Experiences With RTEMS

Till Straumann, SSRL/SLAC
<strauman@slac.stanford.edu>

Overview

□ What is RTEMS

□ Advantages, Shortcomings

□ Experience

□ IOC Example

□ Questions

What is RTEMS

- Modular, hard-RTOS kernel. "Library", linked to app.
- Add-ons
  - CPU support (i386, PPC, 68k, ARM, sparc, mips, ...)
  - BSPs
  - newlibc
  - BSD TCP/IP
  - TFTPfs
  - rdbg (remote debugger stub)
- GNU toolchain

Advantages, Shortcomings

- △ Good performance
- △ Open source
- △ No fees
- △ History > 10y; supportive user community (mailing list)
- △ Maintenance
- △ Supported by EPICS

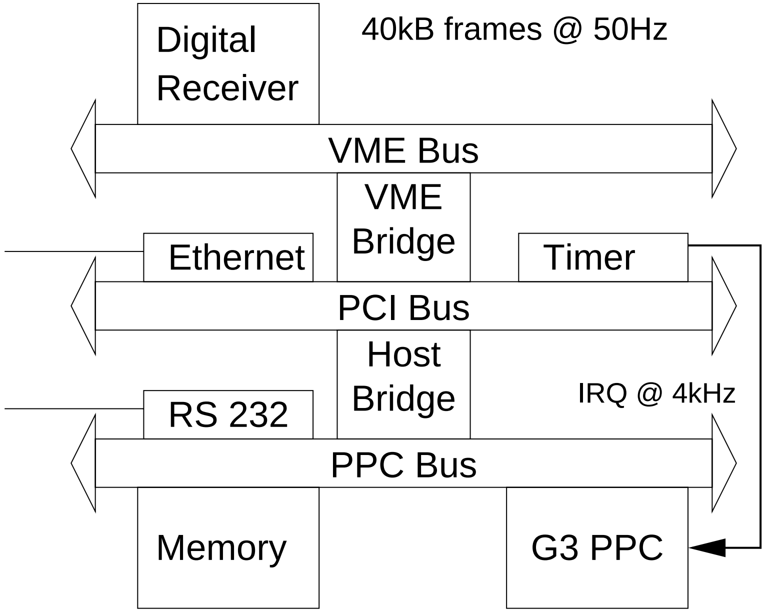
- ▽ Critical mass (of EPICS users) not yet reached
- ▽ Less BSPs available
- ▽ No standardized API for lowlevel resource management and basic BSP services (IRQ, address probe/mapping, VME, PCI)
- ▽ No NFS (for EPICS not very important, has TFTPfs)
- ▽ No shell (but: iocsh, monitor, cexp)

Experience

- Development of RTEMS applications and drivers
- Study / comparison of latency behavior (RTEMS/vxWorks/RTLinux)
- Development of BSP for Synergy VGM series PPC SBC
- Kernel internals
- Port of EPICS to VGM BSP (easy, thanks folks)
- EPICS Applications on RTEMS
- Implemented "devLib" for RTEMS (relies on BSP)
- Fourteenified drvIpac, CAN bus support; runs on RTEMS (< 2days)
- Lack of shell: here comes Cexp

IOC Example

- □ IOC controls digital ADC / receiver VME card
- □ 40kB @ 50Hz bursts over VME/PCI -> memory
- □ Network communication
- □ Serial port communication
- □ Timer interrupts CPU @ 4kHz
- □ ISR schedules max priority task
- □ Measurement of IRQ and dispatching latencies



Demo