UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
DEPARTMAN ZA FIZIKU

Vladimir Jokić

Programski paket za prikupljanje podataka i upravljanje difraktometrijskim sistemom za prah Seifert MZ IV

◊

DIPLOMSKI RAD

MENTOR
Dr Srđan Rakić

Novi Sad, 2004.

Predgovor

Tema ovog diplomskog rada je pre svega praktična: izrada programskog paketa za akviziciju podataka i kontrolu difraktometrijskog sistema za analizu praškastih uzoraka sagrađenog oko goniometra Seifert MZ IV.

Stoga je sam tekst napisan kao kratak uvod u oblast rendgenostrukturne analize kristala, tj. istaknute su osnovne osobine X-zraka, njihovo dobijanje i monohromatizacija, opisana je pojava difrakcije na kristalima, a difraktometrijski metod je predstavljen u onoj meri koja je potrebna za upoznavanje sa geometrijskom postavkom eksperimenta. Ovde se mora skrenuti pažnja na razliku koja postoji u obeležavanju uglova u oblastima optike i kristalografije.

Posle toga je izložen deo koji treba da upozna čitaoca sa temeljima računarskih komunikacija i načinima povezivanja koji se trenutno koriste. Najveći deo ove sekcije je posvećen serijskom portu koji je iskorišćen za povezivanje računara i ostatka difraktometrijskog sistema.

U drugom poglavlju je ukratko predstavljen difraktometrijski sistem koji se koristi, tj. svaki od njegovih sastavnih delova. Ni ovaj deo teksta nije previše detaljan, jer se kompletne specifikacije mogu naći u dokumentaciji proizvođača.

Poslednji deo teksta posvećen je kratkom opisu mogućnosti programa, uz napomene koje se odnose na način na koji su pojedine funkcije implementirane. Izvorni kod programa dat je u dodatku A.

Ideju za rad dao je dr Srđan Rakić, a ceo projekat nastao je u okviru katedre za opštu fiziku iz potrebe da se postojeći difraktometrijski sistem dovede u funkcionalno stanje. Pored ovakvog, odmah vidljivog rezultata ovog projekta, iskustvo stečeno u njegovoj izradi može biti iskorišćeno i za druge slične projekte, a sam programski kod je napisan tako da iz njega može proizaći i platforma za drugačije namene. Konkretno, grafički podsistem bi mogao biti dopunjen i modifikovan za analizu dobijenih podataka, čime bi se zaokružio proces merenja i interpretacije rezultata.

Sadržaj

Poglavlje 1

Uvod

1.1 Osobine X-zraka

1.1.1 Otkriće i osobine X-zraka

U zimu 1895. godine Vilhelm Rendgen[1] je, eksperimentišući sa katodnim zracima u Kruksovoj[2] cevi, primetio fluorescenciju ekrana koji nije bio korišćen u eksperimentu i koji se nalazio na nekoj udaljenosti od aparature. Ostavljajući po strani predavanja i ostale dužnosti na univerzitetu u Vircburgu, gde je inače bio rektor, narednih šest nedelja je proveo u laboratoriji radeći sasvim sam i ispitujući osobine zračenja koje je izazivalo fluorescenciju. Prodorna moć ovih zraka bila je vrlo velika, a njihovo poreklo nepoznato pa ih je Rendgen nazvao X-zracima. Već početkom 1896. godine vest o ovom otkriću se proširila širom sveta, izazvavši interesovanje kako u naučnim i stručnim krugovima, tako i u širokoj javnosti. Narednih par godina se osobine X-zraka detaljno ispituju i uvode se razne pretpostavke o njihovom poreklu, a zbog njihovog otkrića Rendgen 1901. godine postaje prvi dobitnik Nobelove nagrade za fiziku.

Danas je poznato da X-zraci, ili rendgenski zraci, predstavljaju elektromagnetno zračenje čije energije leže između energija ultraljubičastih i γ-zraka, tj. sa talasnim dužinama u intervalu od 0.01 nm do 10 nm, odnosno od 0.1 Å do 100 Å (1 Å $= 10^{-10}$ m). Kao i vidljivo, ultraljubičasto i infracrveno zračenje, i X-zraci imaju dualnu prirodu, manifestujući se kao talasi ili čestice. Proučavanje fenomena njihove difrakcije se najvećim delom zasniva na talasnim karakteristikama zračenja.

1.1.2 Dobijanje X-zraka i njihovi spektri

X-zraci nastaju pri naglom usporavanju brzih elektrona. Ovaj efekat se u laboratorijskim uslovima najčešće dobija ubrzavanjem termalnih elektrona sa katode i njihovim udarom u materijal antikatode. Pojave koje nastaju pri naglom usporavanju ovih elektrona su veoma kompleksne i X-zraci rezultuju iz dva opšta mehanizma interakcije elektrona sa atomima materijala mete, čime se dobijaju i dva tipa spektara X-zračenja—karakteristični i kontinualni spektar.

Karakteristični spektar nastaje ukoliko elektron ubrzan u cevi, pri interakciji sa atomom mete izbije neki od elektrona iz blizine jezgra, čime se atom pobuđuje (najčešće se radi o jonizaciji, ali može doći i do ekscitacije). Rekombinacionim procesima elektron sa

[1] Wilhelm Conrad Röntgen
[2] William Crookes

spoljašnje ljuske popunjava upražnjeno mesto usled čega dolazi do emisije zračenja visoke energije. Ovakav način generisanja X-zraka predstavlja kvantni proces identičan procesima nastanka optičkog spektra. Temelje ove teorije postavio je Kosel[1] na bazi Borove[2] atomske teorije i Mozlijevih[3] merenja spektara X-zraka, a do slične ideje je nezavisno došao i Barkla[4].

Brzi elektroni se mogu usporavati i drugačijim tipom procesa. Umesto sudara sa unutrašnjim elektronima atoma mete, upadni elektron se jednostavno može usporavati prolaskom kroz jako električno polje u blizini jezgra atoma, pri tome emitujući tzv. zakočno zračenje. Energija ΔE koju elektron u tom procesu izgubi dovodi do pojave fotona čija je frekvencija data Ajnštajnovom[5] relacijom:

$$h\nu = \Delta E,$$

gde je h Plankova[6] konstanta ($h = 6.662 \cdot 10^{-34}$ Js). Rendgensko zračenje koje nastaje na ovaj način ne zavisi od prirode materijala koji se bombarduje elektronima, i u spektru se javlja se kao kontinualna traka čija je donja granična talasna dužina, tj. kratkotalasna granica funkcija maksimalne energije upadnih elektrona. Po analogiji sa vidljivim delom spektra, ovo zračenje se naziva i belim zračenjem.

Kontinualni spektar

Raspodela energije u kontinualnom spektru određenog elementa dobija se merenjem intenziteta zračenja na različitim talasnim dužinama. Ukoliko se ovo uradi za nekoliko različitih napona rendgenske cevi, dobija se serija krivih prikazana na slici 1.1, na primeru volframa. Nekoliko važnih karakteristika ovog spektra vidljivo je na prvi pogled. Granična talasna dužina i distribucija intenziteta zavise od veličine primenjenog napona. Sa povećanjem napona u cevi, kratkotalasna granica i maksimum krive se pomeraju ka nižim talasnim dužinama, a intenzitet se povećava.

Prema klasičnoj elektromagnetnoj teoriji, ne postoji donja granica talasne dužine zračenja koju mogu proizvesti elektroni pri naglom zaustavljanju. Međutim, kvantna teorija dozvoljava takvu granicu. Ako se elektron ubrza prelazeći razliku potencijala U i ako mu je početna energija bila jednaka nuli, ili zanemarljivo mala u odnosu na onu koju je dobio usled ubrzavajućeg napona, tada je njegova kinetička energija jednaka radu u električnom polju

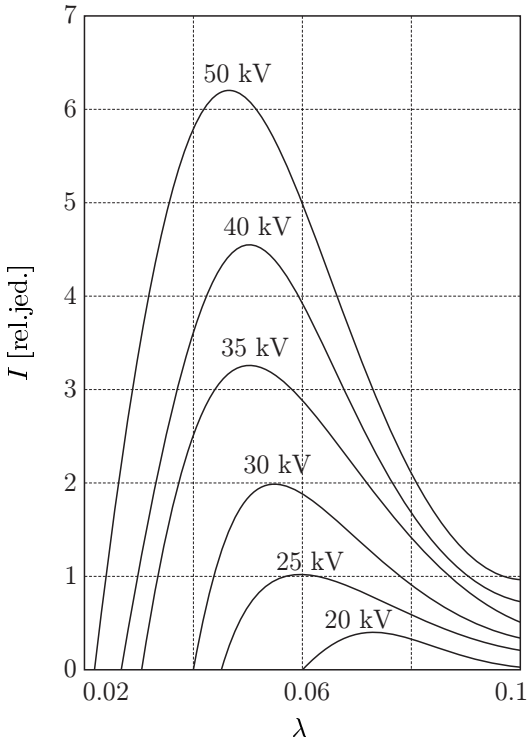
$$\frac{mv^2}{2} = eU,$$

gde je e elementarno naelektrisanje ($e = 1.602 \cdot 10^{-19}$ C).

Elektron može izgubiti celokupnu kinetičku energiju u jednom sudaru, ili u nizu sukcesivnih interakcija. Verovatnoća da se celokupna kinetička energija izgubi odjednom je mala, ali ipak postoji, i foton nastao u ovom procesu će imati maksimalnu energiju ($h\nu_{\max}$), tj. minimalnu talasnu dužinu ($\lambda_{\min}$), te se može pisati:

$$eU = h\nu_{\max} = \frac{hc}{\lambda_{\min}},$$

[1] W. Kossel
[2] Niels Bohr
[3] Henry Moseley
[4] Charles Barkla
[5] Albert Einstein
[6] Max Planck



Slika 1.1: Tipični kontinualni spektri volframa pri različitim naponima rendgenske cevi

odakle je

$$\lambda_{\min} = \frac{hc}{eU},$$

čime se dobija postojanje kratkotalasne granice u saglasnosti sa empirijskom formulom, prema kojoj je

$$\lambda_{\min} = \frac{1239.6}{U},$$

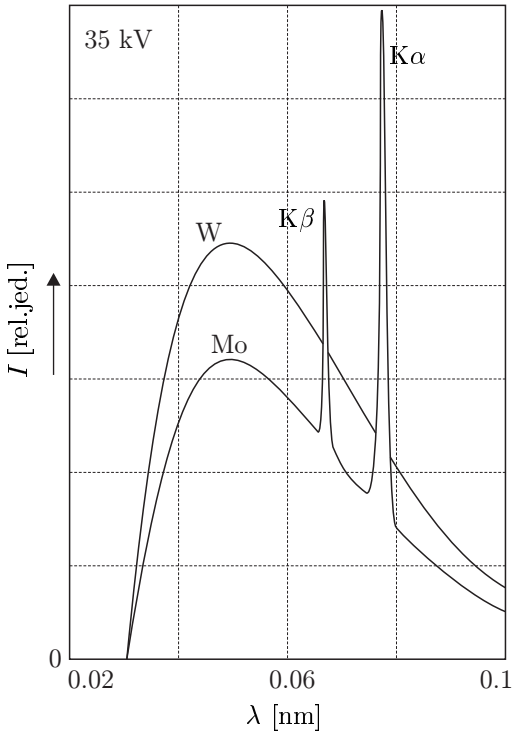
gde se $\lambda_{\min}$ izražava u nm, a U u V.

Maksimum intenziteta u spektru takođe zavisi od napona na cevi, a za vrednost talasne dužine koja odgovara maksimumu zračenja, teorija daje formulu

$$\lambda_{\max} = \frac{3}{2}\lambda_{\min}.$$

Karakteristički spektar

Ako napon cevi, odnosno energija elektrona dostigne ili premaši određenu vrednost karakterističnu za materijal date antikatode (tzv. ekscitacioni potencijal), u spektru X-zraka se pored kontinualnog spektra javljaju i diskretne linije većeg intenziteta. Takve linije čine karakteristični spektar materijala od kojeg je izrađena antikatoda, odnosno, karakteristični spektar atoma antikatode, jer je pokazano da on predstavlja osobina datog elementa, nezavisno od toga u kakvom se hemijskom sastavu element nalazi. Najčešće se karakteristični spektar superponira na kontinualni spektar, što je prikazano na slici 1.2 na primeru molibdena.



Slika 1.2: Kontinualni spektar volframa i karakteristične linije molibdena pri naponu od 35 kV

Linije karakterističnog spektra se najčešće označavaju kombinacijom poput Kα, gde prvo slovo označava krajnji, a drugo slovo početni nivo elektrona čiji prelaz dovodi do emisije; na primer:

- Kα odgovara prelazu L $\rightarrow$ K,
- Kβ odgovara prelazu M $\rightarrow$ K,
- Lα odgovara prelazu M $\rightarrow$ L,

i tako dalje.

Zbog cepanja L nivoa, Kα linija je dubletna, tj. čine je dve veoma bliske linije Kα_1 i Kα_2, čiji se intenziteti odnose kao

$$I(\mathrm{K}\alpha_1) : I(\mathrm{K}\alpha_2) \approx 2 : 1,$$

pa se položaj linije dobija usrednjavanjem:

$$\lambda(\mathrm{K}\alpha) = \frac{2\lambda(\mathrm{K}\alpha_1) + \lambda(\mathrm{K}\alpha_2)}{3}.$$

Povećanjem napona cevi povećava se i intenzitet karakterističnih linija, ali njihova talasna dužina ostaje konstantna. Na primer, za anodu od molibdena, K linije imaju talasnu dužinu oko 0.7 Å, L linije oko 5 Å, a M linije još veću.

Što je veći atomski broj Z materijala anode, to su talasne dužine karakterističnih linija manje (tabela 1.1). Ova zavisnost se izražava Mozlijevim zakonom koji povezuje frekvenciju ν i atomski broj Z:

$$\sqrt{\nu} = K(Z - \sigma),$$

gde su K i σ konstante za određeni tip linije.

Tabela 1.1
Atomski broj, talasna dužina i kritični ekscitacioni napon nekih anoda

Anoda	Atomski broj	Talasna dužina karakterističnih linija i K-apsorpcione ivice [Å]					Kritični ekscitacioni napon
	Z	$\lambda(\mathrm{K}\alpha_2)$	$\lambda(\mathrm{K}\alpha_1)$	$\lambda(\mathrm{K}\alpha)$	$\lambda(\mathrm{K}\beta)$	λ_K	V_K [kV]
Cr	24	2.294	2.290	2.291	2.085	2.070	5.99
Fe	26	1.940	1.936	1.937	1.757	1.743	7.11
Co	27	1.793	1.789	1.790	1.621	1.608	7.71
Ni	28	1.662	1.658	1.659	1.500	**1.488**	8.30
Cu	29	1.544	1.540	**1.542**	**1.392**	1.381	9.89
Mo	42	0.714	0.709	0.711	0.632	0.620	20.0

1.1.3 Monohromatizacija X-zraka

Kao što je rečeno, spektar X-zraka se sastoji od kontinualnog ili belog zračenja i nekoliko karakterističnih linija. Ovakvo, polihromatsko rendgensko zračenje se ređe koristi i uglavnom se u eksperimentalnim metodama zahteva monohromatsko zračenje koje potiče od jedne karakteristične linije. Najčešće su to Kα linije elemenata nabrojanih u tabeli 1.1. Zato je bitno da se dobijeno zračenje monohromatizuje, za šta postoje dve principijelne metode—monohromatizacija filterima i kristalnim monohromatorima.

Monohromatizacija filterima

Ovaj način za dobijanje monohromatskog zračenja zasniva se na pojavi apsorpcije X-zraka. Naime, i pored velike prodorne moći, X-zraci se u interakciji sa materijom apsorbuju, tj. smanjuje se njihov intenzitet, usled sledećih pojava:

- stvarne apsorpcije,
- rasejanja,
- stvaranja para čestica—elektrona i pozitrona.

Smanjenje intenziteta opisuje se dobro poznatom jednačinom:

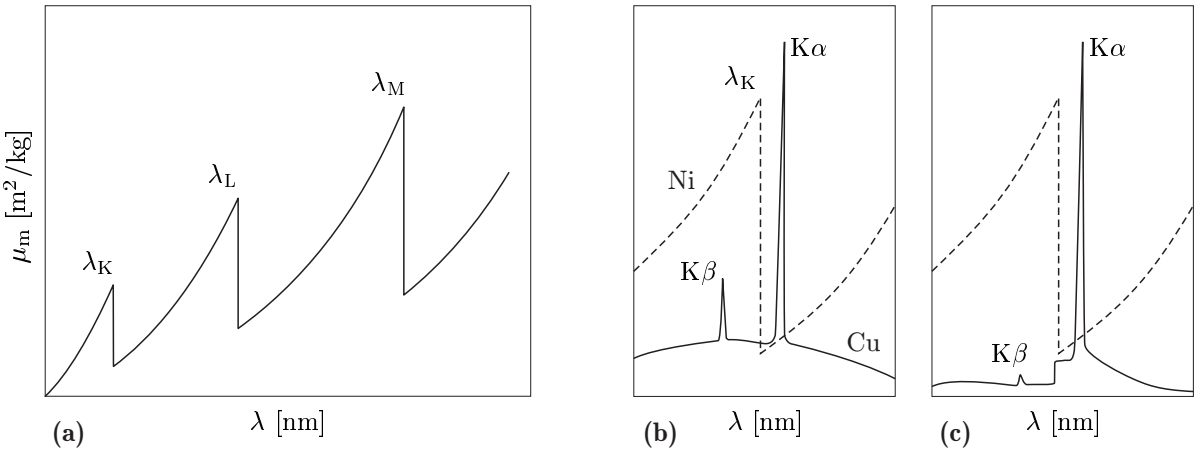
$$I = I_0\,\mathrm{e}^{-\mu x},$$

gde je I_0 intenzitet upadnog snopa, I intenzitet propuštenog snopa, x debljina uzorka, a μ linearni koeficijent apsorpcije. Vidi se da intenzitet zračenja eksponencijalno opada sa debljinom apsorbera.

Kako μ između ostalog zavisi i od prirode supstance, tj. njene gustine, uvodi se maseni apsorpcioni koeficijent, $\mu_\mathrm{m} = \mu/\rho$, pa prethodnu jednačinu možemo pisati kao

$$I = I_0\,\mathrm{e}^{-\mu_\mathrm{m}\rho x}.$$

Zavisnost masenog apsorpcionog koeficijenta prikazana je na slici 1.3a, i sa nje se vidi postojanje oštrih granica u apsorpcionom spektru, tzv. apsorpcionih ivica. Njihovo postojanje se objašnjava time da upravo na tim talasnim dužinama X-zraci imaju dovoljnu



Slika 1.3: Zavisnost masenog apsorpcionog koeficijenta od talasne dužine i princip monohromatizacije filterima

energiju i maksimalni presek za pobuđenje atoma mete, te se najjače apsorbuju. Ako se od ove ivice krene ka većim talasnim dužinama, apsorpcija raste, jer se fotoni manje energije jače apsorbuju. Fotoni energija većih od energije posmatrane apsorpcione ivice će imati sve manji presek za jonizaciju, usled čega će trpeti slabiju apsorpciju, dok se ne dodje do sledeće apsorpcione ivice koja odgovara jonizaciji nekog drugog elektrona.

Činjenica da se položaji apsorpcionih ivica razlikuju za atome sa različitim Z se koristi za izradu filtera, što je prikazano na slici 1.3b i 1.3c. Ukoliko apsorpciona ivica materijala od koga je načinjen filter leži između Kα i Kβ linija atoma antikatode, dolazi do jake apsorpcije upravo Kβ linije, čime se u izlaznom snopu ona skoro potpuno eliminiše. Primer za antikatodu od bakra i filter od nikla je istaknut i u tabeli 1.1.

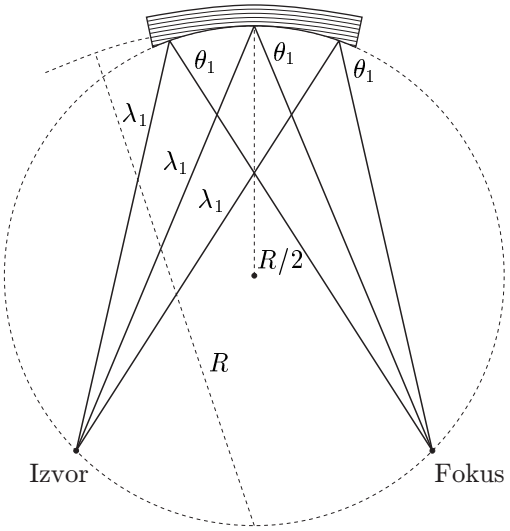
Treba napomenuti da se ovim postupkom još uvek ne dobija striktno monohromatsko zračenje, jer deo kontinualnog spektra i Kβ linije još uvek ostaju prisutni.

Kristalni monohromatori

Za dobijanje sasvim monohromatizovanog zračenja moderni difraktometri uglavnom koriste kristalne monohromatore. Kristalni monohromatori su monokristali koji difraktuju rendgenske zrake određene talasne dužine λ pod odgovarajućim uglom 2θ. Time se dobija izdvojena talasna dužina, ali mogu biti prisutni i viši harmonici talasnih dužina $\lambda/2$, $\lambda/3$ itd., koji potiču od viših redova difrakcije. Njihov intenzitet je dovoljno nizak da uglavnom ne predstavlja smetnju, ali ukoliko je potrebno i oni se mogu ukloniti.

Postoji više izvedbi ovakvih monohromatora, a u difraktometriji praha najčešće se koriste monohromatori sa linijskim fokusom, odnosno savijeni monokristali (slika 1.4). Oni fokusiraju zračenje u određenu tačku, a pored toga daju veći intenzitet snopa od planarnih monokristala. Izrađuju se savijanjem odabranog monokristala tako da paralelne ravni pređu u koncentrične krugove radijusa R, a onda se još i bruse tako da reflektujuća površina ima radijus $R/2$, čime se dobija mnogo bolje fokusiranje negu u slučaju kad se kristal samo savija.

Bilo koji tip monohromatora da je u pitanju, može se postaviti u primarni ili difraktovani snop. Češće se koristi konfiguracija u kojoj se monohromator nalazi u difraktovanom snopu, između uzorka i detektora, jer se time eliminiše i neželjeno zračenje sa uzorka (fluorescencija i nekoherentno zračenje) čime se snižava bazna linija snimka.



Slika 1.4: Principijelna izvedba monohromatora sa linijskim fokusom

1.2 Difrakcija na kristalnom prahu

Difrakcija je fenomen skretanja talasnog fronta pri nailasku na prepreku ili otvor reda veličine talasne dužine upadnog talasa. Ova pojava se dešava bez obzira na tip talasa, bili oni mehanički kao npr. zvuk i talasi na površini vode, ili elektromagnetni poput svetlosti.

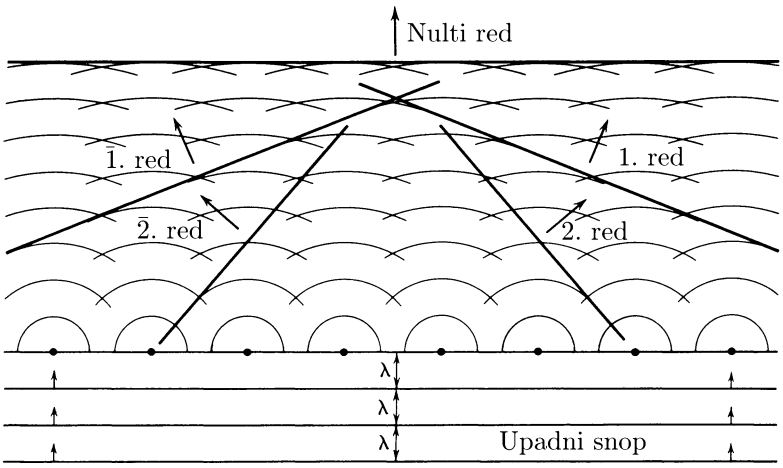
Još po otkriću X-zraka javile su se pretpostavke o njihovoj talasnoj prirodi, ali nisu postojali defintivni fizički dokazi za to. Rasprave o prirodi X-zraka vodile su se sve do eksperimenta koji je pokazao da oni pripadaju elektromagnetnom spektru. Pojavu difrakcije X-zraka predvideo je Laue[1], i na njegov predlog su Fridrih[2] i Kniping[3] 1912. izveli prvi ogled difrakcije X-zraka na monokristalu ZnS.

Difrakcija X-zraka na kristalima nastaje usled rasejanja upadnih zraka na elektronima atoma koji čine kristal bez promene talasne dužine. Do pojave difraktovanih zraka će doći samo kada su zadovoljeni određeni geometrijski uslovi koji se mogu izraziti u vidu Laueovih jednačina ili Bragove[4] formule. Rezultujuća difrakciona slika predstavlja karakteristiku svake supstance i može se iskoristiti kako za njenu identifikaciju, tako i za određivanje njene strukture.

S obzirom da su X-zraci elektromagnetni talasi, elektron na njihovoj putanji će pod uticajem promenljivog električnog polja početi da osciluje, time i sam postajući izvorom elektromagnetnog zračenja iste frekvencije. Iz ove interakcije se javlja sferni talasni front X-zračenja, sa elektronom kao izvorom, pa se kaže da elektron rasejava upadni snop. Kako je atom sačinjen od jezgra okruženog elektronskim oblakom, talasi rasejani na više elektrona se kombinuju, tako da se atom ponaša kao tačkasti izvor. Intenzitet rasejanja zavisi od broja elekrona u atomu, ali se menja i sa pravcem zbog prostorne raspodele elektrona unutar atoma.

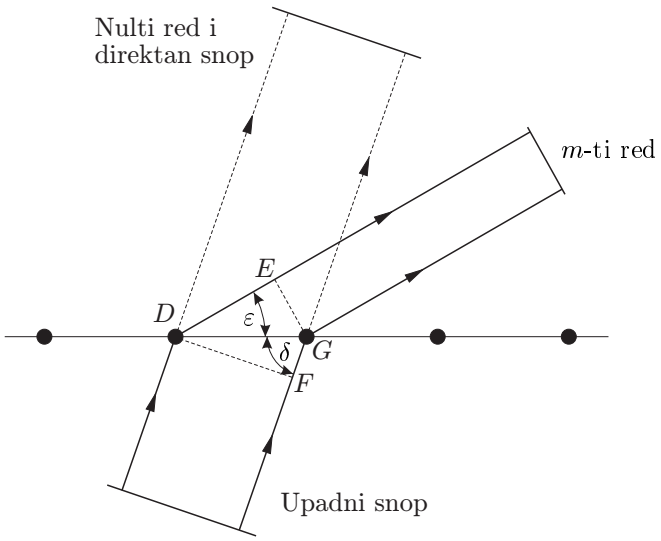
Kao i kod mehaničkih i svetlosnih talasa, između X-zraka rasejanih na različitim atomima može doći do pojave konstruktivne i destruktivne interferencije. Pretpostavimo

[1] Max von Laue
[2] Walther Friedrich
[3] Paul Knipping
[4] William Henry Bragg



Slika 1.5: Konstruktivna interferencija na nizu atoma

da snop X-zraka nailazi na niz jednako udaljenih atoma, kao na slici 1.5. Usled interakcije sa upadnim snopom, svaki atom postaje izvor sfernih talasa. Smer zajedničke tangente susednih sfernih talasa čini novi talasni front, a konstruktivna interferencija se događa pri putnim razlikama jednakim celobrojnom umnošku talasnih dužina. Ovakva konstruktivna kombinacija i pojačanje talasnog fronta u određenim pravcima predstavlja difrakciju. Pri tome, nulti red difrakcije odgovara ravnom talasu paralelnom sa upadnim, odnosno propuštenom talasu. Prvi red se javlja u smeru u kome se kombinuju talasi sa faznom razlikom od jedne talasne dužine, a analogno se dobijaju i drugi i viši redovi difrakcije. Odgovarajući negativni redovi se javljaju simetrično u odnosu na smer nultog reda. Ukoliko putna razlika nije jednaka celom broju talasnih dužina, interferencija je destruktivna i u tom smeru se ne javlja talasni front.



Slika 1.6: Uslovi za difrakciju na nizu atoma

Laueov model difrakcije na monokristalu

Niz atoma koji leže na pravoj, na jednakim međusobnim rastojanjima a_0, čine linearnu rešetku (slika 1.6). Posmatrajmo paralelan snop X-zraka koji nailazi na ovakav niz atoma pod uglom δ, i uočimo dva atoma iz rešetke, u tačkama A i B. Svi atomi deluju kao izvori sfernih talasa, pa će se difraktovani zraci nultog, prvog, drugog i viših redova javljati u određenim pravcima. Da bi do ovoga došlo, zraci rasejani u tačkama A i B moraju biti u fazi, tj. putna razlika ovih zraka mora biti jednaka celobrojnom umnošku talasnih dužina upadnog zračenja λ:

$$DE - FG = m\lambda, \qquad m \in \mathbb{Z}.$$

Iz jednostavnih trigonometrijskih relacija sledi da je

$$DE = a_0 \cos\varepsilon, \qquad FG = a_0 \cos\delta,$$

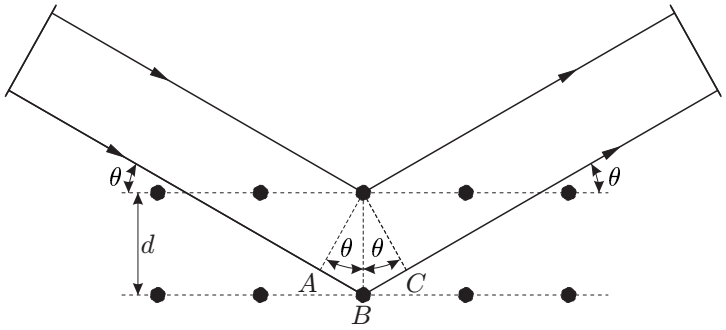
odakle je putna razlika

$$DE - FG = a_0 \cos\varepsilon - a_0 \cos\delta = a_0,$$

pa se Laueov uslov za difrakciju svodi na

$$m\lambda = (\cos\varepsilon - \cos\delta).$$

Ovaj uslov je ispunjen za sve uglove ε u konusu oko linije niza, pa se i difrakcioni maksimumi javljaju duž izvodnica ovih konusa. Jasno je da će, pri nailasku na film ili ekran pod određenim uglom, difrakciona slika biti u obliku konusnog preseka, tj. kruga ili elipse.



Slika 1.7: Uslovi za Bragovu difrakciju

Bragov model difrakcije

Bragov model objašnjava difrakciju pretpostavljajući da se X-zraci, rasejavajući se na nizu paralelnih ravni u kristalu, efektivno reflektuju i pojačavaju u smerovima za koje je zadovoljen uslov konstruktivne interferencije.

Na slici 1.7 je prikazan jedan takav niz paralelnih ravni sa međuravanskim rastojanjem d i snop X-zraka koji se na njima rasejava. Putna razlika dva zraka koji se reflektuju u tačkama D i B je ABC, odnosno

$$ABC = AB + BC = d\sin\theta + d\sin\theta = 2d\sin\theta,$$

a uslov za konstuktivnu interferenciju daje

$$n\lambda = 2d\sin\theta, \tag{1.2.1}$$

što je poznata Bragova jednačina.

Tabela 1.2
Najčešće korišćene metode difrakcije.

Metoda	Uzorak	Zračenje	Detektor	Namena
Laue	M	P	Foto-ploča, miruje	Orijentacija kristala, određivanje grupe simetrije
Obrtni kristal	M	M	Cilindrična kamera	Ako je kristal pravilno orijentisan, može se odrediti jedan parametar elementarne ćelije
Debye-Scherrer	P	M	Nepokretna filmska komora	Određivanje međuravanskih rastojanja, analizom intenziteta se rešavaju se strukture više simetrije
Difraktometrijska	P	M	Gajger-Milerov ili scintilacioni brojač	Kao kod Debaj-Šererove metode, plus identifikacija komponenti smeše, kvantitativna analiza smeše

1.2.1 Eksperimentalne metode difrakcije na kristalima

Postoji nekoliko metoda difrakcije na kristalima koje se međusobno razlikuju po vrsti uzorka, tipu upadnog zračenja i detektora koji se koristi. Različitim eksperimentalnim metodama se dobijaju i različiti podaci o uzorku koji se posmatra, pa svaka od metoda ima i svoju namenu. Kratak pregled metoda koje su najčešće u upotrebi je dat u tabeli 1.2.

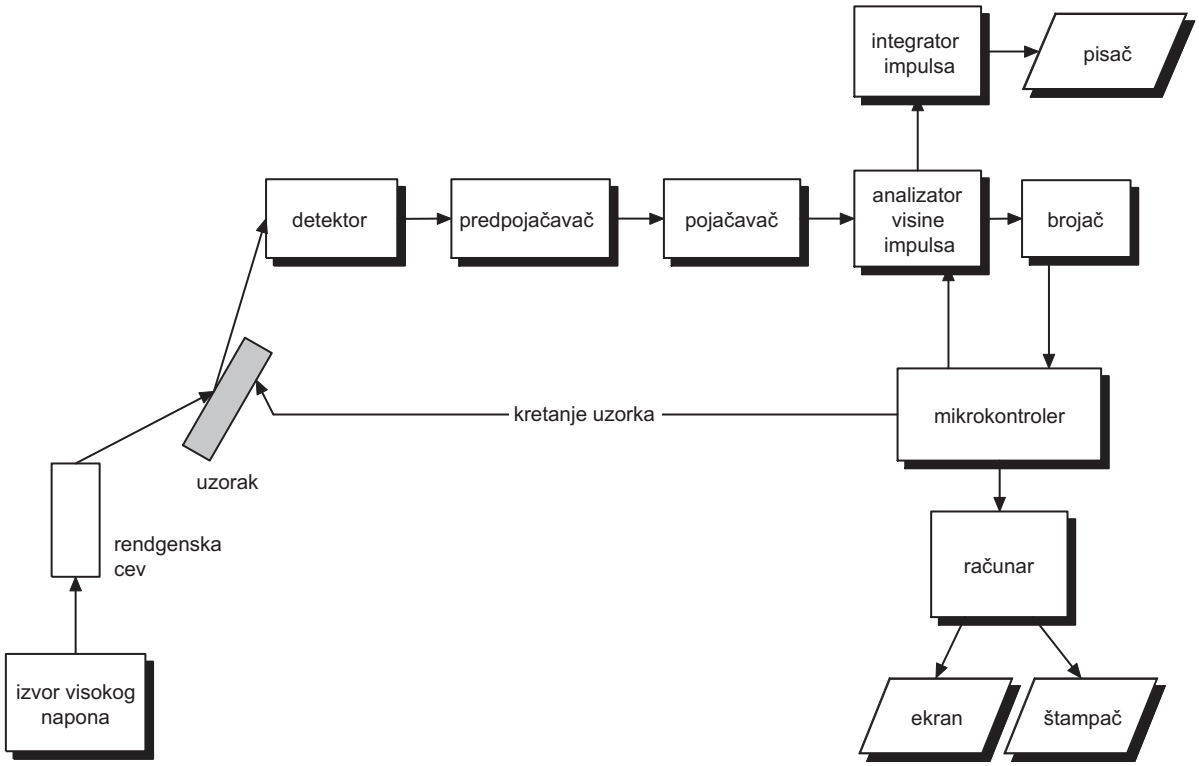
1.2.2 Difraktometar za prah

Za ispitivanje osobina različitih materijala najčešće se koristi difraktometrijska metoda na prahu, odnosno difraktometar za prah kao instrument. Zbog svojih prednosti, kao što su jednostavnija priprema uzorka, kraće vreme eksperimenta, tačniji rezultati i sl., ova metoda je gotovo potpuno zamenila ranije korišćene metode filma.

U najopštijem slučaju, jedan difraktometrijski sistem čine (slika 1.8):

- izvor visokog napona (generator),
- rendgenska cev,
- jednokružni goniometar sa uzorkom u centru,
- uređaj za registrovanje intenziteta.

Tipičan tok merenja ovim sistemom se sastoji od pripreme uzorka, njegovog izlaganja X-zracima i detektovanja zraka difraktovanih sa pogodno orijentisanih kristala u uzorku. Impulsi sa detektora se pojačavaju, a pomoću analizatora visine impulsa (diskriminatora)



Slika 1.8: Šematski prikaz difraktometrijskog sistema

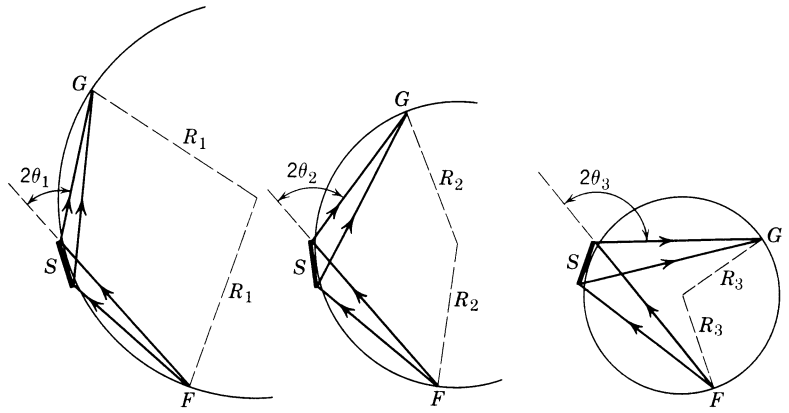
i filtera ili monohromatora odstranjuje se neželjeno belo zračenje, Kβ zračenje i nekoherentno i fluorescentno zračenje. Nakon toga se impulsi beleže na jedan od dva načina: analogno (pomoću integratora impulsa i pisača) ili digitalno (pomoću brojača).

Difraktometri za prah su danas uglavnom automatizovani, što znači da se svim ovim operacijama upravlja pomoću računara, odnosno kompjuterskih programa, a projektovanje i izrada jednog takvog programa obrađeni su u ovom radu.

Pomoću difraktometra za prah mere se intenziteti i uglovi 2θ, a vrednosti za međuravanska rastojanja se dobijaju iz Bragove formule (1.2.1). Zbog toga je potrebno uzeti u obzir faktore koji utiču na tačnost merenja ugla 2θ, a to su:

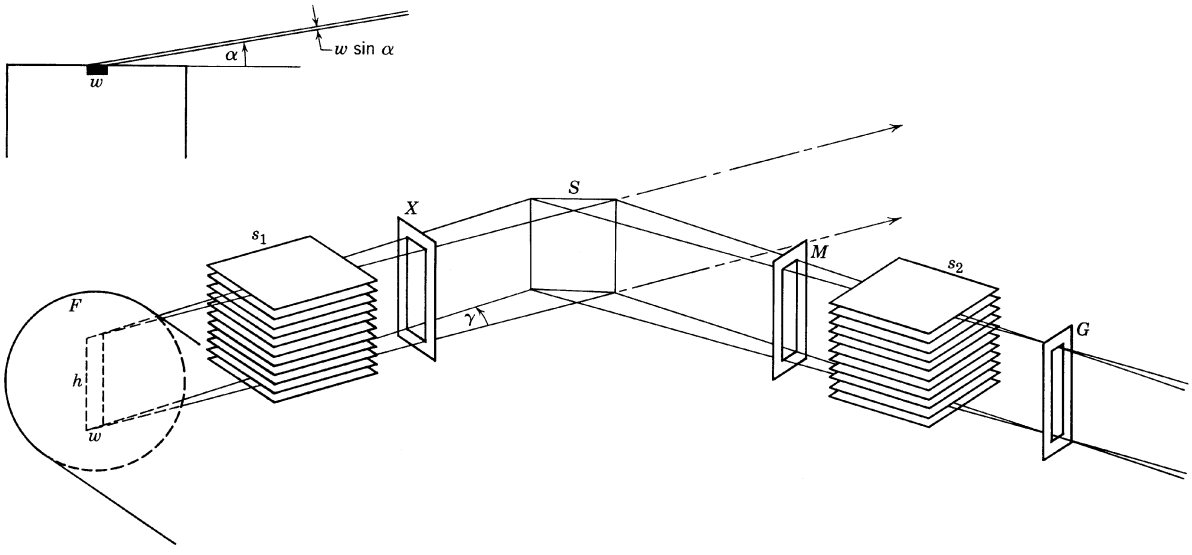
1. podešenost instrumenta,
2. beleženje impulsa,
3. priprema i postavljanje uzorka,
4. određivanje položaja pika,
5. određivanje bazne linije.

Većina komercijalnih difraktometara za prah koristi tzv. parafokusnu Brag-Brentano geometriju, čije su osnove date na slici 1.9. U ovakvoj geometriji detektor G rotira zajedno sa uzorkom S, ali dvostruko većom ugaonom brzinom, čime se pri okretanju uzorka za ugao θ detektor okrene za 2θ. Time se detektor i uzorak drže na konstantnom rastojanju uz održavanje fokusa snopa na detektoru. Izvor F i detektor G se nalaze na tzv. fokusnom krugu, koji predstavlja krivu na čijem je preseku sa snopom snop fokusiran. Primetno je da se sa povećavanjem ugla 2θ radijus fokusnog kruga smanjuje.



Slika 1.9: Parafokusirajuća Brag-Brentano geometrija

Principijelna šema konkretne izvedbe ovake konfiguracije data je na slici 1.10. Divergentni snop X-zraka dolazi sa linijskog izvora rendgenske cevi F, prolazi kroz Solerov[1] kolimator s_1 čija je funkcija da ograniči bočnu divergenciju snopa i time delimično kontroliše oblik difrakcionih linija. Zatim snop prolazi kroz divergentni prorez X, pada na uzorak S, difraktuje se, prolazi kroz još tzv. prijemni prorez M i drugi Solerov kolimator s_2. Uloga proreza M je da zajedno sa drugim Solerovim kolimatorom izdvoji samo difraktovane rendgenske zrake. Veličinu proreza X treba birati tako da snop X-zraka obasjava celu površinu uzorka. Ovakva geometrija uz pravilno podešavanje daje veoma veliku rezoluciju.



Slika 1.10: Optički podsistem zasnovan na Brag-Brentano geometriji

[1] W. Soller

1.3 Povezivanje računara i prenos podataka

1.3.1 Uvod

Osnovna jedinica informacije kojom se koriste računari je *bit*[1], koji predstavlja logičku vrednost nule ili jedinice. Fizički, bit se može manifestovati otvorenim ili zatvorenim tranzistorom ili različitim nivoima napona unutar kola. Tako, na primer, pozitivan napon može označavati logičku jedinicu, a negativan napon logičku nulu.

Kolekcija od osam bita čini *bajt*, i jasno je da se jednim bajtom može predstaviti 256 (2^8) različitih vrednosti ili znakova. Šema kojom se specificira način predstavljanja različitih znakova putem različitih kombinacija bita prestavlja *kod*. Najrašireniji standardizovani kod koji se danas koristi je tzv. ASCII[2].

Komunikacija između računara i perifernih uređaja vrši se preko *interfejsa*, koji predstavlja tačku kontakta između različitih okruženja. Često se kao sinonim koristi i termin *port*.

Dva načina koja se koriste za prenos podataka između računara i perifernih uređaja jesu *serijski* i *paralelni* i u osnovi se razlikuju po količini podataka koja se istovremeno prenosi. Kod serijskog porta se jedan znak šalje ili prima kao sekvenca od osam bitova, dok paralelni port koristi osam linija za prenos te se ceo znak šalje odjednom. Konkretne implementacije ova dva standarda ispoljavaju još značajnih razlika o kojima će biti reči i usled kojih je serijska komunikacija povoljnija za prenošenje podataka na većim razdaljinama.

Zanimljivo je da su prednosti i mane oba ova načina bile iskristalisane još na samom početku razvoja komunikacionih tehnologija. Rani pokušaji komunikacije na daljinu korišćenjem električne struje javili su se krajem osamnaestog veka. Godine 1809. nemački naučnik fon Zomering[3] razvio je elektrohemijski telegraf koji je signal prenosio putem 26 žica (za svako slovo nemačke abecede po jedna). Isti broj prekidača na mestu slanja poruke obezbeđivao je zatvaranje kola sa baterijom, dok se na drugom kraju nalazila posuda sa rastvorom kiseline u kojoj je, po zatvaranju kola, dolazilo do stvaranja mehurića na tačno određenom mestu za svaku žicu, tj. slovo. Na taj način, poruke su se mogle slati slovo po slovo.

Vetston[4] i Kuk[5] su 1837. konstruisali prvi telegraf koji je ušao u komercijalnu upotrebu. Njihov uređaj je imao pet žica preko kojih su se napajali mali elektromagneti koji su pokretali pokazivačke igle. Dvadeset znakova je bilo poređano u matricu 5×5, i različite kombinacije okretanja dve igle "ukrštale" su se na određenom znaku. Ovaj telegraf je predstavljao veliki napredak za svoje vreme i začeo je eru modernih komunikacija. Međutim, kao i fon Zomeringov uređaj, bio je nepraktičan pre svega zato što je zahtevao više žica, tj. pripadao je klasi paralelnih uređaja.

Prvi serijski binarni uređaj bio je Morzeov[6] telegraf iz 1837. koji je koristio odgovarajući kod. U Morzeovom kodu, znaci su predstavljeni tačkama i crtama koje odgovaraju binarnim jedinicama i nulama. Za slanje i prijem poruka je potrebna svega jedna signalna linija, dok se za zatvaranje koristila uzemljena linija. Njegova jednostavnost je

[1] *Binary digit*
[2] *American Standard Code for Information Interchange*
[3] Samuel Thomas von Soemmering
[4] Charles Wheatstone
[5] William Cooke
[6] Samuel Morse

omogućavala relativno pouzdan prenos na velikim razdaljinama, a prva javna linija je uspostavljena 1844. između Vašingtona i Baltimora.

1.3.2 Serijski port

Standardni serijski interfejs je nastao usled potrebe za povezivanjem velikog broja različitih računarskih sistema i komponenti. Sredinom šezdesetih godina dvadesetog veka razvoj kompjuterske tehnologije doveo je do uvođenja sve većeg broja računara u poslovnu upotrebu, a istovremeno se javila i potreba za pristup mejnfrejm sistemima sa daljine. Povezivanje terminala sa kompjuterom najčešće je vršeno putem standardnih telefonskih linija, korišćenjem modema. Različiti proizvođači su, naravno, koristili različite standarde za kablove, signale i komunikacione protokole, pa je međupovezivanje računara i modema različitih firmi bilo moguće samo uz različite prepravke i dodatne uređaje. Preporučeni standard u toj oblasti dali su EIA[1] i *Bell Laboratories*, 1969. godine.

Standard definiše električne, mehaničke i funkcionalne karakteristike. Pod električnim karakteristikama se podrazumevaju parametri poput nivoa napona i impedanse kablova, dok je mehaničkom specifikacijom preporučen raspored pinova; izgled samog konektora nije specificiran, pa postoji nekoliko različitih tipova koji se koriste. Funkcionalni opis definiše funkcije različitih električnih signala.

Kroz nekoliko revizija standard je uskoro postao poznat kao RS-232C[2], dok je u Evropi u upotrebu ušao pod okriljem CCITT[3] sa oznakama V.24 (funkcionalni opis) i V.28 (specifikacija električnih osobina).

RS-232C je definisan kao standard pod nazivom *"Interfejs između DTE[4] i DCE[5] uređaja koji koristi serijsku binarnu razmenu podataka"*. Prema standardu, DTE predstavlja terminal, tj. uređaj sa tastaturom i ekranom, dok je tipičan DCE uređaj modem. Ovo razgraničenje je potrebno zbog jasnog definisanja rasporeda pinova na različitim krajevima kabla i smera protoka podataka. U principu, danas se pod DTE uređajem podrazumeva kompjuter, dok su svi ostali DCE.

Parametri komunikacije

Standard RS-232C dopušta brzine prenosa od 110 do 19200 bita u sekundi, pri čemu postoje dva načina komunikacije: sinhroni i asinhroni. Kod sinhrone komunikacije, uređaji koji su povezani se inicijalno međusobno sinhronizuju i potom kontinualno šalju tzv. *clock* signale kako bi ostali sinhronizovani.

Kod asinhrone komunikacije uređaji se ne sinhronizuju, već se početak i kraj slanja svakog znaka obeležava *start* i *stop* bitovima. Asinhrona linija se identifikuje kao slobodna vrednošću logičke jedinice[6]; korišćenjem ove vrednosti kako bi se označilo da se nikakvi podaci ne šalju obezbeđuje se razlikovanje slobodne linije od prekida u vezi. *Start* bit ima vrednost logičke nule. Stoga, kada linija pređe iz vrednosti 1 u vrednost 0, uređaj koji prima podatke obaveštava se da će uslediti niz bitova koji predstavljaju podatke. Ovaj niz može sadržati 5, 6, 7 ili 8 tzv. *data* bitova, što se može birati pri konfigurisanju

[1] *Electronics Industry Association*

[2] *Recommended Standard number 232, revision C*

[3] *Comité Consultatif Internationale de Télégraphie et Téléphonie*

[4] *Data Terminal Equipment*

[5] *Data Communications Equipment*

[6] U terminologiji koja se koristi u oblasti komunikacija vrednost 1 se označava i kao *mark*, a 0 kao *space*, što potiče od Morzeovog telegrafa koji je pri ispisu na papirnu traku davao liniju ili razmak.

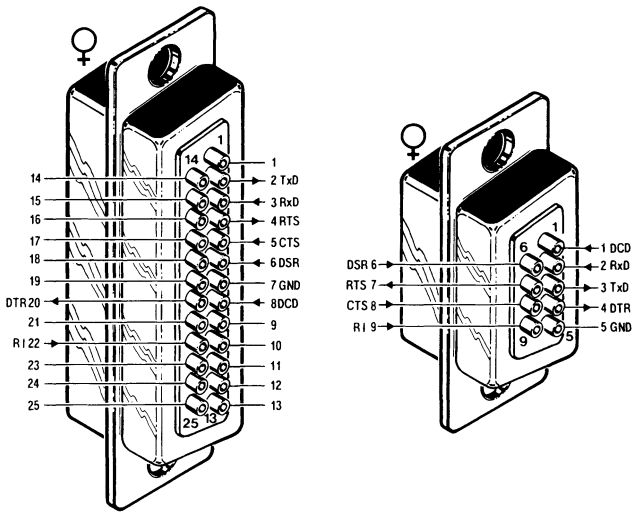
porta. Većina uređaja radi sa 7 ili 8 bitova za podatke. Po prenosu podataka, šalje se *stop* bit, koji ima vrednost logičke jedinice i koji se može ispravno detektovati čak i ako je prethodni *data* bit bio jedinica; ovo se postiže njegovim trajanjem koje je različito od trajanja *data* bitova. *Stop* bit može biti dužine 1, 1.5 ili dva perioda.

Zbog slanja ova dva dodatna bita asinhrona komunikacija je nešto sporija od sinhrone; međutim, zbog jednostavnije implementacije protokola, svi standardni računari i modemi danas koriste upravo ovaj način komunikacije.

Pored start i stop bitova, uz podatke se može slati i bit parnosti (*parity bit*), koji obezbeđuje rudimentarnu korekciju grešaka. Stanje ove vrednosti se može konfigurisati kao parno (*even*) odnosno neparno (*odd*), što znači da će bit parnosti biti setovan ako je u binarnoj reprezentaciji znaka koji se šalje paran, tj. neparan broj jedinica. Postoje i tzv. *mark* ili *space* modovi, ali se oni veoma retko koriste, jer za bit parnosti uvek postavljaju vrednost 1, odnosno 0. Ako je izabrana provera parnosti na paran broj jedinica, onda će, ukoliko je broj jedinica u znaku zaista paran, bit parnosti imati vrednost 1. Ukoliko prijemni uređaj ne primi paran broj jedinica usled greške u prenosu, proverom bita parnosti moći će da detektuje grešku.

Svi parametri komunikacije se mogu koncizno označiti šemom (*brzina, broj data bitova, parnost, broj stop bitova*), pa npr. 9600,8N1 označava prenos brzinom od 9600 bps, sa osam bitova za podatke, jednim stop bitom i bez provere parnosti.

Po standardu je predviđeno da DTE uređaji koriste 25-pinski "muški", a DCE uređaji 25-pinski "ženski" konektor, a u upotrebi su i standardizovani 9-pinski konektori. Stoga se DTE i DCE uređaji međusobno povezuju "pin na pin", dok je za povezivanje dva uređaja istog tipa (na primer, dva računara) potrebno ukrstiti linije za slanje i primanje na različitim krajevima kabla, čime se dobija tzv. *null-modem* kabl.



Slika 1.11: Raspored pinova za 25-pinske i 9-pinske konektore

Raspored pinova je dat na slici 1.11, a njihove funkcije u tabeli 1.3.

Iz tabele se vidi da 9-pinski konektor sadrži podskup u kome su najvažniji tj. najčešće korišćeni pinovi. U minimalnoj konfiguraciji, za slanje i primanje podataka su potrebne samo GND, RxD i TxD linije. Prva predstavlja zajedničku masu, u odnosu na koju se mere svi naponi u kolu. Na TxD liniji DTE uređaj vrši slanje, a DCE primanje podataka, dok RxD linija služi za slanje podataka sa DCE i prijem na DTE uređaju.

Naponski nivoi definisani u standardu su sledeći: logička nula predstavlja se naponom od 3 V do 25 V, logičku jedinicu reprezentuje napon od −3 V do −25 V, dok je oblast između −3 V i 3 V nedefinisana.

Maksimalna brzina prenosa je 20 kbps sa kablom dužine do 16 m. Korišćenjem kvalitetnijih kablova mogu se postići i veće brzine na većim razdaljinama.

Treba pomenuti i da se na izlazu serijskog porta može dobiti struja reda veličine 10 mA, pa se manji potrošači, na primer kompjuterski miševi, mogu pokretati bez dodatnih izvora napajanja.

Tabela 1.3

DB-25 pin	DB-9 pin	Uobičajena skraćenica	Smer DTE–DCE	Formalni naziv
1		FG		Frame Ground
2	3	TD	→	Transmitted Data, TxD
3	2	RD	←	Received Data, RxD
4	7	RTS	→	Request To Send
5	8	CTS	←	Clear To Send
6	6	DSR	←	Data Set Ready
7	5	SG		Signal Ground, GND
8	1	DCD	←	Data Carrier Detect
9				+P
10				−P
11				unassigned
12		SDCD	←	Secondary Data Carrier Detect
13		SCTS	←	Secondary Clear To Send
14		STD	→	Secondary Transmitted Data
15		TC	←	Transmission Signal Element Timing
16		SRD	←	Secondary Received Data
17		RC	→	Receiver Signal Element Timing
18				unassigned
19		SRTS	→	Secondary Request To Send
20	4	DTR	→	Data Terminal Ready
21		SQ	←	Signal Quality Detector
22	9	RI	←	Ring Indicator
23			→	Data Signal Rate Selector
24			←	Transmitter Signal Element Timing
25				unassigned

1.3.3 Paralelni port

Primarna namena paralelnog ili *Centronics* porta je povezivanje računara i štampača. Budući da ima osam linija za podatke, oni se šalju bajt po bajt, čime je paralelni port značajno brži od serijskog (u standardnom modu tipična brzina je 50 kBps, a može ići i do 150 kBps). Pored toga, s obzirom da omogućava istovremeni prijem 9 bita ili slanje 12 bita korišćenjem dodatnih kontrolnih linija kao signalnih, ovaj port se najčešće koristi i za povezivanje hobi projekata sa kompjuterima, jer se na taj način zahteva minimum eksternih kontrolnih kola. Konektor koji se koristi je 25-pinski "ženski" na strani računara.

Noviji paralelni portovi su standardizovani prema IEEE 1284 standardu, koji definiše sledeće modove rada:

- Standardni mod (kompatibilan sa *Centronics* standardom),
- *Nibble* mod (prijem podataka),
- *Byte* mod (prijem podataka),
- EPP (*Enhanced Parallel Port*) mod,
- ECP (*Extended Capabilities Port*) mod.

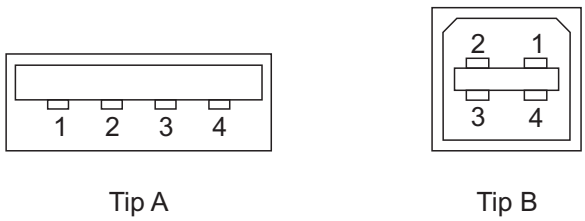
Upotreba paralelnog porta je ograničena činjenicom da usled postojanja velikog broja signalnih linija, na većim rastojanjima dolazi do interferencije i gubitka u kvalitetu signala, pa se ovaj način komunikacije može koristiti samo na kraćim rastojanjima.

1.3.4 USB port

USB[1] port je nastao 1994. godine kao objedinjeni standard za povezivanje različitih perifernih uređaja i računara, uz veću fleksibilnost i brzinu prenosa.

Trenutna revizija standarda, USB 2.0, specificira maksimalnu dužinu kabla od 5 m i definiše sledeće brzine prenosa: *low speed* – 1.5 Mbps, *full speed* – 12 Mbps i *high speed* – 480 Mbps.

Konektori su standardizovani kao tip A koji se priključuje na računar i tip B koji se priključuje na periferni uređaj (slika 1.12 i tabela 1.4). Za prenos podataka se koriste svega dve linije, kao i kod standardnog serijskog porta, a fleksibilnost se postiže paketnim prenosom podataka i različitim oblicima signala. Zbog velikog broja podržanih uređaja i različitih namena, upotreba USB porta je donekle komplikovana pa se za jednostavnije projekte još uvek preferira RS-232C.



Slika 1.12: Raspored pinova na USB konektorima

Tabela 1.4
Funkcije pinova po USB standardu

Pin	Boja	Funkcija
1	Crvena	V_{BUS} (5 V)
2	Bela	D−
3	Zelena	D+
4	Crna	GND

[1] *Universal Serial Bus*

Poglavlje 2

Opis uređaja

2.1 Difraktometrijski sistem

Osnovni elementi ovog sistema su:

- gonimetar MZ IV,
- mikrokontroler,
- izvor napajanja,
- detektorski sistem RAE 1,
- upravljački računar.

Pored ovih komponenti moguća je i nadogradnja sistema, na primer komorom za visokotemperaturska merenja, komorom za niske temperature, dodatkom za snimanje pod visokim pritiscima itd.

2.1.1 Goniometar (Seifert MZ IV)

Seifert MZ IV je dvokružni goniometar namenjen snimanju praškastih uzoraka u reflektujućoj geometriji Brag-Brentana. Sadrži dva step motora pomoću kojih je omogućen nezavisan pogon kako 2θ kruga na kome se nalazi detektor, tako i θ kruga na koji se postavljaju uzorci.

Pomoću odgovarajućih dodataka ovaj goniometar može da se montira horizontalno i da služi za ispitivanje monokristalnih uzoraka.

2.1.2 Mikrokontroler (Seifert μ-CONTROLLER)

Mikrokontroler se može koristiti u ručnom ili automatskom modu. U automatskom režimu je, sa upravljačkim računarom i izvorom napajanja, moguće upravljanje kompletnim difraktometrijskim sistemom.

Postoje dva načina merenja: sa zadatom dužinom trajanja, pri čemu se koristi tajmer koji se može podesiti za vreme merenja od 0.1 s do 999.9 minuta i sa zadatim odbrojem u intervalu od 1 do $10^9 - 1$. Rezultati merenja se mogu slati kako na štampač ili pisač, tako i na upravljački računar.

Mikrokontroler je modularnog dizajna i nalazi se u standardnom kućištu od 19 inča, a sadrži osam dvostrukih i osam jednostrukih "Evropa" kartica.

Komunikacija sa upravljačkim računarom se ostvaruje preko jednog od dva serijska porta po RS-232C standardu. Brzina prenosa podataka može se podešavati u intervalu od 110 do 9600 bita u sekundi. Ostali parametri veze su 8 bitova za podatke, jedan stop bit, bez provere parnosti, tj. 8N1.

Komunikacija kontroler-računar se vrši u jednom od dva moda:

- terminal mod, koji je korišćen u ovom projektu,
- kompjuterski mod, koji se suštinski razlikuje samo u načinu zadavanja komandi i formatu izlaznih podataka.

U mikrokontroler je ugrađen BASIC interpreter koji radi sa programima veličine do 2 kB, a on je i korišćen u izradi ovog programskog paketa.

2.1.3 Detektorski sistem (Seifert RAE I)

Ovaj sistem se sastoji od scintilacionog detektora sa fotomultiplikatorom i brojača; pomoću njega je moguće snimati amplitudni spektar rendgenskog zračenja. Položaj maksimuma intenziteta (pik) je proporcionalan usrednjenoj energiji rendgenskog zračenja. Za određivanje pika koristi se jednokanalni diskriminator čiji je zadatak da prenese samo one impulse čija amplituda leži između dve podešene granične vrednosti. Oblast između pomenutih vrednosti se uobičajeno naziva "prozor" ili "kanal". Ovaj sistem omogućava registrovanja energije zračenja u dva kalibrisana energetska područja i to od 0.1 keV do 10 keV i od 1 keV do 100 keV. Za kalibraciju uređaja je dovoljno kalibrisati samo jedno od ta dva područja pomoću zračenja poznate energije.

Moguća su tri načina merenja:

1. Integralno merenje. Pod ovim se podrazumeva brojanje samo onih impulsa čija je energija veća od neke zadate granične energije E_1.
2. Diferencijalno merenje (I). Broje se samo impulsi između dve zadate granične vrednosti E_1 (donje) i E_2 (gornje).
3. Diferencijalno merenje (II). Registruju se samo impulsi unutar prozora $\pm\Delta E$, koji je simetričan u odnosu na energiju E, koja predstavlja srednju tačku prozora.

Podešavanje širine prozora

Po pravilu se prozor podešava prema maksimumu samog spektra. Međutim, sledeći aspekti se moraju uzeti u obzir prilikom određivanja širine prozora:

- uticaj fona,
- uticaj zračenja rasejanog van uzorka,
- odbroj,
- uticaj nestabilnosti pojačavača.

Uticaj fona i zračenja rasejanog van uzorka, npr. na česticama u vazduhu ili na samim molekulima vazduha se redukuje proporcionalno sa širinom prozora. Međutim, treba voditi računa da merna nesigurnost raste sa smanjenjem širine prozora, pošto se sa smanjenjem odbroja povećava statistička nesigurnost koja se ponaša kao $N^{-1/2}$.

Pri odbrojima unutar ili ispod nivoa šuma (fon i šum uređaja), mera za optimalnu podešenost jeste kvalitativni faktor E^2/B, gde je E rezultat brojanja, a B je intenzitet šuma. Što je ovaj faktor veći, to je kraće vreme koje je potrebno za merenje pri zahtevanoj tačnosti. Optimalno podešavanje prozora se određuje preko niza probnih merenja sa različitim širinama prozora, pri čemu se za svaku određuje kvalitativni faktor.

Pri odbrojima koji su značajno viši nego sam fon, tačnost merenja pri specificiranoj dužini merenja zavisi samo od odbroja. U praksi se često izabira kompromis između maksimalnog faktora kvaliteta i maksimalnog odbroja, pri čemu se prozor tako podešava da njegova donja granica leži na istom odstojanju između talasne dužine koja odgovara maksimalnom pomaku pri Komptonovom[1] rasejanju i talasne dužine upadnog zračenja, čime se eliminiše doprinos Kompotonvog kontinuuma. U tom slučaju nestabilnosti pojačavača imaju neznatan uticaj izmerene odbroje.

Prilikom rasejanja sa nižim energijama čiji spektar ne sadrži Komptonov kontinuum, širina prozora se tako podešava da maksimum leži unutar prozora, radi postizanja visoke stabilnosti merenja.

Karakteristika scintilacionog detektora je takva da pri naponima od 600 V do 1000 V, odbroj u jedinici vremena ne zavisi od primenjenog napona, i u toj oblasti napona se i vrše merenja.

2.2 Upravljački računar

Računar koji se koristi za upravljanje i prikupljanje podataka može biti bilo koji PC računar sa operativnim sistemom Windows 95/98 ili Windows NT/2000/XP, serijskim portom i minimalnom rezolucijom ekrana 800×600. Program je uspešno testiran na sladećim platformama:

- Intel Pentium 150 MHz, 40 MB RAM, OS Windows 95 v4.0.971,
- Intel Celeron 533 MHz, 128 MB RAM, OS Windows 98,
- AMD Athlon XP 1600+, 512 MB RAM, OS Windows XP SP1.

Snimljeni podaci se za potrebe dalje analize mogu prenositi putem mreže ili flopi diskova, s obzirom na malu veličinu izlaznih datoteka.

[1] Arthur H. Compton

Poglavlje 3

Opis i implementacija programa Seifert MZ IV CDA

3.1 Opis programa

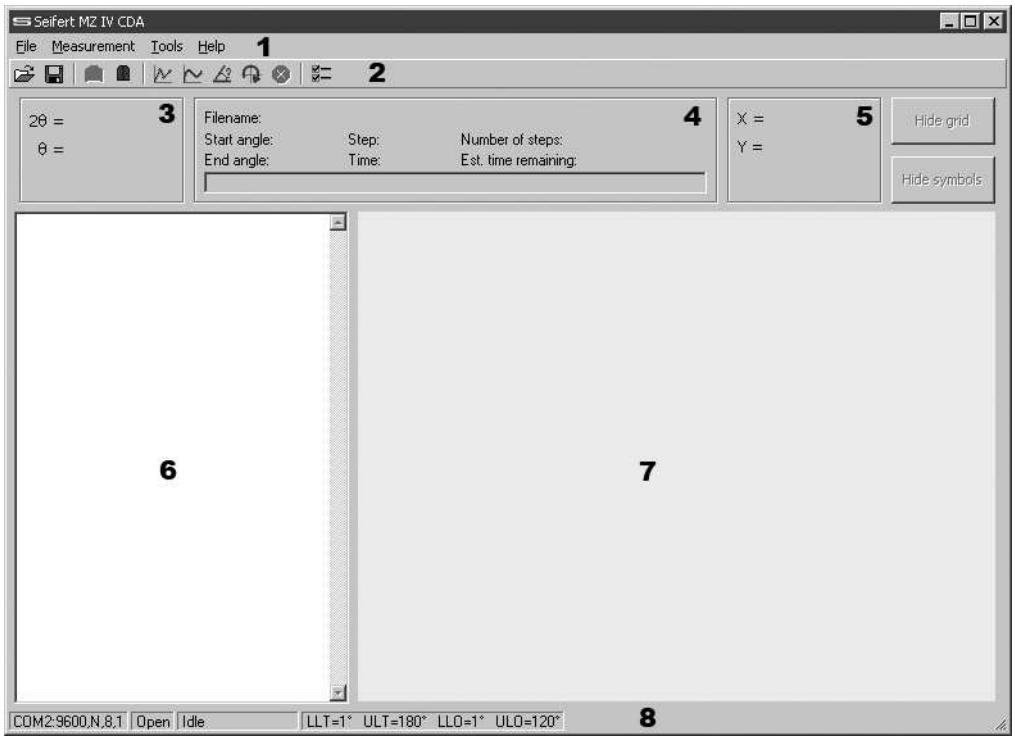
Program Seifert MZ IV CDA (*Control and Data Acquisition*) je, kao što se vidi iz naziva, namenjen prikupljanju podataka i upravljanju difraktometrijskim sistemom Seifert MZ IV.

Program radi pod bilo kojom 32-bitnom verzijom Windows operativnog sistema i ekstenzivno koristi grafičko okruženje kako bi bio što jednostavniji za korišćenje. Sa mikrokontrolerom komunicira putem serijskog porta, pružajući sledeće funkcije

- inicijalizacija difraktometrijskog sistema, pri čemu "pamti" sve bitne varijable iz prethodne sesije (parametri serijskog porta, granični i referentni uglovi, itd.),
- sinhrono podešavanje uglova 2θ i θ, u odnosu 2:1,
- očitavanje uglova 2θ i θ,
- prikupljanje podataka pri step merenju,
- prikupljanje podataka pri kontinualnom merenju sa uzorkovanjem na svakih $0.1°$,
- grafički prikaz merenja u toku, uz jednostavno očitavanje vrednosti tačaka sa grafika,
- zapisivanje podataka u fajl po izboru korisnika ili u fajl sa automatski generisanim imenom na osnovu datuma i vremena sesije,
- zapis podataka u standardne formate najrasprostranjenijih programa za analizu difraktometrijskih rezultata ("slobodni" dvokolonski format i 10-kolonski format sa fiksnom mantisom),
- učitavanje podataka iz fajlova u oba formata i njihov prikaz

Prozor koji se dobija pri učitavanju programa prikazan je na slici 3.1, i podeljen je na osam funkcionalnih celina:

1. Linija menija,
2. *Toolbar*, najčešće korišćene opcije,



Slika 3.1: Funkcionalne celine radnog prozora programa

3. Panel sa uglovima θ, 2θ i odbrojem (u toku merenja),

4. Panel sa podacima o merenju u toku,

5. Panel sa podacima o koordinatama merenih tačaka,

6. Terminal prozor,

7. Grafički prikaz merenih vrednosti,

8. Statusna linija.

Budući da je program jednostavan za korišćenje (ukoliko je korisnik upoznat sa principima difraktometrije praha) ovde se nećemo zadržavati na detaljnijem opisu komandi i procedura pri upotrebi.

3.2 Implementacija programa

Iako uz donekle haotičan razvojni put, program je projektovan sa sledećim osnovnim ciljevima:

- pouzdanost u radu,
- tačnost merenja,
- jednostavnost korišćenja.

Programski jezik u kome je projekat napisan je Microsoft Visual Basic 6.0. Ovaj izbor je učinjen pre svega zbog jednostavnosti same implementacije, pri čemu je programski jezik i izvršni kod koji generiše dovoljno fleksibilan za program ovakvog (relativno niskog) stepena zahtevnosti.

Osim toga, deo programa koji inicira i izvodi samo merenje (u step i u kontinualnom modu) je urađen kao program samog mikrokontrolera koji mu se šalje u terminal modu i izvršava bez daljeg nadzora glavnog programa. Naravno, određeni stepen kontrole je neophodan, pa se merenje može prekinuti, kontroler resetovati i nastaviti sa radom.

Kako je Visual Basic uglavnom objektno orijentisan, nije moguće dati kompletnu algoritamsku šemu za ceo program, ali je kompletan izvorni kod dat u dodatku A.

Dodatak A

Izvorni kod programa

```
'<------------------------ Description

VB.Form dlgDriveTheta
  +>VB.TextBox Text1
  +>VB.CommandButton CancelButton
  +>VB.CommandButton OKButton
  +>VB.Label Label1

'<------------------------ End Of Description
```

```
Attribute VB_Name = "dlgDriveTheta"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Option Explicit

Dim ValidateOK As Boolean
```

```
Private Sub CancelButton_Click()
  frmGlavna.DriveThetaTo = -1
  Unload Me
End Sub
```

```
Private Sub Form_Load()
  ' Ensure that the textbox has primary focus
  Text1.TabIndex = 0
End Sub
```

```
Private Sub OKButton_Click()
  Text1_Validate (True)
 ┌If ValidateOK = True Then
 │  frmGlavna.DriveThetaTo = Val(Text1.Text)
 │  Unload Me
 ├Else
 │  Text1.SetFocus
 └End If
End Sub
```

```
Private Sub Text1_GotFocus()
  SelTxt
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 ┌If KeyAscii = 46 And InStr(Text1, ".") > 0 Then
 │  KeyAscii = 0
 │  Exit Sub
 │  ' Sad proveravam brojeve, tacku i backspace
 ├ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
 │  KeyAscii = 0
 ├ElseIf KeyAscii = 9 Then
 │  Exit Sub
 └End If
End Sub
```

```
Private Sub Text1_Validate(KeepFocus As Boolean)
  ' If the value is not valid, keep the focus and display a message box

 ┌If Val(Text1.Text) < frmGlavna.GLLT Or Val(Text1.Text) > frmGlavna.GULT Then
 │  ValidateOK = False
 │  KeepFocus = True
 │  MsgBox "Theta angle not within limits", , "Theta"
 ├Else: ValidateOK = True
 └End If
End Sub
```

```
'<------------------------ Description

VB.Form dlgInitAttempt
  +>VB.CommandButton btnSave
  +>VB.TextBox txtCLI
  +>VB.TextBox txtLLT
  +>VB.TextBox txtULT
  +>VB.TextBox txtLLO
  +>VB.TextBox txtULO
  +>VB.TextBox txtRAT
  +>VB.TextBox txtRAO
  +>VB.TextBox txtSAT
  +>VB.TextBox txtSAO
  +>VB.TextBox txtDTT
  +>VB.TextBox txtVLI
  +>VB.TextBox txtSPD
  +>VB.Frame Frame2
    +>VB.TextBox txtDTT1
    +>VB.TextBox txtSAO1
    +>VB.TextBox txtSAT1
    +>VB.Label Label3
    +>VB.Label Label2
    +>VB.Label Label1
  +>VB.CommandButton btnEdit
  +>VB.Frame Frame1
    +>VB.Label Label15
    +>VB.Label Label4
    +>VB.Label Label5
    +>VB.Label Label6
    +>VB.Label Label7
    +>VB.Label Label8
    +>VB.Label Label9
    +>VB.Label Label10
    +>VB.Label Label11
    +>VB.Label Label12
    +>VB.Label Label13
    +>VB.Label Label14
  +>VB.CommandButton btnInit
  +>VB.CommandButton btnCancel

'<------------------------ End Of Description
```

```
Attribute VB_Name = "dlgInitAttempt"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim ValidateOK As Boolean
```

```
Private Sub btnCancel_Click()
  Unload Me
End Sub
```

```
Private Sub btnEdit_Click()
  btnEdit.Enabled = False
  btnInit.Enabled = False
  btnSave.Enabled = True
  EnableControlsUp
  txtVLI.SetFocus
  btnSave.Default = True
End Sub
```

```
Private Sub btnInit_Click()
  txtSAT1_Validate (True)
 ┌If ValidateOK = True Then
 │  txtSAO1_Validate (True)
 │ ┌If ValidateOK = True Then
 │ │  txtDTT1_Validate (True)
 │ │ ┌If ValidateOK = True Then
 │ │ │  SaveControls
 │ │ │  frmGlavna.GSAT = Val(txtSAT1.Text)
 │ │ │  frmGlavna.GSAO = Val(txtSAO1.Text)
 │ │ │  frmGlavna.GDTT = Val(txtDTT1.Text)
 │ │ │  frmGlavna.SaveINIFile
 │ │ │  frmGlavna.DoInit = True
 │ │ │  Unload Me
 │ │ ├Else: txtDTT1.SetFocus
 │ │ └End If
 │ ├Else: txtSAO1.SetFocus
 │ └End If
 ├Else: txtSAT1.SetFocus
 └End If
End Sub
```

```
Private Sub btnSave_Click()

  txtVLI_Validate (True)
 If ValidateOK = True Then
   txtCLI_Validate (True)
  If ValidateOK = True Then
    txtLLT_Validate (True)
   If ValidateOK = True Then
     txtULT_Validate (True)
    If ValidateOK = True Then
      txtLLO_Validate (True)
     If ValidateOK = True Then
       txtULO_Validate (True)
      If ValidateOK = True Then
        txtRAT_Validate (True)
       If ValidateOK = True Then
         txtRAO_Validate (True)
        If ValidateOK = True Then
          txtSAT_Validate (True)
         If ValidateOK = True Then
           txtSAO_Validate (True)
          If ValidateOK = True Then
            txtDTT_Validate (True)
           If ValidateOK = True Then
             txtSPD_Validate (True)
            If ValidateOK = True Then
              btnEdit.Enabled = True
              btnInit.Enabled = True
              btnSave.Enabled = False
              SaveControls
              FillControls
              DisableControlsUp
              txtSAT1.SetFocus
              btnInit.Default = True
            Else: txtSPD.SetFocus
            End If
           Else: txtDTT.SetFocus
           End If
          Else: txtSAO.SetFocus
          End If
         Else: txtSAT.SetFocus
         End If
        Else: txtRAO.SetFocus
        End If
       Else: txtRAT.SetFocus
       End If
      Else: txtULO.SetFocus
      End If
     Else: txtLLO.SetFocus
     End If
    Else: txtULT.SetFocus
    End If
   Else: txtLLT.SetFocus
   End If
  Else: txtCLI.SetFocus
  End If
 Else: txtVLI.SetFocus
 End If

End Sub
```

```
Private Sub DisableControlsUp()
  txtVLI.Enabled = False
  txtCLI.Enabled = False
  txtLLT.Enabled = False
  txtULT.Enabled = False
  txtLLO.Enabled = False
  txtULO.Enabled = False
  txtRAT.Enabled = False
  txtRAO.Enabled = False
  txtSAT.Enabled = False
  txtSAO.Enabled = False
  txtDTT.Enabled = False
  txtSPD.Enabled = False
  txtSAT1.Enabled = True
  txtSAO1.Enabled = True
  txtDTT1.Enabled = True
End Sub
```

```
Private Sub EnableControlsUp()
  txtVLI.Enabled = True
  txtCLI.Enabled = True
  txtLLT.Enabled = True
  txtULT.Enabled = True
  txtLLO.Enabled = True
  txtULO.Enabled = True
  txtRAT.Enabled = True
  txtRAO.Enabled = True
  txtSAT.Enabled = True
  txtSAO.Enabled = True
  txtDTT.Enabled = True
  txtSPD.Enabled = True
  txtSAT1.Enabled = False
  txtSAO1.Enabled = False
  txtDTT1.Enabled = False
End Sub
```

```
Private Sub FillControls()
  txtVLI.Text = Format(frmGlavna.GVLI)
  txtCLI.Text = Format(frmGlavna.GCLI)
  txtLLT.Text = Format(frmGlavna.GLLT)
  txtULT.Text = Format(frmGlavna.GULT)
  txtLLO.Text = Format(frmGlavna.GLLO)
  txtULO.Text = Format(frmGlavna.GULO)
  txtRAT.Text = Format(frmGlavna.GRAT)
  txtRAO.Text = Format(frmGlavna.GRAO)
  txtSAT.Text = Format(frmGlavna.GSAT)
  txtSAO.Text = Format(frmGlavna.GSAO)
  txtDTT.Text = Format(frmGlavna.GDTT)
  txtSPD.Text = Format(frmGlavna.GSPD)
  txtSAT1.Text = Format(frmGlavna.GSAT)
  txtSAO1.Text = Format(frmGlavna.GSAO)
  txtDTT1.Text = Format(frmGlavna.GDTT)
End Sub
```

```
Private Sub Form_Load()
  FillControls
  DisableControlsUp
End Sub
```

```
Private Sub SaveControls()
  frmGlavna.GVLI = Val(txtVLI.Text)
  frmGlavna.GCLI = Val(txtCLI.Text)
  frmGlavna.GULT = Val(txtULT.Text)
  frmGlavna.GULO = Val(txtULO.Text)
  frmGlavna.GLLT = Val(txtLLT.Text)
  frmGlavna.GLLO = Val(txtLLO.Text)
  frmGlavna.GRAT = Val(txtRAT.Text)
  frmGlavna.GRAO = Val(txtRAO.Text)
  frmGlavna.GSAT = Val(txtSAT.Text)
  frmGlavna.GSAO = Val(txtSAO.Text)
  frmGlavna.GDTT = Val(txtDTT.Text)
  frmGlavna.GSPD = Val(txtSPD.Text)

  frmGlavna.SaveINIFile
End Sub
```

```
Private Sub txtCLI_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtCLI_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtCLI, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtCLI_Validate(KeepFocus As Boolean)
 If txtCLI.Text = "" Or Val(txtCLI.Text) > 90 Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid current limit (max. 90)", , "Current limit"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtDTT_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtDTT_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtDTT, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtDTT_Validate(KeepFocus As Boolean)
 If txtDTT.Text = "" Or Val(txtDTT.Text) < Val(txtLLT.Text) Or Val(txtDTT.Text) > Val(
 txtULT.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid 'drive theta to' angle (LLT--ULT)", , "DTT"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtDTT1_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtDTT1_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtDTT1, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtDTT1_Validate(KeepFocus As Boolean)
  If txtDTT1.Text = "" Or Val(txtDTT1.Text) < Val(txtLLT.Text) Or Val(txtDTT1.Text) >
  Val(txtULT.Text) Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid 'drive theta to' angle (LLT--ULT)", , "DTT"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtLLO_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtLLO_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
  If KeyAscii = 46 And InStr(txtLLO, ".") > 0 Then
    KeyAscii = 0
    Exit Sub
    ' Sad proveravam brojeve, tacku i backspace
  ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
    KeyAscii = 0
  ElseIf KeyAscii = 9 Then
    Exit Sub
  End If
End Sub
```

```
Private Sub txtLLO_Validate(KeepFocus As Boolean)
  If txtLLO.Text = "" Or (Val(txtLLO.Text)) < Val(txtLLT.Text) Or (Val(txtLLO.Text) >
  Val(txtULT.Text)) Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid omega lower limit (LLT--ULT)", , "LLO"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtLLT_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtLLT_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
  If KeyAscii = 46 And InStr(txtLLT, ".") > 0 Then
    KeyAscii = 0
    Exit Sub
    ' Sad proveravam brojeve, tacku i backspace
  ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
    KeyAscii = 0
  ElseIf KeyAscii = 9 Then
    Exit Sub
  End If
End Sub
```

```
Private Sub txtLLT_Validate(KeepFocus As Boolean)
  If txtLLT.Text = "" Or Val(txtLLT.Text) < 1 Or Val(txtLLT.Text) > 180 Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid theta lower limit (1--180)", , "LLT"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtRAO_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtRAO_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtRAO, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtRAO_Validate(KeepFocus As Boolean)
 If txtRAO.Text = "" Or Val(txtRAO.Text) < Val(txtLLO.Text) Or Val(txtRAO.Text) > Val(
 txtULO.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid reference angle omega (LLO--ULO)", , "RAO"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtRAT_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtRAT_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtRAT, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtRAT_Validate(KeepFocus As Boolean)
 If txtRAT.Text = "" Or Val(txtRAT.Text) < Val(txtLLT.Text) Or Val(txtRAT.Text) > Val(
 txtULT.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid reference angle theta (LLT-ULT)", , "RAT"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtSAO_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtSAO_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtSAO, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtSAO_Validate(KeepFocus As Boolean)
 If txtSAO.Text = "" Or Val(txtSAO.Text) < Val(txtLLO.Text) Or Val(txtSAO.Text) > Val(
 txtULO.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid omega estimate (LLO--ULO)", , "SAO"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtSAO1_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtSAO1_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtSAO1, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtSAO1_Validate(KeepFocus As Boolean)
 If txtSAO1.Text = "" Or Val(txtSAO1.Text) < Val(txtLLO.Text) Or Val(txtSAO1.Text) >
 Val(txtULO.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid omega estimate (LLO--ULO)", , "SAO"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtSAT_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtSAT_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtSAT, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtSAT_Validate(KeepFocus As Boolean)
 If txtSAT.Text = "" Or Val(txtSAT.Text) < Val(txtLLT.Text) Or Val(txtSAT.Text) > Val(
 txtULT.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid theta estimate (LLT--ULT)", , "SAT"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtSAT1_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtSAT1_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtSAT1, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtSAT1_Validate(KeepFocus As Boolean)
  If txtSAT1.Text = "" Or Val(txtSAT1.Text) < Val(txtLLT.Text) Or Val(txtSAT1.Text) >
  Val(txtULT.Text) Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid theta estimate (LLT--ULT)", , "SAT"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtSPD_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtSPD_KeyPress(KeyAscii As Integer)
  ' Proveravam brojeve, tacku i backspace
  If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 Then
    KeyAscii = 0
  ElseIf KeyAscii = 9 Then
    Exit Sub
  End If
End Sub
```

```
Private Sub txtSPD_Validate(KeepFocus As Boolean)
  If txtSPD.Text = "" Or Val(txtSPD.Text) < 1 Or Val(txtSPD.Text) > 7 Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid global speed (1--7)", , "DTT"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtULO_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtULO_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
  If KeyAscii = 46 And InStr(txtULO, ".") > 0 Then
    KeyAscii = 0
    Exit Sub
    ' Sad proveravam brojeve, tacku i backspace
  ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
    KeyAscii = 0
  ElseIf KeyAscii = 9 Then
    Exit Sub
  End If
End Sub
```

```
Private Sub txtULO_Validate(KeepFocus As Boolean)
  If txtULO.Text = "" Or Val(txtULO.Text) <= Val(txtLLO.Text) Or (Val(txtULO.Text) > Val
  (txtULT.Text)) Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid omega upper limit (LLO--ULT)", , "ULO"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtULT_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtULT_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
  If KeyAscii = 46 And InStr(txtULT, ".") > 0 Then
    KeyAscii = 0
    Exit Sub
    ' Sad proveravam brojeve, tacku i backspace
  ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
    KeyAscii = 0
  ElseIf KeyAscii = 9 Then
    Exit Sub
  End If
End Sub
```

```
Private Sub txtULT_Validate(KeepFocus As Boolean)
  If txtULT.Text = "" Or Val(txtULT.Text) > 180 Or (Val(txtULT.Text) <= Val(txtLLT.Text
  )) Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid theta upper limit (LLT--180)", , "ULT"
  Else: ValidateOK = True
  End If
End Sub
```

```
Private Sub txtVLI_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtVLI_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
  If KeyAscii = 46 And InStr(txtVLI, ".") > 0 Then
    KeyAscii = 0
    Exit Sub
    ' Sad proveravam brojeve, tacku i backspace
  ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
    KeyAscii = 0
  ElseIf KeyAscii = 9 Then
    Exit Sub
  End If
End Sub
```

```
Private Sub txtVLI_Validate(KeepFocus As Boolean)
  If txtVLI.Text = "" Or Val(txtVLI.Text) > 90 Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid voltage limit (max. 90)", , "Voltage limit"
  Else: ValidateOK = True
  End If
End Sub
```

```
'<------------------------ Description

VB.Form dlgMeasCont
  +>MSComDlg.CommonDialog CommonDialog1
  +>VB.Frame Frame2
    +>VB.CommandButton cmdSelectFile
    +>VB.ComboBox cboSpeed
    +>VB.TextBox txtEnd
    +>VB.TextBox txtStart
    +>VB.OptionButton optTimestamp
    +>VB.OptionButton optSpecify
    +>VB.Label lblFileName
    +>VB.Label Label6
    +>VB.Label Label3
    +>VB.Label Label4
    +>VB.Label Label5
  +>VB.CommandButton CancelButton
  +>VB.CommandButton btnStart

'<------------------------ End Of Description
```

```
Attribute VB_Name = "dlgMeasCont"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Private Declare Function PathCompactPath Lib "shlwapi" Alias "PathCompactPathA" (ByVal
hDC As Long, ByVal lpszPath As String, ByVal dx As Long) As Long

Dim ValidateOK As Boolean
Dim lhDC As Long, lCtlWidth As Long
Dim FileName As String
Dim i As Integer
```

```
Private Sub btnStart_Click()

  txtStart_Validate (True)
 ┌If ValidateOK = True Then
 │  txtEnd_Validate (True)
 │ ┌If ValidateOK = True Then
 │ │  frmGlavna.gcmsStart = CDbl(txtStart.Text)
 │ │  frmGlavna.gcmsEnd = CDbl(txtEnd.Text)
 │ │  frmGlavna.gcmsSpeed = cboSpeed.ListIndex + 1
 │ │  Unload Me
 │ │
 │ ├Else: txtEnd.SetFocus
 │ └End If
 ├Else: txtStart.SetFocus
 └End If

End Sub
```

```
Private Sub CancelButton_Click()
  frmGlavna.gcmsSpeed = -1
  Unload Me
End Sub
```

```
Private Sub cmdSelectFile_Click()
  CommonDialog1.FileName = ""
  CommonDialog1.Flags = &H2
  CommonDialog1.ShowSave
 ┌If CommonDialog1.FileName <> "" Then
 │  frmGlavna.MEASFileName = CommonDialog1.FileName
 │  FileName = CommonDialog1.FileName
 │
 │  lCtlWidth = lblFileName.Width / 15 - Me.DrawWidth - 10
 │  lhDC = Me.hDC
 │  PathCompactPath lhDC, FileName, lCtlWidth
 │  i = InStr(1, FileName, Chr(0))
 │  If i > 0 Then FileName = Left(FileName, i - 1)
 │  lblFileName = "(" + FileName + ")"
 └End If

End Sub
```

```
Private Sub Form_Load()

  cboSpeed.AddItem "0.1 °/min"
  cboSpeed.AddItem "0.2 °/min"
  cboSpeed.AddItem "0.4 °/min"
  cboSpeed.AddItem "0.8 °/min"
  cboSpeed.AddItem "1.6 °/min"
  cboSpeed.AddItem "3.2 °/min"
  cboSpeed.AddItem "200 °/min"
  cboSpeed.ListIndex = 0
  frmGlavna.gcmsSpeed = -1

  Me.ScaleMode = vbPixels
  lblFileName.Caption = ""
End Sub
```

```
Private Sub optSpecify_Click()
  cmdSelectFile.Enabled = True
  cmdSelectFile.SetFocus
End Sub
```

```
Private Sub optTimestamp_Click()
  cmdSelectFile.Enabled = False
  frmGlavna.MEASFileName = ""
  lblFileName = ""
End Sub
```

```
Private Sub txtEnd_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtEnd_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtEnd, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtEnd_Validate(KeepFocus As Boolean)
  ' If the value is not valid, keep the focus and display a message box

 If txtEnd.Text = "" Or (Val(txtEnd.Text) < frmGlavna.GLLT) Or (Val(txtEnd.Text) >
 frmGlavna.GULT) Or _
    Val(txtEnd.Text) <= Val(txtStart.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid end angle value", , "End angle"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtFilename_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtSpeed_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtStart_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtStart_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtStart, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtStart_Validate(KeepFocus As Boolean)
  ' If the value is not valid, keep the focus and display a message box

 If txtStart.Text = "" Or Val(txtStart.Text) < frmGlavna.GLLT Or Val(txtStart.Text) >
 frmGlavna.GULT Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid start angle value", , "Start angle"
 Else: ValidateOK = True
 End If
End Sub
```

```
'<------------------------ Description

VB.Form dlgMeasStep
  +>MSComDlg.CommonDialog CommonDialog1
  +>VB.Frame Frame2
    +>VB.CommandButton cmdSelectFile
    +>VB.TextBox txtStep
    +>VB.TextBox txtEnd
    +>VB.TextBox txtStart
    +>VB.OptionButton optTimestamp
    +>VB.OptionButton optSpecify
    +>VB.TextBox txtTime
    +>VB.Label lblFileName
    +>VB.Label Label6
    +>VB.Label Label3
    +>VB.Label Label4
    +>VB.Label Label5
    +>VB.Label Label7
  +>VB.CommandButton CancelButton
  +>VB.CommandButton btnStart

'<------------------------ End Of Description
```

```
Attribute VB_Name = "dlgMeasStep"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Private Declare Function PathCompactPath Lib "shlwapi" Alias "PathCompactPathA" (ByVal
hDC As Long, ByVal lpszPath As String, ByVal dx As Long) As Long

Dim ValidateOK As Boolean
Dim lhDC As Long, lCtlWidth As Long
Dim FileName As String
Dim i As Integer
```

```
Private Sub btnStart_Click()

  txtStart_Validate (True)
 If ValidateOK = True Then
   txtEnd_Validate (True)
  If ValidateOK = True Then
    txtStep_Validate (True)
   If ValidateOK = True Then
     txtTime_Validate (True)
    If ValidateOK = True Then
      frmGlavna.gcmsStart = CDbl(txtStart.Text)
      frmGlavna.gcmsEnd = CDbl(txtEnd.Text)
      frmGlavna.gcmsStep = CDbl(txtStep.Text)
      frmGlavna.gcmsTime = CDbl(txtTime.Text)
      Unload Me
    Else: txtTime.SetFocus
    End If
   Else: txtStep.SetFocus
   End If
  Else: txtEnd.SetFocus
  End If
 Else: txtStart.SetFocus
 End If

End Sub
```

```
Private Sub CancelButton_Click()
  frmGlavna.gcmsTime = -1
  Unload Me
End Sub
```

```
Private Sub cmdSelectFile_Click()
  CommonDialog1.FileName = ""
  CommonDialog1.Flags = &H2
  CommonDialog1.ShowSave
 If CommonDialog1.FileName <> "" Then
   frmGlavna.MEASFileName = CommonDialog1.FileName
   FileName = CommonDialog1.FileName

   lCtlWidth = lblFileName.Width / 15 - Me.DrawWidth - 10
   lhDC = Me.hDC
   PathCompactPath lhDC, FileName, lCtlWidth
   i = InStr(1, FileName, Chr(0))
   If i > 0 Then FileName = Left(FileName, i - 1)
   lblFileName = "(" + FileName + ")"
 End If

End Sub
```

```
Private Sub Form_Load()
  'frmGlavna.gcmsTime = -1
  Me.ScaleMode = vbPixels
  lblFileName.Caption = ""
End Sub
```

```
Private Sub optSpecify_Click()
  cmdSelectFile.Enabled = True
  cmdSelectFile.SetFocus
End Sub
```

```
Private Sub optTimestamp_Click()
  cmdSelectFile.Enabled = False
  frmGlavna.MEASFileName = ""
  lblFileName = ""
End Sub
```

```
Private Sub txtEnd_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtEnd_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtEnd, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtEnd_Validate(KeepFocus As Boolean)
  ' If the value is not valid, keep the focus and display a message box

 If txtEnd.Text = "" Or (Val(txtEnd.Text) < frmGlavna.GLLT) Or (Val(txtEnd.Text) >
 frmGlavna.GULT) Or _
    Val(txtEnd.Text) <= Val(txtStart.Text) Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid end angle value", , "End angle"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtFilename_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtStart_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtStart_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtStart, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtStart_Validate(KeepFocus As Boolean)
  ' If the value is not valid, keep the focus and display a message box

 If txtStart.Text = "" Or Val(txtStart.Text) < frmGlavna.GLLT Or Val(txtStart.Text) >
 frmGlavna.GULT Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid start angle value", , "Start angle"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtStep_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtStep_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtStep, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtStep_Validate(KeepFocus As Boolean)
  ' If the value is not valid, keep the focus and display a message box

 If txtStep.Text = "" Or Val(txtStep.Text) > (Val(txtEnd.Text) - Val(txtStart.Text))
 Then
   ValidateOK = False
   KeepFocus = True
   MsgBox "Invalid measurement step value", , "Measurement step"
 Else: ValidateOK = True
 End If
End Sub
```

```
Private Sub txtTime_GotFocus()
  SelTxt
End Sub
```

```
Private Sub txtTime_KeyPress(KeyAscii As Integer)
  ' Dozvoljena je samo jedna tacka
 If KeyAscii = 46 And InStr(txtTime, ".") > 0 Then
   KeyAscii = 0
   Exit Sub
   ' Sad proveravam brojeve, tacku i backspace
 ElseIf (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 46 And KeyAscii <> 8 Then
   KeyAscii = 0
 ElseIf KeyAscii = 9 Then
   Exit Sub
 End If
End Sub
```

```
Private Sub txtTime_Validate(KeepFocus As Boolean)
   ' If the value is not valid, keep the focus and display a message box

  If txtTime.Text = "" Then
    ValidateOK = False
    KeepFocus = True
    MsgBox "Invalid step time value", , "Step time"
  Else: ValidateOK = True
  End If
End Sub
```

```
'<------------------------ Description

VB.Form frmAbout
  +>VB.PictureBox picIcon
  +>VB.CommandButton cmdOK
  +>VB.Label lblDescription
  +>VB.Label lblTitle
  +>VB.Label lblVersion

'<------------------------ End Of Description
```

```
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim i As Integer
Dim s As String
```

```
Private Sub cmdOK_Click()
  Unload Me
End Sub
```

```
Private Sub Form_Load()
  ' Me.Caption = "About " & App.Title
  ' lblTitle.Caption = App.Title
  lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision

  Randomize

  i = Round(Rnd)
  If i = 0 Then
    s = "Vladimir Jokic && Srdjan Rakic"
  Else
    s = "Srdjan Rakic && Vladimir Jokic"
  End If

  lblDescription.Caption = "Copyright (C) 2003-2004 " + s
End Sub
```

```
'<------------------------ Description

VB.Form frmGlavna
  +>VB.Frame FrameXY
    +>VB.Label Label2
    +>VB.Label Label1
  +>VB.CommandButton cmdGrid
  +>VB.Timer Timer1
  +>VB.Frame FrameMeas
    +>MSComctlLib.ProgressBar ProgressBar1
    +>VB.Label lblEstTime
    +>VB.Label lblFileName
    +>VB.Label lblStart
    +>VB.Label lblEnd
    +>VB.Label lblStep
    +>VB.Label lblTime
    +>VB.Label lblNoOfSteps
  +>MSComctlLib.ImageList ImageList1
  +>MSCommLib.MSComm MSComm1
  +>MSComDlg.CommonDialog CommonDialog1
  +>VB.CommandButton cmdSymbols
  +>VB.Frame Frame3
    +>VB.Label lblCount
    +>VB.Label lblTheta
    +>VB.Label lblOmega
  +>VB.TextBox TerminalWindow
  +>VB.PictureBox Grafik
  +>MSComctlLib.StatusBar StatusBar1
  +>MSComctlLib.Toolbar Toolbar1
(0)>VB.Menu mnuFile
    +>VB.Menu mnuFileOpen
    +>VB.Menu mnuFileSave
    +>VB.Menu mnuFileExport
    +>VB.Menu Separator1
    +>VB.Menu mnuFileExit
  +>VB.Menu mnuMeas
    +>VB.Menu mnuMeasStep
    +>VB.Menu mnuMeasCont
    +>VB.Menu Separator2
    +>VB.Menu mnuMeasStop
  +>VB.Menu mnuTools
    +>VB.Menu mnuToolsOpenPort
    +>VB.Menu mnuToolsClosePort
    +>VB.Menu Separator3
    +>VB.Menu mnuToolsReadAngles
    +>VB.Menu mnuToolsDriveTheta
    +>VB.Menu mnuToolsReset
    +>VB.Menu Separator4
    +>VB.Menu mnuToolsOptions
  +>VB.Menu mnuHelp
    +>VB.Menu mnuHelpAbout

'<------------------------ End Of Description
```

```
Attribute VB_Name = "frmGlavna"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Private Declare Function PathCompactPath Lib "shlwapi" Alias "PathCompactPathA" (ByVal
hDC As Long, ByVal lpszPath As String, ByVal dx As Long) As Long

Private DATAFileName As String
Private INIFileName As String
Public MEASFileName As String

Dim MEASFile

Dim Ret As Integer

Const SnapSize = 5
Const PreferredTickDensity = 10
Const BorderWidth = 50
Const AxisExtent = 20
Const TickWidth = 8

Public ScaleFactor As Double

Public COMPort As Integer
Public PortSpeed As Integer
```

```
Public DataBits As Integer
Public StopBits As Integer
Public PortParity As String

Public gfkBackColor As Long
Public gfkLineColor As Long
Public gfkAxisColor As Long
Public gfkGridColor As Long
Public gfkSymbolOColor As Long
Public gfkSymbolFColor As Long
Public gfkDrawSymbols As Boolean    ' Da li da se iscrtavaju tacke na grafiku
Public gfkDrawGrid As Boolean       ' Da li da se iscrtava mreza

Public GVLI As Integer              ' Voltage Limit
Public GCLI As Integer              ' Current Limit
Public GULT As Integer              ' Upper Limit Theta
Public GULO As Integer              ' Upper Limit Omega
Public GLLT As Integer              ' Lower Limit Theta
Public GLLO As Integer              ' Lower Limit Omega
Public GRAT As Double               ' Reference Angle Theta
Public GRAO As Double               ' Reference Angle Omega
Public GSAT As Double               ' eStimated Angle Theta
Public GSAO As Double               ' eStimated Angle Omega
Public GDTT As Double               ' Drive Theta To // Double
Public GSPD As Byte                 ' global Speed (1,2,3,4,5,6,7)
Public GCAT As Double               ' Current Angle Theta //?? da li da koristi CAT ili
SAT
Public GCAO As Double               ' Current Angle Omega //?? da li da koristi CAO ili
SAO

Public OpenPortAtStartup As Boolean
Public DriveThetaTo As Double
Public gcmsStart As Double
Public gcmsEnd As Double
Public gcmsStep As Double
Public gcmsTime As Double
Public gcmsSpeed As Integer

Public gcmsEstTime As Double

Public gcmsCurrentAngle As Double
Public gcmsCount As Integer

Public DoInit As Boolean            ' Da li da se obavi incijalizacija

Private InitOver As Boolean         ' Oznacava da je goniometar odradio inicijalizaciju
Private AnglesRead As Boolean       ' Oznacava da je zavrseno citanje uglova
Private ThetaDriven As Boolean      ' Oznacava da je Theta doveden na zadatu lokaciju
Private OmegaDriven As Boolean      ' Oznacava da je Omega doveden na zadatu lokaciju
Private MotorsDriven As Boolean     ' Oznacava da su Theta i Omega sinhrono dovedeni na
zadati ugao
Private MeasMade As Boolean         ' Oznacava da je izvrseno merenje

Private StartedDrawing As Boolean
Private SafeToExit As Boolean       ' Goniometar je izvrsio komandu, bezbedan izlazak iz
 programa

Dim DataMatrix() As Double          ' Matrica podataka
Dim PlotMatrix() As Double          ' Matrica tacaka za grafik
Dim TicksXData() As Double          ' Matrica ticks-a za X osu // koristice se od 1 do
max, 0 se ignorise, mrzi me da radim ReDim
Dim TicksYData() As Double          ' Matrica ticks-a za Y osu
Dim TicksXPlot() As Double          ' Matrica ticks-a za X osu, pixeli
Dim TicksYPlot() As Double          ' Matrica ticks-a za Y osu, pixeli
Dim TicksXcount As Integer
Dim TicksYcount As Integer
Dim TicksXformat As String
Dim TicksYformat As String

Dim BigBuffer As String
Dim MeasurementCount As Long

Dim XAxeBeg As Double, XAxeEnd As Double
Dim XAxeZero As Double, YAxeBeg As Double
Dim YAxeEnd As Double, YAxeZero As Double

Dim OldSnapIndex As Integer

1Public Enum ContextType            ' Kontekst u kome se trenutno nalazi goniometar - za
```

```
komande
1   Idle = 0                        ' Inicijalizacija je zavrsena, idle
    PreInit = 1                     ' Pre inicijalizacije
    Init = 2                        ' Inicijalizacija
    ReadAngles = 3                  ' Iscitavanje uglova
    DriveTheta = 4                  ' Zadato pomeranje Theta
    DriveOmega = 5                  ' Zadato pomeranje Omega
    DriveMotors = 6                 ' Sinhrono pomeranje Theta i Omega
    Measurement = 7                 ' Zadato merenje
End Enum

Dim Context As ContextType
```

```
Private Sub AddMeasToGraph()

  MeasurementCount = MeasurementCount + 1
  ProgressBar1.Value = MeasurementCount
  If MeasurementCount > 1 Then
    ReDim Preserve DataMatrix(2, 1 To MeasurementCount)
  End If

  DataMatrix(0, UBound(DataMatrix, 2)) = gcmsCurrentAngle
  DataMatrix(1, UBound(DataMatrix, 2)) = gcmsCount

  If (StartedDrawing = False) Then
    If DataMatrix(1, 1) - DataMatrix(1, UBound(DataMatrix, 2)) <> 0 Then
      StartedDrawing = True
    End If
  End If
  If StartedDrawing = True Then
    DrawGrafik
    RefreshShowHideButtons
  End If

End Sub
```

```
Sub AssignCOMPort(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  COMPort = Val(TempString)

End Sub
```

```
Sub AssignDataBits(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  DataBits = Val(TempString)

End Sub
```

```
Sub AssignGCLI(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GCLI = Val(TempString)
End Sub
```

```
Sub AssignGDTT(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GDTT = Val(TempString)
End Sub
```

```
Sub AssignGFKAxisColor(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  gfkAxisColor = Val(TempString)
End Sub
```

```
Sub AssignGFKBackColor(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  gfkBackColor = Val(TempString)
End Sub
```

```
Sub AssignGFKDrawGrid(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  If TempString = "0" Then gfkDrawGrid = False Else gfkDrawGrid = True
End Sub
```

```
Sub AssignGFKDrawSymbols(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  If TempString = "0" Then gfkDrawSymbols = False Else gfkDrawSymbols = True
End Sub
```

```
Sub AssignGFKGridColor(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  gfkGridColor = Val(TempString)
End Sub
```

```
Sub AssignGFKLineColor(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  gfkLineColor = Val(TempString)
End Sub
```

```
Sub AssignGFKSymbolFColor(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  gfkSymbolFColor = Val(TempString)
End Sub
```

```
Sub AssignGFKSymbolOColor(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  gfkSymbolOColor = Val(TempString)
End Sub
```

```
Sub AssignGLLO(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GLLO = Val(TempString)
End Sub
```

```
Sub AssignGLLT(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GLLT = Val(TempString)
End Sub
```

```
Sub AssignGRAO(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GRAO = Val(TempString)
End Sub
```

```
Sub AssignGRAT(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GRAT = Val(TempString)
End Sub
```

```
Sub AssignGSAO(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GSAO = Val(TempString)
End Sub
```

```
Sub AssignGSAT(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GSAT = Val(TempString)
End Sub
```

```
Sub AssignGSPD(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GSPD = Val(TempString)
End Sub
```

```
Sub AssignGULO(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GULO = Val(TempString)
End Sub
```

```
Sub AssignGULT(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GULT = Val(TempString)
End Sub
```

```
Sub AssignGVLI(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  GVLI = Val(TempString)
End Sub
```

```
Sub AssignOpenPortAtStartup(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  If TempString = "0" Then OpenPortAtStartup = False Else OpenPortAtStartup = True
End Sub
```

```
Sub AssignPortParity(CurrentINILine As String)

  Dim EqualSignPos As Integer

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  PortParity = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )

End Sub
```

```
Sub AssignPortSpeed(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  PortSpeed = Val(TempString)

End Sub
```

```
Sub AssignStopBits(CurrentINILine As String)

  Dim EqualSignPos As Integer
  Dim TempString As String

  EqualSignPos = InStr(4, CurrentINILine, "=", vbTextCompare)
  TempString = Mid$(CurrentINILine, EqualSignPos + 1, Len(CurrentINILine) - EqualSignPos
  )
  StopBits = Val(TempString)
End Sub
```

```
Private Sub CalcMarker(MouseX As Single, MouseY As Single)
  Dim i, SnapIndex As Integer
  Dim SnapX1, SnapX2, SnapY1, SnapY2 As Double

  SnapX1 = MouseX - SnapSize
  SnapX2 = MouseX + SnapSize
  SnapY1 = MouseY - SnapSize
  SnapY2 = MouseY + SnapSize

  SnapIndex = -1

┌For i = 1 To UBound(PlotMatrix, 2)
│ ┌If ((PlotMatrix(0, i) > SnapX1) And (PlotMatrix(1, i) > SnapY1) And _
│ │    (PlotMatrix(0, i) < SnapX2) And (PlotMatrix(1, i) < SnapY2)) Then
│ │   SnapIndex = i
│ │   Exit For
│ └End If
└Next i

┌If SnapIndex > -1 Then
│   Label1.Caption = "X = " + Format$(DataMatrix(0, i))
│   Label2.Caption = "Y = " + Format$(DataMatrix(1, i))
│   DrawMarker SnapIndex
│   OldSnapIndex = SnapIndex
├Else
│   Grafik.ToolTipText = ""
└End If

End Sub
```

```
Private Sub CalculatePlotMatrix()

  Dim i As Long
  Dim j As Long
  Dim MaxWidth As Integer, MaxHeight As Integer
  Dim ScaleX As Double, ScaleY As Double
  Dim MinX As Double, MinY As Double, MaxX As Double, MaxY As Double
  Dim DeltaX As Double, DeltaY As Double
  Dim GrWidth As Double, GrHeight As Double
  Dim TicksX As Double, TicksY As Double
  Dim TicksX10 As Double, TicksX20 As Double, TicksX25 As Double, TicksX50 As Double,
  TicksX100 As Double
  Dim TicksY10 As Double, TicksY20 As Double, TicksY25 As Double, TicksY50 As Double,
  TicksY100 As Double
  Dim DeltaPTD As Double
  Dim TicksIntPart As String
  Dim TicksDecPart As String

  If StartedDrawing = True Then

    TicksX10 = 10
    TicksX20 = 20
    TicksX25 = 25
    TicksX50 = 50
    TicksX100 = 100

    TicksY10 = 10
    TicksY20 = 20
    TicksY25 = 25
    TicksY50 = 50
    TicksY100 = 100

    MaxWidth = Grafik.ScaleWidth
    MaxHeight = Grafik.ScaleHeight
    GrWidth = Grafik.ScaleWidth - BorderWidth * 2
    GrHeight = Grafik.ScaleHeight - BorderWidth * 2

    'Debug.Print UBound(DataMatrix, 1)
    'Debug.Print UBound(DataMatrix, 2)

    MinX = DataMatrix(0, 1)
    MaxX = DataMatrix(0, 1)
    For i = 1 To UBound(DataMatrix, 2)
      If DataMatrix(0, i) < MinX Then MinX = DataMatrix(0, i)
      If DataMatrix(0, i) > MaxX Then MaxX = DataMatrix(0, i)
    Next i

    MinY = DataMatrix(1, 1)
    MaxY = DataMatrix(1, 1)
    For i = 1 To UBound(DataMatrix, 2)
      If DataMatrix(1, i) < MinY Then MinY = DataMatrix(1, i)
      If DataMatrix(1, i) > MaxY Then MaxY = DataMatrix(1, i)
    Next i

    DeltaX = MaxX - MinX
    DeltaY = MaxY - MinY

    ScaleX = ScaleFactor * GrWidth / DeltaX
    ScaleY = ScaleFactor * GrHeight / DeltaY

    ReDim Preserve PlotMatrix(2, 1 To UBound(DataMatrix, 2))

    For i = 1 To UBound(DataMatrix, 2)
      PlotMatrix(0, i) = (BorderWidth + ((DataMatrix(0, i) - MinX) * ScaleX))
      PlotMatrix(1, i) = MaxHeight - (BorderWidth + ((DataMatrix(1, i) - MinY) * ScaleY
      ))
    Next i

    'CALCULATE AXES
    XAxeBeg = BorderWidth
    XAxeEnd = BorderWidth + GrWidth
    'XAxeZero = BorderWidth - MinX * ScaleX
    XAxeZero = BorderWidth - 10
    YAxeBeg = BorderWidth
    YAxeEnd = MaxHeight - BorderWidth
    'YAxeZero = MaxHeight - (BorderWidth - MinY * ScaleY)
1   YAxeZero = MaxHeight - BorderWidth + 10
```

```
1
    'CALCULATE TICKS

    ' ************************************************************************
    ' Proracunavanje da li 10, 20 ... odgovaraju osama ili ih treba *10 ili /10

   TicksXcount = 0
   TicksYcount = 0

  ┌Do
  │  Ret = DoEvents()
  │ ┌If DeltaX / TicksX10 < PreferredTickDensity Then
  │ │  TicksX10 = TicksX10 / 10
  │ │  TicksX20 = TicksX20 / 10
  │ │  TicksX25 = TicksX25 / 10
  │ │  TicksX50 = TicksX50 / 10
  │ │  TicksX100 = TicksX100 / 10
  │ └End If
  └Loop Until DeltaX / TicksX10 >= PreferredTickDensity

  ┌Do
  │  Ret = DoEvents()
  │ ┌If DeltaY / TicksY10 < PreferredTickDensity Then
  │ │  TicksY10 = TicksY10 / 10
  │ │  TicksY20 = TicksY20 / 10
  │ │  TicksY25 = TicksY25 / 10
  │ │  TicksY50 = TicksY50 / 10
  │ │  TicksY100 = TicksY100 / 10
  │ └End If
  └Loop Until DeltaY / TicksY10 >= PreferredTickDensity

  ┌Do
  │  Ret = DoEvents()
  │ ┌If DeltaX / TicksX100 > PreferredTickDensity Then
  │ │  TicksX10 = TicksX10 * 10
  │ │  TicksX20 = TicksX20 * 10
  │ │  TicksX25 = TicksX25 * 10
  │ │  TicksX50 = TicksX50 * 10
  │ │  TicksX100 = TicksX100 * 10
  │ └End If
  └Loop Until DeltaX / TicksX100 <= PreferredTickDensity

  ┌Do
  │  Ret = DoEvents()
  │ ┌If DeltaY / TicksY100 > PreferredTickDensity Then
  │ │  TicksY10 = TicksY10 * 10
  │ │  TicksY20 = TicksY20 * 10
  │ │  TicksY25 = TicksY25 * 10
  │ │  TicksY50 = TicksY50 * 10
  │ │  TicksY100 = TicksY100 * 10
  │ └End If
  └Loop Until DeltaY / TicksY100 <= PreferredTickDensity
   ' ************************************************************************

   TicksX = TicksX10
   DeltaPTD = DeltaX / PreferredTickDensity
   If Abs(DeltaPTD - TicksX20) < Abs(DeltaPTD - TicksX) Then TicksX = TicksX20
   If Abs(DeltaPTD - TicksX25) < Abs(DeltaPTD - TicksX) Then TicksX = TicksX25
   If Abs(DeltaPTD - TicksX50) < Abs(DeltaPTD - TicksX) Then TicksX = TicksX50
   If Abs(DeltaPTD - TicksX100) < Abs(DeltaPTD - TicksX) Then TicksX = TicksX100

   TicksY = TicksY10
   DeltaPTD = DeltaY / PreferredTickDensity
   If Abs(DeltaPTD - TicksY20) < Abs(DeltaPTD - TicksY) Then TicksY = TicksY20
   If Abs(DeltaPTD - TicksY25) < Abs(DeltaPTD - TicksY) Then TicksY = TicksY25
   If Abs(DeltaPTD - TicksY50) < Abs(DeltaPTD - TicksY) Then TicksY = TicksY50
   If Abs(DeltaPTD - TicksY100) < Abs(DeltaPTD - TicksY) Then TicksY = TicksY100

   j = Fix((MinX - AxisExtent / ScaleX) / TicksX)
   i = 1                      ' i je samo TicksCounter

  ┌Do
  │  Ret = DoEvents()
  │  ReDim Preserve TicksXData(1 To i)
  │ ┌If (j * TicksX > (MinX - AxisExtent / ScaleX)) And (j * TicksX < (MaxX +
  │ AxisExtent / ScaleX)) Then
  │ │  TicksXData(i) = j * TicksX
1 2 3  'Debug.Print "x:  i=" + Format(i) + ", " + "j=" + Format(j)
```

```
    Debug.Print "tickx: " + Format(j * TicksX)
    i = i + 1
  End If
  j = j + 1

Loop Until (j * TicksX > (MaxX + AxisExtent / ScaleX))

ReDim Preserve TicksXPlot(1 To UBound(TicksXData))
For i = 1 To UBound(TicksXData)
  TicksXPlot(i) = (BorderWidth + ((TicksXData(i) - MinX) * ScaleX))
Next i

j = Fix((MinY - AxisExtent / ScaleY) / TicksY)
i = 1                    ' i je samo TicksCounter
Do
  Ret = DoEvents()
  ReDim Preserve TicksYData(1 To i)
  If (j * TicksY > (MinY - AxisExtent / ScaleY)) And (j * TicksY < (MaxY +
  AxisExtent / ScaleY)) Then
    TicksYData(i) = j * TicksY
    'Debug.Print "y:  i=" + Format(i) + ", " + "j=" + Format(j)
    Debug.Print "ticky: " + Format(j * TicksY)
    i = i + 1
  End If
  j = j + 1

Loop Until (j * TicksY > (MaxY + AxisExtent / ScaleY))

ReDim Preserve TicksYPlot(1 To UBound(TicksYData))
For i = 1 To UBound(TicksYData)
  TicksYPlot(i) = MaxHeight - (BorderWidth + ((TicksYData(i) - MinY) * ScaleY))
Next i

' ODREDJIVANJE FORMATA ZA ISPISIVANJE VREDNOSTI NA OSAMA (npr. 52.5 53.0 53.5 ...)

TicksIntPart = ""
TicksDecPart = ""
TicksXformat = ""
TicksYformat = ""

i = InStr(Format(TicksX), ".")
If i = 0 Then
  TicksDecPart = ""
Else
  TicksDecPart = Right$(Format$(TicksX), Len(Format$(TicksX)) - i)
End If

TicksXformat = "0"

If TicksDecPart <> "" Then
  TicksXformat = TicksXformat + "."
  For j = 1 To Len(TicksDecPart)
    TicksXformat = TicksXformat + "0"
  Next j
End If

i = InStr(Format(TicksY), ".")
If i = 0 Then
  TicksDecPart = ""
Else
  TicksDecPart = Right$(Format$(TicksY), Len(Format$(TicksY)) - i)
End If

TicksYformat = "0"

If TicksDecPart <> "" Then
  TicksYformat = TicksYformat + "."
  For j = 1 To Len(TicksDecPart)
    TicksYformat = TicksYformat + "0"
  Next j
End If

End If
'Debug.Print "TicksXformat = " + TicksXformat
'Debug.Print "TicksYformat = " + TicksYformat
End Sub
```

```
Private Sub CloseCOMPort()

  MSComm1.PortOpen = False

  DisablePortDependentButtons
  RefreshStatusBar
End Sub
```

```
Private Sub cmdGrid_Click()
  If gfkDrawGrid = True Then
    gfkDrawGrid = False
    cmdGrid.Caption = "Show grid"
    DrawGrafik
    Exit Sub
  End If
  If gfkDrawGrid = False Then
    gfkDrawGrid = True
    cmdGrid.Caption = "Hide grid"
    DrawGrafik
    Exit Sub
  End If
  SaveINIFile
End Sub
```

```
Private Sub cmdSymbols_Click()
  If gfkDrawSymbols = True Then
    gfkDrawSymbols = False
    cmdSymbols.Caption = "Show symbols"
    Grafik.ToolTipText = ""
    DrawGrafik
    Exit Sub
  End If
  If gfkDrawSymbols = False Then
    gfkDrawSymbols = True
    cmdSymbols.Caption = "Hide symbols"
    DrawGrafik
    Exit Sub
  End If
  SaveINIFile
End Sub
```

```
Sub CreateINIFile()

  Dim fs, INIFile

  Set fs = CreateObject("Scripting.FileSystemObject")
  On Error GoTo FileError
  Set INIFile = fs.CreateTextFile(INIFileName, True)

  INIFile.WriteLine ("Port=1")
  INIFile.WriteLine ("Speed=9600")
  INIFile.WriteLine ("DataBits=8")
  INIFile.WriteLine ("Parity=N")
  INIFile.WriteLine ("StopBits=1")
  INIFile.WriteLine ("OpenPortAtStartup=1")
  INIFile.WriteLine ("GVLI=60")
  INIFile.WriteLine ("GCLI=40")
  INIFile.WriteLine ("GULT=180")
  INIFile.WriteLine ("GULO=180")
  INIFile.WriteLine ("GLLT=0")
  INIFile.WriteLine ("GLLO=0")
  INIFile.WriteLine ("GRAT=13")
  INIFile.WriteLine ("GRAO=13")
  INIFile.WriteLine ("GSAT=10")
  INIFile.WriteLine ("GSAO=10")
  INIFile.WriteLine ("GDTT=2")
  INIFile.WriteLine ("GSPD=7")
  INIFile.WriteLine ("gfBC=0")
  INIFile.WriteLine ("gfAC=65535")
  INIFile.WriteLine ("gfLC=65535")
  INIFile.WriteLine ("gfSO=65535")
  INIFile.WriteLine ("gfSF=33023")
  INIFile.WriteLine ("gfGC=32896")
  INIFile.WriteLine ("gfDS=1")
  INIFile.WriteLine ("gfDG=1")
  INIFile.Close
  Exit Sub

FileError:
  MsgBox ("File error!")

End Sub
```

```
Private Function CTime(NumberOfSeconds As Double) As String
  Dim Days As Long
  Dim Hours As Long
  Dim Minutes As Long
  Dim Seconds As Long
  Dim CTimeFormat As String

  Days = Fix(NumberOfSeconds / 86400)
  Hours = Fix((NumberOfSeconds - Days * 86400) / 3600)
  Minutes = Fix((NumberOfSeconds - Days * 86400 - Hours * 3600) / 60)
  Seconds = Fix((NumberOfSeconds - Days * 86400 - Hours * 3600 - Minutes * 60))

  If Days = 0 Then
    CTimeFormat = Format$(Hours) + ":" + Format$(Minutes) + ":" + Format$(Seconds)
  Else
    CTimeFormat = Format$(Days) + "d, " + Format$(Hours) + ":" + Format$(Minutes) + ":"
    + Format$(Seconds)
  End If
  CTime = CTimeFormat

End Function
```

```
Private Sub DisablePortDependentButtons()
  Toolbar1.Buttons.Item(4).Enabled = True
  Toolbar1.Buttons.Item(5).Enabled = False
  Toolbar1.Buttons.Item(7).Enabled = False
  Toolbar1.Buttons.Item(8).Enabled = False
  Toolbar1.Buttons.Item(9).Enabled = False
  Toolbar1.Buttons.Item(10).Enabled = False
  Toolbar1.Buttons.Item(11).Enabled = False
  mnuToolsOpenPort.Enabled = True  ' Toolbar(4)
  mnuToolsClosePort.Enabled = False  ' Toolbar(4)
  mnuToolsReadAngles.Enabled = False ' Toolbar(9)
  mnuToolsDriveTheta.Enabled = False ' Toolbar(10)
  mnuMeasStep.Enabled = False        ' Toolbar(7)
  mnuMeasCont.Enabled = False        ' Toolbar(8)
  mnuMeasStop.Enabled = False       ' Toolbar(11)
  mnuToolsReset.Enabled = False
End Sub
```

```
Private Sub DrawGrafik() ' ISCRTAVANJE GRAFIKA

  Dim i As Integer
  CalculatePlotMatrix
  Grafik.BackColor = gfkBackColor
  Grafik.Cls

  ' Draw axis ticks
  Grafik.ForeColor = gfkAxisColor
  For i = 1 To UBound(TicksXPlot)
    Grafik.Line (TicksXPlot(i), YAxeEnd + AxisExtent)-(TicksXPlot(i), YAxeEnd +
    AxisExtent - TickWidth), gfkAxisColor
    Grafik.CurrentX = TicksXPlot(i) - (Grafik.TextWidth(Format$(TicksXData(i),
    TicksXformat)) / 2)
    Grafik.CurrentY = YAxeZero + 12
    Grafik.Print Format$(TicksXData(i), TicksXformat)
  Next i

  For i = 1 To UBound(TicksYPlot)
    Grafik.Line (XAxeBeg - AxisExtent + TickWidth, TicksYPlot(i))-(XAxeBeg - AxisExtent
    , TicksYPlot(i)), gfkAxisColor
    Grafik.CurrentX = XAxeZero - Grafik.TextWidth(Format$(TicksYData(i), TicksYformat))
    - 13
    Grafik.CurrentY = TicksYPlot(i) - (Grafik.TextHeight(Format$(TicksYData(i),
    TicksYformat)) / 2)
    Grafik.Print Format$(TicksYData(i), TicksYformat)
  Next i
  Grafik.ForeColor = gfkLineColor

  ' Draw grid
  If gfkDrawGrid = True Then
    Grafik.DrawStyle = vbDot
    For i = 1 To UBound(TicksXPlot)
      Grafik.Line (TicksXPlot(i), YAxeBeg - AxisExtent + 1)-(TicksXPlot(i), YAxeEnd +
      AxisExtent - TickWidth + 1), gfkGridColor
    Next i

    For i = 1 To UBound(TicksYPlot)
      Grafik.Line (XAxeBeg - AxisExtent + TickWidth, TicksYPlot(i))-(XAxeEnd +
      AxisExtent, TicksYPlot(i)), gfkGridColor
    Next i

    Grafik.DrawStyle = vbSolid
  End If

  ' Draw axes
  Grafik.Line (XAxeBeg - AxisExtent, YAxeEnd + AxisExtent)-(XAxeEnd + AxisExtent + 1,
  YAxeEnd + AxisExtent), gfkAxisColor 'X
  Grafik.Line (XAxeBeg - AxisExtent, YAxeBeg - AxisExtent)-(XAxeBeg - AxisExtent,
  YAxeEnd + AxisExtent), gfkAxisColor  'Y
  Grafik.Line (XAxeBeg - AxisExtent, YAxeBeg - AxisExtent)-(XAxeEnd + AxisExtent,
  YAxeBeg - AxisExtent), gfkAxisColor 'X
  Grafik.Line (XAxeEnd + AxisExtent, YAxeBeg - AxisExtent)-(XAxeEnd + AxisExtent,
  YAxeEnd + AxisExtent), gfkAxisColor 'Y

  ' Draw lines
  Grafik.ForeColor = gfkLineColor
  For i = 1 To UBound(PlotMatrix, 2) - 1
    Grafik.Line (PlotMatrix(0, i), PlotMatrix(1, i))-(PlotMatrix(0, i + 1), PlotMatrix(1
    , i + 1))
  Next i

  ' Draw symbols
  Grafik.FillColor = gfkSymbolFColor
  If gfkDrawSymbols = True Then
    For i = 1 To UBound(DataMatrix, 2)
      Grafik.Circle (PlotMatrix(0, i), PlotMatrix(1, i)), 3, gfkSymbolOColor
    Next i
  End If

End Sub
```

```
Sub DrawMarker(SnapIndex As Integer)
  Dim MarkerSize As Integer
  Dim MarkerX1, MarkerX2, MarkerY1, MarkerY2 As Double

  MarkerSize = 10
  MarkerX1 = PlotMatrix(0, SnapIndex) - MarkerSize / 2
  MarkerX2 = PlotMatrix(0, SnapIndex) + MarkerSize / 2
  MarkerY1 = PlotMatrix(1, SnapIndex) - MarkerSize / 2
  MarkerY2 = PlotMatrix(1, SnapIndex) + MarkerSize / 2

  Grafik.DrawMode = vbXorPen
  Grafik.Line (MarkerX1 + 1, MarkerY1)-(MarkerX2, MarkerY1), RGB(230, 230, 0)
  Grafik.Line (MarkerX1, MarkerY2)-(MarkerX2 + 1, MarkerY2), RGB(230, 230, 0)
  Grafik.Line (MarkerX1, MarkerY1)-(MarkerX1, MarkerY2), RGB(230, 230, 0)
  Grafik.Line (MarkerX2, MarkerY1)-(MarkerX2, MarkerY2), RGB(230, 230, 0)
  Grafik.DrawMode = vbCopyPen

  Grafik.ToolTipText = " (" + Format$(DataMatrix(0, SnapIndex)) + ", " + Format$(
  DataMatrix(1, SnapIndex)) + ") "

End Sub
```

```
Private Sub EnablePortDependentButtons()
  Toolbar1.Buttons.Item(4).Enabled = False
  Toolbar1.Buttons.Item(5).Enabled = True
  Toolbar1.Buttons.Item(7).Enabled = True
  Toolbar1.Buttons.Item(8).Enabled = True
  Toolbar1.Buttons.Item(9).Enabled = True
  Toolbar1.Buttons.Item(10).Enabled = True
  Toolbar1.Buttons.Item(11).Enabled = False
  mnuToolsOpenPort.Enabled = False  ' Toolbar(4)
  mnuToolsClosePort.Enabled = True  ' Toolbar(4)
  mnuToolsReadAngles.Enabled = True ' Toolbar(9)
  mnuToolsDriveTheta.Enabled = True ' Toolbar(10)
  mnuMeasStep.Enabled = True        ' Toolbar(7)
  mnuMeasCont.Enabled = True        ' Toolbar(8)
  mnuMeasStop.Enabled = False       ' Toolbar(11)
  mnuToolsReset.Enabled = True
End Sub
```

```
Private Sub ExportDATA(DATAFileName As String)
  Const ForReading = 1, ForWriting = 2, ForAppending = 3
  Const Tristatefalse = 0
  Dim fs, DATAfile
  Dim i As Integer, j As Integer
  Dim dataXstart As Double, dataXend As Double, dataXstep As Double
  Dim TempLine As String

  Set fs = CreateObject("Scripting.FileSystemObject")
  Set DATAfile = fs.CreateTextFile(DATAFileName)

  dataXstart = DataMatrix(0, 1)
  dataXend = DataMatrix(0, UBound(DataMatrix, 2))
  dataXstep = DataMatrix(0, 2) * 10000

  dataXstep = dataXstep - DataMatrix(0, 1) * 10000
  dataXstep = Round(dataXstep)
  dataXstep = dataXstep / 10000

  TempLine = FormatFix83(dataXstart) + FormatFix83(dataXstep) + FormatFix83(dataXend)

  DATAfile.WriteLine TempLine

  i = 1

  Do While i <= UBound(DataMatrix, 2)
    TempLine = ""
    j = 1
    Do While (j <= 10) And (i <= UBound(DataMatrix, 2))
      TempLine = TempLine + FormatFix8(DataMatrix(1, i))
      j = j + 1
      i = i + 1
    Loop
    TempLine = TempLine + Chr(13) + Chr(10)
    DATAfile.Write TempLine
  Loop

  DATAfile.Close

End Sub
```

```
Private Sub Form_Load()

  frmGlavna.Caption = App.Title

  CommonDialog1.InitDir = App.Path
  CommonDialog1.Filter = "Data files (*.dat;*.asc)|*.dat;*.asc|All Files (*.*)|*.*"
  'Otvaranje INI fajla
  INIFileName = App.Path + "\seifert.ini"
  OpenINIFile
  Context = Idle
  MEASFileName = ""
  RefreshStatusBar

  ScaleFactor = 1
  OldSnapIndex = -1
  Grafik.ScaleMode = vbPixels
  Grafik.BackColor = gfkBackColor

  '***********************************
  'GRAFIK SA MERENJEM
  StartedDrawing = False
  ReDim DataMatrix(2, 1 To 1)
  MeasurementCount = 0

  'GRAFIK BEZ MERENJA
  'StartedDrawing = True
  'DATAFileName = App.Path + "\data.asc"
  'LoadDATA DATAFileName
  'DrawGrafik
  '***********************************

  RefreshShowHideButtons
  DisablePortDependentButtons

  DriveThetaTo = -1
  gcmsTime = -1
  gcmsSpeed = -1

  SafeToExit = True

  frmGlavna.Show

  If OpenPortAtStartup = True Then OpenCOMPort
  'dlgInitAttempt.Show vbModal
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
  If SafeToExit = False Then
    If MsgBox("Program is busy waiting for a device response. Please wait.",
    vbExclamation Or vbOKOnly, "Program busy") = vbOK Then
      Cancel = True
    End If
  End If
  If SafeToExit = True Then
    If MsgBox("Are you sure you want to end this session?", vbQuestion Or vbYesNo, "Exit
     program?") = vbNo Then
      'Stop the exit
      Cancel = True
    End If
  End If

End Sub
```

```
Private Sub Form_Resize()
  Const MIN_WIDTH = 11790
  Const MIN_HEIGHT = 8550
 ┌If WindowState <> 1 Then
 │  If Width < MIN_WIDTH Then Width = MIN_WIDTH
 │  If Height < MIN_HEIGHT Then Height = MIN_HEIGHT
 │  Grafik.Width = frmGlavna.Width - 4335
 │  Grafik.Height = frmGlavna.Height - 2820
 │  Grafik.ScaleMode = vbPixels
 │  Grafik.ScaleWidth = (frmGlavna.Width - 4335) / 15
 │  Grafik.ScaleHeight = (frmGlavna.Height - 2820) / 15
 │  'FrameMeas.Width = (frmGlavna.Width - (11790 - 6135))
 │  FrameXY.Left = (frmGlavna.Width - (11790 - 8400))
 │  cmdGrid.Left = (frmGlavna.Width - (11790 - 10320))
 │  cmdSymbols.Left = (frmGlavna.Width - (11790 - 10320))
 │  If StartedDrawing = True Then DrawGrafik
 │
 │  TerminalWindow.Height = frmGlavna.Height - 2790
 └End If
End Sub
```

```
Private Function FormatFix8(NumToFormat As Double) As String
  Dim i As Integer
  Dim TempString As String

 ┌If Len(Format$(NumToFormat)) > 8 Then
 │  FormatFix8 = "        "
 │  Exit Function
 └End If

 ┌For i = 1 To 8 - Len(Format$(NumToFormat))
 │  TempString = TempString + " "
 └Next i
  TempString = TempString + Format$(NumToFormat)

  FormatFix8 = TempString
End Function
```

```
Private Function FormatFix83(NumToFormat As Double) As String
  Dim i As Integer
  Dim TempString As String

 ┌If Len(Format$(NumToFormat)) > 8 Then
 │  FormatFix83 = "        "
 │  Exit Function
 └End If

 ┌For i = 1 To 8 - Len(Format$(NumToFormat, "#0.000"))
 │  TempString = TempString + " "
 └Next i
  TempString = TempString + Format$(NumToFormat, "#0.000")

  FormatFix83 = TempString
End Function
```

```
Private Sub frmGlavna_Unload(Cancel As Integer)
  MSComm1.Output = ""
  MSComm1.PortOpen = False
  UnloadAllForms

End Sub
```

```
Private Sub gcomControllerReset()
  Dim i As Integer

  MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
  gcomWaitFor (">")
  'Do
  ' Ret = DoEvents()
  ' i = i + 1
  'Loop Until i > 30000
  MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
  gcomWaitFor (">")
  ' Do
  '  Ret = DoEvents()
  '  i = i + 1
  ' Loop Until i > 30000
  MSComm1.Output = "BYE" + Chr$(13) + Chr$(10)
  gcomWaitFor (">")
  'Do
  ' Ret = DoEvents()
  ' i = i + 1
  'Loop Until i > 30000
End Sub
```

```
Private Sub gcomDriveMotors(Angle As Double)
  Dim TextToSend As String
  Context = DriveMotors
  RefreshStatusBar

  BigBuffer = ""
  TextToSend = "/DM " + Format$(Angle) + " "

  MSComm1.Output = TextToSend
  gcomWaitFor "  SP "
  MSComm1.Output = "7 "
  gcomWaitFor Chr$(13)
  BigBuffer = ""
  gcomReadAngles
  gcomWaitFor (Chr$(13))
  InterpretAngles BigBuffer
  BigBuffer = ""

  Context = Idle
  RefreshStatusBar
End Sub
```

```
Private Sub gcomDriveOmega(Angle As Double)
  Dim TextToSend As String

  Context = DriveOmega
  RefreshStatusBar

  TextToSend = "/DO " + Format$(Angle) + " "

  MSComm1.Output = TextToSend
  gcomWaitFor "  SP "
  MSComm1.Output = "7 "
  gcomWaitFor ("7 " + Chr$(13))

  Context = Idle
  RefreshStatusBar
End Sub
```

```
Private Sub gcomDriveTheta(Angle As Double)
  Dim TextToSend As String

  Context = DriveTheta
  RefreshStatusBar

  TextToSend = "/DT " + Format$(Angle) + " "

  MSComm1.Output = TextToSend
  gcomWaitFor "  SP "
  MSComm1.Output = "7 "
  gcomWaitFor ("7 " + Chr$(13))

  Context = Idle
  RefreshStatusBar
End Sub
```

```
Private Sub gcomInitialize()

  Context = Init
  RefreshStatusBar
  MSComm1.Output = "Y"

  gcomWaitFor "/VL "
  MSComm1.Output = Format$(GVLI) + " "
  gcomWaitFor "/CL "
  MSComm1.Output = Format$(GCLI) + " "
  gcomWaitFor "/UL T "
  MSComm1.Output = Format$(GULT) + " "
  gcomWaitFor "/UL T " + Format$(GULT) + "   O "
  MSComm1.Output = Format$(GULO) + " "
  gcomWaitFor "/LL T "
  MSComm1.Output = Format$(GLLT) + " "
  gcomWaitFor "/LL T " + Format$(GLLT) + "   O "
  MSComm1.Output = Format$(GLLO) + " "
  gcomWaitFor "/RA T "
  MSComm1.Output = Format$(GRAT) + " "
  gcomWaitFor "/RA T " + Format$(GRAT) + "   O "
  MSComm1.Output = Format$(GRAO) + " "
  gcomWaitFor "/SA T "
  MSComm1.Output = Format$(GSAT) + " "
  gcomWaitFor "/SA T " + Format$(GSAT) + "   O "
  MSComm1.Output = Format$(GSAO) + " "
  gcomWaitFor ("/DR" + Chr$(13))
  BigBuffer = ""
  gcomReadAngles
  gcomWaitFor (Chr$(13))
  InterpretAngles BigBuffer
  BigBuffer = ""
  gcomDriveTheta Format$(GDTT) + " "
  BigBuffer = ""
  gcomDriveOmega Format$(GDTT / 2) + " "
  BigBuffer = ""
  gcomReadAngles
  gcomWaitFor (Chr$(13))
  InterpretAngles BigBuffer
  BigBuffer = ""
End Sub
```

```
'*************************
'
' KONTINUALNO MERENJE
'
'*************************

Private Sub gcomMeasCont(gcmStart As Double, gcmEnd As Double, gcmSpeed As Integer)
  Dim i As Long

  Const ForReading = 1, ForWriting = 2, ForAppending = 3
  Const Tristatefalse = 0
  Dim fs

  Dim CurrentDate, CurrentTime
  Dim ActualSpeed As Double
  Dim lhDC As Long, lCtlWidth As Long
  Dim FileName As String

  ' Otvaranje datoteke sa podacima
  If MEASFileName = "" Then
    CurrentDate = Date
    CurrentTime = Time
    MEASFileName = App.Path + "\" + Format$(CurrentDate, "YYYYMMDD") + "-" + Format$(
    CurrentTime, "HHMMSS") + ".dat"
  End If

  ' Ako treba, skrati ime fajla za prikaz
  FileName = MEASFileName
  lCtlWidth = lblFileName.Width / 15 - Me.DrawWidth
  lhDC = Me.hDC
  PathCompactPath lhDC, FileName, lCtlWidth

  Select Case gcmSpeed
    Case Is = 1
      ActualSpeed = 0.1
    Case Is = 2
      ActualSpeed = 0.2
    Case Is = 3
      ActualSpeed = 0.4
    Case Is = 4
      ActualSpeed = 0.8
    Case Is = 5
      ActualSpeed = 1.6
    Case Is = 6
      ActualSpeed = 3.2
    Case Is = 7
      ActualSpeed = 200
  End Select

  For i = 1 To 12
    Toolbar1.Buttons.Item(i).Enabled = False
  Next i

  lblFileName = "Filename: " + FileName
  lblStart = "Start angle: " + Format$(gcmStart) + " °"
  lblEnd = "End angle:  " + Format$(gcmEnd) + " °"
  lblStep = "Speed: " + Format$(ActualSpeed) + " °/min"
  lblNoOfSteps = ""
  lblTime = ""

  ProgressBar1.Enabled = True
  ProgressBar1.Min = 0
  ProgressBar1.Max = Round((gcmEnd - gcmStart) / 0.1)
  gcmsEstTime = Round(60 * (gcmEnd - gcmStart) / ActualSpeed)
  lblEstTime = "Est. time remaining: " + Format$(CTime(gcmsEstTime), "HH:MM:SS")

  Set fs = CreateObject("Scripting.FileSystemObject")
  Set MEASFile = fs.CreateTextFile(MEASFileName)

  ReDim DataMatrix(2, 1 To 1)
  StartedDrawing = False
  Grafik.Cls
  Grafik.ToolTipText = ""
  RefreshShowHideButtons

  gcomDriveMotors gcmStart
```

```
BigBuffer = ""
MSComm1.Output = "/MB "
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "10 ST=" + Format$(gcmSpeed) + "; SO=" + Format$(gcmSpeed) + Chr$(13
) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "20 SV=0" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "30 CALL .PS" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "30 TN=TA+0.1" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "40 CALL .ES" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "50 CALL .DM" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "60 CALL .DS" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "70 CALL .TS" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "80 PRINT " + Chr$(34) + "T =" + Chr$(34) + ",TA-0.05," + Chr$(34) +
", C =" + Chr$(34) + ",SV," + Chr$(34) + " ResOK" + Chr$(34) + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "90 IF TA<" + Format$(gcmEnd) + " GOTO 20" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "100 PRINT " + Chr$(34) + "** MEASUREMENT END **" + Chr$(34) + Chr$(
13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "110 STOP" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

Context = Measurement
RefreshStatusBar
mnuMeasStop.Enabled = True
Toolbar1.Buttons.Item(11).Enabled = True
Timer1.Enabled = True

BigBuffer = ""
MSComm1.Output = "RUN" + Chr$(13) + Chr$(10)
BigBuffer = ""
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "BYE" + Chr$(13) + Chr$(10)
gcomWaitFor Chr$(13)
```

```
  Context = Idle
  RefreshStatusBar

  Timer1.Enabled = False
  BigBuffer = ""
  gcomReadAngles
  gcomWaitFor (Chr$(13))
  InterpretAngles BigBuffer
  BigBuffer = ""

  MeasurementCount = 0
  lblCount = ""
  lblFileName = "Filename: "
  lblStart = "Start angle: "
  lblEnd = "End angle: "
  lblStep = "Step: "
  lblTime = "Time: "
  lblEstTime = "Est. time remaining: "
  lblNoOfSteps = "Number of steps: "

  ProgressBar1.Enabled = False
  ProgressBar1.Min = 0
  ProgressBar1.Max = 1
  ProgressBar1.Value = 0
  mnuMeasStop.Enabled = False

  For i = 1 To 12
    Toolbar1.Buttons.Item(i).Enabled = True
  Next i
  Toolbar1.Buttons.Item(11).Enabled = False

  MEASFile.Close
  MEASFileName = ""
End Sub
```

```
'*************************
'
' STEP MERENJE
'
'*************************

Private Sub gcomMeasStep(gcmStart As Double, gcmEnd As Double, gcmStep As Double,
gcmTime As Double)

  Dim i As Long

  Const ForReading = 1, ForWriting = 2, ForAppending = 3
  Const Tristatefalse = 0
  Dim fs

  Dim CurrentDate, CurrentTime
  Dim lhDC As Long, lCtlWidth As Long
  Dim FileName As String

  ' Otvaranje datoteke sa podacima

  If MEASFileName = "" Then
    CurrentDate = Date
    CurrentTime = Time
    MEASFileName = App.Path + "\" + Format$(CurrentDate, "YYYYMMDD") + "-" + Format$(
    CurrentTime, "HHMMSS") + ".dat"
  End If

  ' Ako treba, skrati ime fajla za prikaz *********
  FileName = MEASFileName
  lCtlWidth = lblFileName.Width / 15 - Me.DrawWidth - 55
  lhDC = Me.hDC
  PathCompactPath lhDC, FileName, lCtlWidth

  For i = 1 To 10
    Toolbar1.Buttons.Item(i).Enabled = False
  Next i
  Toolbar1.Buttons.Item(13).Enabled = False

  lblFileName = "Filename: " + FileName
  lblStart = "Start angle: " + Format$(gcmStart) + " °"
  lblEnd = "End angle:  " + Format$(gcmEnd) + " °"
  lblStep = "Step: " + Format$(gcmStep) + " °"
  lblTime = "Time: " + Format$(gcmTime)
  lblNoOfSteps = "Number of steps: " + Format$(Round((gcmEnd - gcmStart) / gcmStep) + 1)
  ProgressBar1.Enabled = True
  ProgressBar1.Min = 0
  ProgressBar1.Max = Round((gcmEnd - gcmStart) / gcmStep) + 1
  gcmsEstTime = gcmTime * (Round((gcmEnd - gcmStart) / gcmStep) + 1) + 6 / 20 * Round((
  gcmEnd - gcmStart) / gcmStep)
  lblEstTime = "Est. time remaining: " + Format$(CTime(gcmsEstTime), "HH:MM:SS")

  Set fs = CreateObject("Scripting.FileSystemObject")
  Set MEASFile = fs.CreateTextFile(MEASFileName)

  ReDim DataMatrix(2, 1 To 1)
  StartedDrawing = False
  Grafik.Cls
  Grafik.ToolTipText = ""
  RefreshShowHideButtons

  gcomDriveMotors gcmStart

  BigBuffer = ""
  MSComm1.Output = "/MB "
  gcomWaitFor ">"

  BigBuffer = ""
  MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
  gcomWaitFor ">"

  BigBuffer = ""
  MSComm1.Output = "10 TI=" + Format$(gcmTime) + Chr$(13) + Chr$(10)
  gcomWaitFor ">"

  BigBuffer = ""
  MSComm1.Output = "20 SV=0; ST=7; SO=7" + Chr$(13) + Chr$(10)
```

```
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "30 CALL .PT" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "40 CALL .PS" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "50 FOR J =" + Format$(gcmStart) + " TO " + Format$(gcmEnd) + " STEP
" + Format$(gcmStep) + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "60 TN = J" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "70 CALL .DM" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "80 CALL .ST" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "90 CALL .TS" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "100 PRINT " + Chr$(34) + "T =" + Chr$(34) + ",TA," + Chr$(34) + ", C
 =" + Chr$(34) + ",SV," + Chr$(34) + " ResOK" + Chr$(34) + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "110 NEXT J" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "120 PRINT " + Chr$(34) + "** MEASUREMENT END **" + Chr$(34) + Chr$(
13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "130 STOP" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

Context = Measurement
RefreshStatusBar
mnuMeasStop.Enabled = True
Toolbar1.Buttons.Item(11).Enabled = True
Timer1.Enabled = True

BigBuffer = ""
MSComm1.Output = "RUN" + Chr$(13) + Chr$(10)
BigBuffer = ""
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
gcomWaitFor ">"

BigBuffer = ""
MSComm1.Output = "BYE" + Chr$(13) + Chr$(10)
gcomWaitFor Chr$(13)

Context = Idle
RefreshStatusBar

Timer1.Enabled = False
BigBuffer = ""
gcomReadAngles
gcomWaitFor (Chr$(13))
InterpretAngles BigBuffer
BigBuffer = ""
```

```
  MeasurementCount = 0
  lblCount = ""
  lblFileName = "Filename: "
  lblStart = "Start angle: "
  lblEnd = "End angle: "
  lblStep = "Step: "
  lblTime = "Time: "
  lblEstTime = "Est. time remaining: "
  lblNoOfSteps = "Number of steps: "

  ProgressBar1.Enabled = False
  ProgressBar1.Min = 0
  ProgressBar1.Max = 1
  ProgressBar1.Value = 0
  mnuMeasStop.Enabled = False

 ┌For i = 1 To 10
 │  Toolbar1.Buttons.Item(i).Enabled = True
 └Next i
  Toolbar1.Buttons.Item(13).Enabled = True
  Toolbar1.Buttons.Item(11).Enabled = False

  MEASFile.Close
  MEASFileName = ""

End Sub
```

```
Private Sub gcomReadAngles()
  MSComm1.Output = "/OA "
End Sub
```

```
Private Sub gcomWaitFor(TempString As String)
  Dim i As Integer

  i = 0
  SafeToExit = False
 ┌Do
 │  Ret = DoEvents()
 └Loop Until InStr(BigBuffer, TempString)

 ┌Do
 │  Ret = DoEvents()
 │  i = i + 1
 └Loop Until i > 10000

  SafeToExit = True
End Sub
```

```
Sub Grafik_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

 ┌If StartedDrawing = True Then
 │ ┌If (UBound(DataMatrix, 2) > 1) And (gfkDrawSymbols = True) Then
 │ │ ┌If OldSnapIndex > -1 Then
 │ │ │  Label1.Caption = "X = "
 │ │ │  Label2.Caption = "Y = "
 │ │ │  DrawMarker OldSnapIndex
 │ │ │  OldSnapIndex = -1
 │ │ └End If
 │ │  CalcMarker X, Y
 │ └End If
 └End If
End Sub
```

```
Private Sub InterpretAngles(TextLine As String)

  Dim TSignPos As Integer
  Dim OSignPos As Integer
  Dim CommaSignPos As Integer
  Dim TempIntPart As Integer
  Dim TempDecPart As Integer

  Dim ThetaString As String
  Dim OmegaString As String
  Dim ThetaAngle As Double
  Dim OmegaAngle As Double

  If TextLine = "" Then Exit Sub

  TSignPos = InStr(3, TextLine, "T")
  OSignPos = InStr(3, TextLine, "O")
  ThetaString = Trim$(Mid$(TextLine, TSignPos + 2, (OSignPos - TSignPos - 2)))
  OmegaString = Trim$(Mid$(TextLine, OSignPos + 2, (Len(TextLine) - OSignPos - 3)))

  CommaSignPos = InStr(1, ThetaString, ",", vbTextCompare)
  TempIntPart = Val(Left$(ThetaString, Len(ThetaString) - CommaSignPos - 1))
  TempDecPart = Val(Right$(ThetaString, Len(ThetaString) - CommaSignPos))
  ThetaAngle = TempIntPart + TempDecPart / 1000

  CommaSignPos = InStr(1, OmegaString, ",", vbTextCompare)
  TempIntPart = Val(Left$(OmegaString, Len(OmegaString) - CommaSignPos - 1))
  TempDecPart = Val(Right$(OmegaString, Len(OmegaString) - CommaSignPos))
  OmegaAngle = TempIntPart + TempDecPart / 1000

  GSAT = ThetaAngle
  GSAO = OmegaAngle
  gcmsCurrentAngle = GSAT

  RefreshThetaOmega
  SaveINIFile

End Sub
```

```
Private Sub InterpretMeasurement(TextLine As String)

  Dim TSignPos As Integer
  Dim CSignPos As Integer
  Dim CommaSignPos As Integer
  Dim TempIntPart As Integer
  Dim TempDecPart As Integer
  Dim strTempIntPart As String
  Dim strTempDecPart As String

  Dim ThetaString As String
  Dim CountString As String

  'TextLine = Left$(TextLine, 26)

  TSignPos = InStr(1, TextLine, "T")
  CSignPos = InStr(3, TextLine, "C")
  ThetaString = Trim$(Mid$(TextLine, TSignPos + 3, (CSignPos - TSignPos - 5)))
  CountString = Trim$(Mid$(TextLine, CSignPos + 3, (Len(TextLine) - CSignPos - 9)))

  CommaSignPos = InStr(1, ThetaString, ".", vbTextCompare)
  TempIntPart = Val(Left$(ThetaString, CommaSignPos - 1))
  TempDecPart = Val(Right$(ThetaString, Len(ThetaString) - CommaSignPos))
  gcmsCurrentAngle = TempIntPart + TempDecPart / 1000

  CommaSignPos = InStr(1, CountString, ".", vbTextCompare)
  strTempIntPart = Left$(CountString, CommaSignPos - 1)
  strTempDecPart = Right$(CountString, Len(CountString) - CommaSignPos)
  gcmsCount = Val(strTempIntPart * 1000 + strTempDecPart)
End Sub
```

```
Private Sub LoadDATA(DATAFileName As String) ' UCITAVANJE PODATAKA ZA GRAFIK
  Const ForReading = 1, ForWriting = 2, ForAppending = 8
  Const Tristatefalse = 0
  Dim DATAfileF10
  Dim fs, DATAfile
  Dim CurrentDATAline As String
  Dim CommaSignPos As Integer
  Dim TempString As String
  Dim dataX, dataY As Double
  Dim DataSetLen As Long

  Dim dataXstart As Double, dataXend As Double, dataXstep As Double, dataXcurrent As
  Double

  Dim i As Integer, j As Integer

  ' Otvaranje datoteke sa podacima
  Set fs = CreateObject("Scripting.FileSystemObject")
  Set DATAfile = fs.OpenTextFile(DATAFileName, ForReading, Tristatefalse)

  'Odredjivanje tipa fajla - F2 ili F10

  DataSetLen = 0
  CurrentDATAline = DATAfile.ReadLine

  If nStrCount(CompressSpaces(CurrentDATAline), " ") > 1 Then DATAfileF10 = True

  ' F10 - Citanje podataka iz datoteke u matricu, ReDim matrice

  If DATAfileF10 = True Then

    ' Prva linija je vec ucitana, uzmi parametre

    TempString = CurrentDATAline

    'i = InStr(1, TempString, " ", vbTextCompare)
    dataXstart = Val(Mid$(TempString, 1, 8))

    'j = InStr(i + 1, TempString, " ", vbTextCompare)
    dataXstep = Val(Mid$(TempString, 9, 8))

    'k  = InStr(j + 1, TempString, " ", vbTextCompare)
    dataXend = Val(Mid$(TempString, 17, 8))

    dataXcurrent = dataXstart

    ' Ucitaj ostale linije
    Do While DATAfile.AtEndOfStream <> True
      CurrentDATAline = DATAfile.ReadLine

      TempString = CurrentDATAline

      For i = 1 To Len(TempString) / 8
        DataSetLen = DataSetLen + 1
        ReDim Preserve DataMatrix(2, 1 To DataSetLen)
        DataMatrix(0, DataSetLen) = dataXstart + dataXstep * (DataSetLen - 1)
        DataMatrix(1, DataSetLen) = Val(Mid$(TempString, (i - 1) * 8 + 1, 8))
        dataXcurrent = dataXcurrent + dataXstep

      Next i
    Loop
  End If

  ' F2 - Citanje podataka iz datoteke u matricu, ReDim matrice

  If DATAfileF10 = False Then
    DataSetLen = 0

    Do While DATAfile.AtEndOfStream <> True

      If DataSetLen > 0 Then CurrentDATAline = DATAfile.ReadLine

      DataSetLen = DataSetLen + 1
      CommaSignPos = InStr(1, CurrentDATAline, ",", vbTextCompare)
      If CommaSignPos = 0 Then CommaSignPos = InStr(1, CurrentDATAline, " ",
      vbTextCompare)
1 2
```

```
1 2     ReDim Preserve DataMatrix(2, 1 To DataSetLen)

        TempString = Mid$(CurrentDATAline, 1, CommaSignPos)
        DataMatrix(0, DataSetLen) = Val(TempString)
        TempString = Mid$(CurrentDATAline, CommaSignPos + 1, Len(CurrentDATAline) -
        CommaSignPos)
        DataMatrix(1, DataSetLen) = Val(TempString)

      Loop
    End If

    ' KRAJ CITANJA
    DATAfile.Close

End Sub
```

```
Sub mnuFileExit_Click()
    Unload Me
End Sub
```

```
Sub mnuFileExport_Click()

    If UBound(DataMatrix, 2) = 1 Then
        MsgBox "Dataset empty, nothing to save", , "No data"
    Else
        CommonDialog1.FileName = ""
        CommonDialog1.Flags = &H2
        CommonDialog1.ShowSave
        If CommonDialog1.FileName <> "" Then
            Debug.Print CommonDialog1.FileName
            ExportDATA CommonDialog1.FileName
        End If
    End If
End Sub
```

```
Sub mnuFileOpen_Click()
    CommonDialog1.ShowOpen
    If CommonDialog1.FileName <> "" Then
        LoadDATA CommonDialog1.FileName
        StartedDrawing = True
        RefreshShowHideButtons
        DrawGrafik
    End If
End Sub
```

```
Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal
End Sub
```

```
Sub mnuMeasCont_Click()
    dlgMeasCont.Show vbModal
    If (gcmsSpeed <> -1) Then gcomMeasCont gcmsStart, gcmsEnd, gcmsSpeed

End Sub
```

```
Sub mnuMeasStep_Click()
    dlgMeasStep.Show vbModal
    If (gcmsTime <> -1) Then gcomMeasStep gcmsStart, gcmsEnd, gcmsStep, gcmsTime

End Sub
```

```
Sub mnuMeasStop_Click()

    mnuMeasStop.Enabled = False
    Toolbar1.Buttons.Item(11).Enabled = False

    Do
        Ret = DoEvents()
        MSComm1.Output = Chr$(3)
    Loop Until InStr(BigBuffer, ">")

    'MSComm1.Output = "NEW" + Chr$(13) + Chr$(10)
    'gcomWaitFor ">"
    'MSComm1.Output = "BYE" + Chr$(13) + Chr$(10)
    'gcomWaitFor ">"

End Sub
```

```
Sub mnuToolsClosePort_Click()
  CloseCOMPort
End Sub
```

```
Sub mnuToolsDriveTheta_Click()
  dlgDriveTheta.Show vbModal

  If DriveThetaTo > 0 Then gcomDriveMotors DriveThetaTo
End Sub
```

```
Sub mnuToolsOpenPort_Click()
  OpenCOMPort
End Sub
```

```
Sub mnuToolsOptions_Click()
  frmSettings.Show vbModal
  RefreshStatusBar
  RefreshShowHideButtons
  If StartedDrawing Then DrawGrafik
End Sub
```

```
Sub mnuToolsReadAngles_Click()
  BigBuffer = ""
  gcomReadAngles
  gcomWaitFor (Chr$(13))
  InterpretAngles BigBuffer
  BigBuffer = ""
End Sub
```

```
Sub mnuToolsReset_Click()
  BigBuffer = ""
  gcomControllerReset
  BigBuffer = ""
End Sub
```

```
Private Sub OpenCOMPort()
  On Error Resume Next
  Dim PortToOpen As String
  Dim PortSettings As String
  ' Format line as COM2:1200,N,8,1

  PortSettings = Trim$(Str(PortSpeed)) + "," + Trim$(PortParity) + "," _
  + Trim$(Str(DataBits)) + "," + Trim$(Str(StopBits))

  'Debug.Print "PortSettings = " + PortSettings
  MSComm1.CommPort = COMPort
  MSComm1.Settings = PortSettings
  MSComm1.PortOpen = True

  If Err Then MsgBox Error$, 48
  If MSComm1.PortOpen = True Then EnablePortDependentButtons

  RefreshStatusBar
End Sub
```

```
Sub OpenINIFile()
  Const ForReading = 1, ForWriting = 2, ForAppending = 3
  Const Tristatefalse = 0
  Dim fs, INIFile
  Dim CurrentINILine As String

  Set fs = CreateObject("Scripting.FileSystemObject")
  On Error GoTo FileError
  Set INIFile = fs.OpenTextFile(INIFileName, ForReading, Tristatefalse)

 ┌Do While INIFile.AtEndOfStream <> True
  │  CurrentINILine = INIFile.ReadLine
  │
  │ ┌Select Case Left$(CurrentINILine, 4)
  │ ├──Case "Port"
  │ │    AssignCOMPort (CurrentINILine)
  │ ├──Case "Spee"
  │ │    AssignPortSpeed (CurrentINILine)
  │ ├──Case "Data"
  │ │    AssignDataBits (CurrentINILine)
  │ ├──Case "Pari"
  │ │    AssignPortParity (CurrentINILine)
  │ ├──Case "Stop"
  │ │    AssignStopBits (CurrentINILine)
  │ ├──Case "Open"
  │ │    AssignOpenPortAtStartup (CurrentINILine)
  │ ├──Case "GVLI"
  │ │    AssignGVLI (CurrentINILine)
  │ ├──Case "GCLI"
  │ │    AssignGCLI (CurrentINILine)
  │ ├──Case "GULT"
  │ │    AssignGULT (CurrentINILine)
  │ ├──Case "GULO"
  │ │    AssignGULO (CurrentINILine)
  │ ├──Case "GLLT"
  │ │    AssignGLLT (CurrentINILine)
  │ ├──Case "GLLO"
  │ │    AssignGLLO (CurrentINILine)
  │ ├──Case "GRAT"
  │ │    AssignGRAT (CurrentINILine)
  │ ├──Case "GRAO"
  │ │    AssignGRAO (CurrentINILine)
  │ ├──Case "GSAT"
  │ │    AssignGSAT (CurrentINILine)
  │ ├──Case "GSAO"
  │ │    AssignGSAO (CurrentINILine)
  │ ├──Case "GDTT"
  │ │    AssignGDTT (CurrentINILine)
  │ ├──Case "GSPD"
  │ │    AssignGSPD (CurrentINILine)
  │ ├──Case "gfBC"
  │ │    AssignGFKBackColor (CurrentINILine)
  │ ├──Case "gfAC"
  │ │    AssignGFKAxisColor (CurrentINILine)
  │ ├──Case "gfLC"
  │ │    AssignGFKLineColor (CurrentINILine)
  │ ├──Case "gfSO"
  │ │    AssignGFKSymbolOColor (CurrentINILine)
  │ ├──Case "gfSF"
  │ │    AssignGFKSymbolFColor (CurrentINILine)
  │ ├──Case "gfGC"
  │ │    AssignGFKGridColor (CurrentINILine)
  │ ├──Case "gfDG"
  │ │    AssignGFKDrawGrid (CurrentINILine)
  │ ├──Case "gfDS"
  │ │    AssignGFKDrawSymbols (CurrentINILine)
  │ └End Select
 └Loop

  INIFile.Close
  Exit Sub

FileError:
  MsgBox ("INI file not found." + Chr$(13) + "Creating a new file with default values.")
  CreateINIFile
  OpenINIFile
  RefreshStatusBar
```

```
End Sub
```

```
Private Sub PortSetup_Click()
  frmSettings.Show vbModal
  RefreshStatusBar
End Sub
```

```
Private Sub RefreshShowHideButtons()
  If gfkDrawSymbols = True Then
    cmdSymbols.Caption = "Hide symbols"
  Else
    cmdSymbols.Caption = "Show symbols"
    Grafik.ToolTipText = ""
  End If
  If gfkDrawGrid = True Then cmdGrid.Caption = "Hide grid" Else cmdGrid.Caption = "Show
  grid"
  If StartedDrawing = True Then
    cmdGrid.Enabled = True
    cmdSymbols.Enabled = True
  Else
    cmdGrid.Enabled = False
    cmdSymbols.Enabled = False
  End If
End Sub
```

```
Sub RefreshStatusBar()
  Dim PortToOpen As String
  ' Format line as COM2:1200,N,8,1
  PortToOpen = "COM" + Trim$(Str(COMPort)) + ":" + Trim$(Str(PortSpeed)) + "," _
   + Trim$(PortParity) + "," + Trim$(Str(DataBits)) + "," + Trim$(Str(StopBits))

  StatusBar1.Panels(1).Text = PortToOpen
  If MSComm1.PortOpen Then StatusBar1.Panels(2).Text = "Open" Else StatusBar1.Panels(2).
  Text = "Closed"
  Select Case Context
    Case Is = 0
      StatusBar1.Panels(3) = "Idle"
    Case Is = 1
      StatusBar1.Panels(3) = "Pre-initialization"
    Case Is = 2
      StatusBar1.Panels(3) = "Initialization"
    Case Is = 3
      StatusBar1.Panels(3) = "Read angles"
    Case Is = 4
      StatusBar1.Panels(3) = "Drive theta"
    Case Is = 5
      StatusBar1.Panels(3) = "Drive omega"
    Case Is = 6
      StatusBar1.Panels(3) = "Drive theta-omega"
    Case Is = 7
      StatusBar1.Panels(3) = "Measurement"
  End Select

  StatusBar1.Panels(4) = "LLT=" + Format$(GLLT) + "°   " + "ULT=" + Format$(GULT) + "°
     " + "LLO=" + Format$(GLLO) + "°   " + "ULO=" + Format$(GULO) + "° "

End Sub
```

```
Private Sub RefreshThetaOmega()

  lblTheta = "2è = " + Format$(gcmsCurrentAngle) + " °"
  lblOmega = "è = " + Format$(gcmsCurrentAngle / 2) + " °"
  If Context = Measurement Then lblCount = "C = " + Format$(gcmsCount)

End Sub
```

```
Sub SaveINIFile()

  Dim fs, INIFile

  Set fs = CreateObject("Scripting.FileSystemObject")
  On Error GoTo FileError
  Set INIFile = fs.CreateTextFile(INIFileName, True)

  INIFile.WriteLine ("Port=" + Trim$(Str(COMPort)))
  INIFile.WriteLine ("Speed=" + Trim$(Str(PortSpeed)))
  INIFile.WriteLine ("DataBits=" + Trim$(Str(DataBits)))
  INIFile.WriteLine ("Parity=" + Trim$(PortParity))
  INIFile.WriteLine ("StopBits=" + Trim$(Str(StopBits)))
  If OpenPortAtStartup = True Then INIFile.WriteLine ("OpenPortAtStartup=1") Else
  INIFile.WriteLine ("OpenPortAtStartup=0")
  INIFile.WriteLine ("GVLI=" + Trim$(Str(GVLI)))
  INIFile.WriteLine ("GCLI=" + Trim$(Str(GCLI)))
  INIFile.WriteLine ("GULT=" + Trim$(Str(GULT)))
  INIFile.WriteLine ("GULO=" + Trim$(Str(GULO)))
  INIFile.WriteLine ("GLLT=" + Trim$(Str(GLLT)))
  INIFile.WriteLine ("GLLO=" + Trim$(Str(GLLO)))
  INIFile.WriteLine ("GRAT=" + Trim$(Str(GRAT)))
  INIFile.WriteLine ("GRAO=" + Trim$(Str(GRAO)))
  INIFile.WriteLine ("GSAT=" + Trim$(Str(GSAT)))
  INIFile.WriteLine ("GSAO=" + Trim$(Str(GSAO)))
  INIFile.WriteLine ("GDTT=" + Trim$(Str(GDTT)))
  INIFile.WriteLine ("GSPD=" + Trim$(Str(GSPD)))
  INIFile.WriteLine ("gfBC=" + Trim$(Str(gfkBackColor)))
  INIFile.WriteLine ("gfAC=" + Trim$(Str(gfkAxisColor)))
  INIFile.WriteLine ("gfLC=" + Trim$(Str(gfkLineColor)))
  INIFile.WriteLine ("gfSO=" + Trim$(Str(gfkSymbolOColor)))
  INIFile.WriteLine ("gfSF=" + Trim$(Str(gfkSymbolFColor)))
  INIFile.WriteLine ("gfGC=" + Trim$(Str(gfkGridColor)))
  If gfkDrawSymbols = True Then INIFile.WriteLine ("gfDS=1") Else INIFile.WriteLine (
  "gfDS=0")
  If gfkDrawGrid = True Then INIFile.WriteLine ("gfDG=1") Else INIFile.WriteLine (
  "gfDG=0")
  INIFile.Close
  Exit Sub

FileError:
  MsgBox ("File error!")

End Sub
```

```
Private Sub SaveMeasurement()

  MEASFile.WriteLine (Format$(gcmsCurrentAngle) + " " + Format$(gcmsCount))

End Sub
' The OnComm event is used for trapping communications events and errors.
Private Static Sub MSComm1_OnComm()
Dim EVMsg$
Dim ERMsg$

' Branch according to the CommEvent property.
Select Case MSComm1.CommEvent
    ' Event messages.
Case comEvReceive
    Dim TempBuffer As String
    Dim Buffer As String

    TempBuffer = MSComm1.Input
    Buffer = StrConv(TempBuffer, vbUnicode)
    BigBuffer = BigBuffer + Buffer
    ShowData TerminalWindow, Buffer
    ' Debug.Print Buffer

    If InStr(BigBuffer, "INITIALIZATION? (Y/N) ") Then 'And (dlgInitAttempt.Visible =
    False) Then
      If dlgInitAttempt.Visible = False Then
        BigBuffer = ""
        dlgInitAttempt.Show vbModal
        If DoInit Then gcomInitialize
      End If
    End If

    If (InStr(BigBuffer, "ResOK" + Chr$(13)) And (Context = Measurement)) Then

      BigBuffer = Left$(BigBuffer, InStr(BigBuffer, "ResOK") + 5)
      InterpretMeasurement BigBuffer
      'Debug.Print BigBuffer
      Buffer = ""
      BigBuffer = ""
      RefreshThetaOmega
      GSAT = gcmsCurrentAngle
      GSAO = gcmsCurrentAngle / 2
      SaveINIFile
      AddMeasToGraph
      SaveMeasurement
    End If

    ' Case comEvSend

Case comEvEOF
    EVMsg$ = "End of File Detected"

    ' Error messages.
Case comBreak
    ERMsg$ = "Break Received"
Case comFrame
    ERMsg$ = "Framing Error"
Case comOverrun
    ERMsg$ = "Overrun Error"
Case comRxOver
    ERMsg$ = "Receive Buffer Overflow"
Case comRxParity
    ERMsg$ = "Parity Error"
Case comTxFull
    ERMsg$ = "Transmit Buffer Full"
Case Else
    ERMsg$ = "Unknown error or event"
End Select

End Sub
' This procedure adds data to the Term control's Text property.
' It also filters control characters, such as BACKSPACE,
' carriage return, and line feeds, and writes data to
' an open log file.
' BACKSPACE characters delete the character to the left,
' either in the Text property, or the passed string.
' Line feed characters are appended to all carriage
```

```
' returns.  The size of the Term control's Text
' property is also monitored so that it never
' exceeds MAXTERMSIZE characters.
Private Static Sub ShowData(Term As Control, Data As String)
On Error GoTo Handler
Const MAXTERMSIZE = 16000
Dim TermSize As Long, i

' Make sure the existing text doesn't get too large.
TermSize = Len(Term.Text)
If TermSize > MAXTERMSIZE Then
  Term.Text = Mid$(Term.Text, 4097)
  TermSize = Len(Term.Text)
End If

' Point to the end of Term's data.
Term.SelStart = TermSize

' Filter/handle BACKSPACE characters.
Do
  i = InStr(Data, Chr$(8))
  If i Then
    If i = 1 Then
      Term.SelStart = TermSize - 1
      Term.SelLength = 1
      Data = Mid$(Data, i + 1)
    Else
      Data = Left$(Data, i - 2) & Mid$(Data, i + 1)
    End If
  End If
Loop While i

' Eliminate line feeds.
Do
  i = InStr(Data, Chr$(10))
  If i Then
    Data = Left$(Data, i - 1) & Mid$(Data, i + 1)
  End If
Loop While i

' Make sure all carriage returns have a line feed.
i = 1
Do
  i = InStr(i, Data, Chr$(13))
  If i Then
    Data = Left$(Data, i) & Chr$(10) & Mid$(Data, i + 1)
    i = i + 1
  End If
Loop While i

' Add the filtered data to the SelText property.
Term.SelText = Data

' Log data to file if requested.
Term.SelStart = Len(Term.Text)
Exit Sub

Handler:
MsgBox Error$
Resume Next
End Sub
```

```
Private Sub TerminalWindow_Change()
  TerminalWindow.SelStart = Len(TerminalWindow.Text)
  TerminalWindow.SelLength = 0
End Sub
```

```
Private Sub Timer1_Timer()
  gcmsEstTime = gcmsEstTime - 1
  If gcmsEstTime > 0 Then
    lblEstTime = "Est. time remaining: " + Format$(CTime(gcmsEstTime), "HH:MM:SS")
  Else
    lblEstTime = "Est. time remaining: almost done"
  End If

End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As Button)
  ' Use the Key property with the SelectCase statement to specify
  ' an action.
 ┌Select Case Button.Key
 ├──Case Is = "FileOpen"
 │    mnuFileOpen_Click
 ├──Case Is = "PortOpen"
 │    OpenCOMPort
 ├──Case Is = "PortClose"
 │    CloseCOMPort
 ├──Case Is = "MeasStep"
 │    mnuMeasStep_Click
 ├──Case Is = "MeasCont"
 │    mnuMeasCont_Click
 ├──Case Is = "MeasStop"
 │    mnuMeasStop_Click
 ├──Case Is = "tbrReadAngles"
 │    mnuToolsReadAngles_Click
 ├──Case Is = "DriveT"
 │    mnuToolsDriveTheta_Click
 ├──Case Is = "Options"
 │    mnuToolsOptions_Click
 └End Select
End Sub
```

```
Public Sub UnloadAllForms()

  Dim Form As Form
 ┌For Each Form In Forms
 │  Unload Form
 │  Set Form = Nothing
 └Next Form
End Sub
```

```
'<------------------------ Description

VB.Form frmSettings
  +>MSComDlg.CommonDialog CommonDialogColor
(3)>VB.Frame FrameCont
    +>VB.CommandButton btnBackColor
    +>VB.CheckBox chkDrawSymbols
    +>VB.CheckBox chkDrawGrid
    +>VB.CommandButton btnGridColor
    +>VB.CommandButton btnSymbolFColor
    +>VB.CommandButton btnSymbolOColor
    +>VB.CommandButton btnLineColor
    +>VB.CommandButton btnAxisColor
    +>VB.CommandButton btnDefaultG
    +>VB.Label Label21
    +>VB.Label Label20
    +>VB.Label Label19
    +>VB.Label Label18
    +>VB.Label Label17
    +>VB.Label Label16
(2)>VB.Frame FrameCont
    +>VB.TextBox txtSPD
    +>VB.TextBox txtVLI
    +>VB.TextBox txtDTT
    +>VB.TextBox txtSAO
    +>VB.TextBox txtSAT
    +>VB.TextBox txtRAO
    +>VB.TextBox txtRAT
    +>VB.TextBox txtULO
    +>VB.TextBox txtLLO
    +>VB.TextBox txtULT
    +>VB.TextBox txtLLT
    +>VB.TextBox txtCLI
    +>VB.Label Label15
    +>VB.Label Label14
    +>VB.Label Label13
    +>VB.Label Label12
    +>VB.Label Label11
    +>VB.Label Label10
    +>VB.Label Label9
    +>VB.Label Label8
    +>VB.Label Label7
    +>VB.Label Label6
    +>VB.Label Label5
    +>VB.Label Label4
  +>VB.CommandButton btnCancel
  +>VB.CommandButton btnOK
(1)>VB.Frame FrameCont
    +>VB.Frame Frame4
      +>VB.CheckBox chkOpenPortAtStartup
    +>VB.CommandButton btnDefault
(0) +>VB.Frame Frame1
      +>VB.ComboBox cboSpeed
    +>VB.Frame Frame2
      +>VB.ComboBox cboDataBits
      +>VB.ComboBox cboParity
      +>VB.ComboBox cboStopBits
      +>VB.Label Label2
      +>VB.Label Label3
      +>VB.Label Label1
    +>VB.Frame Frame3
      +>VB.ComboBox cboPort
    +>VB.Label Label22
  +>MSComctlLib.TabStrip TabStrip1

'<------------------------ End Of Description
Attribute VB_Name = "frmSettings"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Private mintCurFrame As Integer ' Current Frame visible

Public COMPort, PortSpeed, DataBits, StopBits As Integer
Public PortParity As String
```

```
Private Sub btnAxisColor_Click()
  CommonDialogColor.Color = btnAxisColor.BackColor
  CommonDialogColor.ShowColor
  btnAxisColor.BackColor = CommonDialogColor.Color
End Sub
```

```
Private Sub btnBackColor_Click()
  CommonDialogColor.Color = btnBackColor.BackColor
  CommonDialogColor.ShowColor
  btnBackColor.BackColor = CommonDialogColor.Color
End Sub
```

```
Private Sub btnCancel_Click()
  Unload Me
End Sub
```

```
Private Sub btnDefault_Click()
  cboPort.ListIndex = 0
  cboSpeed.Text = "9600"
  cboDataBits.Text = "8"
  cboStopBits.Text = "1"
  cboParity.Text = "None"
  chkOpenPortAtStartup.Value = 0
End Sub
```

```
Private Sub btnGridColor_Click()
  CommonDialogColor.Color = btnGridColor.BackColor
  CommonDialogColor.ShowColor
  btnGridColor.BackColor = CommonDialogColor.Color
End Sub
```

```
Private Sub btnLineColor_Click()
  CommonDialogColor.Color = btnLineColor.BackColor
  CommonDialogColor.ShowColor
  btnLineColor.BackColor = CommonDialogColor.Color
End Sub
```

```
Private Sub btnOK_Click()
  frmGlavna.COMPort = Val(Right(cboPort.Text, 1))
  frmGlavna.PortSpeed = Val(cboSpeed.Text)
  frmGlavna.DataBits = Val(cboDataBits.Text)
  frmGlavna.StopBits = Val(cboStopBits.Text)
  Select Case cboParity.Text
    Case "Even"
      frmGlavna.PortParity = "E"
    Case "Odd"
      frmGlavna.PortParity = "O"
    Case "None"
      frmGlavna.PortParity = "N"
    Case "Mark"
      frmGlavna.PortParity = "M"
    Case "Space"
      frmGlavna.PortParity = "S"
  End Select

  frmGlavna.GVLI = Val(txtVLI.Text)
  frmGlavna.GCLI = Val(txtCLI.Text)
  frmGlavna.GULT = Val(txtULT.Text)
  frmGlavna.GULO = Val(txtULO.Text)
  frmGlavna.GLLT = Val(txtLLT.Text)
  frmGlavna.GLLO = Val(txtLLO.Text)
  frmGlavna.GRAT = Val(txtRAT.Text)
  frmGlavna.GRAO = Val(txtRAO.Text)
  frmGlavna.GSAT = Val(txtSAT.Text)
  frmGlavna.GSAO = Val(txtSAO.Text)
  frmGlavna.GDTT = Val(txtDTT.Text)
  frmGlavna.GSPD = Val(txtSPD.Text)

  frmGlavna.gfkBackColor = btnBackColor.BackColor
  frmGlavna.gfkAxisColor = btnAxisColor.BackColor
  frmGlavna.gfkLineColor = btnLineColor.BackColor
  frmGlavna.gfkGridColor = btnGridColor.BackColor
  frmGlavna.gfkSymbolOColor = btnSymbolOColor.BackColor
  frmGlavna.gfkSymbolFColor = btnSymbolFColor.BackColor
  If chkDrawGrid.Value = 1 Then frmGlavna.gfkDrawGrid = True Else frmGlavna.gfkDrawGrid
  = False
  If chkDrawSymbols.Value = 1 Then frmGlavna.gfkDrawSymbols = True Else frmGlavna.
  gfkDrawSymbols = False
  If chkOpenPortAtStartup.Value = 1 Then frmGlavna.OpenPortAtStartup = True Else
  frmGlavna.OpenPortAtStartup = False
  frmGlavna.SaveINIFile

  Unload Me

End Sub
```

```
Private Sub btnSymbolFColor_Click()
  CommonDialogColor.Color = btnSymbolFColor.BackColor
  CommonDialogColor.ShowColor
  btnSymbolFColor.BackColor = CommonDialogColor.Color
End Sub
```

```
Private Sub btnSymbolOColor_Click()
  CommonDialogColor.Color = btnSymbolOColor.BackColor
  CommonDialogColor.ShowColor
  btnSymbolOColor.BackColor = CommonDialogColor.Color
End Sub
```

```
Private Sub Form_Load()

  frmSettings.Width = 6645
  mintCurFrame = 1

  If frmGlavna.MSComm1.PortOpen = True Then
    Frame1(0).Visible = False
    Frame2.Visible = False
    Frame3.Visible = False
    Frame4.Visible = False
    btnDefault.Visible = False
    Label22.Visible = True
  End If

  FrameCont(1).Top = 600
  FrameCont(1).Left = 240
  FrameCont(1).BorderStyle = 0

  FrameCont(2).Top = 600
  FrameCont(2).Left = 240
  FrameCont(2).BorderStyle = 0

  FrameCont(3).Top = 600
  FrameCont(3).Left = 240
  FrameCont(3).BorderStyle = 0

  frmSettings.Height = 6030

  LoadPropertySettings

  cboPort.ListIndex = frmGlavna.COMPort - 1
  cboSpeed.Text = frmGlavna.PortSpeed
  cboDataBits.Text = frmGlavna.DataBits
  cboStopBits.Text = frmGlavna.StopBits

  Select Case Left(frmGlavna.PortParity, 1)
    Case "E"
      cboParity.ListIndex = 0
    Case "O"
      cboParity.ListIndex = 1
    Case "N"
      cboParity.ListIndex = 2
    Case "M"
      cboParity.ListIndex = 3
    Case "S"
      cboParity.ListIndex = 4
  End Select

  txtVLI.Text = Format$(frmGlavna.GVLI)
  txtCLI.Text = Format$(frmGlavna.GCLI)
  txtULT.Text = Format$(frmGlavna.GULT)
  txtULO.Text = Format$(frmGlavna.GULO)
  txtLLT.Text = Format$(frmGlavna.GLLT)
  txtLLO.Text = Format$(frmGlavna.GLLO)
  txtRAT.Text = Format$(frmGlavna.GRAT)
  txtRAO.Text = Format$(frmGlavna.GRAO)
  txtSAT.Text = Format$(frmGlavna.GSAT)
  txtSAO.Text = Format$(frmGlavna.GSAO)
  txtDTT.Text = Format$(frmGlavna.GDTT)
  txtSPD.Text = Format$(frmGlavna.GSPD)

  btnBackColor.BackColor = frmGlavna.gfkBackColor
  btnAxisColor.BackColor = frmGlavna.gfkAxisColor
  btnLineColor.BackColor = frmGlavna.gfkLineColor
  btnGridColor.BackColor = frmGlavna.gfkGridColor
  btnSymbolOColor.BackColor = frmGlavna.gfkSymbolOColor
  btnSymbolFColor.BackColor = frmGlavna.gfkSymbolFColor
  If frmGlavna.gfkDrawGrid = True Then chkDrawGrid.Value = 1 Else chkDrawGrid.Value = 0
  If frmGlavna.gfkDrawSymbols = True Then chkDrawSymbols.Value = 1 Else chkDrawSymbols.
  Value = 0
  If frmGlavna.OpenPortAtStartup = True Then chkOpenPortAtStartup.Value = 1 Else
  chkOpenPortAtStartup.Value = 0

  CommonDialogColor.Flags = &H1
End Sub
```

```
Sub LoadPropertySettings()
  Dim i As Integer, Settings As String, Offset As Integer

  ' Load Port Settings
  For i = 1 To 4
    cboPort.AddItem "Com" & Trim$(Str$(i))
  Next i

  ' Load Speed Settings
  cboSpeed.AddItem "1200"
  cboSpeed.AddItem "2400"
  cboSpeed.AddItem "4800"
  cboSpeed.AddItem "9600"
  cboSpeed.AddItem "14400"
  cboSpeed.AddItem "19200"
  cboSpeed.AddItem "28800"
  cboSpeed.AddItem "38400"
  cboSpeed.AddItem "56000"
  cboSpeed.AddItem "57600"
  cboSpeed.AddItem "115200"

  ' Load Data Bit Settings
  cboDataBits.AddItem "4"
  cboDataBits.AddItem "5"
  cboDataBits.AddItem "6"
  cboDataBits.AddItem "7"
  cboDataBits.AddItem "8"

  ' Load Parity Settings
  cboParity.AddItem "Even"
  cboParity.AddItem "Odd"
  cboParity.AddItem "None"
  cboParity.AddItem "Mark"
  cboParity.AddItem "Space"

  ' Load Stop Bit Settings
  cboStopBits.AddItem "1"
  cboStopBits.AddItem "1.5"
  cboStopBits.AddItem "2"

End Sub
```

```
Private Sub TabStrip1_Click()
  If TabStrip1.SelectedItem.Index = mintCurFrame Then Exit Sub
  FrameCont(TabStrip1.SelectedItem.Index).Visible = True
  FrameCont(mintCurFrame).Visible = False
  mintCurFrame = TabStrip1.SelectedItem.Index
End Sub
```

```
Attribute VB_Name = "SelTextOnFocus"
```

```
Public Sub SelTxt()

  'Select the entire textbox on GotFocus
 If Screen.ActiveForm.ActiveControl.Tag = "txt" Then
  With Screen.ActiveForm.ActiveControl
    .SelStart = 0
    .SelLength = Len(.Text)
  End With
 End If
End Sub
```

```
Attribute VB_Name = "Various"
Option Explicit
```

```
Function CompressSpaces(ByVal txt As String) As _
    String
  Do While InStr(txt, "  ") > 0
    txt = Replace(txt, "  ", " ")
  Loop
  CompressSpaces = txt
End Function
```

```
Function nStrCount(sSource As String, sChar As String) As Integer
  Dim nPos As Integer
  Dim nCount As Integer

  nCount = 0
  nPos = InStr(sSource, sChar)
  While nPos
    nCount = nCount + 1
    nPos = InStr(nPos + 1, sSource, sChar)
  Wend
  nStrCount = nCount
End Function
```

Literatura

1. Harold P. Klug, Leroy E. Alexander, *X-Ray Diffraction Procedures for Polycrystalline and Amorphous Materials.*
   John Wiley & Sons, New York, 1954.

2. Ljiljana Karanović, *Primenjena kristalografija.*
   Univerzitet u Beogradu, 1996.

3. Ivan Janić, *Osnovi atomske fizike,* I *deo.*
   (skripta), Novi Sad, 1992.

4. Dragoslav M. Petrović, Svetlana R. Lukić, *Eksperimentalna fizika kondenzovane materije.*
   Univerzitet u Novom Sadu, 2000.

5. Burkhard Kainka, *Elektronik am PC - Visual Basic in der Praxis.*
   Elektor-Verlag, Aachen, 2001.

6. *SLAC Beam Line,* Summer 1995., Vol. 25, No. 2.

7. Christopher Evans *Kompjutorski izazov,.*
   Globus, Zagreb, 1983.

8. Craig Peacock, *Interfacing the Serial/RS232 Port.*
   `http://www.beyondlogic.com`

9. *RS-232*
   `http://www2.rad.com/networks/1995/rs232/rs232.htm`

10. *PLCS.net Communications History*
    `http://www.plcs.net/chapters/comhistory29.htm`

11. *Seifert µ-CONTROLLER Bedienungsanleitung und Beschreibung.*
    Rich. Seifert & Co, Ahrensburg, 1986.

12. *Seifert RAE I Beschreibung und Bedienungsanleitung.*
    Rich. Seifert & Co, Ahrensburg, 1986.

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

☐ Redni broj:
RBR

☐ Identifikacioni broj:
IBR

☐ Tip dokumentacije: *Monografska dokumentacija*
TD

☐ Tip zapisa: *Tekstualni štampani materijal*
TZ

☐ Vrsta rada: *Diplomski rad*
VR

☐ Autor: *Vladimir Jokić, br. dos. 541/95*
AU

☐ Mentor: *dr Srđan Rakić*
MN

☐ Naslov rada: *Programski paket za prikupljanje podataka i upravljanje difraktometrijskim sistemom za prah Seifert MZ IV*
NR

☐ Jezik publikacije: *srpski-latinica*
JP

☐ Jezik izvoda: *srpski*
JI

☐ Zemlja publikovanja: *Srbija i Crna Gora*
ZP

☐ Uže geografsko područje: *Vojvodina*
UGP

☐ Godina: *2004.*
GO

☐ Izdavač: *Autorski reprint*
IZ

☐ Mesto i adresa: *Prirodno-matematički fakultet, Trg Dositeja Obradovića 4, 21000 Novi Sad*
MA

☐ Fizički opis rada:
FO

☐ Naučna oblast: *Eksperimentalna fizika kondenzovane materije*
NO

☐ Naučna disciplina: *Rendgenostrukturna analiza—instrumentalna kristalografija*
ND

☐ Predmetna odrednica / Ključne reči: *Difrakcija na prahu, difraktometrijski sistem, akvizicija podataka*
PO
UDK:

☐ Čuva se:
ČU

☐ Važna napomena: *nema*
VN

☐ Izvod:
IZ

☐ Datum prihvatanja teme od strane NN veća:
DP

☐ Datum odbrane:
DO

☐ Članovi komisije: (*naučni stepen / ime i prezime / zvanje / fakultet*)

- mentor: *dr Srđan Rakić, docent, PMF u Novom Sadu*
- član komisije: *dr Božidar Vujičić, redovan profesor, PMF u Novom Sadu*
- član komisije: *dr Miodrag Krmar, vanredan profesor, PMF u Novom Sadu*

KO