



**УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**



**УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД**

Департаман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR)РАД

Кандидат: Ненад Врга

Број индекса: 12555

Тема рада: Управљање кућним уређајима преко HTTP сервера базирано на XMOS платформи

Ментор рада: др Јелена Ковачевић

Нови Сад, 2013



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:	Завршни (Bachelor) рад
Аутор, АУ:	Ненад Врга
Ментор, МН:	др Јелена Ковачевић
Наслов рада, НР:	
Језик публикације, ЈП:	Српски / латиница
Језик извода, ЈИ:	Српски
Земља публикавања, ЗП:	Република Србија
Уже географско подручје, УГП:	Војводина
Година, ГО:	
Издавач, ИЗ:	Ауторски репринт
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО: <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	
Научна област, НО:	Електротехника и рачунарство
Научна дисциплина, НД:	Рачунарска техника
Предметна одредница/Кључне речи, ПО:	
УДК	
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	
Датум прихватања теме, ДП:	
Датум одбране, ДО:	
Чланови комисије, КО:	Председник:
	Члан:
	Члан, ментор: др Јелена Ковачевић
	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 :	Bachelor Thesis
Author, AU :	Nenad Vrga
Mentor, MN :	Jelena Kovacevic, PhD
Title, TI :	
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 :	
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President:
	Member:
	Member, Mentor: Jelena Kovacevic, PhD
	Mentor's sign

Zahvalnost

Zahvaljujem se Institutu RT-RK na pruženoj mogućnosti za realizaciju ovog rada kao i mentoru dr Jeleni Kovačević, stručnom saradniku Nenadu Četiću na savetima i stručnoj pomoći.

SADRŽAJ

1. Uvod	1
2. Teorijske osnove.....	2
2.1 HTTP.....	2
2.2 HTML И HTML5	3
2.3 xCORE mikrokontroler.....	4
2.4 UDP.....	5
2.5 Dart	6
2.6 PWM.....	6
3. Koncept rešenja.....	8
3.1 Klijentske veb stranice	10
3.2 XMOS Server.....	14
3.3 XMOS Uređaj	15
4. Programsko rešenje.....	16
4.1 Modul app_smarthouse_server	17
4.2 Modul app_udp_light.....	19
4.3 Modul module_smarthouse_webserver	19
4.4 Ostali korišćeni moduli	20
4.5 Klasa Wall.....	21
5. Testiranje i rezultati	22
6. Zaključak	24
7. Literatura	25

SPISAK SLIKA

Slika 2.1 xCORE arhitektura	4
Slika 2.2 Struktura datagrama protokola korisničkih datagrama.....	5
Slika 2.3 Analogni signal i rezultujući PWM signal	7
Slika 3.1 SliceKit razvojna ploča.....	8
Slika 3.2 WiFi dodatak	8
Slika 3.3 Ethernet dodatak	9
Slika 3.4 GPIO dodatak	9
Slika 3.5 XTAG2 JTAG adapter.....	10
Slika 3.6 Blok dijagram od Darta do pretraživača.....	10
Slika 3.7 Stranica plana kuće	11
Slika 3.8 format HTTP poruke sa POST metodom i kordinatama zidova	11
Slika 3.9 Stranica za postavljanje uređaja	12
Slika 3.10 Prozor sa podešavanjima za uređaj.....	12
Slika 3.11 format HTTP poruke sa POST metodom i podešavanjima uređaja	13
Slika 3.12 Dijagram klasa.....	13
Slika 3.13 MSC dijagram.....	14

SKRAĆENICE

HTML	- <i>HyperText Markup Language</i> , je jezik za označavanje хипер текста
HTTP	- <i>HyperText Transfer Protocol</i> , протокол за пренос HTML страница
TCP/IP	- <i>Transmission Control Protocol/IP protocol suite</i> , скуп протокола који омогућавају умреженим рачунарима да деле ресурсе путем мреже
PWM	- <i>Pulse-width modulation</i> , импулсно ширинска модулација
WIFI	- <i>Wireless-Fidelity</i> , бежићна локална рачунарска мрежа
WebGL	- <i>Web Graphics Library</i> , je biblioteka koja unapređuje sposobnosti JavaScript-ai omogućava pretraživačima prikaz interaktivnih 3D animacije
OSI	- <i>Open Systems Interconnection Basic Reference Model</i> , референтни модел за отворено повезивање система

1. Uvod

U radu je prikazan primer korišćenja XMOS platforme za upravljanje kućnim uređajima. Samo upravljanje se vrši preko HTTP servera, odnosno preko bilo kog uređaja koji poseduje Veb pretraživač. Glavna ideja rada je da pokaže da je moguće obezbediti samostalno upravljanje uređajima u pametnoj kući koristeći XMOS kontrolere. Potrebno je imati jedan XMOS čip kao server i onoliko XMOS čipova koliko ima uređaja nad kojima želimo da vršimo upravljanje. Svaki uređaj koji je povezan sa XMOS čipom komunicira sa serverom i preuzima svoja podešavanja. Korisniku samo ostaje da preko veb stranice iscrta plan kuće, postavi uređaje u planu kuće i zada podešavanja.

Ideja o daljinskom upravljanja uređajima u domaćinstvu odnosno koncept pametne kuće postoji već više decenija. Ideja je zaživela poslednje decenije jer se ceo koncept zasniva na podizanju kvaliteta življenja i postizanje energetske efikasnosti. Pravilnom kontrolom osvetljenja, grejanja, klimatizacije i ventilacije, postizemo uštede energije, što znači da pametna kuća nije samo praktična već i ekonomična.

Rad je organizovan u nekoliko celina:

- Teorijske osnove – teorijski objašnjava gradivne elemente ovog rada.
- Koncept rešenja – opisuje postupak rešavanja glavnih zadataka, predstavljenih u ovom radu.
- Programsko rešenje – daje opis svih funkcionalnih modula sa najvažnijim funkcijama.
- Testiranje i rezultati – daje opis testiranja i rezultata.
- Zaključak – daje pregled urađenog i mogući predlozi za unapređenje rada.

2. Teorijske osnove

2.1 HTTP

HTTP je protokol za komunikaciju između servera i klijenta, koji funkcioniše po principu zahtev / odgovor. HTTP klijent, koji je najčešće veb pretraživač, inicira prenos podataka nakon što uspostavi TCP / IP vezu s udaljenim veb serverom na određenom portu.

Server konstantno osluškuje zahteve na određenom mrežnom komunikacijskom portu (tipično port 80), čekajući da se klijent poveže i pošalje svoj zahtev . Zahtev se sastoji od osnovne HTTP komande i zaglavlja. Sintaksa komande je propisana standardom i sastoji se od naziva komande, imena traženog dokumenta i verzije podržanog HTTP- a. Zaglavlje se sastoji od određenog broja redova teksta koji određuju aspekte zahteva.

Zahtev klijenta se obrađuje na serveru i u zavisnosti od ispravnosti zahteva i mogućnosti zadovoljavanja istog. Klijentu se šalje odgovarajući odgovor. Odgovor se sastoji od izveštaja o statusu zahteva i od konkretnog odgovora, ukoliko je zahtev moguće zadovoljiti. Status se sastoji od trocifrenog koda i kratkog deskriptivnog teksta statusa, npr 200 OK. Odgovor se sastoji od zaglavlja, koje je iste sintakse kao i zaglavlje zahteva i daje osnovne podatke o prirodi odgovora, i od eventualnog konkretnog sadržaja koji se tražio u zahtevu. U zavisnosti od verzije HTTP protokola kao i od zaglavlja zahteva, veza se može nakon toga prekinuti, a može se ista veza iskoristiti za slanje novog zahteva, radi uštede vremena. [1]

2.2 HTML I HTML5

Jezik za označavanje hiperteksta (*engl. Hypertext Markup Language*), poznatiji kao HTML, je jedan od glavnih računarski jezika na *World Wide Web* globalnoj računarskoj mreži. HTML nije programski jezik, nego jezik za označavanje teksta. Teoretski, internet stranicu koju napravite koristeći HTML mogao bi da vidi svako ko ima računar, bilo kakav veb pretraživač (*engl. Web browsers*, u daljem tekstu pretraživač) i pristup Internetu. U praksi, mogućnost da se pregleda potpun sadržaj stranice zavisi od karakteristika pretraživača. Pretraživači su programi koji prevode HTML u dokumentima veb stranica i prikazuju tekst, slike i animacije na ekranu računara posetioca. Da bi prikazao veb stranicu, pretraživač prevodi HTML kod. Verzija pretraživača koji posetioci vaše veb lokacije koriste određuju koje će karakteristike HTML-a kod njih raditi. To znači da od verzije pretraživača zavisi šta će posetilac imati na ekranu. Tako korišćenje najnovijih, najnaprednijih odlika HTML-a i njegovih mogućnosti formatiranja strana ne garantuje da će svi koji pogledaju stranicu videti sve što ste na nju stavili.

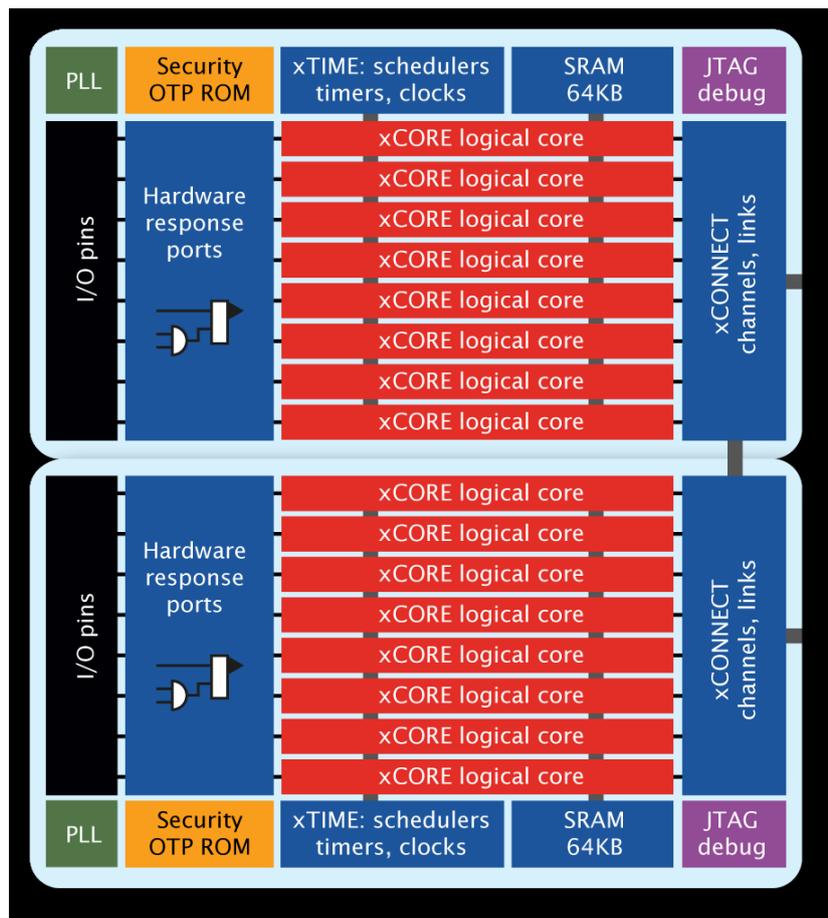
U osnovi, veb stranica je tekstualna datoteka u kojoj se nalaze naredbe u obliku HTML kodova (oznaka) i atributa. Oznake su naredbe koje pretraživač izvršava da bi formatirao tekst i umetnuo slike na stranicu. Neke naredbe u HTML-u moraju da imaju i početnu i završnu oznaku. Takve naredbe se nazivaju uparene oznake (*engl. container tags*), jer se naredba iz početne oznake odnosi na sadržaj stranice sve do završne oznake. Svaka oznaka u HTML-u počinje znakom manje (<) iza čega se navode naziv oznake i atributi, a završava se znakom veće (>). Završna oznaka se razlikuje od početne oznake po kosoj crti (/) ispred samog naziva oznake. Tako su početne oznake oblika <naziv oznake[atributi]>, a završne </naziv oznake>. Naziv oznake veb pretraživaču otkriva namenu te oznake, dok atributi ukoliko postoje daju pretraživaču dodatne informacije pomoću kojih on izvršava naredbu te oznake. [2]

HTML5 je predlog novih standarda koji nam pružaju nove mogućnosti, kao i da isprave i nadomeste neke dosadašnje nedostatke. Radi se o novoj verziji HTML, koja donosi niz novina i mogućnosti koje do sad nisu bile dostupne za izradu veb sajtova bez dodatnih dodataka (*engl. plugin*), kao što su *Flash*, *Java* i *Silverlight*. HTML5 predstavlja evoluciju HTML 4.01 standarda, a ne zamenu. HTML5 je još uvek u fazi razvoja, a standardizacija se očekuje 2014. godine. Kao peta verzija HTML osnovni cilj je bio unapređenje ovog jezika, kao i podrška za multimedijalni sadržaj. Pravljen je za različite uređaje (*engl. cross-platform*), tako da nije bitno da li se dokument pregleda pomoću tableta, pametnog telefona (*engl. smartphona*), laptopa, sve dok postoji podrška za HTML5 na pretraživaču.

Do nedavno, veb programeri su bili ograničeni na CSS i JavaScript kod stvaranja animacije i vizualnih efekata za svoje veb stranice ili prinuđeni da koriste dodatke kao što je *Flash*. Sa dodatkom novih tehnologija kao što su *platno* (engl. *Canvas*), *WebGL* i srazmerna vektorska grafika (engl. *Scalable Vector Graphics*) slike, više nema potrebe za dodacima. [3]

2.3 xCORE mikrokontroler

xCORE uređaji imaju više logičkih procesorskih jezgara (engl. *logical processor core*), tako da mogu da izvršavaju nekoliko zadataka istovremeno i nezavisno. Procesorska jezgra su raspoređena po određenom broju pločica (engl. *tile*). Poseduju fleksibilane ulaze - izlaze i jedinstvenu vremensko determinističku arhitekturu, kao i tehnologiju koja stvara vezu između jezgra kao i spoljašnjeg sveta. Ovo smanjuje latenciju i omogućava xCORE uređaju da odgovori na spoljne događaje u desetinama nanosekundi što je 100x brže nego kod tradicionalne mikrokontrolera. Logička procesorska jezgra komuniciraju međusobno preko kanala (engl. *channel*).



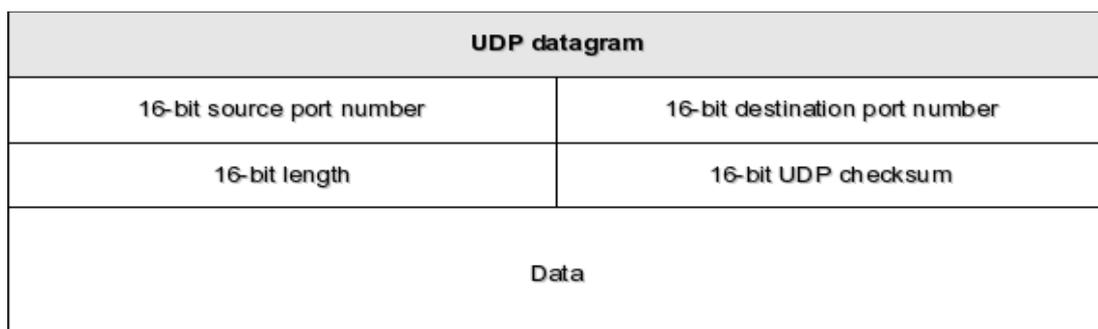
Slika 2.1 xCORE arhitektura

Svako jezgro može da izvršava razne matematičke operacije, kontrolne zadatke (kao što su logičke odluke i automati stanja) ili obradu ulaza - izlaza. xCORE mikrokontroler obezbeđuje 500 MIPS procesne moći (na uređaju radne učestanosti 500 MHz) čineći ga tako mnogo moćnijim od konvencionalnih mikrokontrolera. Logička procesorska jezgra koja se nalaze na istoj pločici, dele memoriju i druge resurse te pločice, ali svako jezgro ima svoje odvojene registarske informacije. Svako logičko jezgro dobija na raspolaganje garantovani deo procesne moći (do 125 MIPS na uređajima od 500 MHz), što je kontrolisano jedinstvenom tehnologijom, nazvanom xTIME. Vremenski odsečki se računaju na potpuno transparentan način – logička procesorska jezgra izgledaju kao odvojeni, paralelni procesori, sposobni za izvršavanje više zadataka u realnom vremenu, istovremeno. Skoro se sve instrukcije izvršavaju u jednom mašinskom ciklusu, tako da se može predvideti vreme potrebno da se izvrši bilo koji blok koda. Izgled xCORE arhitekture prikazan je na Slici 2.1.

xCORE procesori se programiraju u C/C++, assembleru. Da bi još dodatno pojednostavili korišćenje xCORE mikrokontrolera XMOS je napravio i proširenje C-a koje podržava paralelne zadatke i kontrolu vremena, a nazvao ga je XC. [4]

2.4 UDP

UDP (engl. *User Datagram Protocol*) je jednostavan protokol koji obezbeđuje osnovne funkcije transportnog sloja OSI modela. UDP se koristi za razmenu paketa poruka ("datagrama") između računara. Za razliku od protokola TCP, ovaj protokol ne podrazumeva stalnu vezu nego se paketi "bacaju" odredišnom računaru, bez održavanja veze i provere grešaka. Na taj način, ovaj protokol ne garantuje isporuku paketa niti isti redosled isporuke paketa kao pri slanju. Zbog ovih osobina UDP protokol je brz i koristi se za aplikacije kojima je važna brzina, a prispeće paketa i održavanje redosleda nije od velike važnosti. Koristi ga veliki broj aplikacija, naročito multimedijalne aplikacije poput internet telefonije i video konferencije. [5]



Slika 2.2 Struktura datagrama protokola korisničkih datagrama

Sva polja u zaglavlju datagrama UPR protokola imaju dužinu od 16 bitova a u njih spadaju:

- Broj porta na izvorištu (engl. *Source port number*) - određuje port preko koga se komunikacija vrši na strani pošiljaoca.
- Broj porta na odredištu (engl. *Destination port number*) - određuje port preko koga se komunikacija vrši na strani primaca.
- Dužina (engl. *Length*) - bitska dužina podataka koje datagram nosi.
- Kontrolna suma (engl. *Checksum*) - kontrolna suma zaglavljaja i paketa.

2.5 Dart

Dart je novi programski jezik kojeg je razvila kompanija *Google*. JavaScript je dobar za male skripte, ali ako skripta izraste u veliku veb aplikaciju, traženje grešaka i prepravljanje te aplikacije postaje prava noćna mora.

Sam Dart je projekat sa otvorenim kodom (engl. *open-source*) koji omogućava programeru da izgradi mnogo kompleksnije aplikacije za moderni veb. Korišćenjem Dart jezika moguće je brzo otkucati prototipove koji evoluiraju u hodu, postoji i pristup naprednim alatima, pouzdanim bibliotekama i dobroj prgoramskoj podršci.

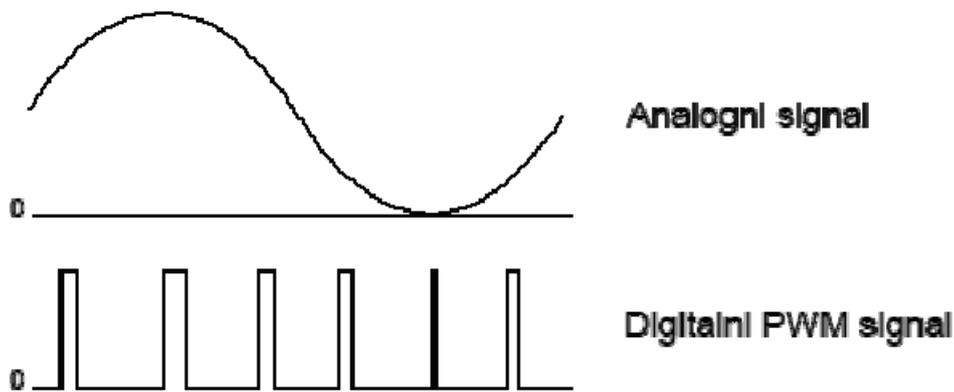
Iako je mlad Dart već ima alate kao što su *Dartboard* (omogućava pisanje i pokretanje Dart koda u pretraživaču) i *Dart Editor* (pomoću kojeg se kreira, prepravlja i pokreće Dart aplikacija). Skoro je izašao i *SDK* koji sadrži alate komandnih linija (engl. *command-line*) kao što su *Dart-to-JavaScript* prevodilac (generiše JavaScript koji može da radi u bilo kom modernom pretraživaču) i *Dart Virtual Machine* (virtualna mašina koja pokreće Dart kod na serveru). Poslednji alat koji je postao dostupan je *Dartium*, koji se prevodi u Dart VM. [6]

2.6 PWM

PWM je tehnika kontrolisanja analognih kola preko digitalnih izlaza procesora (mikrokontrolera). Napon, odnosno struja se dovode do analognog primaoca kao serija impulsa, a informacija o amplitudi analognog signala se predstavlja širinom (trajanjem)

impulsa PWM signala. PWM ima vrlo široku primenu, koristi se pri merenju, komunikaciji, kontroli napajanja, kontroli motora, kao i u raznim vrstama pretvarača.

Analogni naponi i struje se mogu koristiti za direktnu kontrolu uređaja, to je jako jednostavan i jasan način kontrole, ali nije uvek praktičan i isplativ. Analogna kontrola ima razne mane kao što su velike dimenzije komponenti i njihova težina, promenljive karakteristike komponenti tokom vremena, velika potrošnja, grejanje, osetljivost na šum. Digitalnim kontrolisanjem analognih kola, cena sistema za kontrolu i potrošnja energije se mogu drastično smanjiti jer PWM predstavlja ekonomičan, energetske efikasan i imun na šum način za kodiranje analognih signala. Mnogi današnji mikrokontroleri imaju ugrađene PWM periferije što dodatno olakšava implementaciju.

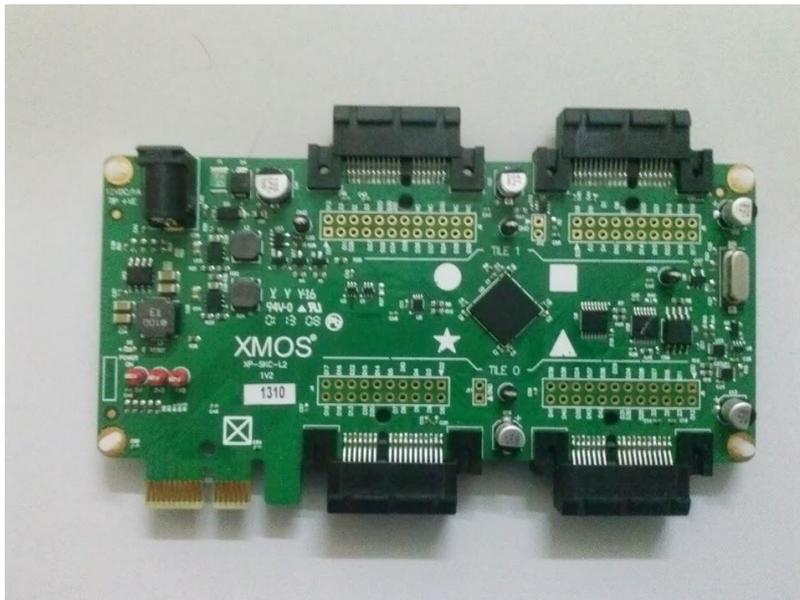


Slika 2.3 Analogni signal i rezultujući PWM signal

Glavna prednosti korišćenja PWM signala je to što signal ostaje digitalan celim putem od procesora do primaoca, tako da konverzija nije potrebna. PWM signal se smatra digitalnim signalom jer je u svakom trenutku vremena nivo signala ili na nuli ili na maksimumu. Korišćenjem čisto digitalnog signala efekat šuma se minimalizuje, što je velika prednost u odnosu na analogni prenos i kontrolu. Šum može da utiče na digitalni signal samo ako je toliko jak da promeni logičku nulu na jedinicu ili obrnuto. Zbog toga se PWM u nekim slučajevima koristi i za komunikaciju.[7]

3. Koncept rešenja

Ovo rešenje je realizovano na *SliceKit* razvojnoj ploči proizvođača XMOS Slika 3.1 koja u sebi ima 2 pločice sa po 8 logičkih procesorskih jezgara na svakoj.



Slika 3.1 SliceKit razvojna ploča

Pored same razvojne poloče korišćeni su i njeni dodaci:



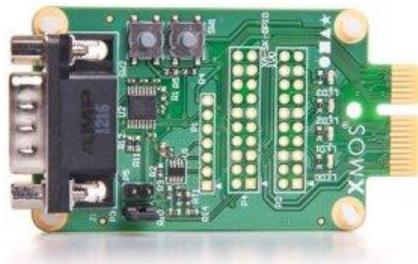
Slika 3.2 WiFi dodatak

- dodatak za bežičnu vezu (Slika 3.2)
 - 802.11b/g
 - 2.4GHz WLAN modul
 - propusna moć do 7Mbps



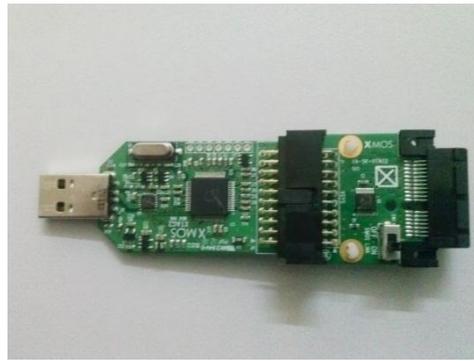
Slika 3.3 Ethernet dodatak

- dodatak za *Ethernet* vezu (Slika 3.3)
 - 10/100Mb
 - MII interfejs za xCORE
 - RJ45 konektor



Slika 3.4 GPIO dodatak

- *GPIO* dodatak sa dodatnim pinovima i LED diodama (Slika 3.4)
 - 4 LED diode
 - 2 tastera
 - Jedan RS232 seriski kabal sa DB9 konektorom
 - 4 kanala A/D preko I2C interfejsa
 - Thermistor input via A/D

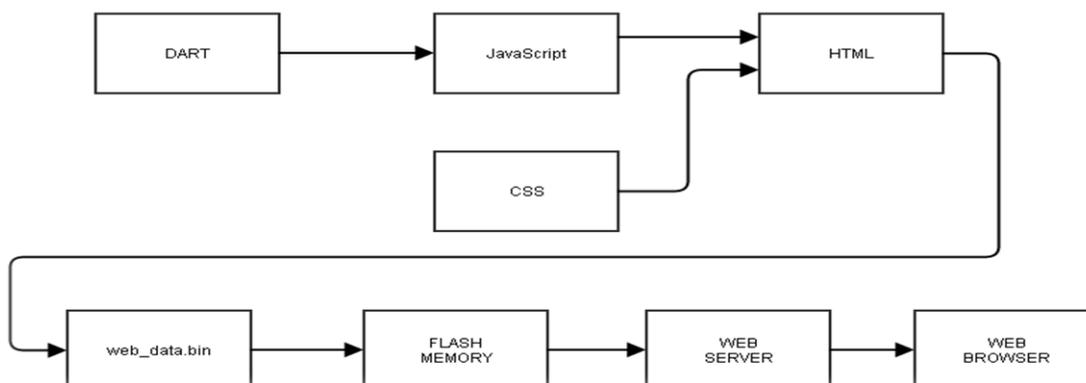


Slika 3.5 XTAG2 JTAG adapter

- otklanjanje grešaka (XTAG2 adapter, slika 3.5)

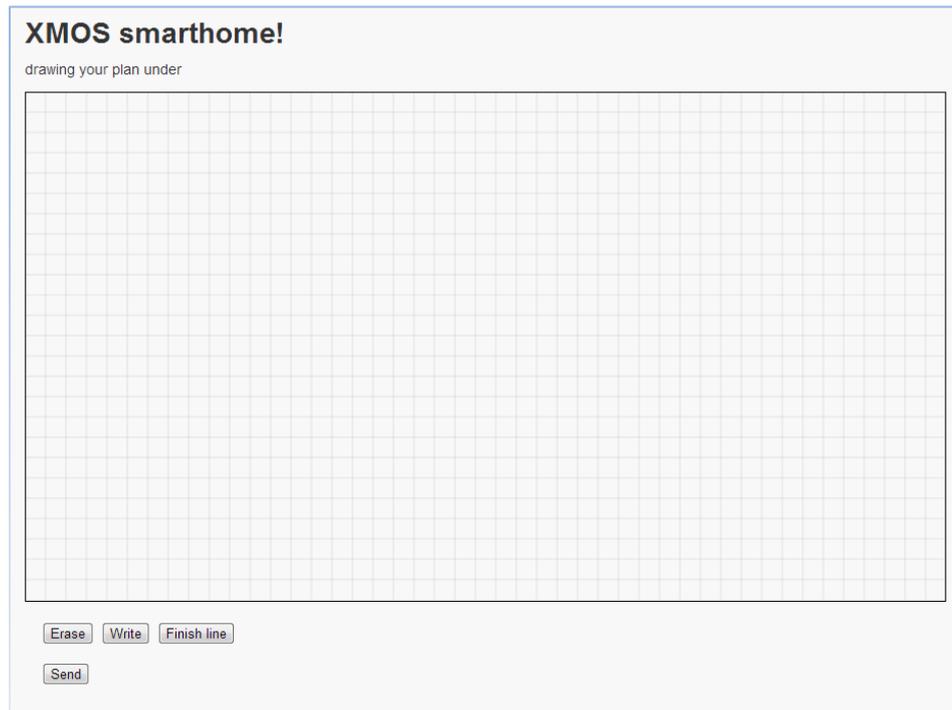
3.1 Klijentske veb stranice

Klijentske veb stranice su rađene u programskom jeziku Dart koji ima mogućnost prevođenja u JavaScript. Pošto pretraživač preko kojeg treba da se vrši iscertavanje plana kuće i podešavanje uređaja ne prepoznaje Dart potrebno je prevesti njegov kod u JavaScript. Preprebacuje se kod prevedenog JavaScripta zajedno sa CSS kodom u HTML. Potom se iz HTML-a generiše binarna datoteka koja se upisuje u *flash* memoriju servera. Kompletan putanja od programa u Dart jeziku do veb pretraživača korisnika prikazana je na blok dijagramu (Slika 3.6).



Slika 3.6 Blok dijagram od Dart-a do pretraživača

Zadatak je da se sa bilo kog uređaja koji ima veb pretraživač može upravljati grejanjem, svetlom i drugim uređajima u domaćinstvu. Potrebno je povezati se sa uređajem pomoću kojeg želimo da vršimo upravljanje na istu mrežu na kojoj je povezan i server. Potom se sa pretraživača šalje HTTP zahtev sa *ip* adresom servera. Kao odgovor od strane servera dolazi početna stranica koja služi za crtanje plana kuće (Slika 3.7).



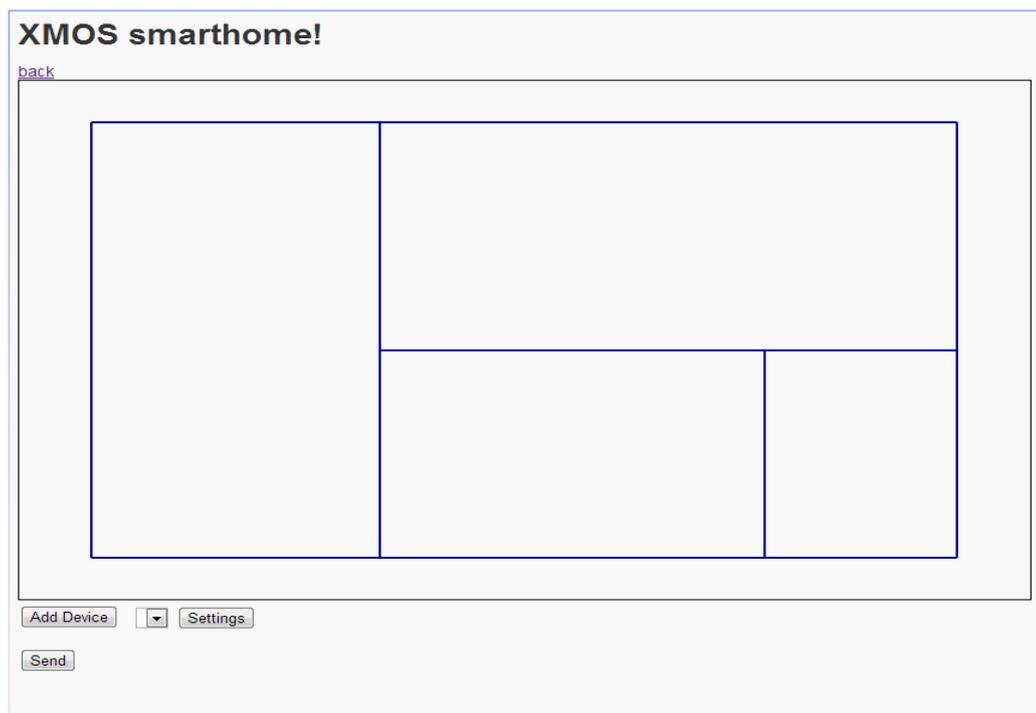
Slika 3.7 Stranica plana kuce

Iscrtavanje se vrši na platnu (engl. *Canvas*) koji je deo HTML5. Pritiskom na platno počinje iscrtavanje zida sve do sledeceg pritiska gde se završava taj zid i od te tačke počinje novi. Moguće je završiti sa crtanjem zida na dva načina, duplim klikom na taster miša ili pritiskom dugmeta *Finish line*. Postoji i opcija brisanja linije do koje se dolazi pritiskom na dugme *Erase*, a zatim selektovanjem zida koji želimo izbrisati. Pritiskom na dugme *Send* formira se HTTP zahtev sa *post* metodom koji šalje poruku sa kordinatama tačaka zida (Slika 3.8). Veb pretraživač počinje sa učitavanjem sledeće stranice koja služi za postavljanje uređaja u predhodno iscrtanom planu kuće (Slika 3.9). Po samom završetku učitavanja stranica šalje serveru HTTP zahtev sa *get* metodom i prima poruku istog formata kao na Slici 3.8. Parsira podarke iz poruke na osnovu kojih iscrtava plan kuće. Postavljanje uređaja se vrši pritiskom na dugme *Add Devices* i na željeno mesto na platnu gde uređaj treba da bude postavljen. Podešavanje samog uređaja je moguće na dva načina duplim klikom na uređaj ili klikom na dugme *Settings* uz predhodno odabran *id* uređaja koji želimo da podesimo u

POST /devices.html HTTP/1.1	wall=5x:180Sy:140Ex:180Ey:360Sx:180Sy:360Ex:320Ey:340Sx:320Sy:340Ex:320Ey:340Sx:320Sy:340Ex:320Ey:340KRAJ
HTTP Headers	body

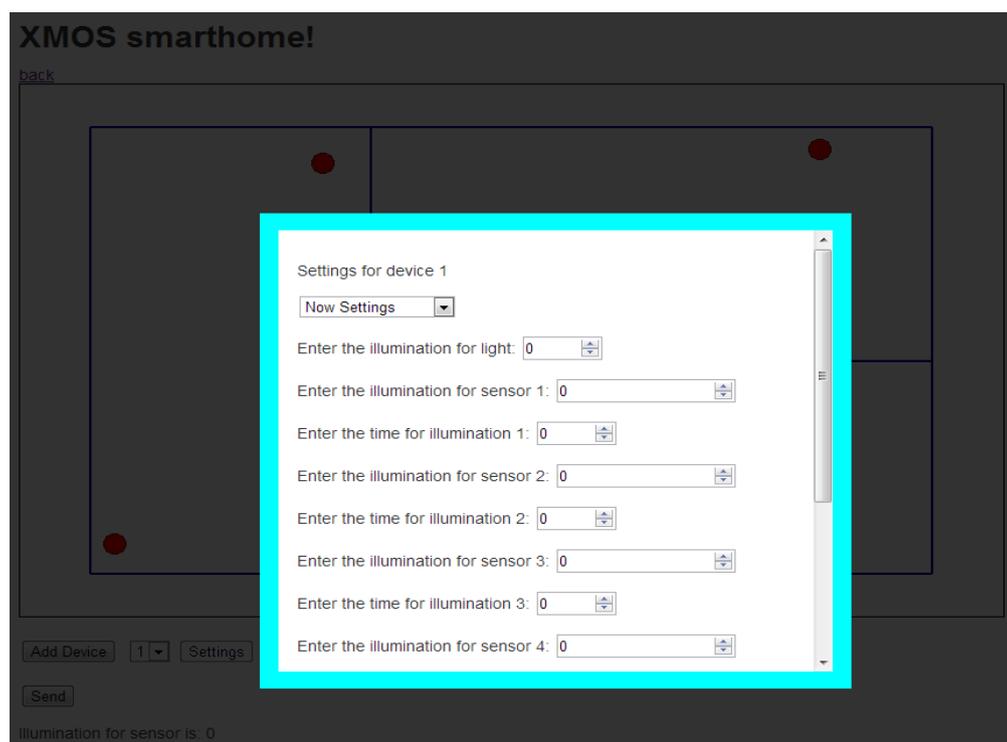
Slika 3.8 format HTTP poruke sa POST metodom i kordinatama zidova

padajućem meniju.

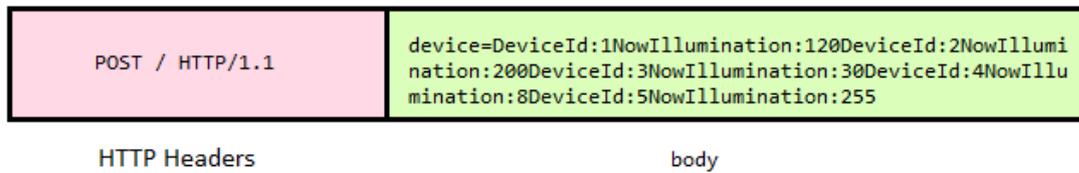


Slika 3.9 Stranica za postavljanje uređaja

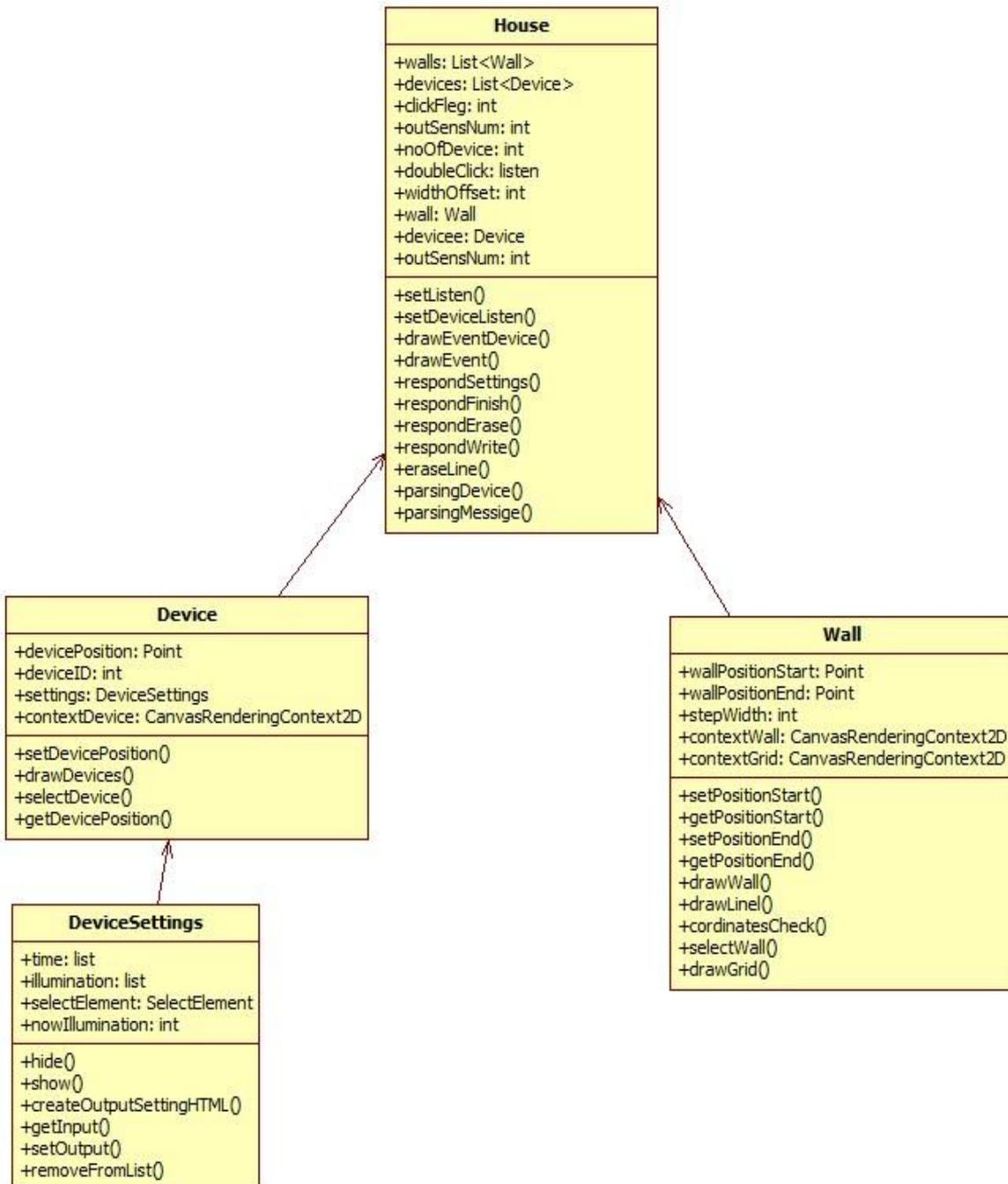
Kao posledica bilo koje od ova dve akcije pojavljuje se novi prozor sa podešavanjima (Slika 3.10). U njemu postoji opcija trenutnog podešavanja i podešavanja za narednih 7 dana koja je ostavljena za unapređenje samog programa. Po završetku podešavanja svih željenih uređaja i snimanja tih podešavanja pritiskom na dugme *Send* vrši se slanje serveru HTTP zahteva sa metodom *post*. Format post zahteva je prikazan na Slici 3.11. Nastavak će biti objasnjen u sledećem poglavlju u okviru samog servera.



Slika 3.10 Prozor sa podešavanjima za uređaj



Slika 3.11 format HTTP poruke sa POST metodom i podešavanjima uređaja

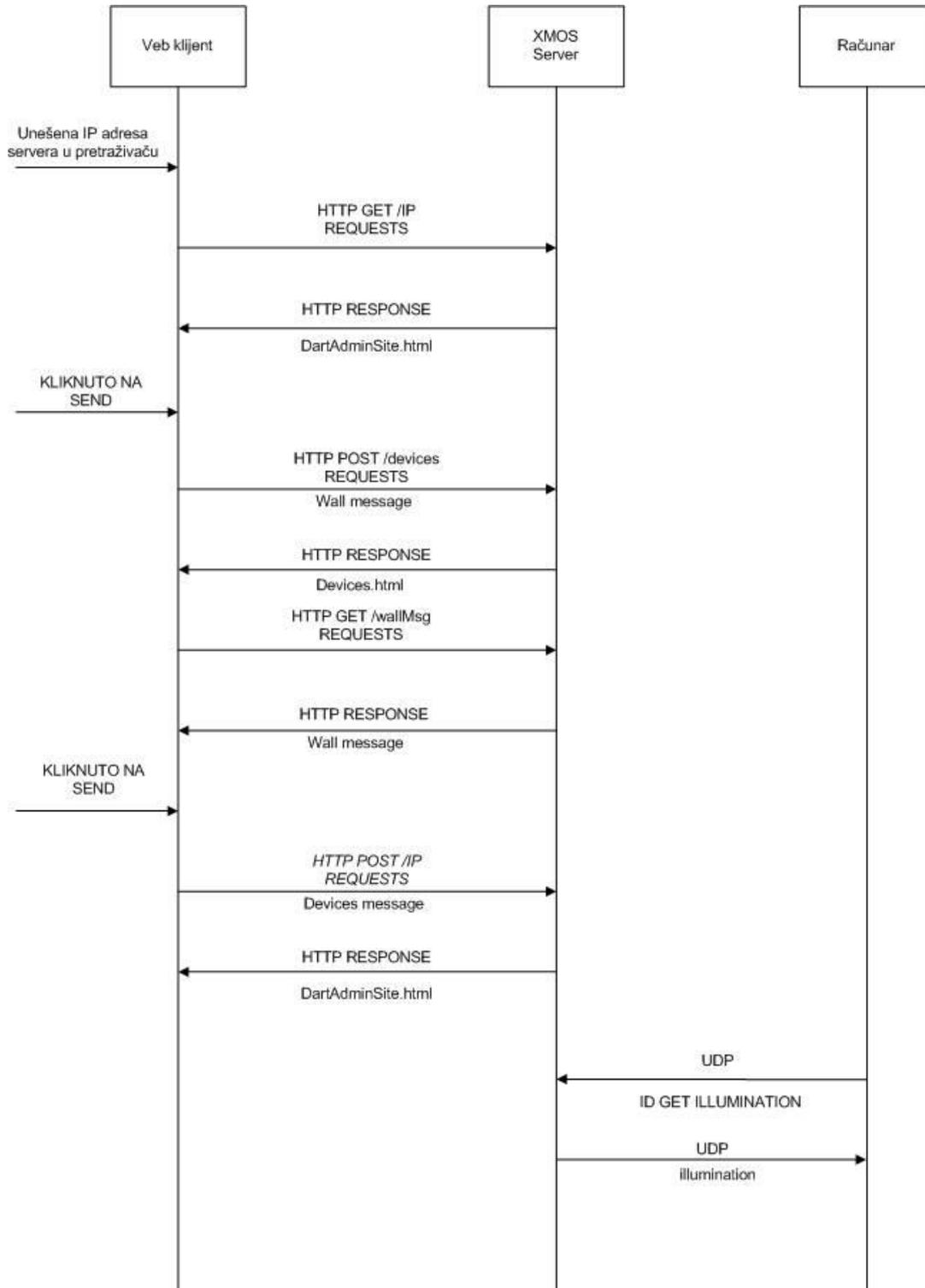


Slika 3.12 Dijagram klasa

Dijagram kalasa kompletnog klijentskog rešenja je prikazan na Slici 3.12.

3.2 XMOS Server

Pri samom startu server vrši se inicijalizacija WiFi dodatka kao i ethernet dodatka. Potom se povezuje sa željenom bežičnom mrežom. Svaki dodatak na ploči dodeljen je jednoj niti. Pločica 0. radi sa WiFi dodatkom kao i njegovim rukovaocem, dok je na pločicu 1. prikazan ethernet dodatak i nit koja obrađuje veb događaje. Sve se to paralelno izvršava.



Slika 3.13 MSC dijagram

Kompletna komunikacija je prikazana na *MSC* dijagramu na Slici 3.13

Prilikom dobijanja prvog HTTP zahteva vrši se njegovo parsiranje i u zavisnosti od metode, kao i adrese zahteva server šalje odgovarajući odgovor. Tako prilikom dobijenog HTTP zahteva sa *get* metodom i *ip* adresom samog servera on posle parsiranja odgovara sa stranicom za iscrtavanje plana kuće. Po prijemu HTTP zahteva *post* metode sa kordinatama zidova vrši se njeno parsiranje i čuvanje u baferu. Kao odgovor server šalje korisniku stranicu za postavljanje uređaja, kao i na sledeći zahtev poruku sa kordinatama zidova radi njihovog iscrtavanja na korisničkoj stranici. Kad server primi zahtev sa *post* metodom i podešavanjima uređaja on te vrednosti parsira i preko kanala prosleđuje na drugo jezgro i drugu nit kako bi ona popunila listu struktura uređaja i sa njihovim podešavanjima.

Nit koja se nalazi na pločici 1. vrši prijem podataka o podešavanjima uređaja od niti koja obrađuje HTTP zahteve i vrši smestanje tih podataka u listu uređaja. Drugi zadatak te niti je da vrši konstantno slusanje određenog porta i po prijemu UDP datagram sa zahtevom za podešavanja i samim *id*-em uređaja koji traži ta podešavanja. Na osnovu *id*-a server pretražuje listu uređaja. Po pronalasku uređaja podešavanja prosleđuje samom uređaju. Pošto nismo imali dve ili više XMOS razvojnih ploča napravljen je program u programskom jeziku C koji se pokreće na računaru i vrši simulaciju servera i uređaja. Zadatak računara koji vrši simulaciju uređaja je da na određeni port pošalje poruku sa svojim *id*. Zatim sačeka da server isparsira tu poruku, primi odgovor koji potom ispisuje na ekranu.

3.3 XMOS Uređaj

Uređaj se sastoji od *SliceKit* razvojne ploče na koju je prikačen ethernet i GPIO dodatak. Oba dodatka se nalaze na istoj pločici ali na različitim nitima. Ovde je urađen primer uređaja koji vrši kontrolu inteziteta jačine svetla na LED diodi, isti princip bi mogao da se primeni i na grejanje ili neki drugi uređaj.

Za ovo je takođe bio potreban dodatni program urađen u C++ na računaru kako bi simulirao server na ploči. Njegov zadatak je da pošalje uređaju podešavanja preko UDP protokola. Po prijemu podešavanja uređaj parsira vrednosti i šalje ih preko PWM-a na LED diode koje se nalaze na *GPIO* dodatku. U zavisnosti od vrednosti koju primi PWM vrši promenu inteziteta jačine svetla na diodi.

4. Programsko rešenje

U ovom poglavlju dat je opis programskog rešenja koje je podeljeno u 8 modula koji su realizovani u programskom jeziku C i XC. Okruženje korišćeno za njihovu realizaciju je xTIME Composer Studio (12.2.0).

- app_smarthouse_server
- app_udp_light
- module_ethernet
- module_mutual_thread_comm
- module_pwm_multibit_port
- module_smarthouse_webserver
- module_smarthouse_xtcp
- module_wifi_tiwisl

Dat je i opis programskog rešenja nekih od klasa realizovanih u Dart programskom jeziku, u okviru Dart Editor okruženja. Tu spadaju sledeće klase

- House
- Wall
- Device
- DeviceSettings

Sledi predstavljanje svakog modula ponaosob kao i njihovih najbitnijih funkcija.

4.1 Modul `app_smarthouse_server`

Ovaj modul predstavlja glavni modul u projektu koji upravlja celokupnim serverom. Sastoji se od sledećih funkcionalnih jedinica:

- **main.xc**- glavna funkcionalna jedinica modula

```
void tcp_handler(chanend c_xtcp, chanend ?c_flash, fl_SPIPorts
&?flash_ports, chanend trans);
```

Funkcija koja poziva funkciju za *WiFi* konekciju, inicijalizaciju internet servera i ukoliko stigne nešto preko *WiFi* poziva funkciju za dalju obradu tih događaja.

Parametri:

1. `c_xtcp` komunikacioni kanal za povezivanje sa xtcp serverom
2. `c_flash` komunikacioni kanal koji se koristi za pristup flash memoriji ukoliko se koriste posebna nit za to. Ukoliko se ne koristi prosleđuje se null.
3. `flash_port` ukoliko se ne koristi posebna nit za rad sa flash memorijom ovaj parameter dobavlja port za pristup memoriji ukoliko se ne koristi prosleđuje se null.

Povratna vrednost: nema

```
void udp(chanend c_xtcp, chanend c_transfer);
```

Funkcija koja prestavlja automat stanja i rukuje, obrađuje sve događaje koji dođu preko UDP-a. Pored toga prima i podatke o uređajima i smešta ih u listu struktura uređaja.

Parametri:

1. `c_xtcp` komunikacioni kanal za povezivanje sa ethernet dodatkom
2. `c_transfer` komunikacioni kanal za prenos podešavanja uređaja

Povratna vrednost: nema

- **message.c** - funkcionalna jedinica čija je uloga da parsira poruke dobijene preko HTTP-a

```
void message_parsing(chanend trans, char *msg, char* rez);
```

Funkcija koja proverava da li je reč o poruci sa kordinatama zidova ili o podešavanjima uređaja. U zavisnosti od tipa poruke poziva određenu funkciju za njeno parsiranje.

Parametri funkcije:

1. `trans` komunikacioni kanal za prenos podešavanja uređaja
2. `msg` pokazivač na poruku koju je potrebno isparsirati
3. `rez` pokazivač na poruku u koju je potrebno upisati rezultate parsiranja

Povratna vrednost: nema

```
void parsWall(char *ptr, char *comper, char *rez);
```

Funkcija koja vrši parsiranje poruke sa kordinatama zidova .

Parametri funkcije:

1. `ptr` pokazivač na poruku koju je potrebno isparsirati
2. `comper` pokazivač na niz karaktera sa kojima se poredi sadržaj poruke
3. `rez` pokazivač na niz karaktera u koji je potrebno upisati isparsirane vrednosti

Povratna vrednost: nema

```
void parsData(char* ptr, char* data, int rez[6]);
```

Funkcija koja vrši parsiranje poruke sa podešavanjima uređaja i rezultate smešta u niz.

Parametri funkcije:

1. `ptr` pokazivač na poruku koju je potrebno isparsirati
2. `data` pokazivač na niz karaktera sa kojima se poredi sadržaj poruke
3. `rez` niz celobrojnih vrednosti u koje je potrebno smestiti isparsirane rezultate

Povratne vrednosti: nema

```
int parsNowIllum(char* ptr, char* illum);
```

Funkcija koja vrši parsiranje intenziteta jačine svetlosti.

Parametri funkcije:

1. `ptr` pokazivač na poruku koju je potrebno isparsirati
2. `illum` pokazivač na niz karaktera sa kojima se poredi sadržaj poruke

Povratne vrednosti: rezultat parsiranja

- **transfer_data.xc** funkcionalna jedinica u kojoj se nalazi funkcija za slanje preko komunikacionog kanala

```
void sendData(chanend trans, Device dev[], int devNo);
```

Funkcija koja šalje podešavanja uređaja preko komunikacionog kanala.

Parametri funkcije:

1. `trans` komunikacioni kanal preko kojeg se šalju podešavanja
2. `dev[]` lista elemenata strukture uređaja
3. `devNo` ukupan broj uređaja

Povratna vrednost: nema

4.2 Modul **app_udp_light**

Modul koji predstavlja uređaj.

- **main.xc** glavna funkciona jedinica ovog modula

```
void udp_pwm(chanend c_xtcp, chanend c);
```

Funkcija koja prima podatke preko UDP protokola i prosleđuje ih na PWM.

Parametri funkcije:

1. `c_xtcp` komunikacioni kanal za primanje podataka sa *ethernet* dodatka
2. `c` komunikacioni kanal za prosleđivanje podataka ka PWM.

Povratna vrednost: nema

4.3 Modul **module_smarthouse_webserver**

Modul koji generiše binarnu datoteku na osnovu html datoteka i kao takva se snima u flash memoriju. Takođe ima i funkciju rukovanja internet stranicama kao i obradom svih HTTP zahteva.

- **web_server.c** funkciona jedinica sa funkcijama koje inicijalizuju internet stranicu, obrađuju HTTP zahteve i sve događaje koje prime preko *WiFi* dodatka

```
void web_server_handle_event(chanend c_xtcp, chanend
c_flash, fl_SPIPorts *flash_ports, xtcp_connection_t *conn, chanendtrans);
```

Funkcija koja predstavlja automat stanja i rukuje HTTP zahtevima.

Parametri funkcije:

1. `c_xtcp` komunikacioni kanal za povezivanje sa xtcp serverom
2. `c_flash` komunikacioni kanal koji se koristi za pristup flash memoriji ukoliko se koriste posebna nit za to, ukoliko se ne koristi prosleđuje se null.
3. `flash_port` ukoliko se ne koristi posebna nit za rad sa flash memorijom ovaj parameter dobavlja port za pristup memoriji ukoliko se ne koristi prosleđuje se null.
4. `conn` Pokazivač na struktura TCP povezivanja sa sadržajem novog događaja.
5. `trans` komunikacioni kanal za prenos podešavanja uređaja

Povratnu vrednost: nema

```
static void parse_http_request(chanend c_xtcp,
xtcp_connection_t *conn, connection_state_t *st, char *buf, int len, chanend
trans);
```

Funkcija koja parsira HTTP zahteve.

Parametri funkcije:

1. `c_xtcp` komunikacioni kanal za povezivanje sa xtcp serverom
2. `conn` Pokazivač na struktura TCP povezivanja sa sadržajem novog događaja.
3. `st` Pokazivač na strukturu sa informacijama o stanju konekcije
4. `len` Pokazivač na dužinu primljenih podataka u bajtovima
5. `buf` Pokazivač na niz karaktera sa primljenim podacima
6. `trans` komunikacioni kanal za prenos podešavanja uređaja

Povratnu vrednost: nema

4.4 Ostali korišćeni moduli

- **Modul `module_ethernet`** - Obezbeđuje ethernet stack za IP/UDP/TCP protokole.

- **Modul `module_mutual_thread_comm`** - Služi za komunikaciju između niti preko kanala.
- **Modul `module_pwm_multibit_port`** - Služi za realizaciju i korišćenje funkcija PWM-a.
- **Modul `module_smarthouse_xtcp`** - Povezuje se sa ethernet dodatkom i vrši obradu TCP/UDP protokola.
- **Modul `module_wifi_tiwisl`** - Vršiti povezivanje preko *WiFi* dodatka na lokalnu mrežu.

4.5 Klasa Wall

Ova klasa sadrži početnu i krajnju tačku zida. Njene *set* i *get* metode.

```
drawLine(int x, int y, String color);
```

Funkcija koja iscrtava zid.

Parametri funkcije:

1. *x* kordinate tačke po x-osi do koje treba iscrtati liniju
2. *y* kordinate tačke po y-osi do koje treba iscrtati liniju
3. *color string* koji sadrži *RGB* informaciju

Poivratnu vrednost: nema

5. Testiranje i rezultati

Testiranje je vršeno na sledećim mobilnim uređajima, LG L9 i LG L7. Bitno je napomenuti da sami sajtovi za podešavanja uređaja, crtanje plana, kao i dodavanje uređaja nisu vezana kako za operativni sistem. Ovo rešenje se oslanja na HTML5 tehnologiju, tako da je konfigurisanje servera moguće iz bilo kod pretraživača sa HTML5 podrškom.

Testiranje internet stranica pored telefona vršeno je i na računaru u sledećim pretraživačima *Chrome*, *FireFox* kao i u *Dartiumu* koji se prevodi u kod za *Dart* virtualnu mašinu. Na osnovu testova na računaru ustanovljeno je da se prilikom iscrtavanja zidova pri pomeraju kursora miša momentalno iscrtava i zid kao što je bilo i predviđeno. Tokom testova na mobilnim uređajima samo učitavanje stranica je trajalo 1 minut i 37 sekundi. Ovo je posledica sporog iščitavanja stranice iz *flash* memorije XMOS servera. Da bi se ovaj postupak ubrzao potrebno je XMOS serveru dodati *SDRAM* dodatak, čija je brzina čitanja znatno veća u odnosu na brzinu čitanja iz *flash* memorije. Zidovi se iscrtavaju na dodir, a ne na pomeranje prsta zbog samog projektovanja pretraživača na telefonima i njihovog fokusiranja na kompletanu stranicu, a ne samo na jedan element stranice. U nastavku testiranja su praćeni HTTP zahtevi kao i samo ponašanje servera po prijemu tih zahteva i njihovoj obradi i parsiranju poruka. Ustanovljeno je da se sve ponaša kako je i predviđeno tokom projektovanja. Sledeća stvar koja je bila testirana jeste komunikacija sa uređajem (računarom) preko UDP protokola. Tokom rada servera nekoliko puta je uređaj zatražio podatke pre nego što su podaci i bili upisani, server je odgovorio sa 0 što je bilo i predviđeno. Tek nakon što je server primio podešavanja uređaja pri sledećem zahtevu uređaja, server je odgovorio sa vrednosti koju je i primio. Ceo postupak je ponovljen nekoliko puta sa različitim *id*-evima uređaja i sve se ponašalo kako je i bilo predviđeno.

Uređaj na XMOS ploči kao i menjanje inteziteta jačine svetlosti je testirano konstantim slanjem poruka od strane servera (računara) sa različitim vrednostima unetim od strane korisnika. U zavisnosti od poslanih vrednosti menjao se intenzitet jačine svetlosti onako kako je bilo predviđeno tokom projektovanja.

6. Zaključak

U ovom radu je realizovano jedana od mnogih mogućnosti upotrebe XMOS platform u projektovanju pametne kuće. Konkretno upravljanje kućnim aparatom uz pomoć bilo kog uređaja koji poseduje *web pretrađivač*.

Rad se može poboljšati i unaprediti na sledeće načine. Memorisanjem podešavanja svakog uređaja u *flash* memoriju da bi i posle gašenja servera podaci ostali sačuvani. Sledeće unapređenje moglo bi biti dodatak stranice sa korisničkim imenom i lozinkom radi identifikacije korisnika. Pored toga moglo bi se realizovati podešavanje vremena i jačine intenziteta svetlosti uređajau za jedan dan, kao i za sedam dana pa i više. Dodavanjem SDRAM dodatka na XMOS server bi ubrzalo čitanje stranica iz memorije i samim time učitavanje stranice na uređaju kojim želimo da postavimo podešavanja.

Mogućnost iscrtavanja plana kuće po spratovima bi moglo da bude jedno unapređenje.

7. Literatura

- [1] “*HTTP The Definitive Guide*” David Gourley, Brian Totty with Marjorie Sayer, Sailu Reddy, and Anshu Aggarwal; September 2002: First Edition. O’Reilly Media. p. ISBN-10: 1-56592-509-2
- [2] “*HTML i Web dizajn kroz praktične primere*”, Kris Jamsa, Konrad King, Andy Anderson; 2003, Mikro knjiga, ISBN: 86-7555-204-1
- [3] “*HTML5 Canvas*”, Steve Fulton and Jeff Fulton, May 2011: First Edition. O’Reilly Media, Inc.p. ISBN: 978-1-449-39390-8
- [4] XMOS Multicore Microcontrollers, <http://www.xmos.com/>, jul 2013
- [5] ”*Računarske mreže*“, Prof. dr Mladen Veinović, Dr Aleksandar Jevremović, 2011, Univerzitet Singidunum, Beograd, ISBN: 978-86-7912-368-8
- [6] “*What is Dart?*” Walrath, Kathy; Ladd, Seth (March 7, 2012). (1st ed.). O’Reilly Media. p. 20. ISBN 978-14493-32327
- [7] “*Pulse Width Modulation, in Flexible Power Transmission: The HVDC Options*”, Arrillaga, J., Liu, Y. H. and Watson, N. R.; Septembar 2007. John Wiley & Sons, Ltd, Chichester, UK. ISBN 978-0-470-05688-2