

RaspberryPi generator funkcija sa AD9850 integrisanim kolom

Projekat iz predmeta Računarska Elektronika

Kandidat
Risto Pejašinović EE19/2015

Mentor
Prof.dr. Ivan Mezei

6. septembar 2019

Sadržaj

1	Uvod	3
1.1	AD9850	3
1.1.1	Reset sekvenca	4
1.1.2	Konfiguracioni registar	5
1.1.3	Sekvenca za učitavanje konfiguracije	5
1.1.4	AD9850 modul	6
2	Aplikacija	7
2.1	Režimi rada	7
2.1.1	Konfiguracija aplikacije	7
2.2	Uputstvo za instalaciju	8
2.3	CLI aplikacija	9
2.3.1	Prednosti CLI aplikacije	9
2.3.2	Parsiranje argumenata komandne linije	9
2.3.3	Uputstvo za upotrebu	10
2.4	GUI aplikacija	10
2.4.1	Opis dizajna	10
2.4.2	Basic kartica	11
2.4.3	Sweep kartica	12
2.4.4	Status widget	12
2.4.5	Dijalog za konfiguraciju	12
3	Zaključak	13
3.0.1	Predlozi proširenja	13
3.0.2	Slike rada	14

Mali rečník

Akronim	Opis
API	= Application programming interface
BASH	= Bourne Again SHell
CLI	= Command Line Interface
DAC	= Digital Analog Converter
DDS	= Direct Digital Synthesis
GUI	= Graphical User Interface
RPi	= RaspberryPi
SBC	= Single Board Computer
UI	= User Interface

1 Uvod

Cilj ovog projekta je projektovanje preciznog i programabilnog generatora sinusnog i pravougaonog signala sa jednostavnim User Interface (UI).

Potrebno je projektovati jednostavan i intuitivan Graphical User Interface (GUI) i Command Line Interface (CLI) korisnički interfejs.

Za potrebne korisničkog interfejsa i zadavanja konfiguracije generatoru funkcija odabran je Single Board Computer (SBC) RaspberryPi (RPi) pod GNU/Linux operativnim sistemom.

Za generisanje sinusnog i pravougaonog signala odabrano je AD9850 integrisano kolo.

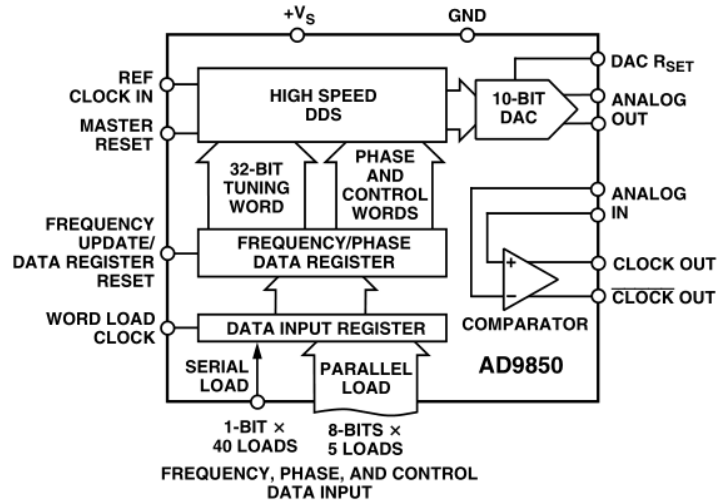
Potrebno je obezbediti iste funkcionalnosti u GUI i CLI aplikaciji. Ovo je postignuto korišćenjem jedinstvenog Application programming interface (API) za AD9850 integrisao kolo. Obe aplikacije trebaju implementirati pozive funkcija iz API .

CLI aplikacija omogućava podešavanje generatora funkcija pomoću skriptnih jezika poput Bourne Again SHell (BASH) i samim tim omogućava automatizaciju i proširenje funkcionalnosti prema potrebi korisnika. Takođe korisnik može implementirati dodatne funkcionalnosti u C++ jeziku korišćenjem AD9850 API.

1.1 AD9850

AD9850 je integrisano kolo firme Analog Devices i koristi Direct Digital Synthesis (DDS) tehnologiju zajedno sa Digital Analog Converter (DAC) i brzim komparatorom generiše precizan sinusni i pravougaoni signal programabilne frekvencije visoke preciznosti.

Pravougaoni signal je generisan pomoću sinusne funkcije i komparatora. Vreme ispunje pravougaone funkcije se podešava analognim ulazom na komparatoru, pa nije moguće digitalno podešavanje bez eksternih komponenti.



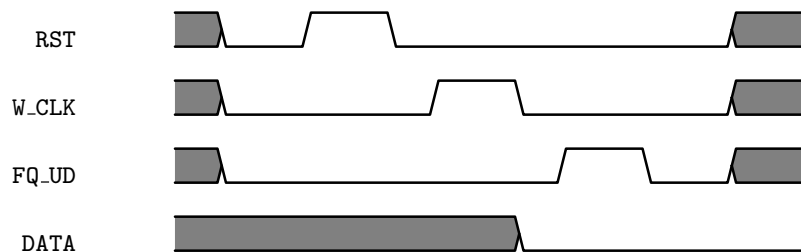
Slika 1.1: Funkcionalni dijagram AD9850[1]

Frekvencija se podešava postavljanjem 32-bitnog registra na željenu vrednost, ovime je postignuta preciznost od 0.0291 Hz pri frekvenciji referentnog oscilatora od 125 MHz. DDS omogućava frekvenciju to polovine referentnog takta, ali je DAC ograničava na 40 MHz. Pored podešavanje frekvencije moguće je i podešavanje faze između dva analogna izlaza.

Moguća je promena izlazne frekvencije brzinom do 23 miliona novih frekvencija u sekundi. Ova brzina će zavisiti od načina upisivanja u konfiguracioni registar. Moguće je upisivati konfiguraciju serijski ili 8-bita paralelno. U slučaju serijskog upisa potrebno je upisati 40-bita što će oduzeti 40 upisnih taktova, prilikom paralelnog upisa potrebno je 5 upisnih taktova.

1.1.1 Reset sekvenca

Pre učitavanja konfiguracionog registra potrebno je resetovati AD9850. Kako bi se AD9850 resetovao potrebno je generisati sekvencu kao na slici(1.2).



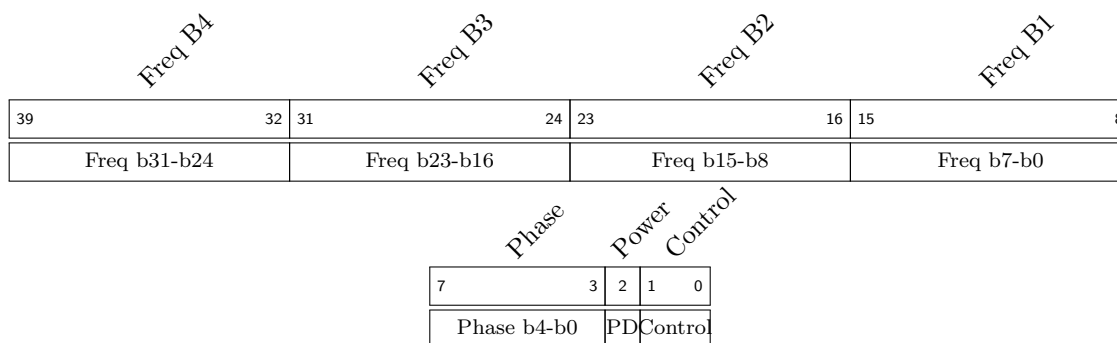
Slika 1.2: Reset sekvenca

Nakon resetovanja AD9850 je spreman za prihvatanje konfiguracije i generisanje signala. Sekvencu za neophodno generisati prilikom uspostavljanja napajanja, dok prilikom promene konfiguracije reset nije neophodno odraditi.

1.1.2 Konfiguracioni registar

AD9850 se u potpunosti upravlja preko jedno 40-bitnog registra u kojem su sadržana frekvencija signala, faza i kontrolni bitovi.

Register 1.1: KONFIGURACIONI REGISTAR



Freq B4-B1

Podešava izlaznu frekvenciju generatora.

Vrednost ovog registra može se dobiti sledećom formulom:

$$Freq[31 : 0] = \frac{freq * 2^{32}}{DDS_CLK}$$

freq je željena frekvencija, DDS_CLK je referentni takt.

Phase

Podešava fazu.

Power

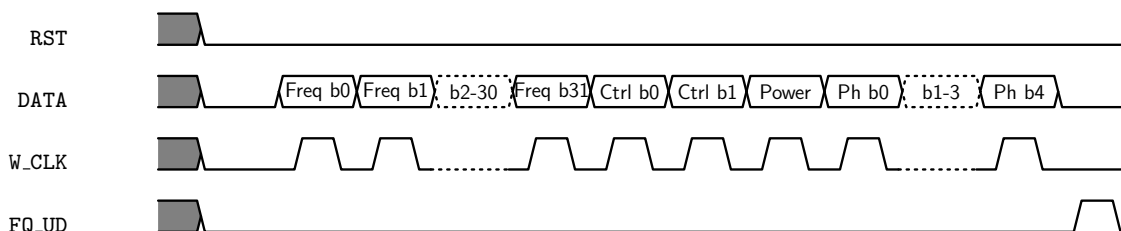
Kada je ovaj bit setovan na jedinicu AD9850 je u Power-Down modu.

Control

Treba biti 0 kako bi oba DAC izlaza bila aktivna.

1.1.3 Sekvenca za učitavanje konfiguracije

Nakon uspešnog resetovanja potrebno je upisati željenu konfiguraciju. Upisivanje u serijskom modu je prikazano na slici (1.3).

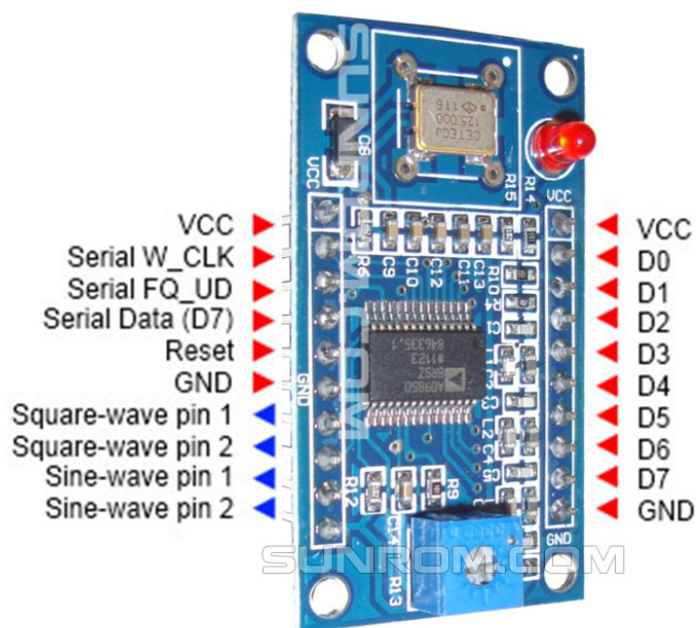


Slika 1.3: Sekvenca serijskog učitavanja

Za detaljne informacije o tajminzima pogledati datasheet [1].

1.1.4 AD9850 modul

Na slici 1.4 prikazan je raspored pinova AD9850 modula korišćenog u ovom projektu.



Slika 1.4: Raspored pinova AD9850 modula[2]

Pinove označene sa prefiksom Serial je potrebno koristiti u slučaju serijske komunikacije. Pinovi obeleženi sa DX se koriste za paralelno upisivanje i u ovom slučaju mogu se ostaviti nepovezani.

Plavom strelicom su označeni izlazni pinovi, od kojih su dva za pravougani signal dva za sinusni.

Na ploči se nalazi referentni oscilator od 125 MHz.

Na ulaz internog komparatora za generisanje pravougaonog signala je doveden napon sa potencijometra na slici, ovo omogućava ručno podešavanje vremena ispuke pravougaonog signala. Na žalost ovaj modul ne dozvoljava programabilno podešavanje vremena ispuke pravougaonog signala.

2 Aplikacija

2.1 Režimi rada

Kao što je već rečeno CLI i GUI aplikacija trebaju da obezbede iste funkcionalnosti. Režimi rada koji su implementirani u ovom projektu su opisani u nastavku.

- **Run** pokretanje generatora sa fiksnom frekvencijom bez vremenskog ograničenja rada, potreban argument:
 - **freq** zadata frekvencija generisanog signala u Hz.
- **Run for** pokretanje generatora sa fiksnom frekvencijom sa određenim vremenom trajanja.
 - **freq** zadata frekvencija generisanog signala u Hz.
 - **time_ms** trajanje generisanja signala u ms.
- **Sweep** linearna promena frekvencije između dve definisane vrednosti.
 - **start_freq** početna frekvencija u Hz.
 - **stop_freq** krajnja frekvencija u Hz.
 - **step_freq** vrednost koraka frekvencije u Hz.
 - **step_time** vreme zadržavanja na svakom koraku u ms.

Ovaj režim rada generatora se često koristi u ispitivanju frekvencijskih karakteristika elektronskih kola ili komponenti. Za prikazivanje frekvencijske karakteristike na osciloskopu, pored signala promenljive frekvencije potrebno je obezbediti i trougaoni signal kao izvor vremenske baze osciloskopa.

Ovakav uređaj se naziva Sweep Generator ili Wobbler. Često se jedan kanal osciloskopa može koristiti kao izvor vremenske baze dok drugi služi kao ulaz za Y-osu.

2.1.1 Konfiguracija aplikacije

Potrebno je obezbediti mogućnost promene pinova preko kojih je RPi povezan sa AD9850 modulom bez ponovnog kompajliranja aplikacije. Takođe ovo podešavanje treba da bude zajedničko za CLI i GUI aplikaciju.

Ovo je rešeno korišćenjem deljenog konfiguracionog fajla.

Konfiguracioni fajl treba da sadrži sledeće parametre.

- **w_clk** broj WiringPi pina povezanim sa **W_CLK** pinom na AD9850 modulu.
- **fq_ud** broj WiringPi pina povezanim sa **FQ_UD** pinom na AD9850 modulu.
- **data** broj WiringPi pina povezanim sa **D7** pinom na AD9850 modulu.
- **rst** broj WiringPi pina povezanim sa **RST** pinom na AD9850 modulu.

- **dds_clk** frekvencija referentnog oscilatora u Hz na AD9850 modulu.

Na **GNU/Linux** operativnim sistemima konfiguracioni fajlovi se trebaju nalaziti u definisanom deljenom direktorijumu.

Prema **XDG Base Directory Specification**[3] od freedesktop projekta treba se poštovati sledeće pravilo za skladištenje konfiguracionih fajlova.

- Ukoliko je promenljiva okruženja **XDG_CONFIG_HOME** postavljena, tada ona sadrži putanju direktorijuma gde se smeštaju korisnički konfiguracioni fajlovi za aplikacije.
- Ukoliko je promenljiva okruženja **XDG_CONFIG_HOME** prazna kao direktorijum za smeštanje korisničkih konfiguracionih fajlova treba koristiti **\$HOME/.config**.
- Svaka aplikacija koja ima potrebe za konfiguracionim fajlom treba imati svoj direktorijum unutar **XDG_CONFIG_HOME** direktorijuma gde će se nalaziti konfiguracioni fajlovi.

Ovo ne deluje kao previše bitna stavka, ali bi poštovanje ovih pravila trebalo biti obavezno. Zbog nepoznavanje ove specifikacije od strane programera nastali su programi koji nemaju zajednički direktorijum za konfiguracione fajlove, već se oni mogu naći svuda po sistemu. Ovo stvara problem prilikom backup-ovanja sistema, ukoliko su svi konfiguracioni i korisnički fajlovi na definisanom mestu backup sistema može biti automatizovan i siguran. Ova specifikacija definiše direktorijume i za ostale korisničke fajlove.

Potrebno je na siguran način parsirati i upisivati ove konfiguracione fajlove. Kako bi se izbeglo ponovno pisanje ovakve biblioteke i izbegli rizici za ovaj projekat izabrana je **libconfig**[4] biblioteka.

Za pristup GPIO pinova RPi korišćena je **WiringPi**[5] biblioteka

2.2 Uputstvo za instalaciju

Kako bi se aplikacija uspešno kompajlirala potrebno je prvo instalirati potrebne pakete i biblioteke koje se koriste u ovom projektu.

Korišćeni paketi bi trebale biti dostupne na repozitorijumu svake poznatije Linux distribucije. U ovom slučaju je korišćena Arch Linux ARM distribucija pa će za nju biti opisana instalacija paketa.

```
1 $ pacman -S qt5-base libconfig
2 $ yay -S wiringpi-git
```

U ovom slučaju koristi se **yay AUR helper** za pristup AUR repozitorijumu. Ukoliko je potrebna samo CLI aplikacija **qt5-base** paket nije potreban.

Nakon ovoga potrebno je klonirati git repozitorijum ovog projekta

```
1 $ git clone https://github.com/Risto97/ad9850_rpi
2 $ cd ad9850_rpi
3 $ make
```

Opciono može se pokrenuti **sudo make install** kako bi se aplikacija instalirala u **/usr/bin** direktorijumu. U suprotnom izvršni fajl će se nalaziti u **bin** direktorijumu kloniranog git repozitorijuma.

Ukoliko je potrebna samo CLI ili GUI aplikacija može se uraditi **make cli_app** ili **make gui_app**. Za dokumentaciju potrebno je izvršiti **make docs**.

2.3 CLI aplikacija

CLI je aplikacija koja se izvršava bez grafičkog okruženja preko komandne linije. Ovo je često bolji pristup za određene aplikacije. Kod nekih aplikacija grafičko okruženje nije neophodno i često ih je lakše i pouzdanije upravljati preko komandne linije.

2.3.1 Prednosti CLI aplikacije

CLI aplikacije su manje kompleksne od GUI aplikacija i samim tim postoji manje mogućnosti za postojanje bug-ova. Funkcionalnost programa je najčešće bitnija od grafičkog interfejsa, jednostavnost CLI aplikacije omogućava programeru da se fokusira više na samu funkcionalnost.

Dodatna prednost CLI aplikacije je lakše pokretanje i podešavanje iz skriptnih jezika (BASH, Python) ovo često kod GUI aplikacija nije moguće. Skriptovanje omogućava automatizaciju poslova i uklanjanje potrebu za ljudskim faktorom.

2.3.2 Parsiranje argumenata komandne linije

Jedan od zadataka CLI aplikacije je parsiranje argumenata komandne linije. Kako bi se skratilo vreme razvoja i izbegla mogućnost grešaka može se koristiti neka od biblioteka za parsiranje argumenata, u ovom projektu je korišćena **cxxopts**[6] biblioteka.

Ova biblioteka se sastoji samo od jednog Header fajla, time uklanja potrebu za kompajliranjem dodatne biblioteke.

2.3.3 Uputstvo za upotrebu

Pokretanjem **ad9850_cli** izvršnog fajla sa argumentom *—help* na **stdout** će se ispisati uputstvo za upotrebu.

Primer 2.1: ad9850_cli help opcija

```
1  $ ad9850 --help
2  AD9850 cli application.
3  Usage:
4  AD9850 [OPTION...]
5
6  --run arg          Run with frequency arg
7  --run-for arg      Run with frequency arg[0] for time arg[1] in ms
8  --sweep arg        Sweep from freq arg[0] to freq arg[1] with step freq
9                    arg[2] and step time arg[3] in Hz and ms
10 --stop             Stop ad9850
11 --read-cfg         Prints tuple of configuration parameters to stdout
12 --write-cfg arg    Write cfg, arg is tuple = w_clk,fq_ud,data,rst,dds_clk
13 --help             Show help
```

Na osnovu uputstva možemo pokrenuti generator u Sweep režimu na sledeći način.

Primer 2.2: ad9850_cli sweep opcija

```
1  $ ad9850 --sweep 1000,10000,100,10
2  Sweeping from 1000 Hz to 10000 Hz, with step 100 Hz and step time 10 ms
```

2.4 GUI aplikacija

Napisana je pomoću Qt Framework-a [7].

Za GUI aplikaciju potrebno je obezbediti intuitivan i jednostavan interfejs. Potrebno je obezbediti pristup svim režimima rada navedenim u sekciji 2.1.

Pored toga treba implementirati ispisivanje i promenu konfiguracionog fajla odnosno promenu pinova kojom RPi pristupa AD9850 modulu i frekvenciju referentnog takta.

2.4.1 Opis dizajna

Za potrebe ovog projekta osmišljen je sledeći dizajn.

Glavni prozor aplikacije je podeljen na dva **Widget**-a [8]. Levi Widget je klase **QTabWidget** koji podržava kartice.

Desni widget je fiksni i prikazuje statusne informacije, na njemu se još nalazi i taster za zaustavljanje generatora.

Na glavnom prozoru se nalazi još i **QMenu**[9], preko kojeg se može pokrenuti dijalog za pristup konfiguracionom fajlu koji je prikazan na slici(2.4).

Widget sa karticama trenutno sadrži Basic i Sweep karticu. Basic kartica prikazana na slici(2.1) pokriva **Run** i **Run for** režime rada.

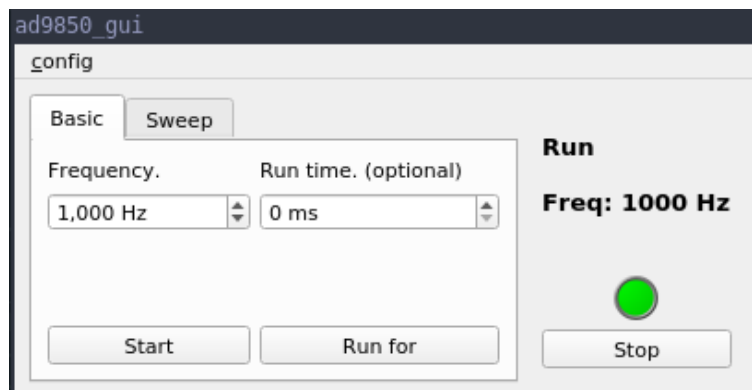
Sweep kartica pokriva prikazana na slici(2.3) pokriva **Sweep** režim rada.

Organizacijom ovog widget-a pomoću kartica ostavlja se mogućnost proširenja programa sa novim režimima rada, a da se pritom ne narušava dizajn interfejsa i izbegne prenatrpanost.

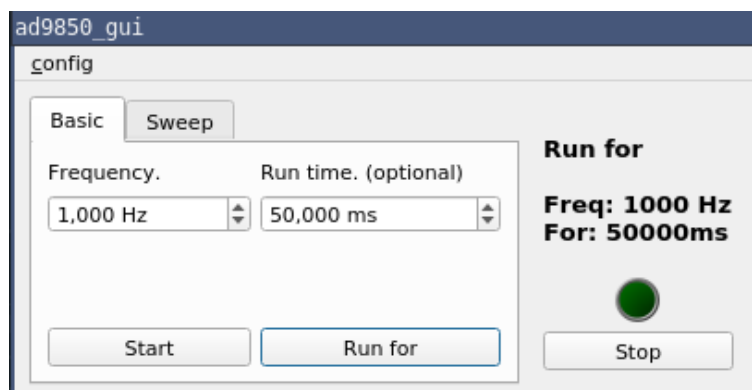
2.4.2 Basic kartica

Kao što se vidi na slikama(2.1, 2.2), ova kartica se sastoji iz dva polja za unos, realizovani pomoću QSpinBox[10] klase i dva tastera QPushButton[11] klase.

Tasteri **Start** i **Run For** pokreću generator u režimima **Run** odnosno **Run For**. U slučaju režima **Run** razmatra se samo polje za unos sa opisom **Frequency**. U slučaju režima **Run For** potrebno je uneti vremensko trajanje rada generatora u polje za unos sa opisom **Run Time**.



Slika 2.1: AD9850 GUI Basic Tab

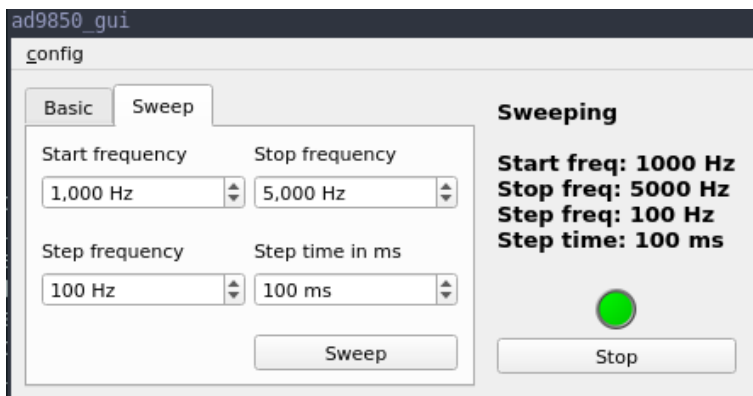


Slika 2.2: AD9850 GUI run for

2.4.3 Sweep kartica

Na slici(2.3) prikazana je Sweep kartica, ona se sastoji iz četiri polja za upis i tastera Sweep.

Opisi parametara u ovom režimu se nalaze u sekciji(2.1)



Slika 2.3: AD9850 GUI Sweep Tab

2.4.4 Status widget

Potrebno je da korisnik ima uvid u trenutno stanje generatora, u tu svrhu je napravljen Status widget koji se nalazi sa desne strane interfejsa. Može se videti na slikama(2.1, 2.2, 2.3).

Na njemu se nalazi QLabel[12] koji prikazuje trenutni režim u kojem generator radi i njegove parametre, QLedIndicator[13] predstavlja indikator da li je generator aktivan ili nije i taster za stopiranje generatora.

2.4.5 Dijalog za konfiguraciju

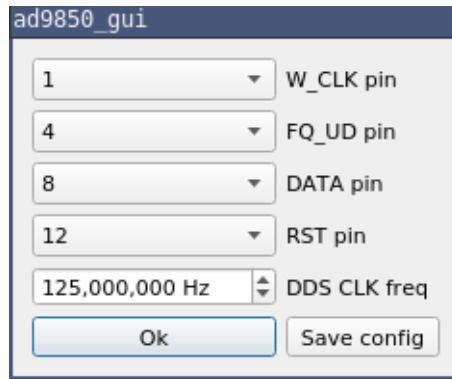
Dijalogu za konfiguraciju može se pristupiti preko config menija na glavnom prozoru, prikazan je na slici(2.4).

Sastoji se od četiri QComboBox[14] padajućih prozora u kojima se nalaze brojevi dostupnih pinova prema WiringPi biblioteci, jednog QSpinBox polja za unos frekvencije referentnog oscilatora i dva tastera.

Prilikom pokretanja ovog dijaloga čita se konfiguracioni fajl opisan u sekciji(2.1.1), zatim prikazuje pročitani parametar u odgovarajuće polje.

Nakon promene vrednosti padajućih prozora ili polja za unos moguće je sačuvati promene u konfiguracioni fajl pritiskom na taster **Save config**.

Pritiskom na taster **Ok** aplikacija učitava novo podešene parametre.



Slika 2.4: AD9850 konfiguracioni dijalog

3 Zaključak

Razvijena aplikacija omogućava jednostavno upravljanje osnovnim režimima rada AD9850 integrisanog kola.

RPi se pokazao kao dobra platforma za razvoj korisničkog interfejsa generatora funkcija i ostavlja mogućnost za lako proširenje funkcionalnosti.

3.0.1 Predlozi proširenja

U nastavku biće predloženi načini na za proširenje ovog projekta. Implementacijom ovih predloga može se dobiti uređaj za testiranje u oblasti audio elektronike i drugih oblasti na višim frekvencijama.

Većina ovih predloga zahteva projektovanje dodatnih analognih elektronskih kola pa zbog toga njihova implementacija nije razmatrana u ovom projektu.

- **Dodavanje izlaznog pojačavača** sa mogućnosti dodavanja napona offset napona kako bi se obezbedili standardni naponski nivoi prisutni kod komercijalnih generatora funkcija (+, - 15V).

Zbog relativno širokog opsega frekvencija generisanog signala (0-40MHz) ovo nije trivijalan zadatak i potrebno je korišćenje odgovarajućeg operacionog pojačavača sa dovoljno širokim frekventnim opsegom, alternativno ograničiti izlaznu frekvenciju na uži opseg.

- **Softversko podešavanje ispune pravougaonog signala.**

Modul korišćen u ovom projektu nema spoljni pristup ulazu internog komparatora i potrebno je izvršiti modifikaciju na ploči ili projektovati novu ploču. Potrebno je obezbediti analogni napon sa mogućnosti podešavanja softverskim putem, najverovatnije korišćenjem integrisanog AD konvertora.

- **Proširenje projekta na Wobbler generator.**

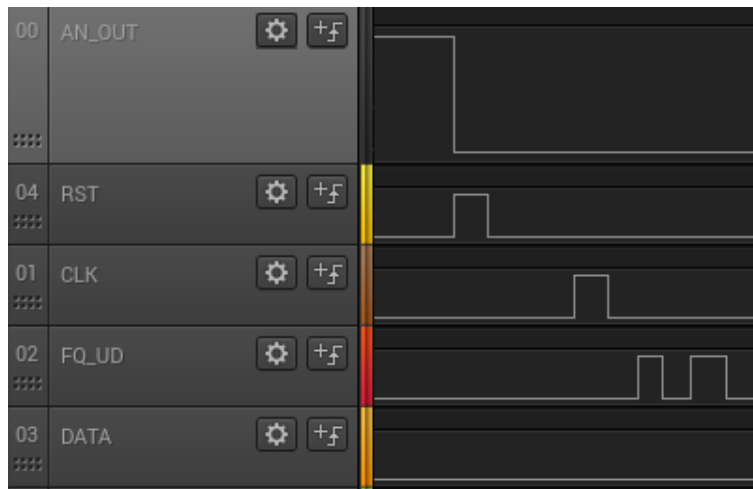
Realizovan sweep režim rada predstavlja osnovu Wobbler generatora.

AD9850 je pogodan za realizaciju ovog uređaja zbog širokog opsega izlazne frekvencije i mogućnost brze promene frekvencije.

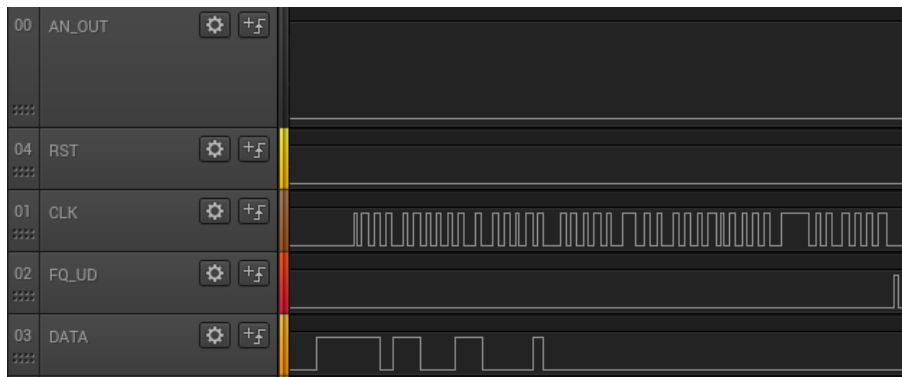
Pored Sweep signala potrebno je obezbediti i trouglasti signal za vremensku bazu osciloskopa kao i potrebne pojačavače za oba signala.

3.0.2 Slike rada

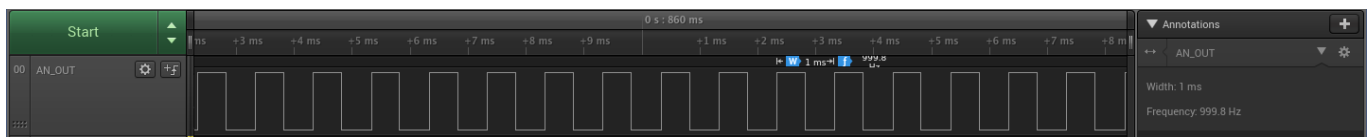
U nastavku su okačene slike sa logičkog analizatora koje prikazuju rad uređaja.



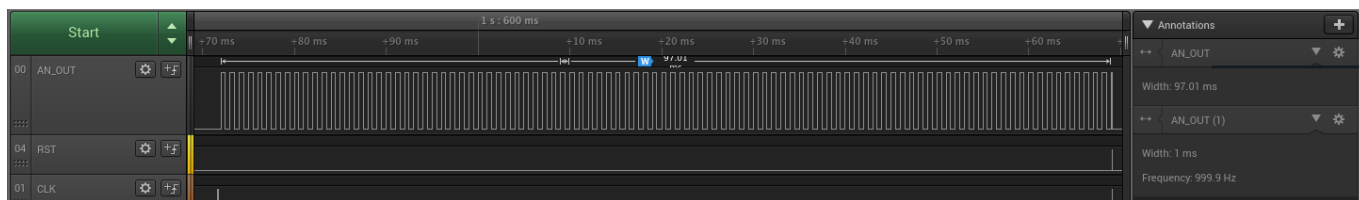
Slika 3.1: Reset sekvenca



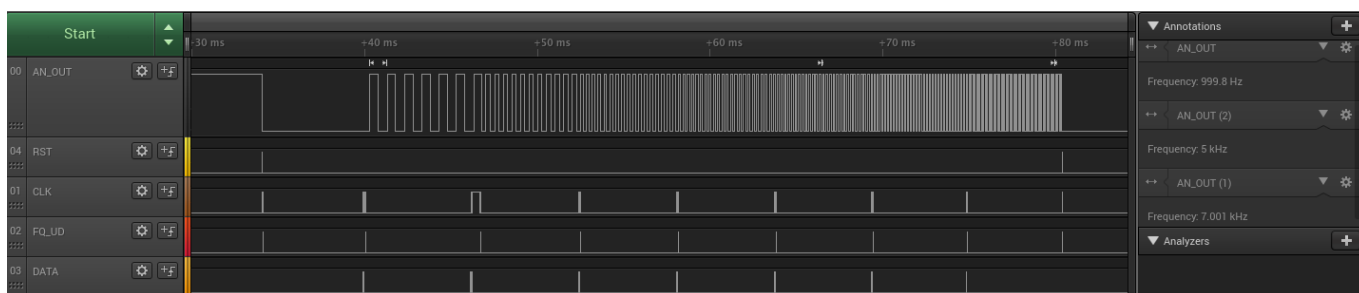
Slika 3.2: Serijski upis



Slika 3.3: Run režim, frekvencija 1kHz



Slika 3.4: Run for režim, 1kHz 100ms



Slika 3.5: Sweep režim, 1kHz do 7kHz, korak od 1kHz i vreme koraka 50ms

Literatura

- [1] A. Devices, “Ad9850 datasheet.” [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9850.pdf>
- [2] R. Butera, “Ad9850 module pinout.” [Online]. Available: <http://robert.butera.org/?p=337>
- [3] L. P. Waldo Bastian, Ryan Lortie, “Xdg specification.” [Online]. Available: <https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>
- [4] hyperrealm, “Libconfig.” [Online]. Available: <http://hyperrealm.github.io/libconfig/>
- [5] G. Henderson, “Wiringpi.” [Online]. Available: <https://doc.qt.io/qt-5/qcombobox.html>
- [6] jarro2783, “cxxopts.” [Online]. Available: <https://github.com/jarro2783/cxxopts>
- [7] T. Q. Company, “Qt.” [Online]. Available: <https://www.qt.io/>
- [8] —, “Qwidget.” [Online]. Available: <https://doc.qt.io/qt-5/qwidget.html>
- [9] —, “Qmenu.” [Online]. Available: <https://doc.qt.io/qt-5/qmenu.html>
- [10] —, “Qspinbox.” [Online]. Available: <https://doc.qt.io/qt-5/qspinbox.html>
- [11] —, “Qpushbutton.” [Online]. Available: <https://doc.qt.io/qt-5/qpushbutton.html>
- [12] —, “Qlabel.” [Online]. Available: <https://doc.qt.io/qt-5/qlabel.html>
- [13] thenobody@poczta.fm, “Qledindicator.” [Online]. Available: https://github.com/Risto97/ad9850_rpi/blob/master/src/gui/qledindicator.h
- [14] T. Q. Company, “Qcombobox.” [Online]. Available: <https://doc.qt.io/qt-5/qcombobox.html>