

Power Development on Embedded Systems

19 Apr 2007

Phil Brownfield, Freescale Semiconductor

Native versus cross program development

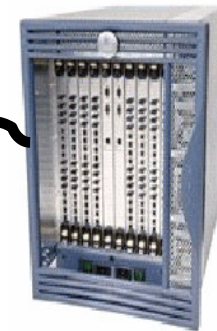
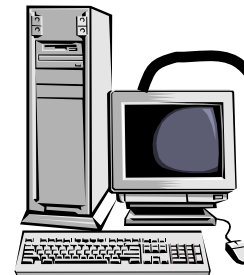
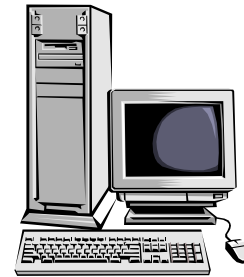
Native, or "self-hosted", development creates, runs and debugs programs on the system where they will run

Cross development creates programs on a "host" system, and runs and debugs them on a different "target" system. Targets are typically the end system hardware

Linux ("GNU/Linux" if you prefer) and *BSD are the only embedded Power Architecture OSes supporting self-hosted development today

Cross development is necessary with other embedded OSes in the target system

Typical cross development host systems for embedded Power Architecture run Windows, Solaris or x86 Linux

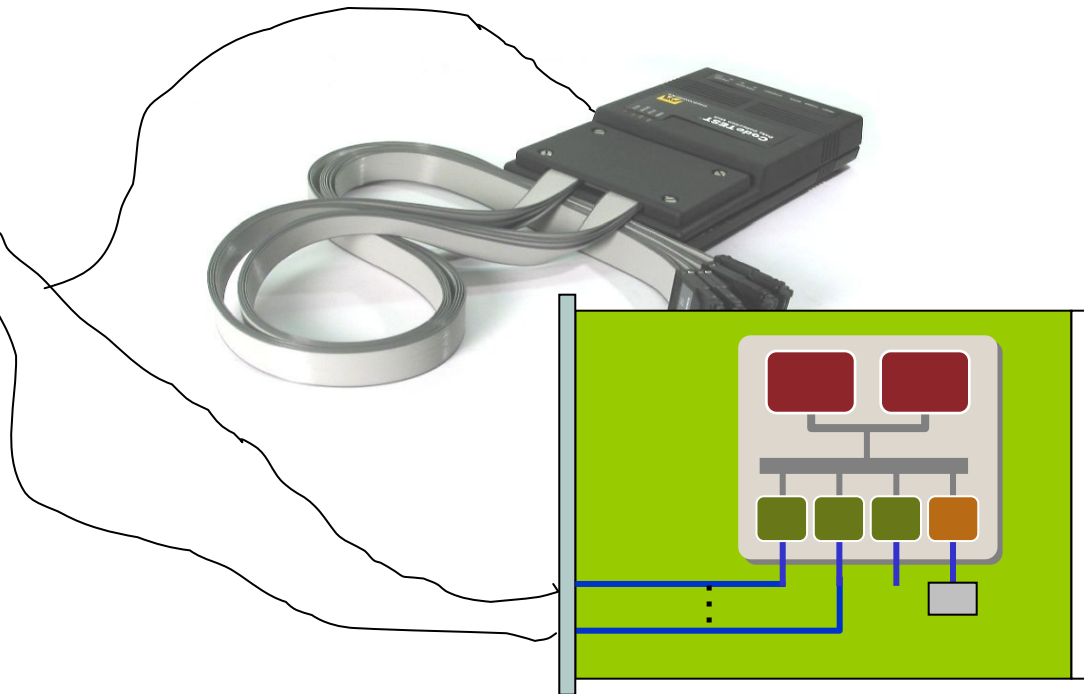


A typical embedded cross development scenario



Host connected to target via serial, ethernet
Run control, trace probes allow non-intrusive target
debugging from IDE/debugger running on host

Target boots by
downloading OS from
host via probe or via
firmware ftp, tftp
May network mount
filesystems resident on
host



Recent Freescale reference system: MPC8349e-mITX



CPU

- MPC8349E TBGA @ 333/667 MHz

Memory subsystem

- DDR-1 Memory Module (default is 256M)
- 16-MByte Flash (MX29LV640BT)
- Compact Flash (type I, 3.3mm, true IDE mode)

Interfaces

- USB
 - 4x USB 2.0 Hub (GL850A)
 - 1x USB 2.0 OTG (USB3300)
- Ethernet ports
 - TSEC 1: connects to Gbe PHY VSC8201 10/100/1000 BaseT copper
 - TSEC 2: connected to 5 ports Gbe layer-2 switch VSC7385
- Connectors
 - 16-bit Local Bus expansion connector, 9-bits addressing
 - GPIO for external 4x 20 LCD module
- On-board PCI 4-port SATA controller SiI3114 (PCI1)
- PCI expansion (PCI2)
 - 1x 32-bit 3.3V PCI slot
 - 1x 32-bit 3.3V mini-PCI slot
- JTAG / COP, 2x RS-232C
- ATX 2x10 power connector

Serial EEPROM and RTC

MCU for soft power up/down, fan/LED control, temperature sensing

Form Factor : mini-ITX (170mmx170mm)

Some strengths of self-hosted, cross development

Tradeoffs are not cut-and-dried, but here are some things to consider

Self-hosted development strengths apply when

- The end product runs the same OS as on the developer desktop
- Software development focuses on application software layers
- Debug tools can depend on a stable OS platform running underneath
- Bugs are unlikely to be hidden when debug tools are co-resident with apps

Cross-development strengths apply when

- The end product runs a different OS than on the developer desktop
- Software development focuses on OS, driver, protocol stack layers
- There is not yet a stable OS platform running
- Heterogeneous asymmetric multiprocessing (Linux + RTOS) in end product
- Bugs may be masked by co-resident debuggers
- Can utilize software tools not ported to target OS / processor

Power.ORG ™