GPIB Device Control with COSYLAB microIOC

- COSYLAB microIOC
- Development for microIOC with SDK
- GPIB device control
- microIOCs in SLAC
- Things learned

Jingchen Zhou at SLAC

COSYLAB microIOC

- microIOC
  - A compact embedded computer for control and monitoring of devices via network
  - A bridge integrating devices with EPICS based control system
    - devices connected to microIOC via serial, GPIB or other ports
    - microIOC (EPICS applications run) available to control system via Ethernet network
- Key features
  - Extensible I/O: Serial (RS232/RS485), GPIB and other interfaces
  - Powered by Debian Linux
  - EPICS enabled
  - SDK for microIOC (develop/build/deploy/console)
  - Compact Flash (CF), no disk, no fan

COSYLAB microIOC

Front side: COM1, LAN1 (Static IP) and LAN2 (DHCP), CF (Linux and EPICS)

Rear side: GPIB port (or Serial) , Controlled Line Power OUT1 & 2 (special hard reset for GPIB devices)

microIOC Software Development Kit (SDK)

- SDK comes with microIOC, Linux based
- Development platform for microIOC
- SDK contains
  - all the libraries (EPICS, Sequencer, ASYN and etc.), compilers and tools required to create/build EPICS control solutions
- Everything (!!!) within the SDK
  - login to SDK with uioc-sdk
  - create a new EPICS project with e.g. makeBaseApp.pl and make
  - set UIOC_TARGET to microIOC's IP
  - uioc-deploy (to package all the required files constituting a solution, distribute over to microIOC , unpack and restart the IOC)
  - uioc-console to access the IOC console
- Backup deployment (just take new CF to microIOC)

```
WELCOME TO THE
                                        #*#   #######  #####
#   #    #    ####   #####   ####      #    #     # #     #
##  ##   #   #    #  #    #  #    #    #    #     # #
# ## #   #   #       #    #  #    #    #    #     # #
#   #    #   #       #####   #    #    #    #     # #
#   #    #   #    #  #    #  #    #    #    #     # #     #
#   #    #    ####   #    #   ####    ###   #######  #####

      SOFTWARE DEVELOPMENT KIT v1.0.2
                        TARGET : epics-2.0.2

  +---+   +---+   +---+   +---+   +---+
  |   |   |   |   |   |   |   |   |   |   Copyright (c) 2005 by Cosylab
---+   +---+   +---+   +---+   +---+   +-------------------------------

Commands: make ; exit ; help ; uioc-deploy ; uioc-console ; uioc-update ; ...

Kerberos Password:
uioc:epics$
```

A GPIB Application

- Create a GPIB device support module
  - makeSupport.pl -t devGpib uiocGPIB
  - using ASYN driver
  - COMMAND, QUERY, RESPONSE supported
- Create a GPIB IOC application to control DVM
  - makeBaseApp.pl -t ioc dvm; makeBaseApp.pl -i -t ioc dvm
  - in dvmInclude.dbd

    include "drvLinuxGpib.dbd”

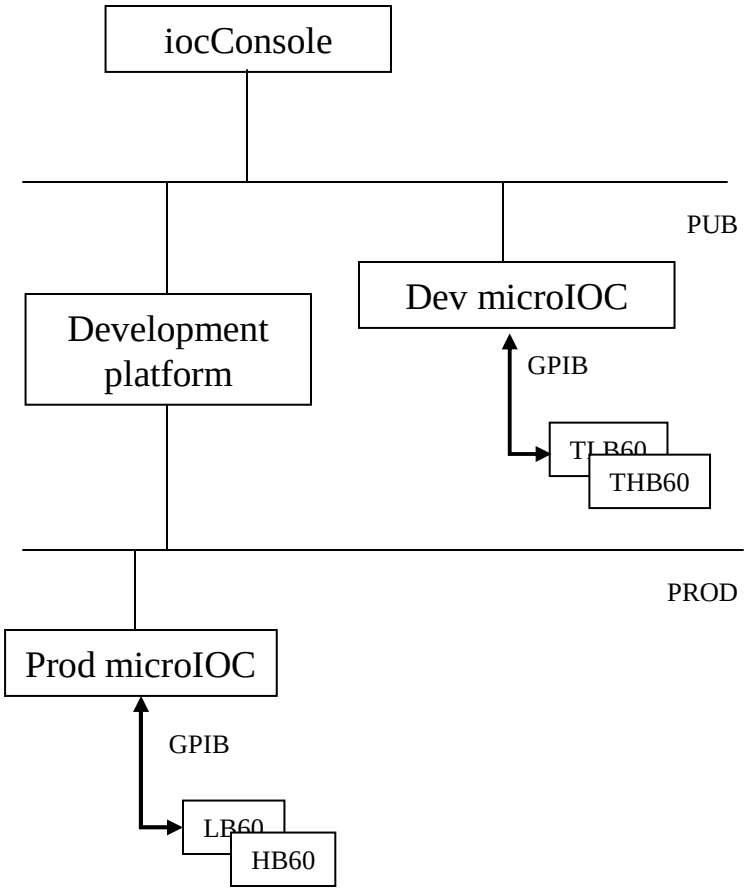
    include "devuiocGPIB.dbd“

    include "uIOCLcdSupport.dbd"
  - setup for GPIB port (in st.cmd)

    GpibBoardDriverConfig("L0",1,0,3,0)
  - define flat database
  - sequence program: to read, monitor, and reset automatically (power-cycle and reinitialize)
- Reference: “How to create EPICS device support for a simple serial or GPIB device” by Eric Norum

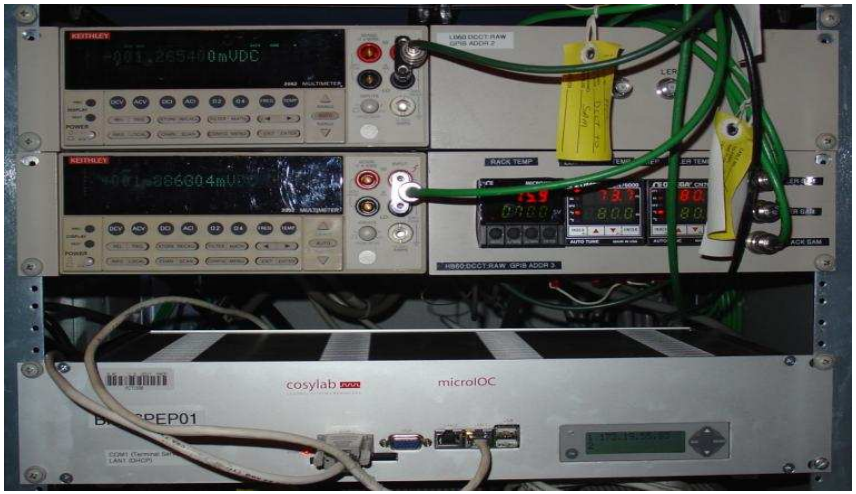
microIOC Setup

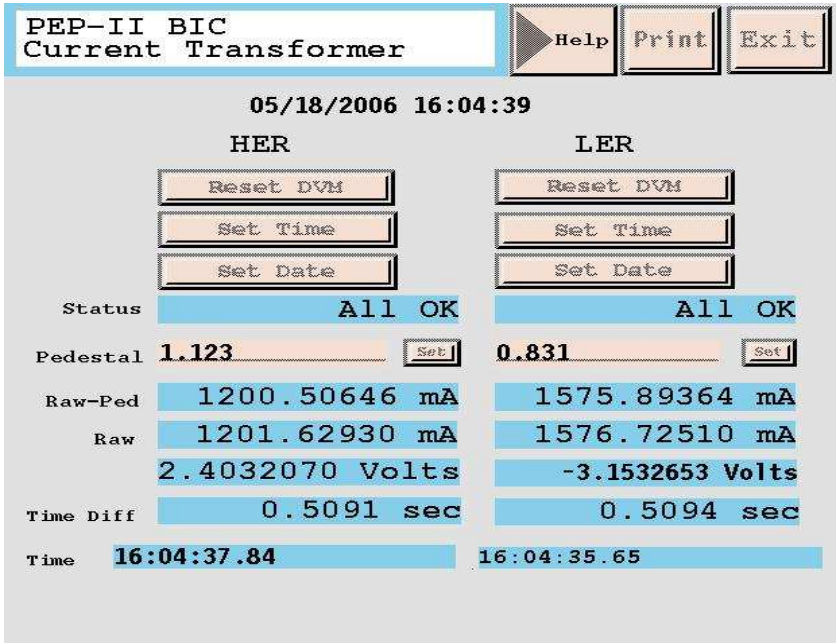
- Development platform
  - Debian Linux recommended
  - RHEL3 tested in SLAC
  - sudo needed to allow login to SDK
  - dual homed: one on public network, one on production network
- SDK and setup backed up to CVS
- IOC Console access with EPICS Extension iocConsole
  - no sudo required
  - access to microIOC host (via terminal server)
  - access to the soft IOC (via telnet microIOC port)
  - access from anywhere
- Production and Development microIOCs
  - SDK can access both
  - DHCP used
  - Quick test/fix, and easy failover (swap of CF)



microIOCs in SLAC

- Two microIOCs for PEPII Bunch Injection GPIB controls
  - To read the DCCT (total ring current) via two Keithley Multimeters (DVM) one for LER, one for HER
  - One microIOC for PROD, one for DEV (backup)
- Two more microIOCs coming
  - Two PCs (DOS) which communicate with the Control Room Knob boxes via RS485





Things Learned

- Easy integration of devices with the rest of EPICS control system
- Easy development and deployment with SDK
- Quick failover
- Robust!!!
- Pitfalls
  - only one soft IOC per microIOC supported
    - multiple IOCs can disable SDK and hang micoIOC
    - login microIOC to stop and remove unwanted soft IOC manually
  - hard to integrate with the rest of development/release environment
  - cumbersome in EPICS upgrade
    - upgrading or adding module in SDK: COSYLAB
    - upgrading in microIOC sometimes manually, can be out of sync
- Thanks to Gaspar Jansa and Klemen Zagar at COSYLAB for their excellent support!