



UPOTREBA LABVIEW PROGRAMSKOG OKRUŽENJA ZA OBRADU I PRIKAZ PODATAKA

Milica Tomić, BI 1/2020, Fakultet tehničkih nauka, Novi Sad
Lidija Renji, BI 4/2020, Fakultet tehničkih nauka, Novi Sad
Andrea Vaštag, BI 5/2020, Fakultet tehničkih nauka, Novi Sad
Iva Marković, BI 20/2020, Fakultet tehničkih nauka, Novi Sad

1. UVOD

Ovaj projekat ima za cilj realizaciju dinamičnog predstavljanja signala kako grafički, tako i tabelarno, u realnom vremenu. Omogućeno je učitavanje signala iz .csv fajlova putem jednostavnog odabira putanje ali i komunikacija sa pločom pomoću portova.

LabVIEW je izuzetno moćan grafički programski jezik koji za razliku od tradicionalnih tekstualno koncipiranih programskih jezika, koristi ikonice koje značajno olakšavaju razumevanje i manipulaciju podacima. Ključna karakteristika *LabVIEW-a* je *dataflow* programiranje, koje omogućava izvršavanje programa na osnovu protoka podataka. Brza i efikasna manipulacija informacijama je omogućena ovom tehnikom, što doprinosi olakšanom razvoju kompleksnih sistema. Kreiranje interaktivnog korisničkog interfejsa je omogućeno putem *Front panela*, koristeći alate i objekte radi stvaranja intuitivnih i funkcionalnih korisničkih iskustava. Prostor za programski kod i funkcije predstavlja *Block diagram*, koji je koncipiran tako da postigne razumevanje i jasnu vizualizaciju procesa izvršavanja programa, kao i sam tok izvršavanja programa.

2. IZBOR SIGNALA

Signali koji su se koristili za prikaz na graphicima su se mogli preuzeti na dva različita načina:

- Učitavanjem odgovarajućeg .csv fajla.
- Pomoću komunikacije između mikrokontrolera i računara

Mikrokontroler koji je korišćen jeste PIC18F87K2 i predstavlja deo razvojne ploče EasyPIC PRO V7 koju je razvila kompanija Mikroelektronika.

Sama inicijalizacija programa je vršena u *MCC-u (MPLAB Code Configurator)*. Podešeni su parametri za UART interfejs kako bi se omogućila komunikacija između mikrokontrolera i računara.

2.1. Opis koda

Kod je implementiran u beskonačnoj while petlji kako bi se omogućilo neprekidno izvršavanje koda dokle god je omogućena komunikacija između mikrokontrolera i računara. Prilikom svakog izvršavanja koda se definišu četiri slučajne vrednosti iz opsega od 0 do 255, koje će se kasnije prikazivati na odgovarajućim graphicima i tabeli. Ove vrednosti se upisuju u listu *temp*, nakon čega se formira paket podataka koji se šalje EUSART-om na računar. Sam paket podataka se sastoji od 6 bajtova. Prvi bajt je start bajt koji je jednak heksadecimalnoj vrednosti 0x03. Naredna četiri bajta odgovaraju slučajno generisanim vrednostima iz liste *temp*. Poslednji bajt, jeste stop bajt i on je jednak heksadecimalnoj vrednosti 0xFC. U listingu 1 je prikazan implementiran kod.

Signali koji su se učitali iz .csv fajlova su takođe formirani na osnovu datog koda. Podaci su se čuvali u .csv format koristeći ovaj projekat, a sam princip na osnovu kog je čuvanje podataka implementirano je opisano u nastavku izveštaja.

```

#include "mcc_generated_files/mcc.h"
#include <math.h>
#include <string.h>
#include <stdlib.h>

void main(void)
{
    SYSTEM_Initialize();

    uint8_t dt;
    uint8_t temp[4];
    uint8_t sig[4];
    while (1)
    {
        dt++;

        for (uint8_t i = 0; i < 4; i++)
        {
            sig[i] = rand()%256;
        }

        memcpy(temp, &sig, 4);

        EUSART1_Write(0x03);
        for (uint8_t i = 0; i < 4; i++)
        {
            EUSART1_Write(temp[i]);
        }
        EUSART1_Write(0xFC);
    }
}

```

Listing 1: Kod realizovan u MPLAB programskom okruženju

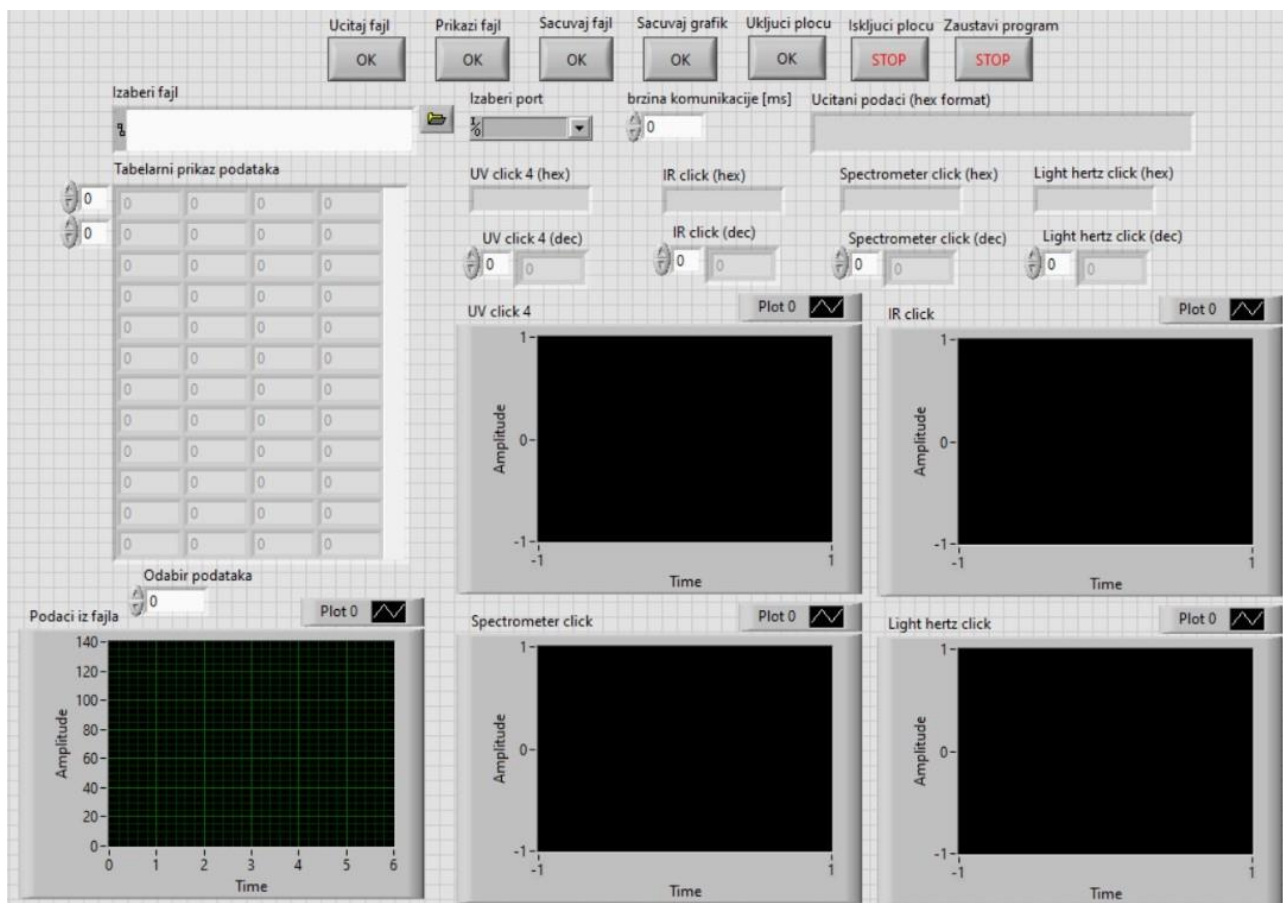
3. REALIZACIJA PROJEKTA - FRONT PANEL

Izgled *Front Panela* je prikazan na Slici 1. Dugmadi *Ucitaj fajl*, *Prikazi fajl*, *Sacuvaj fajl*, *Sacuvaj grafik*, *Ukljuci plocu*, *Zaustavi program* koriste se kada je potrebno aktivirati određenu akciju u *Event Case* strukturi. Dugme *Iskljuci plocu* se koristi kada korisnik hoće da prekine komunikaciju sa mikrokontrolera, posle čega se može nastaviti rad sa ostalim akcijama u *Event Case* strukturi.

Izaberi port se koristi za odabir porta na kojem se nalazi mikrokontroler. U tabeli *Tabelarni prikaz podataka* se prosleđuju real-time podaci sa mikrokontrolera, kao i u poljima *Ucitani podaci (hex)*, *UV click 4 (hex)*, *IR click (hex)*, *Spectrometer click (hex)*, *Light hertz click (hex)*. U *Ucitani podaci (hex)* se prosleđuje trenutni paket podataka, dok se u *UV click 4 (hex)*, *IR click (hex)*, *Spectrometer click (hex)*, *Light hertz click (hex)* prosleđuje po jedna heksadecimalna vrednost sa mikrokontrolera. U poljima *UV click 4 (dec)*, *IR click (dec)*, *Spectrometer click (dec)*, *Light hertz click (dec)* prosleđuju se vrednosti koje su konvertorane iz heksadecimalne vrednosti u decimalnu.

Na graficima *UV click 4*, *IR click*, *Spectrometer click*, *Light hertz click* se iscrtavaju vrednosti sa mikrokontrolera koje su prvobino konvertovane iz heksadecimalne vrednosti u decimalnu. Brzina kojom se podaci prosleđuju u tabelu, pomenuta polja i na graficima zavisi od upisa u kontrolu *Brzina komunikacije [ms]*, gde je potrebno da korisnik unese željenu brzinu kojom će se podaci slati u ms.

Izaberi fajl se koristi kada korisnik želi da učita određeni .csv fajl sa računara pri čemu se prosleđuje putanja datog fajla. Nakon pritiska dugmeta *Prikazi fajl*, prikazuju se podaci koji su prvobitno učitani. Ako postoji više podataka koje je moguće prikazati, promenom vrednosti na *Odabir podataka* moguće je promeniti prikaz na grafiku. U tabeli *Tabelarni prikaz* pored ispisivanja *real-time* podataka sa mikrokontrolera, moguće je ispisati i vrednosti nakon učitavanja .csv fajla, odnosno nakon pritiska dugmeta *Ucitaj fajl*.



Slika 1. Prikaz Front Panel-a

4. REALIZACIJA PROJEKTA – LABVIEW

4.1 Visa

VISA (Virtual Instrument Software Architecture) predstavlja biblioteku u LabView koja omogućava komunikaciju računara sa različitim instrumentima preko različitih protokola, kao što su *GPIB* (General Purpose Interface Bus), *USB* (Universal Serial Bus), *Ethernet*, i tako dalje. Ova biblioteka omogućava korisnicima da komuniciraju sa uređajima kao što su osciloskopi, multimetri i drugi instrumenti koji podržavaju komunikaciju preko pomenutih protokola. Takođe, VISA predstavlja standard za konfigurisanje, programiranje i analiziranje instrumentacionih sistema baziranim na pomenutim portokolima.

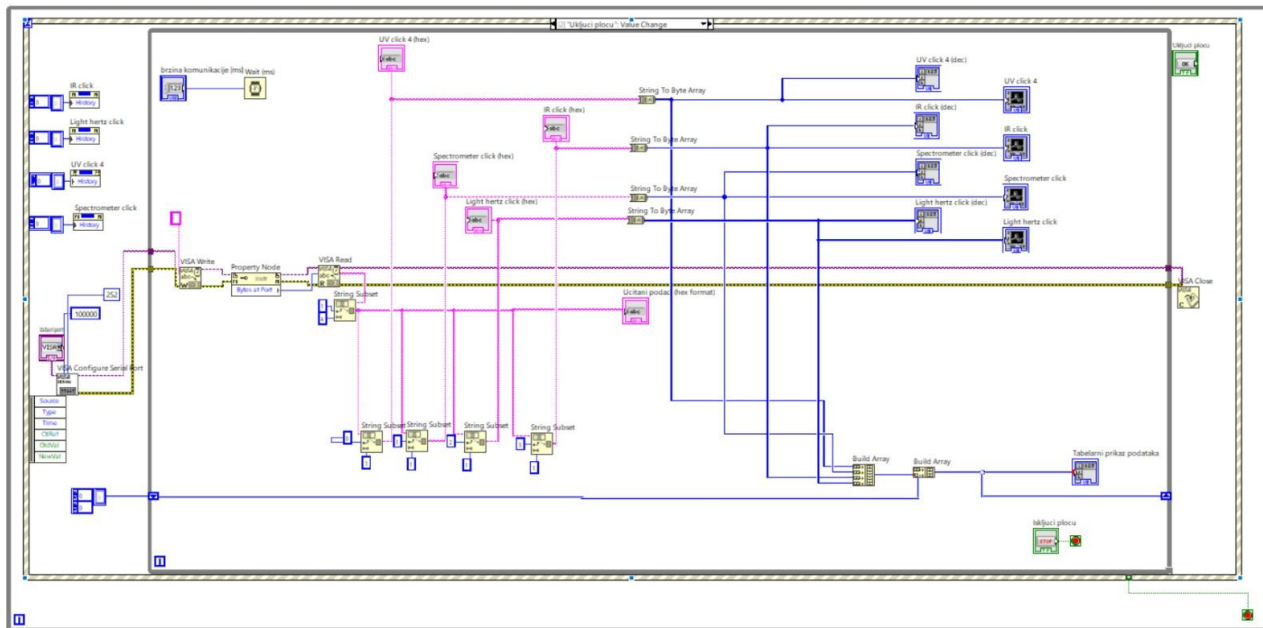
NI – VISA je instrumentacioni drajver koji implementira VISA standard. Osnovne funkcije koje su korišćene:

- *VISA resource name* – funkcija pomoću koje se uspostavlja komunikaciju ka mikrokontroleru tako što se izabere USB port preko kojeg je mikrokontroler povezan sa računarom i tada započinje komunikacija mikrokontrolera i LabView okruženja. Zarad lakšeg rukovanja programom od strane korisnika ova funkcija je preimenovana u naziv *Izaberi fajl*.
- *VISA Configure Serial Port* – funkcija pomoću koje se uspostavlja osnovni parametri za komunikaciju kao što je baud rate (brzina slanja podataka).
- *VISA Write* – funkcija pomoću koje se šalju podaci ili komande ka mikrokontroleru. Ova funkcija omogućava slanje podataka u obliku bajtova, stringova ili drugih formata koji su podržani od strane instrumenta.
- *VISA Read* – funkcija koja se koristi za čitanje podataka sa mikrokontrolera.
- *VISA Close* – funkcija pomoću koje se prekida komunikacija sa mikrokontrolerom.

Na Slici 2 prikazan je deo koda koji predstavlja komunikaciju mikrokontrolera i LabView okruženja. Na izlazu *VISA Read* funkcije, odnosno u indikatoru prosleđuje se string heksadecimalnih vrednosti od 6 bajta sa mikrokontrolera. Prvi bajt predstavlja početak slanja seta podataka (0x03), poslednji bajt predstavlja kraj slanja seta podataka (0xFC), dok 4 bajta koja se nalaze između predstavljaju slučajno generisane vrednosti koje potrebno prikazati u tabeli i na graficima.

Pomoću funkcije *String Subset* su prvo eliminisani prvi i poslednji bajt koji nisu potrebni za prikaz, a zatim je ova funkcija iskorišćena kako bi se dobile 4 zasebne vrednosti (*UV click 4 (hex)*, *IR click (hex)*, *Spectrometer click (hex)*, *Light hertz click (hex)*). U *String Subset* podešavaju se parametri *offset* i *length*. Dužina stringa je 6 bajta, zato u prvoj funkciji *String Subset* za parametar *offset* je zadata vrednost 1, odnosno ispis kreće od drugog bajta, a za parametar *length* postavljena je vrednost 4, odnosno sledeće 4 vrednosti se ispisuju u *read buffer* indikatoru, odnosno na indikatoru će se pojaviti 4 bajta koji predstavljaju slučajno generisane vrednosti sa mikrokotrolera.

Sličnom logikom su preostali bajtovi podeljeni, *length* je postavljen na 1 zato što je potrebno izvući po 1 bajt u svakoj funkciji, dok u *offset* parametru se podešava koji bajt treba preuzeti. *String To Byte Array* funkcija služi da dati string na indikatorima *UV click 4 (hex)*, *IR click (hex)*, *Spectrometer click (hex)*, *Light hertz click (hex)* prevede u niz bajtova, odnosno na izlazu se dobijaju decimalne vrednosti koje se dalje prosleđuju graficima i tabeli.



Slika 2. Komunikacija

Kako bi se vrednosti koje stižu sa mikrokotrolera prosledile u *real-time* tabeli, potrebno je pomoću *Shift registra* prvobitno inicijalizovati vrednost kolone na 0, a nakon svake iteracije, odnosno nakon pristizanja novih podataka, taj broj povećati za jedan. U prvoj *Build Array* funkciji se ubacuju 4 vrednosti sa mikrokotrolera, dok se u drugoj funkciji te vrednosti spajaju sa vrednošću u *Shift registru* pri čemu se pravi 2D niz i takav signal se prosleđuje na indikator za tabelu. Takođe, izlaz druge *Build Array* funkcije koja kreira 2D niz se prosleđuje na izlaz *Shift registra* kako bi se u svakoj iteraciji broj kolona automatski povećao.

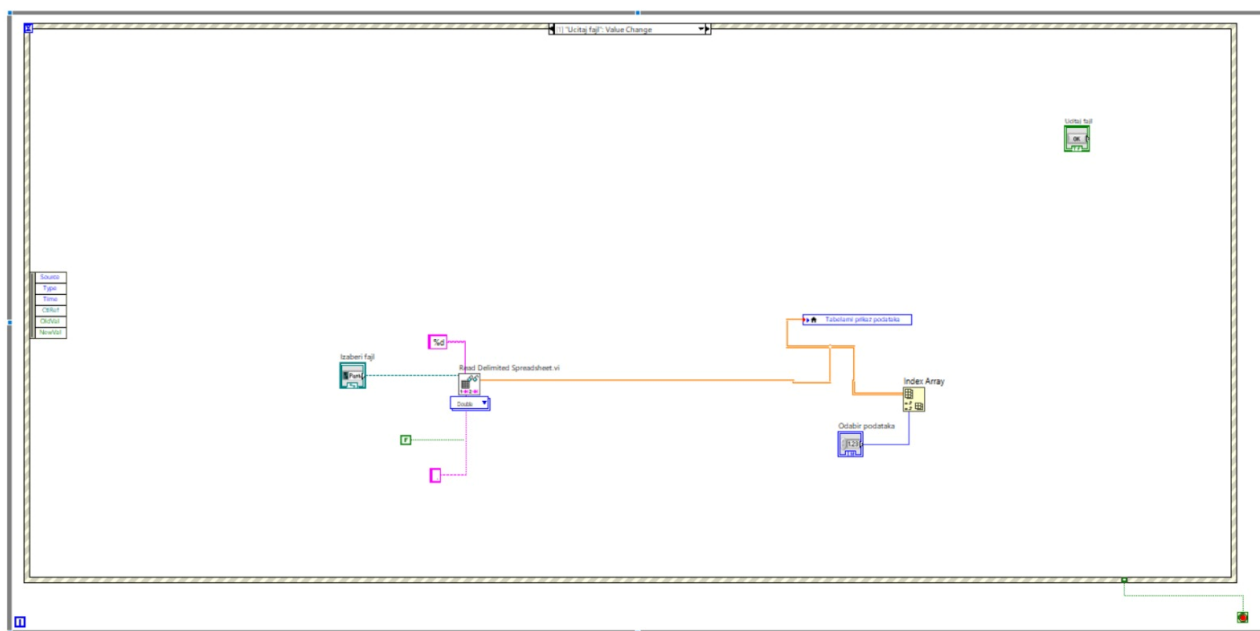
4.2 Učitavanje

Deo koda koji je predviđen za učitavanje .csv fajlova je realizovan u prvom stanju *Event Structure* pod nazivom „Učitaj fajl“.

Učitavanje .csv fajla se započinje izborom putanje na kojoj se nalazi fajl koji je potrebno učitati. Kako bi bio omogućen izbor putanje korišćena je kontrola pod nazivom *Izaberi fajl*. Nakon izbora putanje se vrši razdvajanje podataka koristeći blok koji se naziva *Read Delimited Spreadsheet.vi*. Znak zarez, je postavljen za graničnik na osnovu kog se će vršiti deljenje podataka.

Kako bi se omogućilo ispravno prikazivanje podataka na grafiku, isključena je opcija transponovanja podataka. Takođe, za format konverzije podata je postavljena vrednost %d, što znači da će se tekstualni podaci tipa *string* konvertovati u celobrojan broj. Kako bi se omogućio ispis podataka u vidu tabele postavljena je lokalna promenljiva indikatora *Array* u modu rada *write*.

Kako bi se omogućio prikaz samo jednog signala na grafiku dodat je blok pod nazivom *Index array*. Odabir signala se vrši u *Front Panel* – u pomoću kontrole koja se naziva *Odabir podataka*.

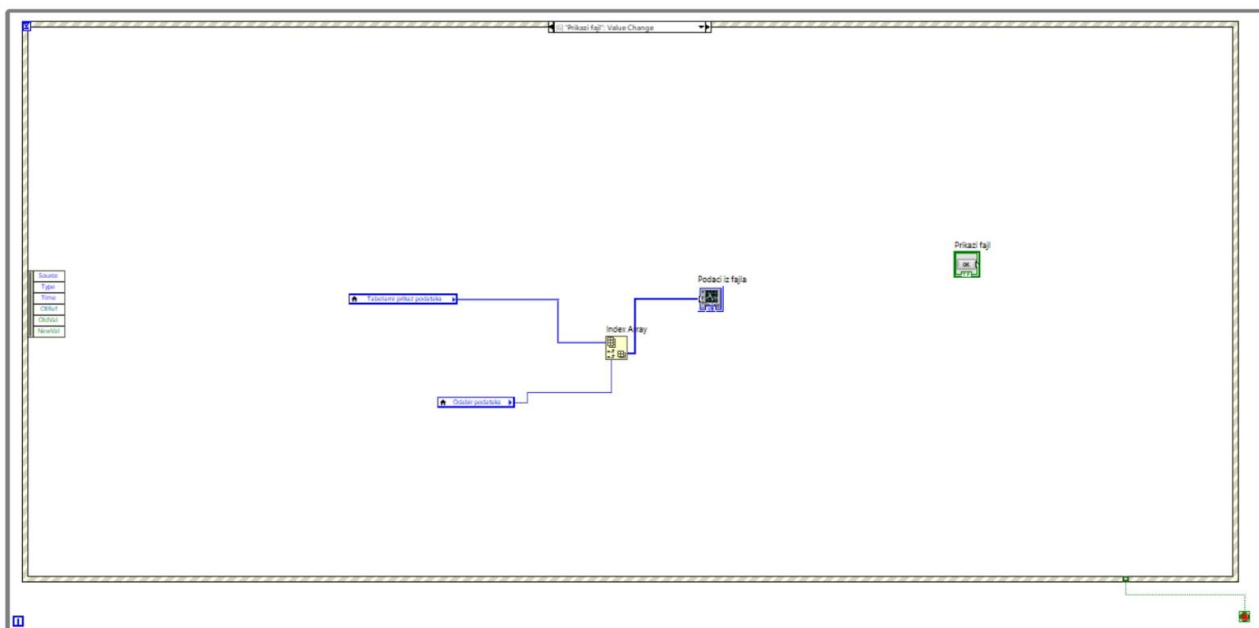


Slika 3. Učitaj fajl

Deo koda predviđen za grafički prikaz signala realizovan je u šestom stanju *Event Structure*, pod nazivom „Prikaži fajl“. U okviru ovog stanja nalaze se dve lokalne promenljive, jedan prekidač i jedan *Waveform Graph*.

Na samom početku izvršavanja koda, pristupa se dvema lokalnim promenljivim *Tabelarni prikaz podataka* i *Odabir podataka*. Obe lokalne promenljive su konfigurisane tako da se obavlja proces čitanja iz objekta. Takođe, odabrane su kao lokalne promenljive, jer podatke, kojima je potrebno pristupiti, iz prethodnog dela koda, nije moguće povezati žicom sa trenutnim kodom, te je ovo jedan od načina realizacije. Dakle, pritiskom na prekidač *Prikaži fajl*, pristupa se lokalnim promenljivim, čiji se terminali nalaze u prvom stanju *Event Structure*, *Učitaj fajl*.

Izlazni podaci funkcije su ustvari ulazni podaci *Wavefrom Graph-a*, čime je kompletirana funkcionalnost ovog dela koda. Naime, *Podaci iz fajla* prihvata niz elemenata i prikazuje ih u grafičkoj formi, gde se prilikom dolaska novog niza vrednosti brišu stare i prikazuju samo te nove. Ova osobina *Graph-a* je korsina, pošto će se na taj način prilikom svakog pritiska tastera ispisivati nove vrednosti na *Graph-u*, a stare se neće pamtit, odnosno, prikazivaće se samo ono što je trenutno na ulazu.

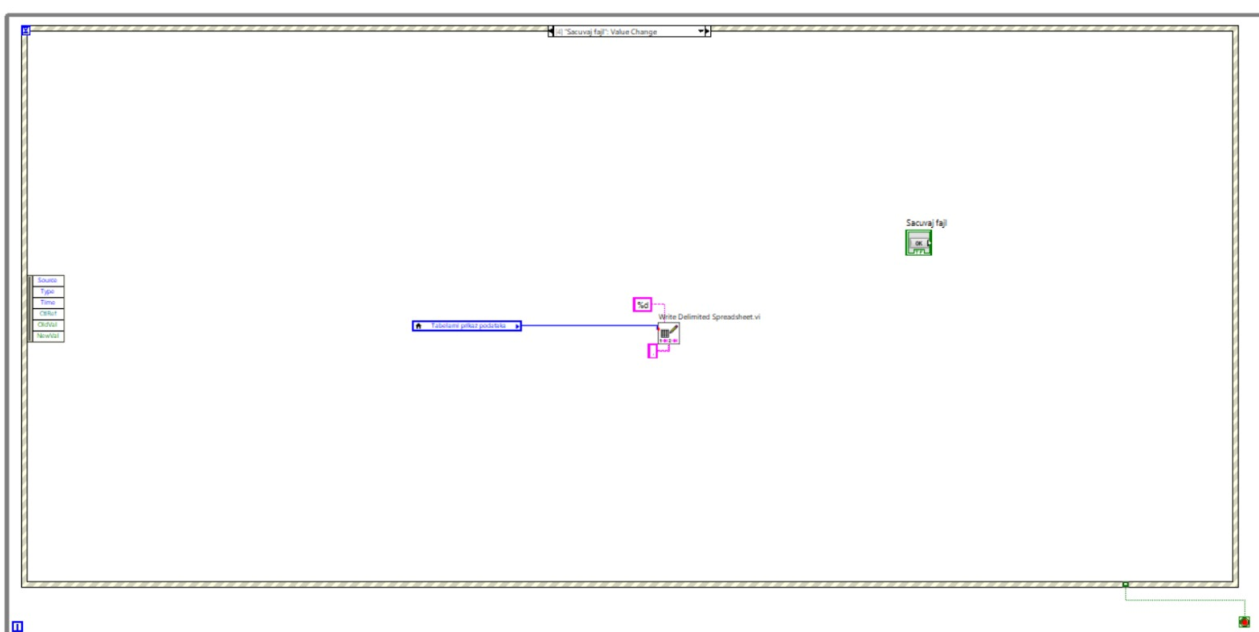


Slika 4. Prikaži fajl

4.4 Čuvanje

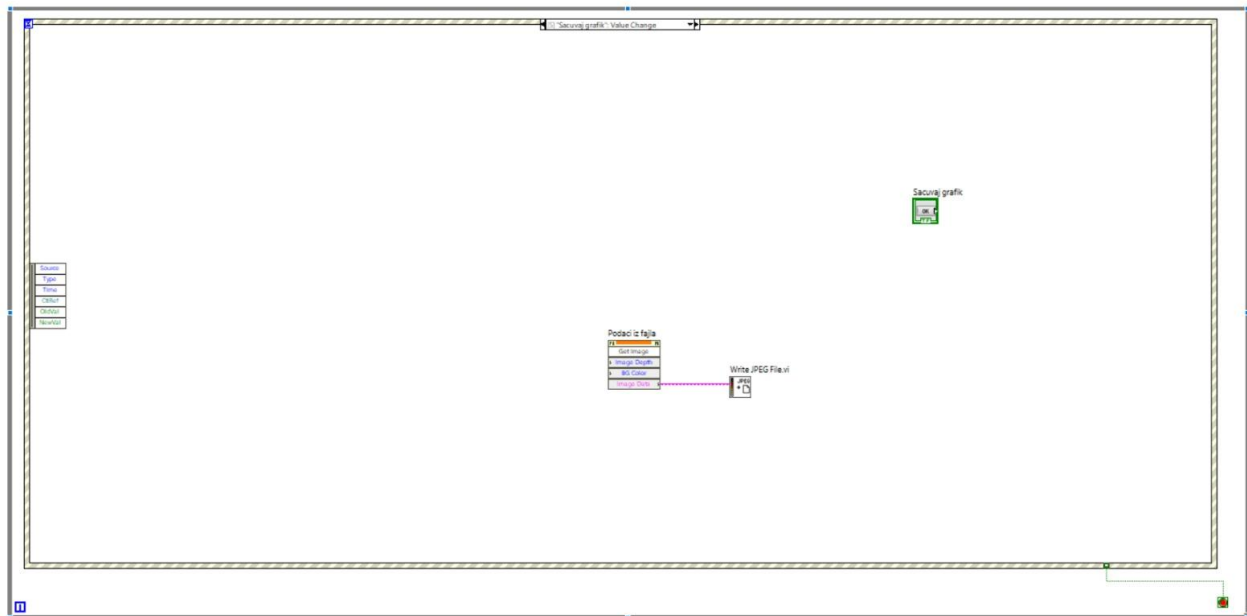
Omogućeno je čuvanje tabelarnog prikaza signala sa .csv ekstenzijom ali i čuvanje grafičkog prikaza signala koji je u .jpeg formatu. Čuvanje CSV-a je realizovano u drugom stanju *Event Structure*, a čuvanje signala u trećem stanju *Event Structure*.

Izgled drugog stanja *Event Structure*, koji se odnosi na čuvanje tabelarnog prikaza signala dat je na Slici 5. U okviru *Block* dijagrama se nalazi prekidač pod nazivom *Sačuvaj fajl*. Prekidač je *Boolean* tip podataka i može imati dva stanja: *True* i *False*. Podrazumevano stanje prekidača je *False* i u tom stanju se ne izvršava program unutar datog stanja *Event Structure*. Pritiskom odgovarajućeg dugmeta na *Front* panelu se vrednost prekidača postavlja na *True* i time započinje izvršavanje odgovarajućeg dela programa. Pristupa se lokalnoj promenljivoj *Tabelarni prikaz podataka* koja sadrži 2D niz *Numeric (Uint)* podataka i prosleđuje se bloku *Write Delimited Spreadsheet.vi* koji čuva signal kao .csv fajl. Ovaj blok spaja u tabelu pomoću stringa „ , „ jer je to spojni string za .csv tip fajla. Takođe se pomoću %d zaokružuje vrednost tipa podataka na *Int64* tip.



Slika 5. Sačuvaj fajl

Izgled trećeg stanja *Event Structure*, koji se odnosi na čuvanje grafičkog prikaza signala dat je na Slici 6. U okviru *Block* dijagrama se nalazi prekidač pod nazivom *Sačuvaj grafik*. Prekidač je *Boolean* tip podataka i može imati dva stanja: *True* i *False*. Podrazumevano stanje prekidača je *False* i u tom stanju se ne izvršava program unutar datog stanja *Event strukture*. Pritiskom odgovarajućeg dugmeta na Front panelu se vrednost prekidača postavlja na *True* i time započinje izvršavanje odgovarajućeg dela programa. *Waveform Property Node* omogućava pristup i manipulaciju podacima talasnih oblika, dok *Image Data* svojstvo omogućava pristup podacima slike koja je povezana sa tim talasnim oblikom. Pristupa se podacima slike grafa iz kog se izvlači slika koja se čuva kao .jpeg fajl.

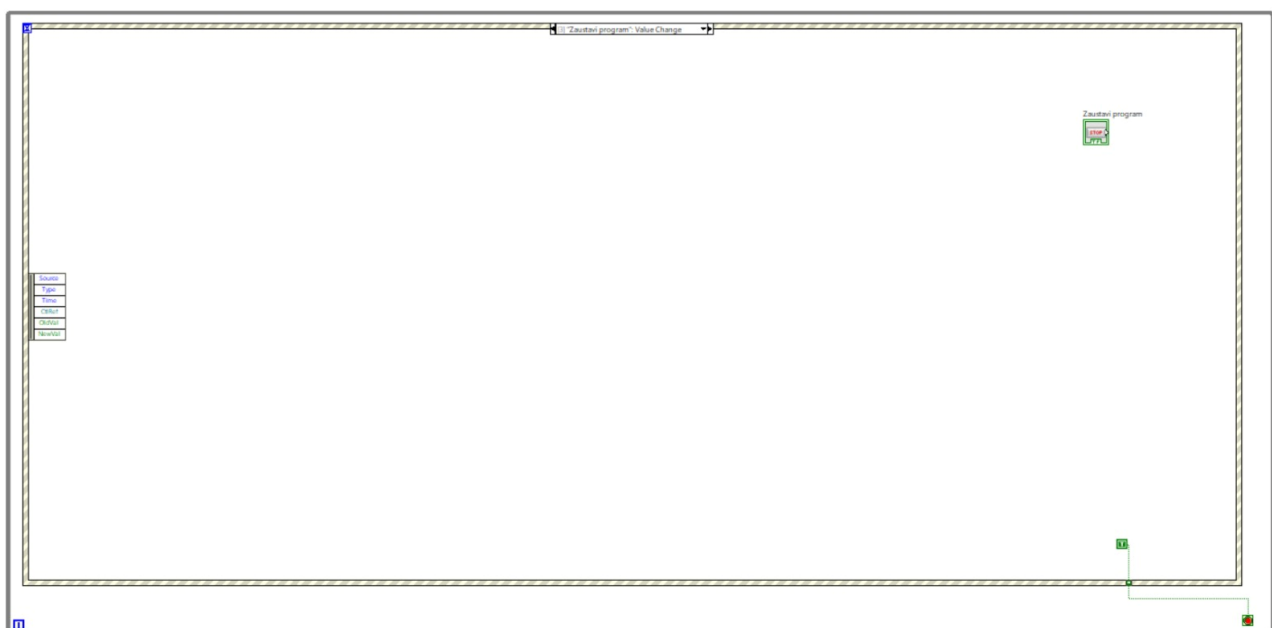


Slika 6. Sačuvaj grafik

4.5 Stop

U sklopu trećeg stanja *Event Structure* se nalazi dugme pod nazivom *Zaustavi program*. Pritiskom na dugme se prosleđuje *Boolean* konstanta koja ima vrednost *True* na terminator spoljašnje *while* petlje koja je deo mašine stanja (*Finite State machine*). Na ovaj način je omogućeno prekidanje izvašavanja ovog programa.

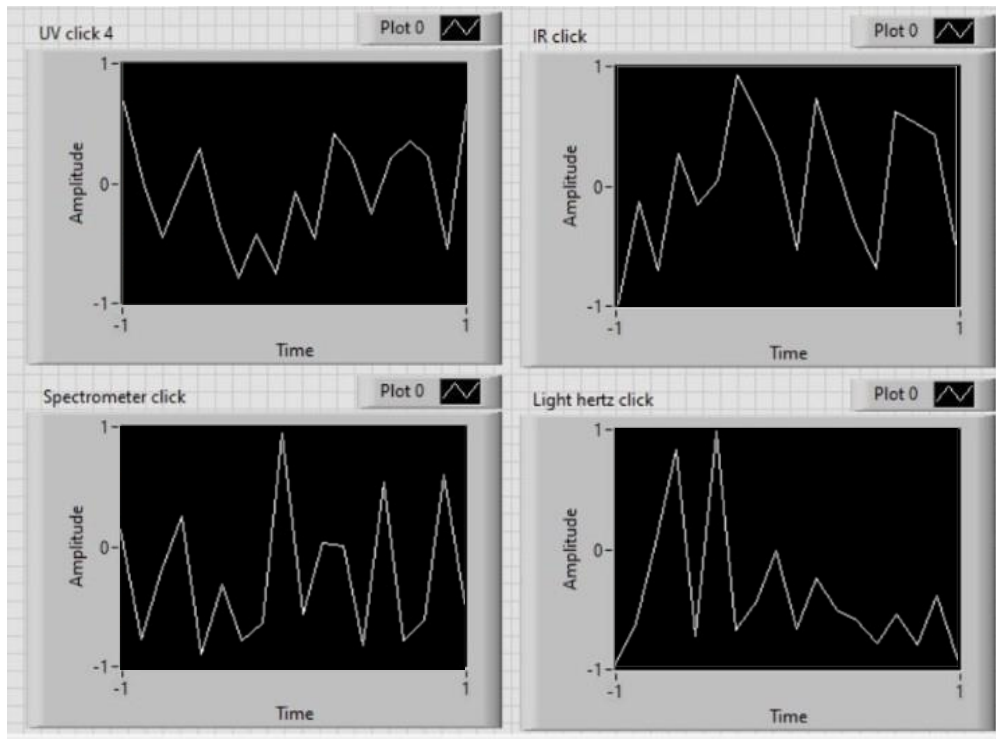
Izgled trećeg stanja *Event Structure* na *Block Diagram-u* u *LABView-u* prikazan je ispod na Slici 7.



Slika 7. Zaustavi program

4. REZULTATI

U sklopu ovog projekta, analiza signala sprovedena je na osnovu četiri random generisana signala. Svaki signal je predstavljen na zasebnom grafiku, kako bi dalja manipulacija i obrada bile lakše. Na Slici 8, u nastavku, prikazan je izgled *Front Panel-a* na kome se nalaze četiri različita grafika. Dodatno, uočava se da je na svakom grafiku prikazan po jedan random signal, gde je na x-osi označeno vreme akvizicije, a na y-osi amplituda svakog od signala.



Slika 8. Random signali

5. ZAKLJUČAK

Zahvaljujući ovom programu omogućeno je učitavanje podataka iz .csv fajlova odabirom putanje, kao i njihov tabelarni i grafički prikaz. Realizovana je i komunikacija sa periferijama pomoću VISA-e (Virtual Instrument Software Architecture), putem COM portova. Na taj način je postignuto tabelarno i grafičko predstavljanje signala, snimljenih u realnom vremenu. Dodatno, implementirano je i čuvanje grafičke predstave signala u .jpeg formatu.

Funkcionalnost programa je testirana na randomizovanim signalima, napisanim u MPLab-u i spuštenim na razvojnu ploču EasyPIC PRO v7, sa kojom komunicira računara putem VISA-e.

S obzirom na implementirane funkcionalnosti, ovaj projekat ima široku oblast primene. Jedna od mogućnosti, jeste upotreba u institutima gde se obavlja detekcija i merenje jonizujućih i nejonizujućih zračenja. Naime, moguće je iskoristiti EasyPIC PRO v7 razvojnu ploču i Click pločice kompanije „Mikroelektronika“ i povezati ih u jednu celinu. Na primer, moguće je napisati drajvere za UV 4 Click, Spectrometer Click, LightHz Click i IR Click, povezati ih na ploču i vršiti akviziciju i obradu signala snimljenih pomoću pomenutih komponenti.

6. LITERATURA

[1] www.ni.com/academic/students/learn-labview/

[2] Dokumentacija sa Sova platforme

