

# Grafičko sučelje u Matlabu za osciloskop TDS 224

---

**Pitinac, Filip**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:895806>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-12-29**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**GRAFIČKO SUČELJE U MATLABU ZA OSCIOSKOP  
TDS 224**

**Diplomski rad**

**Filip Pitinac**

**Osijek, 2020.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMATIČKIH TEHNOLOGIJA ORBITEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 17.09.2020.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime studenta:</b>	Filip Pitinac
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Elektrotehnika
<b>Mat. br. studenta, godina upisa:</b>	D-1194, 20.09.2019.
<b>OIB studenta:</b>	10475882219
<b>Mentor:</b>	Prof. dr. sc. Kruno Miličević
<b>Sumentor:</b>	Ivan Biondić
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Prof. dr. sc. Denis Pelin
<b>Član Povjerenstva 1:</b>	Prof. dr. sc. Kruno Miličević
<b>Član Povjerenstva 2:</b>	Dragan Vulin
<b>Naslov diplomskog rada:</b>	Grafičko sučelje u Matlabu za osciloskop TDS 224
<b>Znanstvena grana rada:</b>	<b>Elektrostrojarstvo (zn. polje elektrotehnika)</b>
<b>Zadatak diplomskog rada:</b>	Objasniti osnovne mogućnosti grafičkog sučelja u Matlabu. Objasniti osnovnu sintaksu protokola za komunikaciju računala s osciloskopom (SCPI). Izrada GUI-a za osciloskop TDS 224. Testirati izrađeni program.
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	17.09.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 30.09.2020.

**Ime i prezime studenta:**

Filip Pitinac

**Studij:**

Diplomski sveučilišni studij Elektrotehnika

**Mat. br. studenta, godina upisa:**

D-1194, 20.09.2019.

**Turnitin podudaranje [%]:**

6

Ovom izjavom izjavljujem da je rad pod nazivom: **Grafičko sučelje u Matlabu za osciloskop TDS 224**

izrađen pod vodstvom mentora Prof. dr. sc. Kruno Miličević

i sumentora Ivan Biondić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

## SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
<b>1.1. Zadatak završnog rada .....</b>	<b>1</b>
<b>2. SCPI PROTOKOL.....</b>	<b>2</b>
<b>2.1. Potreba za standardizacijom računalom upravljanih mjernih sustava .....</b>	<b>2</b>
<b>2.2. Razvoj SCPI protokola .....</b>	<b>3</b>
<b>2.3. Osnovna sintaksa SCPI protokola .....</b>	<b>4</b>
<b>2.4. Grupe naredbi SCPI protokola.....</b>	<b>9</b>
<b>3. IZRADA GRAFIČKOG SUČELJA U MATLAB-U .....</b>	<b>17</b>
<b>4. POSTAV I TESTIRANJE PROGRAMA .....</b>	<b>32</b>
<b>5. ZAKLJUČAK.....</b>	<b>37</b>
<b>LITERATURA .....</b>	<b>38</b>
<b>SAŽETAK.....</b>	<b>39</b>
<b>ABSTRACT .....</b>	<b>39</b>
<b>ŽIVOTOPIS.....</b>	<b>40</b>

## 1. UVOD

Šezdesetih godina prošlog stoljeća počeo se pojavljivati veliki broj mjernih instrumenata s računalnim upravljanjem koji su koristili širok izbor nestandardnih sučelja i protokola za komunikaciju. To je dovelo do problema sa kompatibilnošću između različitih instrumenata. Kako bi se riješila problematika komuniciranja instrumenata s računalom definirani su određeni standardi. Jedan od dijelova te problematike bilo je uvođenje seta naredbi za komunikaciju instrumenata i računala. To je riješeno uvođenjem SCPI protokola koji je standardizirao naredbe za upravljanje mjernih sustava.

U ovom radu prikazane su osnovne značajke SCPI protokola kako u teoriji tako i u praksi na primjeru izrade grafičkog korisničkog sučelja s pomoću programa MATLAB kojim se može kontrolirati osciloskop TDS 224 proizvođača „Tektronix“. Ovim radom nastoji se u pet poglavlja dati uvid u sintaksu SCPI protokola i potpuni opis izrade jednog sučelja te njegove sposobnosti. Nakon uvoda slijedi drugo poglavlje u kojem je opisana sintaksa SCPI naredbi i grupe naredbi. Treće poglavlje vezano je uz izradu grafičkog korisničkog sučelja u MATLAB-u i njegove mogućnosti. U četvrtom poglavlju opisan je postupak testiranja programa. U zadnjem poglavlju nalazi se zaključak rada.

### 1.1. Zadatak završnog rada

Objasniti osnovne mogućnosti grafičkog sučelja u Matlabu. Objasniti osnovnu sintaksu protokola za komunikaciju računala s osciloskopom(SCPI). Izrada GUI-a za osciloskop TDS 224. Testirati izrađeni program. .

## 2. SCPI PROTOKOL

### 2.1. Potreba za standardizacijom računalom upravljanih mjernih sustava

Komercijalni mjerni instrumenti s računalnim upravljanjem, uvedeni u 1960-ima, koristili su širok izbor nestandardnih sučelja i protokola za komunikaciju. Institut inženjera elektrotehnike i elektronike 1975. godine odobrio je IEEE 488-1975. IEEE 488 definirao je standardne električna i mehanička sučelja za priključke i kabele. Također je definirano rukovanje, adresiranje i opći protokol za prijenos pojedinačnih bajtova podataka između instrumenata i računala. Ovaj je standard ažuriran i sada je pod nazivom IEEE 488.1-1987. Iako je riješen problem kako poslati bajtove podataka između instrumenata i računala, IEEE 488 nije odredio značenja bajtova podataka. Proizvođači instrumenata slobodno su izmišljali nove naredbe prilikom razvijanja novih instrumenata pri čemu su postojale razlike i u formatu podataka koji se vraćaju s instrumenata. Početkom 1980-ih započeo je rad na dodatnim standardima kako bi se odredio način tumačenja podataka poslanih putem IEEE 488 [1].

Međutim, IEEE-488, kako je izvorno definirano, ostavio je neke nejasnoće u specifičnostima interakcije i komunikacije između računala i instrumenata. Iako su ova otvorena pitanja vjerojatno namjeravala dizajnerima instrumenata dati širinu pri razvijanju novih instrumenata, rezultat je bio zbunjenost i problemi s kompatibilnošću instrumenata različitih proizvođača [2].

Godine 1987. IEEE je objavio IEEE 488.2-1987, kodove, formate, protokole i zajedničke naredbe za upotrebu s IEEE 488.1-1987 [1].

IEEE-488.2 pruža minimalan skup mogućnosti između kontrolnih uređaja i mjernih instrumenata, kao i specifičniji sadržaj i strukturu poruka i komunikacijskih protokola. IEEE-488.2 u potpunosti je kompatibilan s IEEE-488.1; upotreba kontrolnih uređaja kompatibilnog sa "488.2" pruža mogućnost korištenja novih protokola dostupnih s instrumentima "488.2", uz zadržavanje mogućnosti komuniciranja i kontrole instrumenata koji odgovaraju "488.1" i pridruženim neuobičajenim značajkama dodanih od strane dobavljača [2].

Ovaj je standard definirao uloge instrumenata i kontrolnih uređaja u mjernim sustavima i strukturirao komunikaciju između njih. Konkretno, IEEE 488.2 opisao je način slanja naredbe za instrumente i kako poslati odgovore kontrolnim uređajima. Definirao je neke često korištene naredbe izričito, ali svakom je proizvođaču instrumenata ostao zadatak da imenuje druge vrste

naredbi i definira njihov učinak. IEEE 488.2 odredio je kako bi se neke vrste značajki trebale primijeniti ako su uključene u instrument. Općenito nije odredio koje značajke ili naredbe trebaju biti implementirane za određeni instrument. Dakle, bilo je moguće da dva slična instrumenta mogu bi biti u skladu s IEEE 488.2, a ipak bi mogli imati potpuno drugačiji set naredbi [1].

## 2.2. Razvoj SCPI protokola

SCPI (engl. *Standard Commands for Programmable Instruments*) novi je naredbeni jezik za kontroliranje instrumenata koji nadilazi IEEE 488.2. razvijen za rješavanje širokog spektra mogućnosti mjernih instrumenata na standardni način. SCPI promiče dosljednost s gledišta programiranja na daljinu, između instrumenata iste klase i između instrumenata s istom funkcionalnom sposobnošću. Za dano mjerenje poput frekvencije ili napona, SCPI definira određeni skup naredbi koji je dostupan za tu funkciju. Dakle, dva osciloskopa različitih proizvođača mogu se koristiti za mjerenje frekvencije na isti način. Moguće je i za SCPI brojač izvršiti mjerenje frekvencije koristeći iste naredbe kao i osciloskop. SCPI naredbe lako se mogu naučiti i koristiti uzimajući u obzir početnike i stručne programere. Upoznavanje organizacije i strukture SCPI-a može se postići znatan porast učinkovitosti tijekom razvoja kontrolnog programa, neovisno o odabranom jeziku kontrolnog programa [1].

Cilj SCPI-a je smanjiti vrijeme razvoja programa automatske testne opreme. SCPI postiže ovaj cilj pružajući konstantno programsko okruženje za kontrolu instrumenata i upotrebu podataka. Ovo dosljedno programsko okruženje postiže se korištenjem definiranih naredbi, odgovora instrumenata i formata podataka na svim SCPI instrumentima bez obzira na proizvođača. Konstantno programsko okruženje koristi iste naredbe i parametre za kontrolu instrumenata koji imaju istu funkcionalnost. Ove programske naredbe i parametri su poslani s kontrolnog uređaja na instrument s pomoću sučelja definiranog sa IEEE 488.1, VXIbus, RS-232C itd. [1].

SCPI instrumenti su vrlo fleksibilni u prihvaćanju različitih formata naredbi i parametara, što pruža lakše programiranje instrumenta. Instrument vraća natrag odgovore kontrolnom uređaju koji mogu biti podaci ili status instrumenta. Format odgovora SCPI instrumenta za određeni upit je dobro definiran i smanjuje napor programiranja za razumijevanje podataka o instrumentu i informacija o statusu. Podaci se mogu formatirati tako da su neovisni o uređaju i mjerenju. Konzistentnost SCPI programa je vertikalna i horizontalna. Vertikalna programska dosljednost definira programske poruke unutar instrumentalne klase. Primjer vertikalne dosljednosti je korištenje iste naredbe za očitavanje istosmjernog napona iz više različitih multimetara. Vodoravna dosljednost



koristi istu naredbu za kontrolu sličnih funkcija preko klase instrumenata. Na primjer, naredba okidača bila bi ista za identičnu funkciju okidača pronađenu među brojačima, osciloskopima, generatorima funkcija itd. [1].

Ključ dosljednog programiranja je smanjenje više načina upravljanja sličnim funkcijama instrumenta. Filozofija SCPI-ja je upravljanje istim funkcijama instrumenta s pomoću istih SCPI naredbi. Da bi se pojednostavilo učenje, SCPI koristi industrijski standard imena i izraze koje podržavaju proizvođač i kupac. SCPI pruža nekoliko različitih razina upravljanja instrumentima. Jednostavne mjerne naredbe korisnicima omogućavaju jednostavno i brzo upravljanje SCPI instrumentom, dok detaljnije naredbe omogućuju tradicionalnu kontrolu instrumenta. SCPI je dizajniran tako da se u budućnosti proširi novim definiranim naredbama bez uzrokovanja problema prilikom programiranja. Kako se uvode novi instrumenti, namjera je održati kompatibilnost programa s postojećim SCPI instrumentima. SCPI ispitni programi dizajnirani za izvođenje s novim instrumentima možda nisu kompatibilni s postojećim instrumentima. Drugim riječima, testni programi su kompatibilni prema gore, ali nisu kompatibilni prema dolje [1].

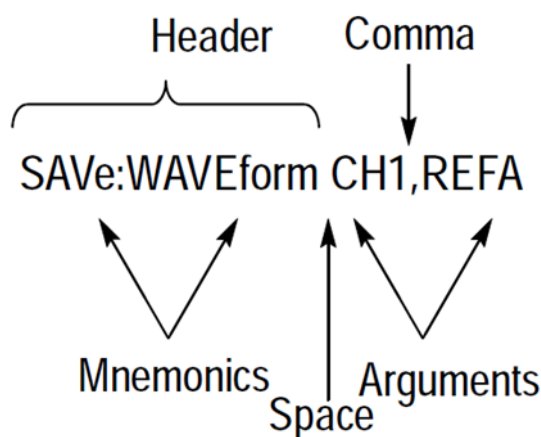
Prednost SCPI-a za programera mjernih sustava je smanjenje vremena za učenje programiranja novih SCPI instrumenata nakon što programiraju svoj prvi SCPI instrument. Programeri koji za slanje naredbi instrumentima koriste programske jezike kao što su BASIC, C, FORTRAN itd. imati će koristi od SCPI-a. Također, programeri koji implementiraju upravljačke programe uređaja ili sučelja instrumenata koristit će SCPI-ove prednosti. SCPI definira instrument, naredbe, parametre, podatke i status. SCPI nije programski paket, programski jezik ili softver namijenjen kontroliranju sučelja instrumenta. SCPI je dizajniran tako da se slojevito nadvisi o hardverskom nezavisnom dijelu IEEE 488.2. SCPI se može koristiti sa sučeljima između kontrolnog uređaja i instrumenta, kao što su IEEE 488.1, VXIbus, RS-232C, itd. [1].

### **2.3. Osnovna sintaksa SCPI protokola**

Osciloskopom se može upravljati putem GPIB-a ili RS-232 sučelja koristeći veliku grupu naredbi i upita. U ovom poglavlju opisana je sintaksa ovih naredbi i upita koje osciloskop koristi za njihovu obradu. Naredbe se šalju na osciloskop koristeći poboljšano ASCII (engl. *American Standard Code for Information Interchange*) kodiranje znakova [3].

Naredbe se sastoje od postavljenih naredbi i naredbi upita (obično jednostavno se nazivaju naredbe i upiti). Naredbe mijenjaju postavke instrumenta ili izvršavaju određenu radnju. Upiti uzrokuju vraćanje podataka s osciloskopa i informacija o njegovom statusu. Većina naredbi ima i

postavljenu formu i formu upita. Oblik naredbe upita jednak je postavljenom obliku, osim što završava s upitnikom. Na primjer, naredba za postavljanje „RS232:BAUD“ ima obrazac upita „RS232:BAUD?“. Naredba upita može dostaviti i podatke o zaglavlju. To se postiže dodavanjem naredbe „HEADer“ u zaglavlje. Isključivanjem naredbe „HEADer“, dobiva se samo povratna informacija o stanju instrument, ali ne i o zaglavlju izvršene naredbe. Nemaju sve naredbe obje forme, neke imaju samo formu postavljene naredbe, a neke samo formu upita. Neke naredbe istovremeno vrše promjenu postavki instrumenta i upit radnje. Na primjer, „\*CAL?“ je naredba koja pokreće program za samokalibraciju na osciloskopu, zatim vraća rezultat kalibracije. Naredbena poruka je ime naredbe ili upita, a slijede je bilo koje informacije koje osciloskopu trebaju da izvrši naredbu ili upit. Naredbene poruke sastoje se (Slika 2.1. [3]) od zaglavlja (engl. *Header*), mnemonika (engl. *Mnemonic*), argumenta (engl. *Argument*), zareza (engl. *Comma*) i razmaka (engl. *Space*) [3].



**Slika 2.1.** Elementi naredbene poruke

Zaglavlje (engl. *Header*) je osnovno ime naredbe. Ako se zaglavlje završi sa upitnikom, naredba je upit. Zaglavlje može započeti znakom dvotočke. Ako je naredba spojena s drugim naredbama potreban je početni znak dvotočke kao što je vidljivo na slici 1. Početni znak dvotočke se nikad ne može koristiti sa naredbama zaglavlja koje započinju zvjezdicom (\*) [3].

Mnemonik (engl. *Mnemonic*) je pod funkcija zaglavlja. Neka zaglavlja imaju samo jednu mnemoničku naredbu. Ako naredbeno zaglavlje ima više mnemonika, oni su uvijek međusobno razdvojeni znakom dvotočke [3].

Argument (engl. *Argument*) Količina, kvaliteta, ili ograničenje koje po potrebi dodatno definiraju naredbe u zaglavlju. Nemaju sve naredbe argument, a neke naredbe imaju više argumenata. Argumenti su odvajaju od zaglavlja razmakom, a jedan od drugog zarezom [3].

Zarez (engl. *Comma*) koristi se između argumenata u naredbama više argumenata. Znakovi razmaka po želji mogu se koristiti prije i poslije zarezom [3].

Razmak (engl. *Space*) koristi se između naredbenog zaglavlja i argumenata. Po želji može se sastojati od više uzastopnih znakova razmaka [3].

Pravila za unošenje naredbi pojedinačno ili ulančano prema [3]:

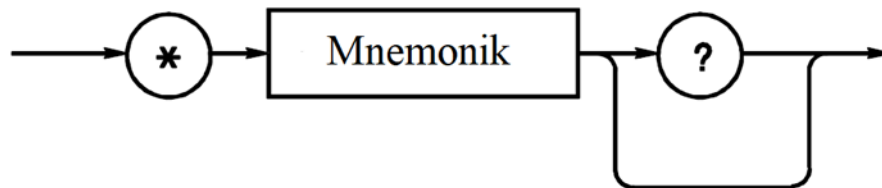
- Unositi naredbe ili velikim ili malim slovima.
- Razmakom se može prethoditi bilo kojoj naredbi.
- Osciloskop ignorira naredbe koje se sastoje samo od razmaka.
- Zaglavlja ulančanih naredbi odvajaju se znakom (;) i početnim znakom pojedinačne naredbe (:).
- Ako se zaglavlja ulančanih naredbi razlikuju samo u zadnjem mnemoniku, moguće je pisati drugu naredbu bez ponavljajućeg mnemonika.
- Naredbama sa zvjezdicom ne prethoditi dvotočkom.
- Ulančaju obavještajnih naredbi, zapis povratne informacije je u jednoj poruci.
- Mogu se ulančati naredbe i upiti,
- Obavještajne naredbe koja vraćaju proizvoljnu vrijednost nalaze se zadnje u lancu.

Naredbe uzrokuju da osciloskop izvrši određenu funkciju ili promijeni jednu od njegovih postavki. Naredbeno zaglavlje sastoji se od jednog ili više raspoređenih mnemonika u hijerarhijskoj ili strukturi stabla. Prvi mnemonik je baza ili korijen stabla i svaki sljedeći mnemonik je nivo ili grana od prethodnog. Naredbe na višoj razini u stablu mogu utjecati na one na nižoj razini. Znak dvotočke uvijek vraća do osnove naredbenog stabla [3].

Pošto se SCPI naredbe temelje na hijerarhijskoj strukturi. To omogućuje da se isto zaglavlje (ključna riječ) instrumenta za upravljanje koristi nekoliko puta u različite svrhe, pod uvjetom da se mnemonik događa u jedinstvenom položaju u hijerarhiji. Odnosno mnemonik se ne sudara s bilo kojom drugim mnemonikom na istoj ili jednoj razini različito tamo gdje postoji zadani čvor.

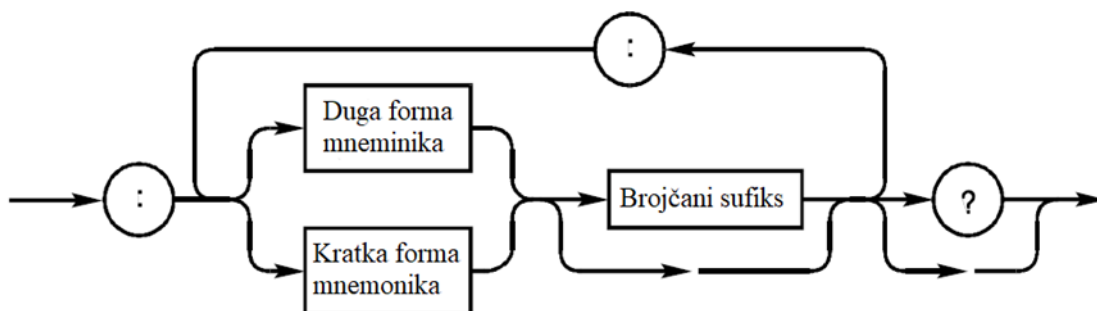
Korištenje hijerarhijskih naredbi trebalo bi eliminirati potrebu za stvaranjem mnemonika iz više riječi [1].

Programska zaglavlja SCPI naredbi možemo podijeliti na zaglavlja osnovnih naredbi, upita i zaglavlja kontrole instrumenta. Sintaksa zaglavlja osnovnih naredbi i upita navedena je u IEEE 488.2 za upotrebu zajedničkim naredbama i upitima definiranim sa IEEE 488.2. Sintaksa im je prikazana na slici 2.2 [1].



Slika 2.2. Sintaksa osnovnih naredbi

Zaglavlja za upravljanje instrumentima koriste se za sve ostale naredbe instrumenta, obično one vezano za kontrolu izvora i mjerenja. Sintaksa zaglavlja za upravljanje instrumentima je prikazana na slici 2.3 [1].



Slika 2.3. Sintaksa naredbi za upravljanje instrumentima

Svako zaglavlje ili ključna riječ za kontrolu instrumenata ima dugu i kratku formu. SCPI instrument će prihvatiti samo točne kratke i točne duge naredbe. Slanije naredbe čije zaglavlje nije kratki niti potpuni dugi oblik rezultira stanjem pogreške SCPI instrumenta. IEEE 488.2 ograničava duljinu zaglavlja na 12 znakova, uključujući bilo koji numerički sufiks koji se može pojaviti.

Zaglavlje dugog oblika je ili jedna riječ ili kratica fraze. Zaglavlje kratkog obrasca kratica je zaglavlja dugog obrasca. Da bi održao dosljedan skup mnemonika, SCPI definira pravila za generiranje mnemonika. Mnemonik dugog oblika generira se iz jedne riječi ili fraze. Ako se koristi jedna riječ tada ta riječ postaje mnemonička. Ako se koristi fraza, tada je mnemonik sastavljen od prvog slova svake riječi i cijele zadnje riječi. Na primjer izraz " relative velocity " generirao bi „RVELOCITY“ kao zaglavlje naredbe dugog oblika. Mnemonic kratkog oblika obično se sastoji od prva četiri znaka naredbe duge forme zaglavlja. Iznimka je kada se dugi obrazac sastoji od više od četiri znaka i četvrti znak je samoglasnik. U takvim slučajevima samoglasnik se spušta i kratak oblik postaje prva tri znaka duge forme. Na primjer, kratki oblik „FREE“ je „FREE“, međutim, kratki oblik „SWEEP“ je „SWE“. Prilikom pisanja mnemonika u literaturi i korisničkim priručnicima često se koristi dugačak oblik ključne riječi sa dijelom kratkog oblika prikazanog velikim slovima, a ostatak ključne riječi prikazan je malim slovima. Tako bi se ključna riječ „relative velocity“ prikazala kao „RVELOCITY“ [1].

Neki mnemotehnički podaci zaglavlja navode jedan od niza mogućih mnemonika. Nazivamo ih formirani mnemonici. Na primjer, mnemotehnika kanala može biti CH<x>. Može se koristiti u naredbi „CH1:VOLTS“ kada se traži intervencija na prvom kanalu instrumenta, odnosno „CH2:VOLTS“ ako je željeni drugi kanal instrumenta [3].

Formirani mnemonici mogu se podijeliti prema [3] na nekoliko podskupina:

- Kanalni mnemonici omogućuju odabir kanala kada se nalaze u zaglavlju CH<x>.
- Mnemonici referentnih valnih oblika omogućuju odabir referentnog valnog REF<x>.
- Mnemonici odabira valnih oblika <wfm>.
- Mnemonici položaja pokazivača POSITION<x>.
- Mnemonici tipa mjerne veličine MEAS<x>.

## 2.4. Grupe naredbi SCPI protokola

Naredbe SCPI protokola na primjeru upravljanja osciloskopom mogu se grupirati te ih možemo prema [3] podijeliti na:

- Akvizicijske naredbe.
- Naredbe za dijagnostiku i kalibraciju.
- Naredbe pokazivača.
- Naredbe prikaza.
- Naredbe tvrdog kopiranja.
- Horizontalne naredbe.
- Naredbe za mjerenje.
- Razne SCPI naredbe.
- RS-232 naredbe.
- Naredbe za spremanje i pozivanje.
- Naredbe stanja i pogrešaka.
- Naredbe okidanja.
- Vertikalne naredbe.
- Naredbe valnih oblika.
- Binarne formate.

Naredbe za akviziciju utječu na dobivanje valnih oblika, odnosno one kontroliraju način prikupljanja, usrednjavanje vrijednost, omotnicu signala i akviziciju valnog oblika. Akvizicijske naredbe prikazane su tablicom 2.1 [3].

**Tablica 2.1.** Akvizicijske naredbe.

Zaglavlje	Opis
<b>ACQUIRE?</b>	Vraćanje unesenih parametara
<b>ACQUIRE:MODE</b>	Postavljanje/vraćanje načina akvizicije
<b>ACQUIRE:NUMAVg</b>	Postavljanje/vraćanje broja akvizicija na srednju vrijednost

<b>ACQUIRE:STATE</b>	Započinjanje ili zaustavljanje akvizicijskog sustava
<b>ACQUIRE:STOPAfter</b>	Postavljanje/vraćanje kontrole akvizicije

Naredbe kalibracije i dijagnostike omogućuju pokretanje rutinske samokalibracije osciloskopa i ispitivanje rezultata dijagnostike osciloskopa. Kalibracijske naredbe prikazane su tablicom 2.2 [3].

**Tablica 2.2.** Kalibracijske naredbe.

Zaglavlje	Opis
<b>*CAL?</b>	Obavlja samokalibraciju i vraća rezultate stanja
<b>CALibrate:ABORT</b>	Zaustavlja tvorničku kalibraciju
<b>CALibrate:FACTory</b>	Započinje tvorničku kalibraciju
<b>DIAG:RESULT:FLAG?</b>	Vraća vrijednosti stanja dobivene dijagnozom sustava
<b>ERRLOG:FIRST?</b>	Vraća prvu vrijednost iz zapisa pogrešaka

Naredbe pokazivača pružaju kontrolu nad pokazivačem osciloskopa te njegovim prikazom i očitavanjem. Naredbe pokazivača prikazane su tablicom 2.3 [3].

**Tablica 2.3.** Naredbe pokazivača

Zaglavlje	Opis
<b>CURSor?</b>	Vraća postavke pokazivača
<b>CURSor:FUNction</b>	Postavlja stanje pokazivač te izabire tip pokazivača
<b>CURSor:HBArs?</b>	Vraća postavke horizontalne trake
<b>CURSor:SElect:SOUrce</b>	Izbor valnog oblika
<b>CURSor:VBArS?</b>	Vraća postavke vertikalne trake

Naredbe prikaza omogućuju promjenu prikazanog stila rešetke, kontrasta i drugih atributa prikaza. Naredbe prikaza prikazane su tablicom 2.4 [3].

**Tablica 2.4.** Naredbe prikaza

Zaglavlje	Opis
<b>DISplay?</b>	Vraća postavke ekrana
<b>DISplay:CONTRast</b>	Podešavanje kontrasta ekrana
<b>DISplay:FORMat</b>	Podešavanje Y-t ili X-Y načina rada
<b>DISplay:PERsistence</b>	Podešavanje vremena
<b>DISplay:STYle</b>	Podešavanje stila valnog oblika

Naredbe na papiru omogućuju kontrolu formata na papiru izlaz i kontrolu pokretanja i zaustavljanja tiskanih kopija. Naredbe tvrdog kopiranja prikazane su tablicom 2.5 [3].

**Tablica 2.5.** Naredbe tvrdog kopiranja

Zaglavlje	Opis
<b>HARDCopy</b>	Početak tvrdog kopiranja
<b>HARDCopy:FORMat</b>	Postavljanje izlaznog formata
<b>HARDCopy:LAYout</b>	Postavljanje orijentacije
<b>HARDCopy:PORT</b>	Postavljanje porta tvrde kopije za izlaz

Horizontalne naredbe kontroliraju vremensku bazu osciloskopa. Moguće je postaviti položaj i vrijeme po podjeli i glavne i vremenske baze prozora. Horizontalne naredbe prikazane su tablicom 2.6 [3].

**Tablica 2.6.** Horizontalne naredbe

Zaglavlje	Opis
<b>HORizontal?</b>	Vraća horizontalne postavke
<b>HORizontal:DELay:POSition</b>	Prozor položaja
<b>HORizontal:DELay:SCALE</b>	Podešavanje vremenskih podioka



<b>HORizontal:MAIn</b>	Postavke osnovne vremenske podjele
<b>HORizontal:POSition</b>	Postavljanje položaja valnog oblika na ekran
<b>HORizontal:RECOrdlength</b>	Vraća snimljenu dužinu valnog oblika
<b>HORizontal:VIEw</b>	Odabir pogleda

Naredbe za mjerenje kontroliraju automatizirani mjerni sustav. Mogu se prikazati do četiri automatizirana mjerenja na zaslonu osciloskopa. Uz prikazana četiri očitavanja mjerenja, naredbe za mjerenje omogućuju i peto mjerenje. Neposredno mjerenje nema ekvivalent na zaslonu, a instrument nikada ne prikazuje neposredna mjerenja zato jer se oni izračunavaju samo kada se zatraže. Neposredna mjerenja uspore brzinu ažuriranja valnog oblika manje od prikazanih mjerenja. Moguće je dodijeliti neke parametre, poput valnog oblika izvora, različito za svako očitavanje mjerenja. Naredbe za mjerenje prikazane su tablicom 2.7 [3].

**Tablica 2.7.** Naredbe za mjerenje

<b>Zaglavlje</b>	<b>Opis</b>
<b>MEASUrement?</b>	Vraća sve mjerene parametre
<b>MEASUrement:IMMed?</b>	Vraća sve trenutne parametre
<b>MEASUrement:IMMed:SOUrce</b>	Uzima trenutne veličine spremljene na nekoj lokaciji
<b>MEASUrement:IMMed:TYPE</b>	Postavka da se mjere samo trenutne veličine
<b>MEASUrement:IMMed:VALue?</b>	Vraća trenutne rezultate mjerenja
<b>MEASUrement:MEAS&lt;x&gt;?</b>	Vraća parametre o periodičkom mjerenju
<b>MEASUrement:MEAS&lt;x&gt;:TYPE</b>	Postavlja tip veličina koje želimo mjeriti

Razne naredbe su skupina naredbi koje ne odgovaraju u bilo koju drugu kategoriju. Prikazane su tablicom 2.8 [3].

**Tablica 2.8.** Razne naredbe.

Zaglavlje	Opis
<b>AUTOset</b>	Automatsko postavljanje instrumenta
<b>FACTory</b>	Vraćanje na tvorničke postavke
<b>HEADer</b>	Postavljanje zaglavlja
<b>ID?</b>	Vraća informacije o identifikaciji
<b>IDN?</b>	Vraća informacije o identifikaciji
<b>LANGUAGE</b>	Postavlja jezik obavijesti
<b>*RST</b>	Resetiranje

Naredbe RS-232 omogućuju postavljanje ili ispitivanje parametara koji kontroliraju RS-232 priključak. Prikazane su tablicom 2.9 [3].

**Tablica 2.9.** RS-232 naredbe.

Zaglavlje	Opis
<b>RS232?</b>	Vraća podatke o parametrima RS-232
<b>RS232:BAUd</b>	Postavlja vrijednost bauda
<b>RS232:HARDFlagging</b>	Postavlja vrstu zastavica
<b>RS232:PARity</b>	Postavlja vrstu parice
<b>RS232:SOFTFlagging</b>	Postavlja vrstu zastavica
<b>RS232:TRANsmit:TERMinator</b>	Postavlja terminaciju RS-232 komunikacije

Naredbe spremi i opozovi omogućuju pohranjivanje i dohvaćanje unutrašnjih podataka valnih oblika i postavki. Prilikom spremanja spremi se većina postavki osciloskopa. Kada se opozove osciloskop vraća se u stanje u kojem je bio prilikom spremanja postavki. Naredbe za spremanje i pozivanje prikazane su tablicom 2.10 [3].

**Tablica 2.10.** Naredbe za spremanje i pozivanje.

Zaglavlje	Opis
<b>*RCL</b>	Vraća određenu postavku
<b>RECALL:SETup</b>	Vrati spremljene postavke instrumenta
<b>*SAV</b>	Spremi postavku
<b>SAVe:SETup</b>	Spremi postavke instrumenta
<b>SAVe:WAVEform</b>	Spremi valni oblik

Naredbe stanja i pogrešaka omogućuju određivanje statusa datoteke osciloskopa i kontrolnih događaja. Prikazane su tablicom 2.11 [3].

**Tablica 2.11.** Naredbe stanja i pogreške

Zaglavlje	Opis
<b>ALLEv?</b>	Povratna informacija o svim događajima
<b>BUSY?</b>	Povratna informacija o dostupnosti instrumenta
<b>*CLS</b>	Čišćenje stanja
<b>EVENT?</b>	Povratna informacija o kodu događaja
<b>EVQty?</b>	Vraća informacija o broju događaja na čekanju
<b>*STB?</b>	Status čitanja bajtova
<b>*WAI</b>	Čekanje za nastavak

Naredbe okidača kontroliraju sve aspekte aktiviranja osciloskopa. Dvije su vrste okidača, okidanje na brid (engl. *Edge*) i video okidanje. Okidanje na brid je zadani tip. Okidač brida omogućuje dobivanje valnog oblika kada signal prolazi kroz razinu napona po izboru. Video aktiviranje dodaje mogućnost aktiviranja na video poljima i linijama. Naredbe okidanja prikazane su tablicom 2.12 [3].

Tablica 2.12. Naredbe okidanja

Zaglavlje	Opis
<b>TRIGger:MAIn:EDGE:COUPling</b>	Postavlja okidanje na brid
<b>TRIGger:MAIn:LEVel</b>	Postavlja razinu okidanja
<b>TRIGger:MAIn:MODe</b>	Postavlja vrstu okidanja
<b>TRIGger:MAIn:TYPe</b>	Postavlja tip okidanja
<b>TRIGger:MAIn:VIDeo?</b>	Povratna informacija o video okidanju
<b>TRIGger:MAIn:VIDeo:SOUrce</b>	Postavlja izvor video okidanja
<b>TRIGger:STATE?</b>	Povratna informacija o statusu sustava okidannja

Vertikalne naredbe kontroliraju attribute kanala. Prikazane su tablicom 2.13 [3].

Tablica 2.13. Vertikalne naredbe.

Zaglavlje	Opis
<b>CH&lt;x&gt;?</b>	Povratna informacija o parametrima kanala
<b>CH&lt;x&gt;:BANdwidth</b>	Postavljanje širine pojasnog propusta
<b>CH&lt;x&gt;:INVert*</b>	Invertiranje kanala
<b>CH&lt;x&gt;:POSition</b>	Postavljanje pozicije kanala
<b>CH&lt;x&gt;SCAle</b>	Postavljanje volta po podioku
<b>MATH?</b>	Povratna informacija o stanju valnog oblika MATH
<b>SElect:&lt;wfm&gt;</b>	Postavljanje stanja prikaza valnog oblika

Naredbe valnog oblika omogućuju prijenos podatkovnih točaka (uzoraka) valnog oblika na i sa osciloskopa. Podatkovne točke valnog oblika zbirka su vrijednosti koje definiraju valni oblik. Jedna vrijednost podataka obično predstavlja jednu podatkovna točka u zapisu valnog oblika. Prije

nego što započne prijenos podataka valnog oblika potrebno je navesti format podataka i lokacije valnih oblika. Naredbe valnih oblika prikazane su tablicom 2.14 [3].

**Tablica 2.14.** Naredbe valnih oblika

<b>Zaglavlje</b>	<b>Opis</b>
<b>CURVe</b>	Prijenos valnog oblika sa ili na osciloskop
<b>DATA</b>	Postavke podatkovnog formata i lokacije
<b>DATA:DESTination</b>	Postavljanje odredišta valnih oblika poslanih na osciloskop
<b>DATA:START</b>	Postavljanje početne točke u slanju valnog oblika
<b>WAVFrm?</b>	Povratna informacija o podacima valnog oblika
<b>WFMPre:XZErO</b>	Postavljanje prve točke valnog oblika
<b>WFMPre:YUNit</b>	Postavljanje vertikalnih jedinica

### 3. IZRADA GRAFIČKOG SUČELJA U MATLAB-U

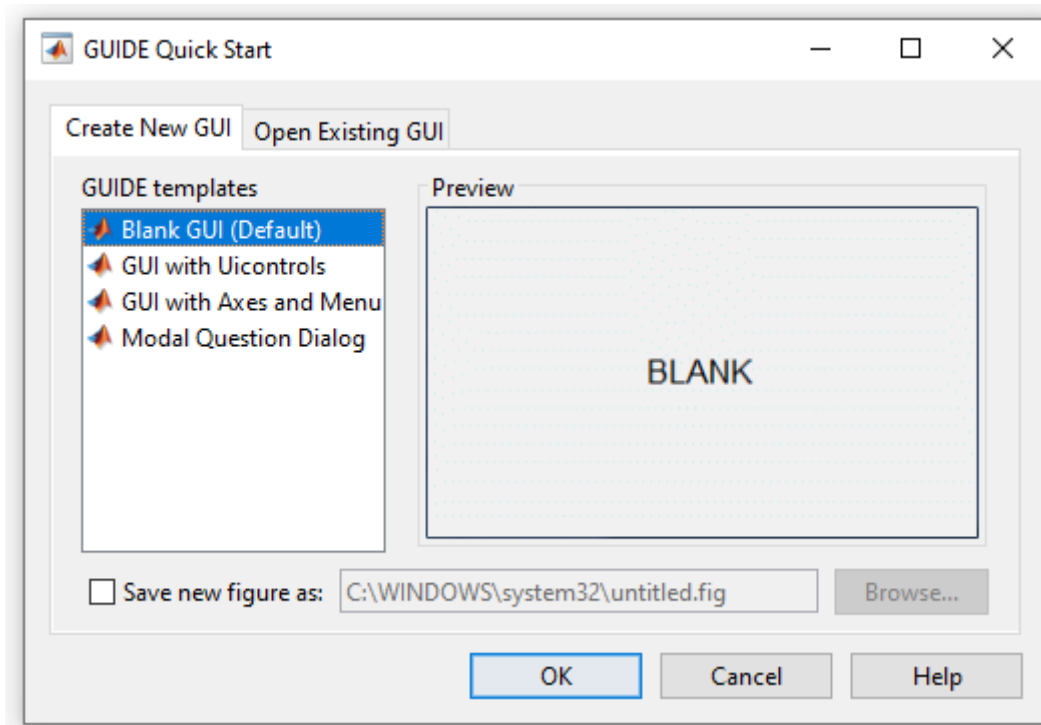
Grafička korisnička sučelja (GUI), također poznata i kao aplikacije, pružaju kontrolu softverskih aplikacija usmjeravanjem i klikom, uklanjajući potrebu da drugi uče jezik ili tipkaju naredbe kako bi pokrenuli aplikaciju. GUIDE je interaktivno okruženje unutar programa MATLAB koje integrira dvije primarne zadaće izgradnje aplikacija. To su postavljanje vizualnih komponenti i programiranje ponašanja aplikacije. Omogućuje brzi prelazak između vizualnog dizajna i razvijanja koda u uređivaču MATLAB. Za dodatnu kontrolu nad dizajnom i razvojem, funkcije MATLAB-a mogu se koristiti za definiranje izgleda i ponašanja aplikacije. Ovom pristupom izrađena FIG-datoteka će poslužiti kao spremnik za korisničko sučelje a komponente se dodaju programiranjem [4].

Prema zadanim postavkama, pri prvom spremanju ili pokretanju aplikacije, GUIDE sprema dvije datoteke:

- FIG-datoteka,
- M-datoteka.

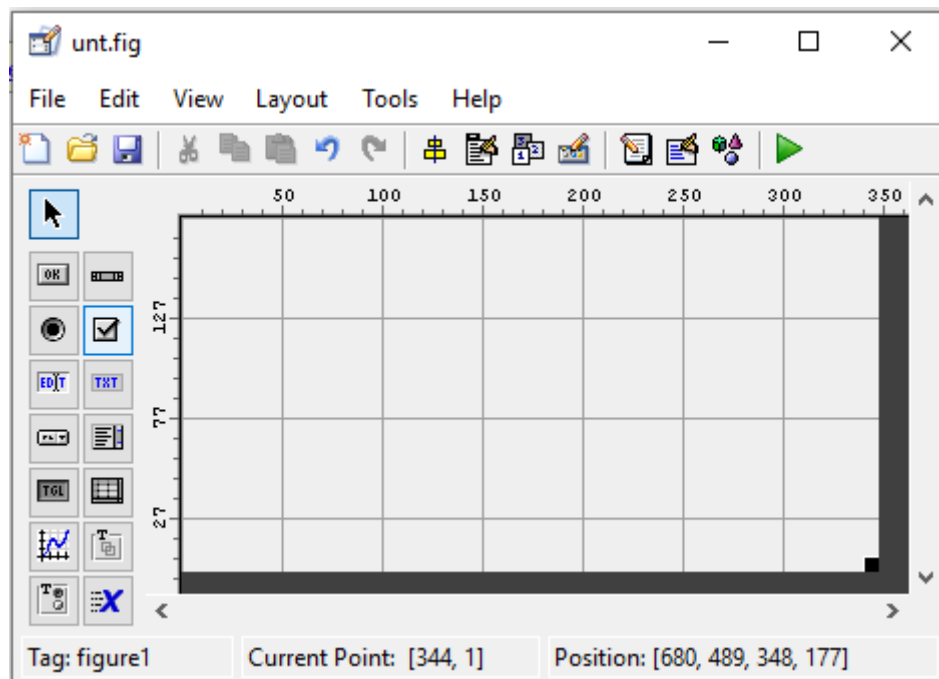
FIG-datoteka s ekstenzijom .fig koja sadrži cjelovit opis izgleda i svake komponente, poput tipki, osi, ploča, izbornika itd. FIG-datoteka je binarna datoteka i ne može se mijenjati osim promjenom izgleda u GUIDE-u. FIG-datoteke su specijalizirane MAT-datoteke. Datoteka koda s nastavkom .m je datoteka koja u početku sadrži inicijalizacijski kod i predloške za neke povratne pozive (engl. *Callbacks*) koji kontroliraju ponašanje. U ovu datoteku obično se dodaju povratni pozivi za komponente u *Layout editoru* [5].

Pokretanje aplikacije GUIDE vrši se upisivanjem naredbe „guide“ u komandni prostor MATLAB-a ili ulaskom u izbornih *New*, klikom na *App* i odabirom naredbe GUIDE. Odabirom naredbe GUIDE pojavljuje se prozor prikazan na slici 3.1. Unutar prozora *GUIDE Quick Start*-a moguće je kreirati praznu fig-datoteku ili otvoriti već postojeću fig datoteku [5].



Slika 3.1. Prozor *GUIDE Quick Start*-a

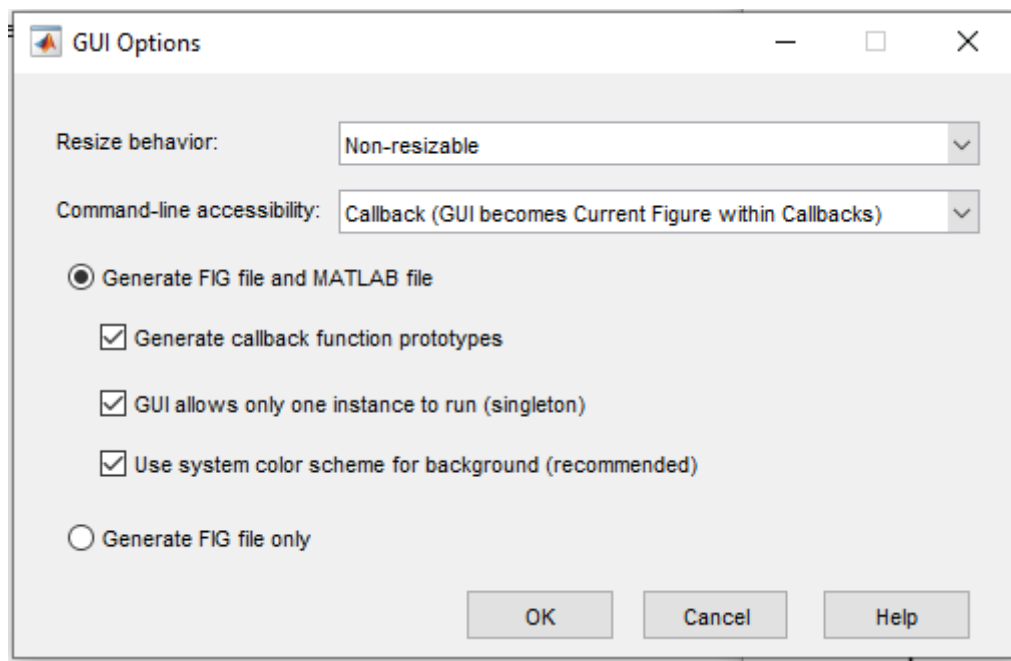
Odabirom prazne fig-datoteke automatski se pojavljuje *Layout Editor* prikazan na slici 3.2. Unutar *Layout editor*-a pristupa se izradi grafičkog dijela aplikacije dodavanjem komponenti i prozora povratnih poziva.



Slika 3.2. *Layout editor*

Prilikom spremanja aplikacije, GUIDE automatski otvara datoteku koda (Slika 3.3) u zadanom uređivaču. FIG-datoteka i kodna datoteka moraju imati isto ime. Te dvije datoteke obično se nalaze u istoj mapi i odgovaraju zadacima postavljanja i programiranja aplikacije. Prilikom izrade aplikaciju u *Layout Editor*-u, komponente i izgled pohranjuju se u FIG-datoteku. Kod programiranja aplikacije, kod se sprema u odgovarajuću datoteku koda [5].

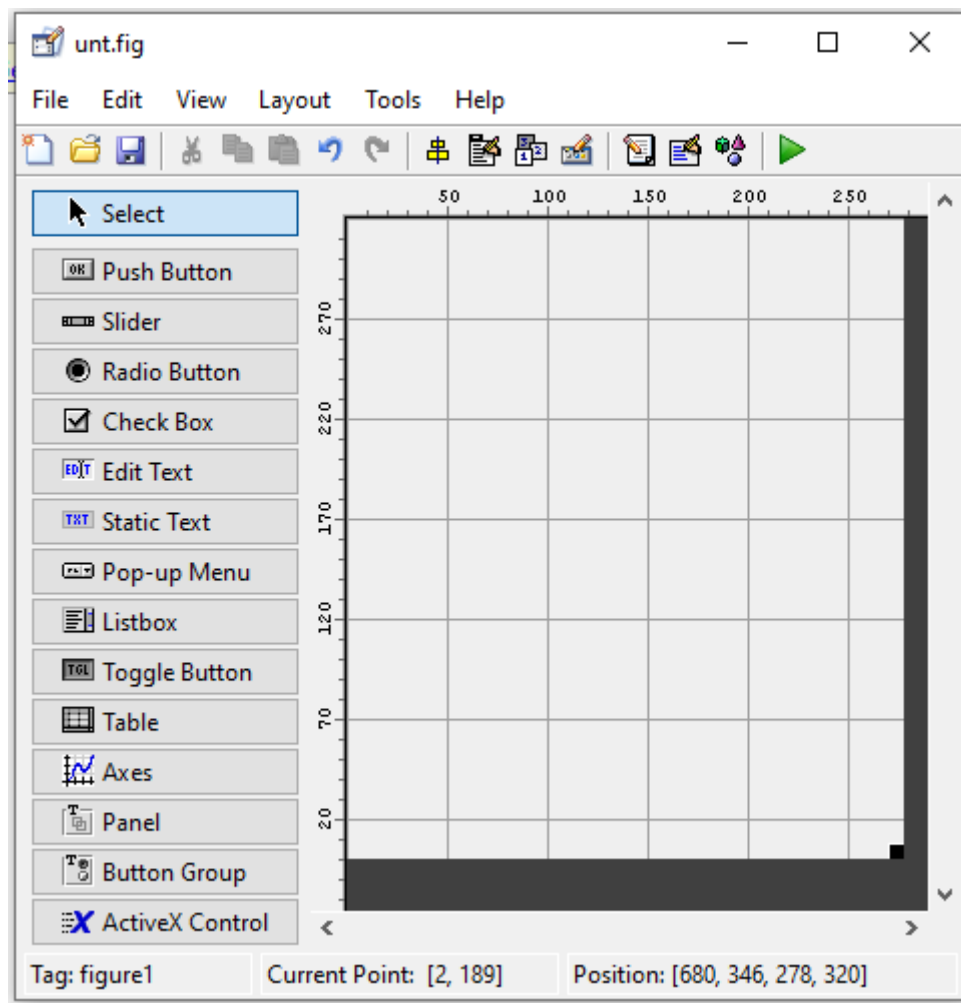
Prije same izrade grafičkog dijela aplikacije potrebno je provjeriti postavke GUI-a. Odabirom padajućeg izbornika *Tools* unutar *Layout Editor*-a klikom na *GUI Options* otvara se prozor prikazan na slici 3.3. Unutar prozora *GUI Options* moguće je postaviti ponašanje promjene veličine komponenata, pristup datoteci koda te datoteke koje MATLAB generira. Postavke je potrebno namjestiti kao na slici 3.3.



Slika 3.3. Opcije GUI-a

Nakon postavljanja željenih postavki pristupa se izradi samog grafičkog sučelja. Radi lakšeg snalaženja unutar *Layout editor*-a moguće je klikom na padajući izbornik *File* odabrati *Preferences* te zatim odabrati opciju *Show names in component palette*. Nakon toga *Layout editor* izgledati će kao na slici 3.4.



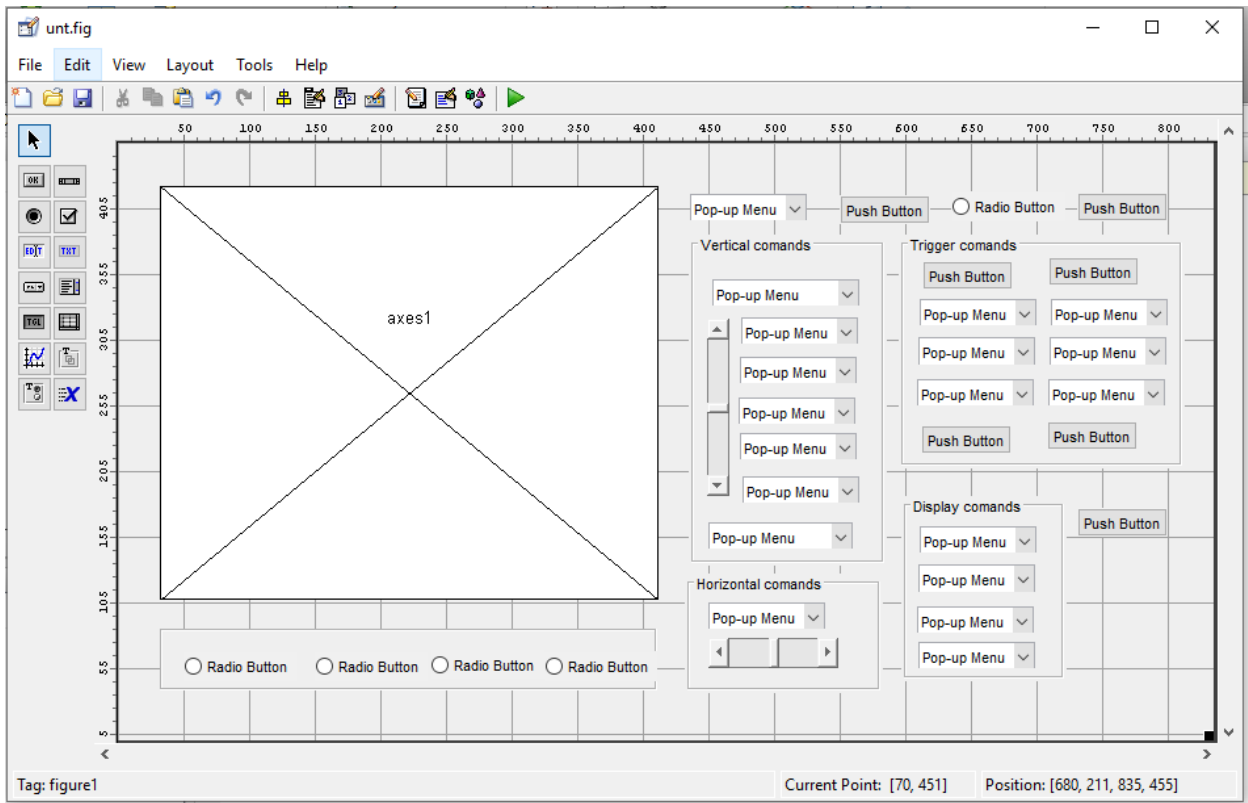


**Slika 3.4.** *Layout editor* sa prikazom naziva komponenata

Radna površina unutar koje će se nalaziti komponente GUI-a moguće je proširiti klikom na crni kvadratić u donjem desnom kutu radne površine. Dodavanje komponenti vrši se jednostavnim povlačenjem željene komponente. Za potrebe željenog programa koristi se komponente:

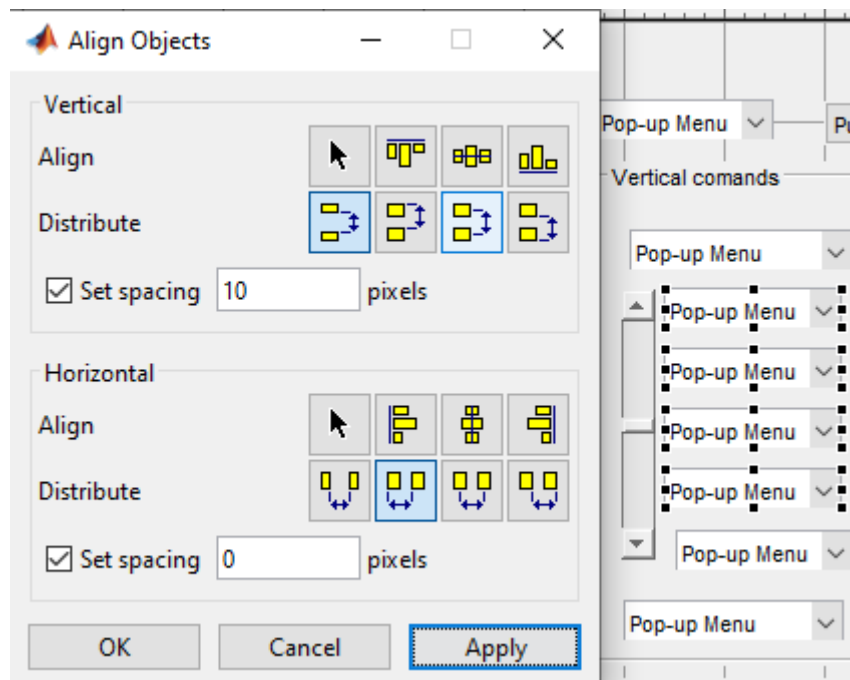
- *Push Button*,
- *Radio Button*,
- *Pop-up menu*,
- *Slider*,
- *Axes*,
- *Panel*.

Nakon dodavanja željenih komponenti *Layout editor* izgleda kao na slici 3.5.



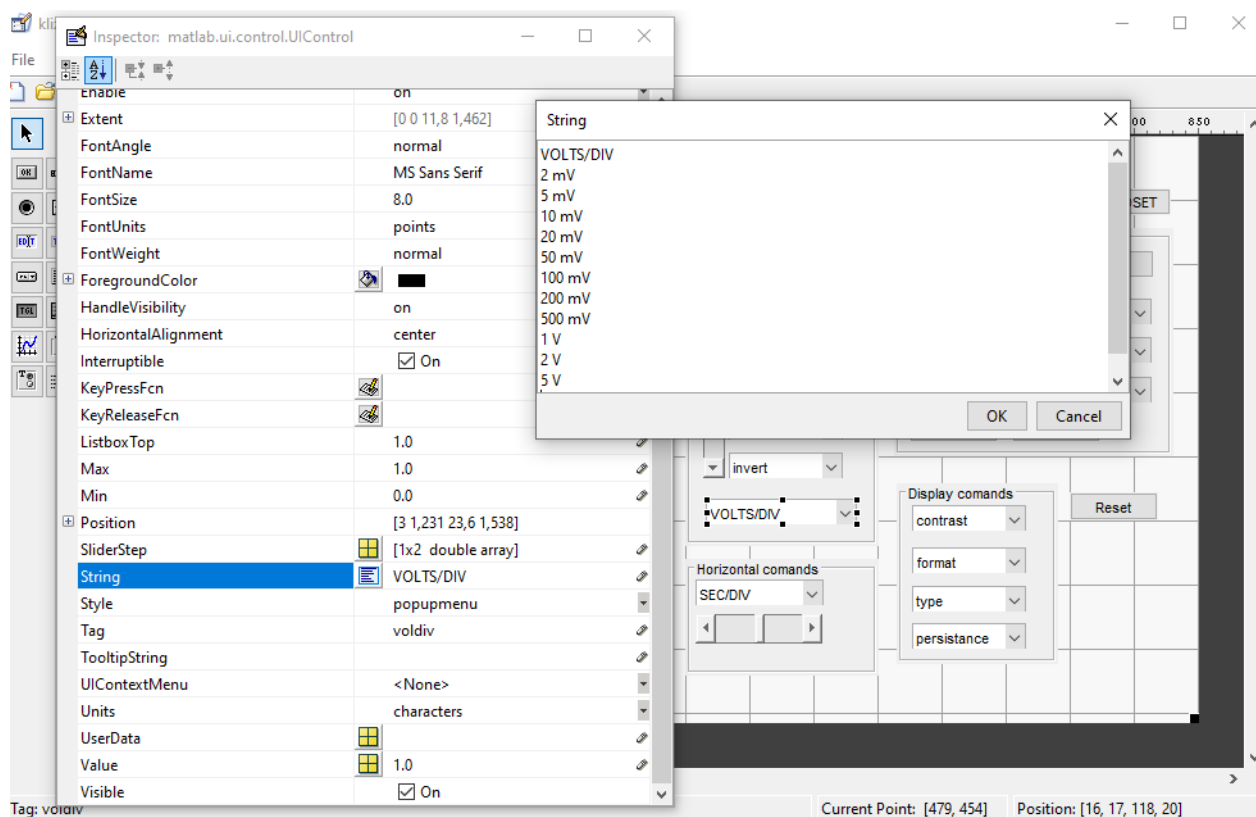
Slika 3.5. *Layout editor* sa željenim komponentama

Nakon prostoručnog dodavanja komponenti pristupa se poravnavanju komponenti. Odabirom ikone *Align Objects* na alatnoj traci *Layout editor*-a otvara se prozor prikazan na slici 3.6.



Slika 3.6. Prozor za poravnanje komponenti

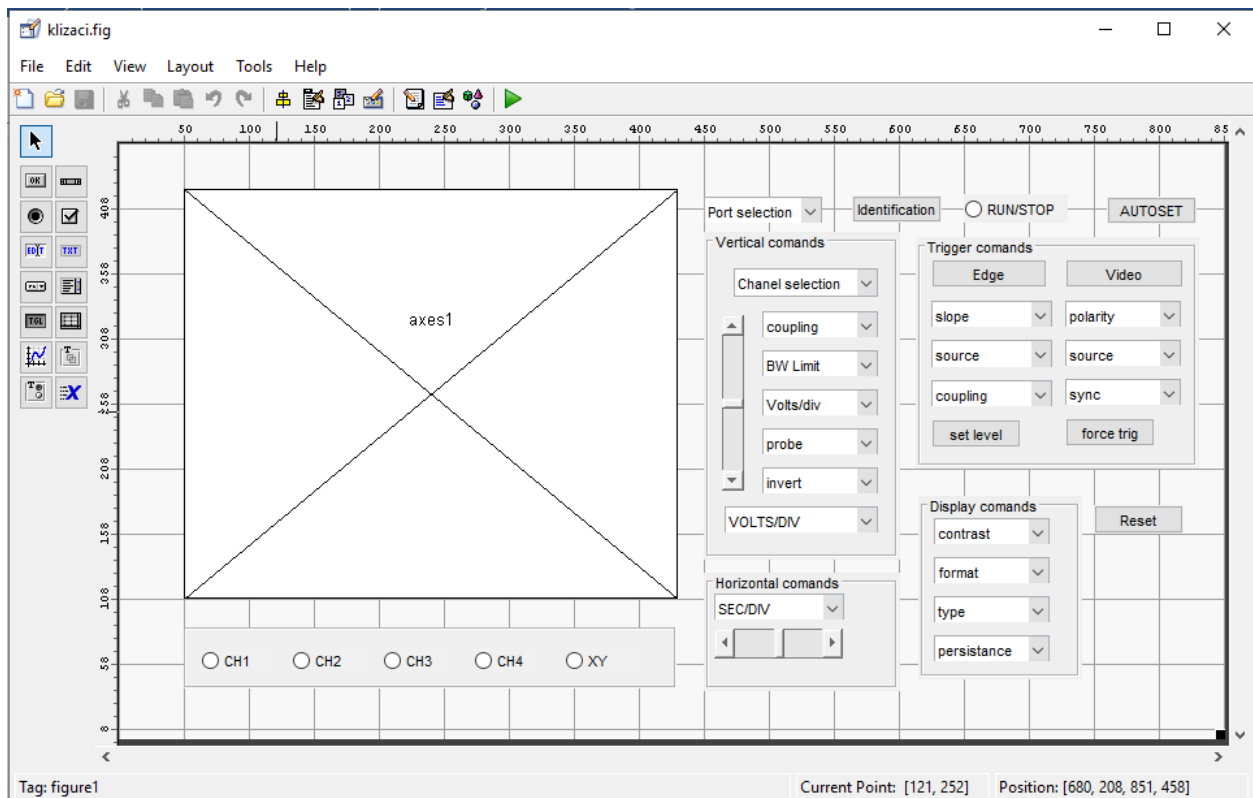
Nakon označavanja željenih komponenti njihov razmak se definira brojem piksela i referentnim položajem komponenti. Slijedeći korak je dodavanje setova podataka unutar komponenti. Dvostrukim klikom na komponentu koju izmjenjujemo pojavljuje se prozor *UI Inspector*-a. Unutar inspektora moguće je izmijeniti grafička svojstva komponenti kao što su boja, font, veličina komponente, vidljivost, tip komponente, poziciju itd. Osim grafičkih svojstava unutar Inspektor-a moguće je dodati setove podataka u obliku *stringo*-va, promijeniti vrijednosna svojstva komponenti i promijeniti *Tag* komponente koji mijenja povratni poziv funkcije koja kontrolira ponašanje komponente. Na slici 3.7. nalazi se primjer unošenja *stringo*-va podataka u komponentu padajućeg izbornika „VOLTS/DIV“.



Slika 3.7. Prozor *UI Inspector*-a

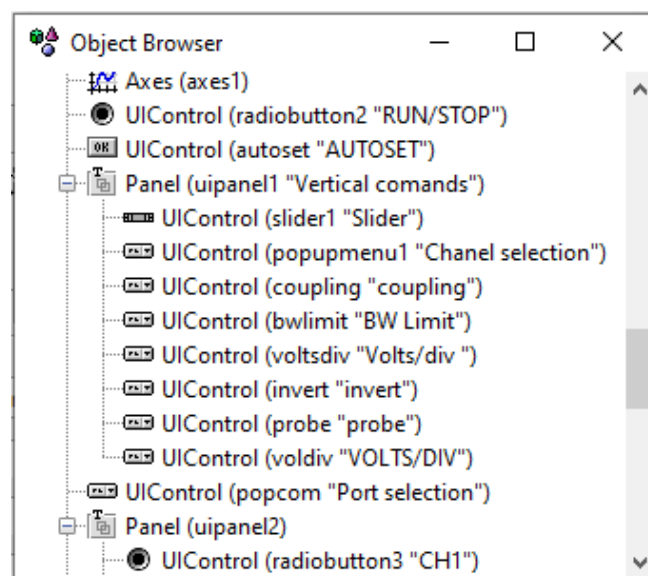
Nakon unosa odgovarajućih *string*-ova za zadane komponente modeliranje grafičkog dijela aplikacije je završeno. Postupci unošenja *string*-ova za svaku komponentu su jednaki zbog čega ih nije potrebno posebno objašnjavati.

Nakon unosa svih *string*-ova konačni raspored *Layout Editor*-a prikazan je na slici 3.8. Izgled *Layout Editor*-a definira konačni izgled korisničkog grafičkog sučelja koji se prikazuje nakon pokretanja programa.



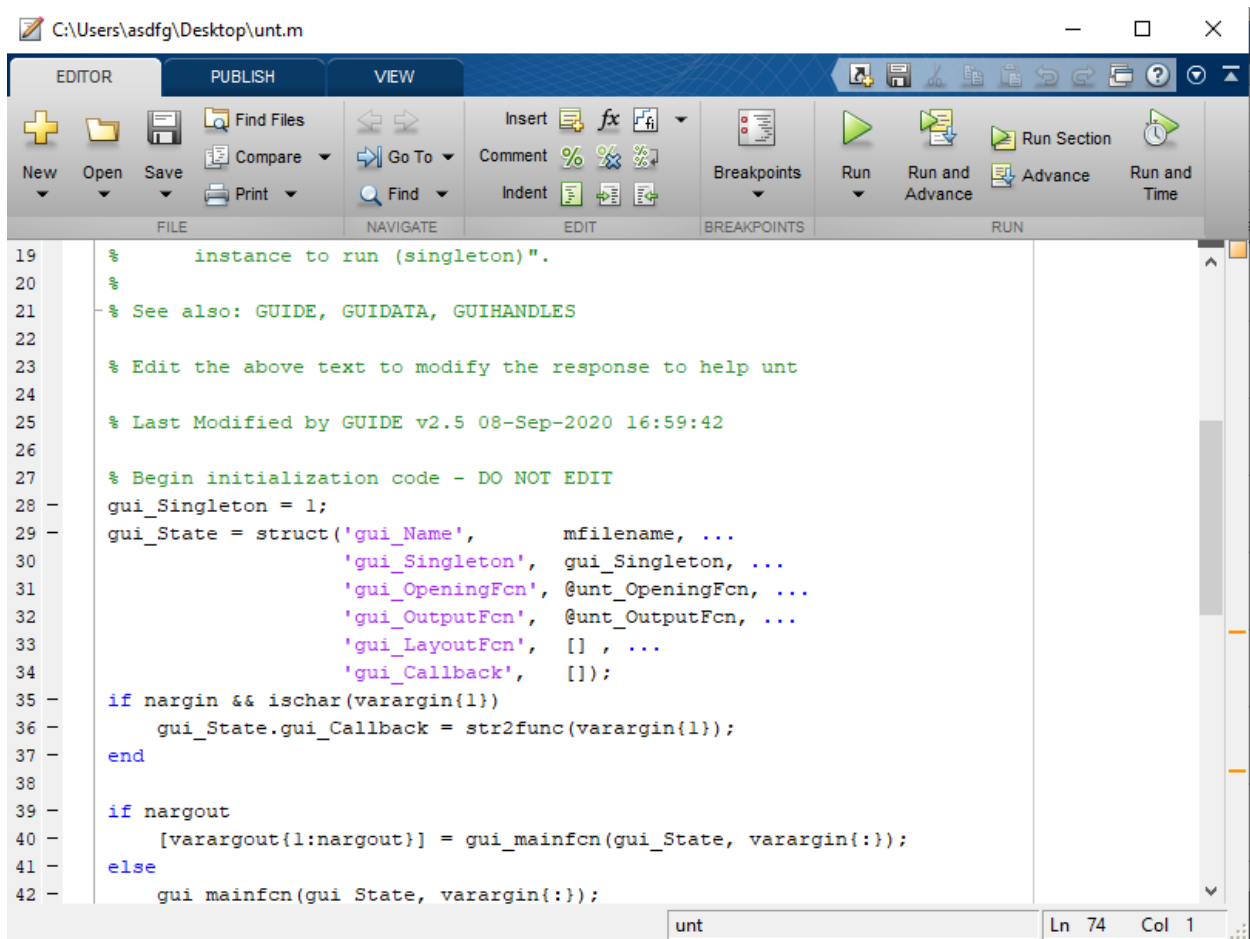
Slika 3.8. Prozor *Layout Editor*-a nakon

Nakon postavljanja rasporeda unutar *Layout Editor*-a pristupa se dodavanju funkcija komponentama grafičkog sučelja. Kod programa sa velikim brojem komponenata radi lakšeg snalaženja preporuča se korištenje *Object Browser*-a prikazanog na slici 3.9. *Object Browser* nalazi se na alatnoj traci *Layout Editor*a. Unutar njega moguće je vidjeti vrstu komponente s pripadajućim *string*-om i *Tag*-om .



Slika 3.9. Prozor *Object Browser*-a

Datoteka koda koju GUIDE generira je datoteka funkcije. Ime glavne funkcije isto je kao i naziv datoteke s kodom. Na primjer, ako je ime datoteke s kodom unt.m, tada je ime glavne funkcije unt. Svaki povratni poziv u datoteci je lokalna funkcija te glavne funkcije [5]. Izgled generirane datoteke koda nalazi se na slici 3.10.



```

19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help unt
24
25 % Last Modified by GUIDE v2.5 08-Sep-2020 16:59:42
26
27 % Begin initialization code - DO NOT EDIT
28
29 gui_Singleton = 1;
30 gui_State = struct('gui_Name',       mfilename, ...
31                  'gui_Singleton',   gui_Singleton, ...
32                  'gui_OpeningFcn',   @unt_OpeningFcn, ...
33                  'gui_OutputFcn',    @unt_OutputFcn, ...
34                  'gui_LayoutFcn',    [], ...
35                  'gui_Callback',     []);
36
37 if nargin && ischar(varargin{1})
38     gui_State.gui_Callback = str2func(varargin{1});
39
40 else
41     gui mainfcn(gui_State, varargin{:});
42

```

Slika 3.10. Datoteka koda

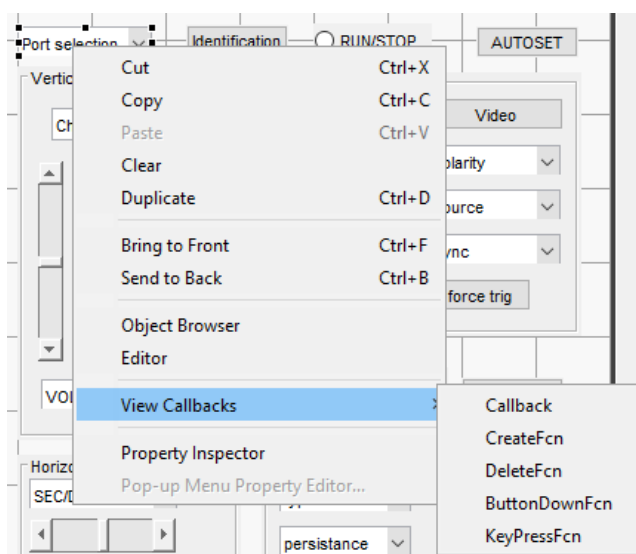
Kad GUIDE generira datoteku koda, ona automatski uključuje predloške za najčešće korištene povratne pozive za svaku komponentu. Datoteka koda također sadrži inicijalizacijski kod, kao i povratni poziv funkcije otvaranja i povratni poziv izlazne funkcije. Potrebno je dodati kod povratnim pozivima komponentata kako bi aplikacija imala traženu funkcionalnost. Također moguće je dodati kod povratnom pozivu funkcije otvaranja i povratnom pozivu izlazne funkcije. Funkcionalnost datoteke koda prikazana je tablicom 3.1 [5].

Tablica 3.1. Sastavnice datoteke s kodom [5].

Odjeljak	Opis
<b>Komentari</b>	Prikazuje se u naredbenom retku kao odgovor na naredbu za pomoć.
<b>Inicijalizacija</b>	Zadaci inicijalizacije GUIDE-a. Ovaj dio koda se ne uređuje.
<b>Funkcija otvaranja</b>	Obavlja zadatke inicijalizacije prije nego što korisnik ima pristup korisničkom sučelju.
<b>Izlazna funkcija</b>	Vraća izlaze u naredbeni redak MATLAB nakon što funkcija otvaranja vrati kontrolu i prije nego što se kontrola vrati u naredbeni redak.
<b>Povratni pozivi komponenata i slika</b>	Kontrolirajte ponašanje prozora i pojedinih komponenata. Softver MATLAB-a poziva povratni poziv kao odgovor na određeni događaj za komponentu ili za sam lik.
<b>Pomoćne funkcije</b>	Razne funkcije koje nisu izravno povezane s događajem za lik ili komponentu.

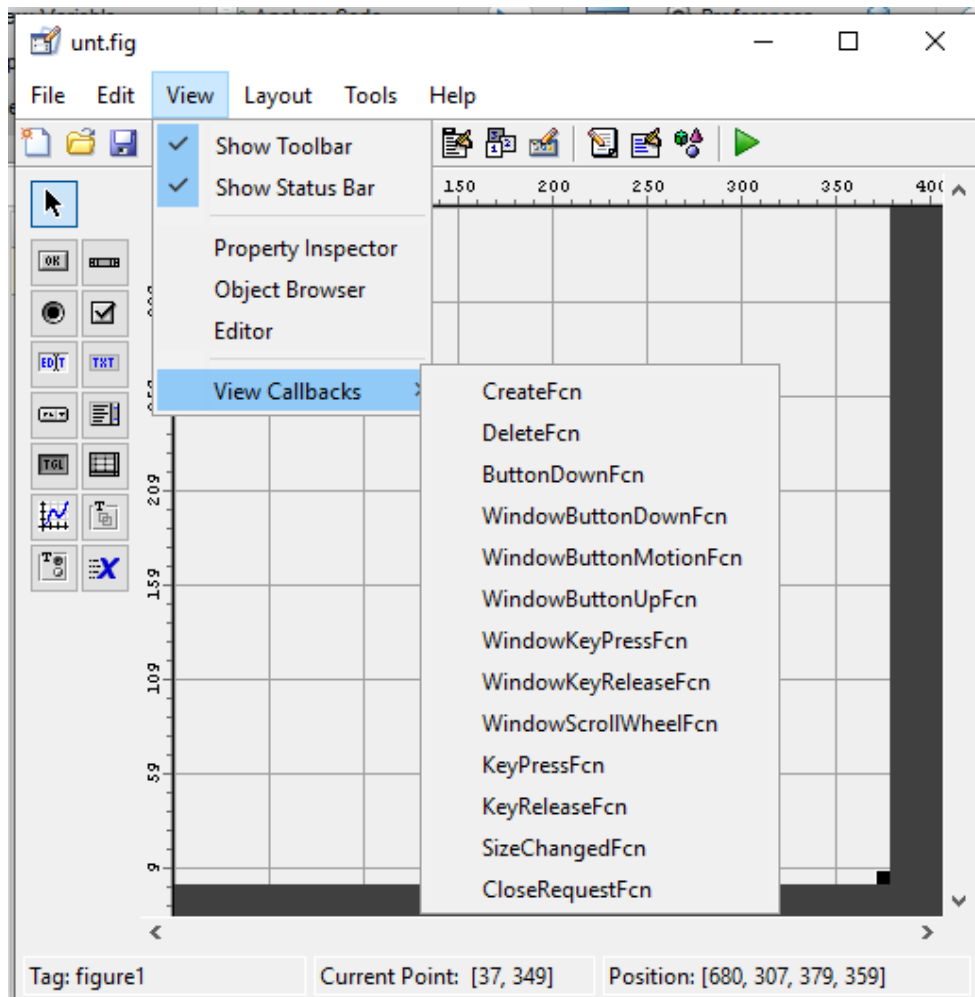
Kod spremanja aplikacije GUIDE automatski dodaje predloške za neke povratne pozive u datoteku koda. Datoteci se po potrebi mogu dodati drugi povratni pozivi. Unutar GUIDE-a kodu se može dodati predložak lokalne funkcije povratnog poziva na nekoliko načina.

Odabirom komponente kojoj je potrebno dodati povratni poziv i desnim klikom na nju otvara se podizbornik. Iz podizbornika klikom na *View Callbacks* moguće je odabrati željeni povratni poziv (Slika 3.11). Klikom na *Callback* GUIDE otvara datoteku koda na mjestu povratnog poziva.



Slika 3.11. Izbornik povratnog poziva komponente

Drugi način dodavanja funkcija GUI-u prikazan je na slici 3.12. Klikom na *View* te zatim na *View Callbacks* moguće je odabrati željeni povratni poziv. GUIDE nudi veliki izbor povratnih poziva koji se mogu pozvati na više različitih načina. Osim osnovnog klika na komponentu povratni poziv može se pozvati i s pomoću tipkovnice, miša itd.

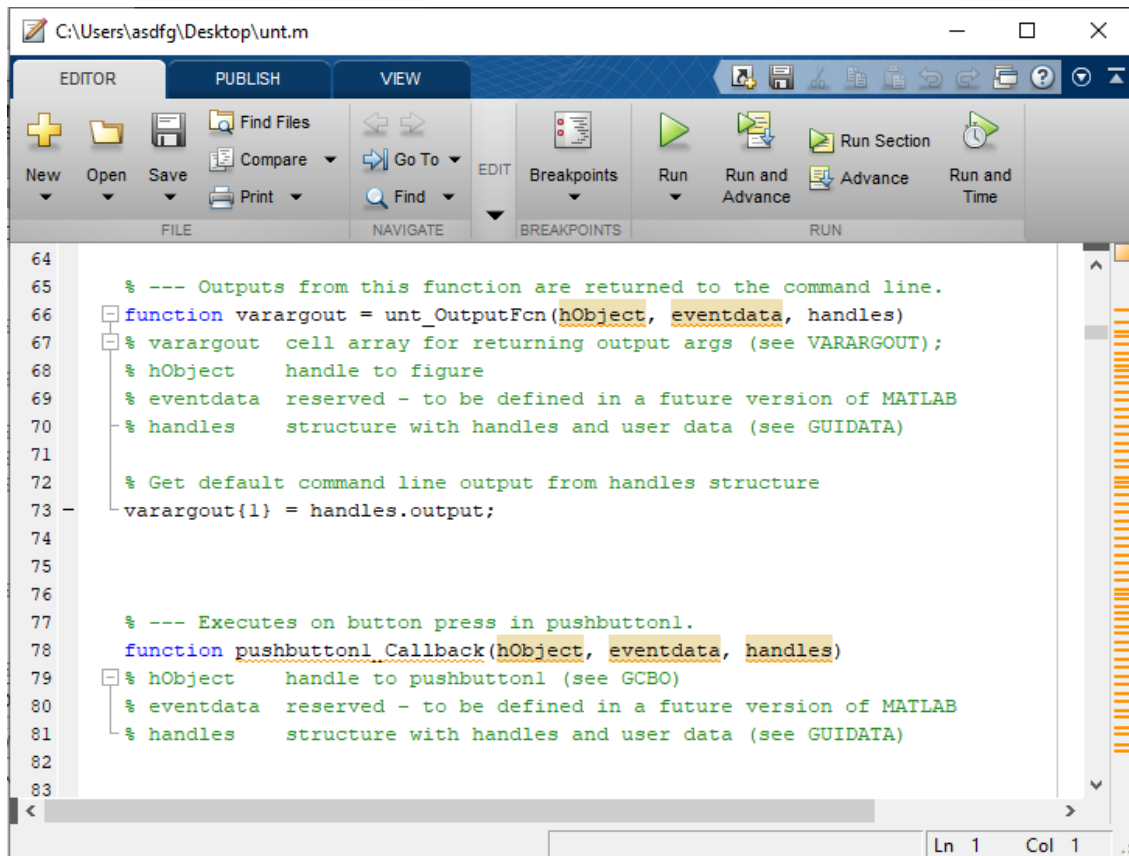


Slika 3.12. Povratni pozivi unutar izbornika *View*

Povratni pozivi mogu se dodavati i pomoću *Property Inspector*-a prikazanog ranije. U *Property Inspectoru* klikom na ikonu olovke i papira pored imena povratnog poziva, povratni poziv pojaviti će se u datoteci koda.

Nakon bilo koje od ovih radnji, GUIDE dodaje predložak za povratni poziv u datoteku koda, sprema ga i otvara za uređivanje na povratnom pozivu koji je upravo dodan. Odabirom povratnog poziva koji trenutno postoji u datoteci koda, GUIDE ne dodaje povratni poziv, već sprema datoteku i otvara je za uređivanje na odabranom povratnom pozivu [5].

GUIDE generira povratne pozive kao predloške funkcija unutar datoteke s kodom. Imenovanje povratnih poziva vrši se na temelju vrste povratnog poziva i svojstva *Tag* komponente. Na primjer, „pushbutton1\_Callback“ je zadani naziv povratnog poziva. Promjenom *Tag*-a komponente, GUIDE će preimenovati sve svoje povratne pozive u datoteci koda kako bi sadržavala novu oznaku. Povratnom pozivu može se promijeniti naziv, zamijeniti ga drugom funkcijom ili ga u potpunosti ukloniti s pomoću *Property Inspector* [5]. Na slici 3.13 nalazi se datoteka koda sa primjerom povratnog poziva za komponentu „pushbutton1“.



```

64
65 % --- Outputs from this function are returned to the command line.
66 function varargout = unt_OutputFcn(hObject, eventdata, handles)
67 % varargout cell array for returning output args (see VARARGOUT);
68 % hObject handle to figure
69 % eventdata reserved - to be defined in a future version of MATLAB
70 % handles structure with handles and user data (see GUIDATA)
71
72 % Get default command line output from handles structure
73 varargout{1} = handles.output;
74
75
76
77 % --- Executes on button press in pushbutton1.
78 function pushbutton1_Callback(hObject, eventdata, handles)
79 % hObject handle to pushbutton1 (see GCBO)
80 % eventdata reserved - to be defined in a future version of MATLAB
81 % handles structure with handles and user data (see GUIDATA)
82
83

```

Slika 3.13. Povratni pozivi unutar izbornika *View*

Nakon brisanja komponente u GUIDE, svi povratni pozivi ostali su u datoteci koda. Ako nijedna druga komponenta ne koristi povratne pozive, tada se može ručno ukloniti kod povratnog poziva [5].



Korisničko sučelje i grafičke komponente imaju određena svojstva koja možete povezati s određenim funkcijama povratnog poziva. Svako od ovih svojstava odgovara određenoj radnji korisnika [6].

Svi povratni pozivi moraju prihvatiti najmanje tri ulazna argumenta prikazana u tablici 3.2.

**Tablica 3.2.** Argumenti povratnih poziva [6].

Argument	Opis
<b>hObject</b>	UI komponenta koja je pokrenula povratni poziv.
<b>eventdata</b>	Varijabla koja sadrži detaljne informacije o određenim radnjama miša ili tipkovnice.
<b>handles</b>	Struktura koja sadrži sve objekte u korisničkom sučelju. GUIDE koristi funkciju „guidata“ za pohranu i održavanje ove strukture.

Da bi funkcija povratnog poziva prihvatila dodatne argumente, dodatni argumenti moraju biti postavljeni na kraju popisa argumenata u definiciji funkcije [6].

Kodiranje ponašanja komponente korisničkog sučelja uključuje određene zadatke koji su jedinstveni za vrstu komponente kojoj se dodaje funkcionalnost. Grafičko sučelje podijeljeno je na nekoliko cjelina. Vertikalne naredbe, horizontalne naredbe, naredbe okidača, naredbe prikaza, naredbe za povlačenje valnih oblika i razne druge funkcije. Zbog velikog broja komponenti prikazane su primjer samo nekih njihovih funkcija.

Prijenos valnog oblika sa osciloskopa vrši se pomoću komponente „radiobutton“. Komponenta ima dva stanja „Max“ i „Min“. Pomoću argumenta „hObject“ dohvaćamo vrijednost trenutnog stanja komponente. Ovisno o stanju komponente vrši se prijenos ili brisanje valnog oblika. Prvo se započinje komunikacija između osciloskopa i računala s pomoću varijable „s“. Varijabla „s“ definira se naredbom „serial“ i nazivom porta preko kojeg se uspostavlja komunikacija. Naredbom „fprintf“ počinje slanje naredbi SCPI protokola osciloskopu. Osciloskopu se šalju SCPI naredbe ovim redoslijedom: pred odabir kanala, postavke enkodiranja, broj bajtova po jednoj točki, početnu i krajnju točku željenog valnog oblika, dohvaćanje pripremnog stanja valnog oblika, dohvaćanje valnog oblika s osciloskopa. Zatim slijedi definiranje varijable „formAY“ u koju se sprema dobiveni niz podataka sastavljen od 2500 točaka. Skalu dobivenog valnog oblika određujemo dohvaćanjem pomoću SCPI naredbi za skaliranje te se ovisno o izvedbi osciloskopa

dobiveni podatci skaliraju i spremaju u varijable „t“ i „a1“. Dobiveni graf sprema se pod varijablu „p1“ kako bi se mogao dohvatiti argumentom „handles“ i obrisati ovisno o stanju komponente.

Programsko rješenje za prijenos valnog oblika s osciloskopa na računalo:

```
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)

if (get(hObject,'Value') == get(hObject,'Max'))

global ComPortName ;
s = serial(ComPortName);
set(s, 'InputBufferSize',8*2500, 'Terminator', 'LF', 'Timeout', 30);
fopen(s)

fprintf(s, 'DATA:SOURce CH1');
fprintf(s, 'DATA:encdng ascii');
fprintf(s, 'DATA:width 2');
fprintf(s, 'DATA:START 1');
fprintf(s, 'DATA:STOP 2500');
fprintf(s, 'WFMPRe?');
form=fscanf(s);
fprintf(s, 'CURVe?');

formaY='';
for i=1:2500
    formaY=[formaY '%d,'];
end
formaY(end)='';
out=fscanf(s, formaY);

fprintf(s, 'horizontal:scale?');
divX=fscanf(s, '%f');
t=[0:2499];
t=t/250*divX;

fprintf(s, 'CH1:scale?');
divY=fscanf(s, '%f');
a1=10*out*divY/(2^16-1)

axes(handles.axes1);
p1=plot(t,a1);grid on;
hold on;
assignin('base', 'p1', p1)
fclose(s)
delete(s)

else
p1=evalin('base', 'p1');
axes(handles.axes1);
delete(p1);
end
```

Druga vrsta komponente korištena u programu naziva se „popupmenu“. Jedna od osnovnih naredbi osciloskopa je promjena skale željenog kanala. Prvo su definirana dva skupa podataka „NapBaza“ i „NapBaza\_1“. Skupove podataka formirani su u tablicu. Prvi skup podataka služi za usporedbu sa *string*-om dobivenim povratnim pozivom komponente dok se drugi skup dodjeljuje SCPI naredbi za promjenu vrijednosti skale. To se postiže s pomoću brojača. Program sadržava dvije globalne varijable koje dodjeljuju kanal i ulaz računala. Nakon dohvaćanja *string*-a i vrijednosti komponente uspostavlja se komunikacija između osciloskopa i računala s pomoću varijable „s“. Brojač uspoređuje sve elemente skupova podataka sa *string*-om dohvaćenim iz komponente. Ako su dohvaćeni *string* i element prvog reda tablice isti u SCPI naredbu upisuje se podatak iz drugog reda tablice koji je definiran po kriterijima SCPI protokola.

Programsko rješenje za mijenjanje skale željenog kanala:

```
function voldiv_Callback(hObject, eventdata, handles)
% hObject    handle to voldiv (see GCBO)

NapBaza = [{'10 mV'} {'20 mV'} {'50 mV'} {'100 mV'} {'200 mV'} {'500 mV'} ...
           {'1 V'} {'2 V'} {'5 V'}];

NapBaza_1 = [{'10e-3'} {'20e-3'} {'50e-3'} {'100e-3'} {'200e-3'} {'500e-3'}
            ...
            {'1'} {'2'} {'5'}];

NapBaza_tablica = [ NapBaza ; NapBaza_1];
contents = cellstr(get(hObject, 'String'));
popvrijeme = contents{get(hObject, 'Value')};

global ComPortName;
global chanelchoice;
s = serial(ComPortName);

set(s, 'InputBufferSize', 8*2500, 'Terminator', 'LF', 'Timeout', 30);

fopen(s)

for str_count = 1 : size(NapBaza_tablica , 2)
    if (strcmp(popvrijeme, NapBaza_tablica(1 , str_count)))
        fprintf(s, [chanelchoice ':SCALE ' NapBaza_tablica{2 , str_count}]);
    end
end

fclose(s)
delete(s)
```

Treća vrsta komponente često korištena u programu je „pushbutton“. To je najjednostavnija komponenta koja klikom na nju izvršava zadani program unutar njene funkcije. Ova vrsta komponente koristi se unutar GUI-a za izvršavanje jednostavnih SCPI naredbi. Jedna od takvih

naredbi je automatsko skaliranje zaslona osciloskopa. Unutar funkcije započinje se s uspostavom komunikacije varijablom „s“ i globalne varijable „ComPortName“ koja pridjeljuje ime ulaza računala. Nakon uspostave komunikacije računalo šalje osciloskopu jednostavnu SCPI naredbu te zatvara komunikaciju.

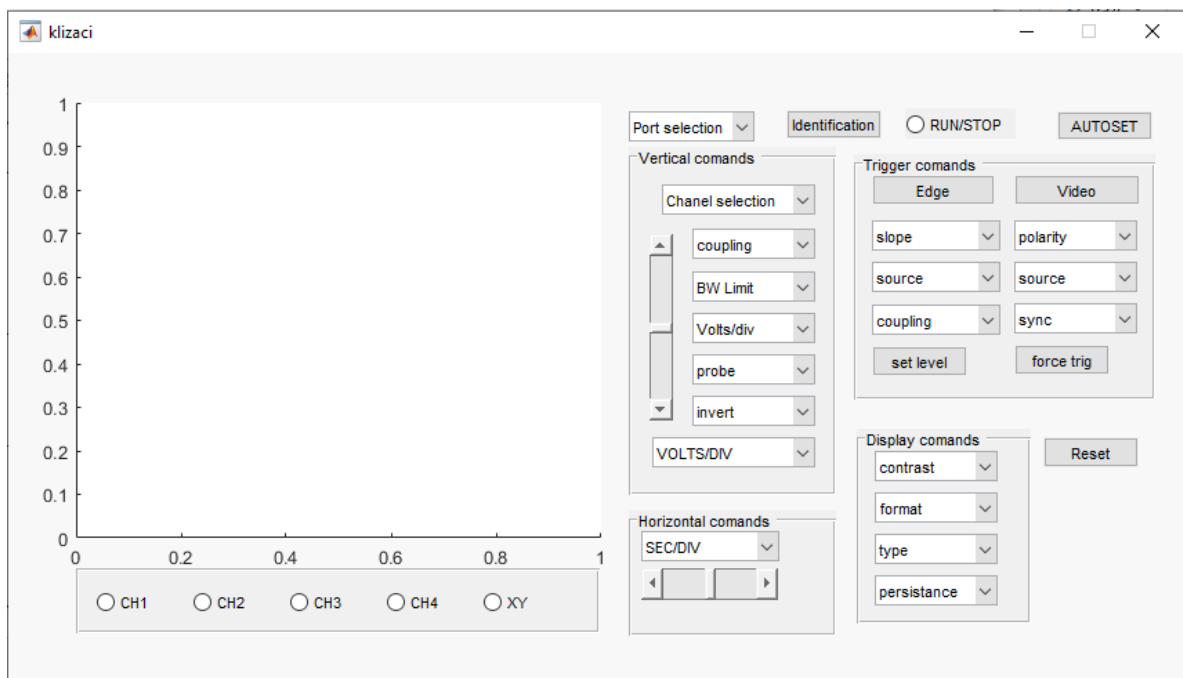
Programsko rješenje za automatsko skaliranje zaslona osciloskopa:

```
function autoset_Callback(hObject, eventdata, handles)
% hObject      handle to autoset (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global ComPortName;
s = serial(ComPortName);

fopen(s)
fprintf(s, 'AUTOSET EXECUTE');

fclose(s)
delete(s)
```

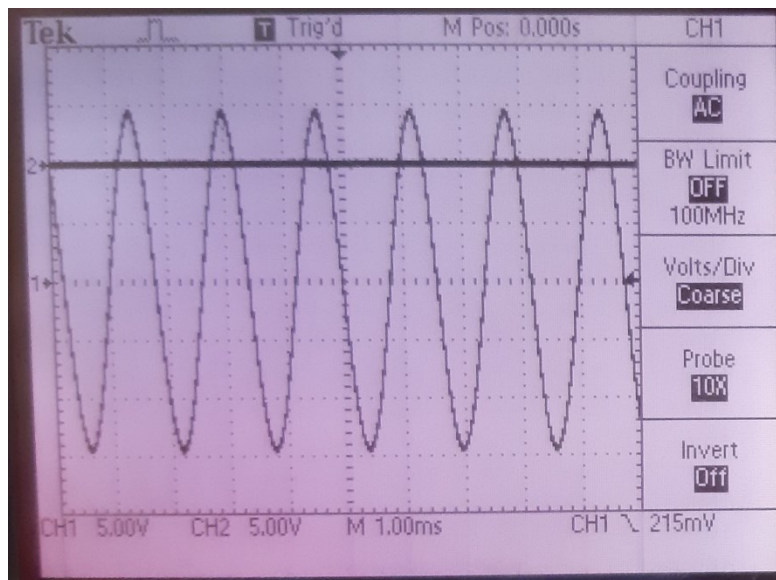
Nakon dodavanja funkcionalnosti svakoj komponenti unutar GUI-a može ga se pokrenuti s pomoću ikone *Run Figure* koja se nalazi na alatnoj traci *Layout Editor*-a. Pokretanjem dobiva se funkcionalno grafičko korisničko sučelje prikazano na slici 3.14. Faza izrade GUI-a je završena i može se započeti s testiranjem njegovih mogućnosti.



Slika 3.14. Grafičko korisničko sučelje za osciloskop tektronix TDS224

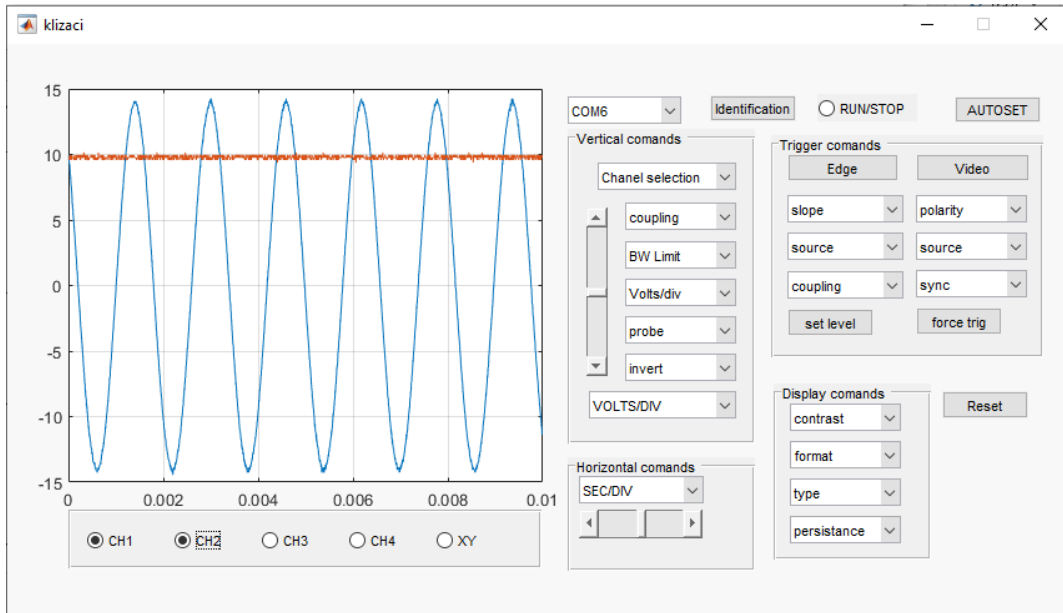
## 4. POSTAV I TESTIRANJE PROGRAMA

Pri testiranju programa prvo je potrebno uspješno uspostaviti komunikacijsku vezu između osciloskopa i računala. Razmjena podataka se vrši s pomoću serijske komunikacije. Sama serijska veza postignuta je adapterom RS232 na USB 2.0. Kako bi se ostvarila komunikacija potrebno je instalirati potrebne *Driver*-e sustava „Windows“. Brzina prijenosa osciloskopa postavljena je na 9600 baud-a. Osciloskop je spojen na frekvencijski generator putem kanala jedan. Zaslom osciloskopa prikazan je na slici 4.1.



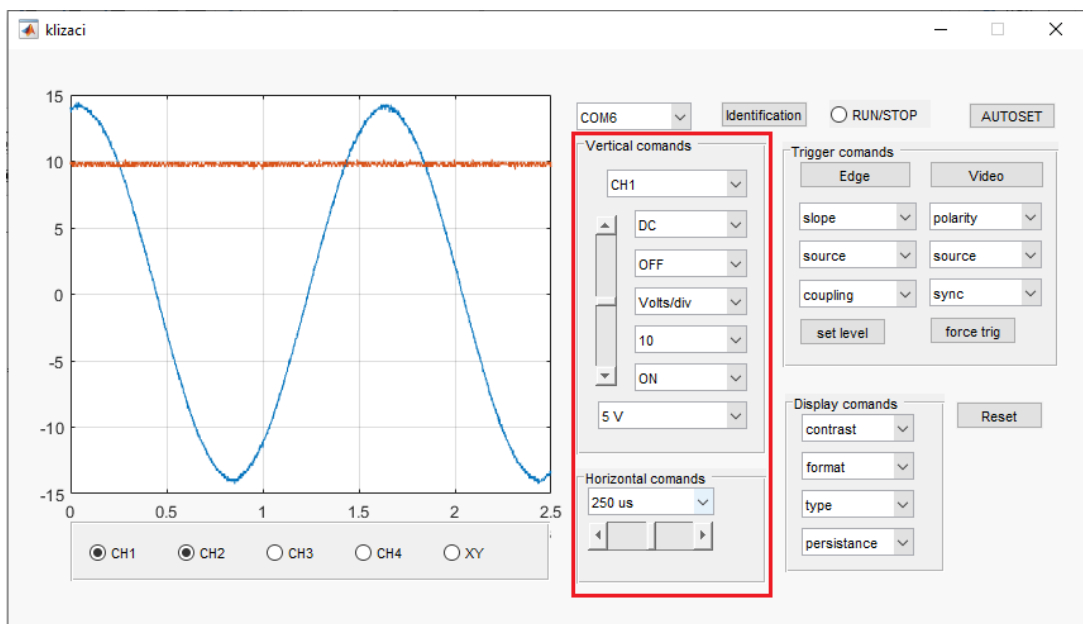
**Slika 4.1.** Zaslom osciloskopa tektronix TDS224

Nakon pokretanja GUI-a može se pristupiti testiranju. Prvo se postavlja padajući izbornik *Port selection* na ime porta kojim je ostvarena komunikacija. To je u ovom slučaju „COM6“. Povezivanje računala s osciloskopom može se jednostavno provjeriti s pomoću tipke *Identification* pri čemu će se unutar naredbenog prozora u MATLAB-u ispisati ime osciloskopa s kojim je komunikacija uspostavljena. Pritiskom na tipke „CH1“ i „CH2“ pokreće se preuzimanje valnih oblika sa prvog i drugog kanala osciloskopa. Valni oblici pojavljuju se u području za crtanje grafova kao što je prikazano na slici 4.2 gdje je prvi kanal označen plavom a drugi kanal crvenom bojom.



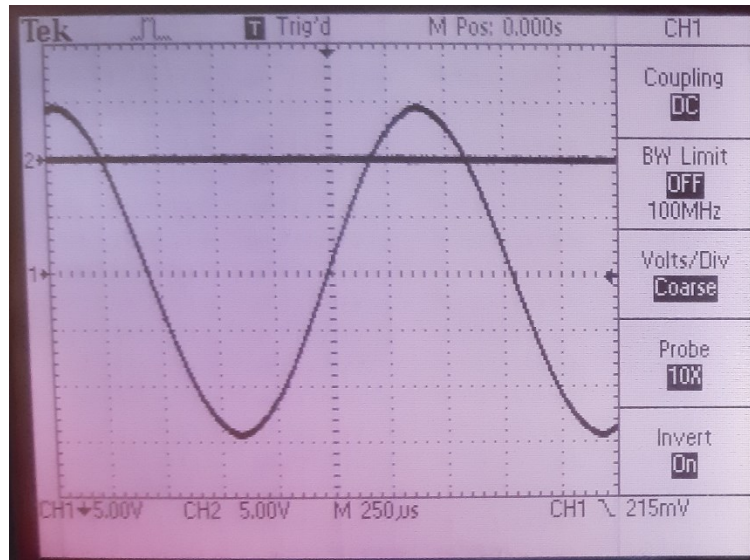
**Slika 4.2.** Prikaz GUI-a nakon preuzimanja valnih oblika prvog i drugog kanala

Nakon uspješno postavljene komunikacije slijedi testiranje vertikalnih i horizontalnih naredbi osciloskopa. Unutar panela *Vertical comands* prikazanog unutar crvenog pravokutnika na slici 4.3 moguće je odabrati željeni kanal i promijeniti njegove postavke. Moguće je mijenjati postavke spajanja, pojasne širine, sonde, invertiranja kanala, vrijednosti napona po podioku i položaja valnog oblika. Panel *Horizontal comands* prikazan unutar crvenog pravokutnika sadrži naredbe za mijenjanje vremenske skale i položaja valnog oblika.



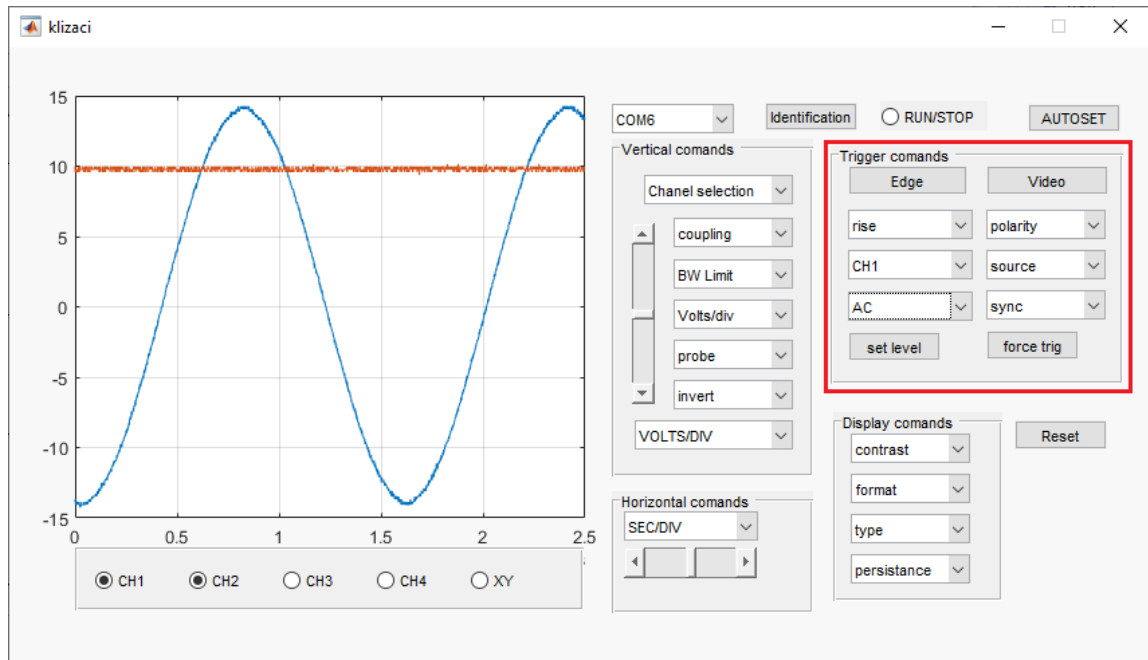
**Slika 4.3.** Prikaz GUI-a nakon promjena vertikalnih i horizontalnih postavki osciloskopa

Nakon odabira novih postavki na izborniku osciloskopa za prvi kanal i nove vremenske skale može se vidjeti da su postavke kanala promijenjene kao što je prikazano na slici 4.4. Ponovnim pritiskom na tipke „CH1“ i „CH2“ brišu se stari valni oblici i dobivaju novi valni oblici identični onima na zaslonu osciloskopa.



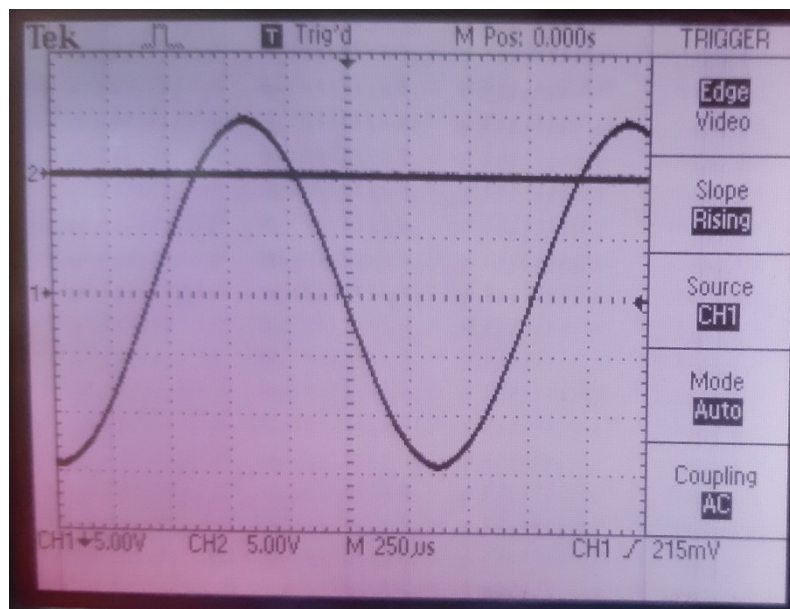
**Slika 4.4.** Zaslon osciloskopa tektronix TDS224 nakon promjene horizontalnih i vertikalnih postavki

Nakon testiranja horizontalnih i vertikalnih naredbi slijedi testiranje okidača osciloskopa. Postavke okidača nalaze se na panelu pod nazivom *Trigger commands*. Postavke okidača označene su na slici 4.4 unutar crvenog pravokutnika. Moguće je mijenjati postavke nagiba, izvora i spajanja ako odaberemo način okidanja na rub, odnosno polarnosti, izvora i sinkronizacije ako se odabere način rada video. Moguće je postaviti razinu i napraviti prisilno okidanje.



**Slika 4.3.** Prikaz GUI-a nakon promjene postavki okidanja osciloskopa

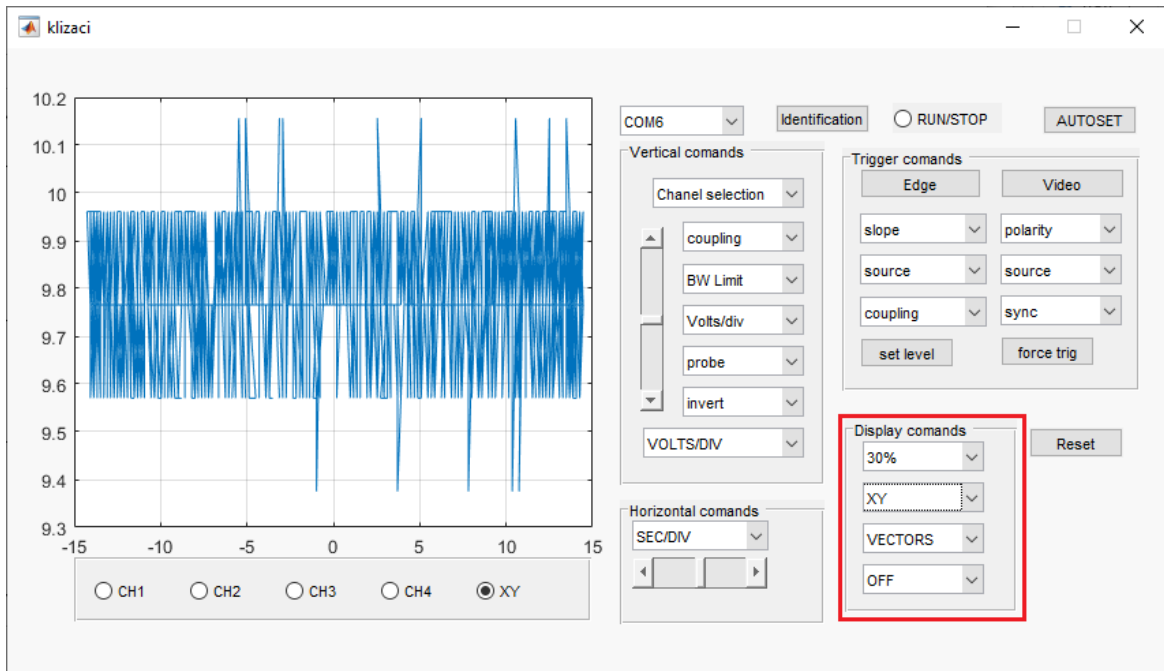
Nakon odabira novih postavki okidača može se vidjeti da su postavke okidača promijenjene kao što je prikazano na slici 4.4. Ponovnim pritiskom na tipke „CH1“ i „CH2“ brišu se stari valni oblici i dobivaju novi valni oblici identični onima na zaslonu osciloskopa.



**Slika 4.4.** Zaslom osciloskopa tektronix TDS224 nakon promjene postavki okidača

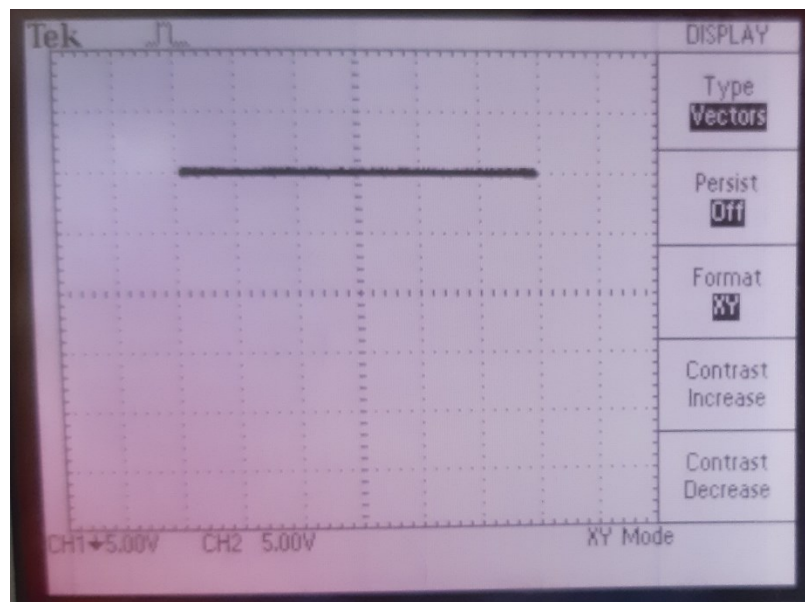
Treći skup komponenti koje treba testirati su naredbe *Display*-a prikazane na slici 4.5 unutar crvenog pravokutnika. Moguće je promijeniti format, tip, perzistenciju i kontrast zaslona.





Slika 4.5. Prikaz GUI-a nakon promjene postavki zaslona osciloskopa

Nakon odabira novih postavki zaslona može se vidjeti da su postavke zaslona promijenjene kao što je prikazano na slici 4.6. Ponovnim pritiskom na tipke „CH1“ i „CH2“ brišu se stari valni oblici i pritiskom na tipku „XY“ dobiva se novi valni oblik identičan onome na zaslonu osciloskopa.



Slika 4.6. Zaslona osciloskopa tektronix TDS224 nakon promjene postavki zaslona

## 5. ZAKLJUČAK

Ovim radom prikazan je proces izrade grafičkog korisničkog sučelja unutar GUID-e MATLAB okruženja. Stečen je uvid u sintaksu SCPI protokola potrebnog za kontrolu osciloskopa putem računala.

Detaljno je opisana tvorba naredbi, grupe naredbi i njihova funkcionalnost. Opisan je proces izrade grafičkog korisničkog sučelja korak po korak sa gledišta izrade grafičkog dijela programa i dodavanja funkcionalnosti komponentama programa u nekoliko primjera.

Izrađeni program daje krajnjem korisniku mogućnost kontrole osciloskopom bez potrebe za znanjem SCPI naredbi.

Rezultat rada je potpuno funkcionalno grafičko korisničko sučelje kojim je moguće mijenjati relevantne postavke potrebne za rad osciloskopa i prijenos podataka sa osciloskopa na računalo što omogućuje daljnju analizu dobivenih valnih oblika. Ovakav oblik obrade podataka ima veliku prednost nad izvođenjem mjerenja samo putem osciloskopa jer mogućnosti analize dobivenih valnih oblika nisu ograničene samim osciloskopom. Mana izrađenog programa je mala brzina prijenosa podataka ostvarena putem serijske komunikacije RS232 standardom. Postoji mogućnost promjene postavljenih vrijednosti brzine komunikacije ili promjene komunikacijskog sučelja (GPIB ili LPT) što bi moglo povećati brzinu prijenosa informacija s osciloskopa na računalo. Ovakav mjerni sustav mogao bi se koristiti u svrhe monitoringa električkih uređaja ili napona gradske mreže.

**LITERATURA**

- [1] Osnovno o SCPI protokolu, sintaksa SCPI protokola:  
<https://www.ivifoundation.org/docs/scpi-99.pdf>(11.9.2020.)
- [2] Kompatibilnost različitih standarda kontrole mjernih uređaja:  
<https://www.omega.de/temperature/z/OverviewIEEE.html>(11.9.2020.)
- [3] Sintaksa SCPI protokola i grupe naredbi: <http://www.tek.com/oscilloscope/tds210-manual/tds200-series-programmer-manual> (11.9.2020.)
- [4] Osnovno o MATLAB-ovom okruženju za izradu grafičkih korisničkih sučelja:  
<https://www.mathworks.com/discovery/matlab-gui.html>(11.9.2020.)
- [5] Osnovno datotekama koje sadrži grafičko korisničko sučelje:  
[https://www.mathworks.com/help/matlab/creating\\_guis/files-generated-by-guide.html](https://www.mathworks.com/help/matlab/creating_guis/files-generated-by-guide.html)(11.9.2020.)
- [6] Povratni pozivi unutar GUID-e okruženja:  
[https://www.mathworks.com/help/matlab/creating\\_guis/write-callbacks-using-the-guide-workflow.html#bq5da4y](https://www.mathworks.com/help/matlab/creating_guis/write-callbacks-using-the-guide-workflow.html#bq5da4y)(11.9.2020.)

**SAŽETAK**

Ovim radom prikazan je proces izrade grafičkog korisničkog sučelja unutar GUID-e MATLAB okruženja. Stečen je uvid u sintaksu SCPI protokola potrebnog za kontrolu osciloskopa putem računala. Detaljno je opisana tvorba naredbi, grupe naredbi i njihova funkcionalnost. Izrađeni program daje krajnjem korisniku mogućnost kontrole osciloskopom bez potrebe za znanjem SCPI naredbi.

**Ključne riječi:** Mjerni instrumenti, osciloskop, SCPI protokol, grafičko korisničko sučelje

**ABSTRACT**

This paper presents the process of creating a graphical user interface within the GUID of the MATLAB environment. An insight into the syntax of the SCPI protocol required to control an oscilloscope via a computer was gained. The formation of commands, groups of commands and their functionality are described in detail. The developed program gives the end user the ability to control the oscilloscope without the need to know SCPI commands.

**Key words:** Measuring instruments, oscilloscope, SCPI protocol, graphical user interface

**ŽIVOTOPIS**

Rođen je 9. studenog 1994. godine u Osijeku. U Osijeku završava 8. razred osnovne škole Antun Mihanović.

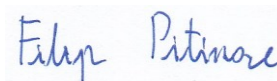
Nakon završene osnovne škole upisuje Tehničku školu i prirodoslovnu gimnaziju Ruđera Boškovića u Osijeku, smjer prirodoslovna gimnazija.

Nakon završetka srednje škole 2013. godine upisuje studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku. Na drugoj godini studija opredjeljuje se za smjer Elektroenergetika. Po završetku preddiplomskog studija upisuje diplomski studij na Elektrotehničkom fakultetu u Osijeku.

U Osijeku, 12. rujna 2020.

Filip Pitinac

Potpis:



---