

Anita Škarica  
0036423668

Ivan Sović  
0036419850

SEMINARSKI RAD - SPVP



# Ethernet i IP komunikacija

Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu  
Zavod za električne sustave i obradu informacija  
Sveučilište u Zagrebu



- △ Za sve one koji žele znati više!
- △ Postupci povezivanja sustava
- △ Korištenje Etherneta i IP protokola
- △ Upute za realizaciju komunikacije

## Sažetak

„Inteligentne kuće“ rastući su trend posljednjih godina, a sama ideja rodila se još prije nekoliko desetljeća. Za razliku od „običnih“ kuća, inteligentna kuća ima sposobnost prilagođavati se potrebama stanara, paziti na sigurnost, ili npr. voditi brigu o kućnim ljubimcima. U inteligentne kuće se danas uglavnom ugrađuju sustavi za regulaciju klimatizacije, osvjetljenja, otvaranje i zatvaranje vrata i prozora te razni sigurnosni podsustavi. Postavljaju se pitanja fizičkog povezivanja podsustava, odabira odgovarajućih tehnologija, te implementacija protokola za komunikaciju između podsustava.

Iako broj protokola i standarda raste iz dana u dan, uglavnom se održe oni koji su jednostavniji i robustniji. Tako su za povezivanje podsustava u projektu „Inteligentna kuća“ odabrani Ethernet i TCP/IP protokol. Nedostatak Etherneta je mogućnost prisluškivanja komunikacije izvana, dok je TCP/IP sporiji od alternativnog UDP-a. Pojedini podsustavi razvijani su na Arduino mikrokontrolerskim pločicama, te je bilo potrebno omogućiti njihovu komunikaciju putem upravo spomenutih tehnologija. Komunikacija je uspješno izvedena i testirana, te je donesen zaključak kako je odabir tehnologije i protokola bio ispravan.

## Sadržaj

|        |   |    |
|--------|---|----|
| 1.     | UVOD .....                                    | 4  |
| 2.     | OPIS SUSTAVA „INTELIGENTNA KUĆA“ .....        | 5  |
| 3.     | KOMUNIKACIJA IZMEĐU PODSUSTAVA .....          | 7  |
| 3.1.   | Povezivanje i protokol .....                  | 7  |
| 3.2.   | Postojeća rješenja .....                      | 9  |
| 4.     | IZVEDBA KOMUNIKACIJE .....                    | 10 |
| 4.1.   | Potrebne komponente .....                     | 10 |
| 4.2.   | Kratki opis komponenata .....                 | 10 |
| 4.2.1. | Arduino mikrokontrolerska pločica .....       | 10 |
| 4.2.2. | Arduino Ethernet Shield .....                 | 11 |
| 4.2.3. | Kabeli .....                                  | 11 |
| 4.2.4. | Switch .....                                  | 12 |
| 4.2.5. | Arduino programski paket .....                | 13 |
| 4.2.6. | Računalo .....                                | 14 |
| 4.3.   | Kôd .....                                     | 14 |
| 4.4.   | Način povezivanja komponenata u cjelinu ..... | 18 |
| 4.5.   | Primjena u drugim projektima .....            | 19 |
| 4.5.1. | Primanje naredbi putem IR signala .....       | 19 |
| 4.5.2. | ZigBee .....                                  | 19 |
| 5.     | TESTIRANJE .....                              | 20 |
| 6.     | ZAKLJUČAK .....                               | 22 |
| 7.     | LITERATURA .....                              | 23 |
| 8.     | POJMOVNIK .....                               | 24 |

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

## 1. Uvod

Inteligentne kuće rastući su trend posljednjih godina, a sama ideja rodila se još prije nekoliko desetljeća. Zamisao je da se spojem tehnologije i stila življenja poveća komfor stanara u kući, te olakša obavljanje svakodnevnih rutinskih, ili možda čak i opasnih zadataka. U kuće se uglavnom ugrađuju sustavi za regulaciju klimatizacije, osvjetljenja, otvaranje i zatvaranje vrata i prozora te razni sigurnosni podsustavi. Svi ovi podsustavi uglavnom se skupljaju na jednom, središnjem mjestu na kojem se omogućuje korisniku (vlasniku kuće) pregled i mijenjanje svih postavki, kao i pregled statusa kuće.

Projekt opisan ovim radom dio je većeg projekta pod nazivom „Inteligentna kuća“, i bavi se upravo takvim povezivanjem raznih podsustava u cjelinu. Samim time nameću se pitanja fizičkog povezivanja podsustava, odabira odgovarajućih tehnologija, te implementacija protokola za komunikaciju između podsustava. Rješavanjem ovih problema trebalo bi se omogućiti jeftinu, a opet pouzdanu i brzu komunikaciju međusobno odvojenih uređaja. Svrha toga je povezivanje podsustava u cjelinu, kako bi oni mogli komunicirati, razmjenjivati informacije, ili slati naredbe jedni drugima. Bez ovakvog rješenja, pojedini uređaji morali bi raditi autonomno, čime bi se ideja pametne kuće reducirala samo na regulaciju nekih procesa.

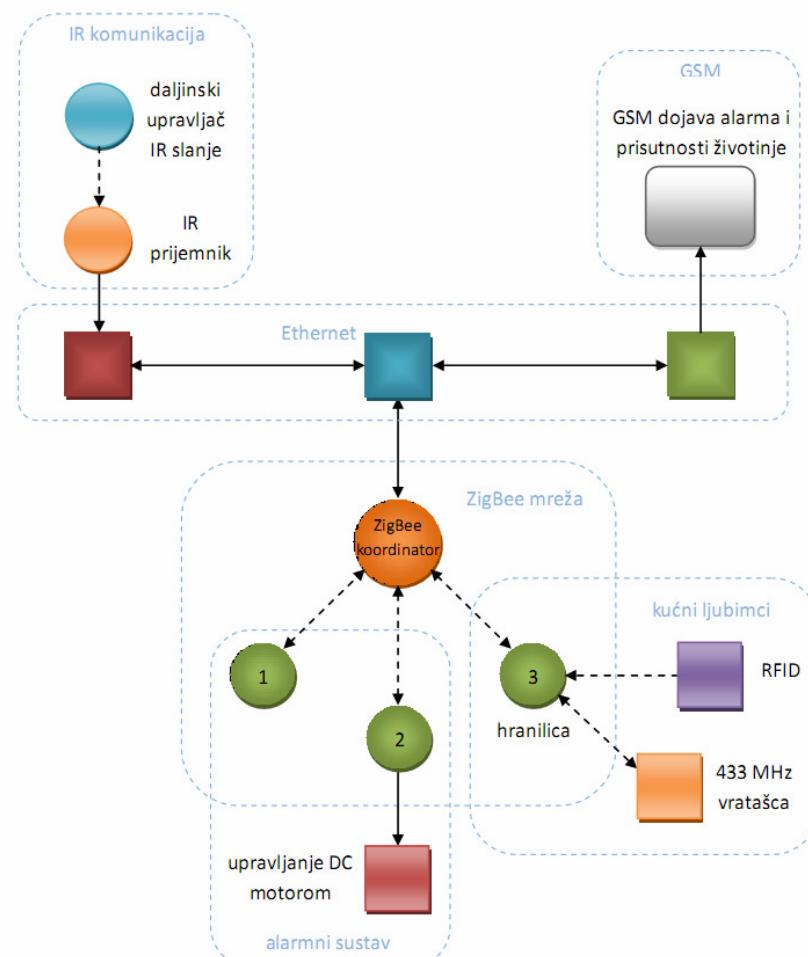
## 2. Opis sustava „Inteligentna kuća“

Razvoj sustava kao što je „Inteligentna kuća“ nije nimalo jednostavan posao. Stoga ga je potrebno podijeliti u manje cjeline, tj. podsustave, koji će biti zaduženi za upravljanje i nadzor pojedinih procesa unutar kuće, te za komunikaciju. „Inteligentna kuća“ je podijeljena na sljedeće podsustave:

- Komunikacija putem modema
- Upravljanje kućom putem web servera
- Primanje govornih naredbi
- Zaštita u slučaju požara
- Klimatizacija
- Grafički prikaz kuće, stanja i procesa
- Primanje naredbi putem IR signala
- Emitiranje naredbi IR signalom
- Digitalni ulazi i izlazi
- Rasvjeta
- Čuvaj kuću
- Obavljanje kupovine
- ZigBee
- Briga o kućnim ljubimcima
- Ethernet i IP komunikacija
- RS232
- Komunikacija putem SMS poruka GSM telefona
- 433MHz
- RFID određivanje prava pristupa

Sve ove podsustave potrebno je međusobno povezati. Dogovoren je, na primjer, da će se projekti „Briga o kućnim ljubimcima“ i „Čuvaj kuću“ povezivati putem ZigBee-ja na ZigBee koordinatorski čvor. S druge strane „Primanje naredbi IR signalom“ i „Emitiranje naredbi IR signalom“ su nužno povezani projekti. Također, bilo bi korisno kada bi se sve promjene i događaji u inteligentnoj kući mogli bilježiti u neku bazu podataka (za što nam treba web server). Ovakvim razmišljanjem možemo uočiti kako bi bilo potrebno sve ove, već djelomično objedinjene cjeline, također međusobno povezati. Upravo o tome će biti riječi u nastavku. Na slici 1 prikazan je dio sustava „Inteligentne kuće“, na kojem se mogu

uočiti međusobne veze nekih podsustava, te tehnologije komunikacije između njih.



**Slika 1. Prikaz dijela sustava "Inteligentne kuće".**

### 3. Komunikacija između podsustava

#### 3.1. Povezivanje i protokol

Podsustave inteligentne kuće potrebno je međusobno žičano povezati, i to radi sigurnosti podataka i brzine upravljanja. Danas za tu svrhu postoje razni načini povezivanja uređaja, kao što su: *Ethernet*, *RS-232*, *modem*,... U sklopu ovog rada, odabrana je komunikacija putem *Etherneta*. To je vrsta mrežne tehnologije koja se koristi za izgradnju lokalnih mreža. Ethernet je dobar izbor iz nekoliko glavnih razloga: velike raširenosti, jednostavnosti i niske cijene postavljanja mreže [1]. Također, brzina komunikacije Ethernetom je neusporedivo veća od ostalih nabrojenih metoda (preko 1 Gbps), što može biti od velikog značaja za regulaciju brzim procesima unutar ili izvan kuće.

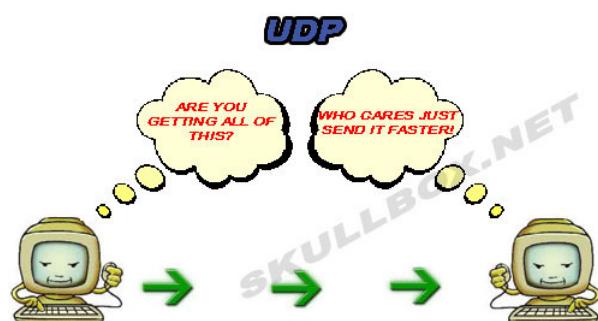
Broj protokola i standarda raste iz dana u dan, a uglavnom se održe oni koji su jednostavnji i robusni. Upravo takav je i *IP protokol* (skraćenica od „*Internet Protocol*“), na kojemu se bazira vrlo veliki broj protokola više razine, te omogućuje prijenos svih vrsta podataka [2]. Za komunikaciju uređaja IP protokolom, potrebno je poznavati njihove *IP adrese* (adrese uređaja na mreži), te *port* na kojem se šalju, odnosno primaju poruke. U doba kada čak i telefoni, televizori i kamere dobivaju IP adresu, bilo je sasvim logično iskoristiti ovakvo provjereno rješenje u intelligentnoj kući. Povežemo li sve podsustave u kući IP protokolom omogućujemo pristup svakom pojedinom podsustavu kuće izvana bez ikakve prilagodbe.

IP omogućuje, na transportnom sloju, komunikaciju pomoću dva protokola: *TCP* (eng. *Transmission Control Protocol*) i *UDP* (eng. *User Datagram Protocol*). TCP protokol omogućuje spor, ali pouzdan prijenos podataka. To je *vezno orijentirani protokol* (eng. *connection oriented*), što znači da će paket, ako ne stigne na odredište, biti ponovno poslan.



Slika 2. TCP/IP protokol.

S druge strane, prijenos podataka pomoću UDP-a je brz ali nepouzdan. On nije vezno orijentiran (eng. *connectionless*), što znači da ne postoji mehanizam ponovnog slanja podataka u slučaju neuspjeha [3]. Zbog potrebe za pouzdanim prijenosom podataka, u ovom projektu koristi se TCP/IP protokol za povezivanje raznih podsustava.



Slika 3. UDP protokol.

### 3.2. Postojeća rješenja

Prilikom razrade ideja o projektu „Inteligentna kuća“, zamišljeno je da se pojedine komponente neće razvijati potpuno iz početka, već da se koriste postojeća rješenja i njihove kombinacije. Tako uglavnom svaki podsustav inteligentne kuće koristi isti mikrokontroler i dodatne sklopove. Konkretnije, riječ je o *Arduino mikrokontrolerskoj pločici*. Ona sadrži mikrokontroler, analogne i digitalne izvode, te sve ostale potrebne komponente da bi se mikrokontroler mogao koristiti [4]. Za upotrebu Arduino pločice, sve što je potrebno napraviti, je priključiti pločicu na *USB sabirnicu ili napajanje*, i ona je spremna za rad.

Iako je ova pločica, sama po sebi, dovoljna za velik broj primjena, za izradu sustava koji komunicira preko Etherneta potrebne su nam i neke dodatne komponente. Arduino pločica nema ugrađene potrebne izvode za RJ-45 priključak, te je stoga za povezivanje potrebno smisliti dodatno rješenje. Arduino također, osim mikrokontrolerskih pločica, nudi i razne dodatke za njih (eng. *shields*). Tako nam za ovu primjenu odgovara *Ethernet Shield* pločica, koja ujedno sadrži potrebne priključke, te pripadajuće sklopovlje za ostvarivanje Ethernet komunikacije [5]. Za povezivanje sa mikrokontrolerskom pločicom, potrebno ju je samo utaknuti u tu pločicu na predviđenim utorima.

Kako bi korisnicima olakšao korištenje svojih proizvoda te omogućio brzo započinjanje i provedbu projekata, Arduino je proizveo vlastiti programski paket za programiranje, kompajliranje (eng. *compiling*) i prebacivanje programa na mikrokontroler (eng. *uploading*) [6]. Programski paket se isporučuje zajedno sa svim potrebnim bibliotekama za korištenje serijskog porta, memorije, pokaznika, itd., pa tako i Etherneta i TCP/IP protokola. Zbog velike popularnosti Arduinovih mikrokontrolerskih pločica, postoji velika zajednica korisnika koja uvelike doprinosi razvoju raznih mogućnosti i softverskih rješenja. Tako se mogu pronaći biblioteke za korištenje UDP protokola, obradu zvučnog signala, slike,...

## 4. Izvedba komunikacije

### 4.1. Potrebne komponente

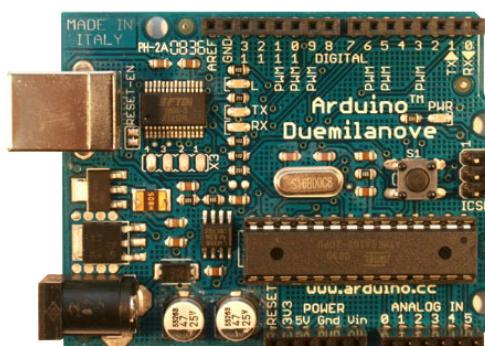
Za izradu ovog sustava korištene su slijedeće komponente:

- Arduino Duemilanove mikrokontrolerska pločica
- Arduino Ethernet Shield pločica
- UTP kabel
- USB kabel sa konektorima A-B
- LAN switch
- Arduino programski paket
- računalo

### 4.2. Kratki opis komponenata

#### 4.2.1. Arduino mikrokontrolerska pločica

U ovom projektu korištena je Duemilanove mikrokontrolerska pločica. Arduino, osim ove serije, nudi još mnogo vrsta sličnih pločica, od kojih su neke: Diecimila, Nano, Mega, LilyPad, itd., a razlikuju se po ugrađenim mikrokontrolerima, broju izvoda, te nekim drugim komponentama. Neke od mogućnosti koje Duemilanove nudi su: ATmega168 ili ATmega328 mikrokontroler, 14 digitalnih ulazno/izlaznih priključaka, 6 analognih ulaza, 16MHz kristalni oscilator, povezivanje putem USB-a, vanjsko napajanje,...



Slika 4. Arduino Duemilanove mikrokontrolerska pločica.

#### 4.2.2. Arduino Ethernet Shield

Ova pločica omogućuje spajanje uređaja na mrežu. Temeljena je na Wiznet W5100 Ethernet čipu, koji sadrži mrežni stog (eng. stack), te je sposoban komunicirati putem TCP-a i UDP-a. Povezuje se sa mikrokontrolerskom pločicom tako da se postavi povrh nje, i time poveže sa priključcima.



Slika 5. Arduino Ethernet Shield.

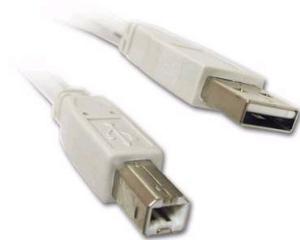
#### 4.2.3. Kabeli

Potrebne su dvije vrste kabela: za povezivanje na mrežu, te za povezivanje na USB sabirnicu. Uredaj se na mrežu pak može povezati sa dvije vrste UTP kabela, i to sa ravnim (eng. *straight*) ili ukriženim (eng. *crossover*) vodovima. Ukriženi se koriste za izravno spajanje uređaja sa drugim uređajem ili računalom, dok ravni služe spajanju na mrežu preko *switch*-eva.

USB kabel je potreban za povezivanje Arduino mikrokontrolerske pločice sa računalom pomoću kojeg je želimo isprogramirati. Osim u postupku programiranja, također se može koristiti i za uspostavljanje serijske veze sa drugim uređajima, te prijenos podataka.



**Slika 6. RJ-45 konektor za UTP kabel.**



**Slika 7. USB kabel sa A i B konektorom.**

## 4.2.4. Switch

Ukoliko želimo povezati više od dva uređaja, ne možemo ih spojiti izravno, već moramo koristiti dodatni uređaj koji se naziva *switch*. To je uređaj koji upravlja protokom podataka između dijelova lokalne mreže. On sprječava koliziju paketa podataka i maksimizira brzinu slanja [7]. Za potrebe testiranja povezivanja podsustava „Inteligentne kuće“ korišteni su switchevi dostupni u laboratorijima Zavoda za električke sustave i obradbu informacija.



**Slika 8. Ethernet switch.**

#### 4.2.5. Arduino programski paket

Arduino je kreirao svoj programski paket koji korisnicima omogućuje unos programa, njihovo jednostavno kompajliranje, te prebacivanje na mikrokontrolersku pločicu. U razvojnog okruženju korisnik dobiva i povratnu informaciju od kompjulera (u slučaju da program sadrži greške), a osim toga može prikazivati i slati podatke preko serijske veze (npr. USB), što može olakšati testiranje uređaja jednom kada je već isprogramiran. Programski paket se može besplatno preuzeti sa Arduinovih *web* stranica.

#### 4.2.6. Računalo

Kako bi se programski paket mogao koristiti, te isprogramirati mikrokontrolerska pločica, potrebno je računalo koje zadovoljava potrebne zahtjeve paketa i pločice. Također, računalo se može koristiti i za testiranje sustava nakon programiranja mikrokontrolera.

### 4.3. Kod

Programski dio ovog rada sastoji se od četiri funkcije, te dodatnih definicija. Slijedi kratki opis tih funkcija:

`void Postavi_Ethernet_IP()` – sadrži sve što se mora postaviti na početku izvođenja koda, tj. prije glavne petlje.

`void Salji_TCP(byte send_cvor, char *send_podatak)` – šalje paket podataka pomoću TCP/IP-a na čvor čiji je redni broj dan parametrom `send_cvor`.

`int Primi_TCP()` – služi za primanje i pohranjivanje paketa putem TCP/IP protokola.

`void Primi_Serijski()` – prima niz podataka preko serije, te ih pohranjuje u memoriju.

Odabir čvora na kojem će se ovaj kod izvoditi mora se napraviti na vrhu programa, i to na sljedeći način:

```
#define ovaj_arduino port_arduino1
```

To je potrebno napraviti kako bi se olakšalo podešavanje komunikacije putem IP-a, jer bi se u protivnome za svaku mikrokontrolersku pločicu morao podešavati kod na većem broju mjesta. Osim postavke `port_arduino1` (koja predstavlja prvi arduino čvor), moguće je koristiti i `port_arduino2` te `port_arduino3`. Svaki od ovih čvorova ima posebno definiranu IP adresu, MAC adresu, te portove. U nastavku je dan kod napisan u sklopu ovog rada.

```
#include <Ethernet.h>
#include <stdio.h>
#include <string.h>

#define ovaj_arduino port_arduino3

byte mac[] = { 0xD3, 0xA1, 0xBE, 0xEF, 0xFE, 0xED };

char podatak1[10];
char podatak_ser[10];
char podatak_prim[10];
char podatak2[10];
char podatak4[10];

int pprim = 0, p1 = 0, p4 = 0, pser=0, j=0;
int velicina_poruke = 2;

byte ip_arduino1[]={ 161, 53, 64, 80 };
byte ip_arduino2[]={ 161, 53, 64, 118 };
byte ip_arduino3[]={ 161, 53, 64, 99 };

#define port_arduino1 9760
#define port_arduino2 300
#define port_arduino3 400

#if ovaj_arduino != port_arduino1
    Client client1(ip_arduino1, port_arduino1); // klijent za slanje na
arduino1
#endif

#if ovaj_arduino != port_arduino2
    Client client2(ip_arduino2, port_arduino2); // klijent za slanje na
arduino2
#endif

#if ovaj_arduino != port_arduino3
    Client client3(ip_arduino3, port_arduino3); // klijent za slanje na
arduino3
#endif

#if ovaj_arduino == port_arduino1
    #define ip ip_arduino1
#elif ovaj_arduino == port_arduino2
    #define ip ip_arduino2
#else
    #define ip ip_arduino3
#endif

Server server(ovaj_arduino);

void Postavi_Ethernet_IP()
{
    Ethernet.begin(mac, ip);
    Serial.begin(9600);
    server.begin();
    delay(10);
    Serial.println("init...");
}
```

```
void Salji_TCP(byte send_cvor, char *send_podatak)
{
    if (send_cvor == 1)
    {
        #if ovaj_arduino != port_arduino1
        if (client1.connect())
        {
            Serial.print("saljem podatak: ");
            Serial.println(send_podatak);
            // popuniti prvi bajt (odrediste) prema send_cvoru
            client1.println(send_podatak);
            client1.stop();
        }
        #else
            Serial.println("ERROR! Ne mogu poslati na isti arduino!!!");
            return;
        #endif
    }
    else if (send_cvor == 2)
    {
        #if ovaj_arduino != port_arduino2
        if (client2.connect())
        {
            Serial.print("saljem podatak: ");
            Serial.println(send_podatak);
            client2.println(send_podatak);
            client2.stop();
            //Serial.print("zatvorio konekciju prema 2 ");
        }
        #else
            Serial.println("ERROR! Ne mogu poslati na isti arduino!!!");
            return;
        #endif
    }
    else if (send_cvor == 3)
    {
        #if ovaj_arduino != port_arduino3
        if (client3.connect())
        {
            Serial.print("saljem podatak: ");
            Serial.println(send_podatak);
            client3.println(send_podatak);
            client3.stop();
        }
        #else
            Serial.println("ERROR! Ne mogu poslati na isti arduino!!!");
            return;
        #endif
    }

    delay(50);
}
```

```
void Primi_Serija()
{
    int j=0;

    if (Serial.available ())
    {
        Serial.print("cekam podatak...");

        podatak_ser[pser] = Serial.read();
        if (pser == velicina_poruke)
        {
            for (j=0; j<pser; j++)
                podatak1[j] = podatak_ser[j];

            p1 = pser;
            p1 += 1;
            podatak1[p1] = '\0';
            pser = -1;

            if (podatak1[0] =='1')
                Salji_TCP (1, podatak1);
            if (podatak1[0] == '2')
                Salji_TCP(2, podatak1);
            Serial.print(podatak1);

        }
        pser++;
    }
    delay(50);
}

int Primi_TCP()
{
    int retVal = 0;
    Client client = server.available();
    if (client)
    {
        while (client.connected())
        {
            if (client.available())
            {
                podatak_prim[pprim] = client.read();
                if (podatak_prim[pprim] == '\0')
                {
                    retVal = 1;
                    pprim = -1; // ocisti podatak_prim
                }
                else
                    retVal = 0;
                pprim++;
                break;
            }
        }
    }
    return retVal;
}
```

#### 4.4. Način povezivanja komponenata u cjelinu

1. Preuzeti Arduinov programski paket sa njihovih *web* stranica, te ga instalirati.
2. Uzeti mikrokontrolersku i Ethernet pločicu, te ih povezati postavljanjem Ethernet pločice povrh mikrokontrolerske.
3. Priključiti UTP kabel u priključak na Ethernet pločici.
4. Priključiti USB kabel na mikrokontrolersku pločicu s jedne strane, te na računalo s druge strane.
5. Pokrenuti programski paket, te upisati kod. Primjer koda koji koristi funkcije dane u prethodnom poglavlju:

```
void setup()
{
    Postavi_Ethernet_IP();
}

void loop()
{
    if (Primi_TCP())
        Serial.println(podatak_prim);
    delay(200);
}
```

U ovom primjeru omogućuje se komunikacija između tri uređaja. Konkretnije, trenutno programirani čvor čeka dolazne podatke preko Etherneta, te ih prolijeduje dalje na seriju.

6. Odabrati opciju Sketch → Verify/Compile, te ako se ona uspješno izvršila, File → Upload to I/O Board.
7. Testirati sustav.
8. **NAPOMENA:** zbog načina na koji je „Inteligentna kuća“ zamišljena, svaki Arduino uređaj mora biti „svjestan“ svih ostalih uređaja spojenih na mrežu (tj. njihovih IP adresa i portova), ako s njima želi komunicirati. Tako je npr. potrebno upravo dani kod prilagoditi za svaku mikrokontrolersku pločicu koja će biti isprogramirana, jer dva uređaja na mreži ne mogu imati iste *IP* i *MAC* (eng. *Medium Access Control*) adrese.

## 4.5. Primjena u drugim projektima

Ethernetom je uspostavljena komunikacija između slijedećih podsustava „Inteligentne kuće“:

- Primanje naredbi putem IR signala
- ZigBee

### 4.5.1. Primanje naredbi putem IR signala

Uloga IR prijamnika u pametnoj kući je da omogući direktno kontroliranje odgovarajućih sustava od strane korisnika. Prijamnik je povezan putem Etherneta sa ostalim čvorovima, na kojima mogu biti priključeni odgovarajući uređaji kojima se želi upravljati. Poruke koje se prenose veličine su 2 okteta, a njihov sadržaj može biti proizvoljan, tj. ovisan o sustavima kojima se upravlja (alarmni sustav, sustav rasvjete,...).

### 4.5.2. ZigBee

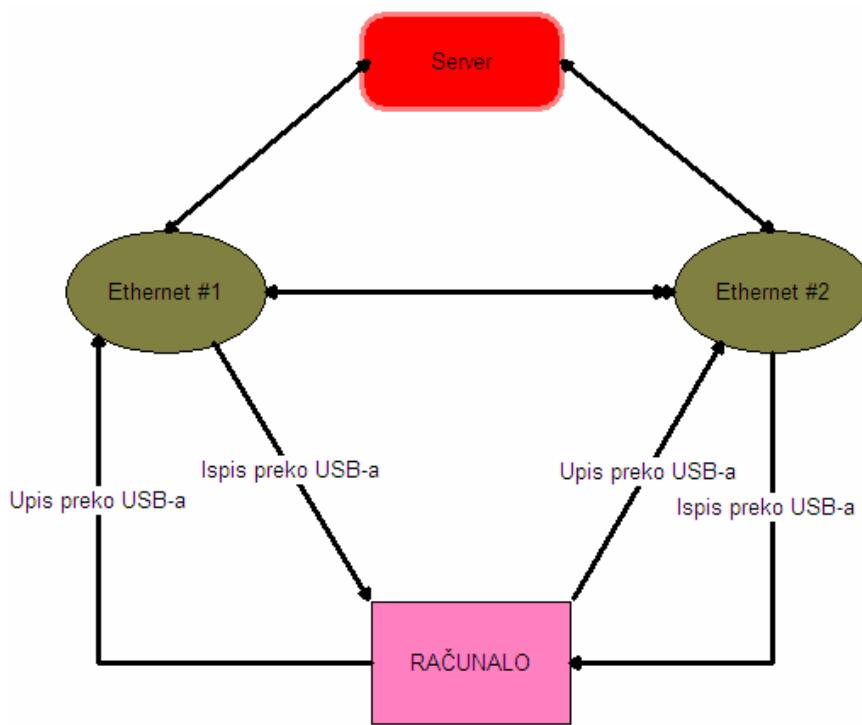
ZigBee mrežom povezani su određeni periferni podsustavi. Sastoje se od nekoliko cjelina:

- koordinatorski čvor
- senzorski čvor alarmnog sustava
- aktuatorski čvor alarmnog sustava
- čvor sustava za brigu o kućnim ljubimcima

Koordinatorski čvor brine o stvaranju bežične mreže, spajanju i autentifikaciji udaljenih čvorova, te se sav promet u ZigBee mreži (koji ide prema ili od ostalih čvorova) odvija preko njega. Za povezivanje sklopova, spojenih putem ZigBee-ja, s ostatkom sustava, koristi se Ethernet. Koordinator preko Etherneta prima naredbe ili šalje podatke širine jednog okteta.

## 5. Testiranje

Testiranje je obavljeno u nekoliko faza. Prvo je bilo potrebno uvjeriti se da Ethernet podsustav ispravno radi, prije nego se pokuša komunikacija s drugim podsustavima. Potrebno je ispitati slanje (odnosno primanje) paketa s jedne Arduino pločice na drugu, te slanje i primanje od poslužitelja. Kao poslužitelj, poslužilo nam je prijenosno računalo s instaliranim TCP monitorom. Shema testiranja prikazana je na slici 2.

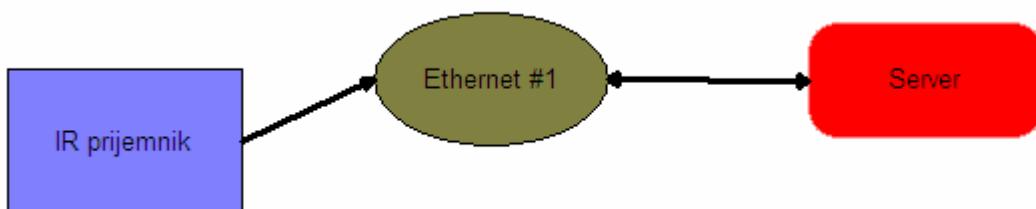


Slika 9. Shema sustava za testiranje

Na računalu je bilo moguće preko USB-a pratiti slanje i primanje poruka, kao i pratiti primanje na poslužitelju. USB sučelje korišteno je za *debug* kôda.

Nakon što je ova konfiguracija ispitana započeta je integracija s drugim podsustavima. Prvo je spojen jedini senzorski podsustav koji komunicira direktno s Ethernet mrežom, a to je IC prijemnik. On je bio

spojen, tj. integriran na jedan već postojeći Ethernet čvor. Nakon što je prijemnik primio poruku, ista bi se proslijedila na poslužitelj.



**Slika 10. Shema sustava za testiranje s IR prijemnikom**

Testiranje je nastavljeno spajanjem sa ZigBee mrežom i preko nje sa drugim podsustavima. Kako je Ethernet-IP komunikacija izvedena preko funkcija *Salji\_TCP()* i *Primi\_TCP()* bilo je potrebno samo u određenim dijelovima kôda ZigBee koordinatora pozvati funkcije za Ethernet. Međutim, dugotrajnim testiranjem, uočeno je da svaki podsustav zasebno dobro radi, ali da je propagacija signala između Ethernet čvora 1 i nekog krajnog ZigBee čvora prevelika (u pojmovima sigurnosnih podsustava moglo bi se čak reći i beskonačna). Zaključeno je da je najvjerojatnije Arduino na tom čvoru preopterećen, te čvor nema dovoljno procesorske snage za obradivanje događaja na oba *shielda*. Time je obustavljena konačna integracija svih podsustava, te se Ethernet mreža odvojila od ZigBee mreže. Posljedica navedenog je nemogućnost slanja podataka prikupljenih u jednoj mreži na drugu. Rješenje ovog problema leži u mogućnosti zamjene postojećeg mikrokontrolera modelom snažnijih karakteristika.

## 6. Zaključak

Sustavom opisanim u ovom projektu omogućena je komunikacija između različitih dijelova inteligentne kuće. Zahvaljujući odabranim standardima i protokolima (Ethernet i TCP/IP), omogućena je sigurna, a opet brza komunikacija. Pouzdanost prijenosa je od vrlo velikog značaja za ovu primjenu, jer se ne smije dopustiti da npr. informacije iz alarmnog sustava ne stignu na odredište. Odabrani TCP/IP protokol nam upravo to omogućava, a sam Ethernet je već dugotrajno korištena i isprobana tehnologija, što nam jamči ispravnost njegovog odabira. Komunikacija između podsustava je uspješno provedena, što potvrđuje ostvarenost ciljeva zadanih u uvodu, te opravdava sredstva kojima smo došli do njih.

U budućnosti, kako potrebe i mogućnosti za automatiziranjem pojedinih funkcija u kućanstvu prerastu trenutno predviđene, ovaj projekt će moći biti primjenjiv uz tek neznatne promjene. Zbog načina izvedbe, biti će potrebno proširiti programski kod iznesen u poglavljiju 4.3. sa podrškom za više mrežnih čvorova, i to po uzoru na već postojeći kod. Također, primjena tako modificiranog koda moći će biti implementirana po uzoru na postupak opisan u poglavljju 4.5. Kako su se javile poteškoće u povezivanju sa ZigBee podsustavom, bilo bi preporučljivo proučiti specifikacije drugih mikrokontrolerskih pločica od istog, ili čak konkurentnog proizvođača. U eventualnom dalnjem radu na ovom projektu također bi trebalo pratiti razvoj komunikacijskih tehnologija, te u slučaju pojave rješenja koje bi bilo bolje od trenutnog napraviti unaprijeđenu inačicu sustava.

## 7. Literatura

- [1] „What is Ethernet”, URL: <http://www.creative-wisdom.com/teaching/network/ethernet.shtml> (2009-07-06)
- [2] „Internet Protocol”, URL: [http://en.wikipedia.org/wiki/Internet\\_Protocol](http://en.wikipedia.org/wiki/Internet_Protocol) (2009-07-06)
- [3] „Understanding TCP/IP”, URL: <http://www.tech-faq.com/understanding-cisco-tcpip.shtml> (2009-07-06)
- [4] „Arduino Duemilanove”, URL: <http://arduino.cc/en/Main/ArduinoBoardDuemilanove> (2009-07-06)
- [5] „Arduino Ethernet Shield”, URL: <http://arduino.cc/en/Main/ArduinoEthernetShield> (2009-07-06)
- [6] „Arduino Software”, URL: <http://arduino.cc/en/Main/Software> (2009-07-06)
- [7] „What are LAN switches? ”, URL: <http://www.tech-faq.com/lan-switches.shtml> (2009-07-06)

## 8. Pojmovnik

| Pojam               | Kratko objašnjenje   | Više informacija potražite na   |
|---------------------|--|---|
| compiler            | Računalni program koji procesira izvorni kod, a kao rezultat daje asemblerski ili izvršni kod.   | <a href="http://en.wiktionary.org/wiki/compiler">http://en.wiktionary.org/wiki/compiler</a>   |
| connection oriented | U telekomunikacijama, ovaj pojam opisuje način slanja podataka pri kojem krajnji uređaji uspostave prijenosni kanal prije početka slanja podataka.   | <a href="http://en.wikipedia.org/wiki/Connection-oriented">http://en.wikipedia.org/wiki/Connection-oriented</a>                     |
| connectionless      | Način slanja paketa podataka bez uspostave prijenosnog kanala.   | <a href="http://en.wikipedia.org/wiki/Connectionless">http://en.wikipedia.org/wiki/Connectionless</a>                               |
| Ethernet            | Vrsta mrežne tehnologije koja se koristi za izgradnju lokalnih mreža.  | <a href="http://www.creative-wisdom.com/teaching/network/ether.html">http://www.creative-wisdom.com/teaching/network/ether.html</a> |
| GSM                 | Eng. <i>Global System for Mobile communications</i> . To je danas najpopularniji standard za mobilne telefone u svijetu.   | <a href="http://en.wikipedia.org/wiki/GSM">http://en.wikipedia.org/wiki/GSM</a>   |
| IP adresa           | Numerički identifikator i logička adresa računala koji koristi Internet protokol.  | <a href="http://en.wikipedia.org/wiki/IP_address">http://en.wikipedia.org/wiki/IP_address</a>                                       |
| IP protokol         | Eng. <i>Internet Protocol</i> . To je protokol za razmjenu podataka preko mreže pomoću paketa.   | <a href="http://en.wikipedia.org/wiki/IP_protocol">http://en.wikipedia.org/wiki/IP_protocol</a>                                     |
| IR                  | Eng. Infrared. Dio spektra elektromagnetskog zračenja valne duljine od 700nm – 1mm.  | <a href="http://en.wikipedia.org/wiki/Infrared">http://en.wikipedia.org/wiki/Infrared</a>   |
| MAC adresa          | Jedinstveni identifikator fizičkog sučelja mrežnog uređaja.  | <a href="http://en.wikipedia.org/wiki/MAC_address">http://en.wikipedia.org/wiki/MAC_address</a>                                     |
| Port (računalni)    | Računalni portovi (ili samo portovi) su virtualno-logičke podatkovne veze koje mogu koristiti programe za razmjenu podataka direktno, umjesto kroz datoteke ili druge lokacije za privremenu pohranu. Najčešći su TCP i UDP. | <a href="http://en.wikipedia.org/wiki/Computer_port_(software)">http://en.wikipedia.org/wiki/Computer_port_(software)</a>           |
| RS232               | Standard za serijsku komunikaciju između dva uređaja, <i>DTE</i> (eng. <i>Data Terminal Equipment</i> ) i <i>DCE</i> (eng. <i>Data Circuit-terminating Equipment</i> ).  | <a href="http://en.wikipedia.org/wiki/RS232">http://en.wikipedia.org/wiki/RS232</a>   |
| SMS                 | Eng. <i>Short Message Service</i> . Komunikacijska usluga za slanje kratkih poruka, standardizirana u GSM mobilnom komunikacijskom sustavu.  | <a href="http://en.wikipedia.org/wiki/SMS">http://en.wikipedia.org/wiki/SMS</a>   |
| TCP                 | Eng. <i>Transmission Control Protocol</i> . To je protokol koji se nalazi na razini transportnog sloja IP protokola, a omogućuje spor, ali pouzdan prijenos podataka.  | <a href="http://www.tech-faq.com/understanding-cisco-tcpip.shtml">http://www.tech-faq.com/understanding-cisco-tcpip.shtml</a>       |

|            |   |   |
|------------|---|---|
| UDP        | Eng. <i>User Datagram Protocol</i> . To je protokol koji se nalazi na razini transportnog sloja IP protokola, a omogućuje brz, ali nepouzdan prijenos podataka.   | <a href="http://en.wikipedia.org/wiki/User_Datagram_Protocol">http://en.wikipedia.org/wiki/User_Datagram_Protocol</a> |
| uploading  | Slanje podataka sa lokalnog sustava na neki udaljeni sustav.  | <a href="http://en.wikipedia.org/wiki/Uploading">http://en.wikipedia.org/wiki/Uploading</a>                           |
| USB        | Eng. <i>Universal Serial Bus</i> . To je standard za serijsku sabirnicu koja se koristi za spajanje raznih uređaja sa računalom.  | <a href="http://en.wikipedia.org/wiki/USB">http://en.wikipedia.org/wiki/USB</a>                                       |
| web        | Eng. World Wide Web je sustav isprepletenih hipertekst dokumenata kojima se pristupa putem Interneta.   | <a href="http://en.wikipedia.org/wiki/World_Wide_Web">http://en.wikipedia.org/wiki/World_Wide_Web</a>                 |
| web server | Računalni program koji prima HTTP zahtjeve od klijenata, te im odgovara sa HTTP odgovorom (najčešće web stranice i povezani objekti). Također može značiti i računalo koje izvodi upravo opisani program. | <a href="http://en.wikipedia.org/wiki/Web_server">http://en.wikipedia.org/wiki/Web_server</a>                         |
| ZigBee     | To je naziv za IEEE 802.15.4 standard za bežičnu tehnologiju vrlo male potrošnje i male brzine prijenosa podataka.  | <a href="http://www.chrsolutions.com/TelecomGlossary.html">http://www.chrsolutions.com/TelecomGlossary.html</a>       |