

# devLib2

## PCI and VME64x extensions for devLib

Michael Davidsaver  
mdavidsaver@bnl.gov

EPICS Collaboration Meeting Fall 2010

<http://epics.sf.net/devlib2>

# What is it?

- An extension of devLib from EPICS Base (v1)
- V1
  - Provides OS independent VME access
- V2
  - Adds helper library for VME64/64x devices
  - New OSI interface for PCI devices (RTEMS, vxWorks and Linux)
  - MMIO macros

# VME64/64x

- New address modifier (CSR)
- Each slot gets an range
  - Geographic addressing
- Standard registers
  - Identification: vendor/model
  - Control status register
  - Address decode registers (64x only)
    - Similar to PCI's BARs

# devLib2 VME64/64x

- Provides macros with offsets of standard registers
- Read/Write macros
- Slot probing
  - Can use identifier list (like PCI)
  - Wildcards supported
- IOCsh diagnostics
  - vmecsrdump(int)
    - Probes all slots and prints ID info.
  - vmeread(int,int,int)
    - vmeread(0x200010e3, 32, 32) # read A32 D32

# PCI in brief

- Supports full automatic configuration!
  - Can be an issue w/ interrupt priority
- Hierarchical bus (PCI-PCI bridges)
- OS support nearly universal
- Several physical form factors: “classic”, cPCI, PMC, miniPCI, PCI-104
- Obsolete?
  - PCI express

# devLib2 PCI

- Allows probing or searching based on identifier lists (w/ wildcards).
- API for BAR mapping, IRQ setup
- Supports RTEMS, vxWorks, and Linux
- Multiple implementations
  - Selectable at runtime
- IOCsh
  - `devPCIShow(verb, vendor, device)`

# Linux UIO

- Linux Userspace I/O framework
  - Added in 2.6.12 (circa 2005)
- Minimal kernel space driver
  - Maps I/O memory into userspace process address space (boiler plate)
  - Handles interrupts
    - Silence IRQ so userspace task can service
- Access via `/dev/uio#`
  - Appropriate permissions needed

# UIO w/ devLib2

- devLib2 UIO implementation can search for PCI devices on Linux systems (via. `/proc/bus/pci/devices`)
- Addition kernel module needed to map BARs and connect interrupts.
  - Manually create `/dev/uio#` (unless `udev` is running)
  - Kernel source contains examples
    - See `drivers/uio/uio_cif.c`

# MMIO Access

- IO memory (not x86 I/O ports) is different then main memory.
- Optimizations dangerous:
  - Reordering
  - Width
  - Combining/splitting
- CPU Architecture and OS specific

# MMIO w/ devLib2

- `be_ioread32(ptr)`, `le_iowrite16(ptr, val)`
  - Conditional byte swap for PCI
- `nat_ioread16(ptr)`, `nat_iowrite16(ptr, bswap16(V))`
  - Unconditional byte swap for VME
- Defaults to volatile void\*
- Specializations:
  - RTEMS on powerpc
  - VxWorks (`vxAtomicLib`)

# Status

- First release available
- Will propose for Base 3.15
- <http://epics.sf.net/devlib2>