



Kamerakiszolgáló működtetése héjprogramból

© Kiskapu Kft. Minden jog fenntartva

Egy beágyazott eszköz programozott működtetése nem föltétlen bonyolult dolog. Akár egy közönséges héjprogram is megteszi...

A svéd *Axis Communications AB* néhány éve egy teljesen új koncepciót vezetett be, amikor piacra dobta új kamerakiszolgálóit. Elképzelhető persze, hogy a cég hálózati kamerái (az egyik típusról 2000 szeptemberében cikk is jelet meg a *Linux Journalban*) nem képesek minden területen kiváltani a hagyományos, analóg készülékeket, az azonban kétségtelen előnyük, hogy a vezérlő és digitalizálást végző elektronika egy különálló, masszív felépítésű, semmiféle mozgó alkatrészt, még hűtőventilátort sem tartalmazó dobozban kapott helyet. A készülék egy és négy analóg videóbemenettel rendelkező változatban kapható, és minden típuson van egy átmenő kivezetés is. Ezekkel a kamerakiszolgálókkal a vállalatok lépésenként modernizálhatják biztonsági rendszerüket.

A kamera

A manapság kapható, *Linuxot* futtató beágyazott rendszerek természetük-

ből adódóan meglehetősen specializáltak. A cikkben szereplő kamerakiszolgálóval azért kerültünk kapcsolatba, mert egyik ügyfelünk igényei bár nem voltak kielégíthetők kizárólag az *Axis* kameráival, a cég kameraszervereivel kapcsolatban nekünk is, és az ügyfélnek is pozitív tapasztalatai voltak.

Így aztán úgy döntöttünk, hogy egy másik, piacvezető cégtől fogunk bekérni olyan biztonsági kamerákat, amelyek képesek együttműködni ezzel a készülékkel. Akárcsak a számítógépek egyéb hardverelemei, a kamerák és a hozzájuk tartozó optikák is amolyan pénznyelő automataként képesek működni. Tulajdonképpen kijelenthetjük, hogy nincs az a pénz, amit el ne lehetne költeni, ha az embernek különleges igényei kezdenek támadni.

Nekünk konkrétan egy olyan kültéri *PTZ* (*pan-tilt-zoom*; fordítás-billentés-nagyítás) kamerára volt szükségünk, aminek elfogadható a nagyítása is. A *PTZ* kamera nem olcsó mulatság. A beltéri egységek ugyan csak néhány száz dollárba kerülnek, az olyan kültériek azonban, amelyek elviselik a tűző napot, a klímát és még némi vandalizmust is, több ezer dollárba kerülhetnek.

Mi egy forgótálcára szerelt átlátszó falú házzal védett típus mellett

döntöttünk. Ezek azok a kamerák, amelyeket manapság a legtöbb repülőtéren is látni. Tekintettel arra, hogy ügyfelünk székhelye *Norvégiában* található, fűtött kameraházra volt szükségünk, hogy a téli jegecsedét elkerülhessük.

Íme az eszköz műszaki paraméterei:

- Elforgathatóság 360 fokban
- Dönthetőség +2 és -92 fok között
- Képzékelő: egynegyed hüvelykes CCD (3,2 x 2,4 mm)
- 22-szeres optikai zoom, 4-től 88 mm-ig állítható fókusztávolság
- Érzékenység: 0,07 lux 1/1,5 másodperces zársebesség mellett
- Zársebesség: 1/1,5-től 1/30.000 s-ig
- Legkisebb F-stop érték: f/1,6
- Környezeti terhelés: -40-től +50 fokos tartós terhelésig

A probléma

Az egész projekt mögötti alapelképzelés, amiért egy *PTZ* kamerát választottunk az volt, hogy bizonyos fix pontokat szerettünk volna fényképezni adott időközönként és a képeket feltölteni egy webkiszolgálóra. Itt aztán beleütköztünk egy problémába. A szerverhez adott szoftver természetesen kiválóan kezelte az *FTP* protokollt, és adott időszakonként is tudott képeket készíteni. Ellenben semmilyen módját nem

találtuk annak, hogy felvétel előtt a kamera pozícióját és nagyítását, vagyis egy adott *PTZ* hármassal beállítsunk. Mivel ez így nem felelt meg az ügyfelünknek, a szállítással várunk kellett, amíg áthidaljuk ezt a gondot.

A készüléket vezérlő szoftvert több úton is lehet módosítani. Először is a firmware részét képező valamennyi nyílt forrású komponens forráskódja rendelkezésre áll. Ebben sajnos a vezérlést végző *CGI* interfész nem tartozik bele, de az annyira nem is volt zavaró. A többségbe bele tudtunk tekinteni, és persze módosítani is lehet az eszköz kódját. Mindezzel együtt egy adott firmware kódját nem lehet csak úgy letölteni, azt írásban kell kérni az *Axis* megfelelő osztályától, amely aztán jelképes összegért *CD*-n elküldi a kódot.

Mivel több ügyfelünk igényeinek kielégítéséhez is szükség lett volna ennek a problémának a megoldására, viszonylag gyorsan kellett cselekednünk. Több olyan programozási felület is felmerült, amelyek segítségével átalakíthatjuk az eszköz működését a saját igényeinknek megfelelően. Vizsgálataink szerint a következő *API*-k jöhettek egyáltalán szóba:

- A készülék saját adminisztrációs felülete (webböngészőn keresztül használható)
- *HTTP API*
- Megfelelő szkriptek írása
- Héjprogramból történő vezérlés
- *PHP3 Lite*
- *GCC SDK for Linux/cris*

Akárcsak a legtöbb beágyazott eszközzel kapcsolatban, itt is számolnunk kellett néhány korlátozással és kényelmetlenséggel. Először és mindenek előtt nagyon kevés a rendelkezésre álló hely. Összesen körülbelül 100 kilobájtnyi írható fájlrendszerrel gazdálkodhattunk, ezen kellett elhelyezni az összes, harmadik féltől származó módosítást. Ennyi maradt ugyanis abból a 4 megabájtos Flash tárolóeszközből, ami a firmwaret tartalmazza. Valamennyi szoftver egy csak olvasható fájlrendszeren van, így nem helyettesíthető egy teljes, testreszabott firmware létrehozása nélkül.

Amikor hozzákezdünk a munkához, sem *SSH* kiszolgáló, sem ilyen ügyfél nem állt még rendelkezésünkre, ami azt jelentette, hogy a héjprogramokat vagy egy külön időzítővel kellett vezélni, vagy *PHP* szkriptből kellett őket futtatni. Akkor még csak egy kísérleti *telnet* kiszolgáló volt a készülékhez, azóta viszont portolták rá a *Dropbear SSH* kiszolgálót is.

Semmiképpen nem akartunk külön szervert üzemeltetni pusztán az automatizálás megoldásához, ezért úgy döntöttünk, hogy mindent belső héjprogramokon keresztül fogunk megoldani, amiket az *utask* nevű ütemező fog majd időzíteni. Szerencsénkre ennek az ütemezőnek valamivel kifinomultabbak a képességei, mint amelyeket a közönséges *cron* magáénak tudhat. Az *utask* például külső jelekkel is vezérelhető, vagyis meg lehet neki tanítani, hogy mit tegyen, ha egy külső digitális jel alacsonyról magasra vált, vagy ha a kettős kamera egyszer csak nem ad videójelet.

A funkciók elérhetővé tételének lehetőségei

A beágyazott rendszereknél az egyik nagy kérdés mindig az, miként tegyük elérhetővé a vezérlési funkciókat. Ettől függ ugyanis, hogy a felhasználónak mekkora erőfeszítéssel jár majd, ha a saját igényeihez akarja alakítani az eszköz működését. Nos, esetünkben elég nagy erőlködésre volt szükség, de rögtön hozzá kell tennem, hogy ebből a szempontból nincs két egyforma biztonsági rendszer.

A böngésző alapú felhasználói felület természetesen könnyen kezelhető, és sok szolgáltatást nyújt az emberi felhasználó számára. Ugyanakkor automatizálási célokra rendszerint tökéletesen alkalmatlan.

A *HTTP API* egy egyszerű, kérés-válasz alapú *API*, amit kifejezetten automatizálási célokra készítettek.

Rajta keresztül a legtöbb olyan funkciót elérhetjük, ami egy böngészőben is rendelkezésünkre állna. A visszatérési értéke általában csak egy állapotkód, vagy maga a kért objektum, például egy a videófolyamból kiemelt képkocka, vagy maga a videó. Mindez természetesen

kiválóan megfelel távvezérlési célokra, így nem véletlen, hogy számos gyártó készít több tíz, vagy akár több száz kamera egyetlen vagy néhány központi helyről történő vezérlésére alkalmas szoftvereket ezzel az *API*-val.

A *PHP3 Lite* nevű nyelvjárása kiválóan megfelel akkor, ha egyedi dinamikus lapokkal akarjuk kiegészíteni a böngésző alapú felhasználói felületet, sőt általános célú belső szkriptek írására is alkalmas.

A készülékhez adott, *sh*-val kompatibilis héj programozása a hozzá tartozó segédprogramokon keresztül akkor a legmegfelelőbb választás, ha valamilyen „helyi intelligenciára” van szükség a működtetés során. Ilyen lehet például az, hogy ha valaki megnyomja az ajtócsengőt, akkor kell róla készíteni egy felvételt, aztán be kell engedni egy relé működtetésével.

Végezetül ha valamilyen kifinomult helyi viselkedést kell megvalósítani, akkor a *GCC SDK* lesz a nyerő választás.

Mi is az tehát, ami a rendelkezésünkre áll?

Amint korábban említettem, mi a héjprogramozás mellett döntöttünk, mivel ez felelt meg leginkább a konkrét szituációnak. Ezen kívül felismertük, hogy a rendelkezésünkre álló firmware változat egészen sok hasznos programot tartalmaz:

- *BusyBox* amely a */bin/sh* szerepében a *sash* héjat használja.
- *mish*: Ez egy a *Minix sh*-val kompatibilis héj.
- *utask*: Feladatütemező, amely azonban nem kompatibilis a *cron*-nal.
- *buffered*: Képkészítő/pufferelő eszköz.
- *sftpclient*: Egyszerű *FTP* ügyfél.
- *shttpclient*: Egyszerű *HTTP* ügyfél.
- *smtpclient*: Egyszerű levélküldő program.

Külön érdemes kiemelni a háttérben szerényen megbújó *shttpclient*-et, hiszen ez lehetővé teszi, hogy a belső héjprogramokból a *HTTP API* szolgáltatásait használjuk, amire szükségünk is lesz a *PTZ* vezérlés megvalósítása során. Ezen kívül természetesen számos más dologra is használhatjuk.

1. Lista A kamerakiszolgáló beépített programjának kiegészítésére készített héjprogram

```
#!/bin/mish
PATH=/bin:/sbin:/usr/bin:/usr/sbin
# stop button pressed? skip the rest.
stopp=`ls /tmp/stopp` ;
if [ "x$stopp" = "x" ]; then
# clear working directory
bufferd -reset -buffername BILDE ;
# read configured positions, '~' signifies
# a unset
# position
grep -v '~' /etc/sysconfig/presetpos.conf \
>/tmp/presets ;
# pos 1-10 reserved for non-public views
num=11;
while [ $num -lt 21 ]; do
# another stop button check - break the loop
stopp=`ls /tmp/stopp` ;
if [ "x$stopp" = "x" ]; then
grep Pos$num /tmp/presets >/tmp/canptz ;
canptz=`cat /tmp/canptz` ;
if [ "x$canptz" != "x" ]; then
# go to position
shttpclient "http://127.0.0.1/axis-cgi/\
com/ptz.cgi?camera=1&gotoserverpresetno=$num";
# wait for picture to settle
sleep 6 ;
# save picture to /tmp/BILDE/
bufferd -start -buffername BILDE -snapshot\
-pre 1 -format snapshot_pos$num.jpg -uri\
'ftp://127.0.0.1/jpg/1/704x576.jpg' ;
bufferd -stop -buffername BILDE ;
tmpwait=20 ;
while [ $tmpwait -gt 0 ]; do
sleep 2;
expr $tmpwait - 2 >/tmp/tmpwait ;
tmpwait=`cat /tmp/tmpwait` ;
if [ $tmpwait -eq 0 ]; then
# timeout creating jpeg, kill process and
# settle for potentially incomplete
# picture
logger "timeout waiting for bufferd
↳-stop,\
killing image_buffer and continuing."
# ps is only available as a builtin
# command in /bin/sh
/bin/sh -c ps >/tmp/kverk ;
grep image_buffer /tmp/kverk >/tmp/
↳kverk2 ;
imbuf_pid=`cut -b 0-5 /tmp/kverk2` ;
rm /tmp/kverk; rm /tmp/kverk2;
kill $imbuf_pid ;
fi;
if [ -f /tmp/BILDE/status ]; then
# status file appeared - picture complete
rm /tmp/BILDE/status ;
tmpwait=0;
fi;
done;
rm /tmp/tmpwait ;
fi;
fi;
expr $num + 1 >/tmp/A ;
num=`cat /tmp/A` ; rm /tmp/A;
done;
rm /tmp/presets ;
rm /tmp/canptz ;
# batch ftp all the pictures from this round
sftpclient -L -m 10.0.0.1 -k /tmp/BILDE -c \
/var/www/pictures/c2 -u web -w P2Ssw1Rd -t \
/var/www/pictures/c2/temp_c2.jpg ;
fi;
```

Jeleket küldhetünk vele egy másik webszervernek vagy kameraszervernek, feltölthetünk képeket **HTTP** protokollon keresztül, de természetesen bármely más műveletet is megvalósíthatunk vele, feltéve, hogy ahhoz az alap hitelesítésen kívül nincs másra szükség. Elvégre ez csupán egy egyszerű **HTTP** ügyfél, nem a **wget**, vagy a **cURL**. Azoknak a problémáknak a többsége, amelyekbe a konkrét megvalósítás során beleütköztünk általában nem a héjprogramozással álltak kapcsolatban. Tapasztaltunk például átmeneti hibákat a képek feltöltésével kapcsolatban. Itt annyi volt a „gubanc”, hogy egy a kertben

található futószőlő indái fölkapaszkodtak a drót nélküli átviteli rendszer egyik antennájára. Ennyit tehát a kameraszerverek héjprogramból történő vezérléséről. Végezetül természetesen meg kell mutassam azt a szkriptet, amivel kiegészítettük a készülék gyári funkcióit (1. Lista). Nem túl elegáns, elismerem, de kicsit és egyszerűen megírható, köszönhetően persze annak a bölcs előrelátásnak, amivel a beágyazott eszköz készítői éltek. Ez az utóbbi amúgy önmagában is olyan lecke, amit bárkinek érdemes megtanulnia, ha ezen a területen dolgozik.

Linux Journal 2006., 142. szám

Erik Inge Bolso

UNIX szakértő és vívó. A Norvégiai Molde-ban él, 1996 óta futtat Linuxot. Másik hobbiját a Google segítségével úgy találhatjuk meg, ha rákeresünk a „balrog genealogy” kifejezésre. Aki írni szeretne neki, az a Ijcomment@tvilsom.org címen érheti el.

KAPCSOLÓDÓ CÍMEK

A cikkhez tartozó egyéb anyagok a következő helyen találhatóak:
www.linuxjournal.com/article/8695