

Elektrotehnički fakultet
Univerziteta u Beogradu

Programabilni generator talasnih oblika
~ *projekat iz predmeta 32-bitni mikrokontroleri* ~

Student:

Predrag Rudić 3312/2012

Predmetni profesor:

dr. Dragan Vasiljević

Beograd, Septembar 2013.

Sadržaj

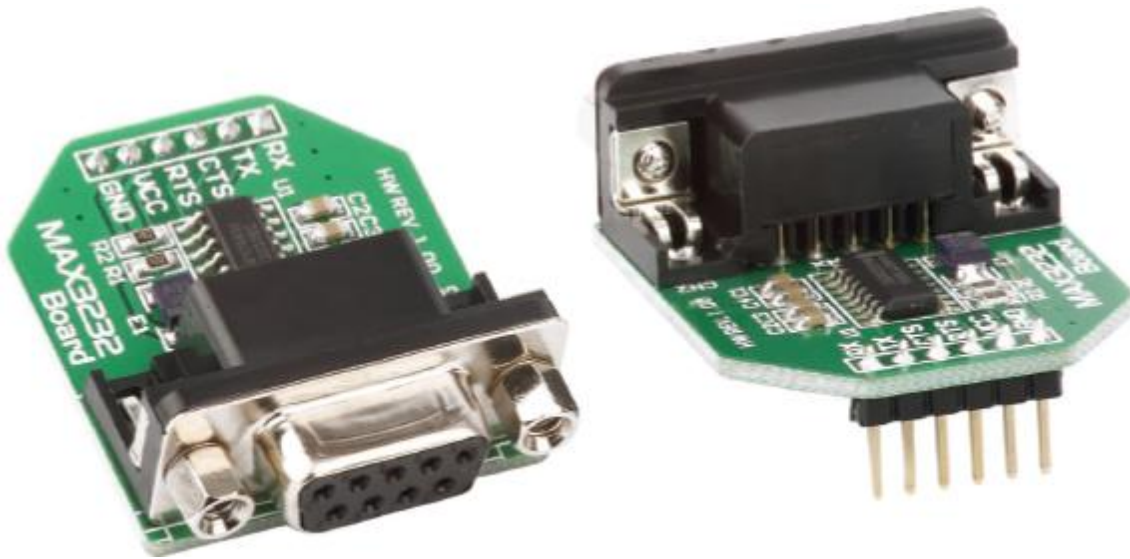
1. Uvod	3
2. Način pravljenja signala.....	4
3. Softverska realizacija.....	6
3.1 Inicijalizacija periferija	6
3.2 Glavna petlja.....	7
3.3 Prekidna rutina USART-a	9
3.4 Program na <i>PC</i> računaru	9
LITERATURA.....	10

1.Uvod

Generator signala takta je realizovan koristeći razvojnu pločicu STM32 *Value line discovery* na kojoj se nalazi mikrokontroler STM32F100RB. Ovaj mikrokontroler radi sa ARM procesorom *Cortex-M3*.

Projekat je zamišljen tako da se generatorom signala takta upravlja sa *PC* računara, a da se komunikacija između mikrokontrolera i računara sprovodi koristeći RS232 port na računaru i UART na mikrokontroleru. Na PC računaru se zadaje specifikacija signala koja se zatim prosleđuje mikrokontroleru.

Za povezivanje sa računarom korišćen je MAX3232 firme Mikroelektronika prikazan na slici 1. Povezivanje MAX3232 sa pločicom na kojoj je mikrokontroler je urađeno preko protoborda.



Slika 1. MAX3232 firme Mikroelektronika

U projektu je realizovan digitalni deo generatora signala tj. do D/A konvertora. Za generator signala koji bi mogao da se koristi, potrebno je dodati još analogni deo. Analogni deo počinje odmah iza D/A konvertora: NF filter kod koga se učestanost slabljenja od 6dB nalazi na polovini učestanosti odabiranja D/A konvertora. Iza NF filtra bi trebalo da ide pomerač DC nivoa koji će signal čiji je opseg $0 \div 3,3V$ pretvoriti u signal čiji opseg može da bude simetričan oko nule, npr: $-3,3V \div 3,3V$.

Generator signala podržava generisanje sinusoide, četvrtki i trougaonog signala. Zbog vremena smirivanja signala iz D/A konvertora koje može da bude do $4\mu s$, odlučeno je da učestanost odabiranja D/A konvertora bude 200kHz. Samim tim, najveća frekvencija sinusoide koja može da se generiše na izlazu D/A je 100kHz. Ali najveća frekvencija sinusoide koja može da se dobije na izlazu analognog dela generatora zavisi od selektivnosti analognog NF filtra. Kada bi ovaj filter bio idealan, najveća frekvencija na izlazu analognog dela bi bila 99,999kHz. Kod realnog filtra ova vrednost je manja i zavisi od kriterijuma najmanjeg slabljenja amplitude sinusoide, kao i od dovoljnog slabljenja u nepropusnom opsegu.

Ova realizacija generatora signala nije uzela u obzir modifikaciju spektra izlaznog signala koja potiče od samog D/A konvertora. Modifikacija amplitudske karakteristike se može predstaviti jednačinom: $\frac{\sin(\frac{\omega T_0}{2})}{\omega \frac{T_0}{2}}$, gde je $1/T_0$ učestanost odabiranja D/A konvertora, a ω je kružna učestanost. Kod sinusoida, ova modifikacija se može ispraviti jednostavnim skaliranjem amplitude. Kod signala četvrtke nije potrebna nikakva modifikacija spektra ovog signala, dok bi kod trouglića morao da se napravi FIR filter da bi se iskompenzovala ovakva modifikacija spektra izlaznog signala.

2. Način pravljenja signala

Da bi se postiglo da sinusoida može da ima učestanosti bilo kog multilpla od 1Hz mora se raditi decimacija signala. Zbog toga sledi kratko objašnjenje decimacije, a formule koje dokazuju ovo mogu se naći u literaturi [1].

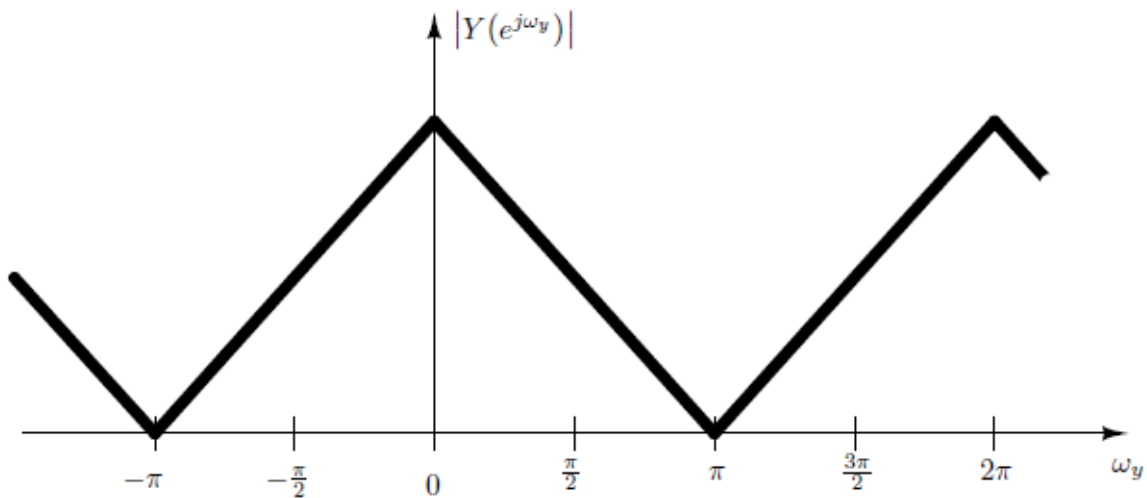
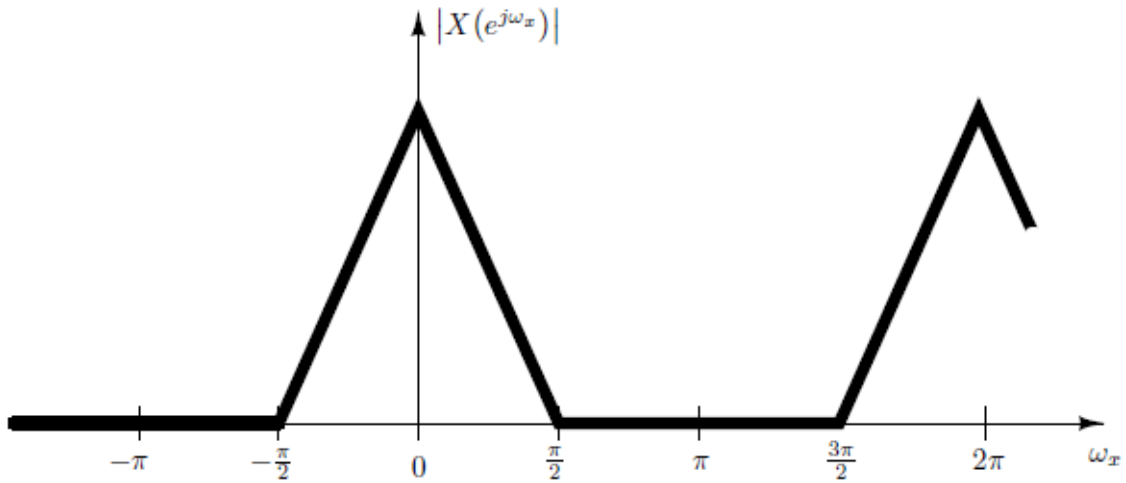
Decimacija predstavlja redukciju učestanosti odabiranja. Ovde će biti reči o redukciji učestanosti odabiranja celobrojnim faktorom D.

Neka je učestanost odabiranja signala $x[n]$, $F_x=1/T_x$ i neka je spektar ovog signala različit od nule u opsegu $0 \leq |F| < F_x/2$, odnosno $0 \leq \Omega < \pi$. Redukcija učestanosti odabiranja signala $x[n]$ može se izvršiti uzimanjem svakog D-tog odbirka signala, čime se dobija izlazni signal $y[n]$. Ali, pošto smanjenjem učestanosti odabiranja prestaje da bude zadovoljena teorema o odabiranju, u spektru izlaznog signala će se pojaviti preklapanje. Da bi se izbeglo preklapanje u spektru izlaznog signala, spektar ulaznog signala mora biti ograničen na opseg učestanosti $0 \leq F \leq F_x/(2D)$, odnosno $0 \leq \Omega \leq \pi/D$. Ovo može da se postigne dodavanjem NF filtra koji će ograničiti spetar ulaznog signala, tj funkcija prenosa takvog NF filtra mora da zadovolji uslov:

$$H(e^{j\Omega}) = \begin{cases} 1, & \Omega \leq \pi/D \\ 0, & \text{drugde} \end{cases}$$

Na slici 2 dat je primer decimacije signala $x[n]$ čiji je spektar ograničen do opsega učestanosti $\pi/2$. Ako se vrši decimacija ovog signala sa faktorom $D=2$, nije potrebno na ulaz dodavati NF filter. Ako je $D>2$, bez NF filtra će doći do preklapanja spektra.

S obzirom na to da je učestanost odabiranja D/A konvertora 200kHz, da bi se dobila sinusoida frekvencije 1Hz, potrebno je da takva sinusoida ima 200.000 odbiraka, signal $x[n]$. Spektar ovakve sinusoida će imati samo jednu komponentu na 1Hz u opsegu učestanosti $0 \leq \Omega < \pi$ i jednu komponentu na 199.999Hz u opsegu učestanosti $\pi < \Omega \leq 2\pi$. Sada, ako se želi sinusoida frekvencije N, gde je N ceo broj za koji važi $N \in \{1,2, \dots, 99999\}$, potrebno je samo da se uzme svaki N-ti odbirak sinusoida koja ima 200k odbiraka, tj. uradi se decimacija sinusoida od 1Hz (signala $x[n]$) sa faktorom decimacije $D=N$ da bi se dobila sinusoida veće učestanosti (signal $y[n]$). U frekvencijskom domenu, u spektru signala $y[n]$ neće doći do preklapanja, jer će se komponenta signala $x[n]$ koja se nalazi na učestanosti 1Hz, u signalu $y[n]$ prebaciti na učestanost od N Hz, odnosno neće preći polovinu učestanosti odabiranja (100kHz). Isto tako, komponenta spektra signala $x[n]$ na učestanosti 199.999Hz se neće spustiti ispod učestanosti 100kHz.



Slika 2. Spektar signala pre i posle decimacije sa faktorom $D=2$

Analogni NF filter na izlazu iz D/A konvertora treba što manje da oslabi komponentu iz opsega $0 \leq \Omega < \pi$, a što više da oslabi komponentu iz opsega $\pi < \Omega \leq 2\pi$. Što je frekvencija sinusoide veća, to je komponenta iz oblasti $\pi < \Omega \leq 2\pi$ bliža komponenti iz oblasti $0 \leq \Omega < \pi$, i samim tim je teže isfiltrirati takav signal.

Zbog jednostavnosti generisanja signala, pravljenje signala četvrtke i trouglića je rešeno na isti način, tj. decimacijom signala od 1Hz. Zbog toga što ovi signali imaju nenulte odbirke spektra na visokim učestanostima, ovim načinom neminovno dolazi do preklapanja spektra, što se manifestuje kao slučajan šum u spektralnom domenu.

U fajlovima **Uporedjivanje Spektra Trouglića.m** i **Cetvrtke Spektra.m** koji mogu da se izvrše u programskom paketu MATLAB, mogu se uporediti spektri signala koje proizvodi ovderealizovani generator signala i spektar signala koji ima idealne odbirke za traženu frekvenciju (i traženi *duty ratio* za četvrtke). Prvi fajl upoređuje signale za trougaoni signal a drugi za četvrtke.

3. Softverska realizacija

Dvesta hiljada odbiraka sinusoide frekvencije 1Hz se dobija iz fleš memorije. Korišćeni mikrokontroler poseduje 128kB fleš memorije, što je dovoljno da se smesti četvrtina 16-bitnih odbiraka ove sinusoide, koji ukupno zauzimaju 100.000 bajtova. Ostale tri četvrtine odbiraka sinusoide lako mogu da se dobiju iz ove prve četvrtine, zato što se u prvoj četvrtini periode sinusne funkcije nalaze sve apsolutne vrednosti sinusoide.

Odbirci sinusoide su upisani u mikrokontroler posebnim programom. *PC* računar izračunava odbirke sinusoide i šalje ih mikrokontroleru koji ih zatim upisuje u fleš memoriju. Pošto glavni program generatora signala ne zauzima više od 3kB memorije, odbirci sinusoide su stavljani u memoriju počevši od desete strane fleš memorije.

Ovaj način upisivanja u fleš memoriju je moguć jer sam mikrokontroler podržava upisivanje u fleš memoriju iz programa koji je napisao korisnik. Kada se nakon završenog upisa odbiraka, u fleš memoriju upiše nov program za izvršavanje, njegov upis neće obrisati odbirke sinusoide, osim ako veličina programa nije veća od 10kB, jer odbirci počinju na desetoj strani fleš memorije.

Odbirci za trougaoni signal i četvrtke nisu čuvani u memoriji jer oni mogu lako da se izračunaju u samom mikrokontroleru.

3.1 Inicijalizacija periferija

Periferije mikrokontrolera koje su korišćene prilikom realizacije projekta su: USART, tajmer 6 i D/A konvertor.

Na početku inicijalizacije potrebno je u RCC bloku mikrokontrolera omogućiti taktni signal svim korišćenim periferijama. Nakon toga se pinovi odgovarajućih portova nameštaju da budu u odgovarajućem modu. Pin D/A konvertora je namešten da bude u *analog mod*-u. Kod USART-a Tx pin i RTS pin su namešteni da budu izlazni pinovi alternativne funkcije.

USART se koristi kao UART i služi da omogući komunikaciju sa *PC* računarom. Namešteno je da se vrši hardverska kontrola protoka, a *baud rate* je 115200 b/s i namešten je neparni bit parnosti. Ovoliki *baud rate* je namešten jer je to najveći *baud rate* koji podržava drajver COM porta u operativnom sistemu *PC* računara. Omogućeno je da se uđe u prekidnu rutinu USART periferije kada dođe podatak u Rx bafer. Da bi se omogućila ova prekidna rutina, potrebno je u kontrolnom registru same periferije omogućiti ovaj prekidni zahtev, kao i u kontroleru prekida NVIC koji se nalazi u samom jezgru procesora.

D/A konvertor ovog mikrokontrolera je 12-bitni. On se koristi tako što se podatak za konverziju ubaci u DHR registar D/A konvertora. Posle toga, kada se javi određeni okidač, sadržaj tog registra se premešta u DOR registar nakon čega se vrši D/A konverzija na izlazni pin konvertora za najviše 4 μ s. Ovo vreme zavisi od spoljašnje kapacitivnosti i otpornosti izlaznog pina. Događaj koji vrši okidanje može da dođe iz određenih tajmera, sa jednog određenog spoljašnjeg pina ili od DMA. Okidač može da bude i sam softverski upis

u DHR registar. Konvertor sadrži i opcioni izlazni bafer kojim se povećava strujni kapacitet izlaznog pina. Ako se izabere izlazni bafer, tada nije potrebno spolja dodavati takav bafer radi povećanja strujnog kapaciteta, ali je tada opseg izlaznog analognog napona: $0,2V \div 3,1V$ [5]. U ovom projektu je namešteno da se D/A konvertor okida sa signalom TRGO iz tajmera 6.

Kao što je već rečeno TRGO izlaz tajmera 6 je direktno povezan na signal za okidanje D/A konvertora. *Prescaler* i *AutoReload* registri tajmera 6 su tako namešteni da brojač ovog tajmera odbroji od nule do vrednosti smeštene u *AutoReload* registru za $5\mu s$. Svaki put kada se to desi, kaže se da se desio *Update Event*. Tajmer je namešten tako da kada njegov brojač dobroji do vrednosti smeštene u *AutoReload* registru, brojač se automatski resetuje i počinje ponovo da odbrojava. To znači da se *UpdateEvent* događa sa učestanošću od 200kHz. Kada se dogodi *UpdateEvent*, ako je omogućeno u njegovom kontrolnom registru, javi se i događaj na njegovom TRGO izlazu. Dakle, da bi D/A konvertor dobio znak sa TRGO izlaza tajmera da počne konverziju, potrebno je u samom tajmeru, u njegovom kontrolnom registru, namestiti da on može da pravi događaje na njegovom TRGO izlazu, što je i urađeno u ovom projektu. Kada brojač tajmera dobroji do vrednosti u *AutoReload* registru (kada se dogodi *UpdateEvent*), UIF bit (*update interrupt flag*) u njegovom statusnom registru se postavlja na logičku jedinicu i ostaje u tom stanju sve dok ga softver ne namesti na logičku nulu.

3.2 Glavna petlja

Nakon inicijalizacije periferija, glavni program ulazi u beskonačnu petlju čija je struktura data na slici 3.

Komunikacija sa PC računarom se obavlja preko USART-a. Ona je u mikrokontroleru implementirana u prekidnoj rutini USART-a koja se izvršava svaki put kada stigne podatak u Rx bafer. Kada se u poruci iz PC računara naznači da se želi drugačiji oblik signala, tada prekidna rutina USART-a uz pomoć promenljive *novostanje* naznačuje petlji u glavnom programu da je došlo do promene promenljive *stanje*, ili da je samo došlo do promene parametara signala.

Ako je vrednost promenljive *stanje* jednako *POCETNO*, tada mikrokontroler ulazi u stanje manje potrošnje i ostaje u njemu sve dok se ne pojavi nova prekidna rutina USART-a. Posle toga, u zavisnosti od promenljive *stanje*, može opet da uđe u početno stanje ili da uđe u neko stanje u kome se izračunavaju odbirci signala koji se šalju na D/A konvertor.

Kada mikrokontroler uđe u neko stanje izračunavanja odbiraka, ulazi se u „beskonačnu“ petlju iz koje se izlazi samo kad prekidna rutina USART-a namesti vrednost promenljive *novostanje* na vrednost različitu od nule. U ovoj „beskonačnoj“ petlji prvo se proverí vrednost promenljive *novostanje* i ako je ona različita od nule, izlazi se iz trenutnog stanja i ulazi u novo stanje određeno promenljivom *stanje*. Ako je promenljiva *novostanje* i dalje nula, prelazi se na izračunavanje odbirka koji se zatim stavlja u registar DHR D/A konvertora. Ovaj deo koda nije naznačen na slici 3, već su stavljene tri tačke. Kada je softver stavio izračunati odbirak u DHR registar D/A konvertora, čeka se da tajmer 6 naznači da je poslao TRGO događaj D/A konvertoru tako

```

while(1)
{
    novoStanje=0;
    ...
    switch (stanje)
    {
        case POCKETNO:
            __WFI();
        case TROUGLICI:
            {
                while(1)
                {
                    if (novoStanje) break;
                    ...
                    TIM6->SR&=~TIM_SR_UIF;
                    while(!(TIM6->SR & TIM_SR_UIF));
                }
                break;
            }
        case CETVRTKE:
            ...
        case SINUSOIDA:
            ...
    }
}

```

Slika 3. Struktura beskonačne petlje glavnog programa

što se proverava vrednost bita UIF statusnog registra tajmera, koga tajmer treba hardverski da setuje kada se dogodi *UpdateEvent*. Ovaj bit mora softverski da se resetuje.

Iako je na slici 3 prikazana samo struktura „beskonačne“ petlje za stanje TROUGLICI, na isti način je realizovana „beskonačna“ petlja za stanja CETVRTKE i SINUSOIDA.

Interno, za predstavljanje vrednosti odbiraka se koristi komplement dvojke. Pošto je opseg izlaznog napona D/A konvertora nesimetričan u odnosu na nulu, tj od 0 do 3,3V, vrednost odbiraka mora da bude predstavljena u pomerenom binarnom kodu pre nego što se stavi u registar D/A konvertora. Kod predstavljanja brojeva u pomerenom binarnom kodu, najnegativnijem broju se dodeljuje vrednost 0x00...0, a najpozitivnijem 0x11...1. Kod ovog koda vrednost MSB je za negetivne brojeve 0, a za pozitivne brojeve 1, što je u suprotnosti sa predstavljanjem u komplementu dvojke, i zbog toga je jednostavno dobiti broj predstavljen u pomerenom kodu iz broja predstavljenog u komplementu dvojke (i obrnuto), na taj način što se jednostavno samo promeni vrednost bita najveće težine broja predstavljenog u komplementu dvojke.

D/A konvertor je 12-bitni i zato je potrebno prvo izvršiti kvantovanje odbiraka na 12 bita. Tom prilikom je vršeno zaokruživanje vrednosti broja na 12 bita kako bi greška zbog kvantovanja odbiraka bila manja.

3.3 Prekidna rutina USART-a

Ova prekidna rutina je zadužena za komunikaciju sa *PC* računarem. Ona se pokreće svaki put kada računar pošalje bajt mikrokontroleru.

Prvo se učitavaju dva bajta kontrolne poruke, a onda, u zavisnosti od kontrolne poruke, učitavaju se dodatni bajtovi podataka ili se izvršava naredba sadržana u kontrolnoj poruci. Posle učitavanja kontrolne poruke i svih podataka, *PC* računaru se šalje poruka o uspešno primljenoj kontrolnoj poruci. Ako bit parnosti nije dobar ili kontrolna poruka nije prepoznata, mikrokontroler šalje *PC* računaru jedan bajt koji označava da je došlo do greške u prenosu. Tada će računar još jednom poslati celu poruku. Ako sam računar pošalje bajt koji označava grešku, onda to znači da on nije prepoznao poruku o uspešno primljenoj kontrolnoj poruci pa se ona još jednom šalje. Program u *PC* računaru određuje koliko ovakvih grešaka zaredom može najviše da se dogodi pre nego što korisnika obavesti o lošoj komunikaciji sa mikrokontrolerom.

3.4 Program na *PC* računaru

Program na *PC* računaru je pavljen koristeći *Microsoft Visual Studio 2005*, na operativnom sistemu *Microsoft Windows XP*. On ne može da se izvršava na ranijim verzijama operativnog sistema *Windows* jer koristi funkcije koje nisu podržane u tim operativnim sistemima.

Napravljen je grafički korisnički interfejs dat na slici 4.



Slika 4. Grafički korisnički interfejs za generator signala

Pokretanjem programa se automatski otvara COM port, namešta *baud rate* na 115200 b/s, neparni bit parnosti i hardverska kontrola protoka.

Na vrhu prozora se nalazi *ComboBox* kojim se bira vrsta signala: sinusoida, četvrtke ili trouglići. U *edit* prozoru se upisuje frekvencija, jednosmerna vrednost i $V_{\text{peak-to-peak}}$. Kada se pritisne dugme “Konfiguriši generator” program čita sadržaje *edit* prozora i trenutnu vrednost *ComboBox*-a i na osnovu toga šalje kontrolnu poruku i podatke mikrokontroleru. Ako mikrokontroler ne odovori odgovarajućom porukom na poslatu poruku u roku od 100ms, korisnik će biti obavešten o tome.

Sa desne strane postoje dva *CheckBox*-a. Ako se klikne na neki od njih, program će istog trenutka poslati poruku mikrokontroleru u skladu sa stanjem pritisnutog *CheckBox*-a.

Napravljena je jedna programska nit koja služi samo zato da čita COM port, a druga nit postoji da opslužuje grafički interfejs.

LITERATURA

- [1] Miodrag Popović „Digitalna obrada signala“ ,pogl. 11 „Promena učestanosti odabiranja“
- [2] Miodrag Popović „Digitalna obrada signala“, pogl. 16 „Uticaj konačne dužine digitalne reči na karakteristike sistema“
- [3] The Definitive Guide to the Cortex-M3, 2nd edition, Joseph Yiu, 2010.
- [4] STMicroelectronics (2011) STM32F100xx *Reference manual* Rev. 4
- [5] STMicroelectronics (2010) STM32F100xx *Datasheet* Rev. 5
- [6] STMicroelectronics (2010) STM32F100xx *Programming manual* Rev. 2
- [7] Mikroelektronika, MAX3232 manual v100