

**SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 3354

**NORME I PROTOKOLI ZA
AUTOMATIZACIJU KUĆANSTVA**

Petar Matković

Zagreb, lipanj 2013.

Sadržaj:

1. Uvod	3
2. <i>Smart Home</i> koncept budućnosti	5
2.1 Inteligentni sustav upravljanja.....	5
2.2 Tehnologija budućnosti	6
3. Norme i protokoli u kućnoj automatizaciji	9
3.1 ISO/OSI referentni model.....	9
3.2 Topologija.....	12
3.3 Komunikacijski mediji u kućnoj automatizaciji	14
3.4 <i>Smart Home</i> protokoli	15
3.4.1 Konnex.....	16
3.4.2 LonWorks	20
3.4.3 BACNet	22
3.4.4 DALI.....	23
3.4.5 X10	23
3.4.6 ZigBee.....	24
4. Izvedba <i>Smart Home</i> sustava	30
4.1 ATmega168 mikrokontroler	32
4.2 Središnji upravljački čvor	34
4.2.1 Matrična tipkovnica	35
4.2.2 LCD prikaznik	36
4.3 Čvor za regulaciju temperature.....	38
4.3.1 Digitalni termometar.....	39
4.3.2 Upravljanje grijačem.....	39
4.4 Čvor za pokretanje muzike	40
4.4.1 Muzički modul WTV020.....	41
4.4.2 Formatiranje i prilagodba pjesama.....	42
4.5 Čvor za uključivanje tereta	44
4.5.1 RSL sklopke.....	45
4.5.2 Daljinski modul i upravljanje trošilima	46
4.6 Podatkovni okvir	47
4.7 Tehnički podaci	48
5. Zaključak	49

Literatura	50
Sažetak.....	52
Dodatak	53
Električna shema.....	53
Sastavnica	57
Programski kod.....	58

1. Uvod

Moderni način života zahtijeva na svim tehničkim razinama optimizaciju, pojednostavljenje, udobnost, brzinu, minimizaciju i uštedu energije. Ovi procesi vode k ostvarivanju kako ekonomskog tako i vremenskog profita. Dvadeseto stoljeće donosi nagli tehnološki napredak, posebice u elektrotehnici, energetici, strojarstvu i informacijskim znanostima. Suvremene tokovine društva teže sveobuhvatnoj integraciji svih tehničkih disciplina u svrhu ostvarivanja gore zadanih ciljeva. Ti postupci su u velikoj mjeri prisutni u industrijskim postrojenjima, elektroenergetskim sustavima i prometu. Unazad dvadesetak godina postupci tehnološke integracije polagano nalaze primjenu i u kućanstvima i zgradarstvu kroz procese kućne i zgradarske automatizacije (engl. *Smart Home – Smart Building*).

Automatsko prilagođenje rasvjete, razine roleta te otvorenosti prozora ovisno o vremenskim uvjetima i prisutnosti ukućana u svrhu uštede energije ili inscenirana reprodukcija muzike i multimedijskog sadržaja uz prilagođenu kućnu temperaturu u svrhu udobnosti, laganim i sigurnim tempom postaje svakodnevnica. Alarmni sustav, osim što detektira neželjen prisup, mjeri i vlagu i dim te hitno dojavljuje vlasniku preko mobitela informaciju o rizičnim pojavama: provali, poplavi ili požaru. Omogućena je i komunikacija u drugom smjeru. Vlasnik preko SMS-a ili elektroničke pošte zadaje radnju u kućanstvu i prije samog dolaska u kuću: zagrijavanje pećnice, namještanje sobne temperature ili uključivanje perilice. Kućna automatizacija vrlo je zanimljiv i koristan proces, no na putu k njezinoj punoj implementaciji stoje brojne ekonomske i tehnološke poteškoće. Mnoge norme i komunikacijski protokoli, tehnološke izvedbe, nepouzdanost i cijena prepreka su širokom korištenju kućne automatizacije u punom smislu te riječi.

Zadnjih godina veliki su naponi uloženi u proces standardizacije protokola za automatizaciju u kućanstvu i zgradarstvu. Sama tehnologija neprestano napreduje i u relativno kratkom vremenu dobra tehnološka rješenja postaju cjenovno prihvatljiva. Stoga postoji veliki potencijal da takva automatizacija kroz dogledno vrijeme postane dio životnog standarda.

Kroz ovaj rad objasnit ćemo koncept pametne kuće i proučiti poznate norme i komunikacijske protokole u kućnoj i zgradarskoj automatizaciji. Upoznat ćemo sabirničke, bežične i ostale sustve veza te načine povezivanja uređaja u takvim sustavima. Završno ćemo realizirati primjer kućne automatizacije sa jednim nadzornim uređajem i tri vlastito napravljena čvora od kojih jedan integrira dva komercijalno dobavljiva čvora.

2. *Smart Home* koncept budućnosti

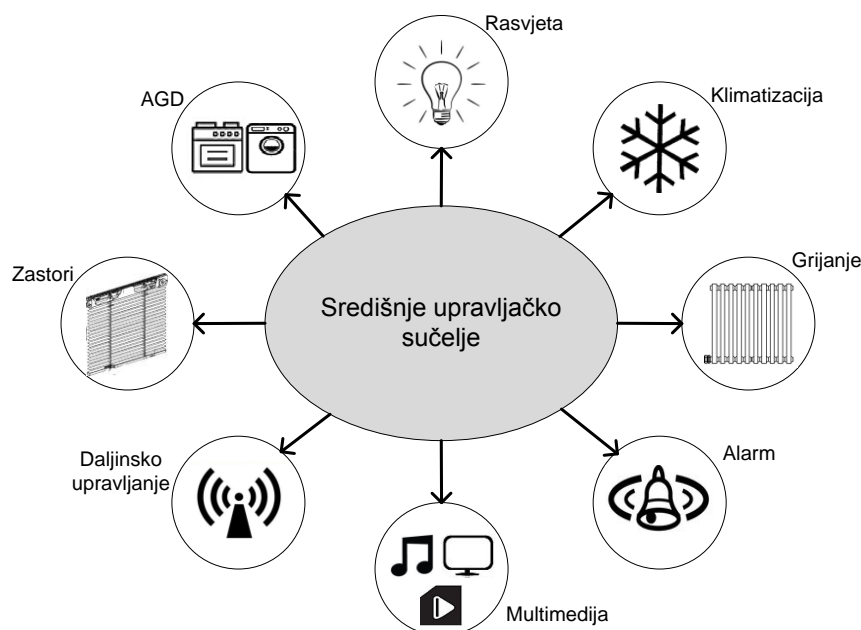
Pametna kuća (engl. *Smart Home*) naziv je za široki pojam automatizacije i integracije funkcija u kućanstvima kojim se osvaruje inteligentni funkcionalni sustav. Naziv se pojavljuje sedamdesetih godina dvadesetoga stoljeća te se tada odnosi na one stambene objekte koji ostvaruju energetska učinkovitost. U osamdesetim godinama pojavljuju se primjeri upravljanja raznim sustavima pomoću osobnog računala. U našem vremenu pojam pametne kuće podrazumijeva energetska učinkovitost i računalno upravljanje obuhvaćajući napredne procese automatizacije.

Općenito se ime “pametna kuća“ odnosi na različite međusobno povezane funkcionalnosti u uskom području u kojemu korisnik obitava. No procesi integracije u jedan pametan sustav provode se na širim stambenim objektima tj. unutar cijele zgrade pa se često takav sustav naziva i pametnom zgradom (eng. *Smart Building*). Sustavi automatizacije odavno su prisutni u industrijskim i energetska postrojenjima kroz procese mjerenja, izvršavanja i kontrole te se iste tehnologije nastoje upotrijebiti u okviru pametne kuće.

2.1 Inteligentni sustav upravljanja

Koncept pametne kuće temelji se na inteligentnom sustavu upravljanja koji objedinjuje sve funkcijske uređaje te automatizacijske i komunikacijske postupke. On se općenito sastoji od središnjeg upravljačkog uređaja (engl. *main interface*), nadzornih funkcionalnih uređaja (engl. *functional executor*) i završnih uređaja (engl. *end devices*). Blok shema *Smart Home* sustava sa prikazanim integriranim funkcionalnostima dana je na slici 1. Središnji uređaj integrira veliki broj kućanskih funkcija istovremeno dopuštajući nadzornim funkcionalnim uređajima samostalno upravljanje izvršnim članovima. Treba naglasiti da je ova struktura samo načelna te da postoje razne decentralizirane strukture sa višim stupnjem slobode pojedinih uređaja. Da bi takav sustav bio cjelovit potrebno mu je pridružiti određeni način povezivanja uređaja: vrstu medija, topologiju veze, komunikacijski protokol i hijerarhiju uređaja. Postoji širok spektar normi koje opisuju

načine povezivanja uređaja u sustavima pametnih kuća i detaljniji prikaz istih ćemo dati u narednim poglavljima.



Slika 1: *Smart Home sustav upravljanja*

Navedimo jedan primjer iz svakodnevnog života kojim ćemo objasniti strukturu inteligentnog sustava upravljanja. U standardnoj regulaciji grijanja glavnu ulogu ima termostat. Korisnik ručno unosi željenu temperaturu preko korisničkog sučelja nakon čega se spregom senzora (termometar) i aktuatora (ventila u komori izgaranja) održava željena temperatura. Kada bi jedan takav sustav integrirali u sustav pametne kuće preko dotičnog sustava veze dobili bismo hijerarhijsku strukturu na čelu sa rukovodećim upravljačkim uređajem koji preko funkcijskog nadzornog sklopa (termostat) i završnih uređaja (senzor i aktuator) obavlja regulaciju temperature.

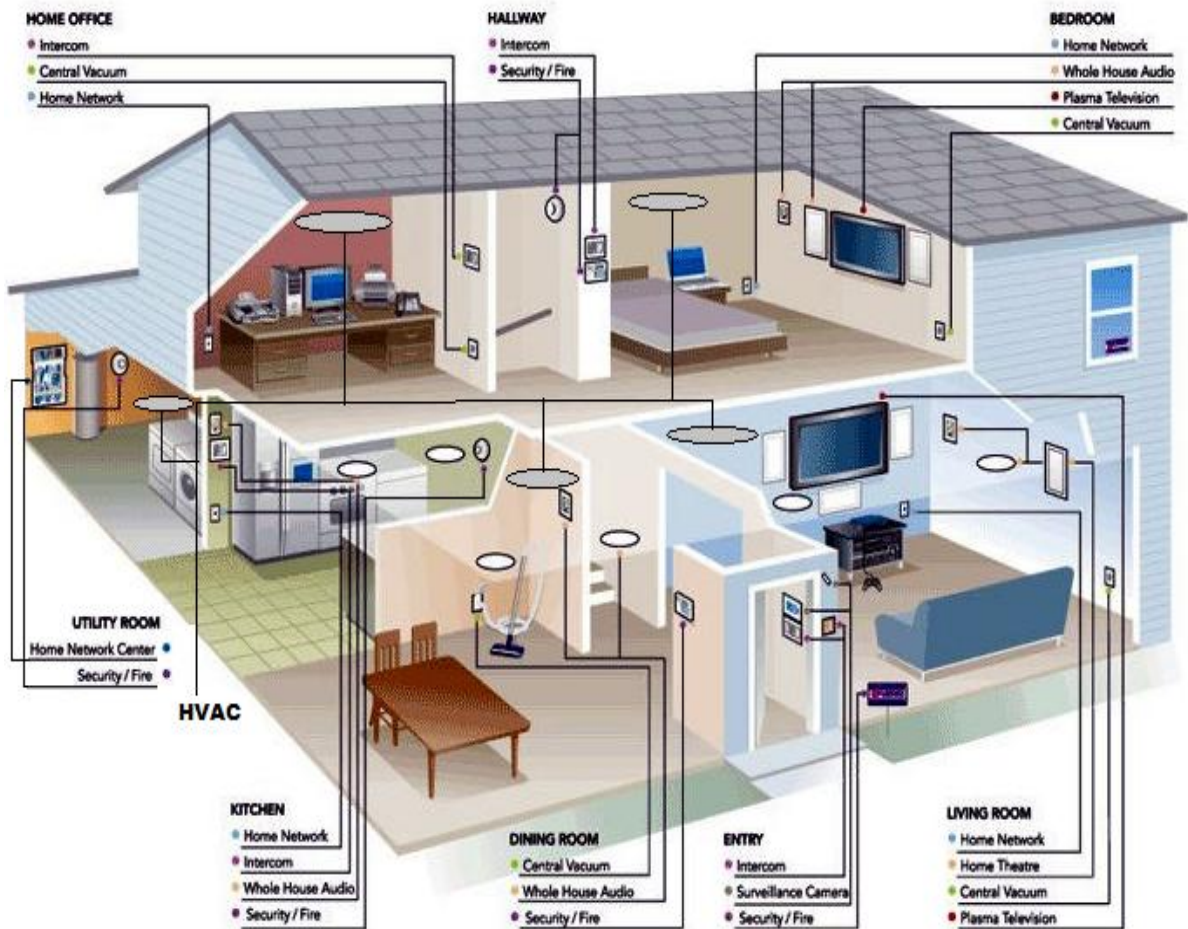
2.2 Tehnologija budućnosti

S tehnološkog aspekta, broj funkcija i zadaća koje pametna kuća povezuje i izvršava poprilično je velik. Ograničenja nastupaju prvenstveno zbog prevelike cijene tehnološke izvedbe. Kompromisom cijene i funkcionalnosti korisnik odabire određenu razinu složenosti pametne kuće. Moderne pametne kuće najčešće objedinjuju sljedeće funkcionalne zadaće:

- HVAC (engl. *Heating, Ventilation and Air Conditioning*) – grijanje, hlađenje i ventilacija tj. energetska učinkovita regulacija temperature
- Rasvjeta – sukus električne i dnevne rasvjete uz detekciju pokreta
- Alarmni sustavi – detekcija i dojava upozorenja
- Automatsko zamračivanje (engl. *Shading*) – automatsko upravljanje roletama i zastorima u svrhu regulacije temperature (pasivno grijanje) i osvjetljenja te iluzije prisutnosti
- Svijet multimedije – koncept već prisutan u osobnom računalu i zasebnim električkim uređajima sada objedinjen, prostorno raspodijeljen, internetski ažuriran i reproduciran po cijeloj kući
- RF upravljanje – uključivanje raznih tereta (AGD kućanski aparati, žarulje,...) na elektroenergetsku mrežu
- Zvučna i vizualna komunikacija unutar i izvan kuće – interfon i kamere
- “*Online Home*“ i “*Mobile Home*“ – zadavanje zadataka preko interneta ili mobilnom mrežom pomoću mobilnog uređaja
- Odabir scenarija – pametna kuća obavlja unaprijed pripremljene i povezane radnje u svrhu jednostavnosti, operabilnosti i opuštanja

U sustavima kućne automatizacije često nalazimo i mnoge druge zadaće i mogućnosti koje odgovaraju korisnikovim željama i mogućnostima: regulacija kvalitete vode bazena, sustav natapanja vrta preko interneta povezan sa vremenskom prognozom, detekcija toksičnih plinova u garaži, automatsko hranjenje kućnih ljubimaca, itd.

Primjer implementacije i alokacije *Smart Home* koncepta u kućanstvu prikazan je slikom 2. Uočavamo da ugradnja koncepta pametne kuće započinje projektiranjem i gradnjom stambenog objekta. Korisnički zahtjevi vezani za kućnu automatizaciju se već tada uzimaju u obzir i određuju oblik “stambene infrastrukture“. U tom slučaju kućna automatizacija može u potpunosti zaživjeti sukladno estetici i ergonomičnosti. U slučaju starijih objekata i onih kod kojih se prilikom same izgradnje nije vodila briga o mogućoj svobuhvatnoj automatizaciji nastaju brojni problemi. Postupak automatizacije postaje višestruko skuplji i često se kao nusprodukt javlja degradacija estetike te smanjena sigurnost. Također se pojavljuje potpuna nemogućnost implementacije određenih funkcija.



Slika 2: Implementacija pametne kuće

Zbog količine zadataka koje mora izvršiti i načina implementacije zaključujemo da pametna kuća kao inteligentni sustav nije jednostavan koncept i ne može se svrstati u jednu tehničku disciplinu. Ostvarivanje takvog sustava iziskuje znanja iz mnogih tehničkih područja. Disciplina automatskog upravljanja odavno je poznata u inženjerstvu i njezina sinergija sa drugim disciplinama rezultira naprednim tehnološkim rješenjima koji su uvelike promijenili sadašnjost i određuju izgled budućnosti. Pametna kuća će još dugo biti, od strane inženjera, objekt usavršavanja i postizanja boljih preformansi te je zasigurno tehnologija budućnosti.

3. Norme i protokoli u kućnoj automatizaciji

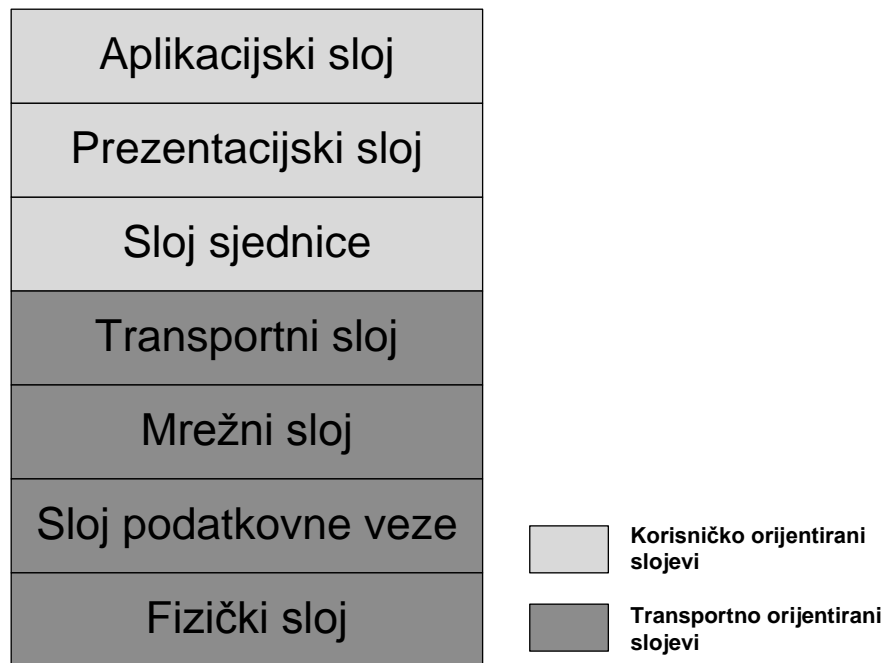
Povezivanje automatizacijskih uređaja se naziva umrežavanje. Umrežavanje omogućuje obavljanje složene funkcionalnosti sustava uz zadane vremenske i prostorne specifikacije. Kako bi se postigao inteligentni sustav upravljanja potreban je pouzdan način povezivanja i komunikacije. Takve procese u kućnoj automatizaciji nazivamo automatizacijskim protokolima koji definiraju određenu automatizacijsku normu tj. standard.

Protokoli u komunikaciji u biti su vrlo složeni i razgranati pa uslijed malih varijacija nastaju veliki problemi s komunikacijom. Stoga se javlja potreba za određenom podjelom i klasifikacijom određenih protokolarnih procesa. Kao glavno rješenje pojavljuje se vertikalna podjela mreže na slojeve (engl. *layer*), a kao glavna referentna norma ustaljuje se model povezivanja otvorenih sustava OSI (engl. *Open System Interconnection*). Za automatizacijske protokole bitna je i topologija mreže tj. uređaja kao i vrsta medija.

3.1 ISO/OSI referentni model

Referentni modeli definiraju koncepte i utvrđuju pravila povezivanja sustava u mrežu. Oni omogućuju standardizaciju komunikacijskih procesa te kompatibilnost uređaja tvoreći tako otvorena rješenja uz pomoć kojih se tvore komunikacijski sustavi neovisno o proizvođaču opreme i samom uređaju. Kao što je navedeno, od brojnih referentnih modela kao osnovni se uzima OSI model sa vertikalnom podjelom na slojeve. Taj model je kreiran 1979. godine od strane Međunarodne organizacije za normizaciju – ISO (engl. *International Organisation for Standardisation*) te nakon određenih revizija poprima današnji oblik. Model se sastoji od sedam slojeva podijeljenih u dvije grupe kao što je prikazano slikom 3. Fizički, podatkovni, mrežni i transportni sloj spadaju u transportno orijentirane slojeve i obavljaju zadatak transporta korisničkih i upravljačkih informacija. Sjednički, prezentacijski i aplikacijski slojevi usklađuju aplikacijske procese te ih ubrajamo u korisničko orijentirane slojeve. Potrebno je naglasiti da je ovakav model, uz sva njegova pojednostavljenja, poprilično složen i u svojoj strukturi nedovoljno precizno definira granice pojedinih slojeva i zadaća. Česta je pojava da pojedini protokoli objedinjuju određene slojeve, no često dolazi i do dodatne formulacije. Stoga je katkad

neke protokole teško strukturirati i implementirati po referentnom modelu. Druga poteškoća jest što ovaj protokol biva kasno normiran. U to doba u punoj snazi korišten je TCP/IP model (engl. *Transmission Control and Internet Protocol*) koji nastaje iz prijenosno upravljivog i internetskog protokola [1][6].



Slika 3: ISO/OSI slojevita arhitektura

Usprkos određenim problemima, ISO/OSI model postaje općenit model za ostvarivanje protokola i predvodnik je standardizacije komunikacijskih sustava. Jasno određuje fizičku vezu, protokolarne usluge i korisnička sučelja pa je pogodan kao komunikacijska referenca u sustavima pametnih kuća. Opišimo stoga поближе zadaće pojedinih slojeva kako bismo mogli dati bolji uvid u protokole koji se koriste u kućnoj automatizaciji.

Fizički sloj (engl. *Physical Layer*)

Sloj koji opisuje fizička tj. električna svojstva informacije i podataka koji se prenose određenim medijem. On opisuje pristup komunikacijskom kanalu (modulacije, sažimanja, naponske razine, impedancije, vrstu konektora i). Sam medij nije dio fizičkog sloja, ali se karakteristike medija uzimaju u obzir u postupcima pristupa mediju.

Sloj podatkovne veze (engl. *Data Link Layer*)

Opisuje direktnu vezu između dva mrežna čvora te osigurava transmisiju podataka. To uključuje generiranje i prijenos podatkovnih okvira (engl. *frame*), ali i procesiranje potvrđujućih okvira u svrhu pouzdanog prijema podataka. Također, ovaj sloj provjerom cikličke redundancije (CRC - engl. *cyclic redundancy check*) upravlja eventualnim pogreškama.

Mrežni sloj (engl. *Network Layer*)

Služi kao odvojnica aplikacijskih i strogo prijenosnih procesa. Provjerava mrežni put između mrežnih čvorova te vrši zadaću usmjeravanja jedinica podataka tzv. paketa. Povezuje funkcionalne uređaje i upravlja funkcionalnim tokom podataka te dodatno upravlja pogreškama.

Transportni sloj (engl. *Transport Layer*)

Definira izvor i odredište informacije te osigurava pouzdan prijenos podataka između dva takva uređaja (tzv. prijenos s kraja na kraj). Odredište ne mora biti jedan uređaj, već skupina uređaja ili pak cjelokupna mreža. Transportni sloj određuje tok podataka obzirom na namjenjene uređaje i vrši dodatno upravljanje pogreškama. Ukoliko uoči da su određeni podaci izgubljeni pokreće retransmisiju.

Sloj sjednice (engl. *Session Layer*)

Dovodi određeni broj mrežnih čvorova u "sjednicu" i sinkronizira komunikaciju. Ovaj sloj je zadužen za pouzdan nastavak veze uslijed pada komunikacije od istog mjesta gdje nastaje prekid. Osigurava izvođenje aplikacije u višim slojevima te se ubraja u korisničko orijentirane slojeve.

Prezentacijski sloj (engl. *Presentation Layer*)

Sloj dobavljene podatke interpretira te strukturira kodirane podatke tj. vrši semantičke i sintaktičke zadaće. Time se aplikacijski procesi dovode u neovisnost u načinu prikaza podataka.

Aplikacijski sloj (engl. *Application Layer*)

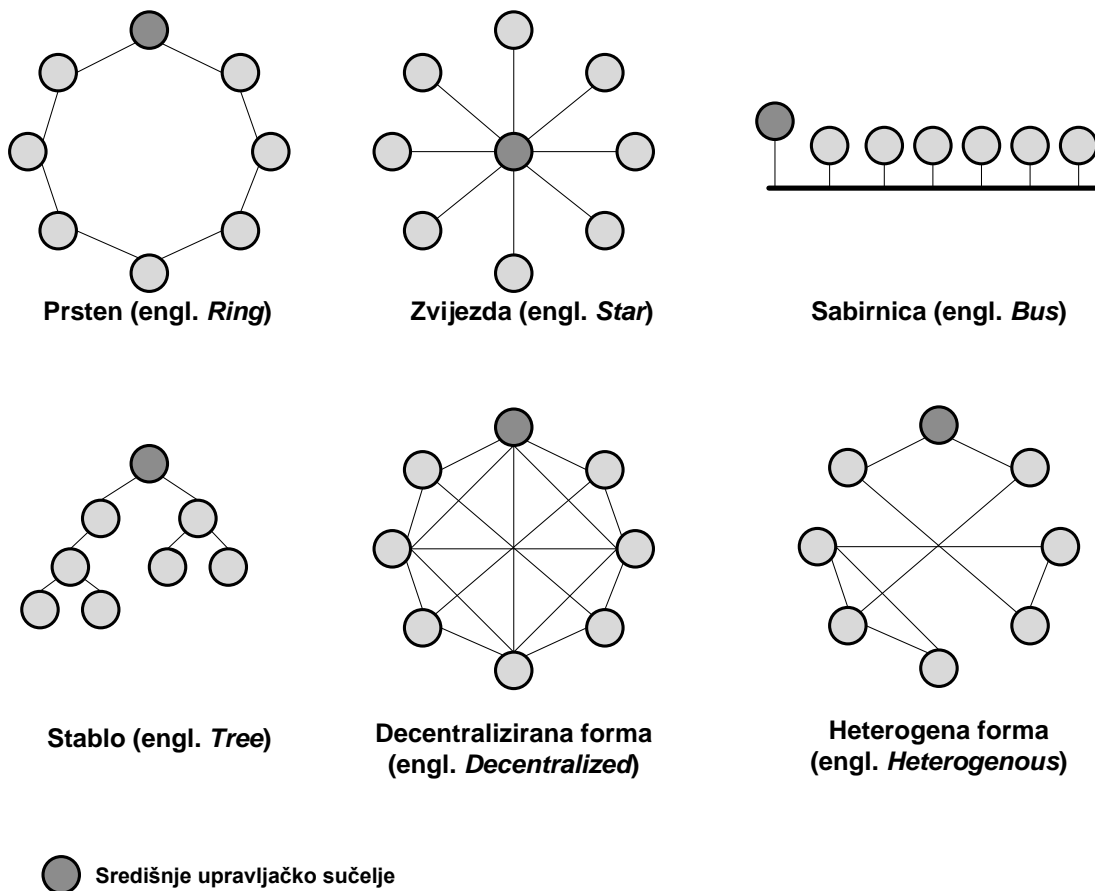
Ovaj sloj zapravo je korisničko sučelje i omogućava korisnicima pristup komunikaciji. Pruža korisnicima usluge, a s druge strane stvara uvjete za realizaciju te

usluge. Aplikacije koje se izvode zahvaljujući donjim slojevima zapravo su svrha cijelog komunikacijskog lanca.

Često se posljednja tri OSI modelom definirana sloja zbog malih zasebnih odgovornosti objedinjuju u jedinstveni aplikacijski sloj koji vrši zadaće sva tri sloja. To je slučaj i kod protokola kućne automatizacije.

3.2 Topologija

Mreža je opisana topologijom čvorova tj. uređaja u slučaju kućne automatizacije. Topologija daje uvid u cjelokupni sustav i određuje implementaciju protokola. Zapravo topologija određuje način povezivanja fizičkih slojeva uređaja u komunikacijskom sustavu. Prema [1] u kućnoj automatizaciji prevladava 6 osnovnih formi: prsten, zvijezda, sabirnica, stablo, decentralizirana i heterogena forma. Na slici 4 prikazan je oblik navedenih topologija.



Slika 4: Topologije u kućnoj automatizaciji

Prstenasta struktura svim čvorovima pridodaje istu važnost jer svi sudjeluju u prijenosu podataka. Izvor informacije generira poruku adresiranu određenom uređaju koja putuje cijelom mrežom. Informacija dolazi do odredišta te se vraća izvoru, ali u ovom slučaju sa potvrdom prijema. Ova topologija traži od uređaja dva registra za pohranu poruke (ulazni i izlazni) što pridonosi složenosti ove strukture. Također, prilikom pucanja veze između dva čvora prekida se cjelokupna veza što je u jednu ruku poželjno jer korisnik odmah uviđa kvar, ali je u drugu ruku vrlo nepoželjno jer se prekida rad eventualno bitnih funkcija u domu.

Zvezdasta struktura zapravo je centralizirana struktura koja glavnom upravljačkom uređaju daje potpuni nadzor i kontrolu veze. Načelno su ovakvi sustavi sporiji u odnosu na drugačije oblike te je potrebno da upravljački uređaj ima velike procesorske performanse kako bi osigurao željenu brzinu i pouzdanost. Rijetka je pojava u sustavima pametnih kuća da središnji upravljački uređaj upravlja baš svim čvorovima u sustavu.

Sabirnička struktura, ovisno o protokolu, omogućava spajanje popriličnog broja uređaja koji u određenim trenucima upravljaju sabirnicom po principu vlasnik – sluga (engl. *master – slave*). Broj uređaja koje je moguće spojiti ovisi o formatu podataka koji se šalje sabirnicom i načelno je manji od broja koji se može spojiti u prstenastu strukturu. Broj linija u sabirnici ovisi o željenoj brzini rada i kreće se od dviju linija (TW – engl. *twisted pair*) kojima se prenose podatkovni i upravljački okviri sa ograničenom brzinom do velikog broja linija (dio adresnih, dio podatkovnih i dio sinkronizacijskih) kojim se postižu znatne brzine. Također je u uređaje potrebno implementirati metodu pristupa sabirnici tzv. arbitraciju. Najčešće se koristi tzv. CSMA/CA arbitracija (engl. *Carrier sense multiple access with collision avoidance*).

Struktura stabla nastaje iz potrebe funkcionalnog grupiranja uređaja i samim time pojednostavljuje izvođenje željenih funkcija. To je vrlo često u sustavima kućne automatizacije gdje određeni uređaj odgovara svom nadzornom uređaju sve do glavnog upravljačkog uređaja. Ovakva hijerarhijska struktura je vrlo djelotvorna, ali postoje situacije gdje je ona nepoželjna. Zamislimo, primjerice, slučaj požara gdje detektor javlja vlasniku upozorenje, a istovremeno ne uključi prskalice.

Decentralizirana topologija omogućava svim čvorovima određenu vrstu autonomije i u samim sustavima ostvaruje velike brzine. Instalacija ovakve forme infrastrukturno je zahtjevana. U pametnoj kući je teško zamisliti da svaki par uređaja bude zasebno žično

međusobno povezan. U slučaju radio komunikacije dolazi do znatne nepouzdanosti zbog velikog broja korisnika istog frekvencijskog područja. Stoga se nastoji u određenoj mjeri heterogenizirati uređaje.

Heterogena struktura u određenoj mjeri objedinjuje iste funkcijske uređaje, no za razliku od stabla osigurava određenu autonomiju i postiže veću brzinu rada. Ovakva struktura sadrži elemente, zvijezde, prstena i stabla. Proces decentralizacije ograničen je prostornim i tehnološkim specifikacijama i kao rezultat nastaje heterogeni sustav.

3.3 Komunikacijski mediji u kućnoj automatizaciji

Protokole je vrlo bitno gledati iz aspekta medija kojim se odvija komunikacija. U kućnoj automatizaciji razlikujemo tri načina spajanja uređaja:

1. Žično spajanje
2. Bežično spajanje
3. Elektroenergetska mreža.

Žično spajanje se provodi sabirnicama ili češće upletenim paricama. Omogućava spajanje velikog broja uređaja u sustav te pouzdanu komunikaciju. Ograničenja nastaju pri velikim udaljenostima i pri velikim brzinama gdje smetnje gradske mreže smanjuju pouzdanost.

Bežično spajanje se odnosi na radio frekvencijsku vezu u onim frekvencijskim područjima i s onim snagama koji su zakonski dopuštene:

- 40 MHz
- 433 MHz
- 870 MHz
- 2440 MHz (tzv. ISM područje - engl. *Industrial, Sceintific and Medical*)

Općenito zbog ograničenih frekvencijskih pojaseva podatkovna propusnost i brzina rada ovakvih sustava jest mala. Obzirom da isti pojas frekvencija koristi veći broj uređaja veća je vjerojatnost pogreške. Također nastaju velika ograničenja sa povećanjem udaljenosti.

Nekonvencionalni način komuniciranja ostvaruje se preko gradske mreže (PLC – engl. *Power Line Communication*). Koristi se načelo superpozicije signala te se uz pomoć

filtra odvajaju signal određene frekvencije od napona mreže (50 Hz) i dovodi se na fizički sloj uređaja. Takav signal je moduliran određenim postupkom i demodulacijom se dobiva primljena informacija. Isti je princip u suprotnom smjeru. U kućnoj automatizaciji se za prijenos signala koristi frekvencija od 20 kHz do 200 kHz uz diskretnu frekvencijsku (FSK – engl. *Frequency Shift Keying*) ili položajno impulsnu modulaciju (PPM – engl. *Pulse Position Modulation*). Ovakav koncept ne traži posebnu veznu infrastrukturu već koristi postojeće resurse. Mala propusnost i kapacitet glavne su mane ovakvog komunikacijskog kanala.

3.4 **Smart Home** protokoli

Pošto smo naveli glavne značajke protokola i komunikacije možemo navesti i opisati one protokole koji se koriste u sustavima kućne automatizacije. Više je osnova po kojima možemo klasificirati protokole: obzirom na topologiju, način povezivanja ili otvorenost sustava.

Zatvoreni standardi vlasništvo su privatnih kompanija koje izrađuju tom standardu kompatibilne uređaje. Postupci povezivanja javnosti su nedostupni u svrhu zaštite od konkurencije [7]. To su sustavi sljedećih kompanija:

- **Control4** – Salt Lake City, SAD [17]
- **Crestron** – New Jersey, SAD [18]
- **AMX** – Richardson, SAD [19]

U našem razmatranju su zanimljivi otvoreni sustavi (engl. *open source systems*) sa poznatom strukturom i komunikacijskim protokolom. Oni omogućuju interoperabilnost i laku zamjenjivost uređaja bez degradacije sustava. U povijesti kućne automatizacije kristalizirala su se dva glavna standarda: europski **EIB** (engl. *European Installation Bus*) i američki **LonWorks**. Devedesetih godina 20. stoljeća, uz EIB, u Europi se još koriste **European Home Systems** i **BatiBUS** standardi. Oni se zajedno sa standardom EIB stapaju u jedinstveni europski standard **Konnex KNX**. U kućanskoj i zgradarskoj automatizaciji pojavljuju se dodatni sabirnički protokoli **BACNet** i **DALI**. Jedan od najstarijih protokola u kućnoj automatizaciji koji primarno koristi elektroenergetsku mrežu kao medij jest **X10** standard. U bežične standarde kućne automatizacije tzv. WHAN-ove (engl. *Wireless Home Automation Network*) ubrajamo **ZigBee**, **Z-Wave**, **Insteon** i **Wavenis**. Specifikacije za

posljednja tri standarda su samo djelomično javno dostupne pa se ne može reći da su ti standardi potpuno otvoreni. Potrebno je naglasiti da oni protokoli koji su strukturno sabirnički mogu koristiti kao medij i gradsku mrežu i radio medij. No, ipak ćemo podrazumijevati radio vezu načinom spajanja bežičnih standarda. Većina standarda u kućnoj i zgradarskoj automatizaciji se bazira na ISO/OSI modelu sa integriranim korisničko usmjerenim slojevima tj. na ISO/OSI modelu sa 5 formalnih slojeva.

3.4.1 Konnex

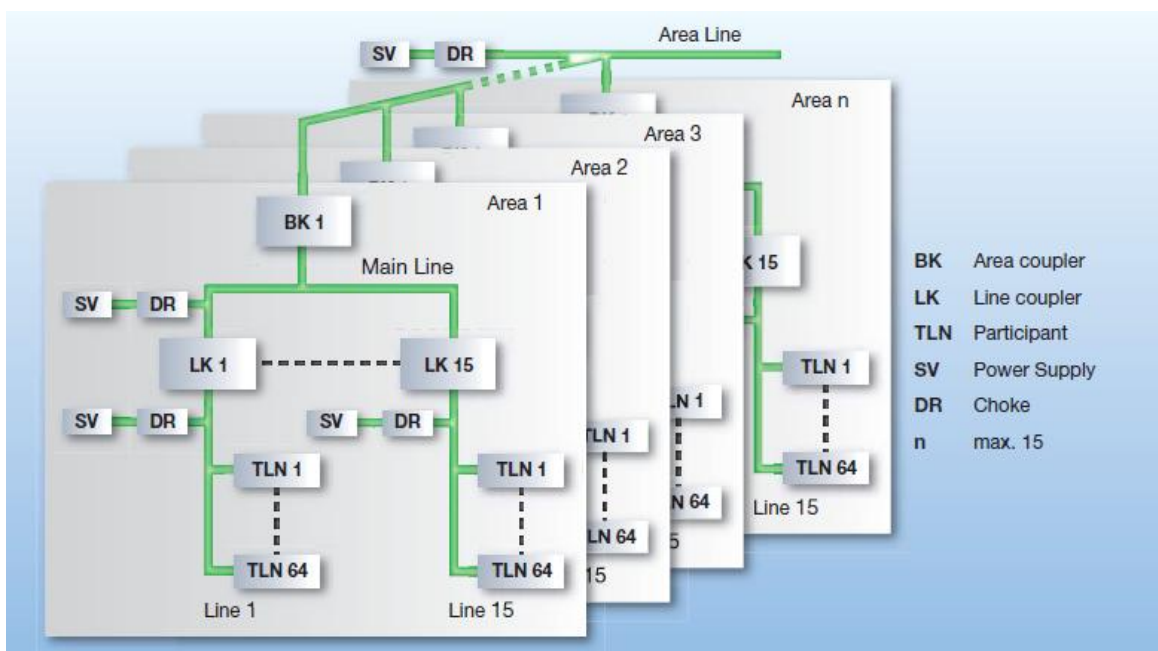
Konnex je naziv za krovnu europsku organizaciju koja se bavi definicijom i unaprijeđenjem standarda kućne automatizacije. Obzirom na objekt svoga djelovanja, taj se naziv odnosi i na samu tehnologiju kojom se postiže sustav baziran na takvom standardu. Ovaj standard najzastupljeniji je u Europskoj uniji, posebice u Njemačkoj. Standard zaživljava 1999. godine kada se, do tada, najpoznatiji standard EIB i nešto manje korišteni standardi BCI (engl. *Batibus Club International*) i EHS (engl. *European Home Systems*) objedinjuju u zasebni Konnex KNX standard. Danas KNX organizacija broji više od 300 proizvođača. Kao otvoreni standard, KNX je priznat po međunarodnom *ISO/IEC 14543-3* i europskom *CENELEC EN 50090* kriteriju. Obzirom na opseg implementacije standarda EIB do tog trenutka, KNX standard poprima gotovo jednaku strukturu kao i EIB. Ostala dva standarda uvode proširenje opsega na određene funkcionalne uređaje i prilagodbu određenih komunikacijskih postupaka. Stoga možemo standard Konnex promatrati kroz standard EIB [2].

European Installation Bus – EIB

1990. godine, vodeće europske tvrtke u domeni kućne i zgradarske automatizacije osnivaju asocijaciju EIBA (engl. *European Installation Bus Association*). Vodeća je tvrtka u ovom procesu bio Siemens. Asocijacija dobiva zadaću izvršenja i nadzora standardizacije tehnologije te certifikacije uređaja dotičnih kompanija. Rezultat djelovanja je otvoreni standard EIB koji novim malim proizvođačima, članovima asocijacije, omogućava lakši razvoj uređaja i pristup tržištu.

Protokol se bazira na ISO/OSI modelu sa 5 slojeva. Obzirom na privrženost kućnoj i zgradarskoj automatizaciji, sama topologija poprima strukturni izgled stambenog objekta (slika 5). Topologija je sabirnička sa određenim osobinama stabla. Standard pridjeljuje katovima ili prostorijama određene zone (engl. *Area*). Broj zona je maksimalno 15.

Određenim zonama pridodijeljeno je do 15 linija (engl. *Line*) koje se mogu klasificirati po funkcionalnosti. Svaka pojedina linija sadrži do 64 ili 256 završnih uređaja tzv. *end devices*, ovisno o verziji standarda. Na slici su oni označeni oznakom TLN. Time se može postići broj od 57 600 uređaja integriranih u sustav pametne kuće. Pojedine linije i zone su međusobno odvojene tzv. odvajačima (engl. *coupler*). Linije su odvojene linijskim odvajačima (engl. *line couplers*) a zone sa zonskim odvajačima (engl. *area coupler*). Odvajači se ponašaju kao usmjerivači (engl. *routers*) tako da filtriraju dobivene podatkovne okvire ovisno o zoni ili liniji kojoj su namijenjeni. Oni tako pridonose brzini rada i uštedi podatkovnog kapaciteta. U EIB strukturama svaka linija i zona ima zaseban izvor napajanja (engl. *power supply*). Obzirom da se linije napajanja vode zajedno sa linijama podataka potrebno je smanjiti utjecaj smetnji napajanja na signale podataka. Za to služe prigušivači smetnji (engl. *choke*).



Slika 5: Topologija EIB sustava

Osnovni medij koji se koristi u ovoj sabirničkoj strukturi je upletena parica koja na određenim daljinama guši signal. Stoga je, na određenim mjestima, potrebno na same linije signala postaviti pojačivače (engl. *repeaters*). Oni omogućuju funkcioniranje sustava sa uređajima do 1000 metara udaljenosti. Sabirnička struktura za medij može koristiti i frekvencijski pojas na kojemu određena frekvencija u dopuštenom pojasu (kanal) određuje liniju sabirnice. Ovaj medij je pogodan za naknadnu implementaciju, ukoliko se nije vodila

briga o nadogradnji sustava. No, ovaj medij iziskuje odašiljače i prijemnike na svakom uređaju što podiže cijenu u odnosu na preinstaliranu žičnu sabirnicu.

EIB mreža je čista slobodna mreža bez poslužitelja (engl. *peer to peer*) gdje svaki čvor ispituje stanje i zauzetost sabirnice bez odobrenja nadležnog računala. No, upravljačko korisničko sučelje vrši određenu kontrolu cijelog sustava kao što to čini upravljački uređaj u sustavu inteligentnog upravljanja. To prije svega vrši time što ima najveći prioritet pristupa sabirnici. Pristup sabirnici se odvija CSMA/CA postupkom arbitracije gdje svaki uređaj šalje prioritete bitove na sabirnicu i nakon svakog poslanog bita čita sabirnicu. Ukoliko se javlja nesklad poslanog i očitog bita, uređaj se povlači sa sabirnice do idućeg pristupa. Prijenos podataka se odvija brzinom od 9.6 kbit/s.

U sustavu EIB uređajima se pridijeljuju fizičke i grupne adrese. U fizičkoj je adresi sadržana informacija o zoni, liniji i rednom broju uređaja u liniji, dok se u grupnoj adresi nalazi informacija o funkcijskoj skupini uređaja (funkcionalni senzori, aktuatori...). Adresa se u uređaj unosi pri konfiguraciji uređaja i zauzima dva bajta u podatkovnom okviru koji se prenosi između uređaja u sustavu EIB. Izgled podatkovnog okvira (engl. *data frame*) koji se prenosi sabirnicom dan je na slici 6.

C field (8 bit)	Source field (16 bit)	Destination field (17 bit)	RL field (7 bit)	Data field (1 - 16 Byte)	P field (8 bit)
--------------------	--------------------------	-------------------------------	---------------------	-----------------------------	--------------------

Slika 6: Podatkovni EIB okvir

C polje je kontrolno polje koje se sastoji od 8 bitova. U njemu je pohranjena informacija o prioritetu poruke (sistemski, alarmni, visoki i niski prioritet). Također govori da li je tekući okvir normalno plasiran ili je ponovljen zbog neke greške. *Source* polje označava adresu pošiljatelja poruke i sastoji se od 16 bita. 4 bita govore u kojoj se zoni nalazi uređaj, 4 bita govore o liniji u toj zoni, a ostalih 8 bitova govore o rednom broju uređaja u liniji. Slično vrijedi i za *Destination* polje koje označava određeni uređaj, s malom razlikom što ovo polje ima 17 bitova. Rekli smo već ranije da se uređajima pridodaje fizička adresa (zona, linija, uređaj), ali i grupna adresa koja uređaj smještava u određenu funkcionalnu grupu (bitovna se reprezentacija određuje prilikom konfiguracije). Stoga prvih 16 bitova označava ili fizičku ili grupnu adresu uređaja, a 17. bit govori o kojoj se vrsti adrese radi. Polje *RL* sadrži informaciju o stanju brojača u vezi (3 bita) i broju

nadolazećih bajtova u podatkovnom polju (4 bita – do 16 bajtova). Brojač služi u komunikacijskom protokolu mrežnog sloja koji na temelju iznosa brojača pristupa slanju podataka ili prestaje sa istim. Sve bitove u navedenim poljima uređuje podatkovni sloj osim brojača kojeg uređuje mrežni sloj. Zatim slijedi podatkovno polje *Data* koje je zapravo jezgra komunikacijskog protokola. Sastoji se od najviše 16 okteta bitova, a prvi broj bitova već je unaprijed pohranjen u *RL* polju. Prvi bajt je uvijek uređen sa strane transportnog sloja i sadrži informaciju o zadaći i karakteru cijele poruke: da li je to zahtjev za otvaranjem ili ukidanjem veze ili je to sam podatak koji se prenosi i prenosi li se on jednom uređaju, grupi uređaja ili cijelom sustavu (engl. *broadcast*). Također, karakter može biti i pozitivna ili negativna potvrda prijema (engl. *acknowledgement*). Ukoliko postoji više bajtova u ovom polju, drugi bajt je kontrolni bajt aplikacijskog sloja. On govori o samoj zadaći koju je korisnik indirektno zatražio: pisanje u uređaj, čitanje iz uređja, postavljanje grupne adrese, unošenje autorizacije, resetiranje uređaja... Ostali bajtovi ovise o zadaći koja se izvršava i predstavljaju čisti podatak kao određenu veličinu, iznos ili šifru. Zadnje polje *P* je polje provjere ispravnosti poruke. Kada odredište dobije poruku provodi paritetno ispitivanje i uspoređuje rezultat sa primljenim *P* poljem. Provodi se vertikalna i horizontalna provjera (tablica 1). Nakon toga se ustanovljuje eventualna pogreška i traži se retransmisija poruke.

Tablica 1: *Provjera pariteta u EIB poruci*

FIELD	D0	D1	D2	D3	D4	D5	D6	D7	P
C field	1	0	1	1	0	0	0	0	1
Source (1. Byte)	0	0	0	0	0	0	0	1	1
Source (2. Byte)	0	0	0	0	1	0	0	1	0
Destiantion (1. Byte)	0	0	0	0	0	0	0	1	1
Destiantion (2. Byte)	0	0	0	0	0	1	1	1	1
RL field	0	0	0	1	0	0	0	1	0
Data (1. Byte)	0	0	0	0	0	0	0	0	0
Data (2. Byte)	1	0	0	0	0	0	0	1	0
P field	1	1	0	1	0	0	0	1	0

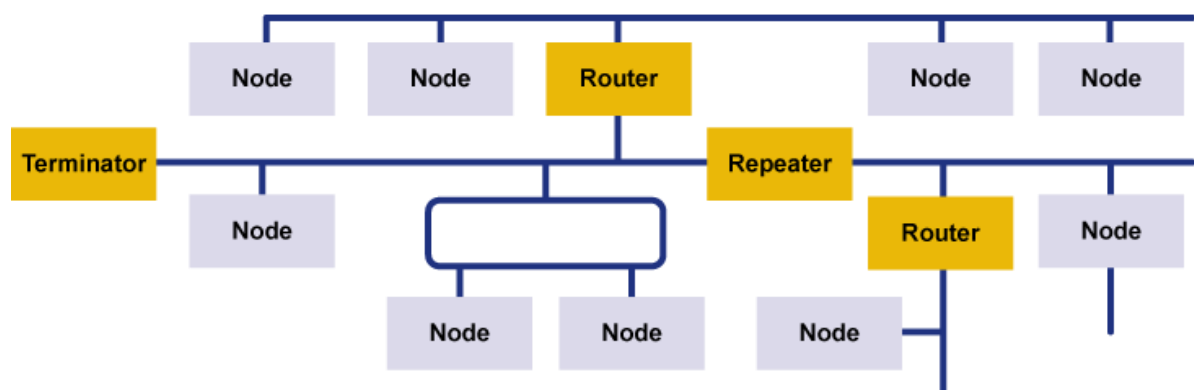
Horizontalno se provodi parna provjera pariteta, a vertikalno neparna. Konačno se *P* polje zaključuje parno provjerenim bitom. Obzirom da smo rekli da *Destination* polje ima 17 bita, a *RL* polje 7 bita, pri provjeri pariteta se zadnji odredišni bit pridodaje *RL* polju. U tablici su, radi primjera, navedena samo dva bajta podatkovnog polja. Jasno je da se

provjera pariteta vrši nad svim prisutnim oktetima u poruci. Ovime smo završili pregled jednog od najbitnijih stambenih automatizacijskih standarda [1].

3.4.2 LonWorks

Godine 1988. osnovana je američka korporacija Echelon nakon što se dio osoblja kalifornijske tvrtke za kontrolne sustave Rolm, nezadovoljno novim vlasnikom IBM-om, udružuje sa Mikeom Markkulom, bivšim predsjednikom tvrtke Apple Computers. Zadaća novonastale korporacije je bila definiranje novog *peer-to-peer* mrežnog protokola za integraciju i kontrolu uređaja, uz realizaciju osnovnog strukturnog sklopa i novog programskog jezika za taj protokol. Odlučuje se da će taj protokol biti otvoren te da će omogućavati korisnicima nesmetani razvoj inteligentnog sustava. Tako 1990. godine nastaje LonTalk protokol baziran na ISO/OSI modelu sa sedam slojeva. Sklopovlje koje se koristi u ovakvom protokolu se naziva *Neuron Chip*, a programski jezik *Neuron C*, baziran na ANSI C jeziku. 1999. godine Echelon na svoje stranice postavlja cijeli postupak implementacije protokola kako bi se uređaji mogli sami razvijati bez potrebe nabavljanja *Neuron Chip* sklopovlja. Cjelokupni standard za automatizaciju naziva se LonWorks.

Ovaj se standard koristi u zgradarskoj automatizaciji, ali više u industrijskim postrojenjima gdje je uređaje potrebno funkcionalno povezati te je potrebna fleksibilnost. Stoga je topologija koja se koristi u ovom standardu slobodna (slika 7). To je zapravo razgranata sabirnička topologija sa elementima stabla i heterogene forme.



Slika 7: LonWorks slobodna topologija

Funkcija usmjerivača i pojačivača jednaka ja kao kod EIB-a, samo što u ovom slučaju usmjerivači filtriraju podatke ovisno o funkcionalnoj grupi. Uloga zaključivača (engl. *terminator*) je prilagodba impedancije u sabirnici. Broj čvorova u mreži ovog protokola je

malen i najviše iznosi 128 što omogućuje slobodnu topologiju sustava. Transmisija podataka je serijska i vrlo prilagodljiva. Za arbitraciju sabirnice koristi tzv. prediktivnu upornu CSMA arbitraciju (engl. *predictive persistent CSMA*) kojom uređaji nasumično pristupaju sabirnici kako bi se izbjeglo istovremeno ponovno pristupanje dva uređaja koji su u prošlom ciklusu pristupa istovremeno bili odbijeni zbog čvora većeg prioriteta. Ovakvim se pristupom smanjuju kolizije i povećava brzina. Obzirom da imamo manji broj čvorova i slobodnu topologiju, podatkovni okvir je manji te također pridonosi ostvarenju većih brzina prijenosa. Mediji koji se koriste u ovom protokolu su upletena parica, radio-frekvencija i gradska mreža.

Tablica 2: Tipovi LonWorks protokola

TIP	MEDIJ	TOPOLOGIJA	BRZINA	MAKS. BROJ ČVOROVA	MAKS. UDALJENOST
TP/FT-10	Upletena parica	Slobodna, sabirnička	78 kbit/s	64-128	500 m(slobodna top.) 2200 m (sabirnička top.)
TP/XF-1250	Upletena parica	Strogo sabirnička	1.25 Mbit/s	64	125 m
PL-20	Gradska mreža	Ovisna o gradskoj mreži	5.4 kbit/s	Ovisi o mreži	Ovisna o izvedbi mreže
IP-10	Internetski protokol	Ovisna o IP mreži	Ovisna o IP mreži	Ovisna o IP mreži	Ovisna o IP mreži

Sama brzina ovisi o načinu izvedbe i mediju koji se koristi za prijenos. Razlikujemo 4 osnovna tipa mreže protokola LonWorks kao što je prikazano tablicom 2. Njih nazivamo kanalima. TP/XF-1250 tip ostvaruje najveću brzinu rada, ali ograničava broj čvorova i traži fizički spoj upletenom paricom. Tip PL-20 koji koristi gradsku mrežu za medij ima najmanju brzinu prijenosa što je rezultat sklopovlja za filtraciju i (de)modulaciju signala koji prenosi podatke i malog podatkovnog propusta. Većim brojem uređaja u ovom tipu izvedbe smanjujemo brzinu zbog povećavnja podatkovnog okvira. Poseban tip je IP-10, sustav baziran na internetskom povezivanju čvorova. Koriste se posebni LonTalk-IP *routeri* koji povezuju međusobno odvojene sabirnice sa čvorovima preko internet veze. Određeni dio uređaja u LonTalk protokolu može biti povezan radio vezom u sustav sabirnice kao što je to opisano u poglavlju EIB sustava. LonWorks kanali sa ograničenim

brojem čvorova mogu se povezivati u veće skupine koje čine određenu domenu. To omogućava integraciju većeg broja čvorova u sustav. Do 256 kanala (mreža) možemo povezati u sustav domene što znači da, obzirom da je najveći broj uređaja u kanalu 128, možemo integrirati do 32 768 čvorova. Po toj se strukturi vrši i adresiranje u komunikaciji. Moguće je i grupno adresiranje (adresiranje po funkcionalnosti).

Za programsku podršku za *Neuron C* jezik Echelon nudi dva razvojna alata: NodeBuilder i LonBuilder. NodeBuilder se koristi za programiranje i razvoj pojedinih čvorova baziranih na *Neuron Chip* strukturi. LonBuilder istovremeno programira i povezuje više čvorova u sustav [3].

3.4.3 BACNet

Razvoj protokola započinje 1987. godine u Nashvilleu, Tennessee, kada skupina članova ASHRAE udruženja (engl. *American Society of Heating, Refrigeration and Airconditioning Engineers*) kreće sa razvojem vrsta podataka, naredbi i aplikacijskih usluga za novi standard u stambenoj automatizaciji. BACNet postaje *ASHRAE/ANSI 135* standard 1995. godine i pod vlasništvom je ASHRAE udruženja. Nedugo nakon objavljivanja standarda, tvrtka za HVAC automatizaciju Alerton objavljuje da je realizirala cjelokupni HVAC sustav pomoću BACNet protokola. Uskoro i ostale vodeće tvrtke za sustave grijanja, hlađenja i klimatizacije preuzimaju BACNet kao temeljni protokol. 2003. godine BACNet postaje i *ISO 16484-5* HVAC standard.

Tablica 3: *BACNet komunikacijski model*

BACNet slojevi					ISO/OSI slojevi
BACNet sloj za aplikaciju					4. - 7. sloj
BACNet sloj za posredovanje					3. Mrežni sloj
IEEE 802.2	BVLL - IP	MS/TP (<i>Master-Slave/Token Passing</i>)	PTP (<i>Point-to-Point</i>)	LonTalk	2. Podatkovi sloj
ARCNET; IEEE 802.3 (Ethernet)		RS485	RS232		1. Fizički sloj

BACNet protokol isključivo se koristi u zgradarskoj automatizaciji, prvenstveno za HVAC zadaće. Prijenos podataka u fizičkom sloju se obavlja raznim standardima: RS232,

RS485, Ethernet, LonTalk serijska veza, itd. Stoga topologija, brzina, broj čvorova i medij ovise o načinu prijenosa podataka. BACNet se temelji na 4-slojnom komunikacijskom modelu kao što je prikazano tablicom 3. Uz BACNet slojeve prikazani su ekvivalentni slojevi ISO/OSI modela [7].

3.4.4 DALI

DALI (engl. *Digital Addressable Lighting Interface*) standard je IEC 60929 standard za mrežu sustava za kontrolu rasvjete u zgradama. Vlasnik protokola je elektrotehničko udruženje ZVEI (njem. *Zentralverband Elektrotechnik und Elektronikindustrie*) iz Njemačke.

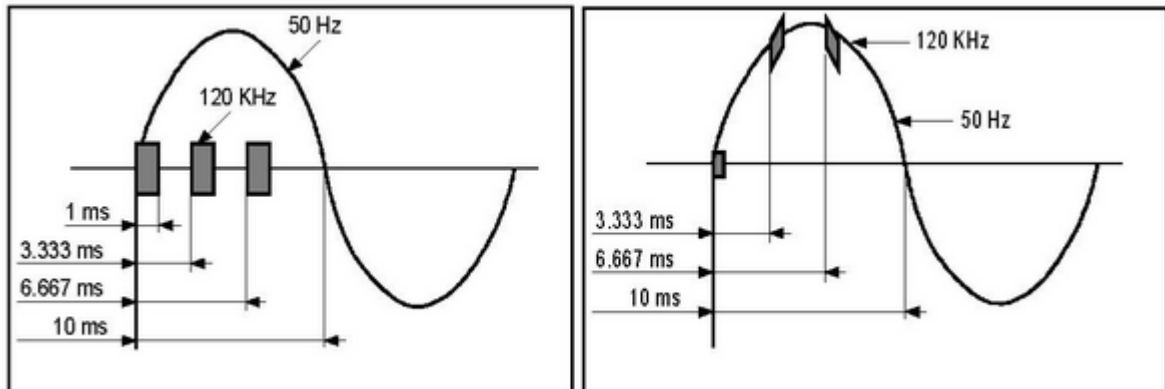
DALI je jednostavan koncept koji se sastoji od kontrolera koji upravlja rasvjetnim uređajima sa DALI sučeljem. Koristi se sabirnička ili zvjezdasta topologija ili pak kombinacija tih dviju topologija. U mrežu sa jednim kontrolerom je moguće povezati do 64 rasvjetna uređaja, no takvu je mrežu moguće povezati preko pristupnika (engl. *gateways*) u sustav više povezanih mreža koji je sličan domeni u LonWorks standardu. Prijenos se ostvaruje asinkronom serijskom vezom pomoću upletenih parica, a brzina prijenosa je mala i iznosi 1.2 kbit/s [14].

3.4.5 X10

Jedan od prvih komunikacijskih protokola u kućnoj automatizaciji je X10. Kao medij prvenstveno koristi gradsku mrežu, a postoje i izvedbe sustava sa X10 protokolom sa radio vezom. Razvijen je 1975. godine od strane male škotske tvrtke Pico Electronics kako bi omogućio daljinsku kontrolu kućanskih aparata. Ta tvrtka je danas dio X10 Ltd. Ovaj protokol je zapravo začetnik kućne automatizacije.

Podaci se kodiraju nosiocem frekvencije od 120 kHz koja se superponira signalu gradske mreže (50/60 Hz). Podatkovni okvir se sastoji od adrese uređaja i zadaće (uključi, isključi, pojačaj,...). Logička jedinica je predstavljena nosiocem frekvencije 120 kHz u trajanju od 1 ms. Nula je predstavljena odsustvom nosioca 120 kHz. Svaki se bit postavlja tri puta zaredom unutar 10 ms kako se ne bi glavnom signalu superponirao u trenutku prolaska kroz nulu (slika 8). To je bitno jer se sinkronizacija u uređajima obavlja točno u trenutku prolaska glavnog signala kroz nulu. Ukoliko bi se bitovi nalazili u nuli glavnog signala u trenutku očitavanja oni ne bi bili pročitani. Sve poruke se šalju dvaput u svrhu

pouzdanosti. Svaki okvir počinje sa startnim kodom 1110, a okviri su međusobno odvojeni sa najmanje šest nula.



Slika 8: Modulacija poruke u X10 protokolu

Nedostaci ovog protokola su brojni. Gradska mreža često sadrži signale smetnje i prenapona što uzrokuje krivu interpretaciju poruke. Moguće su kolizije i preslušavanja ukoliko dva nadzorna uređaja istovremeno komuniciraju mrežom. Zbog zahtjeva za trajanje signala koji reprezentiraju bitove, brzina prijenosa biva mala. Pouzdanost ovisi o vrsti uređaja koji se ugrađuje. Moguće je ugraditi čvorove koji potvrđuju prijem poruke, no oni su višestruko skuplji od primajućih čvorova. Usprkos ograničenim mogućnostima (frekvencijski pojas, brzina), pristupačnom cijenom ugradnje uređaja X10 do danas ostaje vrlo popularan standard u kućnoj automatizaciji [15].

3.4.6 ZigBee

ZigBee je naziv kako za tehnologiju uređaja i veze tako i za standard bežičnog umrežavanja koji omogućuje jednostavnu i cjenovno prihvatljivu integraciju uređaja u mrežu. Razvila ga je ZigBee Alliance za jednostavnije mreže sa malim dometom i manjim brzinama prijenosa. Otvoreno udruženje ZigBee Alliance nastaje 2002. godine sa ciljem unaprijeđenja energetske učinkovitosti i uštede energije te umrežavanja uređaja u stambenim objektima. Otvoreno udruženje označava mogućnost svakome da postane član i da koristi standard za svoj razvoj. 2006. godine plasirani su prvi ZigBee uređaji povezani u kućnu mrežu. Danas je ZigBee standard uvelike korišten u kućnoj i zgradarskoj automatizaciji te broji više od 600 certificiranih uređaja, a više od 400 kompanija-članova diljem svijeta koristi ovaj standard u razvoju svojih sustava.

ZigBee tehnologija se temelji na fizičkom radio standardu *IEEE 802.15.4* koji koristi zakonski dopuštene frekvencijske pojaseve: 2.4 GHz (globalno ISM područje), 915 MHz (SAD) i 868 MHz (EU). Područje od 2.4 GHz koristi do 16 kanala i ostvaruje brzinu do 250 kbit/s, područje od 915 MHz sa 10 kanala ostvaruje brzinu do 40 kbit/s dok područje od 868 MHz u jednom kanalu ostvaruje brzinu od 20 kbit/s. Ovisno o razini smetnji i infrastrukturi ovom se tehnologijom povezuju uređaji na udaljenosti od 10 do 1600 metara. Danas ZigBee Alliance nudi tri otvorene specifikacije za ZigBee tehnologiju (slika 9) :

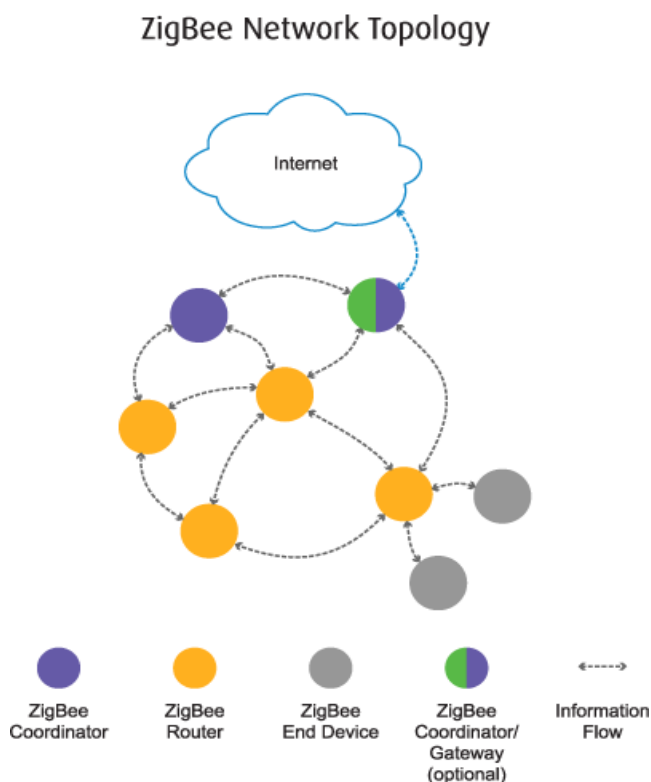
- ZigBee
- ZigBee IP
- ZigBee rf4ce



Slika 9: ZigBee specifikacije standarda

ZigBee je najkompleksnija specifikacija ovog standarda, a zadnja inačica ZigBee 2012 nudi potpuno povezanu mrežu sa više od 64 000 integriranih interoperabilnih uređaja različitih funkcionalnosti. ZigBee Alliance uređaje iste funkcionalnosti svrstava u zajedničku grupu za koje propisuje standard koji je kompatibilan sa svakim drugim ZigBee funkcionalnim standardima. Tako ZigBee specifikacija u cjelokupni sustav povezuje sustav kućne i zgradarske automatizacije sa sustavom energetske učinkovitosti, pametne kupovine,

zdravstvene skrbi, mobilne kontrole i rasvjete. Ova specifikacija standarda koristi zvjezdasto – prstenastu topologiju (slika 10) te međusobnu PAN (engl. *Personal Area Network*) komunikaciju. ZigBee koordinatori kontroliraju formu i sigurnost mreže te ih se dodaje u prsten. Oni zapravo povezuju PAN-ove. ZigBee *routeri* služe kako bi mrežu proširili i u takvoj mreži usmjeravali podatke do krajnjih ZigBee uređaja (čvorova). Mreža može imati pristup internetu (i obratno) za dodatno proširenje funkcionalnosti. Pristup se vrši pomoću *gatewaysa* tj. pristupnika koji istovremeno vrše kontrolu forme i sigurnost mreže.

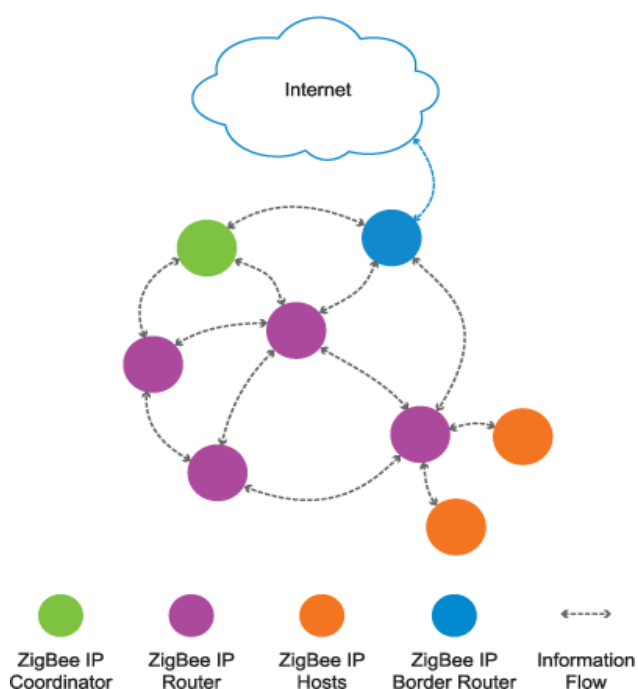


Slika 10: Topologija ZigBee specifikacije

ZigBee IP je prvi otvoreni standard za umrežavanje niskopotrošnih i niskobudžetnih uređaja temeljen na IPv6 protokolu. Osmišljen je kako bi povezivao sustave energetske učinkovitosti u svrhu postizanja uštede, a članovima udruženja je dostupan od 2013. godine. Osim na IPv6 protokolu, ovaj se standard bazira i na ostalim internetskim protokolima: 6LoWPAN, PANA, RPL, TCP, TLS i UDP. Uz poznata frekvencijska područja ovaj standard omogućuje korištenje frekvencijskog područja od 920 MHz u Japanu. Topologija je također zvjezdasto – prstenasta sa malim izmjenama u odnosu na ZigBee specifikaciju (slika 11). Koristi se međusobna PAN komunikacija. Mreža se sastoji

od koordinatora, usmjerivača tj. *routera* te tzv. domaćina (engl. *hosts*). Funkcije koordinatora i usmjerivača jednake su funkcijama kod ZigBee specifikacije- Uloga domaćina je isključivo vezana za energetske zadaće: senzorska mjerenja i kontrola određenim aktuatorima ili drugim sklopovljem. Primjerice, skup domaćina može tvoriti termostat. Domaćini su zapravo specificirani završni uređaji. Granični usmjerivač (engl. *border router*) internetski kontrolira mrežom te internetski ažurira stanje mreže i energetske pokazatelje.

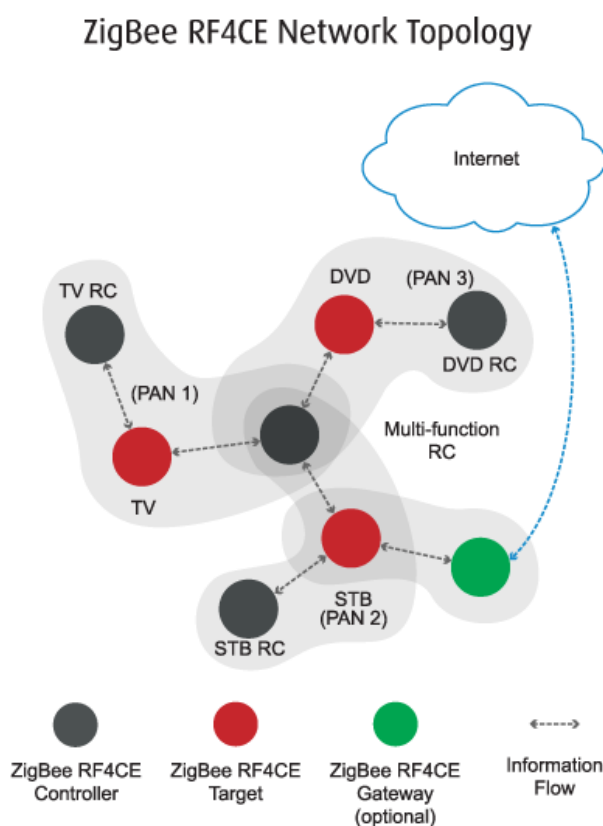
ZigBee IP Network Topology



Slika 11: Topologija ZigBee IP specifikacije

Zadnja specifikacija standarda je ZigBee rf4ce. Ona nudi neposredno, niskobudžetno i jednostavno rješenje za daljinsko upravljanje uređajima i malim sustavima povezivajući sustav daljinskog upravljanja sa ZigBee rješenjem ulaznih jedinica uređaja. Tako je, primjerice, moguće daljinski uključivati i isključivati kućanske aparate, zatvarati i otvarati kućna i garažna vrata, itd. Ova specifikacija je jako popularna i osigurava određenu razinu robusnosti. Obzirom da na frekvencijskom području od 2.4 GHz koristi samo tri frekvencijska kanala ostvaruje malene brzine prijenosa, no i one su i više nego dovoljne za normalan rad ovakvim standardom napravljenog sustava. Poput prvih dviju specifikacija i ZigBee rf4ce koristi međusobnu PAN komunikaciju, no temelji se isključivo na

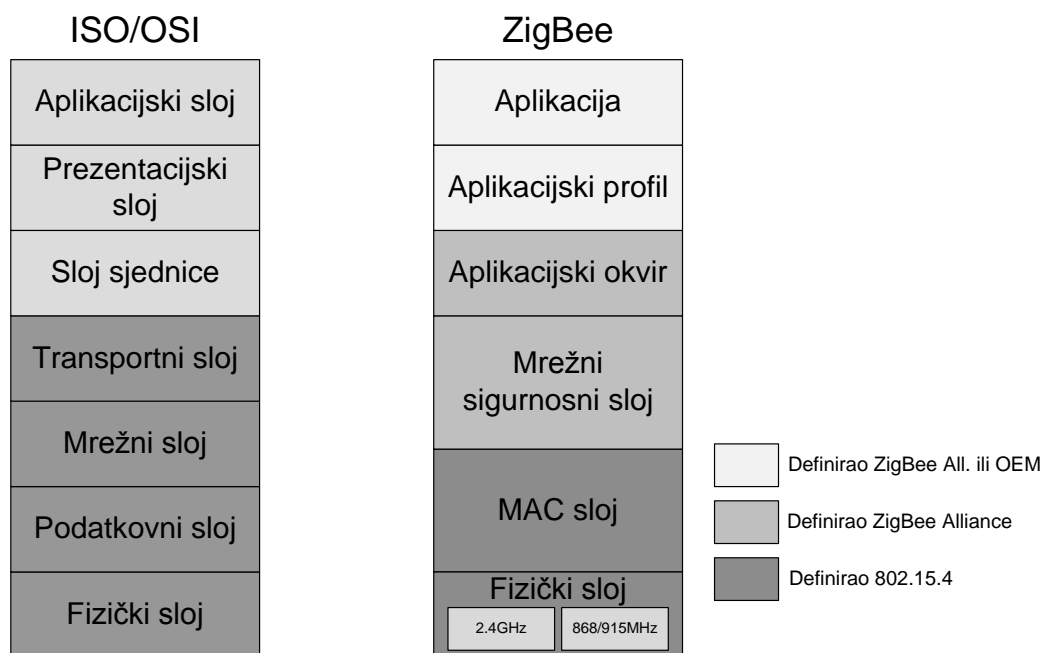
zvjezdastoj topologiji (slika 12). Mreža se sastoji od dviju vrsta uređaja: kontrolnih čvorova (engl. *control*) i odredišnih čvorova (engl. *target*). Odredišni čvorovi mogu biti TV, DVD uređaj, linija, itd. Mreža je podijeljena na zasebne mreže (PAN) gdje je svaki odredišni uređaj povezan sa svojim kontrolnim daljinskim uređajem. Svaki odredišni uređaj ima sposobnost PAN koordinatora tj. može započeti rad PAN mreže zasebno. Kontrolni uređaj se tada u mreži uparuje sa odredišnim uređajem. Multifunkcionalni kontrolni uređaj (engl. *Multi-function RC*) omogućuje upravljanje svim odredišnim čvorovima i tako povezuje odvojene PAN-ove u jednu mrežu (međusobna PAN komunikacija). Mreža se preko pristupnika povezuje na internet [12].



Slika 12: Topologija ZigBee rf4ce specifikacije

ZigBee standard se ne temelji na poznatom ISO/OSI modelu već ZigBee Alliance platformi komunikacijskog protokola kojim se mogu povući paralele sa ISO/OSI modelom (slika 13). Fizički je sloj, kako je već rečeno, određen standardom 802.15.4, a taj se standard primjenjuje i u MAC sloju (engl. *Medium Attachment Unit*). Modulacijski postupak koji se obavlja u fizičkom sloju je binarna diskretna fazna modulacija BPSK (engl. *Binary Phase Shift Keying*) za područje od 868 i 915 MHz te kvaternarna diskretna

fazna modulacija sa odmakom OQPSK (engl. *Offset Quaternary Phase Shift Keying*) za područje od 2.4 GHz. MAC sloj je u ISO/OSI modelima uvijek dio podatkovnog sloja i obavlja zadaću pripreme podatkovnog okvira i pristupa mediju tj. najčešće vrši zadaću arbitracije. U ZigBee modelu on obavlja i arbitracione i mrežne, ali i sinkronizacijske zadaće. Koristi se CSMA/CA arbitracija zajedno sa TDMA pristupom (engl. *Time Division Multiple Access*). TDMA je tehnika pristupa frekvencijskom kanalu u frekvencijskom pojasu veze. On omogućuje većem broju korisnika da rabi isti frekvencijski kanal.

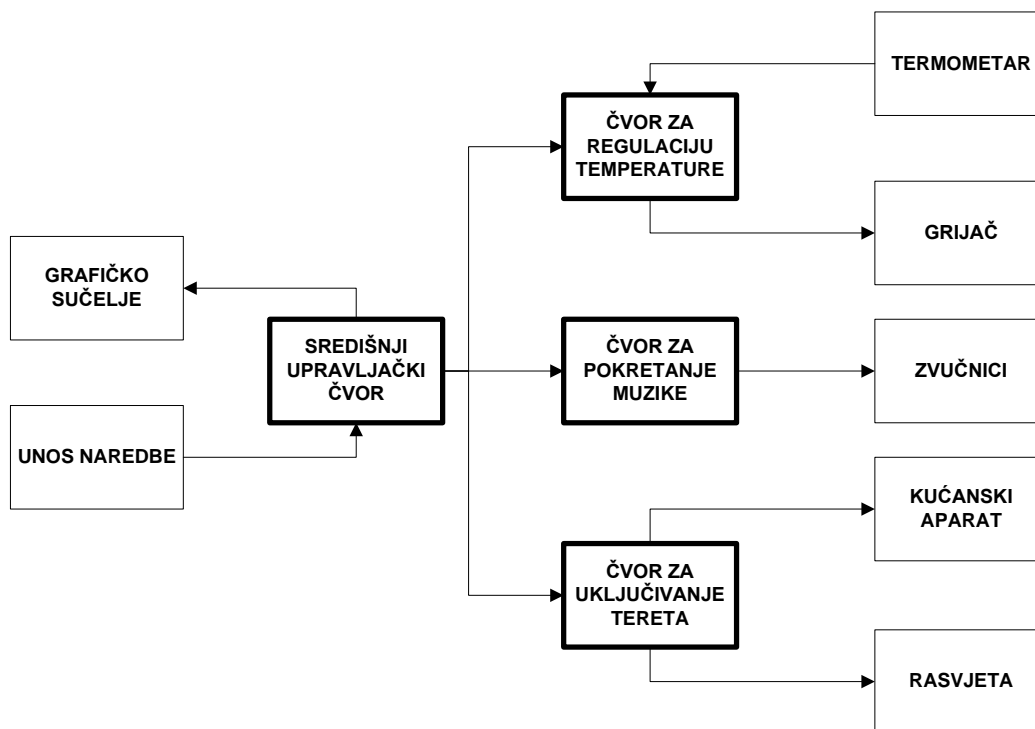


Slika 13: Usporedba ISO/OSI i ZigBee modela

Mrežni sigurnosni sloj određuje režim komunikacije: pojedinačni prijenos podataka (engl. *unicast*) namjenjen samo jednom čvoru, grupni prijenos (engl. *multicast*) namjenjen grupi uređaja ili sveopći prijenos tj. *broadcast* namjenjen svim uređajima u mreži. Gornja tri sloja ZigBee modela su aplikacijski slojevi i definiraju funkcionalne standarde koji se integriraju u ranije navedenim specifikacijama standarda: zadaće kućne i zgradarske automatizacije, HVAC, rasvjetu, energetske mehanizme, mobilni dom, itd. Aplikacijski slojevi mogu biti definirani i od strane OEM proizvođača opreme (engl. *Original Equipment Manufacturer*) [4].

4. Izvedba *Smart Home* sustava

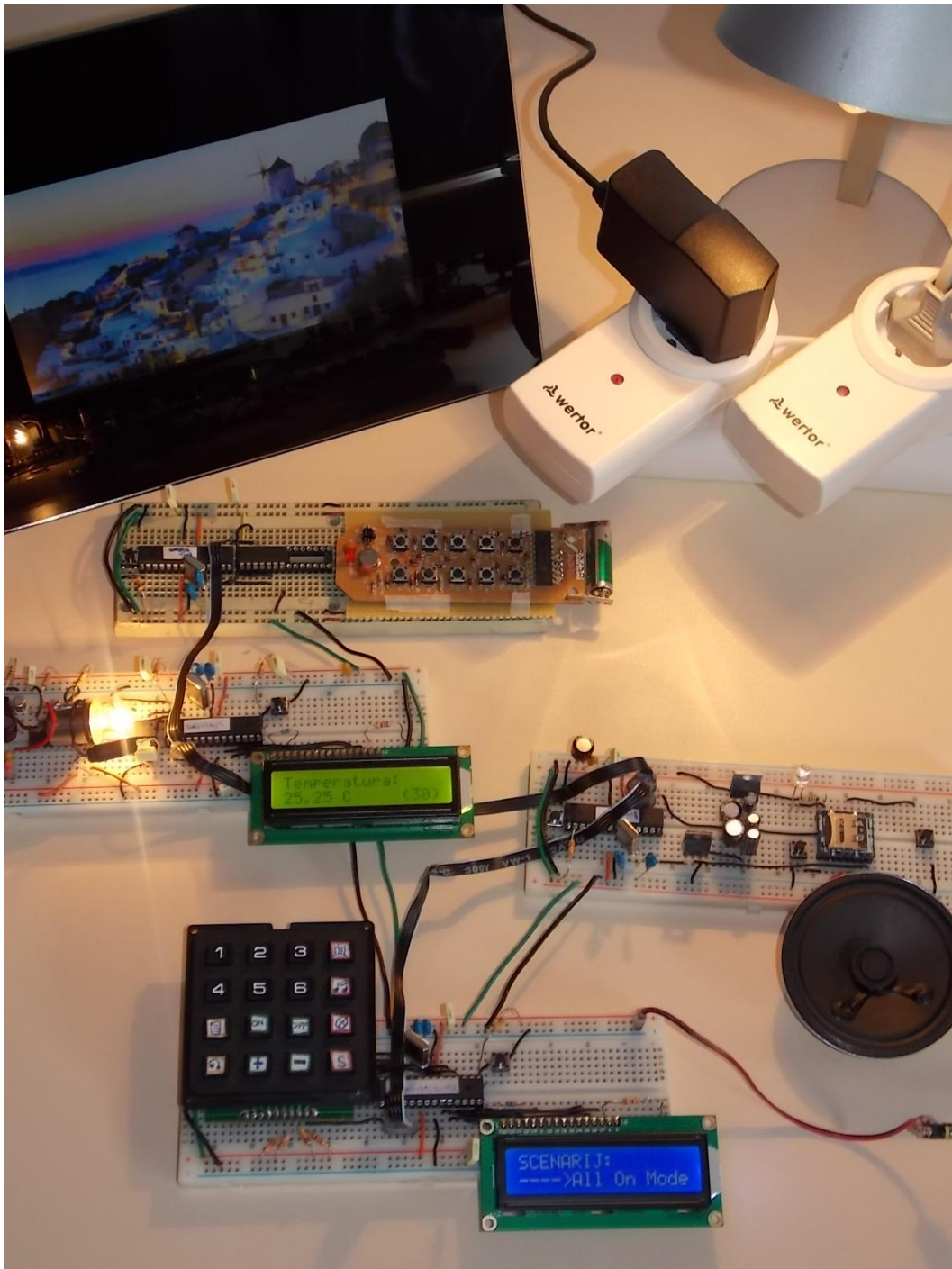
U okviru završnog rada realizirat ćemo jedan načelni sustav kućne automatizacije. Naš sustav će se sastojati od glavnog upravljačkog čvora i tri funkcionalna čvora. Jedan od tri funkcionalna čvora upravlja radio-frekvencijski sa dva komercijalno dobavljiva čvora tj. sa automatskim uključivačima tereta. Glavni upravljački čvor ujedno je i korisničko sučelje kojim korisnik upravlja sustavom pametne kuće. Prvi čvor obavlja funkciju regulacije temperature sukladno zadaći koja dolazi od glavnog čvora. Drugi čvor je zadužen za reprodukciju muzike i stvaranje željenog ambijenta. Treći čvor objedinjuje samostalno napravljeno i komercijalno sklopovlje te ima za svrhu uključivanje i isključivanje tereta na napon gradske mreže. Funkcionalnost uređaja se vidi na blok shemi danoj slikom 14.



Slika 14: Blok shema sustava pametne kuće

Kao napajanje za sklopovlje koristimo istosmjerni napon od 5 V, dok za pogon grijača u čvoru regulacije temperature (žarulja) i pogon daljinskog modula u čvoru uključivanja tereta koristimo istosmjerni napon od 12 V. Kao glavnu procesnu komponentu koristimo mikrokontrolere Atmega168. Obzirom da postoje četiri čvora u našem sustavu koristimo četiri takva mikrokontrolera. Kao topologiju mreže koristimo sabirnicu (vidi

poglavlje 3.2). Čvorovi su međusobno povezani pomoću četiri linije kojima glavni upravljački čvor jednosmjerno šalje podatkovni i sinkronizacijski okvir svim ostalim čvorovima (*broadcast*). Ostali čvorovi očitavaju sabirnicu i detektiraju podatkovne okvire, sve u skladu sa zadanom sinkronizacijom. Izgled cijelog sustava pametne kuće vidljiv je na slici 15.



Slika 15: Sustav pametne kuće

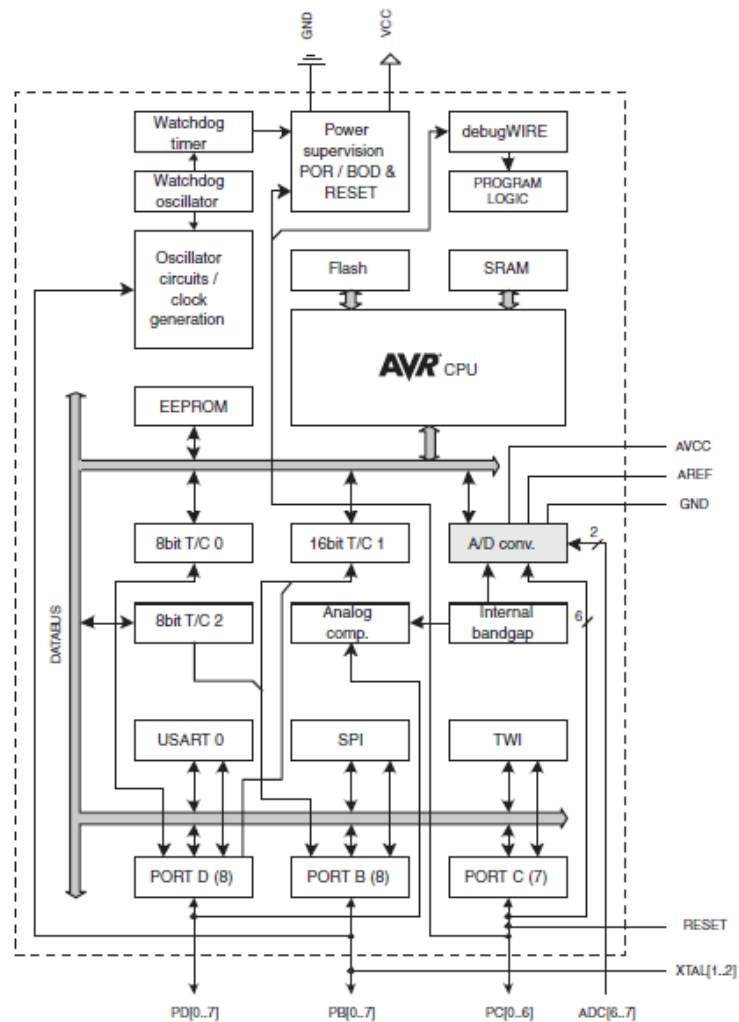
4.1 ATmega168 mikrokontroler

Čvorovi u sustavima kućne automatizacije općenito se temelje na mikrokontrolerima pa su njihove karakteristike vrlo bitne u projektiranju ovakvih sustava. Kao što je već navedeno, naš sustav pametne kuće se temelji na mikrokontroleru ATmega168 porodice AVR kojeg proizvodi tvrtka Atmel. To je visoko preformansni mikrokontroler sa sljedećim karakteristikama:

- 8-bitna RISC arhitektura
- 131 instrukcija, 32x8 registara opće namjene
- Do 20 MIPS-a pri frekvenciji od 20 MHz
- 16 kB programabilna Flash memorija
- 512 B EEPROM
- 1 kB unutarnji statički RAM
- 28 pinsko kućište sa 23 I/O pina
- Logika 0/5 V
- Temperaturno područje: -40°C do +85°C
- Mikrokontroler niske potrošnje

Na slici 16 prikazana je blok shema ATmega168 kontrolera. Napajanje se dovodi na sklopovlje za prilagodbu napajanja. Kontrolna procesorska jedinica jezgre AVR ima zaseban pristup Flash i SRAM memoriji, a s druge strane preko podatkovne sabirnice komunicira sa brojnim perifernim jedinicama. Između ostaloga, u periferne jedinice ubrajamo: dva 8-bitna i jedno 16-bitno brojilo, brojač stvarnog vremena sa zasebnim oscilatorom (RTS – engl. *Real Time Counter*), šest kanala pulsno širinske modulacije (PWM – engl. *Pulse Width Modulation*), po jedan 8-kanalni i 6-kanalni 10-bitni analogno-digitalni pretvornik (ADC – engl. *Analog-to-Digital Converter*), programabilni serijski USART (engl. *Universal Synchronous Asynchronous Receiver Transmitter*), serijsko sučelje SPI (engl. *Serial Peripheral Interface*), dvožično serijsko sučelje I2C, programabilni *watchdog* sklop za nadzor ispravnosti rada, analogni komparator, itd. Od 23 I/O priključka mikrokontrolera, njih šest (pin 23 – pin 28) se mogu koristiti kao analogni ulazni priključci koji se, ako je to definirano, dovode na ulaz analogno-digitalnog

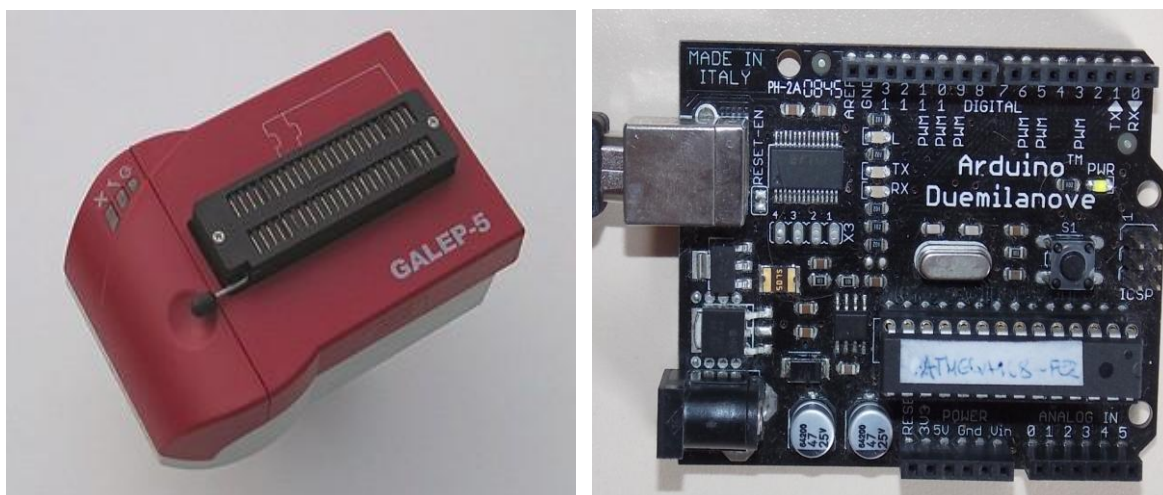
pretvornika. Nakon toga se na analognim signalima vrše operacije. Za pravilan rad AD pretvornika potrebno je dovesti referentno napajanje i referentni napon AD pretvorbe (priključci AVCC, AVREF, GND) [9].



Slika 16: Blok shema ATmega168 MCU

Mikrokontroleri porodice AVR se najčešće koriste u sklopu već izrađene razvojne pločice među kojima su najbrojnije Arduino pločice. Takvi mikrokontroleri već imaju uprženi tzv. *bootloader* koji omogućava pisanje i izvođenje naredbi od predefiniраниh memorijskih lokacija. Međutim, naši čvorovi koriste mikrokontrolere bez razvojnih pločica u tzv. unutarsklopovskoj izvedbi (engl. *in circuit*). Takvi kontroleri dolaze bez uprženog bootladera i potrebno ga je dodatno upržiti. To smo učinili pomoću univerzalnog programatora GALEP 5 koji se može koristiti na Zavodu za elektroničke sustave i obradu informacija Fakulteta elektrotehnike i računarstva u Zagrebu (slika 17a). Samo

programiranje mikrokontrolera vršimo pomoću razvojne pločice Arduino Duemilanove (slika 17b). Kako bi mikrokontroler samostalno radio, osim napajanja, potrebno mu je dovesti i takt iz vanjskog izvora. To postizemo 16 Mhz kristalom spojenim preko kondenzatora (22 pF) na masu. Nakon gore navedenih postupaka, naš je mikrokontroler spreman za rad.



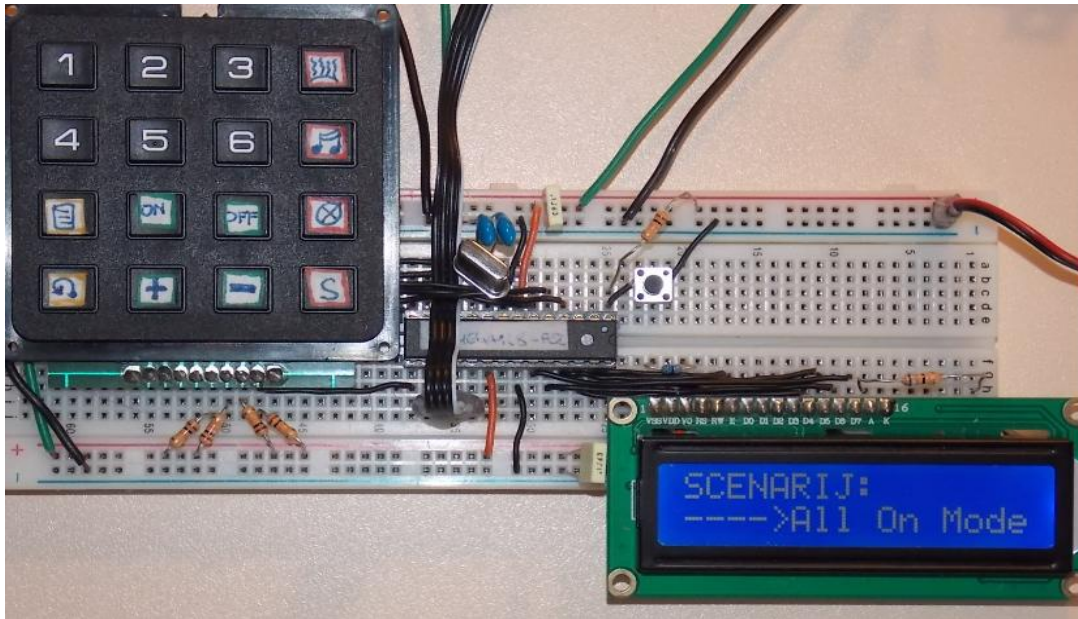
Slika 17: a) Galep 5, b) Arduino Duemilanove

4.2 Središnji upravljački čvor

Glavnu ulogu u sustavu pametne kuće ima središnji upravljački uređaj. Njime korisnik zadaje naredbe preko modificirane matrične tipkovnice, a sve u skladu sa izbornikom prikazanim na grafičkom LCD prikazniku (slika 18). U svakom trenutku je omogućen reset pritiskom na tipkalo. Korisnik može birati direktno upravljanje izvršnim čvorovima, ali može podešavati i scenarije. Naš sustav omogućava direktno podešavanje temperature, pokretanje i gašenje muzike te uključivanje i isključivanje rasvjete i kućanskog aparata (u našem slučaju digitalnog ovira za slike koji glumi televizor). Sam scenarijski odabir čest je koncept u modernim sustavima kućne automatizacije gdje korisnik jednim pritiskom pokreće predefimirane zadatke. Korisniku je osigurana jednostavnost i ušteda vremena. Naš sustav omogućava 5 scenarija:

1. Sleep Mode – scenarij spavanja: isključi rasvjetu i kućanske aparate, smanji temperaturu i pusti laganu opuštajuću pjesmu
2. Read Mode – scenarij čitanja: uključi svjetlo, podesi sobnu temperaturu, pusti laganu klasičnu muziku

3. Fun Mode – scenarij zabave: uključi “televizor“, pusti plesnu pjesmu i podesi sobnu temperaturu
4. News Mode – instant vijesti: pusti zvučni zapis aktualnih novosti (koristimo isti zvučni zapis vijesti – ovaj bi se podatak u modernom sustavu internetski ažurirao)
5. All On Mode – scenarij zimskog ulaska u kuću: uključi sve uređaje, temperaturu podesi na 30°C

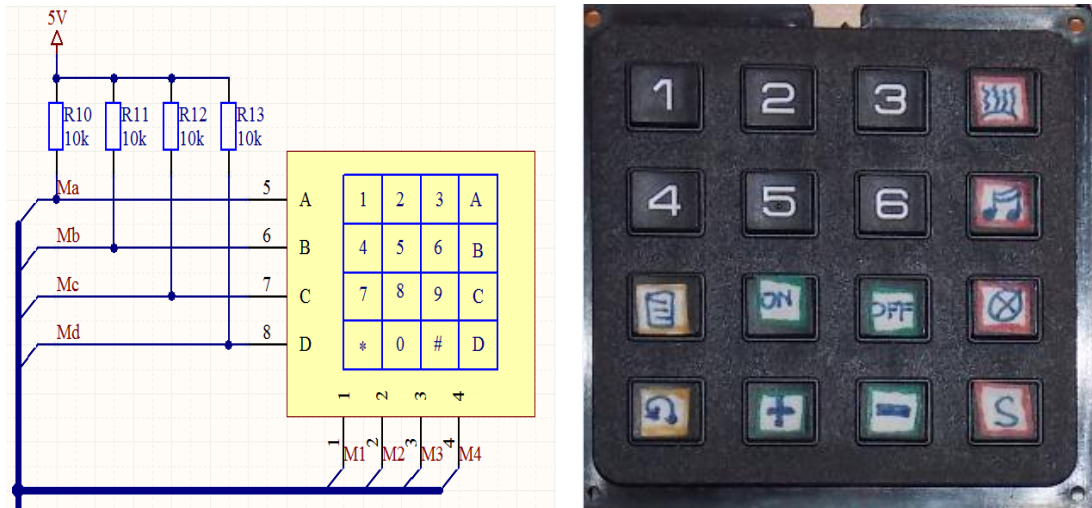


Slika 18: Upravljački čvor

4.2.1 Matrična tipkovnica

Glavna ulazna jedinica u sustav pametne kuće je matrična tipkovnica 4x4. Obzirom da na raspolaganju imamo popriličan broj priključaka mikrokontrolera svaki izvod tipkovnice spajamo direktno na mikrokontroler. Recni matrice su preko *pull-up* otpornika spojeni na napon napajanja. Blok shema matrične tipkovnice dana je na slici 19a.

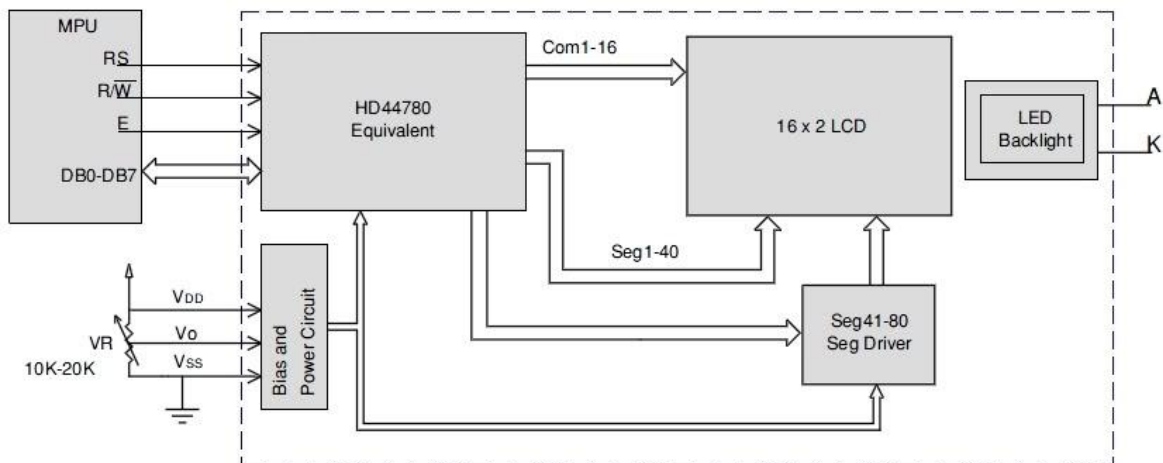
Tipke A, B, C i D preinačene su u tipke za zadaću: podešavanje temperature, muzike, trošila i scenarija (crvene tipke). Tipke od 1 do 5 se koriste za odabir određenog scenarija u slučaju scenarijskog odabira (crne tipke). Tipka 7 je tipka inicijalizacije programa tj. programski reset za cijeli sustav, a tipka “*” je tipka povratka u početni izbornik (žute tipke). Tipke 8, 9, 0 i “#” modificirane su u regulirajuće tipke “ON“, “OFF“, “+“ i “-“ kojima se direktno upravlja izvršnim čvorovima (zelene tipke). Tipka 6 se ne koristi u našem sustavu. Slika 19b prikazuje tipkovnicu modificiranu za sustav pametne kuće.



Slika 19: a) Blok shema tipkovnice, b) Modificirana tipkovnica

4.2.2 LCD prikaznik

LCD prikaznik glavna je izlazna jedinica sustava i u slučaju upravljačkog čvora prikazuje izbornik pametne kuće. Koristi se standardni LCD prikaznik sa ekvivalentnim Hitachi HD44780 kontrolerom (slika 20). Napajan sa naponom 5 V, ovaj prikaznik prikazuje znakove ASCII koda u 2 reda i 16 stupaca. Temperaturno područje rada sklopa je od -20°C do +70°C.



Slika 20: Blok shema LCD prikaznika

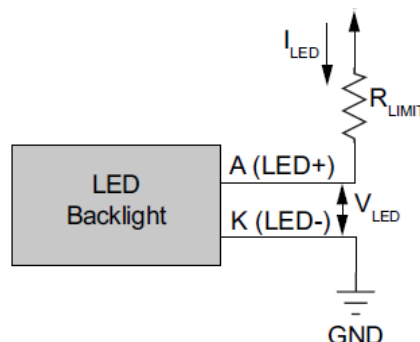
Prikaznik ima 16 priključaka kao što je prikazano slikom 21. Priključci D0-D7 služe za prijenos podataka u tj. iz prikaznika. Priključak RS služi za odabir registra (engl. *Register Select*) koji se koristi. Taj registar može biti podatkovni ili instrukcijski. Iz

podatkovnog se registra dobavljaju podaci, a iz instrukcijskog razne naredbe (čišćenje ekrana, pozicioniranje kursora itd.). R/W priključak određuje da li se iz prikaznika čitaju ili u njega pišu podaci (engl. *Read/Write*). E priključak omogućava sklopu pristup podatkovnoj sabirnici (engl. *Enable*). Dva su načina slanja podataka između LCD prikaznika i kontrolne jedinice: 4-bitno i 8-bitno paralelno sučelje.



Slika 21: LCD prikaznik

Priključci VSS i VDD su priključci napajanja. Priključkom V0 podešavamo kontrast pozadine i poruke tako da podešavamo otpor između priključka i napona napajanja. Za potrebe kontrasta spajamo otpornik od 3.3 k Ω . Ostali priključci A i K služe za LED podešavanje osvjetljenja. Na slici 22 je prikazan način spajanja ovih priključaka na napajanje.



Slika 22: Podešavanje pozadinskog osvjetljenja

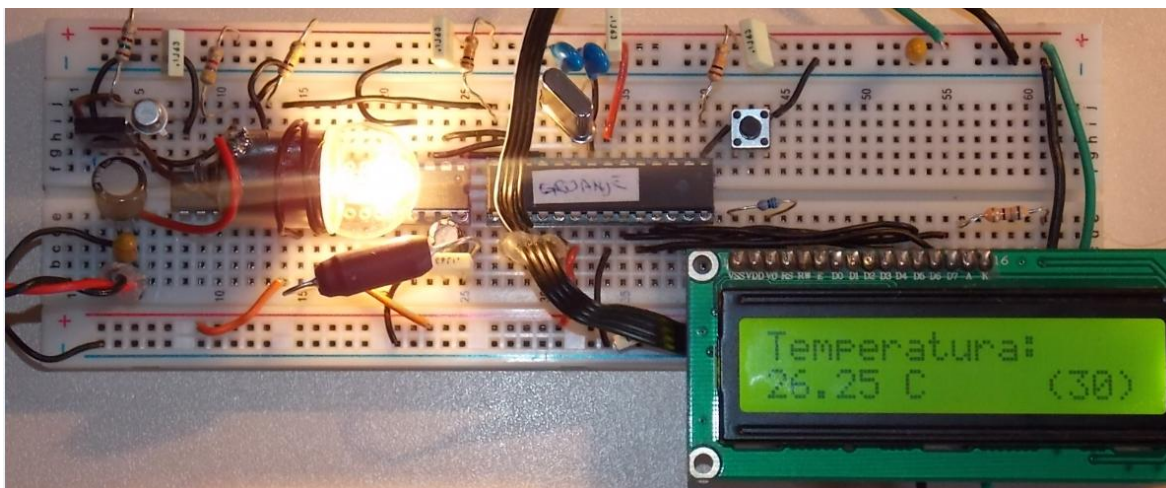
Struja koja protječe krugom je struja diode $I_{LED} = 25 \text{ mA}$. Napon na diodama je $V_{LED} = 3.3 \text{ V}$. Uz poznati napon napajanja $V_{DD} = 5 \text{ V}$ lako izračunavamo ograničavajući otpor R_{LIMIT} kojeg je potrebno staviti u krug da bi ograničavao struju. On iznosi 68Ω i osigurava željeno osvjetljenje:

$$R_{LIMIT} = \frac{V_{DD} - V_{LED}}{I_{LED}} = 68\Omega \quad (1)$$

Pri programiranju mikrokontrolera i povezivanju sa LCD prikaznikom koristimo otvorenu Arduino biblioteku "LiquidCrystal" koja se inicijalno nalazi u Arduino programskom paketu.

4.3 Čvor za regulaciju temperature

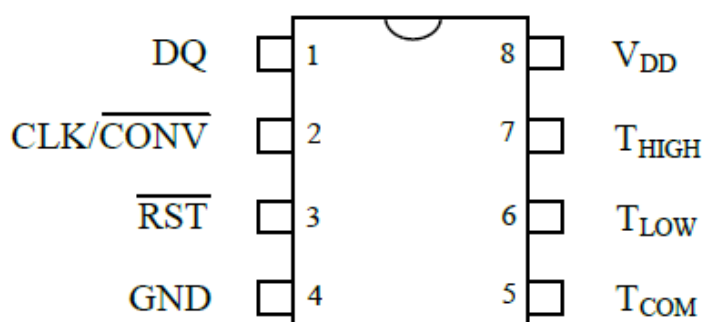
Regulacija temperature je postupak prisutan u gotovo svakom kućanstvu. U mnogim je on i zasebno automatiziran. Njegovo uklapanje u sustav inteligentnog upravljanja prisutan je u svim sustavima pametnih kuća jer on u velikoj mjeri osigurava energetska učinkovitost. U našem primjeru pametne kuće konstruiran je jednostavni digitalni termostat koji se sastoji od digitalnog termometra, grijača tj. žarulje i mikrokontrolera koji povezuje sustav (slika 23). Koristi se i pomoćno sklopovlje koje osigurava regulaciju. Također koristimo kontroler ATmega168 i LCD prikaznik temperature čija je funkcionalnost ranije opisana.



Slika 23: Čvor za regulaciju temperature

4.3.1 Digitalni termometar

Senzorsku i interpretirajuću ulogu u procesu regulacije temperature ima digitalni termometar DS 1620 tvrtke Maxim Integrated. To je 8 pinski sklop (slika 24) koji uz ulogu termometra može zasebno vršiti ulogu termostata. Mjeri temperature u rasponu od -55°C do $+125^{\circ}\text{C}$ sa razlučivosti od 0.5°C . Temperaturu prezentira kao 9 bitnu vrijednost koju serijskom vezom šalje na odredište. Serijska se veza ostvaruje preko tri priključka: DQ, CLK/CONV i RST.



Slika 24: Raspored pinova DS 1620

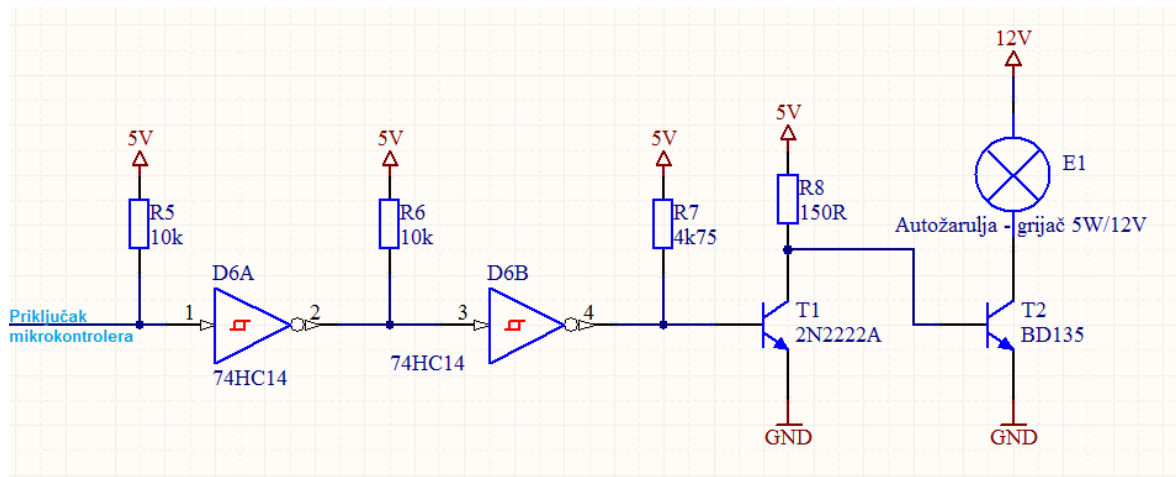
Osim priključaka napajanja postoje tri priključka koji obavljaju funkciju termostata: T_{HIGH} , T_{LOW} i T_{COM} . T_{HIGH} i T_{LOW} okidaju razinu (postaju jedinice) ukoliko temperatura postane viša od definirane T_{H} temperature ili niža od definirane T_{L} temperature. T_{COM} postaje jedinica u trenutku kada temperatura pređe T_{H} i ostaje u toj razini do trenutka spuštanja temperature ispod T_{L} [10].

Kao programsku podršku za komunikaciju sa digitalnim termometrom koristimo otvorenu Arduino datoteku "DS1620" koja se može nabaviti na [16]. Ova biblioteka, uz pravilne postavke, omogućava jednostavno očitavanje temperature pomoću samo jedne naredbe. No, zbog razlučivosti od 0.5°C , očitana temperatura je dvostruko veća od prave vrijednosti te je potrebno taj podatak podijeliti sa dva. Više o programskom kodu se može vidjeti u Dodatku.

4.3.2 Upravljanje grijačem

Kao grijač koristimo automobilsku žarulju nazivne snage 5 W i napona 12 V. Mikrokontroler preko jednog izlaznog priključka, ovisno o temperaturi koju očitava na termometru, pali i gasi žarulju preko dodatnog sklopovlja koji služi za prilagodbu signala i

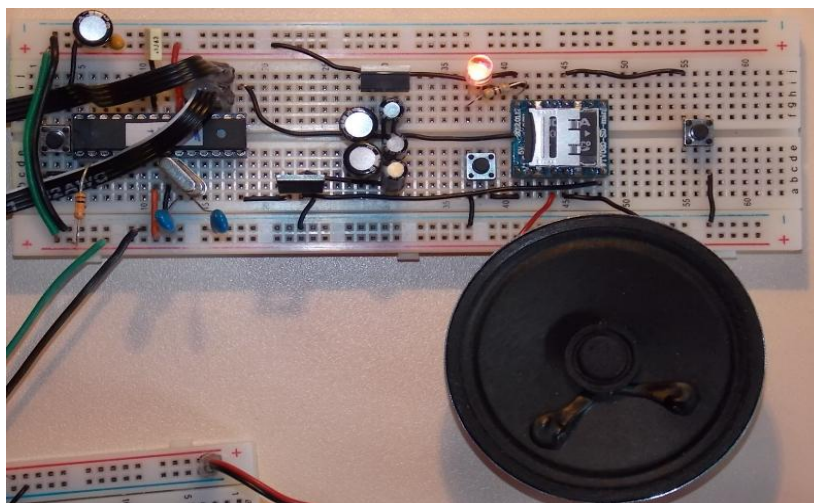
osiguravanje potrebne struje kroz žarulju (slika 25). Izlaz priključka kontrolera dovodimo na dvostruki inverter sa Schmittovim okidačem kako bismo ograničili istitravanja. Kao inverter koristimo sklop 74HC14 tvrtke NXP. Zatim signal dovodimo na tranzistor 2N2222 koji prilagođava razinu da bi završni tranzistor BD135 radio kao sklopka za uključivanje žarulje. Ostali otpornici služe za ograničavanje napona [5].



Slika 25: Shema sklopovlja za upravljanje žaruljom

4.4 Čvor za pokretanje muzike

Ukoliko mikrokontroler ovog čvora (ATmega168) očita sa sabirnice muzički podatkovni okvir, pokreće i upravlja sklopom za reprodukciju glazbe. Osnovni sklop koji se koristi za pohranu i reprodukciju glazbe jest muzički modul WTV020. Na slici 26 prikazan je muzički čvor našeg sustava pametne kuće.



Slika 26: Čvor za upravljanje muzikom

4.4.1 Muzički modul WTV020

Jednostavna reprodukcija muzike u naponskoj razini od 5 V moguća je sa modulom WTV020 koji na spojenom zvučniku reproducira pjesme pohranjene na *microSD* kartici (slika 27). S ovim modulom su kompatibilne SD kartice kapaciteta do 2 GB. Pjesme moraju biti jačinski prilagođene i pohranjene u ADPCM 4-bitnom formatu (tzv. AD4 format). Postupak prilagodbe i formatiranja opisan je u idućem poglavlju. Modul koristi 16-bitni AD pretvornik sa brzinama uzorkovanja od 6 kHz do 36 kHz za AD4 pjesme. Postoje 4 osnovna kontrolna načina rada modula i reproduciranja glazbe od kojih je, prema potrebi, moguće izabrati i kombinirati formu [13]:

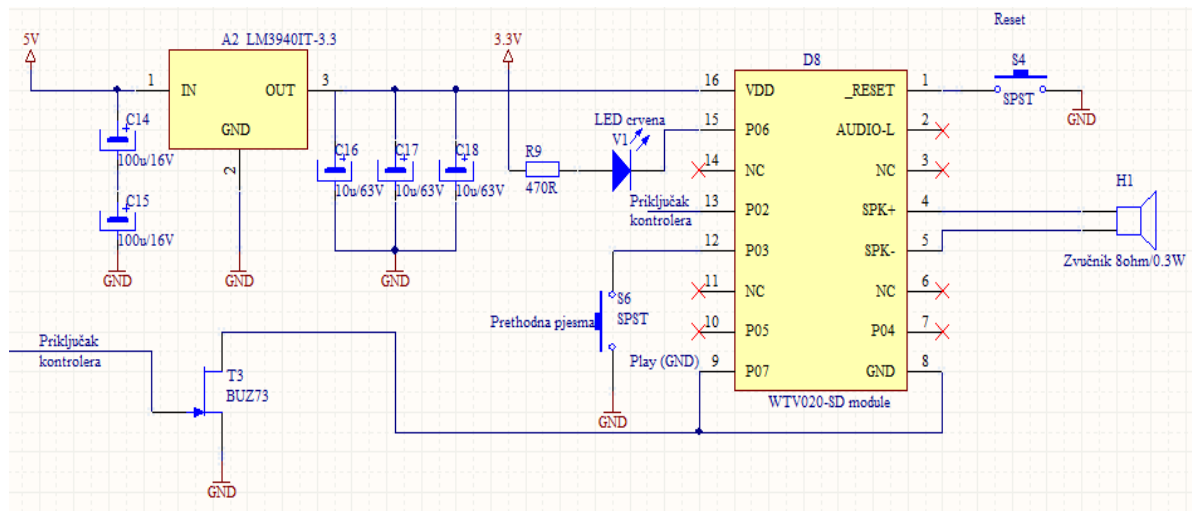
- MP3 Mode
- Key Mode
- Loop Mode
- Serial Mode



Slika 27: WTV020 modul

U našem sustavu koristimo dijelom Loop, a dijelom Key način rada što je prikazano slikom 28. Loop način rada podrazumijeva pokretanje muzike samim dovođenjem napajanja od 3.3 V na sklop, dok Key način rada pomoću tipki prebacuje pjesme unaprijed ili unazad. Za dobivanje napona od 3.3 V koristimo stabilizator napona sa 5 V. Taj napon dovodimo kao napajanje istovremeno tvoreći krug sa diodom koja signalizira pokretanje muzike. Priključak P02 spajanjem na masu prebacuje pjesmu unaprijed, a priključak P03 unazad. Kako uključivanjem napajanja reprodukcija kreće uvijek od prve pjesme na

kartici, moramo spojiti jedan od tih priključaka na mikrokontroler da bismo mogli namjestiti željenu pjesmu. Odlučili smo se za P02.



Slika 28: Kontrolni način rada WTV020 sklopa

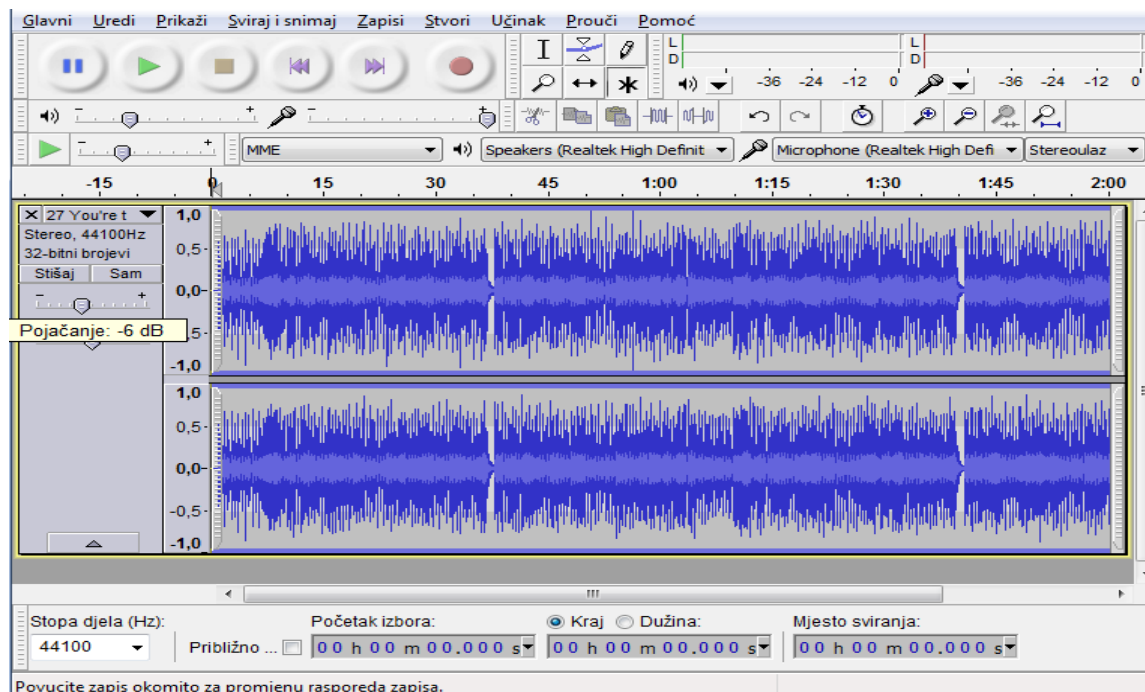
Priključak P07, ukoliko je spojen na masu, dopušta reprodukciju te ga stoga spajamo na GND priključak sklopa. Bitno je primjetiti kako ove priključke ne spajamo na masu već na kolektor tranzistora snage koji nam služi kao sklopka za uključivanje cijelog modula. Koristi se tranzistor snage BUZ73 te preko njegove baze priključak mikrokontrolera pokreće ili zaustavlja muziku. Postavlja se pitanje zašto se jednostavno ulazni napon stabilizatora ne spoji na priključak mikrokontrolera koji bi tako visokom razinom pokretao (odnosno niskom zaustavljao) muziku. Jednostavno zato jer priključak mikrokontrolera ne bi mogao dati potrebnu struju za pokretanje cijelog sklopovlja, već je tu struju potrebno dovoditi iz napajanja. Upravo zato koristimo tranzistor kao sklopku. U našem sustavu, 8-omski piezo zvučnik reproducira glazbu snagom do 3 W, no moguće je koristiti bilo koji komercijalni zvučnik sa pojačalom i trožičnim priključkom kod kojeg masu uklonimo, a diferencijalni par žica spojimo na modul.

4.4.2 Formatiranje i prilagodba pjesama

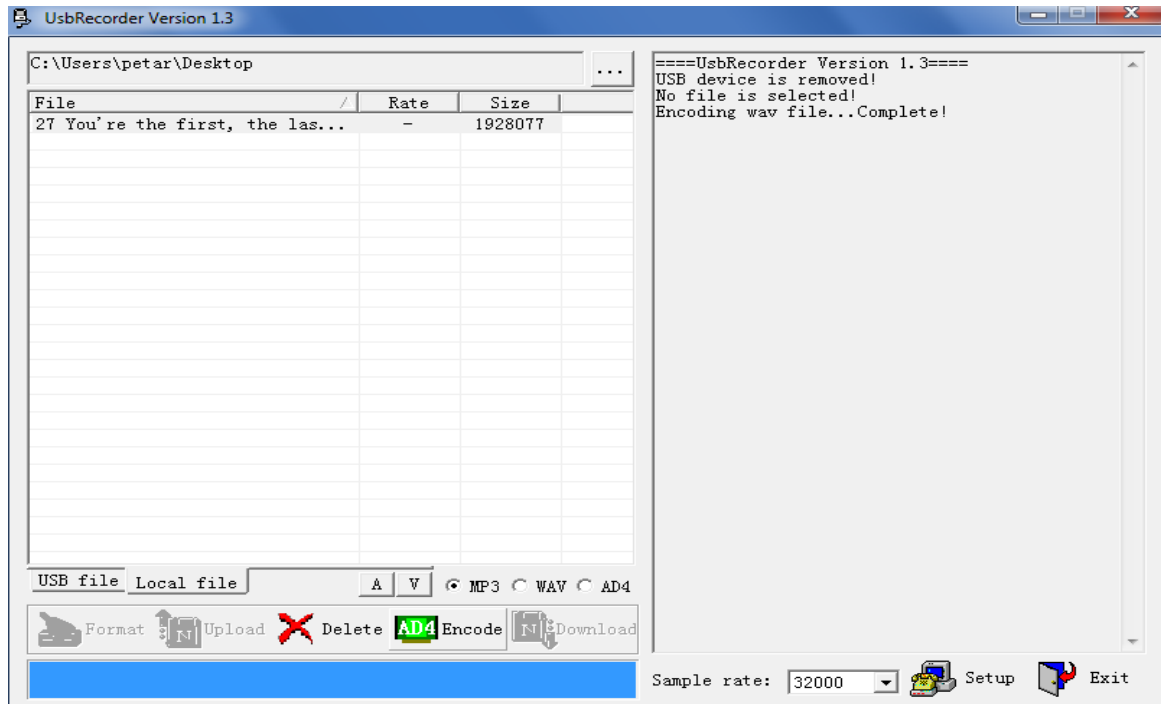
Kao što je navedeno, modul reproducira samo 4-bitne ADPCM pjesme. Pjesme takvog formata moguće je konvertirati iz MP3 i WAV formata u AD4 format pomoću programa *USBRecorder*. No prije toga potrebno je smanjiti jačinu (snagu) audio signala u pjesmama zbog ograničene struje sklopa i snage zvučnika. To vršimo pomoću programa za

uređivanje audio signala *Audacity*. Za primjer, pogledajmo kako pripremiti pjesmu za reprodukciju na WTV020 modulu.

Preko *Audacity* programa otvorimo pjesmu MP3 ili WAV formata koju želimo konvertirati u AD4 format. Otvara nam se vremenska reprezentacija audio signala (slika 29). Potrebno je smanjiti razinu signala (amplitudu – snagu) što činimo pomoću kliznika “Pojačanje“. Smanjimo razinu signala za 6 – 10 dB. Zatim pjesmu izvezemo (Glavni > Izvezi) i pohranimo na željeno mjesto. Kada je pjesma razinski prilagođena možemo pristupiti konvertiranju u ADPCM format. Pokrenemo *USBRecorder* program i odaberemo direktorij u kojem se nalazi prilagođena pjesma. Odaberemo format pjesme (MP3 ili WAV) te odaberemo brzinu uzorkovanja “*Sample rate*“. Obzirom da modul može podržati brzine od 6 kHz do 36 kHz možemo odabrati brzinu od 32 kHz. Pritiskom na “*AD4 Encode*“ opciju pokrećemo postupak konvertiranja pjesme u ADPCM format (slika 30). Time je postupak konvertiranja završen. Pjesma je spremna za pohranu na *microSD* karticu. Još jedna bitna stvar vezana za pjesmu je ta da ona mora biti jednostavno imenovana ukoliko se želi uspješno reproducirati. Stoga se preporuča da se pjesme nazovu brojevima onim redosljedom kojim se žele reproducirati (npr. “0001“, “0002“, ...).



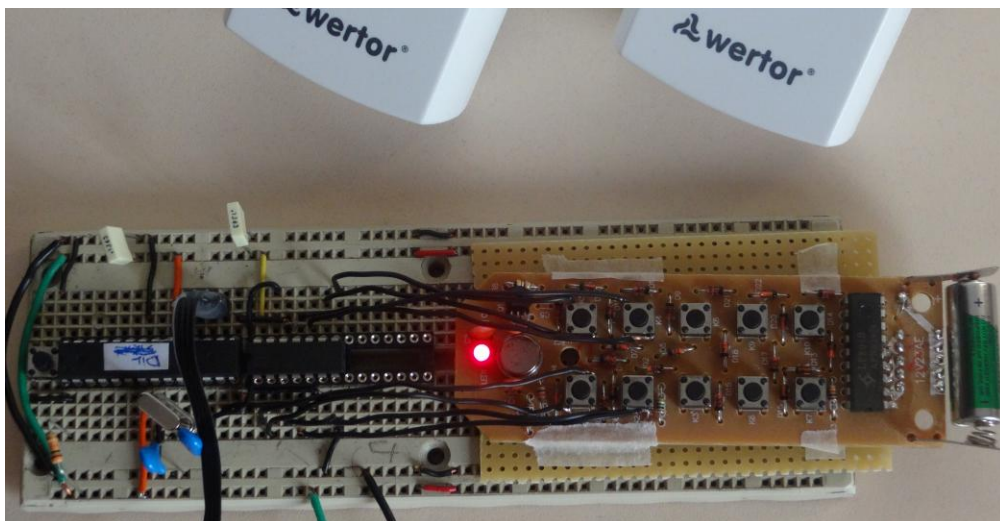
Slika 29: Audacity prilagodba pjesme



Slika 30: UsbRecorder- konvertiranje pjesme u AD4 format

4.5 Čvor za uključivanje tereta

Zadnji čvor u našem sustavu je čvor za uključivanje i isključivanje tereta na napon gradske mreže (slika 31). Gradska mreža izvor je napajanja za većinu kućanskih aparata i rasvjetu. Stoga je potrebno, ukoliko se želi postići sustav automatizacije koji, između ostalog, integrira aparate i rasvjetu, realizirati sofisticirani način uključivanja trošila na napajanje.



Slika 31: Čvor za uključivanje tereta

4.5.1 RSL sklopke

Danas su komercijalno dobavljive elektroničke sklopke koje, daljinski upravljanje, uključuju i isključuju teret na mrežu. Poznate su pod nazivom RSL sklopke (engl. *Remote Control Switch*) i izrađuju ih razni proizvođači (Conrad, Wertor, ...). To su utičnice koje se postavljaju u postojeće utičnice ili pak grla za žarulje koja se postavljaju u već postojeća grla (slika 32). Takve sklopke koristimo u našem projektu za upravljanje rasvjetom i radom digitalnog okvira za slike koji nam oponaša plazma televizor.



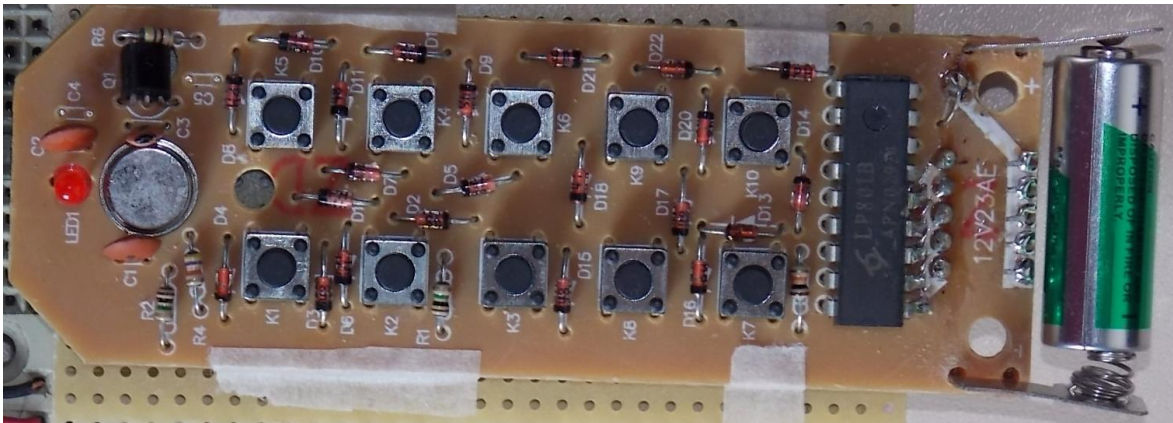
Slika 32: RSL sklopke i daljinski upravljač

Ovakve sklopke možemo razmatrati zasebnim čvorovima i u ovom kontekstu ih smatramo komercijalno dobavljivim čvorovima. One dolaze u paketu sa daljinskim upravljačem koji upravlja njihovim paljenjem i gašenjem preko radio veze. Koristi se frekvencijsko područje od 433 MHz, a svakom se kanalu daljinskog upravljača pridodijeljuje zasebna sklopka. Za upravljanje trošilima u našem automatizacijskom sustavu koristimo dvije RSL utičnice i daljinski upravljač.

RSL sklopke se koriste u konvencionalnim kućanstvima gdje korisnik, u svrhu jednostavnosti, daljinski želi paliti i gasiti rasvjetu i uređaje. Ovakav se koncept u modernim pametnim kućama ugrađuje u sustav gradske mreže pri samoj izgradnji objekta i integriran je u određeni funkcionalni čvor (ZigBee rf4ce specifikacija).

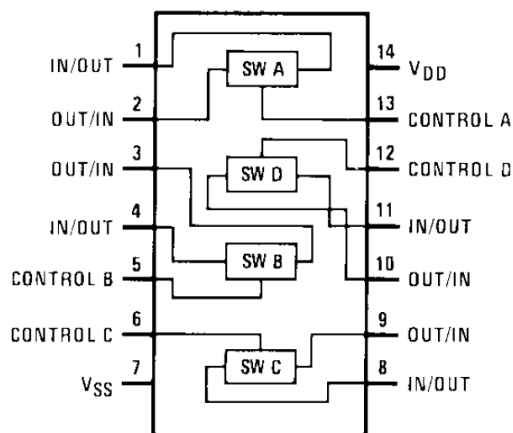
4.5.2 Daljinski modul i upravljanje trošilima

Glavna komponenta u čvoru, uz nadzorni AVR mikrokontroler, je i daljinski modul tj. tiskana pločica daljinskog upravljača (slika 33). Ona osigurava radiofrekvencijsku vezu sa RSL čvorovima (433 MHz) i cjelokupni komunikacijski protokol. Napaja se sa baterijom napona 12 V i sadrži 10 tipkala kojim se svaku od 5 RSL sklopku može uključiti i isključiti.



Slika 33: Daljinski modul

Obzirom da koristimo dvije RSL sklopke, koristit ćemo samo gornja dva kanala modula za koje smo predodredili da upravljaju našim sklopkama. Obzirom da s gornje četiri tipke palimo i gasimo sklopke, njih ćemo, umjesto ručno, pritiskati digitalno. To ćemo ostvariti pomoću sklopa HCF4066BE sa bilateralnim sklopkama (slika 34). Ovaj sklop omogućava kratko spajanje dva ulazno-izlazna priključka dovođenjem visokog napona na kontrolni priključak.



Slika 34: Konekcijski dijagram sklopa 4066

Stoga je koncept zapravo vrlo jednostavan: izvode tipkala koje smo dosad ručno kratko spajali dovodimo na ulazno-izlazne priključke sklopa sa bilateralnim sklopkama. Te izvode je sada moguće digitalno kratko spojiti ili odspojiti preko kontrolnih priključaka kojima upravljamo pomoću mikrokontrolera. Dakle, mikrokontroler u ovisnosti o okviru pročitanoj na sabirnici digitalno pali i gasi terete na gradskoj mreži.

4.6 Podatkovni okvir

Naš sustav pametne kuće koristi sabirničku vezu sa četiri linije preko koje glavni upravljački čvor šalje okvire. Dakle, prenose se 4-bitni podatkovni okviri koji označavaju: uređaj sa pripadajućom zadaćom, sinkronizacijski impuls ili inicijalizacijske zadaće. U tablici 4 prikazane su sve 4-bitne kombinacije podatkovnog okvira sa pripadajućim značenjem u komunikacijskom protokolu.

Tablica 4: Podatkovni okvir sustava pametne kuće

4-BITNI OKVIR	ZADAĆA
0000	Inicijalno stanje
0001	Kraj inicijalnog stanja
0010	X - ne koristi se
0011	Sinkronizacijski okvir
0100	Temperatura - povećaj za 1 °C
0101	Temperatura - smanji za 1 °C
0110	Temperatura - postavi na 25 °C
0111	Temperatura - postavi na 30 °C
1000	Muzika - ugasi
1001	Muzika - upali
1010	Muzika - prebaci pjesmu
1011	X - ne koristi se
1100	TV - ugasi
1101	TV - upali
1110	Rasvjeta - ugasi
1111	Rasvjeta - upali

Prva dva bita označavaju adresu čvora osim u slučaju dviju nula kada se radi o osnovnim zadaćama protokola: inicijalizaciji i sinkronizaciji. Kod okvira za upravljanje čvorovima preostala dva bita određuju zadaću prema tablici. U našem sustavu nema potvrde prijema

među čvorovima što u određenoj mjeri narušava pouzdanost. Određeni stupanj pouzdanosti jednostavno osiguravamo tako da upravljački čvor drži okvir na sabirnici dovoljno dugo da ga upravljani čvor sigurno stigne pročitati. Nakon svake zadaće slijedi sinkronizacijski impuls kako bi se označio kraj zadaće i kako bi se čvorovi pripremili za prijem nove zadaće. Inicijalizacijski se okvir šalje kod podizanja sustava i kod tzv. programskog reseta kojim korisnik ispočetka pokreće sustav pametne kuće.

4.7 Tehnički podaci

Napajanje:	5 V istosmjerno, stabilizirano, 1000 mA 12 V istosmjerno, stabilizirano, 1000 mA
Konektori:	X1 – napajanje 5 V (X1-1: GND, X1-2: +5 V) X2 – napajanje 12 V (X2-1: GND, X2-2: +12 V)
Tipkovnica:	Crvene tipke – odabir funkcije ili scenarija Zelene tipke – odabir zadaće za pojedine čvorove Crne tipke – odabir scenarija Žute tipke – povratak u izbornik i inicijalizacija
Temp. područje:	Komercijalno: 0°C - 70°C
Potrošnja:	0.70 W za neaktivni režim pametne kuće (osnovne funkcije) 4.50 W za All On Mode (svi čvorovi aktivni, 3.50 W troši žarulja)

U obračun potrošnje ne ubrajamo snagu trošila koja se priključuju na gradsku mrežu, već samo potrošnju elektroničkog sklopovlja u sustavu pametne kuće.

5. Zaključak

Razvoj kućne i stambene automatizacije uzrokovao je pojavu velikog broja komunikacijskih protokola. Vremenom su se mnogi od njih standardizirali u svrhu interoperabilnosti i jednostavnijeg razvoja. Otvoreni standardi dostupni su svima ili, u nekim slučajevima, samo članovima udruženja čiji član može postati svatko. Usprkos tome, mnoge korporacije do danas razvijaju svoje zatvorene protokole za tržišno natjecanje što u velikoj mjeri pridonosi skupoći *Smart Home* implementacije i uskosti kruga korisnika. S druge strane, veliki broj korporacija i udruženja, čija je zadaća razvoj ovakvih sustava, svjedoči o plodonosnoj budućnosti kućne automatizacije.

Realizacijom relativno jednostavnog primjera pametne kuće upoznali smo sam pojam i određene tehnološke i komunikacijske postupke koji se koriste u razvoju ovakvih sustava. Pravi *Smart Home* sustavi integriraju neusporedivo veći broj čvorova i podsustava te zahtijevaju sofisticiran sustav veze. To je zadaća koju obavljaju timovi inženjera i cijele korporacije te mnoga udruženja koja se brinu za razvoj i standardizaciju *Smart Home* sustava. To je kompleksan koncept koji ostavlja inženjerima brojne tehnološke zadaće čijim će rješenjima ovi sustavi postati cjenovno prihvatljivi, a ovaj koncept široko primijenjen.

Literatura

- [1] T. Sauter, D. Dietrich, W. Kastner i ostali, *EIB – Installation Bus System*, Munich Njemačka: Publicis KommunikationsAgentur GmbH GWA, 2001.
- [2] A. Kell, P. Colebrook, *Open Systems for Homes and Buildings: Comparing LonWorks and KNX*, Watford Ujedinjeno Kraljevstvo: I&I limited, 2004.
- [3] Echelon Corporation, *Introduction to the LonWorks® System*, Palo Alto SAD: Echelon Corporation – CA 94304 version 1.0, 1999.
- [4] J. Lönn, J. Olsson, *ZigBee for wireless networking*, ožujak 2005.
- [5] M. Vučić, *Ugradbeni računalni sustavi - Upute za laboratorijske vježbe: 2. Ciklus*, Fakultet elektrotehnike i računarstva u Zagrebu, 2007.
- [6] B. Modlic, S. Grgić, G. Šišul, *Elektroničke komunikacije – predavanja*, Fakultet elektrotehnike i računarstva u Zagrebu, 2013.
- [7] E. Prpić, *Automatizacija u zgradarstvu – predavanja*, Veleučilište u Rijeci
- [8] A. Ożadowicz, *Communication Reliability in the Intelligent Building Systems*, Institute of Electrical Drives Automation and Industrial Equipments, AGH – UST, Krakow Poljska
- [9] Atmel Corporation, *8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash*, San Jose SAD: Atmel Corporation – CA 95131, 2011.
- [10] Maxim Integrated, *DS1620 Digital Thermometer and Thermostat*, San Jose SAD: Maxim Integrated – CA 95134
- [11] J. Wang, S. Zhang, *Smart Home System - a wireless, voice-controllable, household system*, završni rad, Cornell University New York, 2012.
- [12] ZigBee Alliance, <http://www.zigbee.org/>
- [13] Elechouse, *WTV010 & WTV020 Datasheet - Elechouse*, <http://www.elechouse.com/elechouse/>
- [14] Wikipedia, *Digital Addressable Lighting Interface*, veljača 2013. godine, http://en.wikipedia.org/wiki/Digital_Addressable_Lighting_Interface,
- [15] Wikipedia, *X10 (industry design)*, travanj 2013. godine, http://en.wikipedia.org/wiki/X10_%28industry_standard%29

[16] GitHub, *Arduino – ds1620*, 2012. godina,

<https://github.com/msparks/arduino-ds1620/blob/master/examples/ds1620/ds1620.pde>

[17] Control4, <http://www.control4.com/>

[18] Crestron , <http://www.crestron.com/>

[19] AMX Corporation, <http://automate.amx.com/>

Sažetak

NORME I PROTOKOLI ZA AUTOMATIZACIJU KUĆANSTVA

Sažetak

Počevši od osnovnih pojmova i funkcionalnosti te pregledom normi i protokola upoznali smo moderni koncept pametne kuće, inteligentnog sustava upravljanja i veza. Primjenjujući elektronička i programerska znanja i jednostavnije načine povezivanja uređaja s aspekta kućne automatizacije realizirali smo zasebni sustav pametne kuće. On se sastoji od upravljačkog čvora i tri izvršna čvora od kojih jedan upravlja sa dvije komercijalno dobavljive daljinske sklopke. Sami čvorovi se temelje na AVR ATmega168 mikrokontrolerima. Sustav integrira reprodukciju muzike, regulaciju temperature te upravljanje rasvjetom i kućanskim aparatima u jedinstveni inteligentni sustav upravljanja.

Ključne riječi: pametna kuća, kućna automatizacija, inteligentni sustav upravljanja, čvor, protokol, standard, sabirnica, ATmega168 mikrokontroler

HOME AUTOMATION STANDARDS AND PROTOCOLS

Abstract

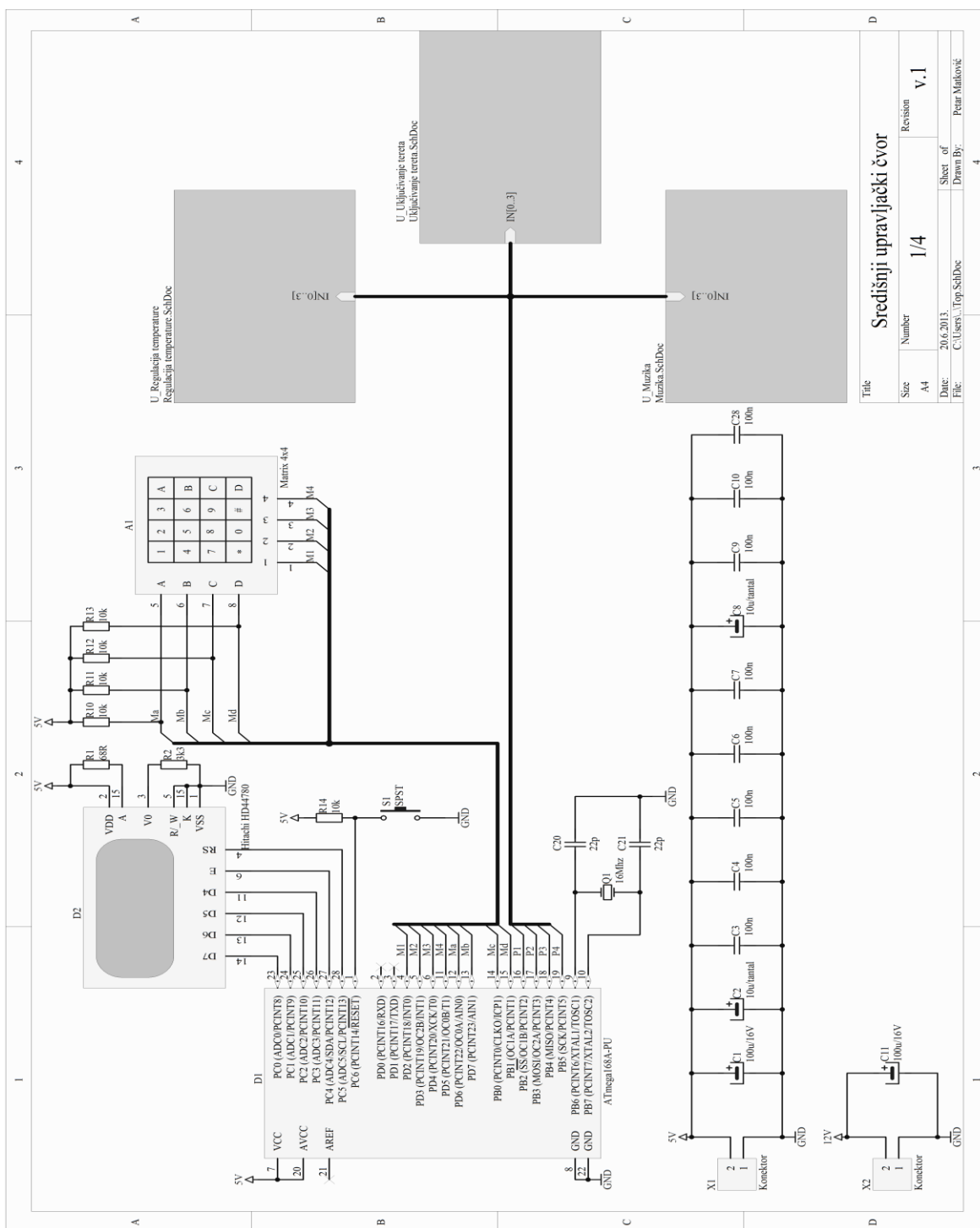
Beginning with basic terms and functionalities across standards and protocols overview we got acquainted with modern Smart Home concept, intelligent control and connection system. Using electronic and programming skills and simpler home automation ways of device connecting we accomplished separate Smart Home system. It contains one control node and three end nodes of which one manages two commercial remote controlled switches. The nodes are based on AVR ATmega168 microcontrollers. The System integrates music reproduction, temperature regulation along with lighting and appliances control in one singular intelligent control system.

Keywords: Smart Home, home automation, intelligent control system, node, protocol, standard, bus, ATmega168 microcontroller

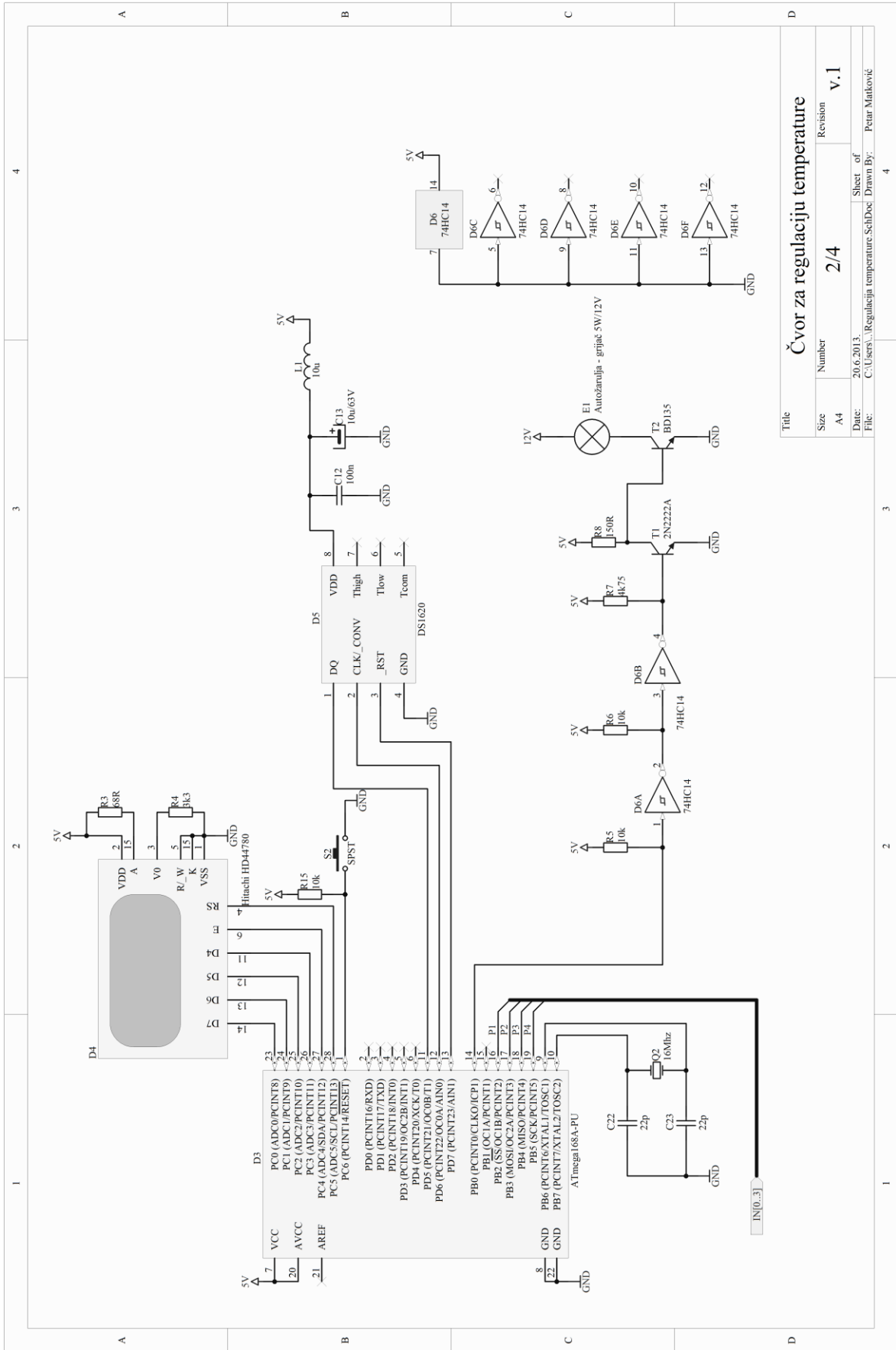
Dodatak

Obzirom da je naš sustav elektronički sustav potrebno ga je detaljno opisati na najbolji način. U tu svrhu prikazujemo električnu shemu i sastavnicu te cjelokupni programski kod.

Električna shema

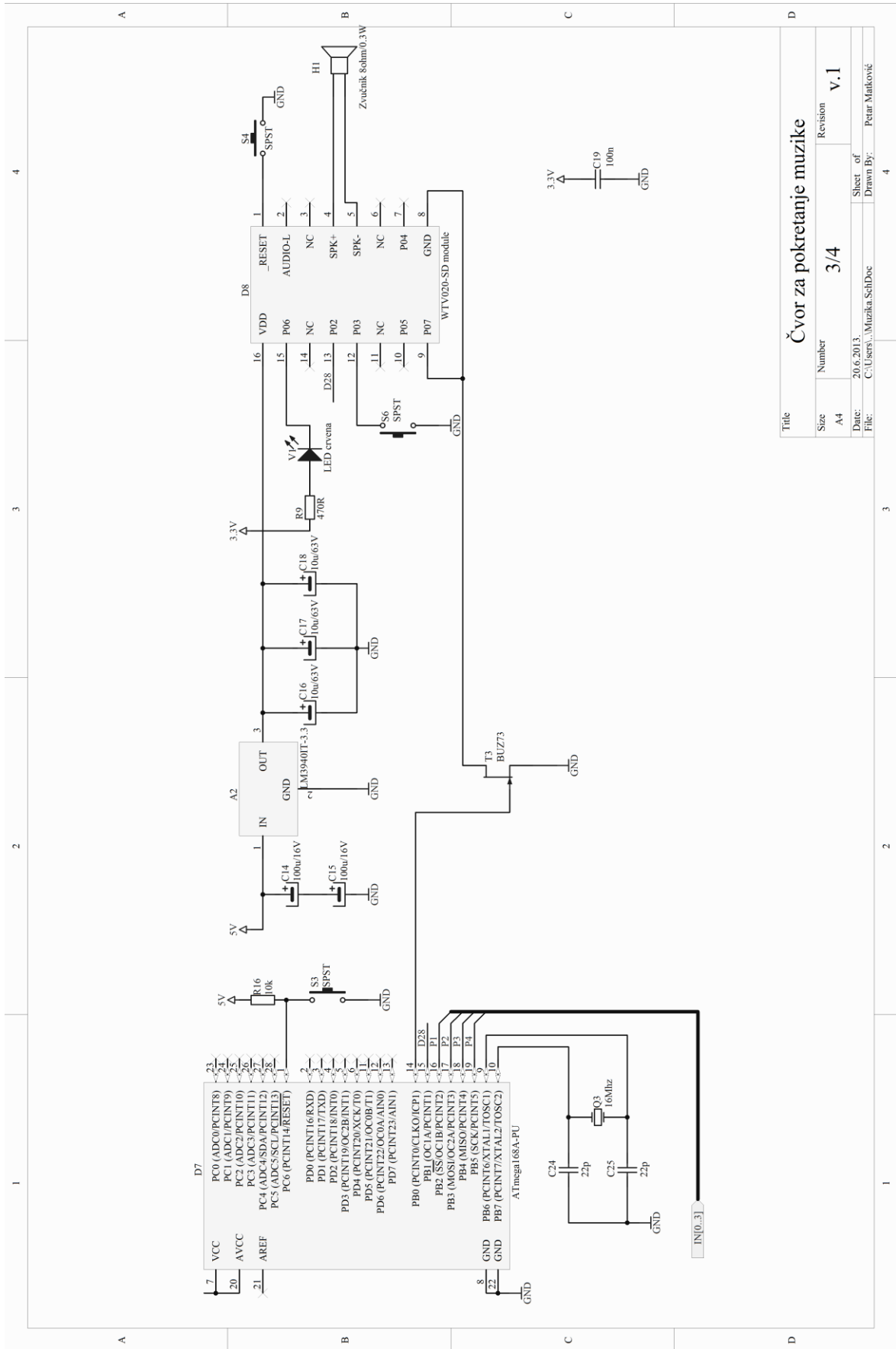


Slika 35: Električna shema sustava pametne kuće sa upravljačkim čvorom



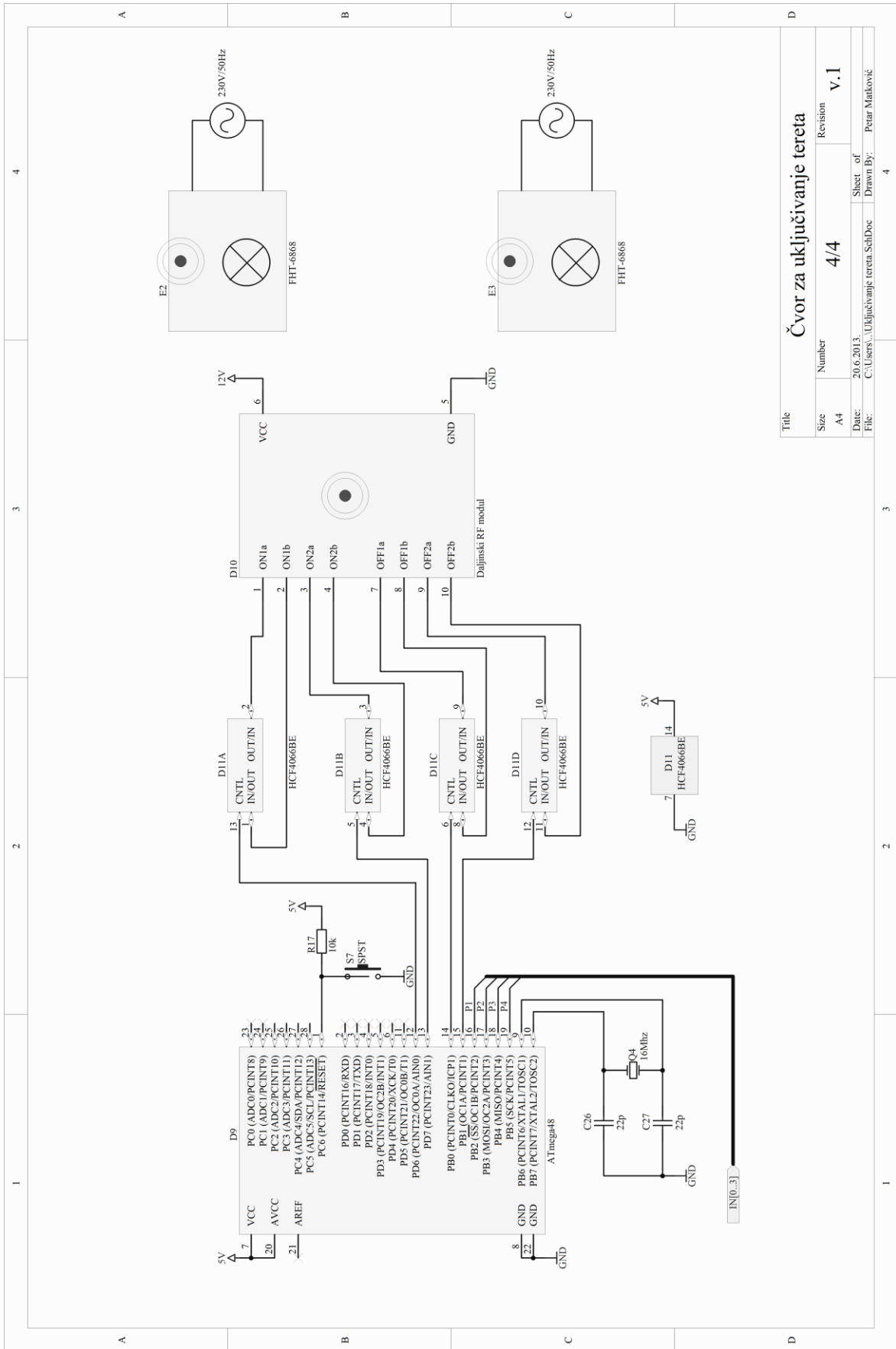
Title		Čvor za regulaciju temperature	
Size	A4	Number	2/4
Revision	v.1	Sheet of	4
Date:	20.6.2013.	SchDoc	Drawn By: Petar Manković
File:	C:\Users\... \Regulacija temperature.SchDoc		

Slika 36: Električna shema čvora za regulaciju temperature



Title		Čvor za pokretanje muzike	
Size	Number	Revision	
A4	3/4		V.1
Date:	20.6.2013.	Sheet of	
File:	C:\Users\...Muzika\SchDoc	Drawn By:	Petar Matković

Slika 37: Električna shema čvora za pokretanje muzike



Title		
Size	Number	Revision
A4	4/4	v.1
Date:	Sheet of	
File:	Drawn By:	
C:\Users\... \Uključivanje tereta.SchDoc	4	Petar Matković

Slika 38: Električna shema čvora za uključivanje tereta

Sastavnica

Redni broj	Designator	Comment	Cijena [Kn]	Dobavljač	Kataloški broj	Količina
1	D1, D3, D7	ATmega168A-PU, mikrokontroler	31.08	Farnell UK Altpro	1841605	3
2	D2, D4	Hlachi HD44780, LCD prikaznik	23.61	Ebay	HD44780	2
3	D5	DS1620, digitalni termometar	45.67	Farnell UK Altpro	1187998	1
4	D6	74HC14, digitalni inverter	2.50	Kronos Zagreb	X7052	1
5	D8	WTV020-SD, muzički modul	28.44	Ebay	WTV020-SD-16P	1
6	D9	ATmega48A-PU, mikrokontroler	28.80	Farnell UK Altpro	9171320	1
7	D10	Dajinski RF modul	142.00	Chipoteka	8071400092	1
8	D11	HCF4066BE, Bilateralna analogna sklopka	ožu.45	Farnell UK Altpro	1463136	1
9	A1	Matrlična tipkovnica 4x4	23.94	Ebay	16 4x4 Matrix	1
10	A2	LM3940IT-3.3, stabilizator napona 3.3V	17.55	Farnell UK Altpro	9494553	1
11	T1	2N2222A, BP tranzistor	5.04	Chipoteka	0730222201	1
12	T2	BD135, BP tranzistor	2.32	Chipoteka	0450013502	1
13	T3	BUZ 73, J-FET tranzistor	16.25	Kronos Zagreb	T6011	1
14	V1	LED crvena	1.61	Farnell UK Altpro	4912433	1
15	Q1, Q2, Q3, Q4	16 Mhz, kristal	3.00	Farnell UK Altpro	2057949	4
16	C1, C11, C14, C15	100u/16V, el. al. kondenzator	2.07	Farnell UK Altpro	2079092	4
17	C2, C8	10u, Tantal kondenzator	5.58	Farnell UK Altpro	1753986	2
18	C3, C4, C5, C6, C7, C9, C10, C12, C19, C28	100n, keramički blokadni kondenzator	1.23	Kronos Zagreb	C5019	9
19	C20, C21, C22, C23, C24, C25, C26, C27	22p, keramički kondenzator	0.61	Kronos Zagreb	C5031	8
20	C13, C16, C17, C18	10u/63V, el. al. kondenzator	0.61	Kronos Zagreb	C2031	4
21	R1, R3	68R, otpornik	0.13	Kronos Zagreb	R0070	2
22	R2, R4	3k3, otpornik	0.15	Kronos Zagreb	R0024	2
23	R5, R6, R10, R11, R12, R13, R14, R15, R16, R17	10k, otpornik	0.13	Kronos Zagreb	R0050	2
24	R7	4k75, otpornik	0.74	Kronos Zagreb	RA002	1
25	R8	150R, otpornik	0.13	Kronos Zagreb	R0004	1
26	R9	470R, otpornik	0.13	Kronos Zagreb	R0020	1
27	L1	10u, zavojnica	4.88	Kronos Zagreb	BL009	1
28	S1, S2, S3, S4, S5, S6, S7	SPST, tipkalo	3.15	Farnell UK Altpro	1217775	7
29	E1	Autožarulja - grijač5W/12V	7.50	Chipoteka	8020803808	1
30	E2, E3	FHT-6868 bežični prekidač tereta	U cijeni D10	Chipoteka	8071400092	2
31	H1	Zvučnik 8ohm/0.3W	12.41	Chipoteka	2631000104	1
32	X1, X2	Konektor napajanja	8.60	Farnell UK Altpro	1217037	2

Programski kod

Program je pisan Arduino 1.0.4 programskim alatom za AVR mikrokontrolere.

Kontrolni čvor

```
#include <LiquidCrystal.h> // LCD biblioteka
LiquidCrystal lcd(A5,A4,A3,A2,A1,A0); //LCD - pinovi: RS ,E ,D4 ,D5 ,D6 ,D7

void setup() {
  digitalWrite(13,0); //Postavljamo riječ '0000' na sabirnicu koja označava
  digitalWrite(12,0); //inicijalno stanje.
  digitalWrite(11,0);
  digitalWrite(10,0);
  lcd.clear(); //Očisti ekran
  lcd.begin(16,2); //LCD - broj stupaca i redaka
  lcd.setCursor(0,0); //Postavi kursor na pocetak prvog reda
  lcd.print("*** Smart Home ***");
  lcd.setCursor(0,1); //Postavi kursor na pocetak drugog reda
  lcd.print("*** Dobrodosli ***");
  delay(700);
  digitalWrite(13,0); //Postavljamo riječ '0011' na sabirnicu koja označava
  digitalWrite(12,0); //Kraj inicijalnog stanja kako se inicijalno stanje nebi bespotrebno očitavalo.
  digitalWrite(11,1); //Bitno kod čvora za trošila
  digitalWrite(10,1);
  delay(1300); //Ukupno uvodno vrijeme neka iznosi 700+1300=2000ms

  pinMode(2,OUTPUT); //Tipkovnica: izlazni pinovi 1,2,3,4
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,INPUT); //Tipkovnica: ulazni pinovi A,B,C,D
  pinMode(7,INPUT);
  pinMode(8,INPUT);
  pinMode(9,INPUT);

  //Pinovi koji upravljaju sabirnicom
  pinMode(13,OUTPUT); //Izlaz 19 iz 168ice - MSB
  pinMode(12,OUTPUT); //Izlaz 18 iz 168ice
  pinMode(11,OUTPUT); //Izlaz 17 iz 168ice
  pinMode(10,OUTPUT); //Izlaz 16 iz 168ice - LSB
}

int tipka=-1; //Trenutno pritisnuta tipka
char s[16]=""; //String koji se unosi tipkovnicom
int Temp=22; //Inicijalna temperatura je 22 stupnjeva C

void CitanjeTipke(){ //****čitanje s tipkovnice + uklanjanje debouncinga****//
  digitalWrite(2,0); //Postavi samo 1. stupac u nulu, ostale u jedinicu.
  digitalWrite(3,1);
  digitalWrite(4,1);
  digitalWrite(5,1);
  if(digitalRead(6)==0){ //Da li je stisnuta "1"?
    delay(150); //sacekaj 150ms pa opet pročitaj pin.
    if(digitalRead(6)==0){ //Ako je i sada stisnuta ista tipka onda
      tipka=1; //je to prava tipka i dodaj ju u string.
      strcat(s,"1");
    }
  }
  if(digitalRead(7)==0){ //Da li je stisnuta "4"?
    delay(150); //sacekaj 150ms pa opet pročitaj pin.
    if(digitalRead(7)==0){ //Ako je i sada stisnuta ista tipka onda
      tipka=2; //je to prava tipka i dodaj ju u string.
      strcat(s,"4");
    }
  }
  if(digitalRead(8)==0){ //Da li je stisnut znak za inicijalizaciju?
    delay(150); //sacekaj 150ms pa opet pročitaj pin.
    if(digitalRead(8)==0){ //Ako je i sada stisnuta ista tipka onda
      tipka=3; //je to prava tipka.
    }
  }
  if(digitalRead(9)==0){ //Da li je stisnuto znak za Povratak?
    delay(150); //sacekaj 150ms pa opet pročitaj pin.
    if(digitalRead(9)==0){ //Ako je i sada stisnuta ista tipka onda
      tipka=4; //je to prava tipka.
    }
  }
}

digitalWrite(2,1); //Postavi samo 2. stupac u nulu, ostale u jedinicu.
```

```

digitalWrite(3,0);
digitalWrite(4,1);
digitalWrite(5,1);
if(digitalRead(6)==0){
    delay(150);
    if(digitalRead(6)==0){
        tipka=5;
        strcat(s,"2");
    }
}
if(digitalRead(7)==0){
    delay(150);
    if(digitalRead(7)==0){
        tipka=6;
        strcat(s,"5");
    }
}
if(digitalRead(8)==0){
    delay(150);
    if(digitalRead(8)==0){
        tipka=7;
        strcat(s,"ON");
    }
}
if(digitalRead(9)==0){
    delay(150);
    if(digitalRead(9)==0){
        tipka=8;
        strcat(s,"+");
    }
}
digitalWrite(2,1);
digitalWrite(3,1);
digitalWrite(4,0);
digitalWrite(5,1);
if(digitalRead(6)==0){
    delay(150);
    if(digitalRead(6)==0){
        tipka=9;
        strcat(s,"3");
    }
}
if(digitalRead(7)==0){
    delay(150);
    if(digitalRead(7)==0){
        tipka=10;
        strcat(s,"6");
    }
}
if(digitalRead(8)==0){
    delay(150);
    if(digitalRead(8)==0){
        tipka=11;
        strcat(s,"OFF");
    }
}
if(digitalRead(9)==0){
    delay(150);
    if(digitalRead(9)==0){
        tipka=12;
        strcat(s,"-");
    }
}

digitalWrite(2,1);
digitalWrite(3,1);
digitalWrite(4,1);
digitalWrite(5,0);
if(digitalRead(6)==0){
    delay(150);
    if(digitalRead(6)==0){
        tipka=13;
        strcat(s,"TEMPERATURA");
    }
}
if(digitalRead(7)==0){
    delay(150);
    if(digitalRead(7)==0){
        tipka=14;
        strcat(s,"MUZIKA");
    }
}
if(digitalRead(8)==0){
    delay(150);
}

```

//Da li je stisnuta "2"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnuta "5"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnut "ON"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnut znak "+"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string

//Postavi samo 3. stupac u nulu, ostale u jedinicu.

//Da li je stisnuta "3"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnuta "6"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnut znak "OFF"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnuto znak "-"?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Postavi samo 4. stupac u nulu, ostale u jedinicu.

//Da li je stisnut znak za postavljanje temperature?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnut znak za muziku?

//sacekaj 150ms pa opet procitaj pin.

//Ako je i sada stisnuta ista tipka onda

//je to prava tipka i dodaj ju u string.

//Da li je stisnut znak za uključivanje tereta?

//sacekaj 150ms pa opet procitaj pin.

```

    if(digitalRead(8)==0){
        tipka=15;
        strcat(s,"TROSILO");
    }
}
if(digitalRead(9)==0){
    delay(150);
    if(digitalRead(9)==0){
        tipka=16;
        strcat(s,"SCENARIJ");
    }
}
}

void OdabirZadace(){
    tipka=-1;
    int a=1;
    lcd.setCursor(0,0);
    lcd.print("Odaberite zadacu");
    lcd.setCursor(0,1);
    lcd.print("TEM-MUZ-TRO-SCEN");
    while(a){
        s[0]='\0';
        CitanjeTipke();
        if((tipka==13)||((tipka==14)||((tipka==15)||((tipka==16)))){
            a=0;
            if(tipka==3){
                Inicijalizacija();
                a=0;
            }
        }
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Odaberite zadacu");
        lcd.setCursor(0,1);
        lcd.print("---->");
        lcd.setCursor(5,1);
        lcd.print(s);
        delay(1250);
        PozivZadace(tipka);
    }
}

void PozivZadace(int x){
    int a,b,i;
    switch(x){
        case(13):
            a=1;
            tipka=-1;
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("TEMPERATURA: C ");
            lcd.setCursor(12,0);
            lcd.print(Temp);
            lcd.setCursor(15,0);
            lcd.print("C");
            lcd.setCursor(0,1);
            lcd.print("+Povecaj/-Smanji");
            while(a){
                s[0]='\0';
                CitanjeTipke();
                if(tipka==8){
                    lcd.clear();
                    lcd.setCursor(0,0);
                    lcd.print("TEMPERATURA");
                    lcd.setCursor(0,1);
                    lcd.print("---->Povecaj 1°C");
                    Temp++;
                    digitalWrite(13,0);
                    digitalWrite(12,1);
                    digitalWrite(11,0);
                    digitalWrite(10,1);
                    delay(650);
                    digitalWrite(13,0);
                    digitalWrite(12,0);
                    digitalWrite(11,0);
                    digitalWrite(10,1);
                    delay(10);
                    a=0;
                }
                if(tipka==12){
                    lcd.clear();
                    lcd.setCursor(0,0);
                    lcd.print("TEMPERATURA");
                    lcd.setCursor(0,1);
                }
            }
        }
}

```

```

lcd.print("---->Smanji 1*C");
Temp--; //Zapamti smanjenje temp. i u ovom čvoru
digitalWrite(13,0); //0100 -> smanji temperaturu za 1 C stupanj
digitalWrite(12,1);
digitalWrite(11,0);
digitalWrite(10,0);
delay(650);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
a=0; //Tipka pritisnuta, vrati se na odabir zadaće nakon odsječka
}
if(tipka==4)return; //Povratak u meni
if(tipka==3){ //Ako je tipka za inicijalizaciju
  Inicijalizacija(); //Inicijalizacija sustava
  a=0; //Tipka pritisnuta, vrati se na odabir zadaće
}
}
break;

case(14): //** Reprodukcija Muzike **//
a=1; //Analogno prvom slučaju
tipka=-1;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("MUZIKA:");
lcd.setCursor(0,1);
lcd.print("ON:Uklj/OFF:Iskl");
while(a){
  s[0]='\0';
  CitanjeTipke();
  if(tipka==7){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MUZIKA:");
    lcd.setCursor(0,1);
    lcd.print("->Pokreni muziku");
    digitalWrite(13,1); //1001 -> pokreni muziku
    digitalWrite(12,0);
    digitalWrite(11,0);
    digitalWrite(10,1);
    delay(400);
    digitalWrite(13,0); //0001 -> sinkronizacijska riječ
    digitalWrite(12,0);
    digitalWrite(11,0);
    digitalWrite(10,1);
    delay(10);
    a=0;
  }
  if(tipka==11){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MUZIKA:");
    lcd.setCursor(0,1);
    lcd.print("->Ugasi muziku");
    digitalWrite(13,1); //1000 -> isključi muziku
    digitalWrite(12,0);
    digitalWrite(11,0);
    digitalWrite(10,0);
    delay(400);
    digitalWrite(13,0); //0001 -> sinkronizacijska riječ
    digitalWrite(12,0);
    digitalWrite(11,0);
    digitalWrite(10,1);
    delay(10);
    a=0;
  }
}
if(tipka==4)return; //Povratak u meni
if(tipka==3){ //Inicijalizacija sustava
  Inicijalizacija();
  a=0;
}
}
break;

case(15): //** Upravljanje trošilima i rasvjetom **//
a=1;
tipka=-1;
lcd.clear();
lcd.setCursor(0,0); //Odabir 1-aparat; 2-rasvjeta
lcd.print("TROSILO:");
lcd.setCursor(0,1);

```

```

lcd.print("1:Aparat 2:Rasvj");
while(a){
s[0]='\0';
CitanjeTipke();
if(tipka==1){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("KUCANSKI APARAT");
lcd.setCursor(0,1);
lcd.print("ON:Uklj/OFF:Iskl");
b=1;
while(b){
s[0]='\0';
CitanjeTipke();
if(tipka==7){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("KUCANSKI APARAT:");
lcd.setCursor(0,1);
lcd.print("->Ukljuci aparat");
digitalWrite(13,1);
digitalWrite(12,1);
digitalWrite(11,0);
digitalWrite(10,1);
delay(600);
digitalWrite(13,0);
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
a=0;b=0;
}
if(tipka==11){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("KUCANSKI APARAT:");
lcd.setCursor(0,1);
lcd.print("->Isklj. aparat");
digitalWrite(13,1);
digitalWrite(12,1);
digitalWrite(11,0);
digitalWrite(10,0);
delay(600);
digitalWrite(13,0);
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
a=0;b=0;
}
if(tipka==4)return;
if(tipka==3){
Inicijalizacija();
a=0;b=0;
}
}
if(tipka==5){
b=1;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("RASVJETA:");
lcd.setCursor(0,1);
lcd.print("ON:Uklj/OFF:Iskl");
while(b){
s[0]='\0';
CitanjeTipke();
if(tipka==7){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("RASVJETA:");
lcd.setCursor(0,1);
lcd.print("->Uklj. rasvjetu");
digitalWrite(13,1);
digitalWrite(12,1);
digitalWrite(11,1);
digitalWrite(10,1);
delay(600);
digitalWrite(13,0);
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
a=0;b=0;
}
}
}

```

//Prva petlja za odabir trošila

//Odabran aparat - TV plazma
//Odabir ON-uključiti; OFF-isključiti

//Petlja za odabir: uključiti/isključiti

//Ako je stisnut ON uključiti aparat

//1101 -> ukljuci kućanski aparat

//0001 -> sinkronizacijska riječ

//Izlazak iz obiju petlji

//Ako je stisnut OFF isključiti aparat

//1100 -> isključiti kućanski aparat

//0001 -> sinkronizacijska riječ

//Izađi iz obiju petlji

//Povratak u meni

//Inicijalizacija sustava
//Izađi iz obiju petlji

//Ako je odabrana rasvjeta učini analogno kao za aparat

//1111 -> ukljuci rasvjetu

//0001 -> sinkronizacijska riječ


```

}
if(tipka==11){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("KUCANSKI APARAT:");
  lcd.setCursor(0,1);
  lcd.print("->lsk. rasvjetu");
  digitalWrite(13,1); //1110 -> isključi rasvjetu
  digitalWrite(12,1);
  digitalWrite(11,1);
  digitalWrite(10,0);
  delay(600);
  digitalWrite(13,0); //0001 -> sinkronizacijska riječ
  digitalWrite(12,0);
  digitalWrite(11,0);
  digitalWrite(10,1);
  delay(10);
  a=0;b=0; //Izađi iz objiju petlja
}
if(tipka==4)return; //Povratak u meni
if(tipka==3){
  Inicijalizacija(); //Inicijalizacija sustava
  a=0;b=0; //Izađi iz objiju petlja
}
}
if(tipka==4)return; //Povratak u meni
if(tipka==3){
  Inicijalizacija(); //Inicijalizacija sustava
  a=0; //Izađi iz petlje za odabir trošila
}
}
break;

case(16): //** Odabir scenarija **//
  a=1; //Variable za čitanje stanja i uvjet petlje
  tipka=-1;
  lcd.clear(); //Prikaži ponudu scenarija
  lcd.setCursor(0,0);
  lcd.print("1:Sleep 2:Read ");
  lcd.setCursor(0,1);
  lcd.print("3:Fun 4:News 5:#");
  while(a){ //Petlja za čitanje tipke
    s[0]='\0'; //Očisti string
    CitanjeTipke();
    if(tipka==1){ //Odabrab scenarij spavanja
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("SCENARIJ:");
      lcd.setCursor(0,1);
      lcd.print("---->Sleep Mode");
      digitalWrite(13,1); //1110 -> ugasi plazmu
      digitalWrite(12,1);
      digitalWrite(11,0);
      digitalWrite(10,0);
      delay(600);
      digitalWrite(13,0); //0001 -> sinkronizacijska riječ
      digitalWrite(12,0);
      digitalWrite(11,0);
      digitalWrite(10,1);
      delay(10);
      digitalWrite(13,1); //1110 -> ugasi svjetlo
      digitalWrite(12,1);
      digitalWrite(11,1);
      digitalWrite(10,0);
      delay(600);
      digitalWrite(13,0); //0001 -> sinkronizacijska riječ
      digitalWrite(12,0);
      digitalWrite(11,0);
      digitalWrite(10,1);
      delay(10);
      Temp=25; //Zapamti temperaturu u ovom čvoru
      digitalWrite(13,0); //0111 -> postavi temperaturu na 25 C stupnjeva
      digitalWrite(12,1);
      digitalWrite(11,1);
      digitalWrite(10,0);
      delay(650);
      digitalWrite(13,0); //0001 -> sinkronizacijska riječ
      digitalWrite(12,0);
      digitalWrite(11,0);
      digitalWrite(10,1);
      delay(10);
      digitalWrite(13,1); //1001 -> pokreni muziku
      digitalWrite(12,0);

```

```

digitalWrite(11,0);
digitalWrite(10,1);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(1500); //Sačekaj da se podigne muzika
for(i=0;i<4;i++){ //Pjesma za spavanje je na 5. mjestu na modulu - potrebna 4 prebacivanja
digitalWrite(13,1); //1010 -> zadaj sljedeći pjesmu
digitalWrite(12,0);
digitalWrite(11,1);
digitalWrite(10,0);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
}
a=0; //Izadi iz petlje na kraju odsječka
tipka=-1; //Zadrži naslov scenarija sve dok se ne pritisne bilo koja tipka
while(tipka===-1)
CitanjeTipke();
}
if(tipka==5){ //Odabir scenarija čitanja
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SCENARIJ:");
lcd.setCursor(0,1);
lcd.print("---->Read Mode");
Temp=25; //Zapamti temperaturu i u ovom čvoru
digitalWrite(13,0); //0110 -> postavi temperaturu na 25 C stupnja
digitalWrite(12,1);
digitalWrite(11,1);
digitalWrite(10,0);
delay(650);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
digitalWrite(13,1); //1111 -> upali svjetlo
digitalWrite(12,1);
digitalWrite(11,1);
digitalWrite(10,1);
delay(600);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
digitalWrite(13,1); //1001 -> pokreni muziku
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(1500); //Sačekaj da se podigne muzika
for(i=0;i<5;i++){ //Klasika je na 6. mjestu na modulu
digitalWrite(13,1); //1010 -> zadaj sljedeći pjesmu
digitalWrite(12,0);
digitalWrite(11,1);
digitalWrite(10,0);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
}
a=0; //Izadi iz petlje nakon odsječka
tipka=-1; //Zadrži naslov scenarija sve dok se ne pritisne bilo koja tipka
while(tipka===-1)
CitanjeTipke();
}
if(tipka==9){ //Odabran scenarij zabave
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SCENARIJ:");
lcd.setCursor(0,1);

```

```

lcd.print("---->Fun Mode");
digitalWrite(13,1); //1101 -> uključi "plasmu" (kućanski aparat)
digitalWrite(12,1);
digitalWrite(11,0);
digitalWrite(10,1);
delay(600);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
Temp=25; //Zapamti temperaturu i u ovom čvoru
digitalWrite(13,0); //0110 -> postavi temperaturu na 25 C stupnja
digitalWrite(12,1);
digitalWrite(11,1);
digitalWrite(10,0);
delay(650);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
digitalWrite(13,1); //0101 -> uključi muziku
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
a=0; //Izađi iz petlje na kraju odsječka
tipka=-1; //Zadrži naslov scenarija sve dok se ne pritisne bilo koja tipka
while(tipka===-1)
  CitanjeTipke();
}
if(tipka==2){ //Odabir scenarija za vijesti
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SCENARIJ:");
  lcd.setCursor(0,1);
  lcd.print("---->News Mode");
  digitalWrite(13,1); //1001 -> pokreni muziku
  digitalWrite(12,0);
  digitalWrite(11,0);
  digitalWrite(10,1);
  delay(200);
  digitalWrite(13,0); //0001 -> sinkronizacijska riječ
  digitalWrite(12,0);
  digitalWrite(11,0);
  digitalWrite(10,1);
  delay(1500); //Čekaj da se podigne muzika
  for(i=0;i<3;i++){ //Tonski zapis vijesti je 4. "pjesma" na modulu
    digitalWrite(13,1); //1010 -> zadaj sljedeću pjesmu (tonski zapis vijesti)
    digitalWrite(12,0);
    digitalWrite(11,1);
    digitalWrite(10,0);
    delay(200);
    digitalWrite(13,0); //0001 -> sinkronizacijska riječ
    digitalWrite(12,0);
    digitalWrite(11,0);
    digitalWrite(10,1);
    delay(10);
  }
  a=0; //Izađi iz petlje na kraju odsječka
  tipka=-1; //Zadrži naslov scenarija sve dok se ne pritisne bilo koja tipka
  while(tipka===-1)
    CitanjeTipke();
}
if(tipka==6){ //Odabran scenarij "Sve uključeno"
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SCENARIJ:");
  lcd.setCursor(0,1);
  lcd.print("---->All On Mode");
  Temp=30; //Zapamti temperaturu i u ovom čvoru
  digitalWrite(13,0); //0111 -> postavi temperaturu na 30 C stupnja
  digitalWrite(12,1);
  digitalWrite(11,1);
  digitalWrite(10,1);
  delay(650);
  digitalWrite(13,0); //0001 -> sinkronizacijska riječ
  digitalWrite(12,0);

```

```

digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
digitalWrite(13,1); //1110 -> uključi "plazmu" (kućanski aparat)
digitalWrite(12,1);
digitalWrite(11,0);
digitalWrite(10,1);
delay(600);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
digitalWrite(13,1); //1111 -> uključi rasvjetu
digitalWrite(12,1);
digitalWrite(11,1);
digitalWrite(10,1);
delay(600);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
digitalWrite(13,1); //1001 -> pokreni muziku
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(1500);
digitalWrite(13,1); //1010 -> postavi drugu pjesmu
digitalWrite(12,0);
digitalWrite(11,1);
digitalWrite(10,0);
delay(200);
digitalWrite(13,0); //0001 -> sinkronizacijska riječ
digitalWrite(12,0);
digitalWrite(11,0);
digitalWrite(10,1);
delay(10);
a=0; //Izadi iz petlje na kraju odsječka
tipka=-1; //Zadrži naslov scenarija sve dok se ne pritisne bilo koja tipka
while(tipka==1)
  CitanjeTipke();
}
if(tipka==4)return; //Povratak u meni
if(tipka==3){
  Inicijalizacija(); //Inicijalizacija sustava
  a=0;
}
}
break;
}
}

void Inicijalizacija(){ //**** Funkcija za inicijalizaciju i ponovno pokretanje sustava (restart) ****//
digitalWrite(13,0); //Postavljamo riječ '0000' na sabirnicu koja označava
digitalWrite(12,0); //inicijalno stanje.
digitalWrite(11,0);
digitalWrite(10,0);
lcd.clear();
lcd.begin(16,2); //LCD - broj stupaca i redaka
lcd.setCursor(0,0); //Postavi kursor na pocetak prvog reda
lcd.print("*** Smart Home ***");
lcd.setCursor(0,1); //Postavi kursor na pocetak drugog reda
lcd.print("*** Dobrodosli ***");
delay(700);
digitalWrite(13,0); //Postavljamo riječ '0011' na sabirnicu koja označava
digitalWrite(12,0); //Kraj inicijalnog stanja kako se inicijalno stanje nebi bespotrebno očitavalo.
digitalWrite(11,1); //Bitno kod čvora za trošila
digitalWrite(10,1);
delay(1300);
tipka=-1; //Trenutno pritisnuta tipka
s[0]='\0';
Temp=22; //Inicijalna temperatura je 22 stupnjeva C
return;
}

void loop(){ //Glavna petlja programa
  OdabirZadace(); //Odaberi zadaću
}

```

Čvor za regulaciju temperature

```
#include <LiquidCrystal.h>
#include <DS1620.h> //**** Open source biblioteka DS1620.h ****//
#include <stdlib.h>

LiquidCrystal lcd(A5,A4,A3,A2,A1,A0); //LCD - pinovi: RS ,E ,D4 ,D5 ,D6 ,D7
//Pinovi za spajanje sa termometrom DS1620
static const uint8_t RST_PIN = 7; //Pin 13 iz 168ice - spojen na RST_ termometra
static const uint8_t CLK_PIN = 6; //Pin 12 iz 168ice - spojen na CLK termometra
static const uint8_t DQ_PIN = 5; //Pin 11 iz 168ice - spojen na DQ termometra
DS1620 ds1620(RST_PIN, CLK_PIN, DQ_PIN);

void setup() {
  lcd.clear();
  lcd.begin(16,2); //LCD - broj stupaca i redaka
  lcd.setCursor(0,0); //Postavi kursor na pocetak prvog reda
  lcd.print(" Regulacija ");
  lcd.setCursor(0,1); //Postavi kursor na pocetak drugog reda
  lcd.print(" grijanja ");
  delay(2000);

  //Pinovi za pristup sabirnici
  pinMode(13,INPUT); //Pin 19 iz 168ice - MSB
  pinMode(12,INPUT); //Pin 18 iz 168ice
  pinMode(11,INPUT); //Pin 17 iz 168ice
  pinMode(10,INPUT); //Pin 16 iz 168ice - LSB
  pinMode(8,OUTPUT); //Pin za upravljanje žaruljom: Pin 14 iz 168ice
  ds1620.config(); //**** KONFIGURACIJA TERMOMETRA - pokreni neprekidno mjerenje temperature ****//
}

int Zad_Temp=22; //Trenutna zadana temperatura - inicijalno 22 C stupnjeva

int Sinkronizacija(){ //Funkcija za ispitivanje sinkronizacijske riječi
  int i=0,t=1000000; //t je velik kako bi osigurao vrijeme u kojem bi se sinkr. impuls sigurno pojavio
  while(!i&&(t)){ //Ako je pročitan sink. riječ (impuls)
    if ((digitalRead(13)==0)&&(digitalRead(12)==0)
        &&(digitalRead(11)==0)&&(digitalRead(10)==1))
      i=1; //Vrati jedinicu
    t--; //Umanji t
  }
  return i;
}

void InicijalnoStanje(){ //Funkcija za ispitivanje inicijalnog stanja
  if ((digitalRead(13)==0)&&(digitalRead(12)==0)
      &&(digitalRead(11)==0)&&(digitalRead(10)==0))
    Zad_Temp=22; //Ako se na sabirnici nalazi riječ 0000 onda inicijaliziraj temperaturu
}

void CitajSabirnicu(){ //Funkcija za čitanje zadaće sa sabirnice
  InicijalnoStanje(); //Ispitaj je li postavljeno inicijalno stanje na sabirnici
  if ((digitalRead(13)==0)&&(digitalRead(12)==1)){ //Ako je adresa za temperaturu
    if(digitalRead(11)==0){ //Ako se temperatura mijenja za 1 C stupanj
      if(digitalRead(10)==0){ //0100 -> smanji za 1 C stupanj
        if(Sinkronizacija) Zad_Temp--; //kad se pojavi sink. riječ smanji temperaturu
      }
    }
    else{
      if(Sinkronizacija) Zad_Temp++; //0101 -> povećaj za 1 C stupanj
    }
  }
  if(digitalRead(11)==1){ //Ako se postavlja definirana temperatura
    if(digitalRead(10)==0){ //0110 -> postavi 25 C stupnjeva
      if(Sinkronizacija) Zad_Temp=25; //kad se pojavi sink. riječ postavi temperaturu
    }
    else{
      if(Sinkronizacija) Zad_Temp=30; //0111 -> postavi 30 C stupnjeva
    }
  }
}

void Grijanje(float Ocitana_temp){ //Funkcija za upravljanje žaruljom
  if(Ocitana_temp<=Zad_Temp-1) digitalWrite(8,0); //Žarulja se pali postavljanjem nule a gasi postavljanjem jedinice na pin 14.
  else if (Ocitana_temp>=Zad_Temp+1) digitalWrite(8,1);
}

void loop(){ //Glavna petlja programa
  CitajSabirnicu(); //Čitaj sabirnicu i podatkovne okvire
  float Temp = ds1620.temp_c()/2; //Pročitaj temperaturu. Razlučivost je 0.5 C stupnjeva i
  Grijanje(Temp); //podatak bi trebalo podijeliti sa 2 za točan prikaz temperature.
  lcd.clear(); //Izgled sučelja LCD za prikaz namještene i trenutne temperature
  lcd.setCursor(0,0);
}
```

```

lcd.print("Temperatura:");
lcd.setCursor(0,1);
lcd.print(Temp);
lcd.setCursor(6,1);
lcd.print("C");
lcd.setCursor(12,1);
lcd.print("(");
lcd.setCursor(13,1);
lcd.print(Zad_Temp);
lcd.setCursor(15,1);
lcd.print(")");
}

```

Čvor za pokretanje muzike

```

void setup(){
  delay(2000); //Pričekati dok se ne podigne sustav
              //Pinovi za spajanje na sabirnicu
  pinMode(13,INPUT); //Izlaz 19 iz 168ice - MSB
  pinMode(12,INPUT); //Izlaz 18 iz 168ice
  pinMode(11,INPUT); //Izlaz 17 iz 168ice
  pinMode(10,INPUT); //Izlaz 16 iz 168ice - LSB
  pinMode(8,OUTPUT); //Pin za uključivanje sklopa za muziku - Izlaz 14 iz 168ice
  pinMode(9,OUTPUT); //Pin za pokretanje muzike i namještanje pjesama - Izlaz 15 iz 168ice
}

int Sinkronizacija(){ //Funkcija za ispitivanje sinkronizacijske riječi
  int i=0,t=1000000; //t je velik kako bi osigurao vrijeme u kojem bi se sinkr. impuls sigurno pojavio
  while(!i&&(t)){ //Ako je pročitan sink. riječ (impuls)
    if ((digitalRead(13)==0)&&(digitalRead(12)==0)
        &&(digitalRead(11)==0)&&(digitalRead(10)==1))
      i=1; //Vrati jedinicu
    t--; //Umanji t
  }
  return i;
}

void InicijalnoStanje(){ //Funkcija za ispitivanje inicijalnog stanja
  if ((digitalRead(13)==0)&&(digitalRead(12)==0)
      &&(digitalRead(11)==0)&&(digitalRead(10)==0)){
    digitalWrite(8,0); //Ako se na sabirnici nalazi riječ 0000 onda ugasi muziku
    digitalWrite(9,1); //Postavi pin za prebacivanje pjesme u neaktivno stanje
  }
}

void CitajSabirnicu(){ //Funkcija za čitanje zadaće sa sabirnice
  InicijalnoStanje(); //Ispitaj je li postavljeno inicijalno stanje na sabirnici
  if ((digitalRead(13)==1)&&(digitalRead(12)==0)){ //Ako je adresa za muziku
    if(digitalRead(11)==0){ //Paljenje ili gašenje
      if(digitalRead(10)==0){ //1000 -> ugasi muziku
        if(Sinkronizacija) digitalWrite(8,0); //Kad se pojavi sink. riječ ugasi muziku
      }
      else{ //1001 -> upali muziku
        if(Sinkronizacija){ //Kad se pojavi sink. riječ
          digitalWrite(8,0); //prvo ugasi muziku
          delay(20); //pa sačekaj 20 ms da modul to registrira
          digitalWrite(8,1); //Zatim upali muziku kako bi reprodukcija krenula od prve pjesme
        }
      }
    }
    if(digitalRead(11)==1)
      if(digitalRead(10)==0) //1011 ->prebaci pjesmu unaprijed
        if(Sinkronizacija){ //kad se pojavi sink. riječ
          digitalWrite(9,0); //spoji kratko WTV priključak za prebacivanje pjesme
          delay(20); // i nakon 20 ms
          digitalWrite(9,1); //opet ga odspoji i prebacio si pjesmu za jednu unaprijed
        }
  }
}

void loop(){ //Glavna petlja programa
  CitajSabirnicu(); //Uvijek čitaj sabirnicu
}

```

Čvor za upravljanje trošilima

```
void setup(){
    //Pinovi za spajanje na sabirnicu
    pinMode(13,INPUT); //Izlaz 19 iz 168ice - MSB
    pinMode(12,INPUT); //Izlaz 18 iz 168ice
    pinMode(11,INPUT); //Izlaz 17 iz 168ice
    pinMode(10,INPUT); //Izlaz 16 iz 168ice - LSB
    //Pinovi za kontrolu RLS uređaja
    pinMode(9,OUTPUT); //Izlaz 15 iz 168ice - OFF2
    pinMode(8,OUTPUT); //Izlaz 14 iz 168ice - OFF1
    pinMode(7,OUTPUT); //Izlaz 13 iz 168ice - ON2
    pinMode(6,OUTPUT); //Izlaz 12 iz 168ice - ON1
}

void InicijalnoStanje(){ //Funkcija za ispitivanje inicijalnog stanja
    if ((digitalRead(13)==0)&&(digitalRead(12)==0)
        &&(digitalRead(11)==0)&&(digitalRead(10)==0)){ //Ako se na sabirnici nalazi riječ 0000 onda izvrši inicijalizaciju.
        digitalWrite(9,0); //Odspoji sve četiri tipke.
        digitalWrite(8,0);
        digitalWrite(7,0);
        digitalWrite(6,0);
        delay(50); //Sačekaj 50 ms da se te zadaće ispune.
        digitalWrite(9,1); //Ako je slučajno koji uređaj bio uključen
        delay(100); //kratko spoji obe OFF tipke kako bi ih sigurno
        digitalWrite(9,0); //ugasio.
        delay(50); //Sačekaj 100 ms pa odspoji tipku OFF pa nakon
        digitalWrite(8,1); //50 ms to ponovi za drugu OFF tipku
        delay(100);
        digitalWrite(8,0);
    }
}

int Sinkronizacija(){ //Funkcija za ispitivanje sinkronizacijske riječi
    int i=0,t=1000000; //Zastavica i veliki brojač nakon kojeg se izlazi iz petlje
    while(!i&&t){
        if ((digitalRead(13)==0)&&(digitalRead(12)==0) //t je velik kako bi osigurao vrijeme u kojem bi se sinkr. impuls sigurno pojavio
            &&(digitalRead(11)==0)&&(digitalRead(10)==1))
            i=1; //Kad se pojavi sink. riječ vrati jedinicu
        t--; //Umanji t
    }
    return i;
}

void CitajSabirnicu(){ //Funkcija za čitanje zadaće sa sabirnice
    InicijalnoStanje(); //Ispitaj je li postavljeno inicijalno stanje na sabirnici
    if ((digitalRead(13)==1)&&(digitalRead(12)==1)){ //Ako je adresa za trošila (11):
        if(digitalRead(11)==0){ //Upravljanje kućanskim aparatom
            if(digitalRead(10)==0){ //1100 - isključi teret
                if(Sinkronizacija){ //Nakon sinkronizacije pritisni OFF1 tipku
                    digitalWrite(8,1);
                    delay(100); //Sačekaj 100ms da se teret izgasi i onda tipku odspoji
                    digitalWrite(8,0);
                    delay(50); //Opet sačekaj 50 ms da bi sklop proveo odspajanje
                } //Analogno za donje slučajeve uključenja i isključenja
            }
        }
        else{
            if(Sinkronizacija){ //1101 - uključi teret
                digitalWrite(6,1); //Pritisni i odspoji tipku ON1
                delay(100);
                digitalWrite(6,0);
                delay(50);
            }
        }
    }
}

if(digitalRead(11)==1){ //Upravljanje rasvjetom
    if(digitalRead(10)==0){ //1110 - isključi teret
        if(Sinkronizacija){
            digitalWrite(9,1); //Pritisni i odspoji tipku OFF2
            delay(100);
            digitalWrite(9,0);
            delay(50);
        }
    }
    else{
        if(Sinkronizacija){ //1101 - uključi teret
            digitalWrite(7,1); //Pritisni i odspoji tipku ON2
            delay(100);
            digitalWrite(7,0);
            delay(50);
        }
    }
}
```

```
}  
}  
}  
}  
}  
void loop(){                               //Glavna petlja programa  
  CitajSabirnicu();                         //Citaj okvire sa sabirnice  
}
```