvojem i projektovanjem ovakvih sistema je poznata pod nazivom internet stvari(eng. Internet of Things). Trenutno je to oblast koja se najviše razvija u domenu informacionih tehnologija sa ciljem da zadovolji sve zahtevnije potrebe korisnika.

U cilju pojednostavljenja korišćenja i olakšavanja ugradnje sistema za kućnu automatizaciju razvijeni su nove tehnologije za komunikaciju između uređaja u ovim sistemima. Tehnologije kao što su KNX ili X10, koje su do sada korišćene i zahtevaju fizičku vezu između uređaja, zamenjuju se novim tehnologijama za bežično povrzivanje uređaja bez intervencija na kućnim instalacijama. Tehnologije koje dominiraju u ovoj oblasti su ZigBee, Zwave i BLE.

Na slici 1 je prikazana arhitektura sistema za kućnu automatizaciju koja se tipično sastoji iz sledećih elemenata: pametnih uređaja, centralnog kontrolera, udaljenih servisa, mobilnih i desktop aplikacija za pristup i konfiguraciju sistema. Komunikacija sa uređajima se postiže korišćenjem protokola niske potrošnje kao što su BLE, Z-wave, ZigBee itd. Pored njih koristi se i IP protocol za komunikaciju sa IP uređajima kao što su kamere. Za komunikaciju sa korisničkim aplikacijama i udaljenim servisima se koriste različiti protokoli visokog nivoa. Najzastupljeniji je HTTP ali i noviji protokoli kao što su MQTT ili AMQP.



Slika 1: Sistem za kućnu automatizaciju

Na tržištu postoji dosta rešenja za kućnu automatizaciju koji imaju ograničen skup krajnjih uređaja kako bi izbegli probleme nekompatibilnosti. U ovom radu ću predstaviti jedno rešenje programske podrške za centralni kontroler koje komunicira preko ZigBee protokola i koje je u skladu sa HA 1.2 specifikacijom. Cilj specifikacije je da centralni kontroler bude kompatibilan sa drugim uređajima usklađenim sa HA 1.2 specifikacijom. Univerzalna podrška daje dodatnu vrednost sistemu iz dva razloga:

1. Proširuje listu podržanih uređaja i
2. Smanjuje trud za dodavanje novih uređaja sa svim njihovim funkcionalnostima

Takođe, pored usklađenosti sa specifikacijom u radu smo obratili pažnju na performanse sistema kao što su odziv sistema, proširivost i otpornost na greške.

U radu će biti objasnjeni osnovni pojmovi ZigBee protokola njegova organizacija koja je korišćenja prilikom izrade ovog rada. Na kraju će biti objašnjeni zahtevi sertifikacije i potvrda usklađenosti testovima koje propisuje HA 1.2 specifikacija.

2. Teorijske osnove

U teorijskim osnovama objašnjeno je funkcionisanje ZigBee protokola, arhitektura sistema za kućnu automatizaciju kao i opis samog problema. Detaljno su objašnjeni pojmovi klastera, krajnje tačke, profila i tipova uređaja. Na tim pojmovima će se zasnovati rešenje problema.

2.1 ZigBee

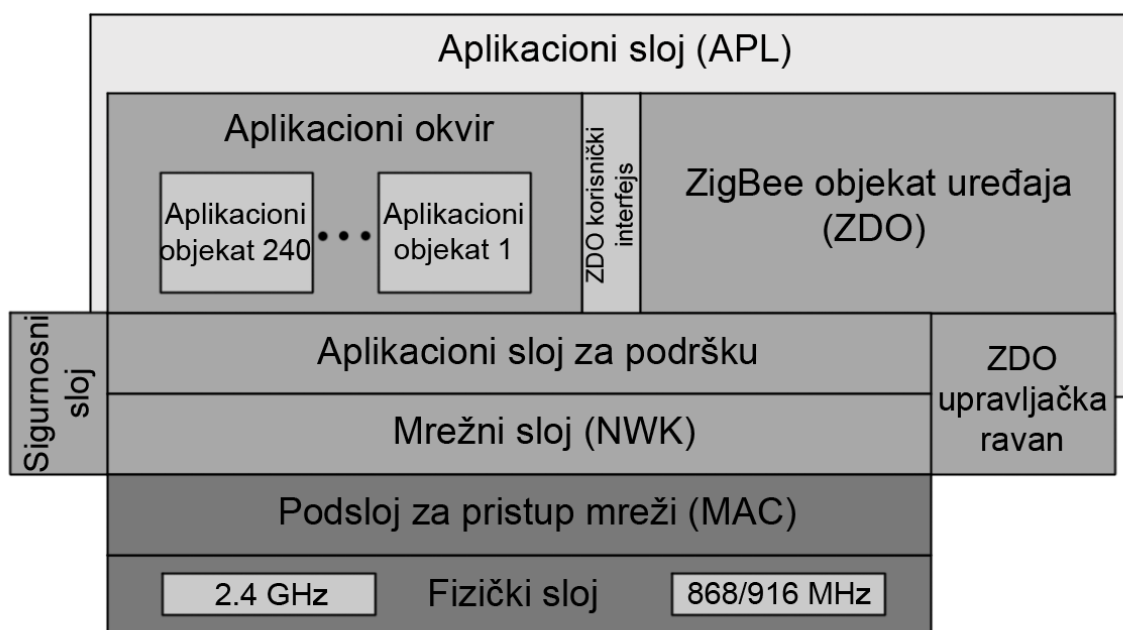
ZigBee[4] je specifikacija za bežičnu komunikaciju malog protoka podataka i male potrošnje. Najzastupljeniji je u oblastima kućne automatizacije, telekomunikacija i automatizaciji zgrada. ZigBee specifikacija se zasniva na 802.15.4 standardu[3] za WPAN[5] mreže sa mrežastom(*eng. mesh*) topologijom. Definisan je od strane udruženja kompanija pod nazivom ZigBee konzorcijum. Specifikacija je definisala aplikativni, sigurnosni i mrežni sloj dok su preostala dva sloja ISO OSI modela definisana 802.15.4 standardom.

ZigBee specifikacija se odlikuje jednostavnošću i fleksibilnošću. Glavne prednosti u odnosu na druge WPAN mreže su:

- Mala potrošnja
- Jednostavno održavanje mreže
- Jednostavno postavljanje mreže
- Niska cena uređaja
- Siguran prenos podataka

2.1.1 ZigBee stek specifikacija

ZigBee stek specifikacija[7] definiše četiri sloja: fizički, sloj pristupa medijumu, mrežni i aplikativni(Slika 2). Standard 802.15.4 definiše fizički i nivo za pristup medijumu dok je ZigBee specifikacija definisala mrežni i aplikativni sloj.



Slika 2: ZigBee stek[7]

Fizički sloj ZigBee protokol stek ima mogućnost slanja signala u tri opsega frekvencija. Radio frekvencijski opsezi su postavljeni na sledeće frekvencijama: 2400 MHz, 868 MHz za Evropu, 915 MHz za Ameriku i Australiju. Brzine prenosa podataka se razlikuju od opsega do opsega i kreću se od 20 do 250 KBPS.

Sloj pristupa medijumu predstavlja spregu između fizičkog i mrežnog sloja. Odgovoran je za pouzdanu komunikaciju između uređaja u mreži koristeći CSMA/CA protokola. Sloj za pristup medijumu formira i dekomponuje ramove podataka koje se šalju i primaju na fizičkom nivou.

Mrežni sloj ima zadatak da upravlja adresiranjem i usmeravanjem poruka u mreži. Na ovom nivou se određuje uloga uređaja koja može da bude: usmerivač, koordinator, i krajnji uređaja. Mrežni sloj dodoljuje mrežnu adresu, dodaje i uklanja uređaje u mreži i vodi računa o putanjama do uređaja. Primena sigurnosnih mehanizama spada u zadatak ovog sloja.

Aplikativni nivo ZigBee steka je slojevit i sastoji se od aplikativnog okvira, u okviru koga se mogu definisati korisničke aplikacije, i ZigBee uređaj objekta koji raspolaže sa informacijama o uređaju u okviru mreže kao i funkcijama za održavanje same mreže. Ova dva pod sloja se oslanjaju na sloj aplikacione podrške koji nudi podršku aplikativnim slojevima sličnu onoj koju pruža TCP sloj aplikativnom nivou u IP mrežama. Razlika između TCP-a i aplikacionog sloja za podršku je u tome što ovaj sloj ne obezbeđuje kontrolu toka kao TCP. Sigurnosni sloj je zadužen za mehanizme šifrovanja i svoje usluge pruža ostalim slojevima.

Kao što je gore napomenuto u okviru ZigBee mreže uređaji imaju različite uloge i po tom osnovu uređaji se dele na tri tipa:

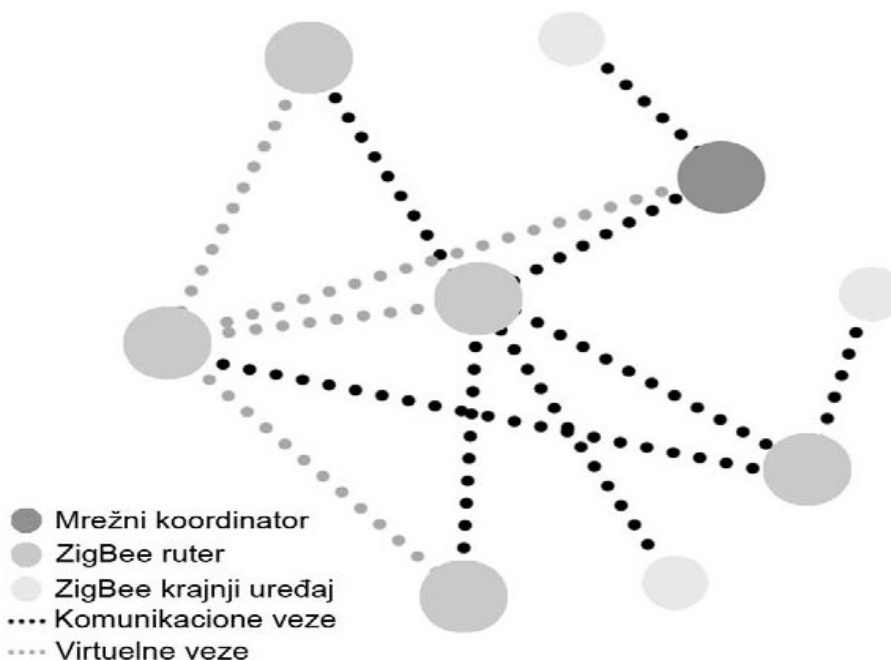
- Koordinator
- Usmerivač(eng. Ruter)
- Krajnji uređaj

Koordinator formira i održava mrežu. Ima ulogu centra od poverenja(eng. Trust Center), skladišti sigurnosne ključeve.

Usmerivač proširuje pokrivenost mreže, formira putanje kako bi isporučio poruku do ciljanog uređaja kao i pronalaženja nove putanje do ciljanog uređaja ako uređaj nije dostupna po staroj. Usmerivač se povezuje ili sa drugim usmerivačem ili sa koordinatorom.

Krajnji uređaji se nalaze na rubu mreže i uglavnom mogu da se pomeraju po mreži. To su uređaji koji su uglavnom bateriski napajani i radi očuvanja iste većinu vremena su u neaktivnom(spavajućem)režimu. Krajnji uređaji primaju poruke preko najbližeg usmerivačkog uređaja na način što se probude i zahtevaju isporuku poruka koje su pristigle dok su bili u režimu spavanja. Krajnji uređaj može da se poveže samo sa koordinatorom ili usmerivačem koji su dužni da čuvaju poruke dok je krajnji uređaj neaktivan. Krajnji uređaj ne može dalje da usmerava poruke kroz mrežu.

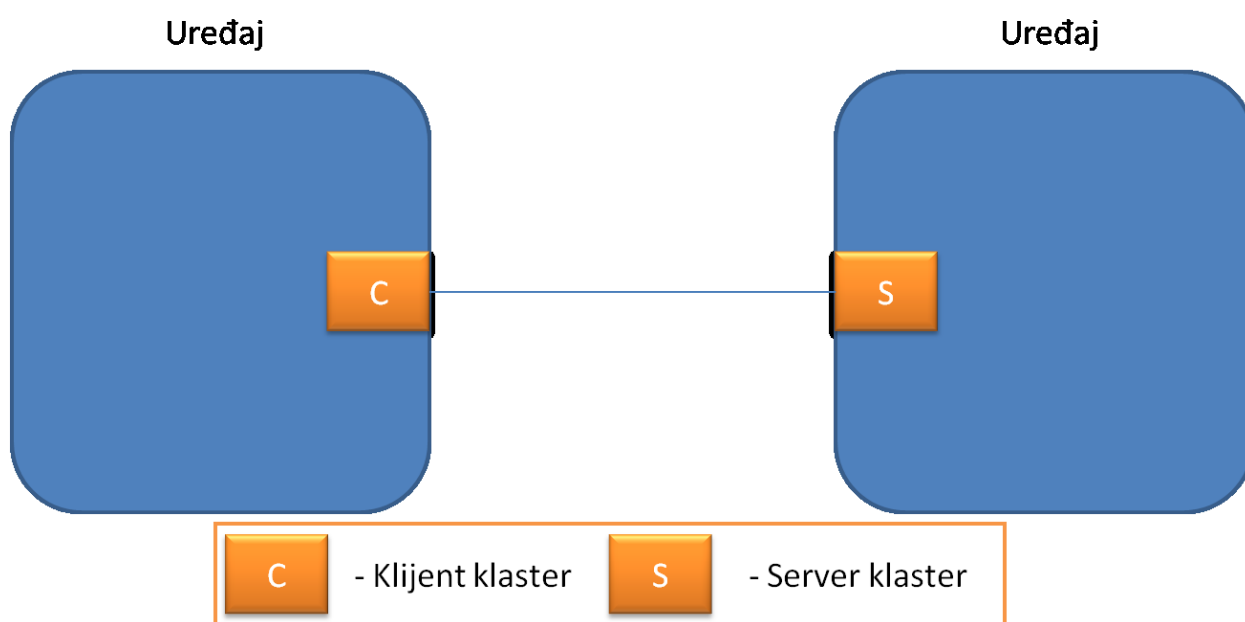
Topologija ZigBee mreže, sa komunikacionim i virtuelnim vezama između uređaja, je prikazana na Slika 3.



Slika 3: Topologija mreže[7]

2.1.2 Klaster

Klaster predstavlja jednu funkcionalnost uređaja kao naprimer temperaturu. ZigBee specifikacija je definisala klaster kao skup atribura i komandi koje zajedno predstavljaju komunikacionu spregu između uređaja. Stanja uređaja su predstavljena kroz attribute dok se komunikacija odvija korišćenjem komandi. Klaster se sastoji od serverske i klijentske strane(slika 4) koje međusobno razmenjuju komande na predefinisani način. Klijentska strana klastera uglavnom podržava samo komande pomoću kojih upravlja stanjima(atributima) na serverskoj strani određenog klastera. Kontrolerski uređaja uglavnom realizuje klijentske delove.



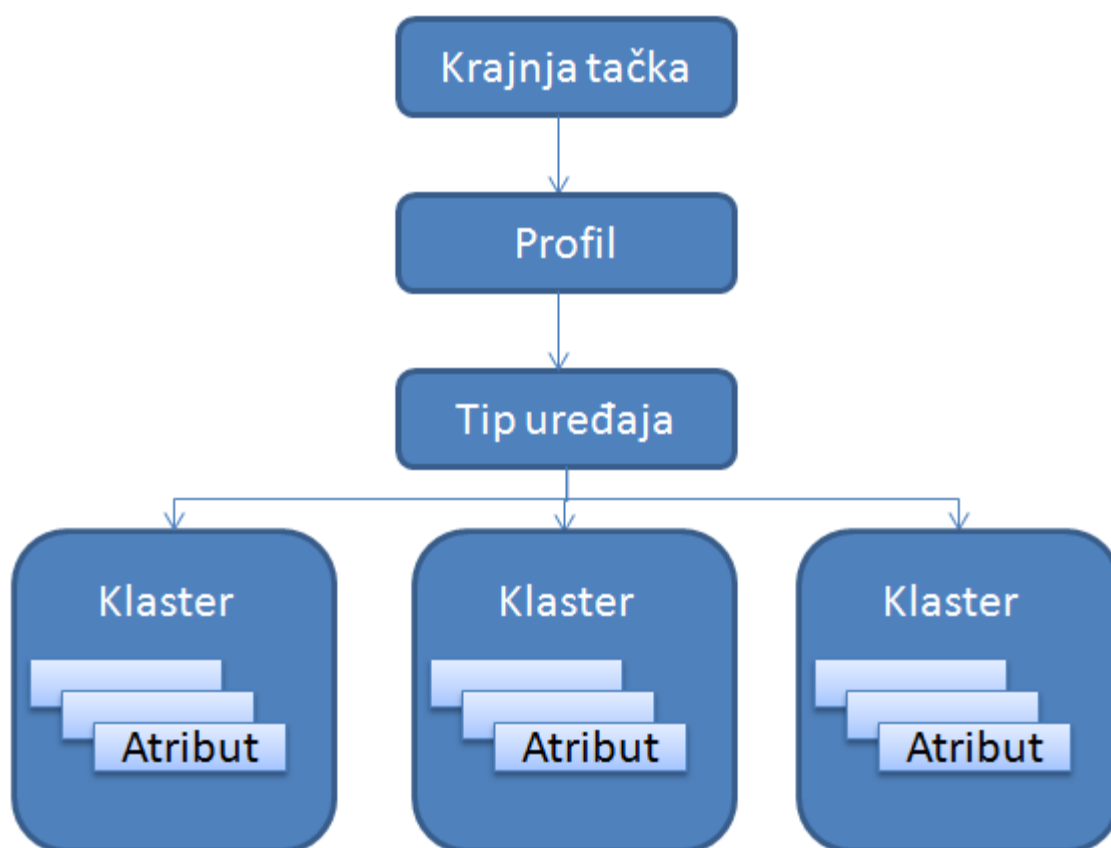
Slika 4: Klijent/Server model klastera

ZigBee konzorcijum je definisala biblioteku klastera[6] u kojoj su specifikovani klasteri koji opisuju najčešće korišćene funkcionalnosti uređaja. Neki primeri definisanih funkcionalnosti su: upali/ugasi, kontrola nivoa, kontrola boje, merenje temperature, itd. Specifikacija klastera podrazumeva attribute i komande koji moraju biti implementirane na uređajima za podržane klasterne. Pored obaveznih postoje i opcioni atributi i komande koje ne moraju biti implementirane od strane uređaja. Broj specifikovanih klastera je veliki i pokriva gotovo sve funkcionalnosti uređaja. Ako je potrebna neka specifična funkcionalnost koja nije definisana u okviru biblioteke klastera moguće ju je dodati specifikacijom novog klastera. Međutim, ovakve funkcionalnosti su specifične za svakog proizvođača i svaki uređaj pa ih je teško portžati na generičan način.

2.1.3 Profil

Funkcionalnosti koje pripadaju istim funkcionalnim domenima se grupišu u profile[8]. Profil specificira skup tipova uređaja koje on obuhvata i skup klastera koji podpadaju pod domen

funkcionalnost profila. Tip uređaja specificira obavezne i opcione klastere koje uređaj datog tipa mora da implementira kao i njegovu ulogu u formiranju i održavanju mreže. Profil dodatno može da dopuni specifikaciju klastera koja je definisana u bibliotekci klastera i da zahteva neke attribute i komande koje su opcione u njoj. Profili omogućavaju da uređaji koji koriste iste ili kompatibilne profile međusobno interaguju bez problema. Svaki aplikacioni objekat na određenoj krajnjoj tački treba da implementira određeni tip uređaja iz profila. Krajnja tačka ima sličnu ulogu kao i prolaz u TCP-u, da razdvoji aplikacije na logičkom nivou u okviru aplikativnog sloja. Na jednoj krajnjoj tački može biti registrovan samo jedan tip uređaja iz profila (slika 5). Neki od najčešće korišćenih profila su: HA[9], ZLL, SE itd.



Slika 5: Hijerarhiska organizacija krajnje tačke

2.2 Opis problema

Oblasti IOT-a i kućne automatizacije se poslednjih godina brzo razvijaju i postoji sve više rešenja za sisteme pametnih kuća. Takođe, broj uređaja koji se koriste u njima je jako veliki i stalno se proširuje. Glavna mana ovih rešenja je loša interoperabilnost sa sistemima i uređajima od drugih proizvođača iako koriste iste protokole za komunikaciju pa su korisnici često uslovljeni da koristimo samo jedno rešenje. Drugi problem se ogleda u potrebni kombinovanja više rešenja da bi se dobila željena funkcionalnost što uglavnom nije jednostavno i potrebno je da imamo više upravljača, aplikacija, kontrolera itd. Neka rešenja i pružaju mogućnost

kombinovanja sa drugima ali ona zahtevaju dodatnu implementaciju kako bi radili zajedno. Performanse mešovutih sistema neretko su znatno lošije nego sistema koja su iz istog rešenja.

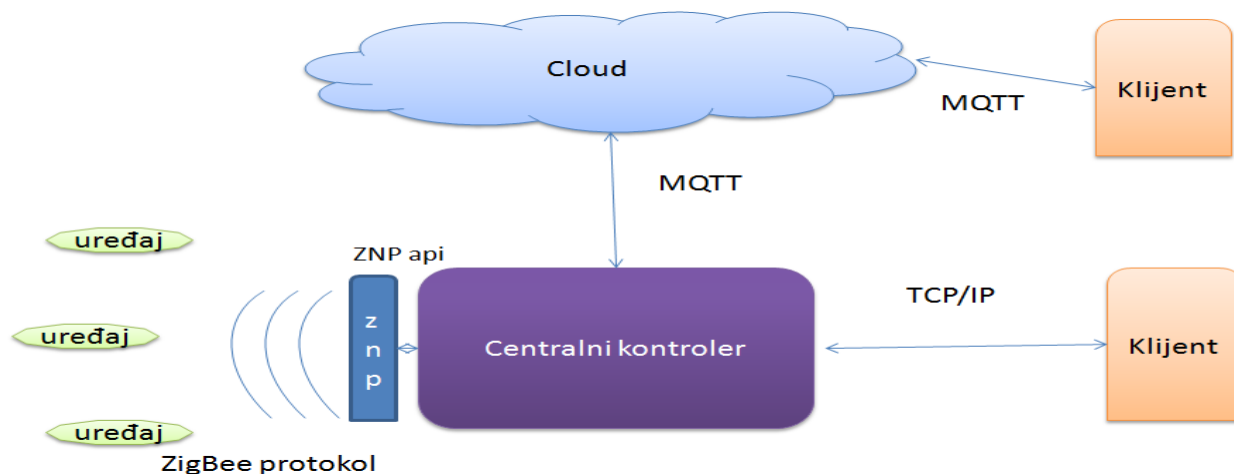
U ovom radu ćemo pokušati da odgovorim da probleme koji nastaju prilikom kombinovanja više nezavisnih rešenja. Predstavićemo rešenje koje treba da odgovori na zahteve kao što su fleksibilnost, proširivost, jednostavnost, lakoća korišćenja, uniformnost na koje dosadašnja rešenja nisu našla adekvatan odgovor. Uz to, pokušaćemo da performanse sistema ostanu na nivou namenskih sistema.

3. Koncept rešenja

Analizom postojećih rešenja za centralni kontroler uočeni su nedostaci u pogledu interoperabilnosti sa drugim sistemima i uređajima kao i u lakoći korišćenja. Sagledavanjem mogućnosti koje nam pruža ZigBee HA1.2 specifikacija i potreba korisnika definisan je koncept rešenja koji treba da reši navedene probleme. ZigBee specifikacija je izabran zbog velikog broja uređaja koji trenutno postoji i jednostavnosti korišćenja. Na kraju ovog poglavlja definisan je način verifikacije rešenja na osnovu čega se mogu potvrditi postavljeni zahtevi.

3.1 Arhitektura sistema za kućnu automatizaciju

Centralni kontroler se sastoji od SOC(*eng. System on Chip*) modula kao glavnog procesora na ploči i pratećih modula sa stekovima za različite protokole. Protokoli koji su podržani na centralnom kontroleru su NFC, Zwave, WiFi, ZigBee, Ethernet. Ti protokoli se koriste za komunikaciju sa pametnim uređajima u sistemu. Dodatno, IP(*eng. Internet protocol*) protokol se koristi i za komunikaciju sa WEB servisima i mobilnim klijentima. Sprega klijenata u lokalnoj mreži ili preko udaljenih(*eng. cloud*) servisa realizovana je preko MQTT protokola.



Slika 6: Arhitektura sistema za kućnu automatizaciju

Sistem za automatizaciju kuće se sastoji iz četiri glavne sistemske celine:

- „pametni“ uređaji koji komuniciraju sa sistemom koristeći protokole koji su prilagođeni njihovim potrebama i resursima kojima raspolažu. Neki od tih protokola su Zwave, ZigBee, BLE itd. Njihova uloga je da dostavljaju informacije o stanjima u kući i izvršavaju korisničkih komandi.
- Centralni kontroler je centralna figura sistema za kućnu automatizaciju. On prosleđuje korisničke zahteve do pametnih uređaja koji ih izvršavaju, pruža korisniku informacije o trenutnom stanju kuće i održava i nadgleda ceo sistem. Klijentske aplikacije mu pristupaju upotrebom HTTP, MQTT ili TCP/IP.
- Udaljeni servisi omogućavaju korisnicima udaljeni pristup sistemu. Za komunikaciju između udaljenih servisa i centralnog kontrolera koristi se niz internet protokola kao što su XMPP, MQTT ili TR069 u zavisnosti od konfiguracije sistema.
- Mobilne ili dekstop klijentske aplikacije realizuju korisničku spregu za pristup ili predstavljanju terminale za pristup sistemu preko udaljenih servisa ili u lokalnoj mreži. Koriste HTTP, MQTT ili TCP/IP za komunikaciju sa sistemom.

3.1.1 Arhitektura programske podrške na centralnom kontoleru

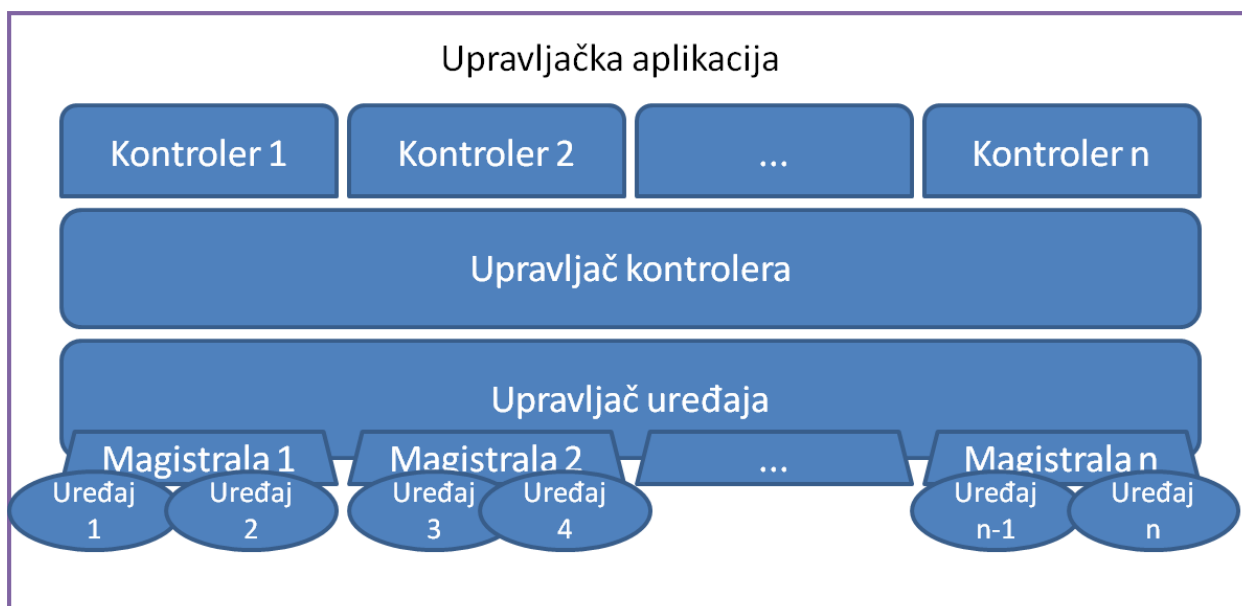
Programska podrška na centralnom kontoleru sastoji se od tri nezavisne aplikacije. To su:

- upravljač sistema(*eng. System Manager*) - upravlja sisetemskim resursima na centralnom kontoleru kao što su LED diode, dugmići, memorija, stanje mreže
- upravljačka aplikacija za kontrolu pametne kuće (*eng. Home Manager*) – upravlja uređajim koji se nalaze u sistemu i izvršava korisničke akcije
- poslužilac za razmenu poruka(*eng. Message Broker*) – posrednik u komunikaciji između upravljačke aplikacije i ostalih aplikacija na centralnom kontoleru

Tema ovog rada će biti realizacija programske podrške za komunikaciju sa uređajima koji koriste ZigBee protokol u skladu sa HA 1.2 specifikacijom koja je deo uprvljačke aplikacije.

3.1.2 Arhitektura upravljačke aplikacije za kontrolu pametne kuće

Upravljačka aplikacija konfiguriše, nadgleda i upravlja „pametnim“ uređajim. Arhitektura upravljačke aplikacije se sastoji od sledećih modula: kontroler, magistrala, uređaj i upravljač za svaki od modula. Arhitektura visokog nivoa je prikazana na Slika 7.

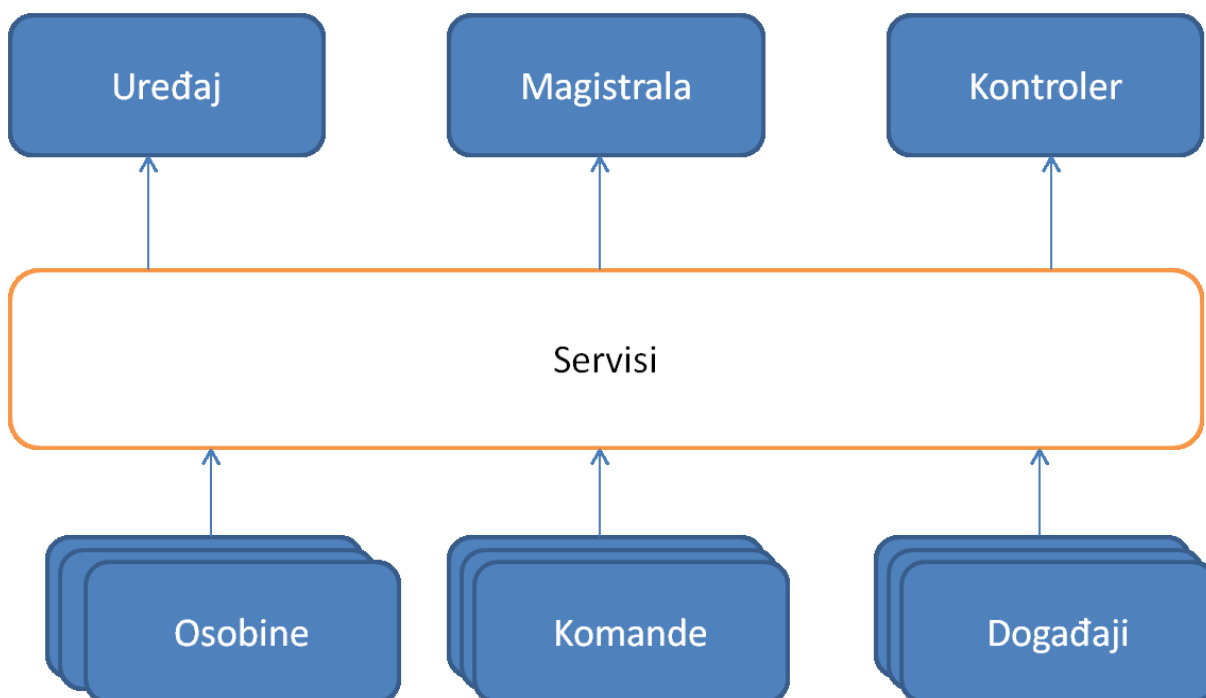


Slika 7: Blok diagram organizacije upravljačke aplikacije

Kontroler omogućavaju proširenje funkcionalnosti upravljačke aplikacije. nadodaju logiku višeg nivoa u vilju proširenja sposobnosti kontrolera i olakšavanja korišćenja sistema. Oni od korisnika sakrivaju kompleksnost sistema i korisniku pružaju jednostavne funkcije. Takođe nadodaju različite sprege za povezivanje sa kontrolerom preko različitih protokola. Neki od kontrolera koji su na raspolaganju korisnicima su HTTP kontroler, scenario kontroler, kontroler TCP korisničke sprege. Upravljač kontrolera raspoređuje zahteve po kontrolerim, pokreće ih i zaustavlja. Sa druge strane, pametni uređaji su apstrakovani modulom uređaja. U okviru ovog modula nalaze se sve zajedničke osobine uređaja nezavisno od protokola preko kog komuniciraju. Sam protokol za komunikaciju sa uređajima apstakovan je modulom magistrale u okviru koga se implementiraju specifičnosti za dati protokol. Neke od zajedničkih osobina koje imaju svi protokoli su otvori/zatvori mrežu, pošalji/primi poruku i prosledi primljenu poruku na uređaj. Modul magistrala i uređaj su pod kotrolom upravljača uređajima. Upravljači, kako kontrolerski tako i uređajski zaduženi su za čuvanje podataka o modulima koji su pod njihovom upravom prilikom gašenja sistema. Podaći se čuvaju u XML-u datotekama.

Komunikacija između modula u programskoj podršci će se odvijati na uniforman način upotrebom servisa. Servis[10] objedinjuje osobine, komande i događaje vezane za određenu funkcionalnost koja je predstavljena tipom servisa. Servis je definisan sa imenom i tipom. Jedan objekat može da poseduje više servisa istog tipa sa različitim imenom. Modul koji poseduje funkcionalnost koja je predstavljena servisom treba da registruje servis kako bi ostali moduli mogli da je koriste. Modul može da ima više servisa istog tipa. Primer za to su uređaji koji imaju više funkcionalnosti istog tipa. Tada se kao identifikator servisa koristi ime servisa. Servisi treba da opišu što više funkcionalnosti preko funkcionalnosti uređaja pa sve do funkcionalnosti

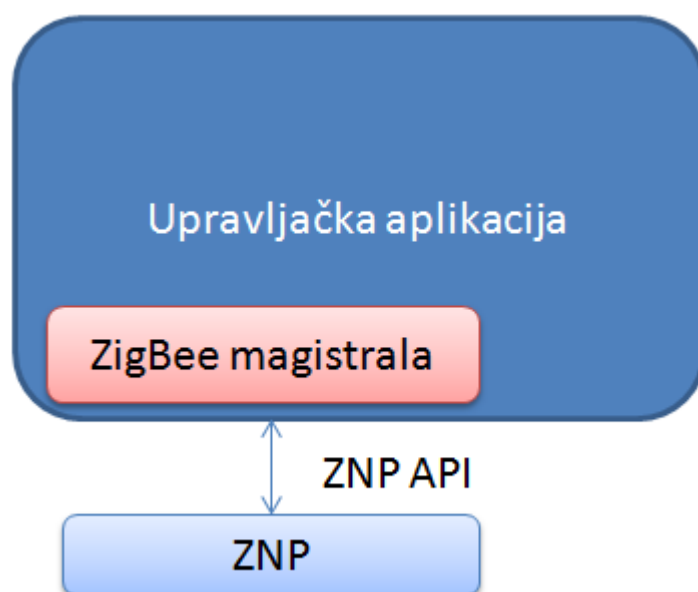
komunikacionih protokola. Slika 8 ilustruje mesto servisa u arhitekturi programske podrške centralnog kontrolera.



Slika 8: Mesto servisa u arhitekturi programseke podrške centralnog kontrolera

3.2 Programska podrška za ZigBee protokol

Na centralnom kontroleru je predviđeno mesto za modul koji će vršiti razmenu podataka sa uređajima koji koriste ZigBee protokol. Modul treba da bude realizovan kao ZNP . ZigBee mrežni procesor treba da izvršavao ZigBee stek i komunicirao bih sa glavnim procesorom na modulu preko serijske veze kroz API pozive koji su definisani od strane proizvođača modula. U okviru upravljačke aplikacije potrebno je realizovati magistralu koja komunicira sa ZNP modulom(Slika 9).



Slika 9: Arhitektura programske podrške za ZigBee protokol

3.2.1 ZigBee magistrala

Modul ZigBee magistrale treba da obavlja komunikaciju sa ZigBee mrežnim procesorom. On proširuje osnovne funkcionalnosti magistrale realizujući ZigBee specifične funkcionalnosti. Sastojace se od pet programskih niti. Niti za slanje i primanje komandi od ZNP-a, niti koja obrađuje primljene komande, niti koja startuje ZNP modul i niti koja kontroliše izvršavanje komandi i vrši ponovno slanje istih. ZigBee magistrala će implementirati funkcije za formiranje i raščlanjivanje komandi iz ZNP API-ja koje definiše proizvođač. Komunikacija između procesa treba da se odvijati preko redova i semafora. Sve globalne promenljive kojima mogu da pristupaju više procesa istovremeno treba da budu zaštićene uzajamno isključivim bravama.

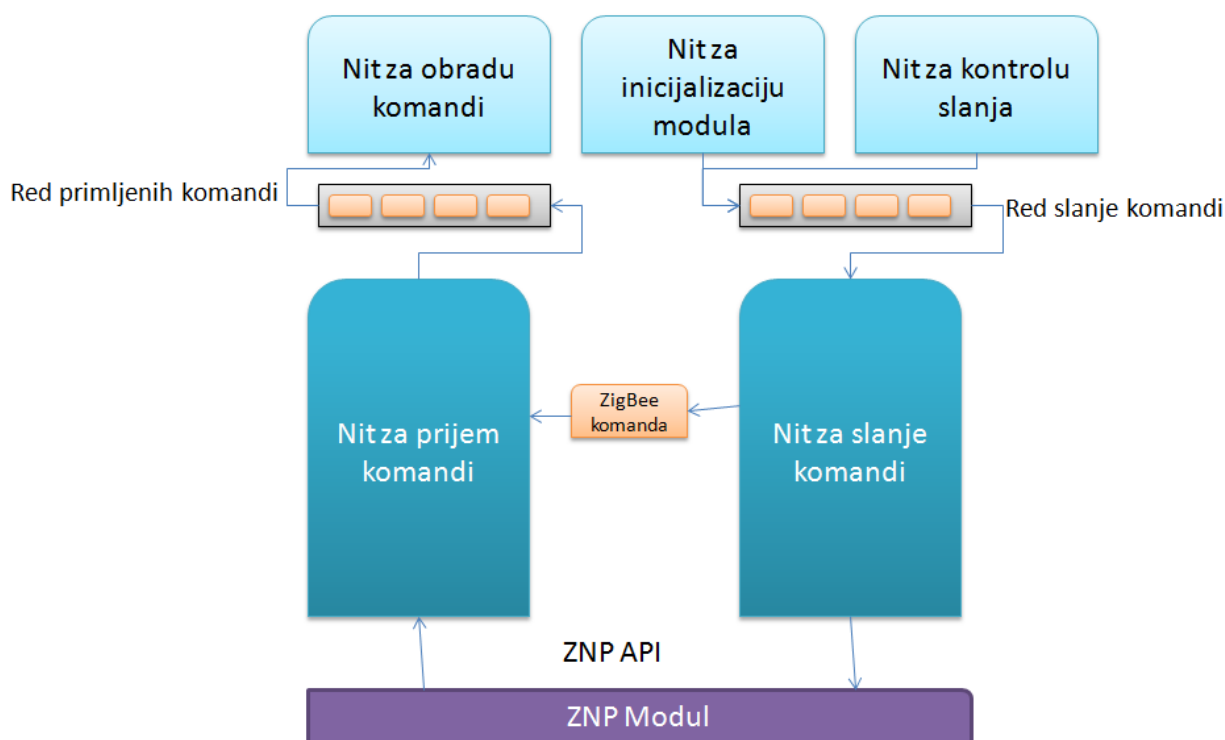
Niti za slanje i prijem komandi od ZNP modula treba da bude sinhronizovan objektom ZigBee komande koji pored niza bajtova i imena komande koje se šalju imaju sinhronizacione primitive. Te primitive omogućavaju blokiranje procesa slanja dok se ne primi odgovor na poslata komandu. ZigBee komanda ima stanja koja govore o tome da li je komanda uspešno poslata, da li je potrebna ponovno slanje ili je komanda neuspešno poslata. Na osnovu ovih stanja se odlučuje da li komanda treba da bude ponovo poslata ili treba da se pređe na obradu naredne komande. Programska nit za slanje komandi ima red iz koga uzima ZigBee komande koje su drugi moduli poslali ka ZNP modulu. Niz bajtova iz objekta ZigBee komande se šalje na serijsku vezu nakon čega se poziva primitiva za blokiranje nit za slanje. Programska nit koja čita niz bajtova sa serijske veze koji prosleđuje metodi za obrađivanje komande tog tipa. Obradivač raščlanjuje niz bajtova, pakuje ga u strukturu i ubacuje u red za komande pristigle od ZNP modula. Obradivač treba da pozvati primiticu poslate komande koja će odblokirati nit za slanje komandi i promeniti stanje poslate komande u zavisnosti od odgovora.

Programska nit koja obrađuje pristigle komande od strane ZNP modula uzima iz reda strukturu ZigBee poruke i u zavisnosti od pošiljaoca poruke poziva se funkcija logičke reprezentacije uređaja koji je poslao komandu.

Programska nit zadužena za inicijalizaciju ZNP modula prilikom pokretanja upravljačke aplikacije u red za slanje komandi ka ZNP modula ubacuje komande koje treba da pripreme modul za obradu korisničkih zahteva. Nakon izvršene inicijalizacione procedura ovaj proces se završava i ostali moduli mogu da ubacuju poruke u red za slanje. Programska nit nakon toga prestaje sa radom.

Programska za kontrolu slanje komande ka ZNP modulu radi ponovno slanje neuspešno poslate komande. Komande koje se ne isporuče uređajima više od 10 puta origlašavaju se za neuspešnim a uređaj se prostavlja u stanje „van mreže“(*eng. Offline*).

Organizacija procesa u okviru ZigBee magistrale je prikazana na Slika 10.



Slika 10: Organizacija ZigBee magistrale

3.2.2 ZigBee uređaj

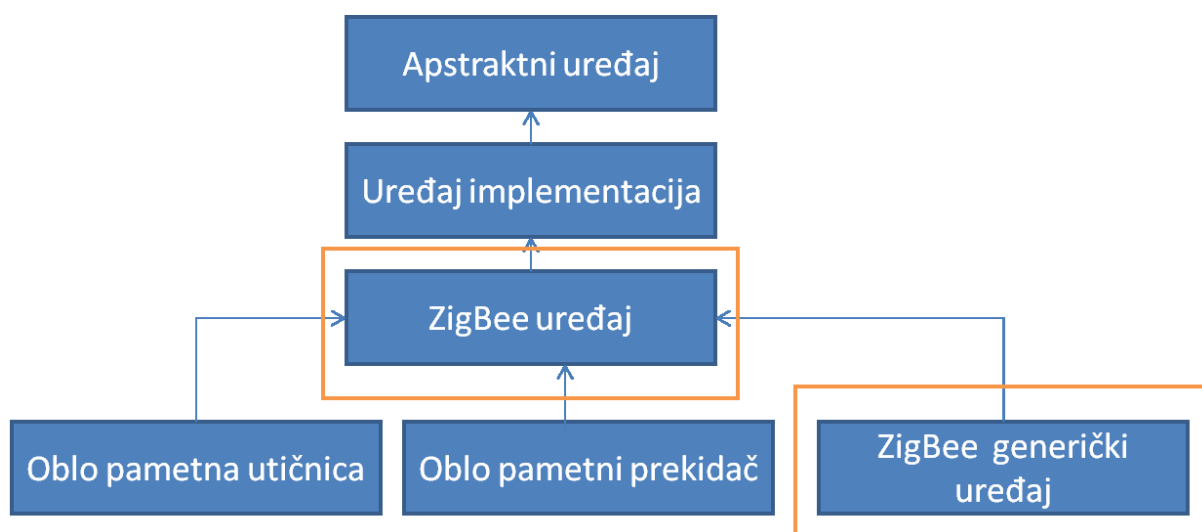
ZigBee uređaj proširuje osnovne funkcionalnosti modula uređaja sa ZigBee specifičnim funkcionalnostima. Ovaj modul predstavlja apstrakciju fizičkog uređaja. Korisnik uređaje koristi kroz apstrakciju koristeći servise sa kojima se uređaj prijavio. ZigBee modul uređaja korisničke zahteve pretvara u komande koje se prosleđuju ZigBee magistrali koja ih dalje šalje na ZNP modul. U isto vreme modul procesira pristigle komande od fizičkog uređaja i ažurira vrednosti servisa koji predstavlja datu funkcionalnost. Na ovom nivou se vodi računa o isporučivanju

komandi do fizičkih uređaja i vrši ponovno slanje. Uređaj u jednom trenutku obrađuje samo jedan korisnički zahtev i dok se on ne završi ostali zahtevi se smeštaju u red. Kada uspešno izvrši zahtev prelazi se na sledeći zahtev. U zavisnosti od tipa uređaja zahtevi korisnika se šalju ka uređaju, ako je uređaj usmerivači, ili se čuvaju u redu dok se uređaj ne javi, krajnji uređaji.

Ako uređaj nije u mogućnosti da ispuni korisnički zahtev, nije dostupan na merži, logička reprezentacija uređaja prelazi u stanje „van mreže“ (*eng. Offline*). U ovom stanju se brišu svi pristigli korisnički zahtevi i ne prihvataju se novi sve dok se uređaj ne vrtati u operabilno stanje. ZigBee uređaj ima mehanizme za vraćanje uređaja u operabilno stanje koji se periodično pokreću sve dok se uređaj ne odazove. Ovi mehanizmi se razlikuju od uređaja do uređaja.

U ZigBee modulu se pored osnovnih informacija o uređaju kao što su: adresa, ime, model, proizvođač itd. i servisa nalaze i putanje do uređaja koje se navode prilikom slanja korisničkih komandi da bi ZNP modul znao da isporuči komandu do fizičkog uređaja. Fizički uređaji centralnom kontroleru periodično šalju putanje po kojoj komande mogu da im se šalju. ZNP modul može da zahteva od uređaja da mu dostave putanje do njih.

Po ugledu na ZigBee specifikaciju, gde se svaki fizički uređaj sastoji od skupa klastera koji opisuju funkcionalnost uređaja, u okviru referentnog sistema definisan je ZigBee uređaj koji predstavlja fizički uređaj i sastoji se od skupa klastera koji su podržani na fizičkom uređaju. Klasteri koji se nalaze na uređaju su vezeni za krajnju tačku te se svakoj krajnjoj tački dodaje lista klastera koji se nalaze na istoj. Na osnovu oznake model uređaja moguće je razlikovati uređaje i napraviti posebne module koji imaju predefinisane klastere za dati model uređaja. Pored toga moguće je od uređaja tražiti spisak podržanih klastera i na osnovu toga dinamički kreirati uređaj. Na Slika 11 se nalazi dijagram nasleđivanja za ZigBee uređaje.



Slika 11: Dijagram nasleđivanja za ZigBee uređaje

3.2.3 Komunikacija između magistale i uređaja

Instanciranje uređaja je zadatak upravljača uređaja. Kada magistrala prikupi sve potrebne podatke za instanciranje uređaja obraća se upravljaču uređaja. Na osnovu imena klase koji magistrala šalje upravljaču uređajima instancira se objekat uređaja, pokreće se i inicijalizuje. Na kraju se instanca uređaja dodaje magistrali preko koje se vrši komunikacija sa uređajem. Magistrala sve komande koje stižu od uređaja prosleđuje logičkoj reprezentaciji uređaja na dalju obradu. Sa druge strane uređaj korisničke zahteve proseleđuje na magistralu.

Postupak dodavanja i izbacivanja uređaja uključuju više modula. ZigBee magistrala za komunikaciju sa uređajima koristi ZigBee deskriptor uređaja. Preko njega magistrala pristupa modulu uređaja. Da bi se uređaj dodao u mrežu neophodna je korisnička akcija koja će otvoriti mrežu za ulazak novih uređaja. Nakon toga uređaj se prijavljuje ZigBee magistrali i u tom momentu se instancira ZigBee opis uređaja koji ima ulogu u čuvanju informacija o uređaju sve do instanciranja logičke reprezentacije istog. U okviru tog modula je realizovana mašina stanja kroz koju svaki uređaj prolazi od ulaska do izlaska iz mreže.

Nakon instanciranja opisa uređaja on pokušava da otkrije da li je uređaj kompatibilan sa referentnim sistemom. Ako nije biće izbačen iz njega. Ako je kompatibilan sa referentnim sistemom prelazi se na otkrivanje informacija o uređaju kao što su model, proizvođač, napajanje i tip. Kada uređaj odgovori na ove zahteve proverava se da li u sistemu postoji namenska klasa koja obrađuje zahteve za dati model uređaja. Ako postoji izdaje se zahteve upravljaču uređaja da instancira objekat namenske klase. U slučaju da ne postoji realizacija namenske klase za dati uređaj prelazi se na otkrivanje klastera koje uređaj implementira. Kada se otkriju klasteri pravi se instanca generičnog uređaja i prosleđuju mu se klasteri koje uređaj podržava da ih ubaci u listu podržanih klastera.

Završna faza dodavanja uređaja u sistem se sastoji od inicijalizacije uređaja. Svaki uređaj implementira funkciju u kojoj definiše komande koje moraju biti poslate kako bi on bio inicijalizovan. Ako bilo koji korak prilikom ulaska uređaja ne uspe uređaj nakon dva minuta pokušavanja biva izbačen iz mreže.

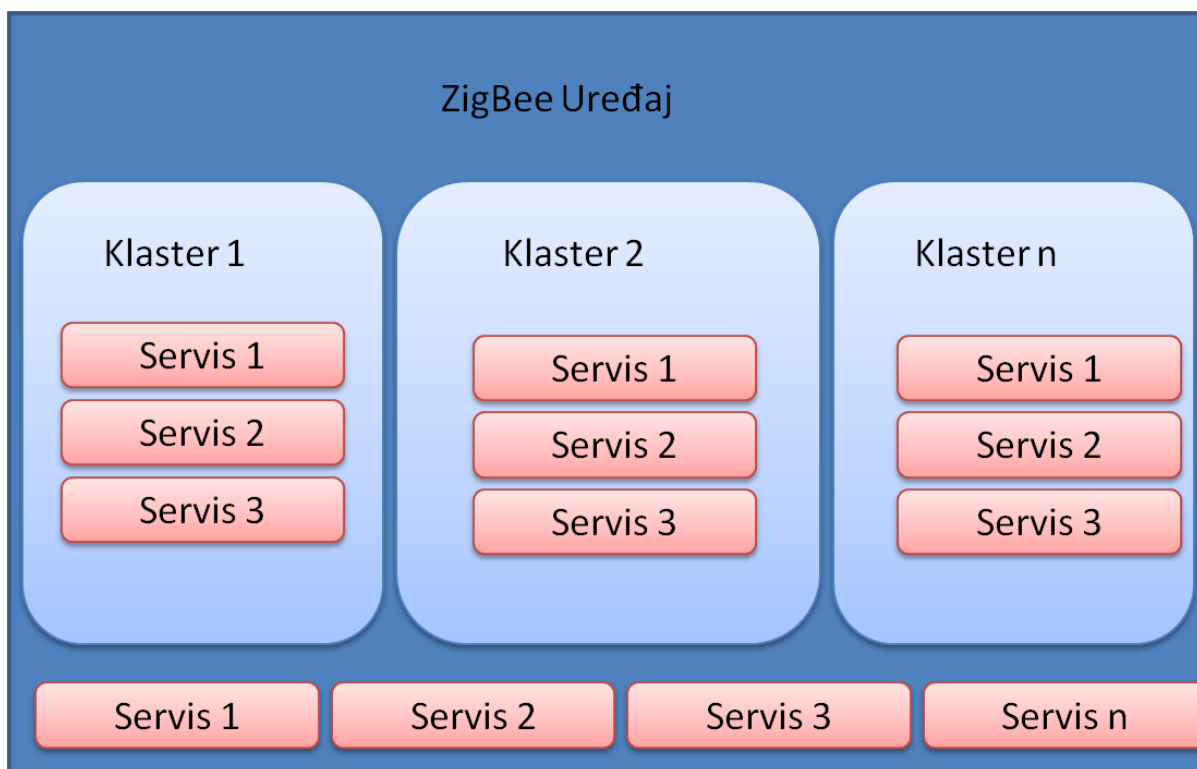
3.2.4 Klaster modul

Klaster modul predstavlja apstrakciju klastera koju definiše ZigBee specifikacija. Ovaj modul se sastoji od imena klastera, identifikatora klastera i profila kom klaster pripada. Funkcija ovog modula je da raščlanjuje komande koje stižu od fizičkih uređaja koji podržava taj klaster. Klaster modul ima funkcije za formiranje komande koje se šalju na fizički uređaj. Ovaj modul je deo logičke reprezentacije uređaja. Klaster modul može da ima servise preko kojih se korisniku omogućava da izdaje zahteve na osnovu kojih se formiraju komande koje se šalju na

uređaja. Kada se klaster dodaje na uređaj njegovi servisi se dodaju u listu servisa datog uređaja i korisnik ih koristi kao servise uređaja.

Klaster modul izdvaja metode koje su zajedničke za sve klasterne koji su definisani ZigBee specifikacijom. Iz ovog modula se izводе namenski klasteri koji imaju svoje specifične komande i atribute. Na osnovu uređaja koji će biti podržani od strane upravljačke aplikacije napraviće se spisak klastera koji će biti implementirani u programskoj podršci. Oni će biti razvrstani u posebne strukture u zavisnosti od profila kome pripadaju.

Uvođenje klaster modula će programsku podršku za ZigBee specifikacija učiniti modularnom i omogućiti lako proširivom novim funkcionalnostima i klasterima. Što više klastera se implementira to je veći skup uređaja koje podržava referentni sistem. Modul ZigBee uređaja će registrovati klasterne koje fizički uređaj podržava i nakon toga je moguće komunicirati sa fizičkim uređajem preko registrovanog klastera. Primenom ovog koncepta moguće je dinamičko dodavanje uređaja u sistem na osnovu podržanih klastera nezavisno od proizviđača. Prilikom ulaska uređaja u sistem on će biti pitan za podržane klasterne ako nema implementacije namenske klase. U zavisnosti od odgovora dinamički se instanciraju klasteri dodatom uređaju. Ako uređaj ima neke funkcionalnosti koje nisu podržane od strane sistema on će biti dodat bez tih funkcionalnosti. Prikaz organizacije uređaja je prikazan na Slika 12. Na njoj možete uočiti da se svaki uređaj sastoji od klastera koje podržava i servisa koje podržava. Servise koje podržava klaster se pridružuju servisima uređaja.



Slika 12: Organizacija ZigBee uređaja

3.3 Provera rešenja

Na kraju ovog poglavlja postavice mo zahteve koji treba da budu provereni nakon implementacije ovog koncepta. Parametri koji ce biti mereni i pokazati da li je koncept bio dobro definisan i da li je implementacija zadovoljila ocekivanja ce ove biti razmotreni.

Testirace se dva aspekta:

1. Uskladenost sa ZigBee HA 1.2 specifikacijom
2. Performanse sistema

Kako bi proverili uskladenosta sa HA 1.2 specifikacijom bice potrebno da centralni kontroler prode skup testove koji su specifikovani od strane ZigBee alijanse. Testovi koje uređaj treba da prode zavise od tipa uređaja koji izaberemo u implementaciji i klastera koji budu podržani od strane istog.

Performanse sistema ce biti merene u nekoliko aspekata. Meri se broj uređaja koji moze da bude podržan od strane sistema, brzina odziva sistema, stres testovi u vidu velikog broja poruka koji se šalju istovremeno na meržu. Takođe ce se meri brzina isporuke poruka do najudaljenijih uređaja u sistemu. Ocekivani broj uređaja koje sistem podržava je 50. Ovaj broj uređaja je utvrđen analizom drugih sistema za kućnu automatizaciju i potrebama prosečnog korisnik. Ocekivani odziv sistema za 50 poruka istovremeno je oko 14 sekundi dok što je oko 0,2 sekunda po komandi. Ocekivani broj skokova do najudaljenijih uređaja najviše 5-7 uređaja. Zadovoljavajuće vreme za isporuku poruke ovim uređajima je 0,5 sekundi.

Zahtevane performanse su utvrđenje empirijski, na osnovu iskustava i ocekivanja korisnika sistema, i performansi drugih sistema. Nakon implementacije rešenja za programsku podršku za ZigBee protokol po HA 1.2 specifikaciji navedeni provere moraju biti uređene kako bi se proverila korektnost rešenja. Rezultate testova treba uporediti sa postavljenim zahtevima i ako uređaj bude ispunio zahteve koncept rešenja koji smo ovde postavili i implementacija ce biti potvrđena kao uspešna.

4. Programsko rešenje

Na osnovu koncepta rešenja koje je zasnovano na teorijskim osnovama i opisu problema nastalo je programsko rešenje koje će biti predstavljeno u ovom poglavlju. Biće predstavljene implementacije modula programske podrške iz koncepta rešenja i njihove glavne karakteristike i funkcije. U implementaciji se koriste projektni obrasci kako bi rešenje bilo lako održavati i proširivati sa novim funkcionalnostima. Obrazaci koji su korišćeni prilikom implementacije su: fabrika[12], prototip[13] i stanje.

4.1 Implementacija programske podrške za ZigBee protokol

Odlučili smo da na centralnom kontroleru ZigBee mrežni procesor bude SOC CC2530 od kompanije Texas Instruments. Za ovaj čip smo se odlučili zbog kvalitetne implementacije ZigBee steka i jednostavnog i dobro specifikovanog API-ja za korišćenje. Sa ovim modulom treba da komunicira magistrala pod nazivom *ZnpBus* koji omogućava API pozive ka modulu i rasčlanjivanje odgovora. Komunikacija se odvija preko serijske veze.

Na ZNP modul je trebalo registrovati tip uređaja i klastere koji su podržani na odgovarajućim krajnjim tačkama. Iz HA 1.2 specifikacije izabrali smo kombinovani interfejs(eng. Combined Interface). Ovaj tip uređaja se najbolje uklapa u specifikaciju centralnog kontrolera koji je zadužen za upravljanje drugim uređajima. Za kombinovani interfejs samo su osnovni (eng. Basic) i identifikacioni(eng. Identify) klasteri obavezni na serverskoj strani. Na klijentskoj strani ima samo opcione klastere. Koji će klasteri biti implementirani zavisi od toga koji skup uređaja želimo da podržimo u okviru referentnog sistema. Analizom potreba korisnika došli smo do zaključka da su dovoljni sledeći klijentski klasteri:

- Uključi/Isključi(eng. On/Off) klaster
- Kontrola nivoa(eng. Level Control) klaster
- Kontrola prozivanja(eng. Poll Control) klaster

- Sigurnosne zone (eng. IAS Zone) klaster
- Merenje (eng. Metering) klaster
- Električna mera (eng. Electrical Measurement) klaster

Navedeni klasteri pokrivaju funkcionalnosti koje su zahtevane od korisnika sistema za kućnu automatizaciju kao što su uključivanje i isključivanje uređaja, kontrolu nivoa otvorenosti roletni ili osvetljaja sijalica, senzorski uređaji koji detektuju pokret ili otvorenost i zatvorenost vrata, kontrolu potrošnje struje, gasa itd. Lista podržanih klastera se može na jednostavan način proširivati. Na osnovu odabranih klastera biće određen skup testova koje centralni kontroler mora da prođe kako bi se utvrdilo da je uređaj u skadu sa HA 1.2 specifikacijom.

4.1.1 ZigBee magistrala

U okviru ZigBee magistrale nalaze se procesi za slanje i raščlanjavanje API komandi koje se razmenjuju sa ZNP modulom. Klase koje implementiraju ta dva procesa su respektivno *CSendToPortRunnable* i *CReadFromPortRunnable*. Kao i svi procesi u okviru upravljačke aplikacije i ovi proširuju klasu *Runnable* iz *Poco[11]* biblioteke koja je korišćena u implementaciji programske podrške. Pored ova dva osnovna modula za komunikaciju sa ZNP modulom preko serijske veze implementirani je proces za inicijalizaciju čipa koji je deo klase *CInitBusManagerRunnable* kao i process za obrađivanje pristiglih komandi koji je deo klase *CProcessIncomingMsgRunnable*. Za ponovno slanje neuspešno izvršenih komandi zadužena je klasa *CRetransmitMsgRunnable*.

4.1.1.1 Proces za slanje API komandi na ZNP modul

Proces koji šalje API komande na ZNP modul predstavlja potrošača u konceptu proizvođač-potrošač dok su inicijalizacioni proces i proces koji obrađuje komande proizvođači. Ovi procesi komuniciraju preko reda za komande. Proizvođački procesi u taj red ubacuju komande koje je potrebno upisati na setijsku vezu. Objekat komande koji se nalazi u redu biće kasnije objašnjen. Red je ograničen na 150 komandi što se smatra dovoljno za potrebe referentnog sistema. Metode koje manipulišu sa redom su:

- `CZigBeeCommand* CreateZigBeeCommand(const string& _szCommand, const Parametars* _pCmdParams);`

Metoda konstruiše objekat ZigBee komande u zavisnosti od imena komande i parametra. Metoda vraća objekat konstruisane komande.

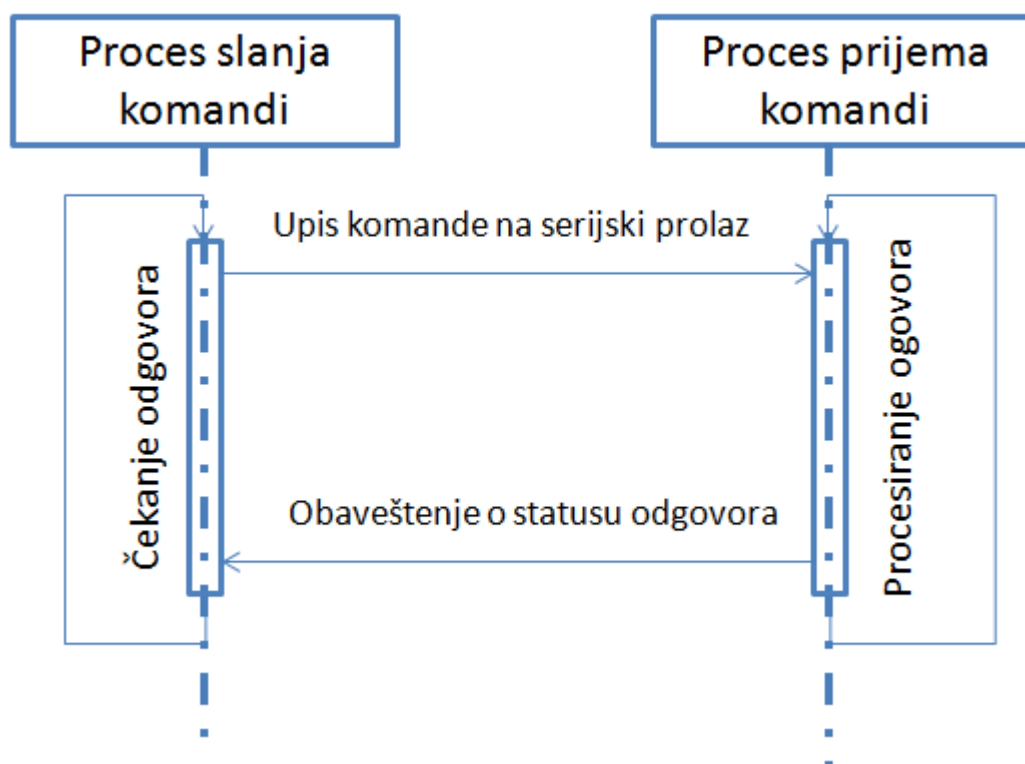
- `bool AddCommandToQueue(CzigBeeCommand* zigBeeCommand);`

Metoda ubacuje komandu u red za slanje ako red nije pun. Ako je red pun metoda vraća netativan odgovor.

Proces za slanje komandi iz reda za slanje uzima jednu po jednu komandu za slanje sve dok red ne bude prazan. Ako je uspešno izvadio komandu iz reda šalje je na serijski prolaz. Nakon toga čeka odgovor da je komanda uspešno upisana na serijski prolaz. Sinhronizacija je implementirana pomoću binarnih semafora. Kako je red deljena promenljiva između više procesa štiti se uzajamno isključivim bravama.

4.1.1.2 Proces za prijem API komandi sa ZNP modula

Zadatak ovog procesa je čitanje podataka sa serijskog prolaza i njihovo raščlanjavanje. Ovaj proces komunicira sa procesom za slanje komandi preko ZigBee komanda objekta. Proces slanja u globalni objekat komande upiše pokazivač na komandu koju je upisao na serijski prolaz i blokira se dok ne stigne odgovor za poslatu komandu. Kada odgovor stigne u proces prijema komandi ovaj proces će u zavisnosti od statusa odgovora promeniti stanje poslate komande i odblokirati proces slanja. Komunikacija između procesa prikazana je na slici 13. Kako bi propusnost procesa koji čita komande sa serijskog prolaza bila što veća dalja obrada komande se prebacuje na proces za obradu komandi. Ova dva procesa su takođe u odnosu proizvođač-potrošač gde proces za prijem komandi ima ulogu proizvođača a proces koji obrađuje komande ima ulogu potrošača. Za razmenu komandi između ova dva procesa koristi se red koji je ograničen na 500 komandi.



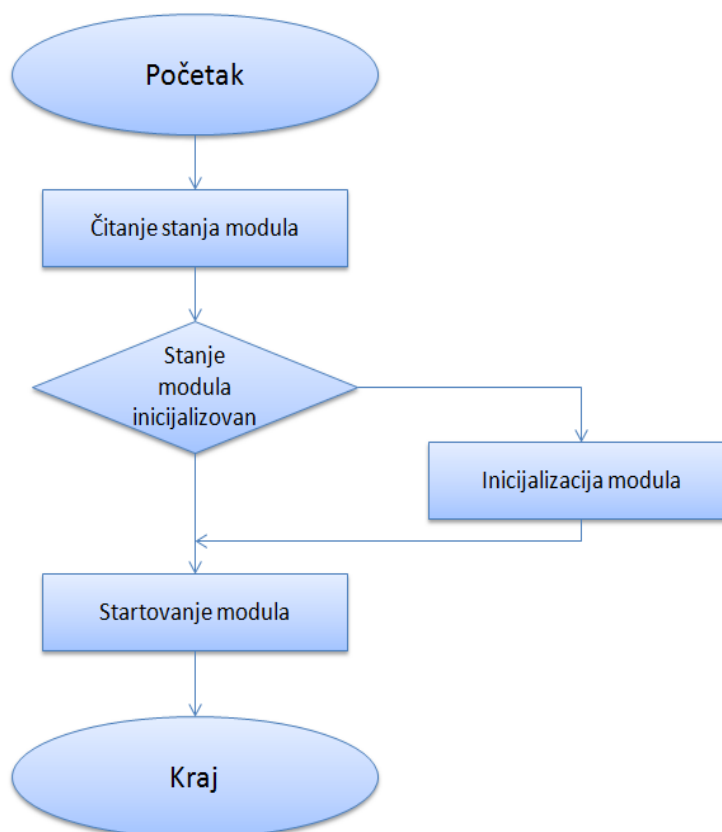
Slika 13: Komunikacija između procesa za slanje i prijem komandi

4.1.1.3 Proces za obradu API komandi ZNP modula

Proces za obradu pristiglih komandi iz reda komandi za obradu vadi komandu po komandu. Na osnovu adrese pronalazi uređaj od koga je stigla poruka i prosleđuje mu korisne podatke na dalju obradu. Uređaji koji su u okviru ZigBee mreže se čuvaju u mapi opisa uređaja. Ključ u ovoj mapi je adresa uređaja i po njoj se pronalazi odgovarajući opis uređaja. Ako se opis uređaja nalazi u mapi onda će se pozvati njegova metoda *RecvFromBus* koja će biti objašnjena u okviru ZigBee uređaja. Ako uređaja nema u ovoj mapi to znači da uređaj nije ušao na regularan način u mežu i šalje mu se zahtev za izlazak iz mreže.

4.1.1.4 Proces za inicijalizaciju modula i ponovno slanje komandi

Ova dva procesa imaju ulogu u startovanju modula i ponovno slanje neuspelo poslatih komandi. Inicijalizacioni proces u red za slanje komandi ubacuje komande koje moraju biti izvršene prve kako bi se ZNP modul startovao i dok se ove komande ne izvrše modul nije pokrenut i neće obrađivati druge zahteve. Iz tog razloga dok se proces inicijalizacije ne završi ostali procesi ne mogu da ubacuju komande u red za slanje. Sistem se neće pokrenuti sve dok se inicijalizacioni proces ne završi uspešno. Inicijalizacione komande se razlikuju u zavisnosti od toga u kom je stanju modul pa se prvo pročita trenutno stanje modula i na osnovu toga određuje dalji tok komandi (slika 14).



Slika 14: Prikaz inicijalizacione procedure

Proces za ponovno slanje komandi ima ulogu da vodi računa da se nit za slanje ne blokira na nekoj komandi koja je neuspešno poslata na ZNP modul. Ako neka komanda nije uspešno poslata ovaj modul ima ulogu da je ponovo pošalje nakon 10 sekundi dok proces za slanje nastavlja nesmetano da šalje ostale komande. Ako komanda ni posle tri pokušaja ne uspe da se pošalje onda se od te komande odustaje.

4.1.1.5 ZigBee komanda

ZigBee komanda se sastoji od niza bajtova koji će se upisivati na serijski prolaz pored toga, ima i dva binarna semafora koji služe za sinhronizaciju sa odgovorom na poslatu komandu. Koji semafor će biti korišćen zavisi od toga da li će odgovor na komandu biti sinhron ili asinhron. Kog tipa je odgovor i koji semafor se koristi zavisi od tipa komande i to se interno određuje u okviru objekta komande. Komanda ima i stanja u zavisnosti od kojih se određuje da li komandu treba ponovo slati ili je komanda uspešno poslata. Metode od značaja za rad sa ZigBee komandom su:

- `void Wait();`
Čeka na potvrdu o prijemu odgovora. U okviru ove metode se odlučuje, u zavisnosti od tipa komande, koji se odgovor čekati tj. Koji semafori će biti korišćeni.
- `void SyncRespRecieve();`
Poziva se kad je stigao sinhroni odgovor na komandu. Podiže se semafor i menja se stanje komande.
- `void ArspRecieve();`
Poziva se kada je stigao asinhroni odgovor na komandu. Podiže se semafor i menja se stanje komande.

Kako bi se sprečilo blokiranje procesa slanja, u slučaju da ne stigne poruka, korišćeni su semafori sa vremenskim ograničenjem koji će nakod određenog vremena, u zavisnosti od tipa komande, da se podignu iako nije stigao odgovor. Ovakva situacija se registruje kao problem prilikom slanja komande i proces za ponovno slanje ponovo šalje komandu nakon 10 sekundi. Sinhroni semafor ima vreme čekanja na odgovor od 3 sekunde dok se na asinhronom semaforu čeka 10 sekundi.

4.1.1.6 ZigBee deskriptor

Klasa *CZigBeeDeviceDescriptor* vodi računa o prikupljanju osnovnih informacija o uređaju na osnovu kojih se instancira namenska klasa za kontrolu datog modela uređaja. Implementirana je kao mašina stanja. Uslov za prelazak iz stanja u stanje je da uređaj pošalje podatke koji se od njega zahtevaju za dato stanje. Podaci se periodično, na svakih 10 sekundi,

traže od uređaja sve dok ne stigne ogovor a najduže 2 minuta. Ako u roku od 2 minuta ne stignu svi podaci koji su neophodni za instanciranje uređaja on se izbacuje iz mreže. Kada se prikupe svi podaci potrebni za instanciranje adekvatnog uređaja oni se prosleđuju upravljaču uređajima kako bi on instancirao namenski uređaj i dodao ga u sistem. Instancirani uređaj postaje deo opisa uređaja i ZigBee magistrala mu pristupa posredstvom opisa uređaja. Najvažnija metoda u ovom modulu je:

- `void goToNextState();`

U zavisnosti od stanja i podataka koji su stigli metoda određuje naredno stanje i narednu komandu koju treba poslati uređaju. Poziva se uvek kada stignu nove informacije od uređaja ili kada istekne 10 sekundi od prethodnog zahteva za informacijama

4.1.2 Klaster

Klaster modul je implementiran u okviru *CCluster* klase. To je prototip klastera i sve ostale klase za klastere se izvode iz ove klase. Svaki klaster ima svoje ime i jedinstveni identifikator u okviru profila. Ove vrednosti se nalaze kao polja u *CCluster* klasi. Pored toga u klasi se nalazi i mapa atributa koji svaki klaster popunjava u okviru svog konstruktora i ima mapu pokazivača na metode za raščlanjavanje komandi koje se rezmenjuju između klastera. Ključ u ovim mapama su identifikatori atributa i komandi. Ovakva realizacija je izabrana jer je postupak obrade pristigle komande isti za sve komande i razlikuje se samo u tumačenju korisnog dela poruke. Osnovna klasa klastera ima još i metode za raščlanjavanje komandi koje su iste za sve klastere. To su metode iz skupa generalnih okvira komandi koje svi klasteri u ZigBee biblioteki klastera moraju da implementiraju. Takođe, postupak raščlanjavanja i formiranja zaglavlja komande isti su za sve komande i njihove implementacije su deo osnovne klase klastera. Metode za raščlanjavanje podataka zavisno od tipa podatka ne zavise od klastera pa su implementirane u osnovnoj klasi. Na kraju klaster ima listu servisa kojima pruža svoje funkcionalnosti ostatku sistema. Osim servisa metoda koja se najviše koristi je:

- `void parseClusterCommands(const unsigned char* _pCommandBuffer, unsigned int _uiCommandLength)`

Metoda poziva ZigBee uređaj kada mu stigne komanda od klastera. U okviru nje su definisani koraci za raščlanjavanje komande koji su zajednički za sve klaster komande. Na osnovu identifikatora komande dalje poziva odgovarajuću metodu za raščlanjavanje korisnog dela komande.

U programskoj podršci za ZigBee protokol implementirani su klasteri iz HA profila koji će podržati referentni sistem. Pored njih implementirani su i klasteri iz privatnog profila koji je razvijen u okviru ovog projekta kako bi poboljšao performanse sistema pod imenom OHN profil.

Uređaji koji podržavaju ovaj profil efikasnije će raditi sa izabranim rešenjem ali to nije ograničenje za uređaje koji to ne podržavaju. Na grupu klastera iz istog profila primenjen je fabrika obrazac u kombinaciji sa prototip obrascem. Implementirana je klasa *CHAProfilClusterFactory* koja ima mapu pokazivača na prototip klastera a kao ključ se koristi jedinstveni identifikator klastera. Fabrika klastera u svom konstruktoru napuni mapu sa pokazivačima na sve do sada implementirane klastere. Za rad sa fabrikom korisniku je dovoljna samo jedna metoda:

- `Ccluster* createCluster(unsigned short _usClusterId);`

Metoda za dati identifikator klastera vraća objekat klastera koji je u mapi registrovan pod tim identifikatorom

4.1.3 ZigBee uređaj

ZigBee uređaj proširuje klasu uređaja sa specifičnostima koje imaju ZigBee uređaji. Klasa ZigBee uređaja u sebi objedinjuje sve zajedničke osobine fizičkih uređaja koji su bitna za rad sa uređajem u okviru referentnog sistema. Ona uvodi mapu klastera na datoj krajnjoj tački. Kao identifikator u toj mapi koristi se broj krajnje tačke koja je aktivna a sadržaj je mapa klastera koji se nalaze u izlaznoj listi klastera uređaja. Metoda koje rade sa mapom su:

- `void addCluster(Ccluster* _pCluster, unsigned char _ucEndPoint = HA_PROFILE_ENDPOINT);`

Dodaje klaster na krajnju tačku u mapi klastera na uređaju. Pored toga registruje servise klastera kao servise uređaja kako bi korisniku bili na raspolaganju. Predefinisana vrednost za krajnju tačku je krajnja tačka koju najčešće koristi HA profil tj. 1.

- `Ccluster* getCluster(unsigned short _ucId, unsigned char _ucEndPoint = HA_PROFILE_ENDPOINT)`

Metoda vraća pokazivač na klaster koji se nalazi u mapi klastera sa datim identifikatorom na krajnjoj tački.

Metode koje su nasleđene od klase uređaj a koje su proširene u okviru ZigBee uređaja su:

- `virtual bool Start();`

Metoda se poziva prilikom pokretanja uređaja i menja stanje uređaja u stanje radi.

- `virtual bool Stop();`

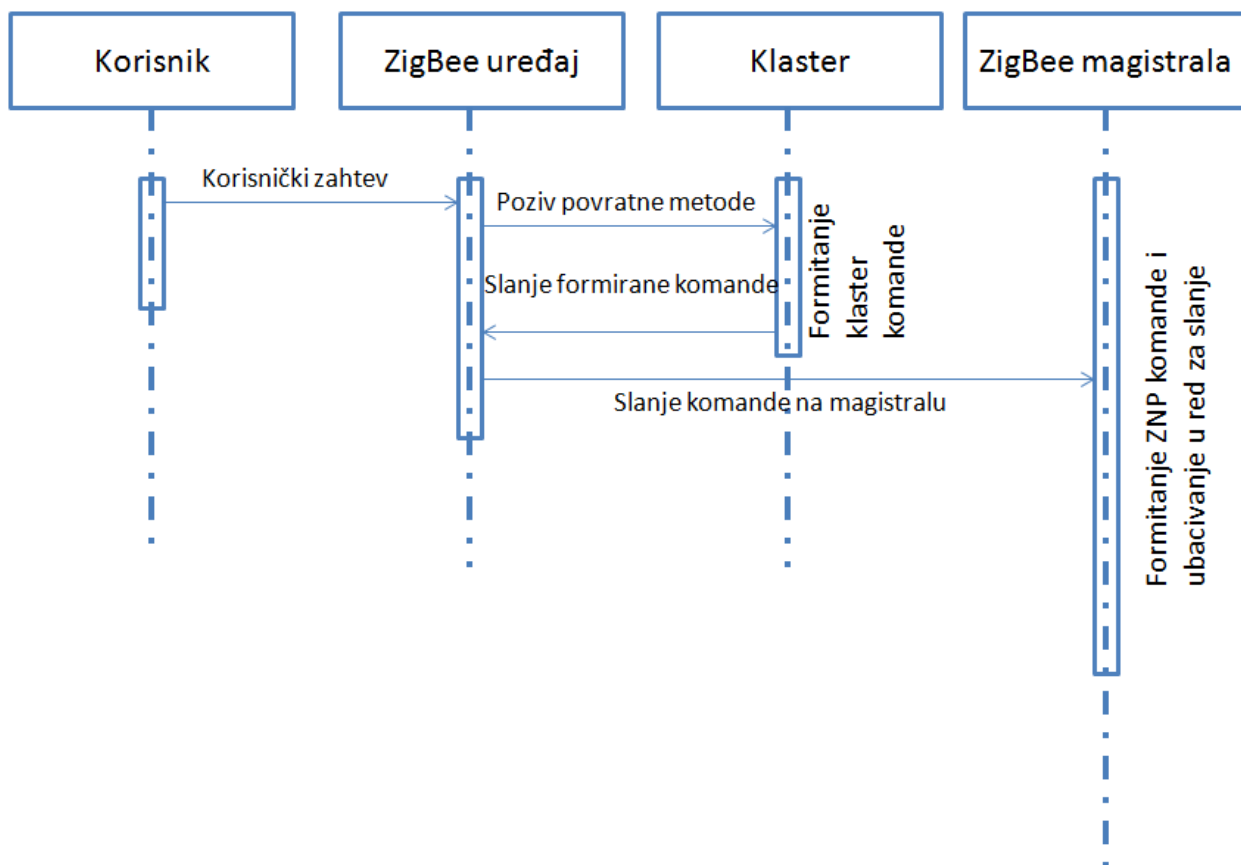
Metoda menja stanje uređaja u stanje spreman ako je uređaj u stanju radi.

- `bool RecvFromBus(const unsigned char* _pZigbeeCommand, const unsigned int _uiSize, const Parametars* _pMsgParams);`

Metoda koju poziva magistrala kada od fizičkog uređaja stigne komanda.

Metoda je za svaki uređaj ima iste korake. Iz mape klastera se pronalazi objekat klastera na osnovu identifikatora koji se prosleđuje kroz parametre i poziva metodu za raščlanjavanje komande na klasteru.

Korisnički zahtevi koji su upućeni ka uređaju u želji korisnika da promeni stanje na fizičkom uređaju ili da pročita trenutne vrednosti sa uređaja izazivaju poziv povratne funkcije(eng. call-back) koja je registrovana na servis koji je korisnik svojom akcijom aktivirao. Ova povratna funkcija implementirana je u okviru klastera ili uređaja. U okviru nje se obrađuje korisnički zahtev i uglavnom rezultuje formiranjem klasterske komande koja se šalje na uređaj. Pored korisnog dela komande kroz parametre prosleđuju se još i identifikator klastera koji formira komandu, krajnje tačke na fizičkom uređaju i centralnom kontroleru. Ako se do uređaja dolazi preko nekih drugih uređaja u parametre se smešta i putanja do uređaja. Ovako formiran korisnički zahtev se prosleđuje ZigBee magistrali ako već ranije nije poslat neki korisnički zahtev na izvršavanje. Ako postoji korisničkih zahteva koji se izvršavaju novi zahtev se smešta u red zahteva koji čekaju na izvršenje. Veličina ovog reda je 50 zahteva. Kada trenutno izvršavani zahtev završi sa izvršavanjem prelazi se na sledeći. Na slici 15 je prikazana obrada jednog korisničkog zahteva od strane zigBee uređaja. Korisnička akcija se završava ili ubacivanjem komande u red za slanje u okviru uređaja, ako se trenutno obrađuje neki zahtev, ili u red za slanje komandi na ZNP modul.



Slika 15: Obrada korisničkog zahteva

Iz ove klase se izvode klase za sve ostale module uređaja koje referentni sistem podržava. One u svom konstruktoru dodaju klastere u mapu klastera. Klase za poznate uređaje implementiraju i specifične osobine za dati modul uređaja.

4.1.4 ZigBee generični uređaj

Pored klase za različite modele uređaja, koje su unapred definisane, iz ZigBee uređaja izvodi se klasa koja može dinamički da se popunjava sa klasterima prilikom ulaska uređaja u sistem pametne kuće. ZigBee generični uređaj iz fabrike klastera, za profil koji implementira, uzima objekat klastera preko identifikatora klastera koji dobija od fizičkog uređaja. Informacije o uređaju koji treba da bude dodat u referentni sistem čuvaju se u ZigBee opisu uređaja u okviru magistrale. Podaci o tome koji se klasteri nalaze na kojoj krajnjoj tački se uređaju isporučuju u XML formatu. Ove podatke generički uređaj rasčlani i popuni svoju mapu klastera sa podacima iz XML-a. XML format za prosleđivanje podataka do uređaja koristi se zbog potrebe da se podaci prilikom gašenja aplikacije sačuvaju na disku kako bi se prilikom pokretanja ponovo učitali. Metode koje definiše klasa ZigBee generički uređaj su:

- `void initialize(const string& _initData);`
 Poziva se prilikom dodavanja uređaja u sistem bilo prilikom ulaska uređaja u mrežu ili učitavanja iz datoteka. Metoda rasčlanjuje XML i popunjava mapu klastera na krajnjim tačkama. Nakon toga vrši inicijalna podešavanja za klastere i dobavljanje trenutnih vrednosti sa uređaja.
- `string getInitData();`
 Metoda formira XML sa informacijama koje se čuvaju u datoteci a neophodne su za inicijalizaciju uređaja prilikom ponovnog podizanja aplikacije. To su podaci o tome koje klastere uređaj podržava na datoj krajnjoj tački.
- `void setCluster(unsigned short _usProfileId, unsigned char _ucEndPoint, unsigned short _usClusterId);`
 Na osnovu profila određuje se fabrika klastera koja će biti korišćena da se na osnovu identifikatora klastera dobavi odgovarajući klaster i ubaci u mapu klastera.

Ovo su najbitnije metode koje se koriste od strane referentnog sistema. Bitno je napomenuti da su prve dve metode definisane u okviru klase uređaja koja objedinjuje sve zajedničke funkcionalnosti za pametne uređaje u sistemu nezavisno od protokola komunikacije.

5. Testiranje i rezultati

Programsko rešenje je testirano u dva aspekta. Prvi je usklađenost sa HA 1.2 specifikacijom a drugi su performanse rešenja. Zahteve koje programsko rešenje mora da zadovolji smo definisali u okviru koncepta. Zahtevi su upoređeni sa rezultatima testiranja i izvršena je provera dobijenih rezultata.

5.1 HA 1.2 testiranje

HA 1.2 specifikacija definiše skup testova koje uređaj treba da prođe da bi se potvrdilo da je uređaj u skladu sa specifikacijom. Testovi su definisani u oficijalnom dokumentu od strane ZigBee alijanse. Kako referentno rešenje koristi već implementiran ZigBee stek na modulu CC2530 i kako je taj modul već poročao platformske testove vezane za sam ZigBee protokol. Ti testovi nisu morali ponovo da se prolaze. Bilo je dovoljno testirati samo aplikativni nivo koji smo mi implementirali tačnije klijentske i serverske klastere koje centralni kontroler podržava. Testovi se sastoje od testiranja funkcionalnosti klastera definisani specifikacijom i funkcionalne testove koji su uglavnom realizovani na nivou ZNP modula i koji bi trebalo da prođu uspešno ali će, zbog testiranje funkcionalnosti celog uređaja, biti ponovo pokrenuti.

Za testiranje centralnog kontrolera koristili smo alat koji je priznat od strane ZigBee alijanse pod nazivom Tracer. U okviru ovog alata dolazi usb dongla, programska podrška i aplikacija. U okviru aplikacije moguće je odabrati tip uređaja koji se testira i navesti njegove karakteristike i klastere koje uređaj podržava. Kada izaberete funkcionalnosti koje uređaj podržava alat će formirati skup testova koji za taj konkretni uređaj mogu da budu pokrenuta. U okviru ovog alata neki testovi su izostavljeni pa smo to testirali pomoću scripta koje su izvršavane na nekom drugom računaru sa ZNP modulom u vidu usb dongle. Testovi su uglavnom bili automatski ali bilo je i testova koji su zahtevali korisničku interakciju. Uspešnost testova je proveravano pomoću alata za nadgledanje protokola(eng. Sniffer). Na osnovu datoteke

evidencije koju je formirao alat za nadgledanje protokola i testne specifikacije utvrđivana je uspešnost izvršenih testova. U tabeli 1 se nalazi spisak izvršenih testova i njihovi rezultati.

Broj testa	Naziv	Uspešnost
3.1.1	Uređaj koji se testira kao kordinator	Prošao
3.3.2	Sigurnosna mogućnost: uređaj koji se testira kao centar od poverenja	Prošao
3.5.1	Otkrivanje komandi(server)	Prošao
3.5.3	Otkrivanje atributa proširen(server)	Prošao
3.6.1	Test topologije – krajnji uređaj	Prošao
3.6.2	Test topologije – kordinator	Prošao
3.7.5	Test bajndovanja	Prošao
3.7.6	Veličina bajnd tabele	Prošao
5.1.1	Osnovni klaster atributi	Prošao
5.1.2	Osnovni klaster(server)	Prošao
5.4.1	Identifikacija klaster atributi	Prošao
5.4.2	Identifikacija klaster(server)	Prošao
5.4.3	Identifikacija klaster(klijent)	Prošao
5.7.2	Uključi Isključi klaster(klijent)	Prošao
5.9.2	Kontrola nivoa klaster(klijent)	Prošao
5.16.2	Kontrola prozivke klaster(klijent)	Prošao
5.25.2	Sigurnosne zone klaster(klijent)	Prošao
6.1.1	Otvaranje mreže	Prošao
8.1.1	Skeniranje i formiranje mreže	Prošao
9.1.1	Pronalaženje i bajndovanje: Uređaj koji se testira kao meta	Prošao

Table 1: Lista urađenih testova

Urađeni testovi potvrđuju niz kompleksnih funkcionalnosti koje pokazuju da je uređaj kompatibilan sa drugim uređajem koji podržava testiranu funkcionalnost. Testovi proveravaju celokupnu funkcionalnost klastera kao što su mogućnost obrađivanje i slanje obaveznih i

opcionih komandi sa korektno postavljenim parametrima, čitanje i pisanje opcionih i obaveznih atributa kao i slanje izveštaja o promeni stanja.

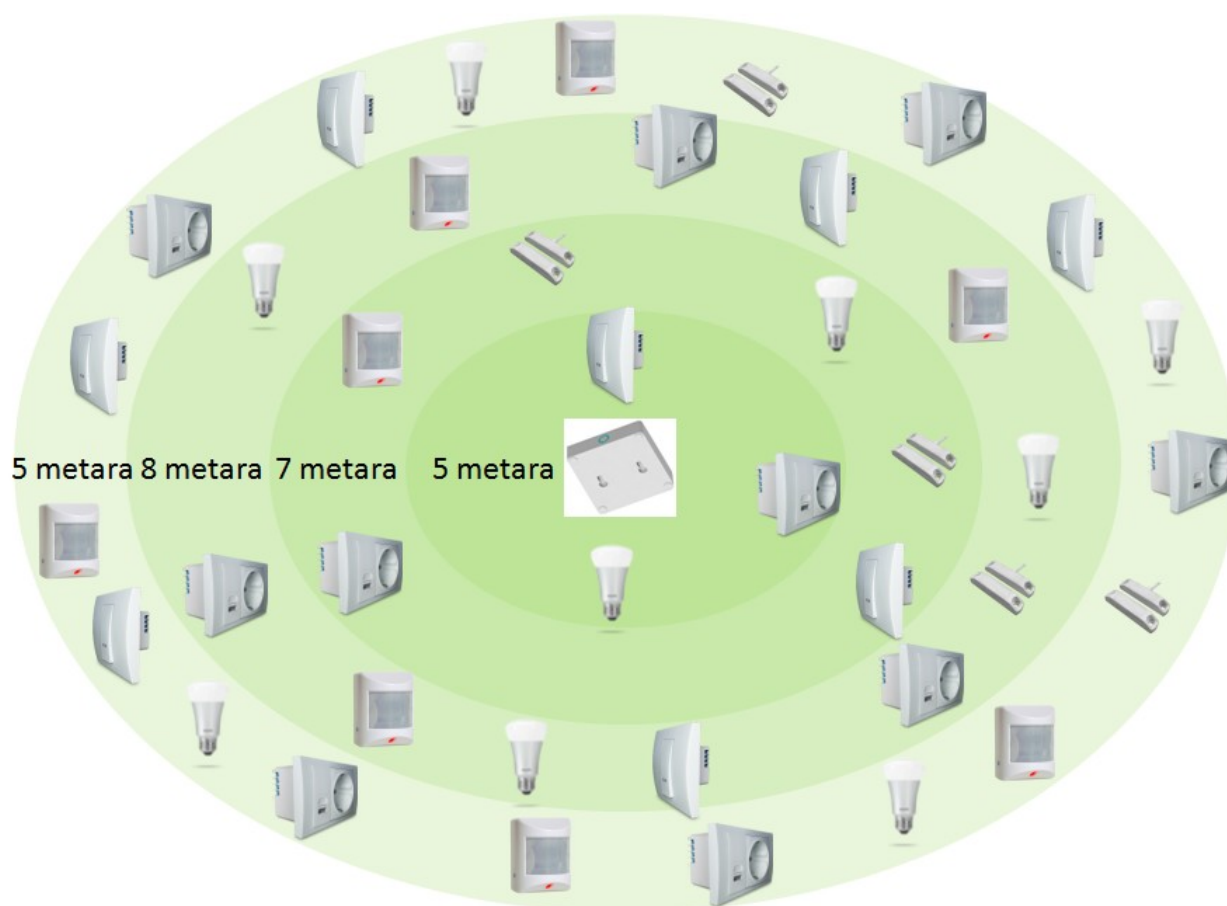
Pošto je centralni kontroler uspešno prošao sve testove koji su zahtevani HA 1.2 specifikacijom možemo da kažemo da je on u skladu sa istom i da je kompatibilan sa drugim uređajima koji su sertifikovani.

5.2 Testiranje performansi

Usklađenost sa HA 1.2 specifikacijom nam ne garantuju dobro korisničko iskustvo nego samo kompatibilnost sa drugima pa smo odlučili da proverimo da li centralni kontroler zadovoljava zahteve u performansama koji su definisani u konceptu rešenja.

Parametri koji su mereni ovim testovima su maksimalan broj uređaja koji je podržan, brzina odziva u zavisnosti od broja uređaja i broja uređaja na putanji do ciljnog kao i stres testovi u vidu velikog broja korisničkih zahteva koji se upućuju ZigBee programskoj podršci.

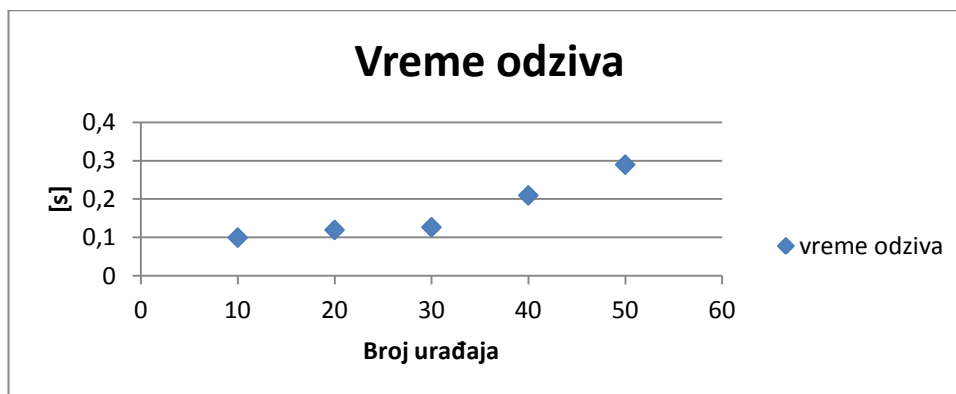
Topologija mreže (slika 16) na kojoj je izvršeno testiranje se sastojala od usmerivačkih uređaja koji su bili postavljeni u nekoliko prstena oko centralnog kontrolera. Prvi krug uređaja je bio postavljen na udaljenost od 5 metara od centralnog kontrolera, sledeći krug na 7 zatim na 8 i poslednji krug je bio na 5 metara udaljenosti od prethodnog. Najudaljeniji uređaj nalazio se na 25 metara od centralnog kontrolera. Uređaji koji su korišćeni za testiranje bili su HA 1.2 sertifikovani. Pored njih korišćene su i ZLL sijalice kako bi se proverila kompatibilnost sa drugim protokolima. Takođe, korišćeni uređaji su bili od različitih proizvođača. Neki od uređaja koji su korišćeni bili su Philips i Osram sijalice, Oblo dimmer i outlet i Netwoks senzorski uređaji.



Slika 16: Topologija testnog okruženja

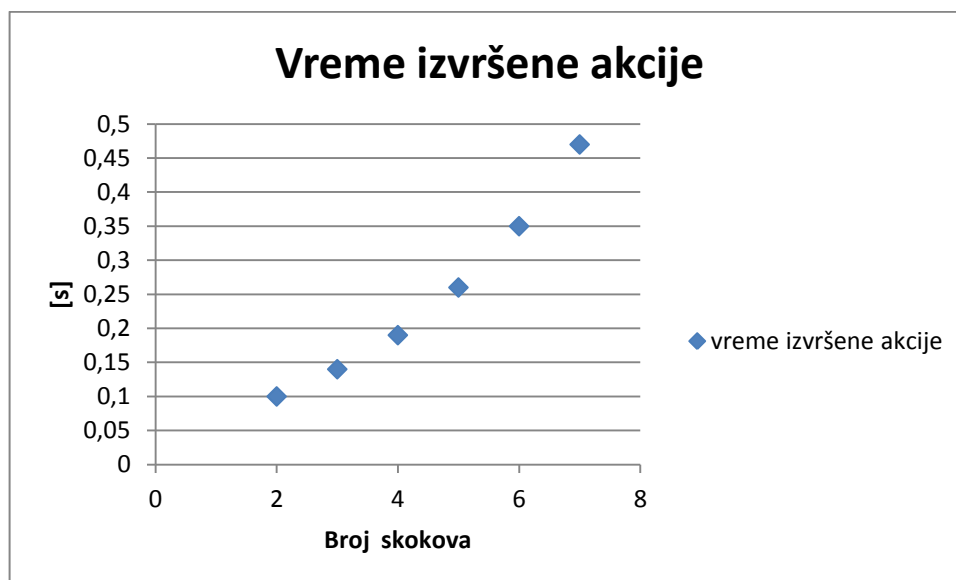
Maksimalan broj uređaj je ograničen kako mogućnostima ZigBee protokola tako i količinom memorije na centralnom kontroleru. Protokol ne ograničava broj uređaja tačnije broj uređaja je ograničen veličinom mrežne adrese koja je dva bajta. Sa druge strane izabrani centralni kontroler ima ograničenu količinu memorije koja iznosi 64 megabajta i to ograničava broj uređaja na 50. S obzirom da smo u konceptu rešenja a na osnovu istraživanja došli do zaključka da je 50 uređaja dovoljan prosečnim korisnicima možemo reći da centralni kontroler zadovoljava ovaj parametar performansi.

Drugi parametar koji smo merili jeste zavisnost odziva od broja uređaja koji se nalaze u mreži. Rezultati su prikazani grafiku 1. Na x-osi se prikazuje broj uređaja dok se na y-osi prikazuju vremena odziva u sekundama. Na grafiku se može primetiti da do 30 uređaja u mreži nema drastičnih promena u odzivu sistema dok preko 30 uređaja odziv je malo skočio ali opet je u granicam očekivanog odziva do 0.5 sekundi. Zahtevana vremena su empirijski utvrđeno na osnovu korisničkog iskustva. Možemo reći da je referentni sistem u granicama očekivanog.



Grafik 1: Vreme odziva

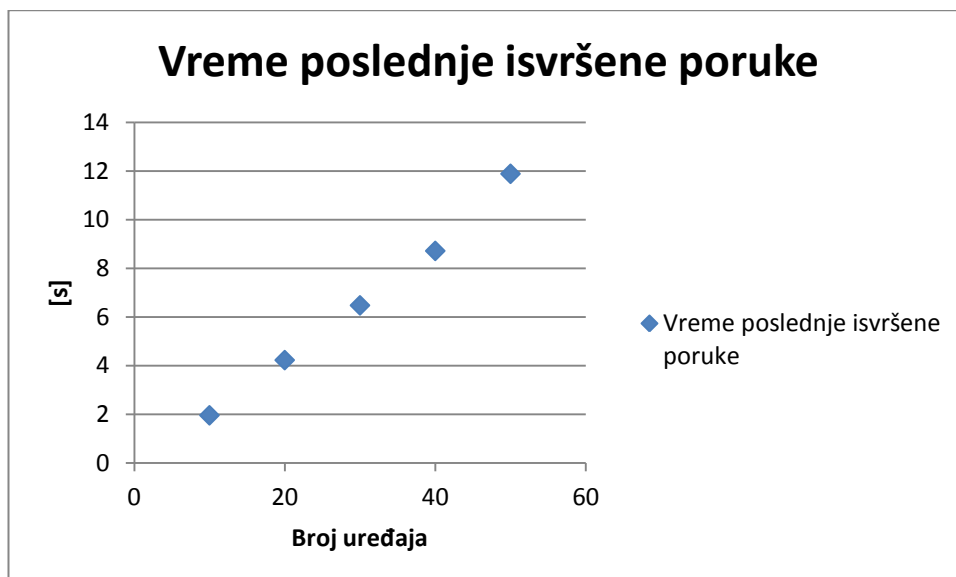
Sledeći parameter koji je meren je koliko kašnjenje unose skokovi u mreži. Merenja su izvršena sa usmerivačkim uređajima koji su postavljeni u liniji i bez prepreka. Cilj ovog merenja je bio da proverimo da li skokovi unose kašnjenje u sistem i koliko. Rezultati će biti prikazani na sledecem grafiku 2.



Grafik 2: Vreme izvršene akcije

Sa grafika se vidi da vreme isporuke raste eksponencijalno sa brojem uređaja koji se nalazi na putu do krajnjeg uređaja. Vreme isporuke poruke krajnjem uređaju kod najvećeg broja skokova je manje od očekivanih 0.5 sekundi tako da centralni kontroler zadovoljava i ovaj kriterijum.

Ostalo je još da vidimo kako će se sistem ponašati u stresnim situacijama kada se istovremeno šalje veliki broj poruka. Kako se istovremeno može obrađivati jedan zahtev od strane korisnika po uređaju i kako je maksimalan broj uređaja 50 testiraće se slanje velikog broja poruka u malim vremenskim intervalima. Rezultati meranje su prikazani na sledecem grafiku.



Grafik 3: Vremena isoruke poslednje poruke

Vreme isporuka poslednje poruke raste linearno sa brojem poruka i u najvećem mogućem broju poruka poslatih od jednom ne prelazi 12 sekundi. Ovakvo vreme je očekivano s obzirom na količinu podataka koji treba poslati kroz mrežu. Sa strane korisničkog iskustva ovo je zadovoljavajuće vreme.

6. Zaključak

Rezultati testiranja su pokazali da realizovano rešenje zadovoljava korisničke zahteve i da je u skladu sa HA 1.2 specifikacijom što je i bio cilj ovog rada. Kao glavne prednosti ovog rešenja su njegova fleksibilnost, laka proširivost novim funkcionalnostima i uređajima, kompatibilnost sa drugim HA 1.2 uređajima i jednostavnost korišćenja. Ubrzanim razvojem tehnologije u oblasti kućne automatizacije i pojave velikog broja rešenja i uređaja koji se primenjuju u kućnoj automatizaciji uslovio je razvoj ovakve programske podrške. Ovim rešenjem pruža se mogućnost jednostavnog dodavanja i kontole uređaja nezavisno od proizvođača.

Usklađenost sa specifikacijom omogućava kompatibilnost sa velikim broje uređaja koji su trenutno dostupni dok će jednostavnost korišćenja povećati broj korisnika ovog sistema. Performanse koje su izmerene u merenjima garantuju vrhunsko korisničko iskustvo. U poređenju sa drugim rešenjima realizovano rešenje je znatno jednostavnije i fleksibilnije rešenje. Jednostavna proširivost ovog sistema novim funkcionalnostima je još jedna bitna prednost ovog rešenja zbog brzog razvoja novih uređaja i funkcionalnosti u oblasti kućne automatizacije.

Dalja poboljšanja i usavršavanja ovog rešenja ići će u pogledu prelaska na ZigBee 3.0 specifikaciju koja ukida profile i uvodi drugačiju organizaciju funkcionalnosti. Svakako prelaskom na novu specifikaciju zadržaće se kompatibilnost sa starom specifikacijom. Prelazak na novu specifikaciju će zahtevati i promenu ZNP modula koji realizuje protokol stek.

7. Literatura

- [1] Ilija Bašičević, Miroslav Popović, Vladimir Kovačević: *Osnove računarskih mreža I*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2013
- [2] Miroslav Popović: Inženjering komunikacionih protokola, eng. *Communication Protocol Engineering*, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2006
- [3] 802.15.4 standard, <https://standards.ieee.org/getieee802/download/802.15.4-2011.pdf>, jul 2016
- [4] ZigBee, <http://www.zigbee.org/zigbee-for-developers/network-specifications/>, jul 2016
- [5] WPAN, <https://cs.wmich.edu/~alfuqaha/Fall09/cs6030/lectures/WPAN.pdf>, jul 2016
- [6] Specifikacija ZigBee klaster biblioteke, eng. *ZigBee Cluster Library Specification*, Joson Choong, David Clark, Ettore Colicchio, Kent Crouse, http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2011/kjb79_ajm232/pmeter/ZigBee%20Cluster%20Library.pdf, maj 29, 2008
- [7] Arhitektura protokola I topologija senzorskih ZigBee mreža, Vanja Eličić, Nada Basiljević; konferencija Infoteh, Jahorina, mart 2010
- [8] *ZigBee Application Profile*, Phil Jamieson, ZigBee Open House, Beijing, China, septembar 13, 2007
- [9] Specifikacija ZigBee profila za kućnu automatizaciju, eng *ZigBee Home Automation Public Application Profile*, Don Sturek, Mads Westergreen, Tim Hirou, Bob Old, Tim Gilman, <https://docs.zigbee.org/zigbee-docs/dcn/07/docs-07-5367-02-0afg-home-automation-profile-for-public-download.pdf>, februar 8, 2010
- [10] *Unified Device Access in Home Automation Enviroment*, Milan Pandurov, Ivan Lazarević, Nikola Smiljković, konferencija Telfor 2014, strana 971-974
- [11] *Poco C++* biblioteka, <http://www.pocoproject.org/>, jul 2016

-
- [12] Fabrika obrazac, <http://www.oodesign.com/factory-pattern.html>, jul 2016
 - [13] Protorup obrazac, <http://www.oodesign.com/prototype-pattern.html>, jul 2016
 - [14] *An approach for uniform representation and control of ZigBee devices in home automation software*, Vladimir Moravčević, Milan Tucić, Roman Pavlović, Aleksandar Majdak, konferencija ICCE Berlin 2015, strana 237-239