



# UVOD

Cilj laboratorijske vežbe je da se demonstriraju mogućnosti upravljanja funkcijama programabilnog instrumenta preko IEEE-488 instrumentacionog komunikacionog interfejsa. Interfejs IEEE-488 naziva se i instrumentaciona magistrala opšte namene (*General Purpose Instrument Bus*) i skraćeno se označava sa GPIB. S obzirom da PC računar standardno nije opremljen IEEE-488 interfejsom, on se PC računaru dodaje tako što se u neki od slobodnih ISA ili PCI konektora na osnovnoj ploči računara instalira odgovarajuća IEEE-488 kartica. Druga mogućnost je da se povezivanje IEEE-488 instrumentacije sa PC računarom vrši preko nekog od standardnih komunikacionih interfejsa PC računara (na raspolaganju su serijski RS-232, paralelni port i USB interfejs) posredstvom konvertora standardnog komunikacionog interfejsa PC računara u IEEE-488 interfejs. Ova druga mogućnost iskorišćena je za povezivanje multimetra HP 34401A proizvodnje Agilent sa PC računarom. Veza je ostvarena posredstvom IEEE-488/USB konvertora 82357A čiji je proizvođač takođe Agilent. Izgled IEEE-488/USB konvertora prikazan je na slici 1.



Slika 1. Izgled USB / GPIB interfejsa, Agilent 82357 A

Sa strane PC računara komunikacija sa multimetrom ostvaruje se korišćenjem standardnih IEEE-488 funkcija i komandi posredstvom drajvera koji je isporučen uz USB/GPIB konvertor 82357A. Neizmenjen program mogao bi se koristiti za upravljanje multimetrom koji je sa PC računarom povezan posredstvom interfejsne kartice, s tim što bi u tom slučaju bilo potrebno koristiti drajvere za instaliranu karticu.

U ovom tekstu opisan je postupak povezivanja i podešavanja hardvera i instalacije i podešavanja softvera za upravljanje programabilnom instrumentacijom preko IEEE-488 interfejsa. Upravljanje programabilnim multimetrom HP 34401A, koji se koristi u ovoj vežbi, demonstrirano je preko dva primera. U prvom primeru multimetar se, posredstvom PC računara, koristi za merenje jednosmernog napona koji je doveden na odgovarajuće priključke instrumenta. U ovom primeru detaljno je opisan postupak izrade programa aplikacije na PC računaru. Ovaj program realizovan je korišćenjem National Instruments programskog paketa CVI koji radi pod Windows operativnim sistemom. Drugi primer odnosi se na nešto kompleksniju primenu programabilnog multimetra HP 34401A. Naime, preko ovog primera dato je jedno rešenje periodičnog očitavanja vrednosti

amplitude signala koji se dovodi na multimeter i prikazivanje talasnog oblika ovog signala na PC računaru.

## INSTALACIJA HARDVERA

Na zadnjoj strani HP 34401A multimetra je potrebno identifikovati GPIB priključak i na njega povezati odgovarajući kraj USB/GPIB konvertora 82357A. Drugi kraj USB/GPIB konvertora 82357A je potrebno povezati na USB priključak PC računara. Napajanje HP 34401A multimetra vrši se preko mrežnog priključka, dok se napajanje USB/GPIB konvertora 82357A vrši preko USB magistrale. Ovim je proces instalacije hardvera završen.

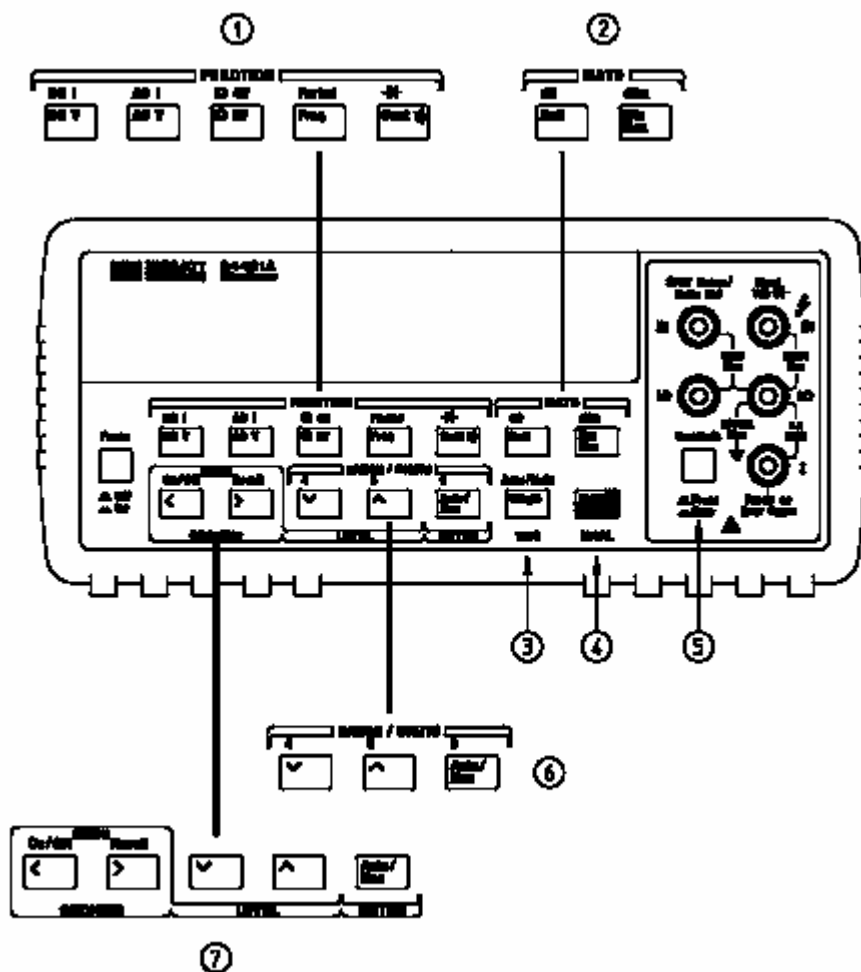
## INSTALACIJA SOFTVERA

Instalacija potrebnog softvera obavlja se sledećim redom:

1. Instalirati National Instruments CVI verziju 7. 1 (programsko okruženje u kojem će se vršiti programiranje softvera za kontrolu multimetra).
2. Instalirati odgovarajuće drajvere koji će omogućiti kontrolu multimetra upotrebom USB/GPIB interfejsa. Ovo je potrebno uraditi pokretanjem fajla **iolibs\_suite\_14\_0\_7202\_1.exe**, koji je ustvari instalacioni fajl za Agilent VISA podršku (*Virtual Instrument Software Architecture*) koja omogućuje pristup USB kontroleru, i odgovarajuću konverziju GPIB u USB protokol.
3. Instalirati HP 34401A programming drajver pokretanjem fajla **driver\_vxipnp\_34401\_a\_02\_14.exe**. Nakon ovoga koraka fajlovi koje sadrži navedeni paket biće instalirani u direktorijumu C:\Program Files\Visa\winnt.

# PODEŠAVANJE HARDVERA

Da bi bilo moguće upravljati HP 34401A multimetrom potrebno je izvršiti odgovarajuću konfiguraciju samog instrumenta. Na slici 2 prikazan je prednji panel multimetra.



Slika 2. Prednji panel instrumenta

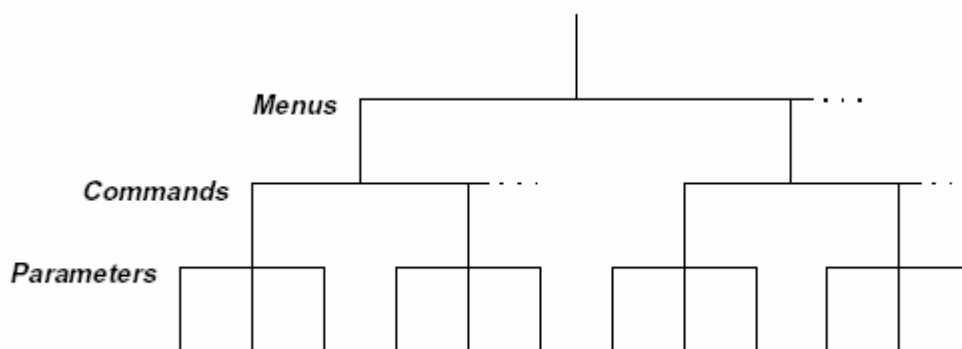
Grupa tastera 7 omogućuje kretanje kroz stablo menija i selektovanje odgovarajuće funkcije koja se želi podesiti. Za upravljanje PC računom korišćenjem GPIB instrumentacionog interfejsa potrebno je u okviru menija selektovati prvo grupu **Input / Output MENU** a potom postaviti vrednosti sledećih podgrupa:

1. **HP-IB ADDR** (adresa koja se dodeljuje instrumentu u procesu upravljanja), u primeru softvera koji će kasnije biti komentarisano usvojeno je da instrument ima adresu 15, pa je dakle na ovu vrednost potrebno postaviti ovo polje. Ovde je potrebno reći da se za naziv

IEEE-488 interfejsa pored skraćenice GPIB koristi i skraćenica HP-IB, prema firmi Hewlett-Packard koja je uvela ovaj standard.

2. **INTERFACE** (način na koji se kontroliše multimeter, moguće varijante su HP-IB i RS-232 jer pored IEEE-488 multimeter raspolaže i RS-232 interfejsom) potrebno je selektovati **HP-IB** varijantu interfejsa pošto se koristi GPIB magistrala.
3. **LANGUAGE** (selektuje se jezik interfejsa moguće varijante su SCPI, HP 3478, ili Fluke 8840/42) potrebno je izabrati **SCPI**.
4. **BAUD RATE** i **PARITY** podgrupe nema potrebe konfigurisati jer one imaju značaja samo ako se koristi serijski **RS-232** interfejs.

Ulazak u meni se ostvaruje selektovanjem tastera **Shift , Menu On/Off**. Za izvršavanje komande menija potrebno je selektovati **Enter**. Organizacija menija je prikazana na slici 3.



Slika 3. Organizacija menija

## PODEŠAVANJE SOFTVERA

Pošto se koriste VISA drajveri firme Agilent, potrebno je izvršiti odgovarajuća prilagođenja CVI programskog okruženja, kako bi kompajlirani fajlovi sa korektnim bibliotekama koje linkuju mogli da preko odgovarajućih drajvera pistepe instrumentu. Koraci koje je potrebno ostvariti u tom smislu su sledeći:

1. Iz direktorijuma **c:\Program Files\VISA\winnt\include\** potrebno je prekopirati sledeće fajlove  
**visa.h**  
**visaext.h**  
**visatype.h**  
**vpptype.h**  
u direktorijum **c:\Program Files\National Instruments\CVI71\include\** i na taj način prepisati postojeće fajlove iz tog direktorijuma.

2. Iz direktorijuma **c:\Program Files\VISA\winnt\bin\** potrebno je prekopirati sledeće fajlove  
**hp34401\_32.dll**  
**hpgpvx32.dll**  
 u direktorijum **c:\Program Files\National Instruments\CVI71\bin\**.
3. U okviru direktorijuma **c:\Program Files\National Instruments\CVI71\toolslib\** potrebno je napraviti direktorijum **Hp34401** i u taj direktorijum prekopirati sve fajlove iz **c:\Program Files\VISA\winnt\HP34401\** direktorijuma.
4. Otvoriti CVI programsko okruženje, u opciji **Library** (padajući meni, potrebno je selektovati polje **Customize**. Selektovati **Browse** i kretati se kroz direktorijume dok se u polju **Look In** ne nađe **c:\Program Files\National Instruments\CVI71\toolslib\ Hp34401** direktorijum. Selektovati fajl **hp34401.fp** (fajl koji dolazi sa hp34401 programming drajverom). Na ovaj način je biblioteka hp34401 uključena u CVI programsko okruženje.
5. Potrebno je napraviti još statičku biblioteku za programming driver hp34401. Napraviti projekat **hp34401.prj** u CVI programskom okruženju. Taj projekat snimiti u direktorijum **c:\Program Files\National Instruments\CVI71\toolslib\ Hp34401** (gde se nalaze svi fajlovi Hp34401 programming drajvera). U projekat uključiti:  
**hp34401.c**  
**hp34401.h**  
**hp34401.fp**  
 selektovanjem redom opcija iz **File-Open, Source, Include, Function Tree** i biranjem navedenih fajlova iz **c:\Program Files\National Instruments\CVI71\toolslib\ Hp34401** direktorijuma. Potom je potrebno iskompajlirati statičku biblioteku **hp34401.lib**, u meniju **Build** selektovati **CreateStaticLibrary**, posle izvršenja ove komande u direktorijumu **c:\Program Files\National Instruments\CVI71\toolslib\ Hp34401**, će se pojaviti fajl **hp34401.lib**.  
 U okviru direktorijuma **c:\Program Files\National Instruments\CVI71\toolslib\ Hp34401** je potrebno napraviti direktorijum **msvc**, dakle **c:\Program Files\National Instruments\CVI71\toolslib\HP34401\msvc\** i u taj direktorijum prekopirati **hp34401.lib**.

Ovim je završeno konfigurisanje softvera i može se preći na izradu programa za kontrolu multimetra.

# PRIMER 1

Program koji će biti realizovan predviđen je za jednostavnu kontrolu multimetra, merenje i prikaz vrednosti jednosmernog napona koji je povezan na odgovarajuće priključke multimetra. Sve funkcije koje služe za upravljanje instrumentom (a koje su podržane od strane programming drivera) su opisane u help fajlu koji je instaliran zajedno sa programming driver-om. Help fajl se nalazi u **c:\Program Files\VISA\winnt\HP34401\** direktorijumu (hp34401.hlp).

Pokrenuti CVI, u **File** meniju izabrati **New, WorkSpace**. Imenovati ovaj fajl sa npr. **Voltmetar** i snimiti ga u direktorijum npr. **c:\Program Files\National Instruments\CVI71\Projects\Voltmetar\**. Snimiti sve fajlove, **Untitled.prj** preimenovati u **Voltmetar.prj**.

Otvoriti novi **UserInterface** iz **File** menija. Duplim klikom na oblast panela moguće je selektovati njegove attribute. Podesiti **Panel Title** atribut kao **Voltmetar- HP 34401**. U meniju **Code** selektovati **Generate, All Code...** posle čega će biti potrebno snimiti sve fajlove (to okruženje automatski radi, potrebo je samo imenovati fajlove kao **Voltmetar**). Nakon toga u direktorijumu **c:\Program Files\National Instruments\CVI71\Projects\Voltmetar\** se nalaze fajlovi:

**Voltmetar.c**

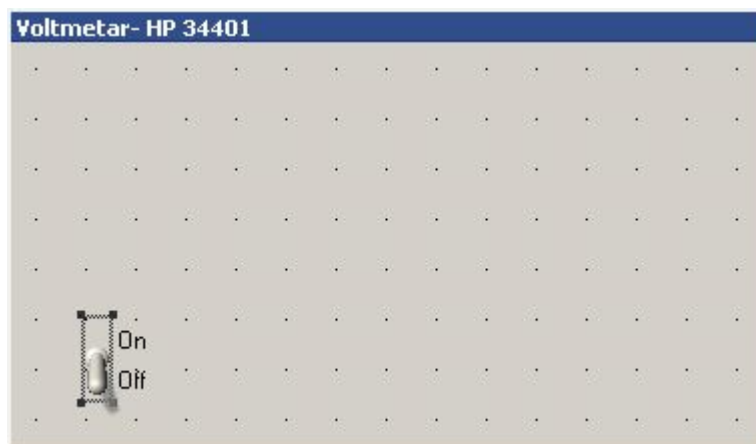
**Voltmetar.cws**

**Voltmetar.h**

**Voltmetar.prj**

**Voltmetar.uir**

U okviru glavnog panela potrebno je dodati **BinarySwitch** elemenat kojim će se inicijalizovati instrument (uspostaviti daljinsku kontrola nad multimetrom). U okvir menija **Create** izabrati **BinarySwitch** i postaviti ga na površinu panela, slika 4.



*Slika 4. Panel virtuelnog instrumenta sa kontrolom za inicijalizaciju*

Posle postavljanja bilo kog elementa na površinu panela potrebno je izgenerisati **Callback** funkcije koje se pozivaju kad god se klikne na odgovarajući element koji se nalazi na panelu. Dvoklikom na novopostavljeni element mogu se selektovati njegovi atributi, u odeljku za **Callback Function** potrebno je ukucati ime funkcije npr. **PowerOn**. Posle toga je potrebno zabeležiti promene, Save opcija i generisati callback funkciju, desni klik na **BinarySwitch** element i selekcija **Generate Control Callback**, posle čega će se odgovarajuća funkcija pojaviti u **Voltmetar.c** fajlu. Posle svake ovakve procedure potrebno je snimiti odgovarajuće fajlove u kojima je nastala promena.

Izgled generisane callback funkcije je dat na slici 5.

```
int CVICALLBACK PowerOn (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            break;

    }
    return 0;
}
```

*Slika 5. Callback funkcija za Binary Switch*

U ovom obliku funkcija ne obavlja nikakav posao, potrebno je dodati elemente koji bi omogućili da se instrument inicijalizuje u stanju **ON** prekidača, odnosno prekine rad sa njim u stanju **OFF** prekidača.

Kompletan izgled funkcije dat je na slici 6.

```
int CVICALLBACK PowerOn (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2){
    int state;
    switch (event)
    {
        case EVENT_LEFT_CLICK:
            GetCtrlVal (panelHandle, PANEL_BINARYSWITCH, &state);
            switch (state) {
                case 0 :
                    hp34401_close (instrumentHandle); break;
                case 1 :
                    hp34401_init ("GPIB::15", VI_TRUE, VI_TRUE, &instrumentHandle);break;
            }break;
    }
    return 0;}
```

*Slika 6. Callback funkcija za Binary Switch (kompletan izgled)*



Funkcija:

```
int GetCtrlVal (int panelHandle, int controlId, void *value);
```

je definisana u okviru CVI programskog okruženja, njena svrha je da vrati trenutnu vrednost elementa na panelu koji služi za kontrolu. U ovom slučaju je to **BinarySwitch** koji ima dva stanja (ON Value, OFF Value) koji su integer tipa i koji se mogu podesiti u okviru podešavanja svojstva kontrolnog elementa. Kada je u stanju **ON** funkcija **GetCtrlVal** će kao vrednost na koju pokazuje **&state** vratiti 1 a kada je prekidač u stanju **OFF** funkcija **GetCtrlVal** će vratiti vrednost 0.

Funkcija **GetCtrlVal** ima još dva parametra, to su **panelHandle**, i **controlID**. Prvi parametar je vezan za glavni panel na kojem se nalazi kontrola (**BinarySwitch**), dok je drugi parametar identifikator kontrole koji se dobija kao **PANEL\_IMEKONTROLE**, gde je **IMEKONTROLE** zapravo constant name u okviru svojstva kontrole (svojstva kontrole se vide desnim klikom na oblast kontrolnog elementa u okviru glavnog panela, i selektovanjem **Edit Control** u padajućem meniju).

Funkcija:

```
ViStatus_VI_FUNC hp34401_init(ViRsrc InstrDesc, ViBoolean id_query, ViBoolean reset, ViPSession vi);
```

je definisana u okviru HP 34401A programming drajvera i služi za inicijalizaciju softverske konekcije sa instrumentom, stoga verifikuje da li je instrument u sistemu veze sa PC računarnom. Opciono može sadržati elemente koji služe za resetovanje instrumenta i njegovo postavljanje u poznato stanje.

Parametar **InstrDesc** je opis instrumenta. Npr. **GPB::15::INSTR**, gde je broj u sredini adresa koja je ukucana u instrment prilikom podešavanja hardvera na početku rada.

Parametar **id\_query** služi za slanje identifikacionog niza. Ako je **VI\_TRUE** niz se šalje ako je **VI\_FALSE** niz se ne šalje.

Parametar **doReset** služi za izvršenje reseta nad instrumentom, ako je **VI\_TRUE** instrument će se resetovati, ako je **VI\_FALSE** instrument se neće resetovati.

Parametar **vi** je handle instrumenta, on dobija vrednost **VI\_NULL** ako se desila geška prilikom inicijalizacije.

Funkcija inicijalizacije vraća vrednost tipa **ViStatus** i povratna vrednost govori o tome da li je inicijalizacija pravilno izvršena. Ako je ta vrednost **VI\_SUCCESS** inicijalizacija je izvršena bez greške a ako nije desila se greška.

Gore napisan primer funkcije ne vodi računa o tome da li je došlo do greške ili ne. Međutim, podatak da je došlo do greške je veoma koristan i tu funkciju treba uključiti u program. Jedno rešenje koje vodi računa o rezlatu izvršene akcije je prikazano na slici 7.

```

int CVICALLBACK PowerOn (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2){
    int state;
    switch (event)
    {
        case EVENT_LEFT_CLICK:
            GetCtrlVal (panelHandle, PANEL_BINARYSWITCH, &state);
            switch (state) {
                case 1 :
                    errStatus =hp34401_close (instrumentHandle);
                    if (VI_SUCCESS > errStatus){
                        MessageBoxA(NULL,"Unable to close instrument","Error",MB_OK);
                    }
                    break;
                case 0 :
                    errStatus =hp34401_init ("GPIB::15", VI_TRUE, VI_TRUE, &instrumentHandle);
                    if (VI_SUCCESS > errStatus){
                        MessageBoxA(NULL,"Unable to open instrument","Error",MB_OK);
                    }
                    break;
            }break;
    }
    return 0;
}

```

*Slika 7. Callback funkcija za Binary Switch (sa ispitivanjem statusa)*

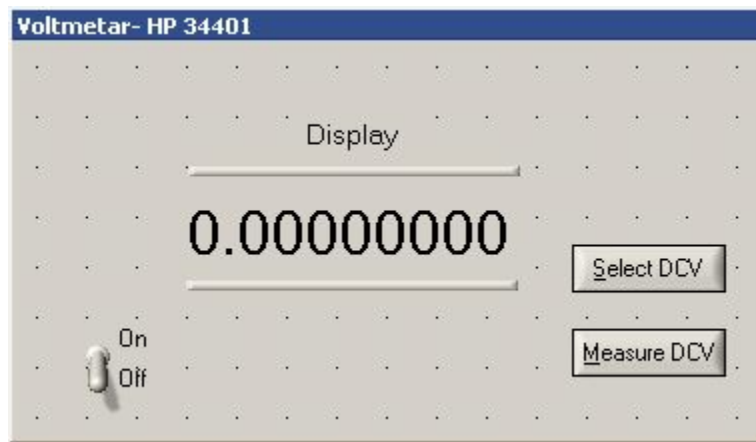
Funkcija:

```
ViStatus _VI_FUNC hp34401_close(ViSession vi);
```

prekida softversku vezu sa instrumentom i dealocira resurse sistema (memorije) dodeljene za upravljanje instrumentom. Ova funkcija je takođe definisana u okviru HP 34401A programming drajvera, ima jedan parametar tipa **ViSession**, onaj isti koji je vratila funkcija **hp34401\_init**.

Pošto dati program treba do omogućiti kontrolu multimetera tako da on meri jednosmerni napon potrebno je dodati još i kontrolu za prebacivanje instrumenta u režim merenja napona i kontrolu za zadavanje merenja napona, kao i displej za prikazivanje izmerene vrednosti.

Izgled panela virtuelnog instrumenta sa pomenutim kontrolama je prikazan na slici 8.



Slika 8. Panel virtuelnog instrumenta sa svim predviđenim kontrolama

Callback kontrole Select DCV, funkcija SelectDCV je prikazana na slici 9.

```
int CVICALLBACK SelectDCV (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int digits ;
    double sampleRate=0.02;
    ViReal64 manualRange;
    switch (event)
    {
        case EVENT_COMMIT:
            hp34401_confVoltDc(instrumentHandle);
            hp34401_voltDcRangAuto(instrumentHandle,VI_TRUE);
            hp34401_voltDcNplc(instrumentHandle, sampleRate);
            break;
    }
    return 0;
}
```

Slika 9. Callback funkcija za Select DCV

Funkcija:

```
ViStatus _VI_FUNC hp34401_confVoltDc(ViSession vi);
```

konfigurira instrument da meri DC napone, ulazni parametar je onaj isti koji je vratila funkcija **hp34401\_init** (softverska veza sa instrumentom).

Funkcija:

```
ViStatus _VI_FUNC hp34401_voltDcRangAuto(ViSession vi, ViBoolean autoRange);
```

konfiguriše opseg merenja u režimu auto. Parametar **autoRange** treba postaviti da ima vrednost **VI\_TRUE**.

Funkcija:

```
ViStatus _VI_FUNC hp34401_voltDcNplc(ViSession vi, ViReal64 nplc);
```

konfiguriše brzinu semplovanja ulaznog napona kroz vreme integracije signala u odnosu na broj ciklusa mrežnog napona. Moguće vrednosti u 0.02T, 0.2T, 1.0T, 10.0T, 100.0T, gde je T perioda mrežnog napona. Parametar **nplc** stoga uzima vrednosti 0.02, 0.2, 1.0, 10.0, 100.0.

U ovom primeru je uzeta maksimalna brzina semplovanja (minimalno vreme integracije) od 0.02T.

Callback kontrole Mmeasure DCV, funkcija MeasureDCV je prikazana na slici 10.

```
int CVICALLBACK MeasureDCV (int panel, int control, int event,
                             void *callbackData, int eventData1, int eventData2)
{
    ViReal64 value;
    char str[100];
    ViInt32 measurementNumber;
    int triggerType;
    switch (event)
    {
        case EVENT_COMMIT:

            hp34401_timeOut (instrumentHandle, 100000);
            hp34401_autoZero(instrumentHandle, VI_TRUE);
            hp34401_trigSour(instrumentHandle,2);
            hp34401_sampCoun(instrumentHandle,1); //Broj semplova=1
            hp34401_trigCoun(instrumentHandle,1); //Broj trigeru =1
            hp34401_trigDelAuto(instrumentHandle, VI_FALSE); //Auto Delay

            hp34401_initImm(instrumentHandle);//Pocetak merenja
            hp34401_fetc_Q(instrumentHandle, &value, &measurementNumber);

            Fmt (str, "%f", value);
            SetCtrlVal (panelHandle, PANEL_TEXTMSG_2, str);

        }
    return 0;
}
```

*Slika 10. Callback funkcija za Measure DCV*

Funkcija:

```
ViStatus _VI_FUNC hp34401_timeOut(ViSession vi, ViInt32 timeOut);
```

postavlja minimalno granično vreme za I/O manipulacije od strane drajvera izraženo u milisekunadama. Ako se prekorači ovo vreme doći će do greške u komunikaciji, tako da je poželjno posatviti ovu vrednost na što veći nivo.

Funkcija:

```
ViStatus _VI_FUNC hp34401_autoZero(ViSession vi, ViBoolean ZeroAuto);
```

uključuje ili isključuje autoZero opciju za DC merenja. Ova mogućnost je ostavljena u slučaju da postoji određeni DC naponski nivo na ulazu koja je superponiran mernom signalu. Instrument će (ukoliko je ova opcija uključena) pre svakog merenja izvršiti jedno merenje u prazno (otkačeni ulazi) i od izmerene vrednosti (kada su ulazi za merni signal priključeni) oduzeti vrednost koja je dobijena u zero mode čime se dobija konačan rezultat merenja.

Funkcija:

```
ViStatus _VI_FUNC hp34401_sampCoun(ViSession vi, ViInt32 sampCoun);
```

specificira broj čitanja po jednom trigeru. Pošto se u ovom slučaju radi o jednom merenju potrebno je da parametar koji specificira broj čitanja bude jedan (**sampCoun=1**).

Funkcija:

```
ViStatus _VI_FUNC hp34401_trigCoun(ViSession vi, ViInt32 trigCoun);
```

specificira broj trigeru. Pošto se u ovom slučaju radi o jednom merenju potrebno je da parametar koji specificira broj trigeru bude jedan (**trigCoun=1**).

Funkcija:

```
ViStatus _VI_FUNC hp34401_trigDelAuto(ViSession vi, ViBoolean trigDelAuto);
```

omogućuje postavljanje automatskog kašnjenja trigeru. **trigDelAuto=VI\_TRUE** automatsko kašnjenje uključeno, u suprotnom kašnjenje je isključeno.

Funkcija:

```
ViStatus _VI_FUNC hp34401_trigSour(ViSession vi, ViInt16 trigSour);
```

vrši izbor trigerera. Postoje tri mogućnosti za izbor trigerera, 0-Bus,1-Ext Trig BNC Connector, 2-Immediate. Ovde je uzeto da se instrument trigeruje odmah po početku merenja (interno).

Funkcija:

```
ViStatus _VI_FUNC hp34401_initImm(ViSession vi);
```

postavlja instrument u stanje wait for trigger, pošto je ovde izabrano da se instrument trigeruje momentalno, nije potrebn voditi računa o trigeru.

Funkcija:

```
ViStatus _VI_FUNC hp34401_fetc_Q(ViSession vi, ViReal64 readings[ ], ViPInt32  
numReadings);
```

vrši preuzimanje očitanih vrednosti iz memorije instrumenta. Parametar **numReadings** označava koliko je vrednosti smešteno u **readings[ ]** niz (u ovom slučaju je broj trigerera i broj čitanja po trigeru bio jedan pa je **numReadings=1**. Ovaj podatak vraća instrument.

Funkcija:

```
int Fmt (void *target, char *formatString, source1, . . . , sourcen)
```

je deo CVI okruženja koja vrši konverziju podatka iz bilo kog formata u string tip podataka koji je pogodan za ispisivanje na displeju.

Funkcija:

```
int SetCtrlVal (int panelHandle, int controlID, ...);
```

je deo CVI okruženja koja vrši postavljanje kontrole na panelu onako kako mi to specificiramo parametrima u okviru poziva ove funkcije.

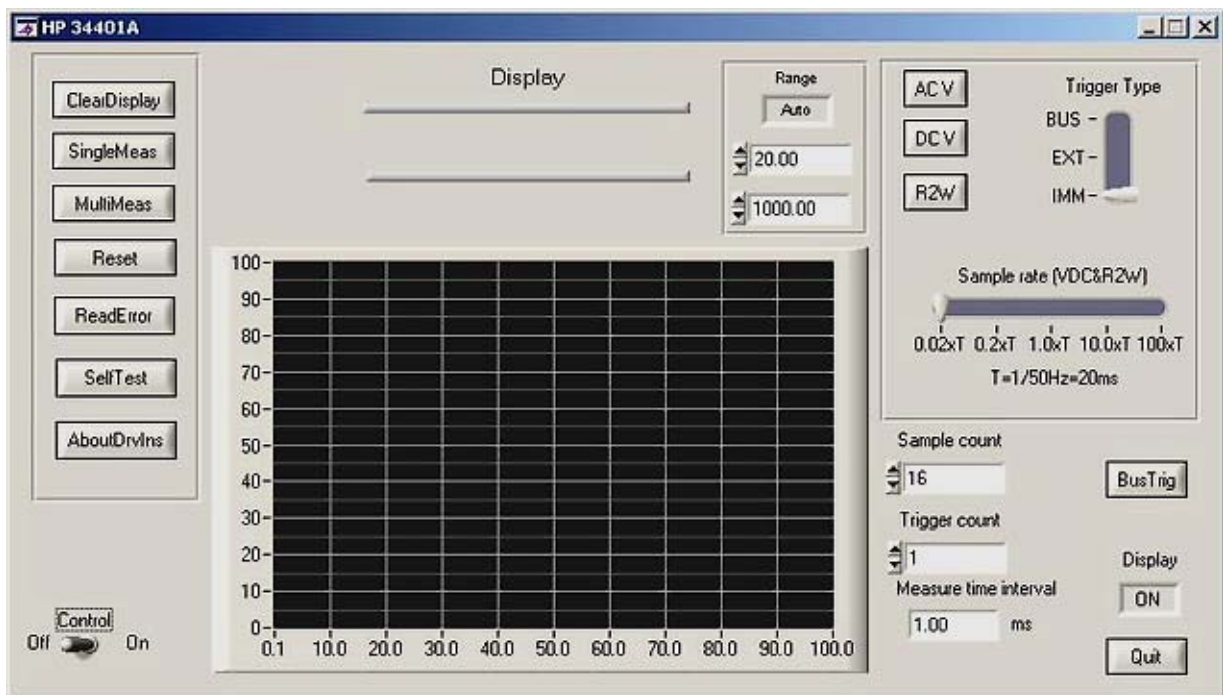
Ovim je završena realizacija jednostavnog programa za kontrolu Agilent HP 34401 multimetara.

## PRIMER 2

U ovom primeru ostvarene su sledeće funkcije:

- Merenje jednosmernog napona u režimu pojedinačnih merenja.
- Merenje naizmeničnog napona u režimu pojedinačnih merenja.
- Merenje otpornosti u režimu pojedinačnih merenja.
- Sukcesivno merenje jednosmernog napona u režimu više uzastopnih merenja sa mogućnošću zadavanja broja mernih tačaka i prikaz merenih rezultata tj. prikaz talasnog oblika signala.
- Očitavanje greške iz bafera instrumenta i prikaz odgovarajuće poruke na displeju.
- Očitavanje verzije drajvera instrumenta.
- Zadavanje komande samotestiranja instrumentu.
- Zadavanje komande resetovanja instrumenta.
- Mogućnost brisanja displeja na panelu virtuelnog instrumenta.
- Mogućnost gašenja i paljenja displeja na instrumentu.

Panel virtuelnog instrumenta koji zadovoljava gore navedene zahteve je prikazan na slici 11.



Slika 11. Panel virtuelnog instrumenta u primeru 2

Postoji mogućnost zadavanja različitih vremena integracije ulaznog signala. Instrument poseduje mogućnost izbora jedne od pet predefinisanih vrednosti, koji su umnošci periode mrežnog napona. Ova opcija je označena kao **Sample Rate** na panelu virtuelnog instrumenta. Ako se u režimu sukcesivnih merenja izabere minimalno vreme integracije ulaznog signala dolazi se do brzine semplovanja. Takođe, postoji mogućnost izbora izvora triger: interni triger, eksterni i softverski triger. U svim merenjima je korišćen interni triger instrumenta.

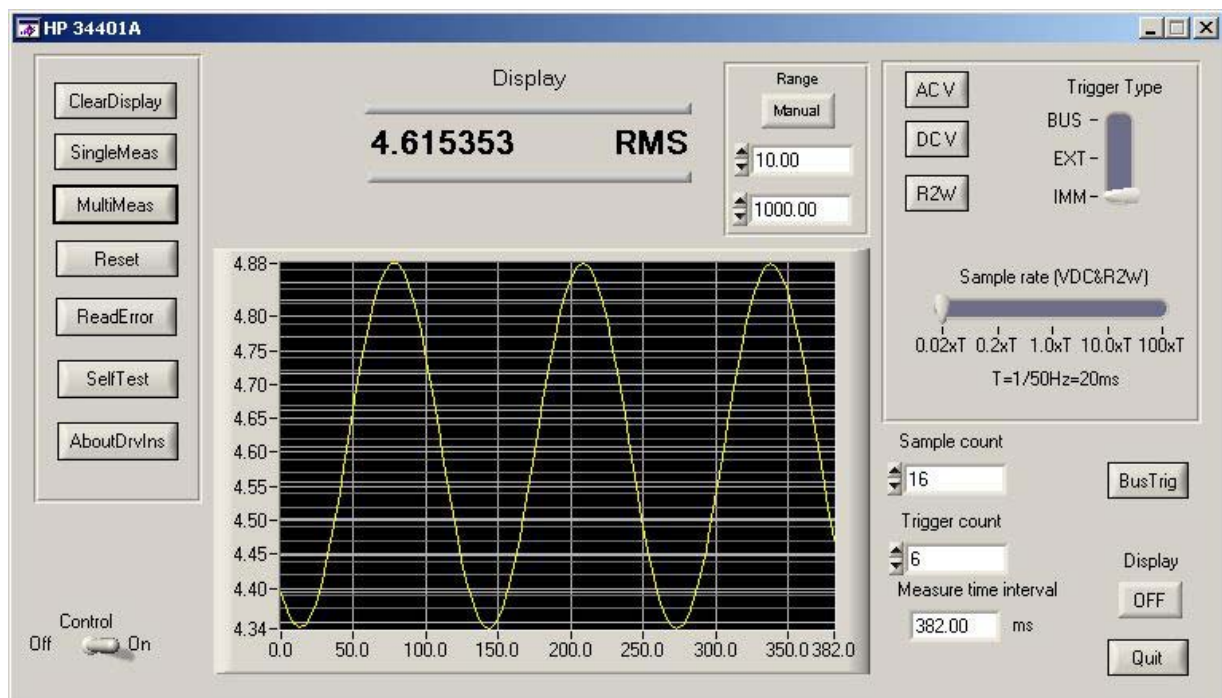
Uvedena je i funkcija podešavanja opsega instrumenta pri merenju napona i otpornosti. Opseg može biti postavljen na **Auto** pri čemu instrument sam određuje mereni opseg u zavisnosti od veličine ulaznog signala koji se snima i **Manual**, gde je opseg određen vrednostima u sastavu text box-ova, prvi text box je zajednički za jednosmerni i naizmenični napon, dok je drugi predviđen za podešavanje opsega pri merenju otpornosti.

U režimu sukcesivnih merenja napona, koji funkcioniše samo za jednosmerne napone, potrebno je ugasiti displej instrumenta i opseg merenja prebaciti na **Manual**, vreme integracije postaviti na minimalno (0.02 T), podesiti **Sample count** i **Trigger count** na željeni broj (broj merenih tačaka se dobija kao  $n = \text{Sample count} \times \text{Trigger count}$ ). Ovim operacijama se postiže maksimalna brzina merenja jer instrument ne troši dodatno vreme na osvežavanje displeja pri svakom merenju i na eventualno prebacivanje opsega ukoliko merna veličina prelazi iz manjeg opsega u veći u toku merenja.

Kontrole **ACV**, **DCV**, **R2W** vrše konfigurisanje instrumenta na merenje odgovarajuće veličine, dok kontrole **SingleMeas** i **MultiMeas** vrše postavljanje instrumenta u režim merenja i očitavaju izmerene vrednosti koje potom prikazuju na displeju odnosno grafiku, ukoliko se radi o režimu sukcesivnih merenja.

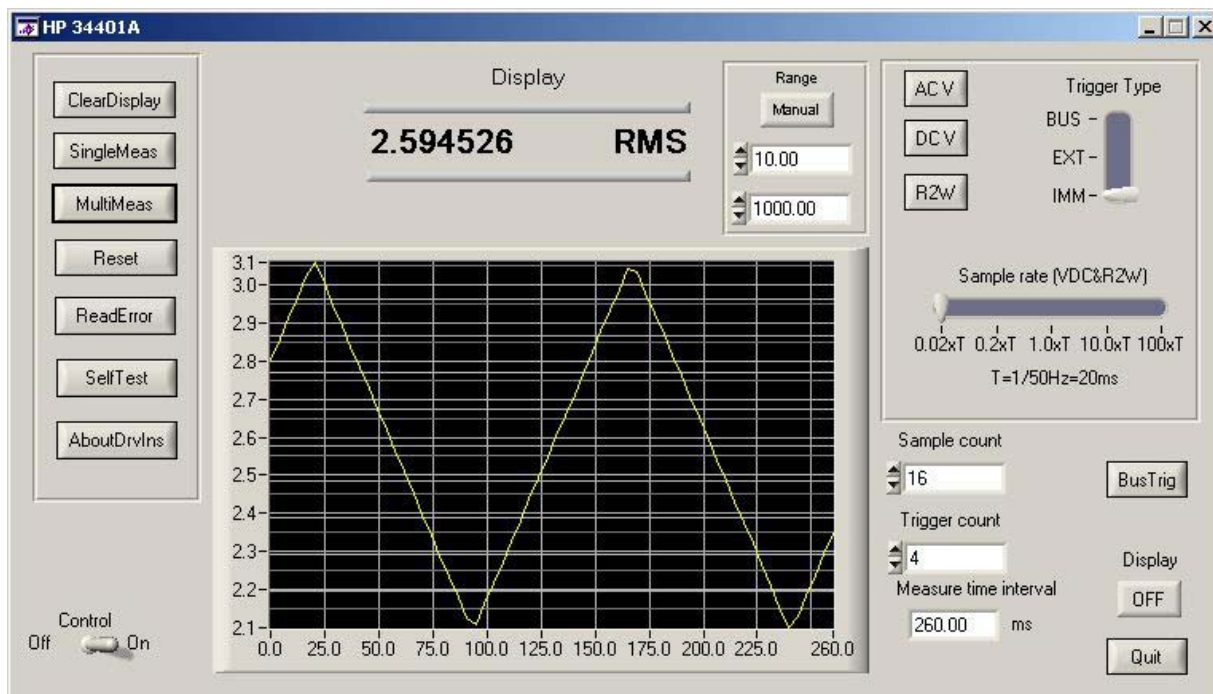
Sve navedene funkcije su dostupne na uvid u kodu softvera koji je priložen, i uz help programming drajvera nije teško prepoznati delove programa koji su od interesa.

Na slikama koje slede prikazani su neki specifični signali koji su izmereni upotrebom multimetra i softvera koji je realizovan i ovde opisan.

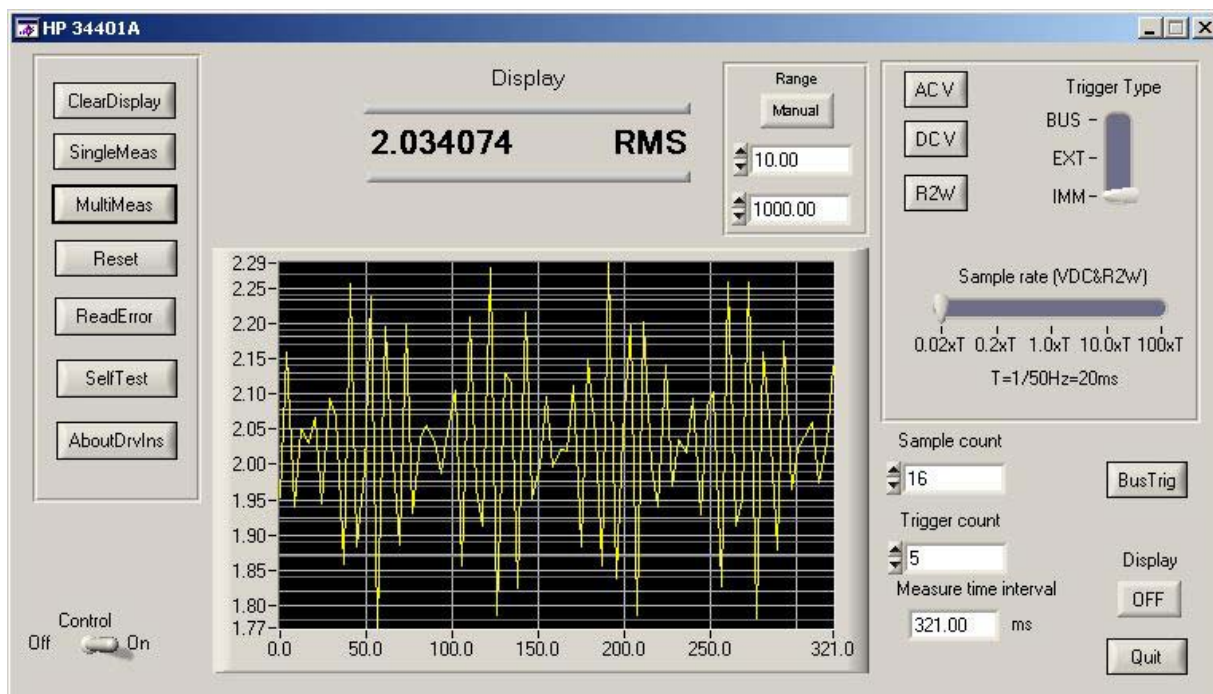


Slika 12. Merenje sinusoidalnog signala korišćenjem softvera iz primera 2





Slika 13. Merenje testerastog signala korišćenjem softvera iz primera 2



Slika 14. Merenje modulisanog sinusnog signala četvrtkama korišćenjem softvera iz primera 2

# LITERATURA

- [1] HP 34401A Programming Reference, Part Number 34401-90004, February 1996, Hewlett-Packard Company 1991, 1996
- [2] HP 34401A User's guide, Part Number 34401-90004, February 1996, Hewlett-Packard Company 1991, 1996
- [3] Dr Vujo Drndarević, „Personalni računari u sistemima merenja i upravljanja”, Akademska misao, Elektrotehnički fakultet, Saobraćajni fakultet, Beograd 2003.