

FM rádió hallgatása programmal lépésről lépésre

Kezdjük a szoftveres rádiók témáját egy olyan projekttel, amely két FM állomást képes egyszerre befogni.

A *Linux Journal* 2004 júniusi számában megjelent „GNU Radio: A rádiófrekvenciás világ felfedezésének eszközei” című cikkemben (*Linuxvilág*, 2004 július 68-72. oldal) áttekintést adtam a GNU rádiórendszer működéséről és ismertettem néhány hardveres módszert a rádiófrekvenciás (RF) jelek számítógépes digitalizálására. Ebben a cikkben azt vizsgálom, miképpen használhatjuk a *GNU Radiot* FM műsorok hallgatására.

Az 1. ábrán a cikkhez használt eszköz felépítése látható. Ez a nyers és minden sallangtól mentes megközelítés tökéletesen alkalmas a működés magyarázatára. A cikk későbbi részében összefoglalom az *USRP (Universal Software Radio Peripheral, univerzális programrádió-periféria)* legfontosabb tudnivalóit is.

A készülékünk egy közönséges FM dipólantennából, egy próbatáblára szerelt kábelmodemes vevőegységből és egy 20 millió jel/másodperc mintavételi sebességű analóg-digitális átalakítást végző PCI kártyából áll. A vevőegység bemenetére az antenna csatlakozik. A vevőegység IF-kimenete egy darab koaxiális kábellel csatlakozik a számítógép hátlapján lévő ADC-bemenetre. A vevőegység próbatáblája a PC párhuzamos kapujára van kötve, így lehetőségünk van az egység vezérlésére is.

A használt egységek: a *Microtune 4937 DI5 3X7702* kábelmodemes vevőegység és egy *Measurement Computing PCI-DAS 4020/12 ADC* kártya. Ez a vevőegység-típus már nehezen beszerezhető, de mások – mint például a *Sharp Microelectronics* által gyártottak – is jó szolgálatot tesznek (lásd a kapcsolódó címeket tartalmazó részt).

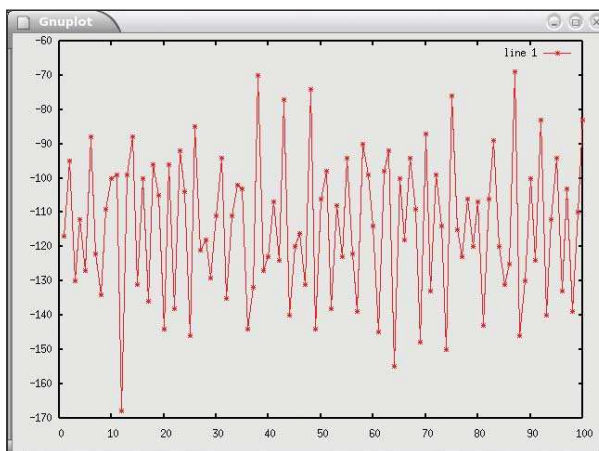
A kábelmodemes vevőegység tölti be az RF előtag szerepét és ez felelős azért, hogy a számunkra fontos rádiófrekvenciás jeltartományt olyan frekvenciasávba konvertálja, amit az ADC-egységünk kezelni tud. Esetünkben az egység az 50 MHz-800 MHz-ig tartó frekvenciatartomány egy választható 6 MHz-es szeletét alakítja át egy 5,75 MHz frekvenciaközéppel bíró 6 MHz-es szeletre. Az elvről részletesebben az említett júliusi lapszámban megjelent cikkben olvashatunk.

Vágjunk bele!

Elsőnek vizsgáljuk meg, mi történik, ha az előtagunkat az FM frekvenciasáv közepére, mondjuk 100,1 MHz-re hangoljuk. A 2. ábra mutatja a kapott mintákat az idő függvényében. Ez az időtartományban mutatott nézet az, amit egy oszcilloszkópon is láthatunk. Ebből sok mindent nem tu-

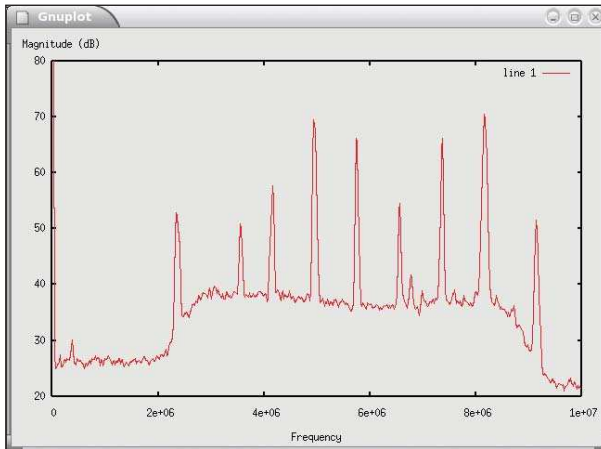


1. ábra A kábelmodemes vevőegység rádiófrekvenciás előtagja

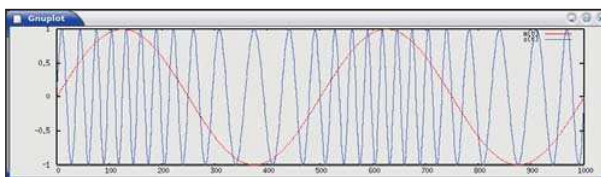


2. ábra Az ADC-minták képe az időtartományban

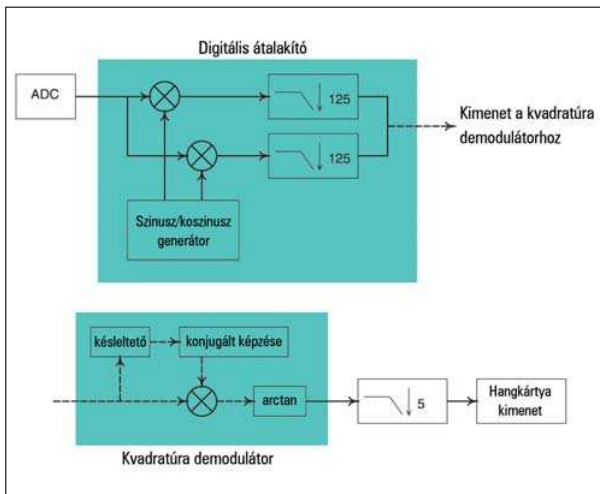
dunk kiolvasni, az mindene esetre biztató, hogy a minták a 170-től 70-ig tartó intervallumba esnek. Ideális esetben a jelek a null értékhez képest szimmetrikusan helyezkednének el, de a céljaink szempontjából ez az eltolás nem lényeges. A frekvenciatartománybeli vizsgálat további információkat szolgáltat. Ebben az esetben egyazon időben 1024 mintát veszünk és ennek számítjuk ki a Fourier-transzformáltját, a gyors Fourier-transzformáció (FFT) algoritmusának alkal-



3. ábra Az FM sáv gyors Fourier-transzformáltja kilenc állomással



4. ábra Egy egyszerű frekvenciamodulált jel



5. ábra Az FM vevő blokkdiagramja

mazásával. Ezzel képet kapunk azokról a frekvencia-összetevőkről, amelyek a bemeneti jelben szerepelnek. A 3. ábrán láthatjuk a kapott spektrumot. Az x-tengely a frekvenciát jelöli, az y-tengelyen pedig a teljesítmény decibelben kifejezett ($10 \cdot \log_{10}$ teljesítmény) értékei szerepelnek. Az alsó határ nulla Hz, a felső pedig 10 MHz, a mintavételi frekvenciánk fele.

A 3. ábrán látható tűskék mindegyike egy-egy rádióállomást jelent. A programunk elsőre látja mindegyiket! Egy állomás hallgatásához módot kell találnunk arra, hogy az állomás jeleit elválasszuk az összes többi állomásétól, alakítsuk át az alapsávra (egyenáram, 0 Hz) és alakítsuk vissza a frekvenciamoduláció hatását. Lépésről lépésre végig fogunk haladni ezeken a folyamatokon, de mindezek előtt nézzük, mit is takar az FM rövidítés.

1. lista A kvadrátúra-demodulátor fejállománya

```
#ifndef INCLUDED_GR_QUADRATURE_DEMOD_CF_H
#define INCLUDED_GR_QUADRATURE_DEMOD_CF_H

#include <gr_sync_block.h>

class gr_quadrature_demod_cf;
typedef boost::shared_ptr<gr_quadrature_demod_cf>
    gr_quadrature_demod_cf_sptr;

gr_quadrature_demod_cf_sptr
gr_make_quadrature_demod_cf (float gain);

/*
 * quadrature demodulator: complex in, float out
 */
class gr_quadrature_demod_cf : public
    gr_sync_block
{
    friend gr_quadrature_demod_cf_sptr
    gr_make_quadrature_demod_cf (float gain);
    gr_quadrature_demod_cf (float gain);

    float        d_gain;

public:

    int sync_work (
        int noutput_items,
        gr_vector_const_void_star &input_items,
        gr_vector_void_star &output_items);
};

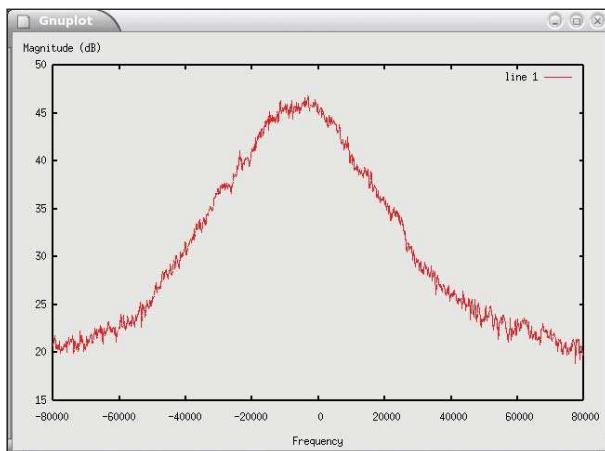
#endif /* INCLUDED_GR_QUADRATURE_DEMOD_CF_H */
```

Mi az a frekvenciamoduláció?

Ahhoz, hogy megértsük egy FM vevő működését, nem árt ismernünk az FM jelek keletkezésének módját. A frekvenciamoduláció (FM) során a vivőfrekvencia pillanatnyi értékét a bemeneti jel frekvenciájának a függvényében változtatjuk. A 4. ábrán láthatjuk az $m(t)$ bemeneti jelet (beszéd vagy zenei jel), és a létrejövő $s(t)$ modulált jelet. Matematikai precizitással felírva, a pillanatnyi frekvencia mindenkor az alábbi összefüggés szerint számolható:

$$f(t) = k \cdot m(t) + f_c$$

Az $m(t)$ jelöli a bemeneti jel frekvenciáját, a k egy állandó érték, amely a frekvenciaérzékenységet szabályozza, az f_c pedig a vivőjel frekvenciája (például 100,1 MHz). Emlékezzünk rá, hogy a frekvencia mértékegysége radián/másodperc, így a frekvenciát tekinthetjük forgási gyorsaságnak is. Ha integráljuk a frekvenciát, fázist vagyis szögértéket kapunk. Megfordítva pedig a fázis idő szerinti differenciálása frekvenciát ad eredményül. Ezek azok a kulcsmotívumok, amiket a vevő megépítésekor felhasználunk.



6. ábra Gyors Fourier-transzformált a digitális frekvenciaváltó kimenetén

A blokkdiagram

Az 5. ábrán látható az FM állomások hallgatására szolgáló módszerünk. Ha eltávolítjuk a vivőhullámot, egy olyan alapsávú hullámunk marad, amelynek pillanatnyi frekvenciája arányos az eredeti jel $m(t)$ értékével. Vagyis az előtűnk álló feladat nem más, mint eltávolítani a vivőhullámot és meghatározni a pillanatnyi frekvenciát.

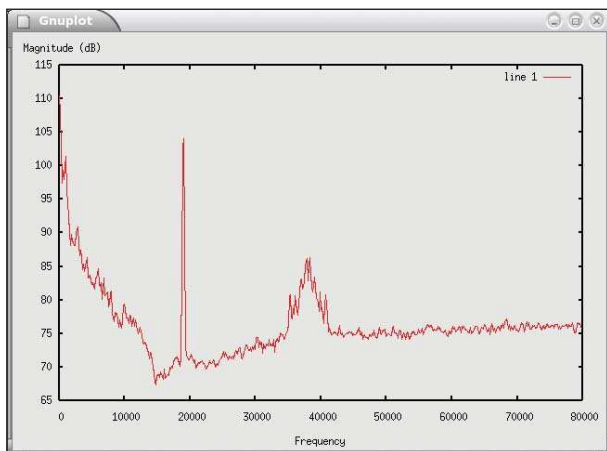
Az első rész nem okoz nehézséget, a vivőhullámtól a programmal megvalósított digitális frekvenciaváltó (DDC) segítségével szabadulunk meg, amelyet `freq_xlating_fir_filter_scf` elnevezéssel illetünk.

Az egység egy numerikus vezérlésű oszcillátorból áll, amely azokat a szinusz és koszinusz hullámokat állítja elő, amelyeket ki szeretnénk oltani, valamint egy keverőből (ez a programunk számára egy szorzónak felel meg), és egy tizedelő véges impulzusválaszú szűrőből. Az `scf` utótag jelzi, hogy ez az egység a bemenetén *short* (rövid) jeleket fogad, a kimenetén komplex jelfolyamot állít elő, a szűrő megadásához pedig lebegőpontos mintákat használ.

A digitális frekvenciaváltó azt a trigonometrikus azonosságot használja ki, hogy amikor két f_1 és f_2 frekvenciájú szinuszhullámot összeszorozunk, az eredmény két másik szinuszhullámból áll elő, amelyek frekvenciái f_1+f_2 illetve f_1-f_2 . Esetünkben a bemenő jelet szorozzuk a vivőjel frekvenciájával. A kimenő jel két összetevőből áll, az egyik a vivőjel kétszerese, a másik pedig egy nullszintű jel. A kétszeres komponensből egy aluláteresztő szűrővel szabadulunk meg, s így megmarad az alapsávú jelünk.

A számítási gyorsaság a lényeg!

A digitális frekvenciaváltó egyenes szoftveres megvalósítása rendkívül számításiigényes feladat. Ekkor a teljes bemenő frekvencián elvégeznénk a szinuszos és koszinuszos jelek előállítását és ezek összeszorozását. Egy *Pentium 4* processzorral a szinusz és koszinusz jelek számítása 150 utasításciklust igényel. Egy 20 millió jel/másodperc sebességű bemenettel számolva ez $20e6 * 150 = 3e9$ ciklust jelentene másodpercenként csupán a szinusz és koszinusz kiszámítására! Ez nyilvánvalóan járhatatlan út. Jó hír azonban, hogy a DDC szoftveres megvalósításának létezik egy jobb módszere is. *Vanu Boseet* a „*Virtual Radios*” (*Virtuális rádiók*) című írásában – lásd a kapcsolódó címet – vázolt eljárás szerint a művele-



7. ábra A demodulált FM jel gyors Fourier-transzformáltja

tek sorrendjének átszervezésével és a valóságos együttathatók helyett a frekvenciaspecifikus komplex-szűrő együttathatóinak használatával elegendő az összes számítást tizedekkora frekvencián elvégezni. Ennek a módszernek az eredménye meghatározó, ezt már valós időben is el tudjuk végezni!

Kvadratúra-demoduláció

A következő feladat az alapsávú jel pillanatnyi frekvenciájának meghatározása. Erre a `quadrature_demod_cf` blokkot használjuk. A fázist közelítő módszerrel választjuk le a szomszédos minták közti szögeltérés meghatározásával. Emlékezzünk vissza, hogy a frekvenciaváltó blokk kimeneteként komplex számok álltak elő. Egy kis trigonometriai ismeret felhasználásával, két egymást követő minta közti szögeltérést meghatározhatunk oly módon, hogy az egyiket megszorozzuk a másik komplex konjugáltjával és az eredménynek vesszük az arkusz tangensét. Az 1. és 2. listán láthatjuk a `quadrature_demod_cf` blokk megvalósítását. Ha már tudjuk, hogy mit akarunk, nem kell sokat kódolnunk. A jelfeldolgozás zömét a `sync_work` ciklusában lévő három sor végzi. Mindezeket kipróbálva a 6. ábrán láthatjuk a digitális frekvenciaváltó kimenetét, a 7. ábra pedig a kvadratúra-demodulátor kimenetét mutatja. A 7. ábrán láthatjuk az FM hullámforma összes összetevőjét. A 0-tól körülbelül 16 kHz-es frekvenciáig terjedő rész a jobb és bal oldal (L+R) együttes hangtartománya. A 18 kHz-nél látható túske a sztereo pilotjel. A bal és jobb oldal különbségele (L-R) a pilotjel kétszeres frekvenciájának értékére középpontozva jelenik meg AM-modulált jelként az FM felső tartományában. Időnként további segéd-vivőfrekvenciákat is találhatunk az 57 kHz és 96 kHz közé eső frekvenciatartományban. Hogy ne bonyolítsuk túl a dolgokat, a kvadrát-demodulátor kimeneti jelét 16 kHz-es frekvenciánál levágjuk, amivel egy monó jelet kapunk, s ezt a hangkártya kimeneteire kapcsoljuk.

Egy többszörös vevőegység

A *LinuxJournal* FTP kiszolgálójáról (lásd a kapcsolódó címet) letölthető 3. lista egy általános vevő megvalósításának Python-kódját mutatja. Valójában arra is képes, hogy két FM állomást fogjon egy időben, egyiket a bal oldali hangszórón keresztül, másikat a jobb oldalin. Nem állítom, hogy ennek különösebben nagy hasznát vesszük, mindenesetre

2. lista A kvadratúra-demodulátor megvalósítása

```

#ifdef HAVE_CONFIG_H
#include "config.h"
#endif

#include <gr_quadrature_demod_cf.h>
#include <gr_io_signature.h>

gr_quadrature_demod_cf::
  gr_quadrature_demod_cf (float gain)
  : gr_sync_block (
    "quadrature_demod_cf",
    gr_make_io_signature(1,1,sizeof
      (gr_complex)),
    gr_make_io_signature(1,1,sizeof (float))),
    d_gain (gain)
  {
    set_history (2); // provide 1 sample look
                    // ahead
  }

gr_quadrature_demod_cf_sptr
gr_make_quadrature_demod_cf (float gain)
{
  return gr_quadrature_demod_cf_sptr (
    new gr_quadrature_demod_cf (gain));
}

int
gr_quadrature_demod_cf::sync_work (
  int noutput_items,
  gr_vector_const_void_star &input_items,
  gr_vector_void_star &output_items)
{
  gr_complex *in = (gr_complex *) input_items[0];
  float *out = (float *) output_items[0];
  in++; // ensure that in[-1] is valid

  for (int i = 0; i < noutput_items; i++){
    gr_complex product = in[i] * conj (in[i-1]);
    out[i] = d_gain * arg (product);
  }

  return noutput_items;
}

```

jól mutatja a programmal megvalósított rádióban rejlő lehetőségeket. A több csatorna egyidejű vételének ez az ötlete felhasználható rádiós *TiVo*-szerű eszközök alapjaként.

A kód három függvényre bomlik. A main kezeli a paraméterellenőrzést, irányítja az RF-előtagot és vezérli a fő jelfeldolgozó ciklust. Ha csak egyetlen állomást veszünk, az RF előtagot arra utasítjuk, hogy az állomást tegye a vevőegység kimeneti frekvenciájának közepére (az IF-re). Ha két állomást fogunk, azoknak egymástól 5,5 MHz-es távolságon belül kell lenniük. Ennek a megszorításnak a kábelmodemes vevőegységben lévő SAW-szűrő az oka. Ez egy 5,75 MHz-re beállított sáváteresztő szűrő, melynek értéke a 6 Mhz-es érték – az észak-

amerikai TV-csatornák sávszélességének – közelében van. Ebben az esetben a különbséget megfelezzük, és az előtagot pontosan a két állomás közé hangoljuk. A `build_graph` az általános jelfeldolgozó blokkokat tartalmazza és kapcsolja egymáshoz. Mind az egy, mind pedig a több csatorna vételi eljárásánál egyetlen nagy sebességű analóg-digitális átalakítót használtunk a bemeneten és egy hangkártyát a kimenet számára. Minden egy időben hallgatni kívánt csatorna számára külön digitális frekvenciaváltó, kvadratúra-demodulátor és aluláteresztő szűrő került megvalósításra.

Többféleképpen is megvalósítható!

Az egyszerre fogható adók száma a számítógépünk sebességének függvénye. Még eme ötletes megvalósításunkra is igaz, hogy a processzorciklusok nagy része a `freq_xlating_fir_filter` blokkokkal terhelt. Az általunk most ismertetett módszer a nyers ADC-módszer nevet kaphatná. A számítási igény csökkentésének egyik módszere a digitális frekvenciaváltás végrehajtásának hardveres megvalósítása lenne. Több gyártó, így a *Texas Instruments*, *Intersil* és *Analog Devices* cégek is kínálnak olyan céláramköröket (*ASIC*), amelyek képesek e feladat betöltésére. A *Universal Software Radio Peripheral (USRP)* által követett módszer a digitális frekvenciaváltó *Verilog* hardverleíró nyelven való lekodolása, majd a keletkezett bitfolyam *USB*-kapun keresztül az *FPGA*-ba való áttöltése. Ezzel egy olyan kombinált megoldást kapunk, amely megőrzi a maximális rugalmasságot, ugyanakkor lehetővé teszi a számításiigényesebb részek végrehajtásának hardverre történő átterhelését. Az *URSP* vonatkozásában további információk a *GNU Radio Wikin* érhetők el.

Összegzés

A cikkben egy egyszerű, de teljes funkcionalitású többsávú FM vevőt vizsgáltunk meg, miközben sikerült egy több ezer dolláros hardvert két ötdolláros tranzisztort tartalmazó rádióval egyenértékűvé tennünk, és képet kaptunk a feldolgozó folyamatokról is. Akiket mélyebben is érdekel az FM rádiózás, a GNU Radio kódtárában egy lényegesen jobb minőségű FM vevőre (*hifi_fm.py*) és egyéb finomságokra bukkanhat. A *GNU Radio* terén mostanra rengeteg értékes munka született, amelyek némelyike a mozgó ideiglenes hálózatokat veszi célba, mások az amatőr rádiózás hagyományait követik vagy a szoftveres *GPS* megvalósításán fáradoznak, egy csoport pedig lázasan dolgozik a következő generációs föld-űr amatőr műholdas kommunikációs rendszer megteremtésén. Habár a *GNU Radio* eszközkészlet nagymértékben független a ki- és bemeneti eszközöktől, ezeknek a törekvéseknek a nagy része az *USRP*-t használja, vagy tervezi használni a rádiófrekvenciás egységek és a számítógép közti csatolófelületként.

Linux Journal 2004. október, 126. szám



Eric Blossom a GNU Radio Project alapítója.

Évekig dolgozott a biztonságos telefonszolgáltatások területén mielőtt programalapú rádiózással kezdett foglalkozni. Ha éppen nem programalapú rádiót bütyköl, akkor valószínűleg jógázással vagy ju-jitsuval tölti idejét.

Az `eb@comsec.com` címen érhető el.