

Nálad mi van a falban?

Alex elmeséli, hogyan derítheted ki mindezt néhány drótdarabbal és egy linuxos számítógép hangkártyájával.

Szögkeresőt szinte fillérékért lehet venni, kinek jutna hát az eszébe, hogy egy linuxos gépet használjon a falban lévő vezetékek, szögek, csavarok vagy egyébek felderítésére?

A CanDetect Project egy összetettebb munka melléktermékeként jött létre, ahol jóval nehezebb, egy olcsó kis kütyүнél többet igénylő feladatot kellett megoldani. A CanDetect a Corroding Aircraft Non Destructive Evaluation using software Tools and an Eddy Current Tester (Repülőgépek korróziójának károsítás nélküli mérése programok és örvényárammérő műszer segítségével) rövidítése, a tervezet célja pedig az, hogy a repülőgép-szerelőket olcsó, a gépeken fellépő korrózió vizsgálására alkalmas eszközzel lássa el. A CanDetect révén megtekarítható a drága célgépek, külső erősítők, modulátorok és tápegységek ára, a szerelők mégis felderíthetik az apró hibákat a festés vagy más fém alkatrészek alatt is.

A berendezés egy bármely Linux alapú gépen futtatható programból, valamint egy érzékelőből áll, ami a `/dev/dsp` eszközt aljzatába közvetlenül csatlakozik – magyarul a hangkártyának megfelelő karaktereszközhöz vezet. Az Embedded Linux Journal NIC (New Internet Computer) versenyére készült egy rendszerindításra is alkalmas CD-ROM lenyomat, ami a célprogramok futtatására használható.

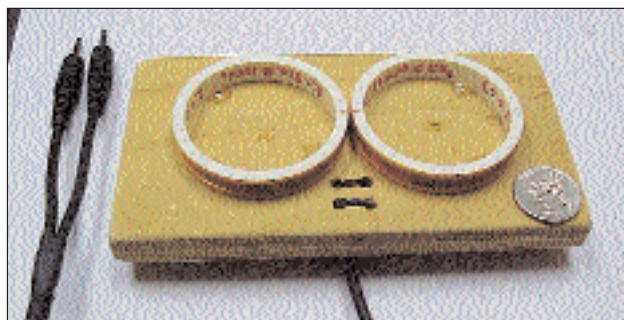
Az érzékelő

Ahol gyakran van földrengés, például Kaliforniában, ott nem biztonságos tégelából építkezni. Sokkal célszerűbb erős és könnyű, alumíniumból vagy fából készülő, függőleges tartóelemekből építeni (kb. 40 cm térközzel), a vázra pedig vékony, sima farostlemez, gipsz vagy valamilyen hasonló, csupán önmagát megtartani képes anyag kerül. Földrengéskor viszont az ilyen fal felborítja a mellé állított bútorokat, hacsak nincsenek biztonságosan hozzárögzítve. Amikor kisebb-nagyobb bútorokat erősítünk a falhoz, a csavarokat mindig a belső vázba kell becsavarni, mert csak ez képes elviselni a terhelést. A falakban futnak az elektromos vezetékek és a csövek is, ezeket viszont – különösen hosszabb csavarokkal – ajánlatos elkerülni.

Az 1. képen látható érzékelőt munkatársam, **Brian Whitecotton** készítette. Az ő munkája szebb, mint az enyém. Az érzékelő két tekercsből áll, amelyeknek a közepe 6 centire van egymástól.

A tekercseket a 2. képen látható módon egy falaphoz ragasztottuk. Mindkét tekercs átmérője 5,58 cm, és 80 menetnyi szigetelt drótból állnak. A két tekercs sorba van kötve – az áram az ellenkező irányban halad bennük körbe – a bal és jobb kimenő vezetékek között. A tekercsek közti felezőpont a mono mikrofonbemenetre csatlakozik. A kimenet és a bemenet földje közös, amint az 1. ábrán is látható. Az itt szereplő elrendezés ellenállása 6,2 Ω, induktivitása 0,8 mH, tényleges impedanciája pedig 50 W 10 KHZ-en.

Az érzékelő építésekor ügyelni kell arra, hogy az összeköttetések egyezzenek, a két tekercs pedig ugyanolyan legyen. Bármennyire is kézenfekvő lenne, az érzékelő építésekor más fém alkatrészeket nem szabad használni, mivel az érzékelő



1. kép 10 KHz-es frekvenciához és 2 cm-es mélységhez tervezett induktív tekercspar

pont a falban lévő fémek felderítésére készül. Egy ilyen érzékelővel egyébként nem lehet repülőgépeket vizsgálni.

A fal felől vett jelek

A fizika iránt kevésbé érdeklődő Linux-felhasználók időnként megriadnak a tervezetünk mögött álló fizikai törvényektől, holott erre igazából semmi okuk nincs. Ha mást nem is, annyit érdemes megjegyezni, hogy a pontszerű források, például a szegecses kerek jelet adnak, a hosszúkás formájúak pedig inkább pontszerűt.

Ahogy a falat a kívülről láthatatlan szegecses után kutatva átvizsgáljuk, a program pozitív vagy negatív változást jelez, ha a szegecs valamelyik tekercs középpontjába esik (lásd a 2. ábrát).

Ha nagyobb falfelületet kell átvizsgálni, a kapott jel jellegéből megállapítható, hogy az egymás mellett húzott tekercsek melyikével találunk például egy szegecseset. Ha a szegecs pontos helyét is meg akarjuk határozni, a két tekercs egymás mögött halad, így követve, hogy a szegecs mikor kerül a két tekercs közé, ekkor ugyanis nulla jelet kapunk. Itt néhány milliméteres pontossággal dolgozhatunk. Az elektromos kábelek és a fém szerkezeti elemek, amint az az 3. ábrán is megfigyelhető, jóval erőteljesebb jelet adnak.

A jelalak alapján lehet megkülönböztetni egymástól a pontszerű (például egy szegecs feje) és a hosszúkás (például valamilyen vezeték) forrásokat. A pontszerű forrás csaknem ugyanolyan jelet ad, bárhol is van a tekercs hatókörén belül, ennél

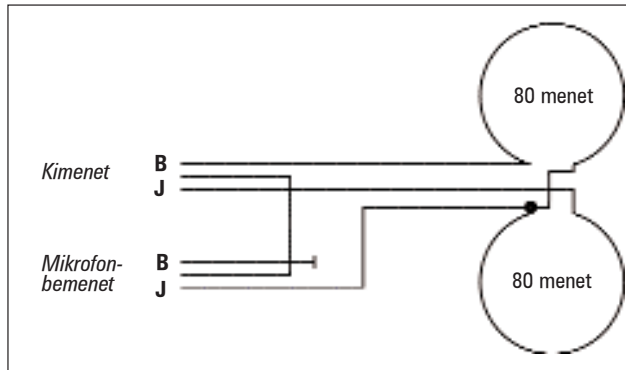


2a

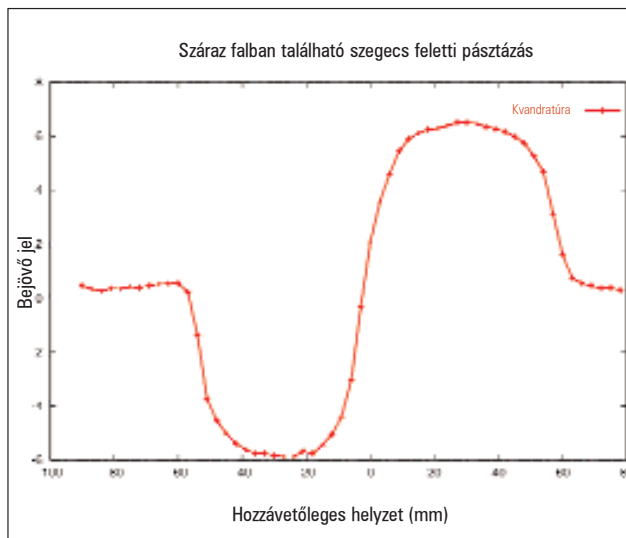


2b

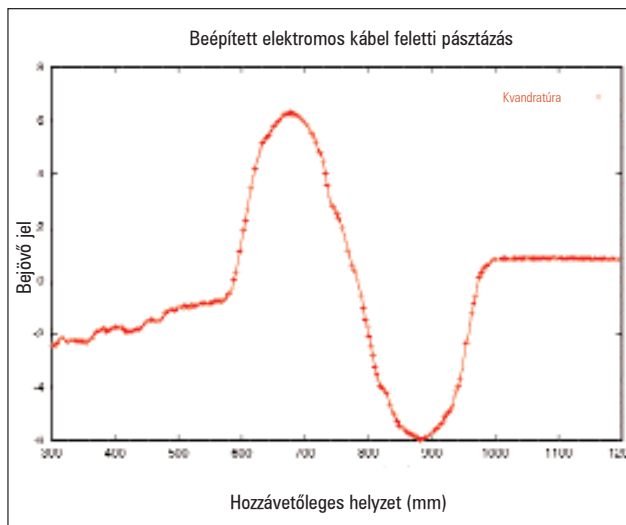
2a, 2b. kép Segítség az 1. ábrán látható érzékelő összeállításához



1. ábra Az 1. képen látható érzékelő kapcsolási rajza

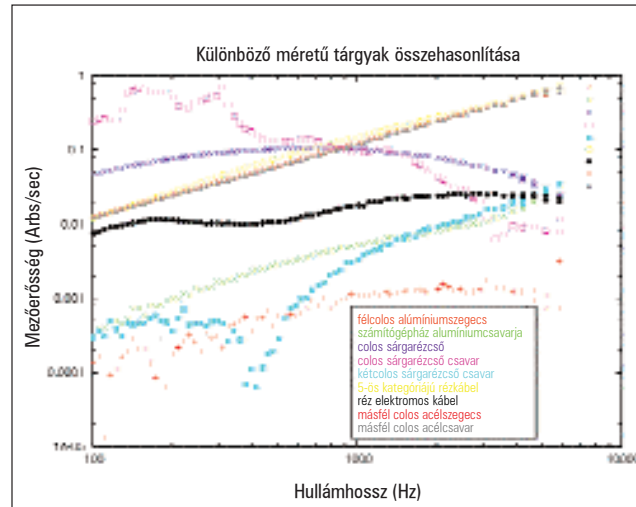


2. ábra Prostar hordozható gép, 1 KHz-es lineáris letapogatás egy száras falban található szegecs felett

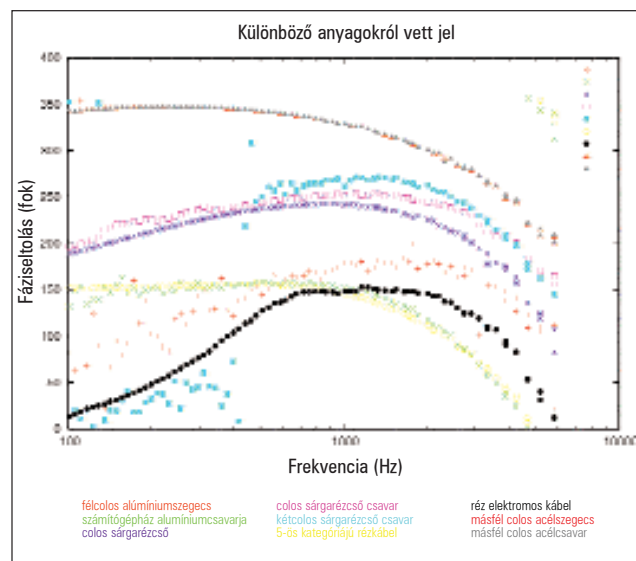


3. ábra Prostar hordozható gép, 1 KHz-es lineáris letapogatás egy 110 voltos elektromos vezeték felett

tehát a jel laposabb jellegű. A hosszúkás forrás jele attól függően változik, hogy mekkora szakasza esik a tekercs hatókörébe, és akkor a legerősebb, ha a forrás pontosan a tekercs közepén



4. ábra A jel mérete utal a tárgy nagyságára. Az alumínium kisebb jelet ad, mint az acél



5. ábra A jel fázisa elárulja, milyen fajtájú fémtárgyat találtunk. A vas és az acél grafikonja magasabb, a réz és az alumíniumé alacsonyabb, a sárgaréz és a hasonló fémeké pedig nagyjából középre kerül. 1 KHz környékén a fázisok jól láthatóan három csoportba tömörülnek

halad keresztül. Ennek jele tehát hegyes, kiugró jellegű. A fémtárgyakat ugyan egy hasonló olcsó keresővel is meg lehet találni, de ha az alapvető helymeghatározáson kívül többre van szükség, finomabb műszert kell beszerezni. Az olcsóbb eszközök arra sem képesek, hogy megkülönböztessék egymástól például az épület vázának elemeit, a vízvezetékeket és az elektromos vezetékeket.

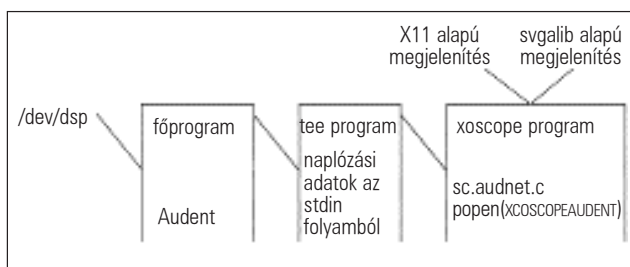
A drágább és nagyobb tudású műszerek apró kijelzőn keresztül jelentik meg a mérések eredményét. Programalapú műszerünk viszont képes kihasználni a nagyobb képernyő adta lehetőségeket, illetve a mérés finomításával a különféle tárgytípusokat is azonosítja. Ha sikerült megtalálni egy tárgyat a falban, a jel változásai alapján állapítható meg, hogy pontosan mire bukkantunk (lásd a 4. ábrát). A jel méretéből megtudható, hogy az adott tárgy mekkora (5. ábra). A jelek hullámhosszváltással módosulnak, ez alapján megbecsülhető a tárgy alakja.

© Kiskapu Kft. Minden jog fenntartva

A számítógépes oldal

A CanDetect Project szerény követelményeket támaszt a gépezemben, nincs különösebb gépígénye. Ha éppen windowsos gépet sikerült kölcsönözni valahonnan, elég egy hajlékonylemezzel vagy CD-ROM-ról elindítani a rendszert. Mi egy olcsó, CD-ről történő indításra alkalmas New Internet Computer (NIC ➔ <http://www.thinknic.com>) gépet használtunk az Embedded Linux Journal versenyén.

Ahhoz, hogy a ki- és bemenő hullámformákat egyszerre lehessen kezelni, a hangkártyának (illetve az alaplapra épített hanglapkának) képesnek kell lennie a teljes kétirányú működésre. Sok régebbi hangkártya és még több jelenlegi hordozható számítógép gyakran csak úgy tud teljes kétirányú módban működni, hogy közben csökkenti a mintavételi felbontást illetve gyakoriságot. A kártyának 16 bites módban kell üzemelnie. A mérések során jobb elkerülni a régebbi vagy rossz hangú kártyákat. A mikrofonbemenetről még a teljes hangerőt beállítva sem szabad hallható sustorgásnak jönnie.



6. ábra A programrészek kapcsolatát szemléltető blokkdiagram

Ha nagyszámú gép közül választhatsz, a Linux Terminal Server Project (LTSP ➔ <http://www.lts.org>) segítségével a mérőprogramot inetd alól induló szolgáltatásként is futtathatod.

Miután minden hálózati kártyához elkészítetted a megfelelő EtherBoot (➔ <http://www.etherboot.org>) hajlékonylemezt, egyetlen este alatt több száz gépet is végigpróbálhatsz. A hajlékonylemezzel az összes gépet hálózatról indítsd el, X-terminálról jelentkezz be távoli munkaállomásodra, majd futtass le egy parancsfájlt. A parancsfájlnak fel kell dolgoznia a DISPLAY változót, meg kell állapítania az IP-címet, csatlakoznia kell a számítógépen futó inetd szolgáltatáshoz, le kell futtatnia a tesztet, majd naplóznia kell, és meg kell jelenítenie az eredményt.

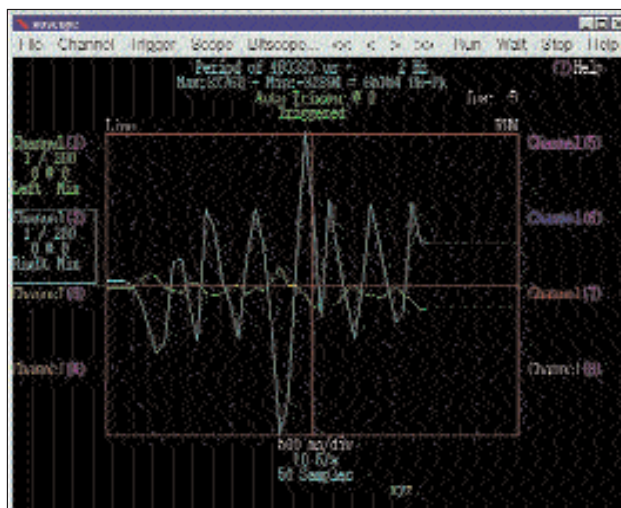
A programok

AudNet fejlesztésünk feladata a hangkártya működtetése. A folyamat megjelenítését az xoscope egy átszabott változata végzi (lásd a 6. ábrát).

Az AudNet feladatának lényege az, hogy a hangkártyán keresztül folyamatosan lejátszik valamit, és a bejövő jeleket is fogadja. A kimenő és bejövő csatornák összhangban tartása nagyon fontos. Általában két bemenő és két kimenő elektromos csatorna van a hangkártyákon, és mindegyik csatornához egy hullámforma tartozik. Minden hullámformaérték a teljes -1...+1 tartományból származhat.

Számos különböző típusú mérési kérés lehet tetszőleges időpontban átadni, amelyek teljesítése a lehető leghamarabb megtörténik. A könyvtár végighalad a feladatok várakozási során, amíg a végére nem ér – már ha a végére ér valaha. Ha a feladatlista kiürül, az utolsó feladatot addig ismételteti, amíg újat nem kap. Minden kérdésben szerepel, hogy a hozzá tartozó hullámformát hány alkalommal kell megismételni.

Mivel a lejátszandó adatokat legalább egy másodperccel a hang-



3. kép Képernyőkép az xoscope ablakáról adatbázis-készítés közben

várakozási sorba kell helyezni, a felvett adatok pedig picivel a tényleges felvétel után jelennek meg a sorban, a könyvtár figyelemmel követi, hogy mennyi adat található a rendszermagnak eme átmeneti táraiban. Az éppen kiküldött kérés gyakran eltér attól, amelynek az adatait a bemenő csatornákon begyűjtjük. Mivel sok elektromágneses rendszer csak kevésbé hajlamos rezonálni, a feladatban azt is meg kell adni, hogy a hullámformát hányszor kell lejátszani, hogy az érzékelő és a minta kellő valószínűséggel nyújtson megbízható eredményt. Ezekre az ugróciklusokra csak akkor van szükség, ha a könyvtár az egyik feladatról egy másik kérésre vált át, feladat ismétlésekor szükségtelenek.

Egyszerű hullámhosszméréshez a példaprogram (main) mindössze két átadott értéket vár: a hullámhossz értékét Hz-ben, és a kimeneti fáziseltolást fokban. A „main” két tiszta, a megadotthoz a lehető legközelebb eső hullámhosszú, a két hangszóró csatornán ellenkező előjellel megjelenő szinuszhullám előállítására utasítja az AudNetet. Ezután az ismétlések számát úgy állítja be, hogy másodpercenként tíz mérési eredményt kapjon.

xoscope – grafikus megjelenítés

Az eredmények megjelenítésére az xoscope módosított változatát használjuk. Egy oszcilloszkóp túlterheltsége esetén nem tudja kirajzolni a bejövő jeleket, ezért elveti őket – az eredeti xoscope is így tesz. A CanDetect számára ez elfogadhatatlan, mivel itt hosszan figyelemmel kell követni egy lassan változó jelet; az xoscope módosításával elhárítottuk ezt a gondot. A Linux-hangkártya illesztőprogramot AudNet alapúra cseréltük le. A NIC-en bármely fájlrendszer-hozzáférés megakadhat, amíg a CD-ROM-meghajtó felpörgésére kell várni, és bármelyik képernyőrajzolásnak addig várakoznia kell, amíg a processzor a gyorsítás nélküli grafikus lapkakészlettel dolgozik. Emiatt az xoscope gyakran több másodpercig nem hívta meg az illesztőprogramunkat. Az időnként öt másodpercet is elérő késleltetés miatt a rendszermag átmeneti tárja betelt, és a mérés kiesett a szinkronból.

Az xoscope másik eszközüllője az XOSCOPEAUDNET környezeti változót olvassa ki, amitől egy 10 Hz mintavételi idejű adatfolyam átadására képes bemenetet várunk. A környezeti változó életre hívása az xoscope indítása előtt történik meg, így a főprogram minden esetben a kívánt beállításokkal indul. Mivel a „main” külön programként fut, fájlokat nem ér el,

a képernyőt nem használja, futását semmi sem akaszthatja meg. Az illesztőprogram arányosítja a lebegőpontos értékeket, így a képernyő egy osztása 1/100 000 részét jelképezi a hangkártya teljes bemeneti tartományának. A teljes tartományból 32 százalékkal kilépő jelek az *xoscope* belső átmeneti tárainak korlátozott dinamikatarományára miatt elvesznek.

Az *xoscope* CVS-en keresztül elérhető módosításait *src* modulunkban folyamatosan követjük. Az *xoscope.patch* fájlt még a beállítások megadása előtt be kell illeszteni. A folt új illesztőprogramokat tartalmaz, illetve néhány alapbeállítást is módosít. A beállító parancsfájl új parancssori kapcsolója, a `--without-vga` letiltja az *svgalib* keresését – a lehetőség különösen akkor hasznos, ha *svgalib*-bel rendelkező gépen végezzük a fordítást, de a program futtatását ilyenekkel nem rendelkező gépen tervezzük.

A bináris állomány összeállítása

Ha az alábbiakat követve létrehozol egy rendszerindításra alkalmas hajlékonylemezt, könnyedén elvégezheted egy próbamérést:

```
$ mkdir candetect.sf.net
$ cd candetect.sf.net
$ cvs -d :pserver:anonymous@cvs.candetect.sf.net :
  ↪ /cvsroot/candetect login
$ cvs -d :pserver:anonymous@cvs.candetect.sf.net :
  ↪ /cvsroot/candetect co .
$ make /dev/fd0
```

Így – ha netán módosítani szeretnéd – a teljes forráskód a rendelkezésedre áll. A tervezet felső szintű *Makefile* állománya

- lefordítja a könyvtárat és a főprogramot;
- letölti az *xoscope* egy különleges változatát CVS-ről;
- bedolgozza a foltot, majd nem X11 alapú használatra állítja be;
- gondoskodik az *oscope* lefordításáról (ez az *svgalib*-hez tartozó bináris);
- letölt és kibont egy 2.4-es sorozatú Linux-rendszermagot;
- többféle hangkártya használatára állítja be, majd lefordítja;
- *BusyBox*ot is tartalmazó fájlrendszert állít össze;
- összegyűrja az egészet, a rendszerindításhoz pedig *syslinux*-ot ad hozzá;
- az eredményt átmásolja egy hajlékonylemezre.

A folyamat kéri a rendszergazdai jelszót, ha rendszergazdai jogosultságokkal kell futnia; a megjelenő parancssort mindig érdemes átolvasni. Megjegyezzük, hogy a *CanDetect* modulokat tartalmazó könyvtár mellett egy *xoscope* nevű is létrejön (így elkerülhető a keveredés a CVS tartalmával). Debian alapú rendszeren a *Makefile* állományok felismerik, ha valamelyik segédprogram hiányzik, és megpróbálkoznak a telepítésével. Egyéb terjesztéseken először a függőségeket kell feloldanod, vagy módosítanod kell a *Makefile* fájlokat.

A Linux előnyei és a további kihívások

A rendszermag több géptípuson is futtatható. A fejlesztést asztali és laboratóriumi munkaállomásokon végeztük, célunk egy hordozható és beágyazott számítógépeken is futtatható alkalmazás készítése volt; ezek ugyanazt a rendszermag API-t használják. Az illesztőprogramok több eszköz meghajtására is képesek, alapállapotban akár nyolc hangkártyát is kezelni tudnak. Egyetlen számítógéppel különböző hangkártyákat használva egyszerre több mérés is végezhető, amelyek ugyanazt a kód API-t használják.

A hangbemenet elérése eszközközön keresztül történik,

ami ugyanúgy fájlként kezelhető, mint például a hálózati kapcsolatok. A hangkártyák vezérléséhez nincs szükség különleges hívásokra. Ez különösen olyan egyszerű gépeknél jelent fontos fegyvertényt, amelyeknél a kijelző mérete vagy a processzor teljesítménye a programok részéről némi alkalmazkodást kíván. A roncsolásmentes mérőeszközök fejlesztésénél a munka leginkább időigényes része a helyes működés igazolása. Az algoritmusok üzembiztonságát csak úgy lehet bizonyítani, hogy a méréseket és a jelfeldolgozást különféle mintákon, különféle operációs rendszerek alatt sorra elvégezzük. Az Open Sound System a Linux-rendszermag más szolgáltatásaival egybefogva alkalmazásunkhoz a legkülönbözőbb számítógépeken biztosít általános futtatási környezetet. Így – mivel a forráskódban nem történtek változások – a program ellenőrzésének eredményei hitelesek lesznek.

A rendszermag átmeneti tárai miatt jelentkező késleltetések révén az algoritmusok alkalmazási köre viszonylag szűk, és mivel időnként különösen nagy értékek mérésére is szükség van, a repülésnél történő használatuk is korlátozott. A jelenlegi rendszermag-fejlesztések éppen ezeknek a korlátoknak a lebontását célozzák – ám a falvizsgálatot ez már nem érinti.

További feladatok

A *CanDetect* készen áll annak igazolására, hogy a repülésben is használható. Az eredmények megjelenítése továbbra is roppant nagy kihívást jelent. A többcsatornás hangkártyákat is szeretnénk támogatni, és ugyanezt az alkalmazást más jellegű mérések elvégzésére is alkalmassá kívánjuk tenni, ezek a következők:

- az alkatrészek ellenállása, teljesítmőképessége, induktivitása méréséhez;
- a páratartalom, a nedvesség és vegyi anyagok méréséhez;
- a tekercskapcsolások mágneses helyzetérzékeléséhez;
- az örvényáramokhoz való roncsolásmentes vizsgálatoknál;
- Hall-mágneses ellenállás iránytűként való használatához;
- a különféle anyagok zajcsillapítási mutatóinak mérésére;
- a pilóták beszélgetéseinek felvételére, mobil helyzetérőztetésre;
- a szonárokhoz és hasonló visszaverődésen alapuló célkereső módszerekhez;
- a mechanikai nyúlás- és alakzatváltozás-mérőknél;
- sugárrezonátorok esetében, például MEMS (mikroelektromechanikus) érzékelőkben.

Köszönetnyilvánítás

A NIC gépet az *Embedded Linux Journal* 2001-es pályázatának döntős résztvevőjeként kaptuk. Győztes pályaművünkről bővebben az *Embedded Linux Journal* (☞ http://embedded.linuxjournal.com/advertising/press/nic_winners.php) 2001. májusi-júniusi számában, illetve a ☞ <http://candetect.sourceforge.net> oldalon lehet olvasni. Az indukciós érzékelőt *Brian Whitecotton* készítette, a fénykép *Keith Ostrom* munkája.

Linux Journal 2002. október, 102. szám



Alex Perry

(alex.perry@ieee.org) PhD-fokozatot szerzett Cambridge-ben, az IEEE egyik vezető munkatársa. Hivatalos pilóta, repülésirányító, és repülésbiztonsági tanácsadó San Diegóban (Kalifornia). A nyílt forrású FlightGear

(☞ <http://www.flightgear.org>) repülőszimulátor egyik fejlesztője.