

UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET

Lazar Caković, 3083/2016

Poredjenje vremena filtriranja na STM32F4
mikrokontroleru i PC-u

projekat iz predmeta 32bitni mikrokontroleri

mentor:
prof. dr Nenad Jovičić

Beograd, jul 2018.

Sadržaj

1	Uvod	3
2	Opis sistema	4
3	Softverska implementacija	5
3.1	MATLAB implementacija	5
3.2	Implementacija na mikrokontroleru	6
4	Poredjenje rezultata	9

Sl like

3.1	MATLAB: Karakteristika filtra	6
3.2	MATLAB: Vremena izvršavanja filtriranja u sekundama	7

Glava 1

Uvod

Cilj projekta iz predmeta 32-bitni mikrokontroleri, na master studijama odseka za Elektroniku, bio je da se uporedi filtriranje signala u vremenskom i frekvencijskom domenu na mikrokontroleru STM32F407VG i PC-u korišćenjem MATLAB programskog paketa. Poredjenje filtriranja se vršilo na brzinu izvršavanja filtriranja, kao i na odstupanje signala dobijenog filtriranjem na mikrokontroleru od onog dobijenog u programskom paketu MATLAB. Projekat je radjen u softverskim paketima MATLAB i IAR Embedded Workbench for ARM, a na ploči STM32F4 Discovery.

Glava 2

Opis sistema

U implementaciji projekta, sistem je konfigurisan tako da se deo projekta izvršava na PC-u, u okviru MATLAB programskog paketa. Dok se drugi deo izvršava na samoj razvojnoj ploči. Na PC-u je izgenerisan signal koji je potrebno obraditi. Razvojna ploča Discovery je povezana na PC, i korišćenjem *semihosting-a* implementirana je emulacija *file system-a* tako da je omogućeno čitanje i upis u fajlove PC-a.

Glava 3

Softverska implementacija

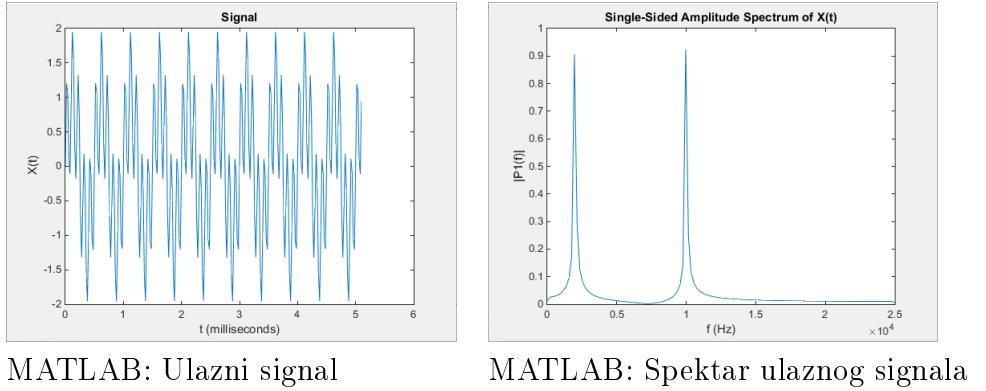
Kao što je već navedeno, softverska implementacija je izvršena na PC-u, u okviru MATLAB programskog paketa, i na strani mikrokontrolera u okviru IAR razvojnog okruženja. Pa će tim redom i biti opisane.

3.1 MATLAB implementacija

U ovom delu softverske implementacije, izgenerisan je signal kao zbir dve sinusoide različitih frekvencija koji je potrebno izfiltrirati. Konkretno, kao primer, uzet je zbir dve sinusoide od 2KHz i 10KHz, sa frekvencijom odabiranja od 50KHz, u 256 tačaka. Ovaj broj tačaka je uzet zbog ograničenja razvojne ploče koja se koristi u drugom delu implementacije, pošto se ovde izgenerisan signal koristi u daljoj implementaciji. Takođe, u ovom delu se generišu i odbirci filtra koji će biti korišćen u filtriranju kako u ovom delu, tako i u drugom delu implementacije. Filter koji se koristi je filter propusnik niskih učestanosti, 120og reda, sa graničnom učestanosti od 8kHz, i slabljenjem u propusnom opsegu od 0.01dB i u nepropusnom opsegu od 80dB. Za dobijanje odbiraka filtra koristi se ugradjena funkcija u MATLAB-u **fircegrip**.

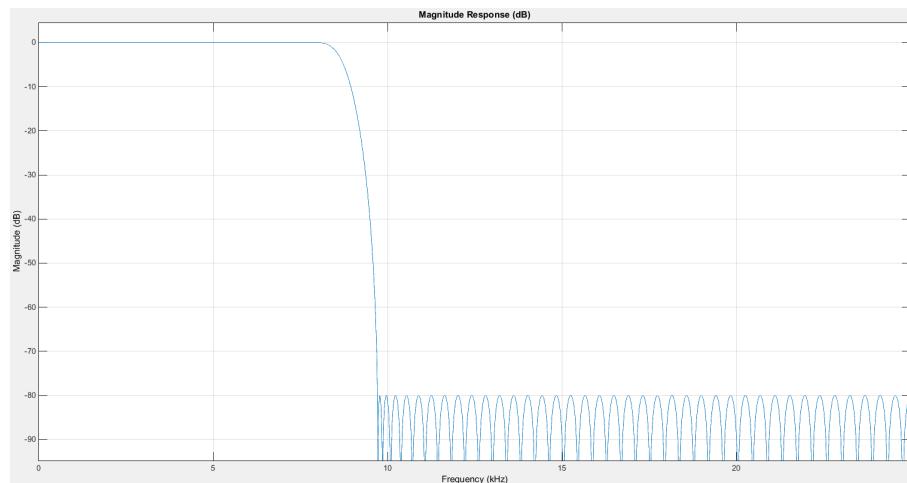
Filtriranje signala u MATLAB-u se vrši u vremenskom i frekvencijskom domenu. Filtriranje signala u frekvencijskom domenu se vrši korišćenjem ugradjene funkcije *filter*, uz pomoć koje se dobija isfiltriran signal kao na slici ispod. Filtriranje u frekvencijskom domenu vrši se množenjem FFT predstave signala i filtra. Nakon čega se dobija isti rezultat kao u prethodnoj implementaciji.

Kao što je navedeno, odbirci signala koji su generisani, upisuju se u binarne fajlove koji će se koristiti u implementaciji na mikrokontroleru. Ispod je prikazano vreme izvršavanja delova koda u programskom paketu MATLAB.

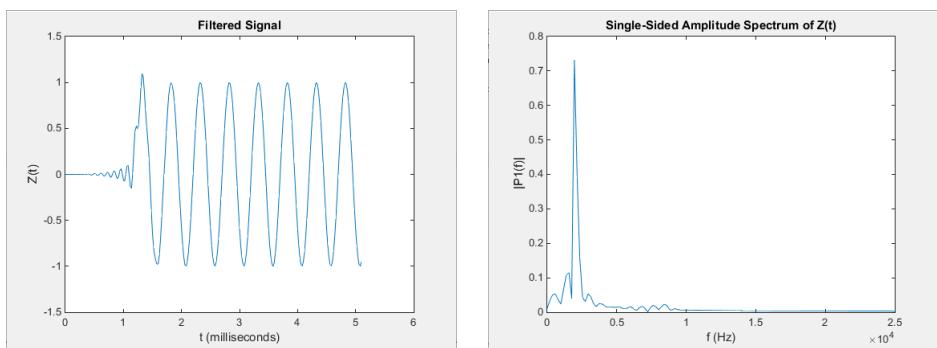


MATLAB: Ulazni signal

MATLAB: Spektar ulaznog signala



Slika 3.1: MATLAB: Karakteristika filtra

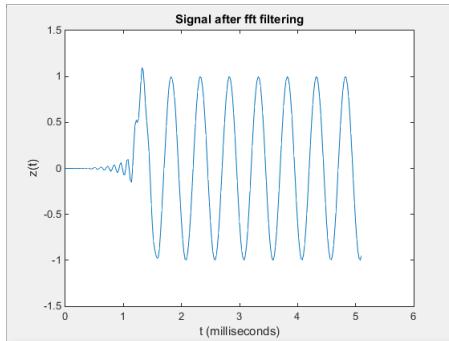


MATLAB: Izlazni signal (FIR)

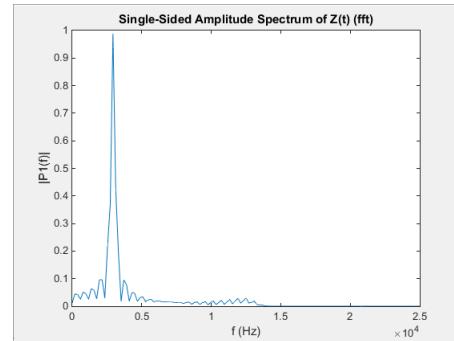
MATLAB: Spektar izlaznog signala (FIR)

3.2 Implementacija na mikrokontroleru

U ovom delu softverske implementacije, filtriranje već izgenerisanih signala se vrši na strani mikrokontrolera. Ova implementacija je izvršena u



MATLAB: Izlazni signal (FFT)



MATLAB: Spektar izlaznog signala (FFT)

```
>> make_signal
Filter function MATLAB:4.8808e-05
FFT filtering MATLAB:9.8851e-05
```

Slika 3.2: MATLAB: Vremena izvršavanja filtriranja u sekundama

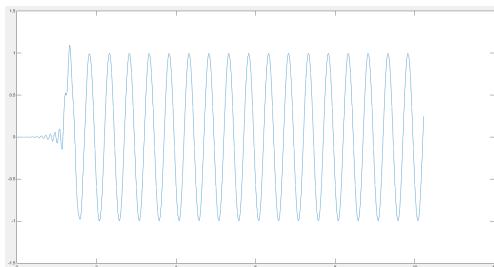
IAR Embedded Workbench for ARM razvojnom okruženju koji sadrži IAR kompjuler za ARM arhitekture.

Kako su odbirci signala koji je potrebno izfiltrirati, kao i odbirci filtra koji se koristi za filtriranje smešteni u binarne fajlove, na strani mikrokontrolera korišten je mehanizam *semihosting-a*. *Semihosting* je mehanizam koji omogućava kodu koji se pokreće na ARM platfomi da komunicira i koristi ulazno-izlazne mehanizme na **host** PC-u, odnosno na PC-u na kome se pokreće razvojno okruženje. Ovaj mehanizam omogućava sve operacije sa fajlovima na ARM mikrokontroleru, kao i ispis i čitanje u Debug terminalu unutar razvojnog okruženja. Korišćenjem ovog mehanizma, pojednostavljen je pristup podacima koji se nalaze na samom PC-u, ali ovaj mehanizam ima ograničenje, a to je da se može koristiti samo u delu projektovanja i debugovanja sistema. Odnosno, samo ukoliko se debugger koristi. Ukoliko se želi postići da sistem radi samostalno, potrebno je naći novi način za dobijanje podataka na mikrokontroleru, bilo to prenosom preko serijskog interfejsa, ili na neki drugi način. Nakon dobijanja svih potrebnih podataka na mikrokontroleru, implementirana su dva načina filtriranja, kao i u prethodnom delu. A to je filtriranje u vremenskom i frekvencijskom domenu. Oba načina filtriranja su implementirana korišćenjem CMSIS biblioteke i njenih funkcija za **fir** filtriranje, kao i funkcija za **fft**.

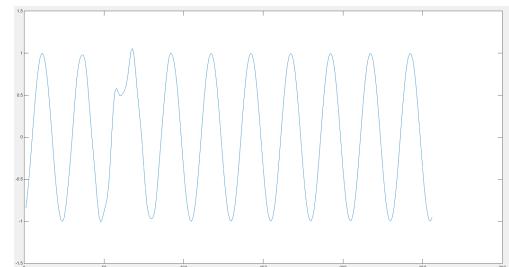
Za filtriranje u vremenskom domenu, korišćena je funkcija **arm_fir_f32** koja vrši filtriranje signala **fir** filtrom. Funkcija za filtriranje se izvršava nad blokovima velicine 16 podataka, i formira izlazni signal. Nakon čega se izfiltriran signal smešta u izlazni binarni fajl. Nakon čega se može poreediti u

MATLAB-u.

Takodje, filtiranje u frekvencijskom domenu se vrši korišćenjem funkcija iz CMSIS biblioteke, i to konkretno funkcije **arm_cfft_f32**, kojom se dobija FFT ulaznog signala i filtra. Nakon čega se vrši množenje FFT predstava signala i filtra u kompleksnom domenu. I na posletku, istom funkcijom **arm_cfft_f32** se radi inverzna FFT. Kao i u prethodnom delu, izlazni signal se smešta u binarni fajl, nakon čega se može porediti u MATLAB-u.

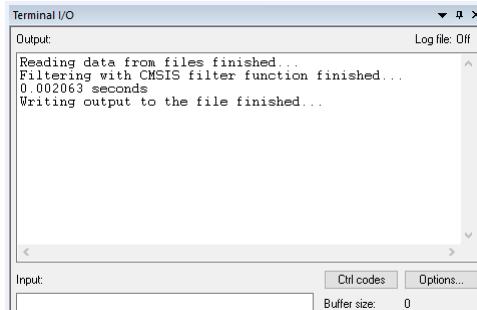


MCU: Izlazni signal (FIR)



MCU: Izlazni signal (FFT)

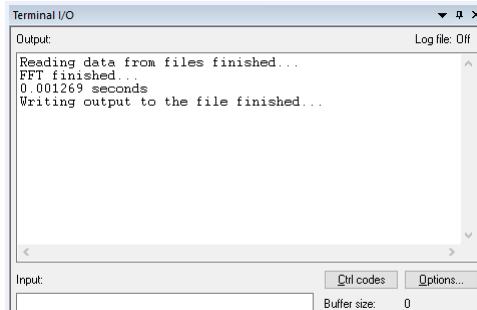
Ispod je prikazano vreme izvršavanja delova koda na strani mikrokontrolera.

A screenshot of a terminal window titled "Terminal I/O". The "Output" pane shows the following text:

```
Reading data from files finished...
Filtering with CMSIS filter function finished...
0.002063 seconds
Writing output to the file finished...
```

The "Input" pane is empty. At the bottom, there are buttons for "Ctrl codes" and "Options...".

MCU: Vreme filtriranja (FIR)

A screenshot of a terminal window titled "Terminal I/O". The "Output" pane shows the following text:

```
Reading data from files finished...
FFT finished...
0.001269 seconds
Writing output to the file finished...
```

The "Input" pane is empty. At the bottom, there are buttons for "Ctrl codes" and "Options...".

MCU: Vreme filtriranja (FFT)

Glava 4

Poredjenje rezultata

Rezultati dobijeni na PC-u i mikrokontroleru su isti. Tj. rezultat filtriranja ulaznog signala je isti. Korišćenjem datih signala, dobija se identičan rezultat na kraju filtriranjem u vremenskom i u frekvencijskom domenu. Takođe, treba uzeti u obzir i vremena izvršavanja. Uvidja se da je filtriranje u frekvencijskom domenu u programskom paketu MATLAB sporije nego filtriranje u vremenskom domenu. To se može objasniti optimizacijom funkcija koje su korišćene za vremensko filtriranje. Vremena izvršavanja na mikrokontroleru su drugačija od vremena u MATLAB-u, i vidi se da je izvršavanje filtriranja u vremenskom domenu dva puta sporije od filtriranja u frekvencijskom domenu, gde treba uzeti u obzir da podataka koji se filtriraju u frekvencijskom domenu ima dva puta manje, zbog ograničenja razvojne ploče. Takođe, i u ovom slučaju je korišćena specijalizovana biblioteka za obradu signala, CMSIS DSP. Iz ovoga se može zaključiti da korišćenje funkcija za filtriranje određuje u mnogome brzinu sistema.