Elektronski merni sistemi

Laboratorijska vežba 3

Očitavanje analognog kanala simulirane akvizicione kartice u *LabWindows/CVI* paketu

Vladimir Rajović, 2008/09

Sadržaj vežbe

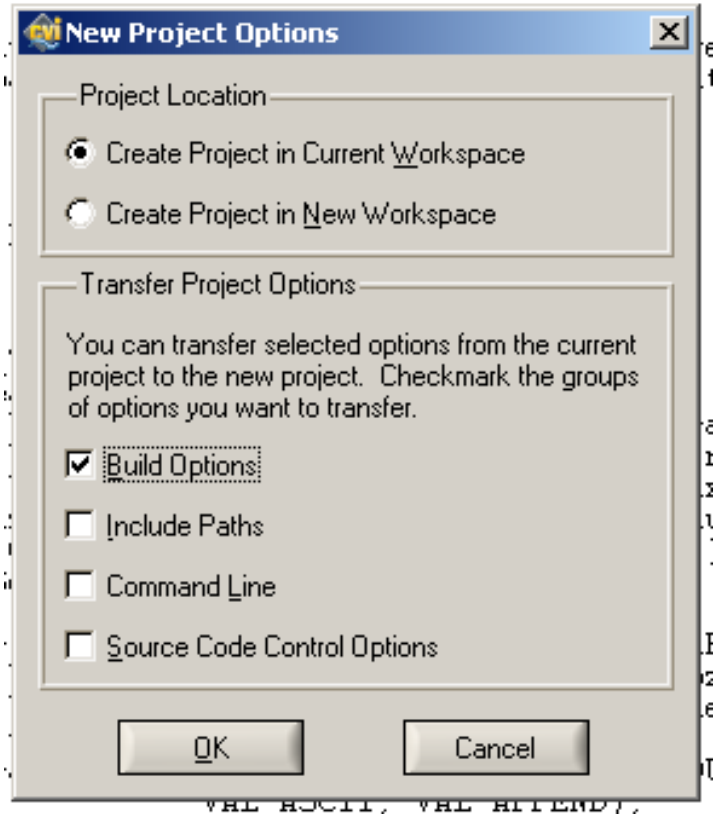
- Kreiranje simulirane DAQ-mx kartice tipa PCI-6025E, u programskom paketu MAX.
- Kreiranje dva merna taska u programu MAX za akviziciju analognih napona.
- Kreiranje korisničkog Interfejsa u paketu Labwindows/CVI.

- Generisanje C koda koji omogućava da traženi program pomoću izabranog kreiranog taska vrši akviziciju signala sa simulirane kartice.
- Prikazivanje signala na kontroli tipa StripChart. Za vreme akvizicije neprekidno se kontroliše nivo signala u odnosu na zadate granice, i na osnovu toga se uključuje ili isključuje alarmna LE dioda.
- Snimanje signala u fajl vrši se ako je uključena dozvola snimanja

Struktura i organizacija CVI programa

Kompletan CVI program se sastoji iz korisničkog interfejsa kreiranog u okviru .uir fajla i koda

Kreiranje projekta

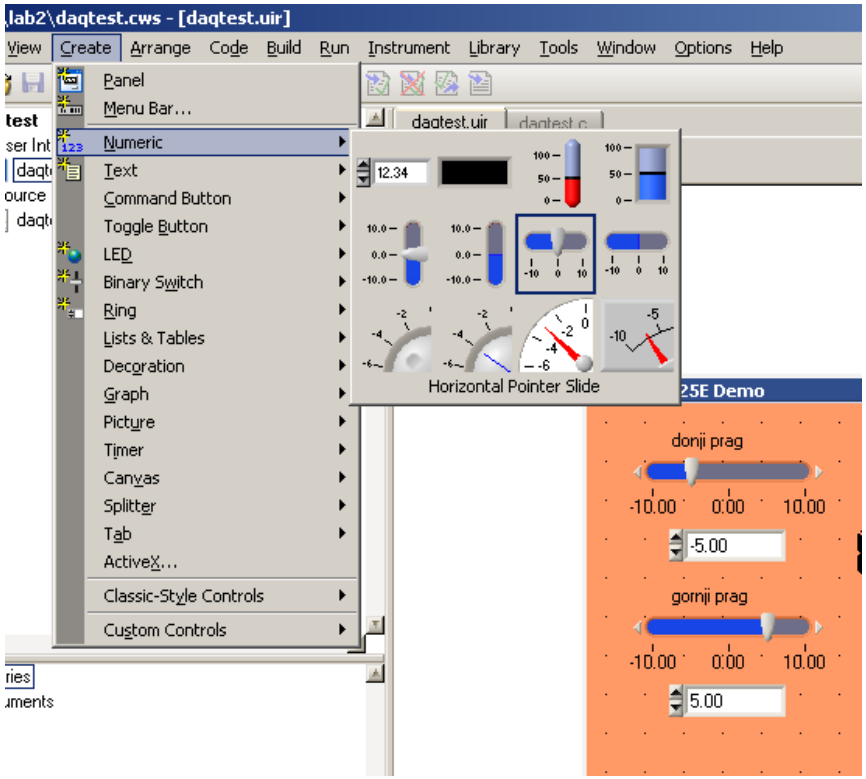


Tipovi falova

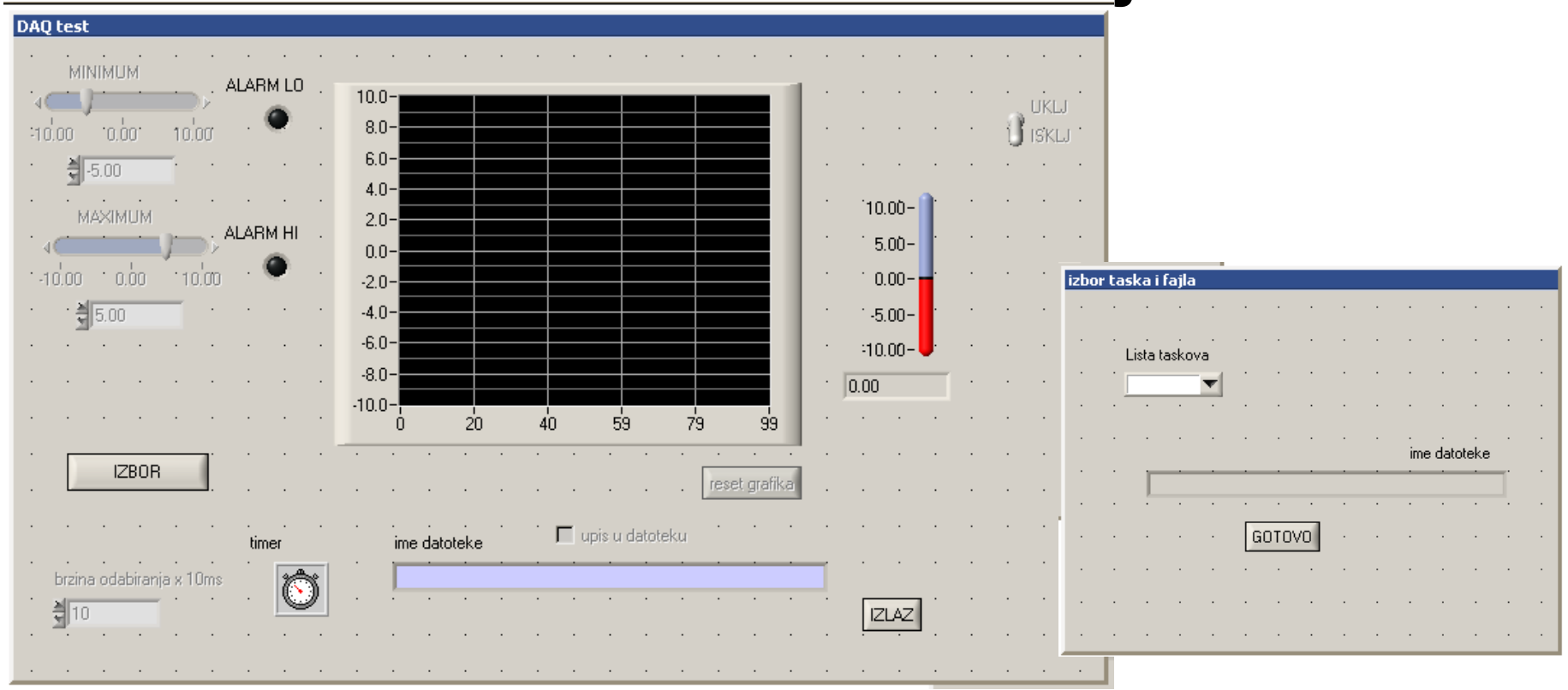
- .UIR fajl – Fajl koji se kreira interaktivno u saradnji sa Visual okruženjem CVI paketa i koji sadrži definicije grafičkog interfejsa korisničkog programa
- .C fajl – Tekstualni fajl u kom se nalazi kod aplikacije pisan u programskom jeziku C
- .H fajl - Tekstualni fajl u kome se nalaze deklaracije prototipova funkcija i programskih konstanti

Kreiranje grafičkog korisničkog interfejsa (GUI)

- Osnovni element svakog programa koji obuhvata i komunikaciju sa korisnikom je GUI – Grpahical User Interface koji se kreira u fajlu sa ekstenzijom .uir
- Korišćenjem ugrađenih kontrola brzo i jednostavno se kreira korisnički interfejs



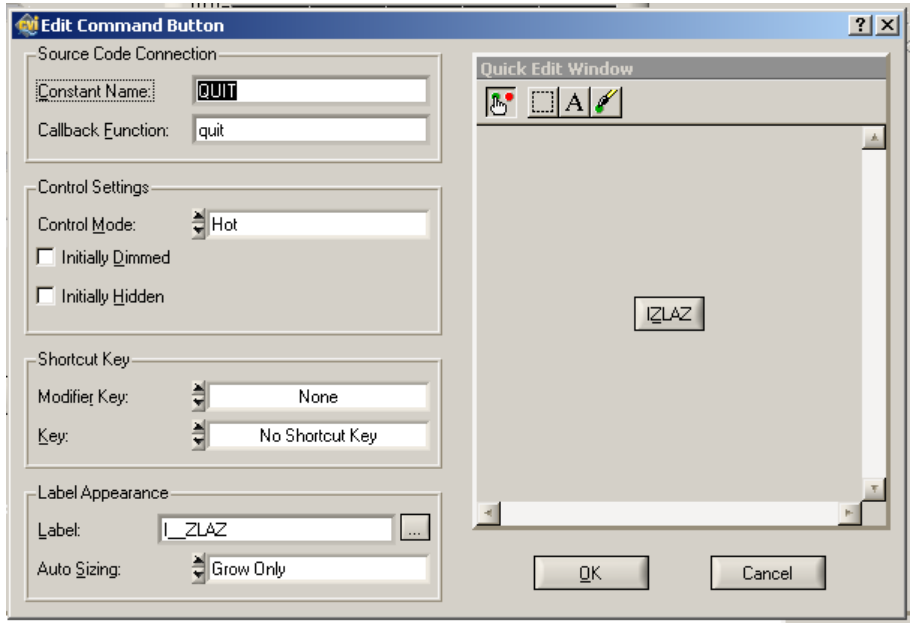
Korisnički interfejs



- Korisnički interfejs sadrži:
  - Kontrole čije akcije zadaje korisnik. Na primer dugme za izlazak iz programa.
  - Kontrole čije vrednosti se zadaje program
  - Kontrole čije vrednosti zadaje korisnik, a očitava program

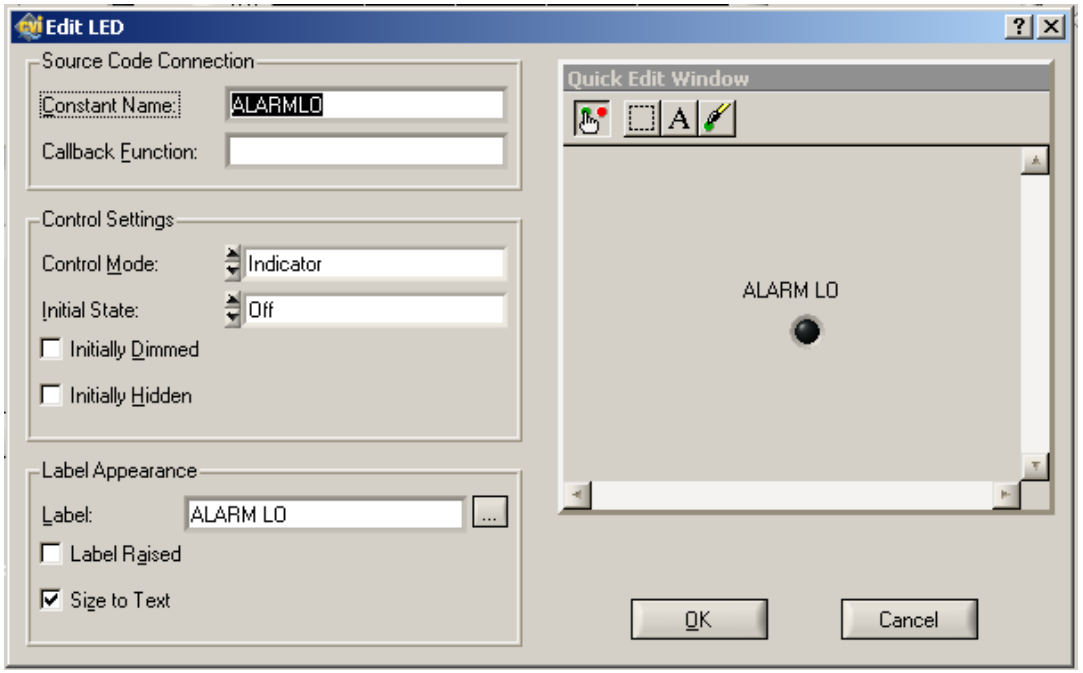
Definisanje parametara kontrola

- Svakom tipu kontrole čiju funkciju korisnik može da pokrene klikom miša ili na neki drugi način, se dodeljuje CallBack funkcija
- Na slici je dat primer funkcije za izlaz iz programa



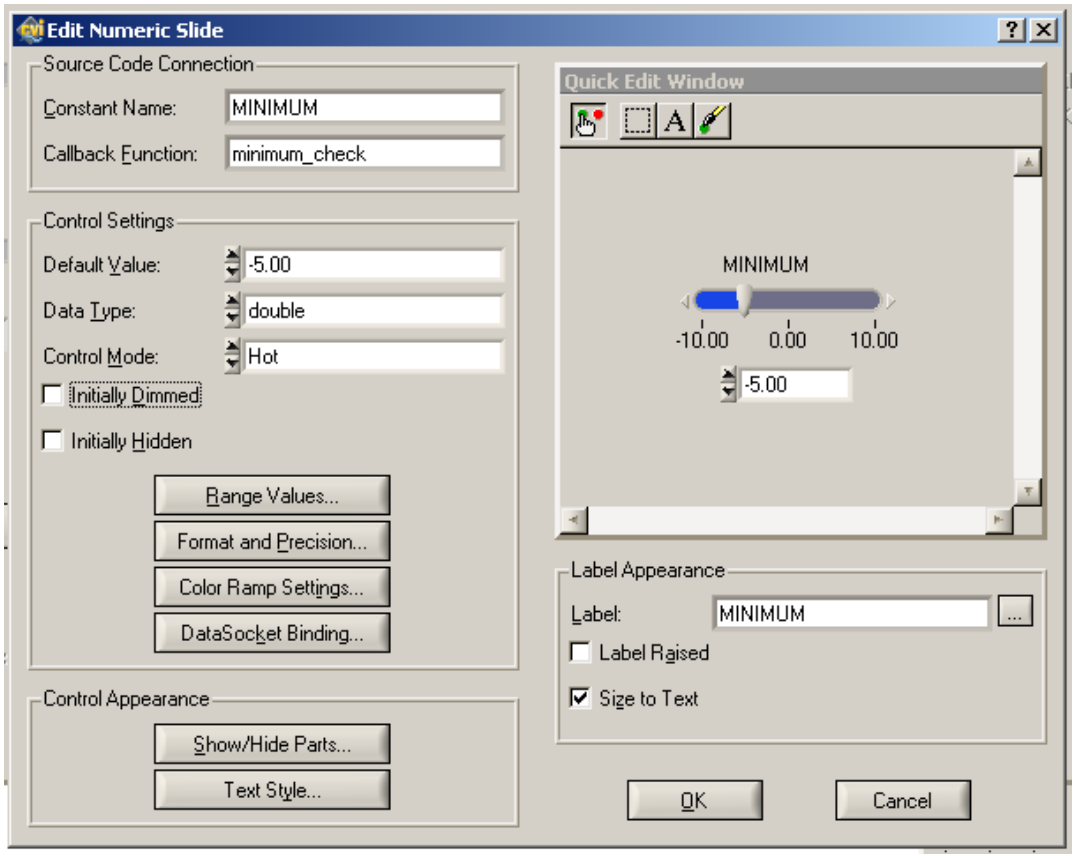
Definisanje parametara kontrola

- Svakoj kontroli čiju vrednost postavlja program se dodeljuje simbolička konstanta, koja služi kao jedinstven identifikator te kontrole
- Na slici je dat primer kontrole tipa LED-a.



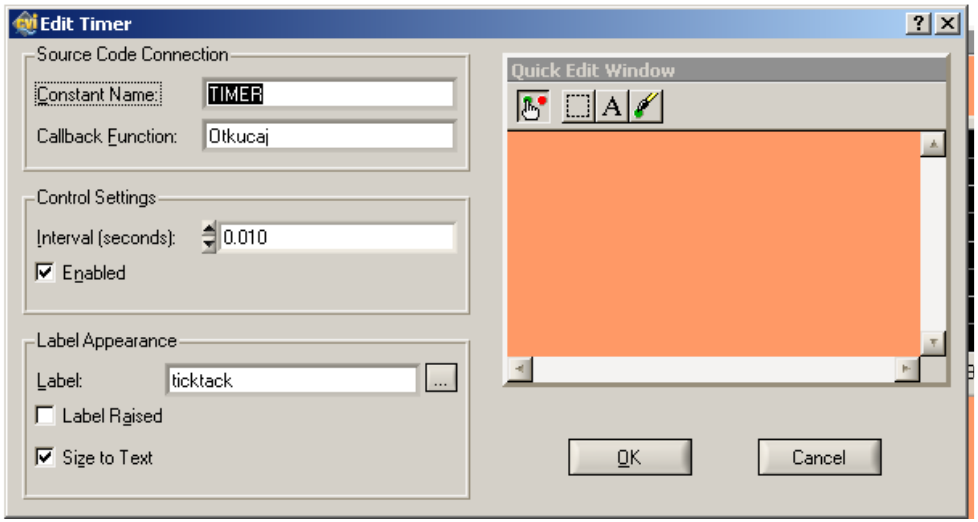
Definisanje parametara kontrola

- Svakoj kontroli čiju vrednost postavlja korisnik i koju program očitava kao parametar neophodan za funkcionisanje se dodeljuje simbolička konstanta, koja služi kao jedinstven identifikator te kontrole
- Na slici je dat primer kontrole tipa klizača za zadavanje numeričke vrednosti.



Definisanje parametara kontrola

- Poseban vid kontrole je Timer kontrola, koja je na korisničkom intefejsu gotove aplikacije nevidljiva.
- Zadatak ove kontrole je da periodično generiše poziv svoje CallBack funkcije nezavisno od korisnika.
- Najčešće se koristi kada je periodično potrebno vršiti akviziciju, obradu ili prikazivanje podataka.

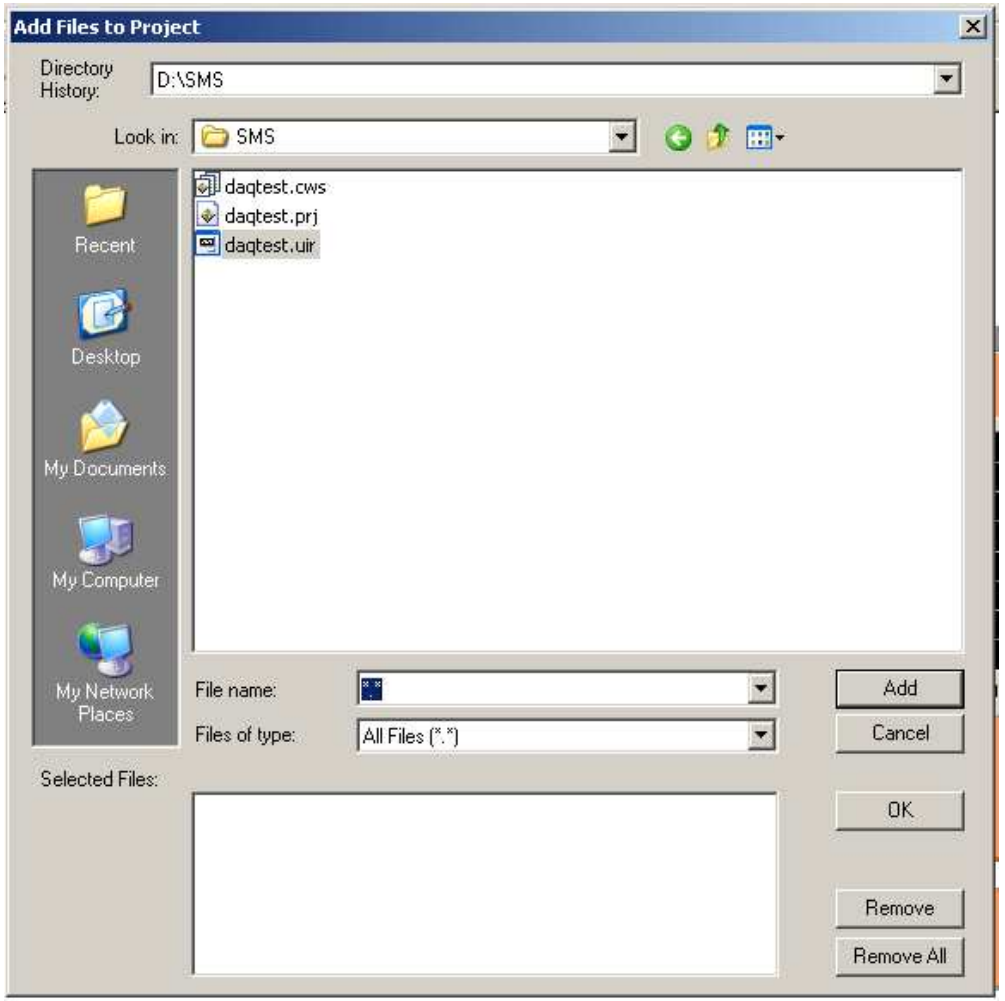


Korišćenje CallBack funkcija i simboličkih konstanti kontrola

- Svaka akcija korisnika se izvrašava u okviru CallBack funkcije koja je povezana sa tom kontrolom. Na primer pritiskom na dugme QUIT poziva se definisana funkcija izlaz() u okviru koje se obavljaju određene radnje koje definiše korisnik
- Svako očitavanje parametara od strane programa koje korisnik podešava preko kontrola tipa numeric ili slično, se obavlja pristupanjem kontrolama preko njihovih simboličkih konstanti tj. identifikatora. Na primer očitavanje vrednosti donjeg praga se vrši očitavanjem kontrole čiji je ID=MINIMUM.
- Svako postavljanje parametara nekoj kontroli tipa indikatora iz programa se vrši preko simboličke konstante tj. identifikatora te kontrole. Na primer uključivanje LED za napon ispod donjeg praga vrši se postavljanjem vrednosti kontrole čiji je ID= ALARMLO na 1.
- Periodične akcije kao što se prikupljanje podataka i prikazivanje na grafu se vrši sistemskim pozivom CallBack funkcije dodeljene kontroli tipa Timer.

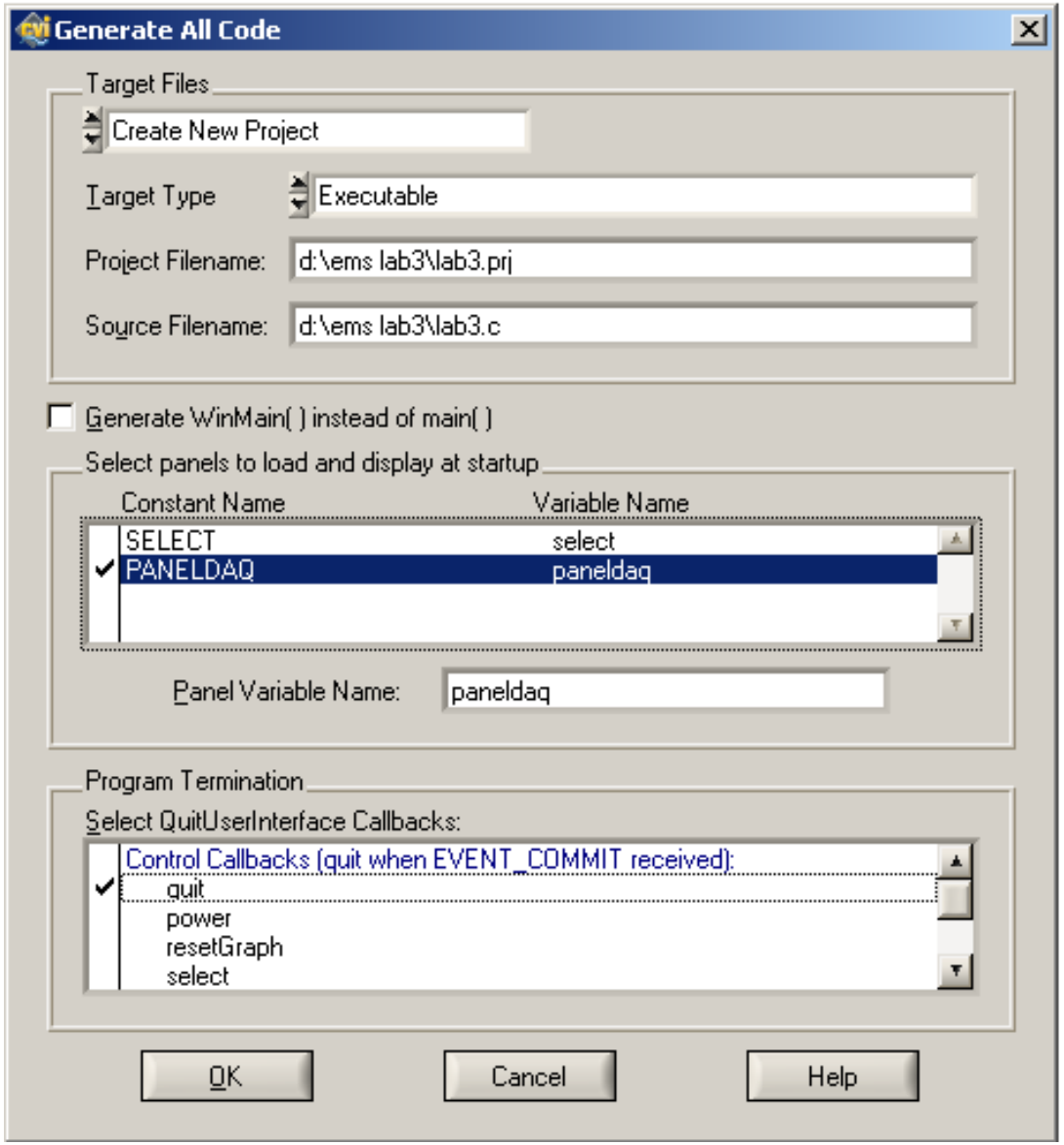
Dodavanje fajlova u projekat

- Nakon kreiranja .UIR fajl je potrebno dodati u projekat



Kreiranje kostura koda

- Nakon kreiranja GUI-a moguće je automatski generisati C kod korišćenjem wizard-a kao na slici.
- Pored main() funkcije kao osnove svakog C programa generišu se i tela svih deklarisanih CallBack funkcija



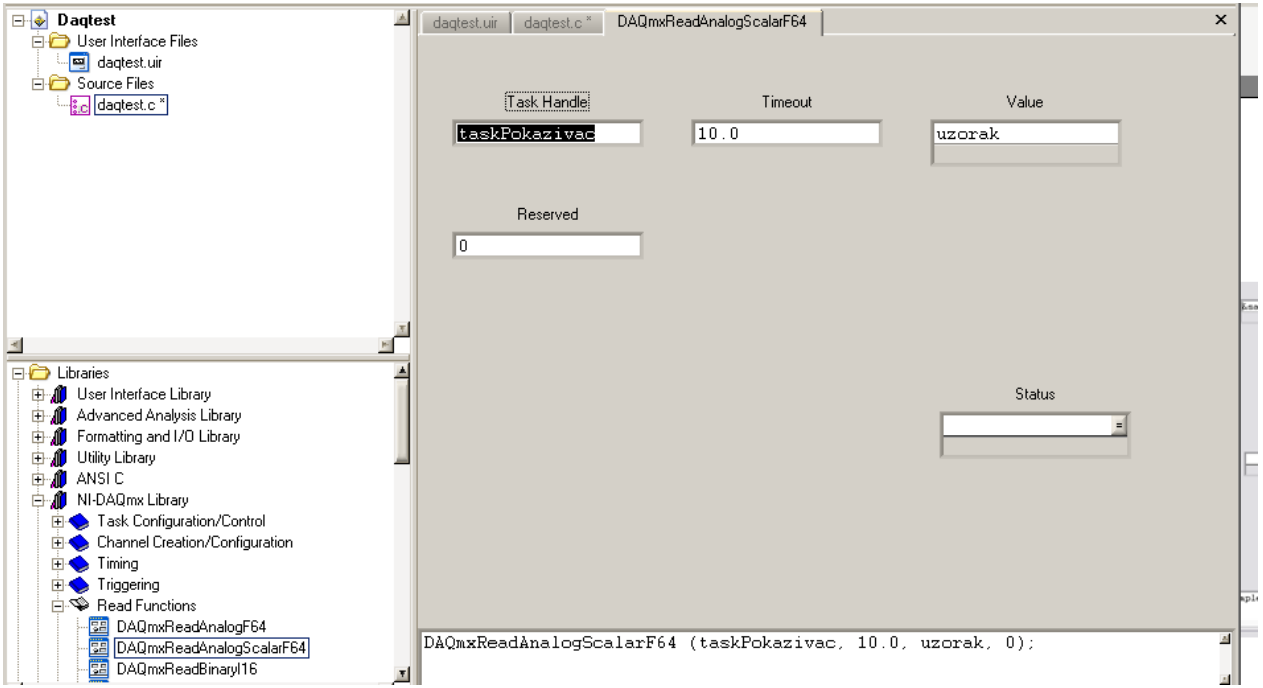
Kod

- Na slici je prikazan automatski generisan deo koda. S obzirom da je kontrola QUIT u wizard-u za generisanje koda odabrana za izlazak iz programa, automatski je ubačena funkcija za zatvaranje korisničkog interfejsa
- Sa druge strane funkcija kotrole Timer je prazna i potrebno je da dizajner programa ubaci kod koji obavlja željenu funkciju

```
int CVICALLBACK izlaz (int panel, int control, int event,
        void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
        case EVENT_LEFT_CLICK:

            break;
    }
    return 0;
}


int CVICALLBACK Otkucaj (int panel, int control, int event,
        void *callbackData, int eventData1, int eventData2)
{
    double uzorak;
    double minimum, maximum;
    int dozvola;
    char ime_fajla[MAX_PATHNAME_LEN];

    switch (event)
    {
        case EVENT_TIMER_TICK:


            break;
    }
    return 0;
}
```

Funkcije iz Biblioteke

- Dodavanje specifičnog koda u automatski kreiran kostur je moguće uz pomoć interaktivne grafičke biblioteke
- Na slici je dat primer kreiranja funkcije za očitavanje analognog napona iz Taska kreiranog u okviru programskog paketa MAX



Kod

- Na slici je dat kompletan kod popunjene CallBack funkcije kontrole Timer

```
int CVICALLBACK Otkucaj (int panel, int control, int event,
        void *callbackData, int eventData1, int eventData2)
{
    double uzorak;
    double minimum, maximum;
    int dozvola;
    char ime_fajla[MAX_PATHNAME_LEN];

    switch (event)
    {
        case EVENT_TIMER_TICK:
            DAQmxReadAnalogScalarF64 (taskPokazivac, 10.0, &uzorak, 0);
            SetCtrlVal (panelHandle, PANEL_TEKUCA, uzorak);
            GetCtrlVal (panelHandle, PANEL_MINIMUM, &minimum);
            GetCtrlVal (panelHandle, PANEL_MAXIMUM, &maximum);
            if ( (uzorak<minimum)  ||   (uzorak > maximum)  )
                SetCtrlVal (panelHandle, PANEL_ALARM, 1);
            else  SetCtrlVal (panelHandle, PANEL_ALARM, 0);

            PlotStripChartPoint (panelHandle, PANEL_GRAFIK, uzorak);
            GetCtrlVal (panelHandle, PANEL_DOZVOLA, &dozvola);
            GetCtrlVal (panelHandle, PANEL_FILENAME, ime_fajla);
            if (dozvola)
                ArrayToFile (ime_fajla, &uzorak, VAL_DOUBLE, 1, 1, VAL_GROUPS_TOGETHER,
                    VAL_GROUPS_AS_COLUMNS,VAL_CONST_WIDTH, 10,VAL_ASCII, VAL_APPEND);
            break;
    }
    return 0;
}
```

Izvršavanje i debug-ovanje programa

- Program počinje i završava svoje izvršavanje u funkciji main().
- Korisnički Interfejs se aktivira pozivom funkcije RunUserInterface() koja se automatski generiše pri kreiranju kostura koda.
- Od momenta pozivanja te funkcije pa sve dok se ne izađe iz korisničkog interfejsa pozivom funkcije QuitUserInterface(), kontrola programa je prebačena na operativni sistem i ceo kod se izvršava samo u okviru CallBack funkcija dodeljenih kontrolama.
- Pre i posle poziva korisničkog interfejsa se mogu obaviti inicijalizacije ili brisanje određenih resursa kao što je urađeno u primeru na slici, gde se pre korisničkog programa pozivaju funkcije za povezivanje aplikacije sa taskom kreiranim u okviru paketa MAX

```
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;  /* out of memory */
    if ((panelHandle = LoadPanel (0, "daqtest.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);

    DAQmxLoadTask ("napon", &taskPokazivac);

    RunUserInterface ();

    DAQmxClearTask (taskPokazivac);

    DiscardPanel (panelHandle);

    return 0;
}
```

Izvršavanje i debagovanje programa

- Program je moguće izvršavati u DEBUG modu, pri čemu je dozvoljeno postavljati sve vrste prekidnih tačaka i pristupati svim tipovima podataka
- Nakon završenog uspešnog debagovanja i testiranja aplikacije moguće je kreirati instalacioni paket koji daje mogućnost prenošenja aplikacije na računare koji nemaju ugrađen CVI paket, uz instalaciju neophodnih drajvera niskog nivoa za specifične periferije.

Kreiranje instalacionog paketa