

Komunikacija putem modema

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



- Δ Za razvoj sustava koji komuniciraju na daljinu pomoću modema
- Δ Potrebno temeljno poznavanje modema, programiranja u jeziku C te Arduino razvojnog okruženja

Sažetak

Ovaj dokument opisuje temeljni način komunikacije putem modema: Hayes AT sustav naredbi te način na koji se one koriste za uspostavljanje veze između dva modema i prijenos podataka. Potom je dan pregled razvijenog programskog koda i funkcija koje se mogu integrirati u daljnje projekte, kako bi im se omogućilo komuniciranje putem modema.

Sustav je razvijen za „Arduino“ pločicu.

Sadržaj

1. UVOD	3
2. NAČELA IZVEDBE SUSTAVA ZA KOMUNIKACIJU PUTEM MODEMA	4
2.1. Povezivanje modema i računalnog sustava	4
2.2. Hayes AT sustav naredbi	4
2.3. Komandni i podatkovni način rada	5
3. REALIZIRANI SUSTAV ZA KOMUNIKACIJU PUTEM MODEMA	6
3.1. Globalne varijable	6
3.2. Primanje podataka	6
4. ZAKLJUČAK	11
5. LITERATURA	11
6. POJMOVNIK	12

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

Modem (kratica od **modulator-demodulator**) je elektronički uređaj koji omogućuje prijenos digitalnih infomacija analognim putem. Konkretno, on omogućuje prijenos digitalnih informacija između dva računala putem analognih telefonskih veza (bakrenih parica). Stoga je modem povjesno odigrao važnu ulogu u razvoju računalnih mreža, posebno interneta.

U kontekstu projekta „Inteligentna kuća“, komunikacija putem modema zanimljiva je zbog mogućnosti jednostavnog povezivanja fizički udaljenih računalnih sustava putem postojećih veza – analognih telefonskih linija. Takvo povezivanje vrlo je važno za nadzor i upravljanje geografski udaljenim nekretninama te pretvaranje i njih u inteligentne kuće, koje vlasnik može nadzirati iz svog vlastitog doma. Jedini preuvjet za povezivanje jest da udaljena nekretnina ima vlastiti telefonski priključak. Dostupnost ostalih računalnih mreža, posebno interneta, nije bitna, što čini sustav prilagodljivim za vrlo široke primjene.

Ovaj projekt pruža programski kod za „Arduino“, koji se može vrlo jednostavno integrirati u bilo koji drugi program pisan za Arduino, te mu omogućiti komunikaciju putem modema.

2. Načela izvedbe sustava za komunikaciju putem modema

2.1. Povezivanje modema i računalnog sustava

Vezu između modema i računalnog sustava koji upravlja modemom potrebno je ostvariti RSR232 serijskom vezom. To povezivanje izvan je okvira ovog projekta (za implementaciju RS232 serijske veze, vidjeti projekt „RS232“)

2.2. Hayes AT sustav naredbi

Jednom kad je modem povezan sa računalnim sustavom, njime se upravlja putem skupa naredbi koje se zovu *Hayes* naredbe. To je niz jednostavnih tekstualnih naredbi, kojima je moguće naložiti modemu da bira određen telefonski broj, javi se na telefonski poziv, tj. uspostavlja i prekida vezu, resetira se itd.

Zajedničko svim ovim naredbama je to što počinju prefiksom AT (koji dolazi od riječi *attention* - pozor). Neke od važnijih naredbi, korištene u ovom dokumentu, su:

ATH – otvaranje/zatvaranje veze prema telefonskoj centrali (odgovara podizanju/spuštanju slušalice na telefonu)

ATH0 – prekid veze

ATH1 – uspostava veze

ATD – biranje telefonskog broja, odnosno,

ATA – odgovor na ulazni telefonski poziv

AT&L1 – prelazak u *leased line* način rada

Sve naredbe završavaju znakom *carriage return* (CR; ASCII kod 13).

Modem također odgovara određenim odgovorima. Za ovaj dokument važni su sljedeći:

OK – odgovor modema na neke naredbe; označava da je naredba primljena i izvršena

NO CARRIER – signal kojim modem signalizira da je druga strana prekinula uspostavljenu vezu (poklopila slušalicu)

CONNECT – signal kojim modem signalizira da je uspostavio vezu sa drugim modemom. Obično slijedi brzina veze, npr. CONNECT 9600.

2.3. Komandni i podatkovni način rada

Modem, gledan sa strane računalnog sustava koji njime upravlja, može raditi u dva načina rada. Jedan je tzv. komandni mod, u kojem modem sluša naredbe koje mu šalje računalni sustav na koji je spojen te odgovara po potrebi. Drugi je podatkovni način, u kojem modem jednostavno prenosi sve podatke koje primi. Iz komandnog načina modem prelazi u podatkovni po uspostavi veze sa drugim modemom, što je popraćeno porukom CONNECT koju modem šalje „svom“ računalnom sustavu. Iz podatkovnog se u komandni način prelazi tako što računalni sustav modemu pošalje poruku „+++“.

3. Realizirani sustav za komunikaciju putem modema

Prilikom razvoja ovog sustava, nit vodilja bila je da se omogući što jednostavnija upotreba i kasnija nadogradnja programskog koda. Stoga je on organiziran u funkcije čije su se uloge i imena nastojale odabrati tako da budu sama za sebe što informativnija i razumljivija.

3.1. Globalne varijable

Programski kod sadrži nekoliko globalnih varijabli, koje u stvari služe kao zastavice za pojedina stanja u kojima komunikacija, odnosno program, trenutno nalaze.

Varijabla *mode* označava trenutni način rada u kojem se modem nalazi. Deklarirana je kao

```
char mode;
```

ona je jednaka:

- konstanti MODE_COMMAND, kad je modem u komandnom načinu rada
- konstanti MODE_DATA, kad je modem u podatkovnom načinu

Niz *command* je *buffer* u kojeg se sprema primljena naredba, odnosno njezin dio. Deklariran je kao

```
char command[20];
```

3.2. Primanje i slanje podataka

connect_leased()

Prije bilo kakvog prijenosa podataka, potrebno je uspostaviti vezu sa udaljenim modemom. Ako su modemi povezani iznajmljenom linijom (*leased line*), veza se uspostavlja pozivom funkcije *connect_leased()*. Ta funkcija najprije će modemu poslati naredbu za prelazak u mod rada sa iznajmljenom linijom (naredba AT&L1), a potom izdati naredbu za poziv drugom modemu (ATD). Ukoliko modemi potom uspješno uspostave vezu

(modem je odgovorio sa CONNECT 9600), funkcija vraća 1; u suprotnome, vraća 0.

U nastavku je naveden izvorni kod funkcije.

```
int connect_leased()
{
    char rcvd, complete = 0;
    char i;

    send_command("AT&L1");
    if(get_modem_response())
    {
        if(strcmp(command, "OK"))
        {
            return 0;
        }
    }

    send_command("ATD");
    analogWrite(ledPin, 20);
    if(get_modem_response())
    {
        if(strcmp(command, "OK"))
        {
            return 0;
        }
    }

    if(get_modem_response())
    {
        if(strcmp(command, "CONNECT 9600"))
        {
            return 0;
        }
    }
}

return 1;
}
```

recieve_all()

Jednom kad su modemi spojeni, prelazi se u podatkovni način rada u kojem računalni sustavi na udaljenim lokacijama preko modema mogu primati i slati podatke.

Funkcija *recieve_all()* prima ulazne podatke dok je modem u podatkovnom načinu rada i u skladu s njima radi određene operacije, poput pozivanja funkcije za čitanje analognih ulaza na Arduino pločici i slanja pročitanih vrijednosti preko serijske veze na modem.

Funkcija *recieve_all()* prepoznaje ulazne podakte u tri formata:

- za primljen znak „A“ (ASCII 65), *recieve_all()* pozivanjem funkcije *read_analog_input()* čita i preko modema šalje vrijednosti analognih ulaza sa Arduino pločice. Poruke koje šalje oblika su

```
Analog broj_pina = vrijednost_pina
```

npr.

```
Analog 1 = 143
```

- za primljen znak „D“ (ASCII 68), *recieve_all()* pozivanjem funkcije *read_digital_inputs()* šalje vrijednosti pet digitalnih ulaza na Arduino pločici. To su priključci označeni sa brojevima 2 – 7 na Arduino pločici. Poruke koje se šalju su oblika

```
Digital broj_pina = vrijednost_pina
```

npr.

```
Digital 2 = HIGH
```

Vrijednosti digitalnih pinova mogu biti konstante HIGH i LOW.

- Za primljen znak „S“ (ASCII 83), *recieve_all()* očekuje još 2 znaka. Prvi je broj 0 – 5, koji odgovara pinovima označenima sa 8 – 12 na Arduino pločici. Drugi je broj 0 ili 1, koji označava treba li se zadani pin postaviti u visoko ili nisko stanje. Na primjer, niz

```
S10
```

označava da pin 9 na Arduino pločici treba postaviti u nisko stanje.

- Ako su primljena tri znaka „+“ (ASCII 43) za redom, suprotna strana prekida vezu, te će i funkcija *recieve_all()* prekinuti vezu pozivanjem funkcije *end_connection()*.

U nastavku slijedi izvorni kod funkcije.

```
void recieve_all()
{
    int complete;
    int data_value;
    char rcvd_data;

    // -----
    // Data mode
    if(mode == MODE_DATA)
    {
        while(mode == MODE_DATA)
        {
```

```
complete = recieve_data(&rcvd_data);

if(complete)
{
    if(complete == DATA_EXIT)
    {
        end_connection();
    }

    // For letter "A", read all the analog inputs
    if(rcvd_data == 65)
    {
        for(int cnt = 0; cnt <= 5; cnt++)
        {
            read_analog_input(cnt);
        }
    }

    // For letter "D", read digital inputs
    if(rcvd_data == 68)
    {
        for(int cnt = 0; cnt <= 5; cnt++)
        {
            read_digital_input(cnt + 2);
        }
    }

    // For letter "S", set digital outputs
    if(rcvd_data == 83)
    {
        char pin, value;
        while(1)
        {
            if(Serial.available())
            {
                pin = Serial.read();
                pin = pin - 48;
                break;
            }
        }
        while(1)
        {
            if(Serial.available())
            {
                value = Serial.read();
                value = value - 48;
                break;
            }
        }
        set_digital_output(pin+8, value);
    }

}
}
```

end_connection()

Ova funkcija prekida vezu s drugim modemom tako što najprije pošalje niz „+++“, a po odgovoru modema, i naredbu ATH0 za prekid veze. Prototip funkcije je:

```
void end_connection();
```

read_analog_input()

Čita stanje određenog analognog ulaza i šalje ga na modem. Ulaz koji treba pročitati specificira se ulaznim parametrom; to treba biti broj od 0 do 5. Prototip funkcije je:

```
void read_analog_input(char inputPin);
```

read_digital_input()

Čita stanje određenog digitalnog ulaza i šalje ga na modem. Ulazni parametar je broj digitalnog ulaza, broj od 2 do 7. Prototip funkcije je:

```
void read_digital_input(char inputPin);
```

set_digital_output()

Postavlja stanje specificiranog digitalnog izlaza u stanje HIGH ili LOW. Ulazni parametri su broj pina koji se postavlja (od 8 do 13), te vrijednost u koji se postavlja (HIGH ili LOW). Prototip funkcije je:

```
void set_digital_output(int outputPin, int value);
```

4. Zaključak

Napisani programski kod omogućava jednostavno i osnovno korištenje modema za prijenos podataka putem analogne telefonske linije. Programski kod je razvijen za „Arduino“. Prilikom razvoja programa za Arduino, kod se na jednostavan način integrira u ostatak projekta. Osim integracije ovog koda, potrebno je i povezati Arduino RS232 vezom sa modemom.

Budući rad na sustavu i njegovo poboljšanje svakako mora najprije biti usmjereni na ekstenzivno testiranje i povećanje robusnosti sustava, a potom i na proširenje seta podržanih naredbi. Posebno bi zanimljivo sustav bilo prilagoditi za mogućnost korištenja javne telefonske mreže, a ne samo komunikacije preko iznajmljene linije.

5. Literatura

- [1] Jeren, Branko . Pale, Predrag: Modem, predavanja iz predmeta Sustavi za praćenje i vođenje procesa, 2009.
- [2] Wikipedia, the free encyclopedia: Hayes command set, 2009.
http://en.wikipedia.org/wiki/Hayes_AT
- [3] Arduino Reference, <http://arduino.cc/en/Reference/HomePage>

6. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Modem	Uređaj koji omogućuje prijenos digitalnih podataka preko analognih telefonskih linija.	http://en.wikipedia.org/wiki/Modem
Arduino	Razvojno okruženje temeljeno na Atmel Atmega168 mikrokontroleru.	http://arduino.cc/
Hayes AT	Sustav naredbi razvijen za upravljanje modemom.	http://en.wikipedia.org/wiki/Hayes_AT
Komandni način rada	Način rada u kojem modem ulazne podatke tretira kao naredbe.	
Podatkovni način rada	Način rada u kojem modem ulazne podatke modulira na analogni signal i proslijeđuje na analogni izlaz.	