

Savremeni merni sistemi

Laboratorijska vežba 1

Distribuirani merno upravljački
sistem baziran na RS485
serijskom prenosu

Vladimir Rajović, 2008/09

Sadržaj vežbe

- U okviru vežbe se koriste četiri različita modula iz Advantech-ove serije modula za akviziciju podataka ADAM 4000
- Računar na kome je instaliran *LAB Windows/CVI* je serijskom RS232 vezom povezan sa ADAM 4502 konvertorom signala sa RS232 na RS485 i dalje sa zajedničkom serijskom magistralom na koju su povezani ostali moduli.
- Komunikacija se odvija po poludupleks master/slave protokolu. Računar proziva određenu periferiju na osnovu njene adrese.
- Najpre računara šalje zahtev za dobijanje trenutne vrednosti temperature, nakon čega analogni ulazni modul šalje izmereni podatak.
- Zatim računara izvršava akcije vezane za prikazivanje izmerene temperature i signalizaciju prekoračenja dozvoljenog opsega

Advantech ADAM moduli

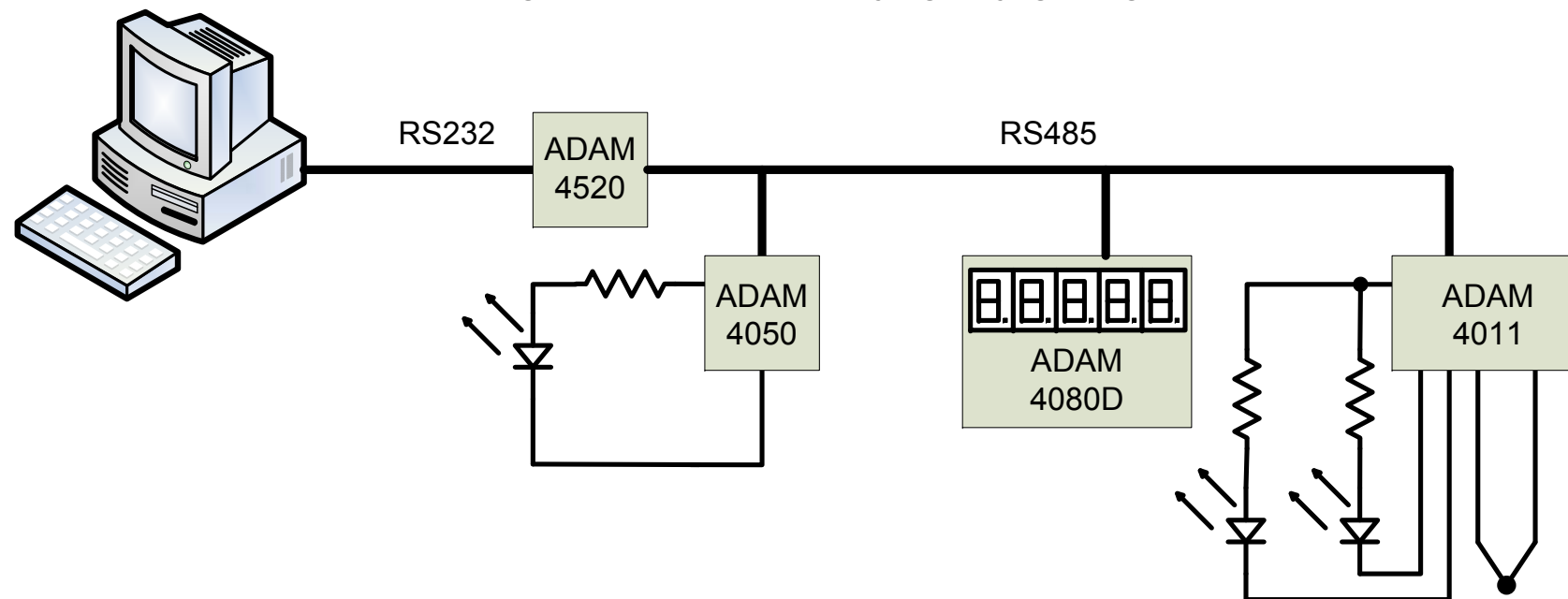
- ADAM 4502 predstavlja konvertor nivoa sa RS232 na RS485
- ADAM 4011 predstavlja analogni ulazni modul koji ima i digitalne izlaze. U konkretnom primeru koristi se na sledeći način:
 - Merena veličina je temperatura, senzor je T tip termopar
 - Koriste se lokalni komparatori koji upoređuju izmerenu veličinu sa vrednostima pragova (gornjeg i donjeg). Izlazi komparatora uključuju dve LE diode
 - Izlazi komparatora nisu lečovani, tj. verno prate promenu temperature.

Advantech ADAM moduli

- ADAM 4080D predstavlja brojački modul koji na sebi ima petocifreni LE displej. U konkretnom primeru koristi se za prikazivanje temperature, tj. koristi se samo displej.
- ADAM 4050 predstavlja digitalni ulazno izlazni modul. U primeru se koristi jedan digitalni izlaz preko koga se uključuje LE dioda

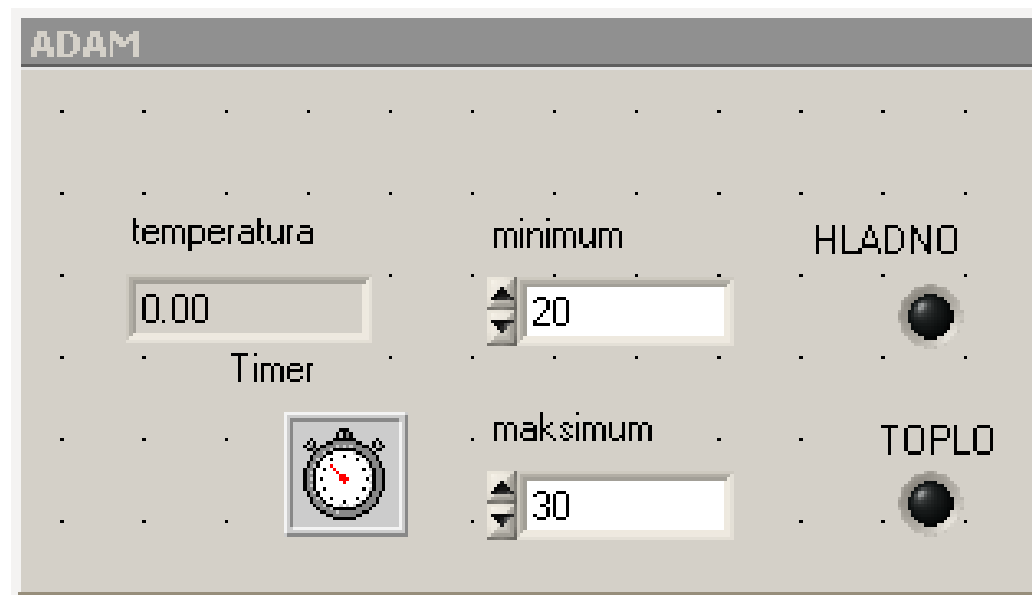
Principijska šema mernog sistema

- Merena veličina je temperatura koja se meri pomoću termopara.
- Računar:
 - iz korisničkog interfejsa uzima vrednost gornjeg i donjeg praga za temperaturu i šalje ih modulu 4011
 - očitava vrednost temperature sa modula 4011 i prikazuje je na ekranu.
 - očitanu vrednost temperature šalje modulu 4080D za prikaz na LED displeju
 - upoređuje očitanu vrednost temperature sa pragovima i na osnovu toga uključuje LE diode na korisničkom interfejsu, i uključuje / isključuje LE diodu na modulu 4050 (signalizira prekoračenje gornjeg praga)



CVI Aplikacija

- Prvi korak u razvoju aplikacije je kreiranje glavnog korisničkog panela, koji je prikazan na slici.
- Koriste se tri kontrole tipa Numeric, jedna kao indikator (za prikazivanje trenutne temperature), ostale dve kao ulazi (donji i gornji prag)
- Dve kontrole tipa LED koriste se za indicaciju da je temperatura previše niska ili previše visoka
- Kontrola Timer kontroliše izvršavanje programa, sa periodom 1.5s



Inicijalizacija serijskog porta

- Osnovni parametri serijske komunikacije su:
 - Baud rate = 9600
 - Format podatka: 8 bita, bez parnosti, 1 stop bit
 - Timeout – 1 sec
 - Komunikacioni port COM2
- Ove veličine se podešavaju pomoću funkcije `OpenComConfig()` kojom se istovremeno i otvara port:

```
OpenComConfig (2, "COM2", 9600, 0, 8, 1, 512, 512);
```

- Po izlasku iz aplikacije port je potrebno zatvoriti komandom `CloseCom()`

Periodično merenje - sampling

- Timer ima dodeljenu Callback funkciju u okviru koje se ubacuju specifične funkcije za serijsku komunikaciju iz RS232 biblioteke

```
int CVICALLBACK otkucaj (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    int promena_temp;
    char string[3];
    switch (event)
    {
        case EVENT_TIMER_TICK:
            //brise sve dosadasnje odzive kao nebitne, radi jednostavnosti programa
            FlushInQ(5);

            CmtGetLock(lock); //zabrana prebacivanja u drugi thread, da se ne promeni promenljiva promena
            promena_temp=promena;
            CmtReleaseLock(lock); //dozvola prebcivanja u drugi thread

            switch (promena_temp)
            {
                case 0:
                    //ocitavanje analognog ulazam format #AA
                    strcpy(send_data, "#03\r");
                    ComWrt (5, send_data, 4);

                    //odziv ima format >+0xy.zw<cr> //
                    ComRdTerm (5, read_data, 9, '\r'); //sa porta 5 se u niz read_data cita 9 bajtova, sa ocekivanjem
                    //9. bajt bude <CR>

                    //prebacivanje niza read_data od cifre x u broj u floating point predstavi
                    temperatura = strtod (&read_data[3], &garbage);

                    //prikaz na UI
                    SetCtrlVal (panelHandle, PANEL_TRENUTNA, temperatura);
            }
    }
}
```


Komande za analogni modul 4011

- #AA<cr>
 - zahtev za očitavanjem analognog ulaza (odgovor je >+0xy.zw<cr>)
- \$AA2<cr>
 - upit o statusu (odgovor je !AA106000 <cr>; 10 za T tip termopar, 06 se odnosi na brzinu komunikacije, 00 inženjerske jedinice)
- @AADOYY<cr>
 - postavljanje digitalnih izlaza, YY=00,01,02,03
- @AALO+0XY.00<cr>
 - definisanje donjeg praga komparatora
- @AAHI+0XY.00<cr>
 - definisanje gornjeg praga komparatora
- @AAEAT<cr>
 - definisanje tipa alarma na modulu -> M trenutni, L lečovani

Komande za brojački modul 4080

- \$AA82<cr>
 - podešavanje da prikazuje podatke koje dobije sa računara
- \$AA2<cr>
 - upit o statusu (odgovor je !AA500600; 50 tip modula, 06 se odnosi na brzinu komunikacije, 00 ulazni mod)
- \$AA9XXXX.X<cr>
 - slanje podataka na displej

Komande za digitalni modul 4050

- #AA00DD<cr>
 - Istovremeni upis na sve kanale
- \$AA2<cr>
 - upit o statusu (odgovor je !AA400600; 40 tip modula, 06 se odnosi na brzinu komunikacije, 00 čeksuma)
- #AA1X0Y<cr>
 - upis na pojedinačan kanal, X broj kanala, Y stanje kanala

Multithreading izvršavanje

- Multithreading (kvaziparalelno izvršavanje) se u CVI može implementirati na više načina
- U primeru se koristi asinhroni tajmer (nije vidljiv u UIR fajlu), čija se CALLBACK funkcija izvršava “paralelno” sa User Interface-om:

NewAsyncTimer (0.5, -1, 1, DrugiOtkucaj, 0);// perioda 0.5s, funkcija DrugiOtkucaj

```
int CVICALLBACK DrugiOtkucaj (int reserved, int timerId, int event, void *callbackData, int eventData1, int eventData2)
{
    int promena_temp;

    CmtGetLock(lock);
    promena_temp=promena;
    CmtReleaseLock(lock);

    if (promena_temp==0)
    {
        CmtGetLock(lock);
        GetCtrlVal(panelHandle, PANEL_MINIMUM, &minimum);
        CmtReleaseLock(lock);
        CmtGetLock(lock);
        if (minimum!=minimum_stari)
        {
            promena=1;
            minimum_stari=minimum;
        }
    }
}
```

Kontrola promenljivih prilikom multithreadinga

- Ako više threadova koristi istu promenljivu potrebno je obezbediti da dok jedan thread manipuliše sa datom promenljivom drugi thread ne počne da se izvršava
- Ovo se rešava definisanjem “ključa”:
`CmtNewLock (NULL, 0, &lock); //pokazivac na kljuc je promenljiva lock`
- Pri izlasku iz aplikacije ključ je potrebno uništiti:
`CmtDiscardLock (lock);`

Kontrola promenljivih prilikom multithreadinga

- Pre nego što će dati thread da koristi neku kritičnu promenljivu potrebno je zabraniti da neki drugi thread počne da se izvršava
- Nakon što dati thread završi sa datom promenljivom potrebno je dozvoliti mogućnost da neki drugi thread počne da se izvršava

```
CmtGetLock(lock);
```

```
//zabrana prebacivanja u drugi thread, zbog promenljive "promena"
```

```
promena_temp=promena;
```

```
//promena_temp je inače sigurna, ne koristi se u više threadova i dalje se radi sa njom
```

```
CmtReleaseLock(lock);
```

```
//dozvola prebacivanja u drugi thread
```