

# Asterisk on Embedded systems



Stelios S. Koroneos  
CEO Digital-OPSiS



# About Digital-OPSiS

- Based in Athens, Greece
- Specializes in embedded design & open source solutions
- Participates and contributes in many Open Source projects

# What is an embedded system ?

- Wikipedia

An **embedded system** is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually *embedded* as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems control many of the common devices in use today.

# Benefits of Embedded Systems

- Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

# Characteristics of Embedded Systems

- Architecture
- RAM
- FLASH
- TDM Hardware (analog,digital BRI/PRI)
- External Storage capabilities

# Asterisk Embedded systems

- Available embedded designs types based on architecture used
  - CPU (CISC/RISC based)
  - DSP
  - Hybrid (cpu+dsp, cpu+cpu)
  - ASIC (FPGA based)

# Asterisk Embedded systems

- Available embedded designs types based on software used
  - Linux
    - Uclibc (12 MB footprint)
    - Glibc (30+ MB footprint)
  - Uclinux
  - FreeBSD

# The Good

- Embedded designs are build to target specific tasks/markets
  - SMB or SOHO markets
  - 0-12 TDM lines
  - Up to 30 sip calls
- Low power consumption
  - 10-30W based on design and number of lines
  - Green footprint = lower electricity costs=lower Total Ownership Cost
- Small size / Appliance “look”
  - No problems with customers “hey this PBX looks like my PC”
- Reduced cost

# The Bad

- Most embedded systems are < 1Ghz
- Storage is limited in size and write endurance
- Development complexity
- Component/ board /material sourcing
- Certain applications are not suited for use in an embedded design

# The Ugly

- Developing software for an appliance requires cross-compiling (most of the times)
- Changing platforms can bring to the surface obscure bugs
- Endianness or alignment / access issues
- Low level drivers for the TDM hardware might be required
- Development time is longer and costs are higher

# Things to watch

- Not all CPU cores created equal
- RISC faster than CISC (same clock,same RAM type) but not for all tasks \*
- Asterisk is heavily threaded. MMU is a must
- RAM Memory type  
    DDR is 2 to 6 times faster than SDRAM
- Asterisk requires media processing capabilities (echo cancellation, codecs)
- FPU or DSP capabilities will make things go faster

# Things to watch

- Optimization
  - Avoid the “One size fits all” mentality
  - Gcc is not a speed demon (unless you tweak both the code and its command line switches)
  - Check Asterisk Makefiles to enable some optimization
- Embedded does not always mean “low-cost”

# The little, dirty (tech) details

- Cache sizes are small (10K-100K)  
Cache thrashing is an issue with Asterisk
- Cache controllers are rather “low-end”  
no dma snooping
- FPU's (if present) are usually not very efficient
- Data size and alignment
- Increased network irq traffic due to small packet size  
irq switching time plus cache thrashing
- Network is BIG ENDIAN

# Selecting the hardware core

- Market (i.e currently deployed appliances) use mostly DSP
- My preference PowerPC
  - AMCC 405 or AMCC 440
  - Fast RISC core with good switching, mmu
  - Big Endian
  - Has DSP instructions
    - 32 bit Multiply Accumulate in 1 cycle
  - Reference Designs available with fxs ports

# Selecting a build system

- Most existing distro's are based on BuildRoot
- DSP's can use ONLY uClinux (no mmu)
- My preference is OpenEmbedded
  - Targets both glibc and uclibc
  - Multi-platform support (x86,arm,powerpc,mips,xscale,sh)
  - Package Management system

# Which Asterisk version ?

- Asterisk 1.2
  - Stable can be tricky in cross-compiling
  - Needs patches for uclibc targets
- Asterisk 1.4
  - Easier to cross-compile
  - Smaller memory footprint
  - Stability problems due to large code base changes.
- Asterisk 1.6
  - New beast , still testing

# Embedded GUI

- Common Web interfaces use LAMP (Linux Apache Php)
- Embedded targets need lightweight web interface (shell script, binary for cgi, web server is usually busybox's http or boa)
- Perl, PHP are too resource intensive (both in size and cpu)
- There is no “silver bullet” when it comes to GUI

D I G I T A L  
O P S I S



**Questions ?**