



Rádióamatőr tanfolyamot segítő jegyzet, egyelőre kidolgozás alatt
Szabó Áron HA1FLX, Mérő László HA1CNT, Bazsó Márton HA7BM
Retzler András HA7ILM szoftverrádiós vevője alapján

Szoftverrádió gyakorlat

Tartalom

1. Bevezető	2
2. Előkészületek	2
3. Elméleti összefoglaló	2
3.1. IQ demodulátor	2
3.2. Komplex jelek	3
3.3. Frekvencia moduláció	4
4. Feladatok	5
4.1. FM demoduláció	5

1. Bevezető

A gyakorlat célja a szoftverrádió lehetőségeinek megismerése, és bemutatása egy C nyelven megírt FM demodulátoron keresztül.

2. Előkészületek

A gyakorlathoz Linux operációs rendszerre lesz szükség.

Szükséges csomagok:

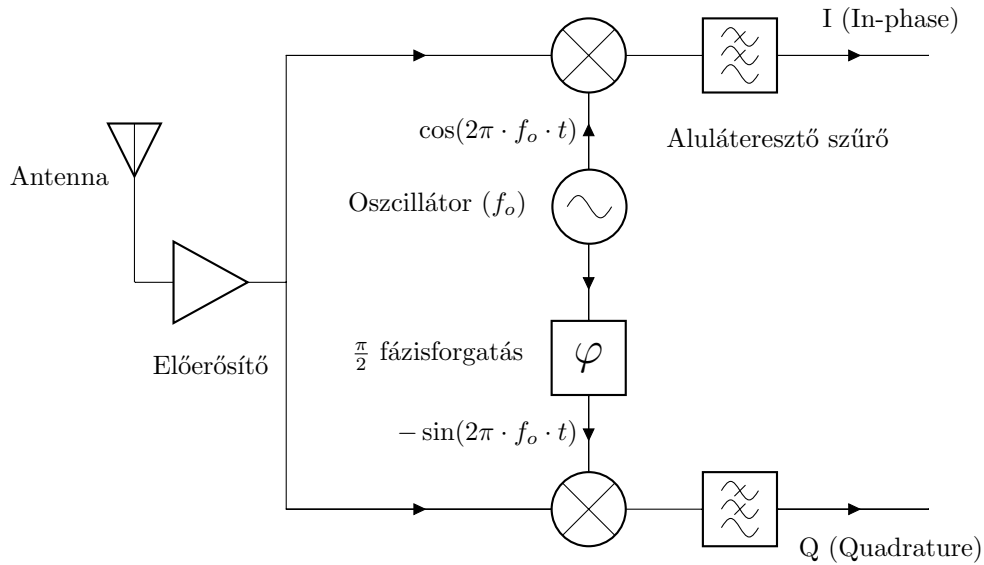
- netcat
- rtl-sdr *(saját Realtek alapú szoftverrádiós eszköz esetén)*
- tcc
- sox
- aplay
- *ízlés szerinti szövegszerkesztő program*

3. Elméleti összefoglaló

3.1. IQ demodulátor

Az IQ demodulátor egy szoftverrádiós eszközben gyakran használt demodulátor. Ilyen hardver található a gyakorlaton használt RTL-SDR USB stickben is. Az IQ demodulátorban a bejövő jelet két azonos frekvenciájú, de 90° -kal eltolt jellel keverjük le (3.1. ábra). A felső keverési termékeket egy low-pass filterrel kiszűrjük, így kapjuk az I (in phase) és a Q (quadrature) jeleket.

Ezen jeleknek a frekvenciája a helyi oszcillátor és a vett jel frekvenciájának különbsége lesz. Az I és Q jelet lehet külön-külön digitalizálni, ezen a ponton már nem szükséges nagyon gyors analóg-digitális átalakítás, hiszen alacsony frekvenciájú jelekkel dolgozunk. Az I és Q jelet együtt, mint $I + jQ$ komplex jelet értelmezzük. Ez már digitálisan feldolgozható, elvégezhető a demoduláció.



1. ábra. Az IQ demodulátor egyszerűsített blokkvázlata

3.2. Komplex jelek

A komplex jelek szoftveres feldolgozása gyakran könnyebb, mint a csak valós számokból állóké. Az eredeti vett jel szinuszos komponensei komplex szinuszoidként jelennek meg.

A komplex szinuszoidok a következő alakot veszik fel:

$$Ae^{j(\phi+2\pi ft)} = A(\cos(\phi + 2\pi ft) + j \sin(\phi + 2\pi ft))$$

Az IQ vektorokat ábrázolva ezeket A hosszúságú f frekvenciával az origó körül körbeforgó vektorokként fogjuk látni. A arányos a jel amplitudójával, f pedig a helyi oszcillátor és a vett komponens frekvenciája közötti eltérés.

Amennyiben a vizsgált komponens frekvenciája nagyobb volt, mint a helyi oszcillátoré, f pozitív lesz, tehát az IQ vektor az óramutató járásával ellentétesen forog. Ha a vizsgált komponens frekvenciája kisebb volt, mint a helyi oszcillátoré, f negatív lesz, tehát az IQ vektor az óramutató járásával megegyezően forog.

Ha az LO és a vett komponens frekvenciája megegyezik, $f = 0$ lesz, így a demodulált IQ vektor konstans. Ilyenkor ϕ az LO és a vett jel közötti fáziseltérés. (Általánosságban a $\phi + 2\pi ft$ kifejezés értelmezhető a két jel fáziseltéréseként, azonban ez időben változó, ha azok nem azonos frekvenciájúak.)

3.3. Frekvencia moduláció

Frekvencia modulációnál egy vivő jel frekvenciájába kódoljuk az $x_m(t)$ moduláló jel által hordozott információt. Az $y(t)$ modulált jelünket a következő képpen számolhatjuk ki:

$$y(t) = A \cos(2\pi t(f_c + f_\Delta x(t)))$$

Itt A az amplitúdónk, f_c a középfrekvenciánk, ez a "vivő eredeti frekvenciája", illetve az állomás névleges frekvenciája (pl. 97.1 MHz). f_Δ a frekvenciatöket, ± 1 közé normált moduláló jelnél az kimeneti jel energiájának nagy része $f_c \pm f_\Delta$ közé esik.

4. Feladatok

4.1. FM demoduláció

A feladathoz egy RTL-SDR hardvert fogunk felhasználni. Az USB stick a digitalizált IQ jelet egy bytestreamként küldi: egy 1 byte-os I majd egy 1 byte-os Q mintát küld felváltva. A 8 bites minták offszettel ábrázolják a negatív értékeket (így például a jel 0 értékét valójában a 127-es bináris érték jelenti). A bytestreamet a demoduláló programunk a standard inputon fogja megkapni, majd a standard outputra kell, hogy kiadja az abból kiszámolt hangmintát.

Segítségképpen a minta beolvasás, illetve kiírás már szerepel a kiinduló kódban. A fogadott IQ jel magasabb mintavételi rátával fog jönni, mint amit a hangkártya kezelni tud, azonban ezt egy külső programmal a demoduláció után fogjuk csökkenteni. Ezért minden fogadott IQ mintára pontosan egy hangmintát kell a programnak kiadni.

Lista 1. Kiinduló kód

```
1 #include <math.h>
2 #include <stdio.h>
3
4 int main()
5 {
6     double i, q, s;
7     for(;;) //vegtelen ciklus
8     {
9         // beolvassuk az I mintát, az offszetet ←
10         // levonjuk, hogy a 0 tenyleg 0 legyen
11         i=((unsigned char)getchar()-127);
12         // Q-val ugyan ez
13         q=((unsigned char)getchar()-127); //beolvassuk
14
15         // s-be kell kiszámolni a demodulált hangot
16         // a kiírás miatt s-t úgy kell skalazni hogy ←
17         // kb. -127 es 128 koze essen.
18         // s = ??;
19
20         // s-t visszaalakítjuk offszetese majd ←
21         // kiírjuk
22         putchar((unsigned char)(s+127));
23     }
24 }
```

A programot saját hardverrel a következő parancssorral lehet futtatni:

```
1 rtl_sdr -s 240000 -f 89500000 -g 20 - | tcc -lm -run ↵  
    minidemod-wfm.c | sox -t raw -r 240000 -e unsigned -b 8↵  
    -c 1 - -t raw - rate 48000 | aplay -f U8 -c1 -r 48000 ↵  
    --buffer-size=200000
```

Ha nincs saját hardvered, használhatod a Kafu `rtl_mus` szerverét ezzel a paranccsal:

```
1 nc teto.sch.bme.hu 7373 | tcc -lm -run minidemod-wfm.c | sox↵  
    -t raw -r 240000 -e unsigned -b 8 -c 1 - -t raw - rate ↵  
    48000 | aplay -f U8 -c1 -r 48000 --buffer-size=200000
```

Mindkét parancssorban az egyes programokat a `|` (pipe) karakter választja el, ami azt jelzi a shellnek, hogy a balra lévő program standard outját adja át a jobbra lévő program standard injére.

Az `rtl_sdr` program az USB stickről olvas mintákat, amiket a standard outra ír. A `-s` kapcsoló a mintavételi rátát, a `-f` a helyi oszcillátor frekvenciáját, a `-g` pedig az erősítést állítja. Az `nc` (netcat) egy TCP kapcsolatot hoz létre a megadott szerverrel a megadott porton, és az onnan jövő adatot a standard outra írja. A megadott gépen és porton egy `rtl_mus` nevű program fut, ami a szerverbe dugott RTL-SDRből jövő mintákat fogja továbbítani.

A `tcc` lefordítja a C kódunkat, és egyből futtatja is az elkészült programot. A `sox` az újramintavételezést végzi el, a 240 kHz-s jelből 48 kHz-eset csinál. Az `aplay` pedig kijátsza a kapott mintákat a hangkártyán.