Embedded IP-PBX

Switch Analog and VoIP Calls

David Rowe describes the design of a µClinux-powered IP-PBX capable of switching both analog and VoIP calls. With an Analog Devices Blackfin processor, some custom hardware, and Asterisk PBX software, you can build a similar system.

I have always been fascinated with embedded systems and telephony. I really like working “close to the machine” on embedded systems, and I have a parallel interest in speech and telephony that dates back to my ham radio days as a teenager. One project I had been dreaming about for years is an embedded telephony platform for VoIP or IP-PBX applications.

One problem with telephony is that it requires a lot of processing power. Unfortunately, most embedded processors are not too powerful. So, a common design approach for embedded telephony consists of a general-purpose microcontroller combined with a special-purpose DSP chip to handle the hard-core number crunching. This two-CPU approach increases the system cost and complexity.

About 18 months ago, I stumbled across the Blackfin processor from Analog Devices. The Blackfin is a powerful host processor and a DSP [i.e., it can run µClinux applications and DSP code efficiently and simultaneously on one processor].

A typical embedded processor will struggle to run more than one or two channels of G.729 speech compression or echo cancellation software. The Blackfin, running at 500 MHz, is capable of running 60 channels of G.729 speech compression or echo cancellation. This makes it possible to build low-cost, high-performance embedded telephony systems, like an IP-PBX that can support multiple analog and VoIP channels at the same time. For about $500, you can now build an IP-PBX with features that only $10,000 PBXs had a few years ago.

This work is part of the free telephony project (www.rowetel.com/ucasterisk). The goal of the project is to build open-source telephony hardware and software [e.g., an embedded Asterisk IP-PBX] at a low cost. Both the hardware and software are open source. You are free to copy and reuse the hardware designs.

IP-PBX OVERVIEW

What is an IP-PBX and how does it work? Figure 1 is a block diagram of the embedded IP-PBX that shows the major hardware and software components.

There are two types of analog ports: FXO ports that connect to your local telephone exchange and FXS ports that connect to analog handsets. This jargon can be confusing. The way I remember is the “S” in FXS stands for “station” handset.

The analog ports convert the voice and signaling information to digital signals that the IP-PBX can process. Examples of signaling information are the ring detection and on/off hook status signals. Analog ports are surprisingly difficult to build because they use a mixture of rather old technologies. For example, a ring signal may be 200 Vpp (at low current), on and off hook status is indicated by loop current flowing, and the transmit and receive audio is mixed together on just two wires, which leads to echo problems. All of this must be reliably and safely connected to low-voltage digital systems. Fortunately, there are excellent chipsets available to help build cost-effective and reliable analog interfaces.

Telephone calls can also flow through the Ethernet port using VoIP. Both local [e.g., using SIP phones] and trunked calls can be performed using VoIP. The magic of an IP-PBX is that analog and VoIP calls can be tied together. You can route a call from an analog handset over the Internet to save money on long-distance calls.

What happens when you make a regular analog phone call? First, pick up an analog phone’s handset. This generates an “off-hook” event in the FXS port that tells the Asterisk software to generate a dial tone in your phone. When you dial 9, DSP software decodes the DTMF tones and presents Asterisk with the digit. Asterisk then connects your FXS port to an FXO port so you can reach the local exchange where you can dial a phone number as usual.

What about VoIP calls? Well, say you use the same analog phone to make the call, but this time you dial
long distance. In this case, Asterisk is programmed to route the call over the Internet. To save bandwidth, the DSP software compresses the speech samples from the FXS port from 64 kbps down to 8 kbps. The Asterisk software then puts the speech samples in packets and squirts them out the Ethernet interface onto the Internet.

A lot of the IP-PBX’s power is provided by the Asterisk software, whose primary sponsor is Digium. The ability to run a powerful operating system such as µClinux is also very useful. For example, you can telnet into the IP-PBX while it is running to debug and configure it.

THE HARDWARE

The hardware is built around four PCBs. The first board is an Analog Devices Blackfin STAMP card. It is a development system that is available off-the-shelf from Digi-Key. It contains an Analog Devices ADSP-BF537 Blackfin chip, 64 MB of RAM, 4 MB of flash memory, and connectors that break out many of the ports. It runs µClinux and is supported by www.blackfin.uclinux.org.

One great feature is that an open-hardware/software community exists around the Blackfin. Reference designs for Blackfin hardware (such as the STAMP boards) and various daughter boards are freely available.

On top of the STAMP sits a 4fx daughter board. (Why are boards always girls?) It holds some glue logic and sockets for the modules and also supports an MMC to provide extra flash memory storage. The 4fx’s name comes from the fact that the daughter board can support up to four FXS or FXO modules.

The modules are the little boards that plug into the daughter board. There are two types of modules, FXS and FXO. Photo 1 shows the 4fx daughter board and modules disassembled. There are two modules of each type in the photo. Photo 2a features

![Figure 2](image-url) Take a look at the FXO module schematic. It provides an interface between a telephone line and the Blackfin processor. Its most important task is to provide high-voltage isolation, which is achieved using the capacitive barrier formed by C1 and C2.

![Photo 1](image-url) Here you see the 4fx daughter board and modules before assembly. Each of the four modules at the bottom mate with matching connectors in the middle of the 4fx daughter board. Note the large vacant area in the middle of the PCB. It is necessary to provide physical isolation between the FXO ports and the rest of the system.

![Photo 2a](image-url) The IP-PBX is assembled and configured for four analog ports. From top to bottom, you can see the FXS/FXO modules, the 4fx daughter board, and the BF537 STAMP. There is an optional MMC connector on the left. Green indicates an FXS port, red an FXO.
The schematics for the FXS and FXO modules are shown in Figures 2 and 3. Both modules are based on chipsets from Silicon Laboratories. The circuit designs for the modules are derived from the reference circuits provided in the Silicon Laboratories datasheets.

The most important function of the FXO module is to isolate the “line side” of the port from the low-voltage digital side. There are safety reasons for the isolation. If lightning hits the phone lines, you will want a degree of physical isolation between the phone line and the rest of the hardware. This isolation is rigorously tested during FCC-68 compliance testing by the application of high voltages that simulate lightning. The Silicon Laboratories chips achieve it with a capacitive isolation barrier (C1 and C2). The Silicon Laboratories Si3050/Si3019 chipset used for the FXO module also performs other functions, such as line voltage and current monitoring, impedance matching, A/D and D/A conversion of the speech signal, ring detection, and DC termination.

A screenshot of the PCB layout for the FXO module is posted on the Circuit Cellar FTP site. The PCB measures about 50 × 25 mm. Note the “isolation barrier” between the two chips, which only C1 and C2 are allowed to bridge. The PCB layout was carefully designed to keep the line side and low-voltage side physically isolated by a 3- to 4-mm gap. Even the ground plane is absent in this area, because it would violate the physical isolation requirements.

The FXS module has similar functions to the FXO module (see Figure 3). In this case, the Silicon Laboratories Si3210/Si3201 chipset is used. The main difference with the FXS module is that it generates ring and “battery” DC supply voltages. Silicon Laboratories employs a clever switched-mode power supply that is modulated to generate the 200-Vpp sinusoidal ring voltage. Although the datasheets...
Put your creativity to the test!
(Sep. 20 ~ Jan. 31)

Join the Ethernet Revolution!

Circuit Cellar magazine is pleased to bring you the WIZnet iEthernet Design Contest 2007. Now you can easily add Ethernet capability to your embedded project and win fame and fortune in the process. WIZnet’s W5100 hardwired TCP/IP Ethernet controller will be the key to your contest success. This 3-in-1 chip solution brings TCP/IP implementation without an OS. Both MAC and PHY are embedded.

Show us how you use the impressive W5100 to usher your embedded project into the Ethernet revolution. Your creativity and design skills could win you a share of $15,000 in cash prizes and recognition in Circuit Cellar magazine.

For details, visit www.circuitcellar.com/wiznet.
Figure 4—This is the 4fx daughter board schematic. This board breaks out the TDM and SPI bus signals from the STAMP card and feeds them to each FXS/FXO module. U4 is a Xilinx CPLD that expands the number of SPI chip select signals available from the STAMP.
Renesas Technology

No.1* supplier of microcontrollers in the world

presents a wide range of M16C microcontrollers. M16C is the only fully code-
compatible platform in the industry that addresses the entire 8-bit through 32-
bit price/performance application space.

Your application can range between 24K Bytes and 1M Bytes of code size,
with between 42pins and 144pins of package size, while keeping the same
code base and development tools.

The consistency and compatibility of the M16C Platform enables you to
reduce your development time while still allowing the flexibility to adapt to
changing system requirements.

M16C Product Roadmap

Top Reasons To Select M16C

- **Broad Platform** - Wide range of selection; 24KB to 1MB on-board Flash, 42pin to 144pin
- **Compatibility** - Pin and Code compatibility across the platform enables you to
  upgrade/downgrade MCU in same package without changing the board design and peripherals.
- **Powerful** - High speed interrupts (M16C: 0.61µs/32MHz), optimized instructions for 1cycle
  operations, Hi-speed hardware multiplier.
- **Versatile** - Specialized on-chip peripherals; CAN, LIN, 3ph-PWM, DMA, etc.
- **Efficient** - Multiple clock sources and multiple power saving modes
- **Quiet** - Built-in noise cancellation circuits provides excellent EMI/EMC characteristics that outperforms
  IEEE standard spec.
- **Easy** - Simple Tool Chain for evaluation and development across the board
- **Reliable and secured** - Trusted flash and built-in
  fail safe features such as oscillation-stop detection
  circuit, protect registers, enhanced WDT, etc.

Get Started Today -
Go online and register to be eligible for a FREE Starter Kit
www.america.renesas.com/ReachM16C/d

Renesas Technology Corp.
claim that the DC/DC converter can work well down to 5 V, I found that a 12-V supply was required for the DC/DC converter to avoid excessive analog noise.

The Si3210/Si3201 FXS chipset also takes care of other functions, such as line voltage monitoring, loop current detection, overload protection, impedance matching, and A/D and D/A conversion of the speech signal.

The FXS module’s PCB layout is quite challenging because there are three distinct signal areas in the small 50 × 25 mm area. [A screenshot of the FXS module PCB is posted on the Circuit Cellar FTP site.] The first area handles line-level analog signals from the telephone handset. The second region is a digital interface containing a 2.048-MHz TDM bus and a SPI bus that connects to the fast rise time (i.e., noisy) signals from the Blackfin. The third area is a switch-mode DC/DC converter that converts a low-voltage 12-VDC rail to the –90 VDC required for the telephone “battery” supply and also generates the 200-VPP ring voltage. To generate the high voltages at line voltage monitoring, loop current measuring, and also generates the 200-VPP ring voltage. To generate the high voltages at the Blackfin. The third area is a switch-mode DC/DC converter that converts a low-voltage 12-VDC rail to the –90 VDC required for the telephone “battery” supply and also generates the 200-VPP ring voltage. To generate the high voltages at even small currents, large pulsed currents must flow in the 12-VDC supply line, which is a potential source of noise.

It is important to prevent the DC/DC converter and digital side from injecting noise into the sensitive low-level analog section. Based on tips from the Silicon Laboratories application notes, a few tricks were used. The DC/DC converter ground was kept isolated from the ground plane and connected only at a single point. This prevents large ground current spikes from entering the ground plane. Large current pulses in the ground plane get converted to voltages (because the ground plane impedance is small but nonzero), which then get superimposed on the low-level analog signals as unwanted noise. You can see the lack of a ground plane in the upper-left hand part of the FXS module PCB screenshot on the Circuit Cellar FTP site.

A ground plane was used throughout the analog section and in the digital section. The two ground planes are connected only at a single point to prevent digital currents from flowing through the analog section and inducing noise. This point is as far away from the DC/DC converter as possible.

Bringing the two module designs together was surprisingly straightforward. The first FXO module I built had some problems with clicks and pops in the audio; however, a good cleaning using flux remover fixed that! The flux I use is conductive so any residue tends to upset circuits that depend on series termination resistors. The resistors are initially loaded as 0 Ω, but this can be increased to combat EMI or ringing issues if required. A 74LV244 buffer (U2) is also used to reduce the edge rates of high-speed digital signals from the Blackfin STAMP card. Reducing the edge rates reduces EMI because slow rise and fall times mean reduced high-frequency energy.

**XILINX CPLD FIRMWARE**

On the 4fx card, a Xilinx CPLD is used to expand the number of SPI select lines available and provide a register to drive the LEDs (see Figure 5). The STAMP SPORT connectors have only a limited number of SPI select lines available. However, on the 4fx design, I have many SPI devices (e.g., four telephony ports and a register to drive the LEDs). I would also like to stack the 4fx cards to build eight port telephony systems. So, the problem is accessing multiple SPI devices across multiple cards using only a small number of SPI lines available on the SPORT connector.

### Table 1

<table>
<thead>
<tr>
<th>A2</th>
<th>A1</th>
<th>A0</th>
<th>SPI Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>nCS0: spare</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>nCS1: Port 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>nCS2: Port 2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>nCS3: Port 3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>nCS4: Port 4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>nCS5: LED register</td>
</tr>
</tbody>
</table>

**4fx DAUGHTER BOARD DESIGN**

Now, take a look at the 4fx daughter board schematic (see Figure 4). The FXO and FXS modules communicate with the host processor via two serial buses. The signaling and control data flows via a SPI bus. The actual transmit and receive speech samples flow on a time division multiplexed (TDM) bus.

The 4fx breaks out the SPORT connector from the STAMP board and feeds the SPI and TDM signals to each module. Because the STAMP has only a limited number of SPI chip-select signals available, a Xilinx CPLD (U4) is used to expand the number of SPI devices that can be addressed. This is described in the next section. The CPLD also supports a “stacking” architecture where several boards can be stacked on top of each other to obtain extra analog ports.

To reduce EMI, the suppression components (e.g., ferrites and capacitors) for each port were placed on the daughter board as close as possible to the RJ-11 connectors. Each digital line in the daughter board also has series termination resistors. The resistors are initially loaded as 0 Ω, but this can be increased to combat EMI or ringing issues if required. A 74LV244 buffer (U2) is also used to reduce the edge rates of high-speed digital signals from the Blackfin STAMP card. Reducing the edge rates reduces EMI because slow rise and fall times mean reduced high-frequency energy.

### Table 2

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>LED1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Off</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Red</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Green</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Off</td>
</tr>
</tbody>
</table>
The Newest Semiconductors

Experience Mouser’s time-to-market advantage with no minimums and same-day shipping of the newest products from more than 335 leading suppliers.

www.mouser.com
Over 860,000 Products Online

(800) 346-6873

Mouser Electronics
a tti company

The Newest Products
For Your Newest Designs
I decode a large number of SPI devices using two SPI select lines called nCSB and nCSA. nCSB is asserted and a byte is sent that selects the SPI device you want to address. The value determines the status of each device external to the CPLD. nCSA is asserted to talk to the actual SPI device. nCSA may be asserted as many times as required (e.g., to perform a block transfer on the SPI device).

Note that once the device has been selected, SPI transfers proceed normally (i.e., assert nCSA and read/write as desired). When access to another SPI device is required, nCSB is asserted and the byte sent can be used to select another device (or card).

The LEDs appear as an 8-bit SPI register at address five on the card. When you write to the register, the value determines the status of each LED (see Table 2). And similarly for the other LEDs:

D3[7:6] LED4
D5[5:4] LED3
D4[3:2] LED2

SOFTWARE

The Silicon Laboratories chips are commonly used with Asterisk on PCI-card-based line interface hardware. Open-source drivers for the chips exist, and several Linux-μClinux porting issues. One area that needed optimization was the DTMF detection routines. Figure 6 shows the software architecture of the Asterisk implementation used for the embedded IP-PBX. Asterisk is a user-mode application that communicates with a kernel-mode driver called zaptel. Zaptel communicates with lower-level drivers that talk to the actual hardware.

The Asterisk wcfxs driver was chosen as the starting point for the embedded IP-PBX. The driver was originally written to communicate with the Silicon Laboratories chips operating on a PCI bus. To port the driver to the Blackfin, the PCI interface component was carefully removed and replaced with code that interfaces to the Blackfin’s SPORT (TDM serial port) and SPI hardware.

Surprisingly, it is actually easier (and cheaper) to interface the Silicon Laboratories chips to the Blackfin compared to an x86 PC because no PCI bridge is required. The driver is also simpler.

Most of the software changes were confined to the wcfxs driver, although some changes to the Asterisk application were also required, such as architecture-specific word-alignment issues and several Linux-μClinux porting issues. One area that needed optimization was the DTMF detection routines. The code was written to run on a general-purpose x86 CPU where it is assumed floating-point support is available in the form of a hardware floating-point unit (FPU). The Blackfin does not have an FPU and is in fact optimized for fixed-point operation. Thus, the original float DTMF code ran very slowly, consuming 31 MIPS per channel.

The problem was traced to the “inner loop” of the DTMF detector (see Listing 1). The code implements a digital filter and is called eight times.

```c
static inline void goertzel_sample(goertzel_state_t *s, short sample) {
  float v1;
  float fsamp = sample;
  v1 = s->v2;
  s->v2 = s->v3;
  s->v3 = s->fac * s->v2 - v1 + fsamp;
}
```

```c
#define AMP_SCALE 8
#define FAC_SCALE 14

static inline void goertzel_sample(goertzel_state_t *s, short sample) {
  int mpy;
  v1_fix = s->v2_fix;
  int16_t v1;  
  int mpy;
  s->v2_fix = s->v3_fix;
  mpy = ((int)v1 * (int)s->fac_fix) >> FAC_SCALE ;
  s->v3_fix = mpy - v1_fix + (sample)>>AMP_SCALE;
}
```

---

**Figure 6**—The IP-PBX runs the μClinux operating system and Asterisk PBX software. To run Asterisk on the Blackfin, the DTMF detector was modified to use fixed-point arithmetic. The wcfxs FXS/FXO port device driver was also modified to support the Blackfin serial port and SPI hardware.
for each input sample (eight filters are
required for each DTMF detector).
Because each channel has 8,000 input
samples per second, the total number of
function calls is $8 \times 8,000 \times$ the number
of channels per second. So, it is impor-
tant to make sure the “inner loop” code
runs as efficiently as possible.

The trick is to replace the floating-
point code with equivalent fixed-point
code (see Listing 2). The code maps
directly to the assembler instruction
set of the Blackfin and compiles down
very efficiently. The scale factors were
chosen to keep the dynamic range of
the variables within a range easily rep-resented by a 16-bit integer. The fixed-
point port was tested with a unit test
from the Spandsp library that puts the
DTMF detector through its paces
[www.soft-switch.org]

The result was that the fixed-point
DTMF detector worked just as well as the
floating-point version, but it con-
sumed about 1.75 MIPS compared to the
31 MIPS required for the floating-point
version. Standard vanilla C code was
used (see Listing 2). With a little Black-
fin assembler replacing the C code, the
performance could be improved even
further. However, with 500 MIPS avail-
able on the Blackfin, 1.75 MIPS for the
DTMF decoder is probably fast enough.

GO BUILD

In this article, I explored the archi-
tecture of an embedded IP-PBX and
described in detail the schematic-level
hardware design and PCB layout.
Using embedded techniques and open-
source hardware and software, it is
possible to build a low-cost IP-PBX
with features (VoIP, IVR, and flexibili-
ty) that rival those of $10,000 PBXs.

Other areas of the project that have
not been mentioned due to space limi-
tations include the Asterisk software
configuration, a custom DSP mother-
board and configuration of the PBX. For
more information, visit the project web
site [www.rowetel.com/ucasterisk].

The software and hardware designs
for the project are open-source and
contributions by corporations or indi-
viduals are welcome. Currently, a
team of companies and individuals is
working on the project in areas like
DSP, µClinux software, and hardware
development. Refer to the project web
site for more information.

If you would like to get started with
embedded IP-PBX development, I have
a fully assembled and tested IP04 IP-
PBX for $450 plus shipping. [Q]

Author’s Note: Thanks to the Linux,
Blackfin, Asterisk, and gEDA commu-
nities. My wife Rosemary did a great
deal of the schematic entry for the FXS
module, and Jerry Zeng from Analog
Devices was very helpful in checking
the design and brainstorming the
CPLD requirements for the 4fx.

David Rowe has 20 years of experience
in the development of DSP-based teleph-
ony and satellite communications hard-
ware/software. He has a wide mix of
skills, including software, hardware, and
project management. He earned a Ph.D.
in DSP Theory. David has held execu-
tive-level positions in the satellite commu-
nications industry [www.dspace.com.au]
and has built and successfully exited a
small business [www.voicetronix.com].
However, he has decided that he is better
at debugging machines than people, so
he currently hacks telephony hardware
and software full time.

PROJECT FILES
To download code, go to ftp://ftp.circuit

RESOURCES
D. Rowe, “Free Telephony Project: Open Embedded Telephony,” 2007,
www.rowetel.com/ucasterisk.
S. Underwood, “Spandsp Project,”

SOURCES
Blackfin STAMP
Analog Devices, Inc.
www.analog.com
Si3050/Si3019 and Si3210/Si3201 chipsets
Silicon Laboratories, Inc.
www.silabs.com