

USING EPICS ENABLED INDUSTRIAL HARDWARE FOR UPGRADING CONTROL SYSTEMS

E. Björklund, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

A. Veeramani, T. DeBelle, National Instruments, Austin, TX 78759, USA

Abstract

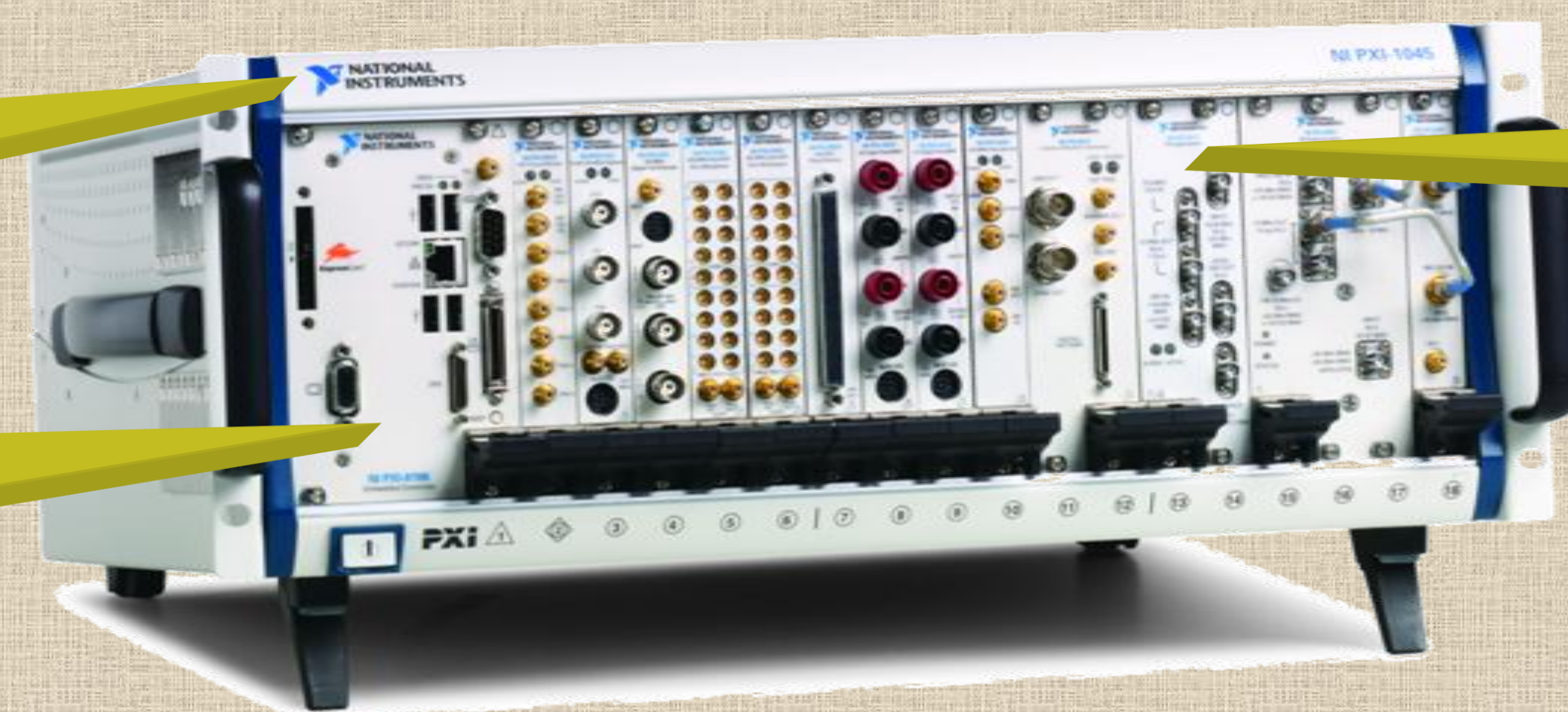
Los Alamos National Laboratory has been working with National Instruments (NI) and Cosylab to implement EPICS Input Output Controller (IOC) software that runs directly on NI CompactRIO Real Time Controller (RTC) and communicates with NI LabVIEW through a shared memory interface. In this presentation, we will discuss our current progress in upgrading the control system at the Los Alamos Neutron Science Centre (LANSCE) and what we have learned about integrating CompactRIO into large experimental physics facilities. We will also discuss the implications of using Channel Access Server for LabVIEW which will enable more commercial hardware platforms to be used in upgrading existing facilities or in commissioning new ones.

Two Options

EPICS Channel Access Server

- Channel Access Interface Only
- Record Fields Available
- Available on all LabVIEW platforms
- Simple to Set Up
- VAL, TIME, STAT, SEVR, HI, LOW, HIHI, LOLO

Any Commercial Hardware

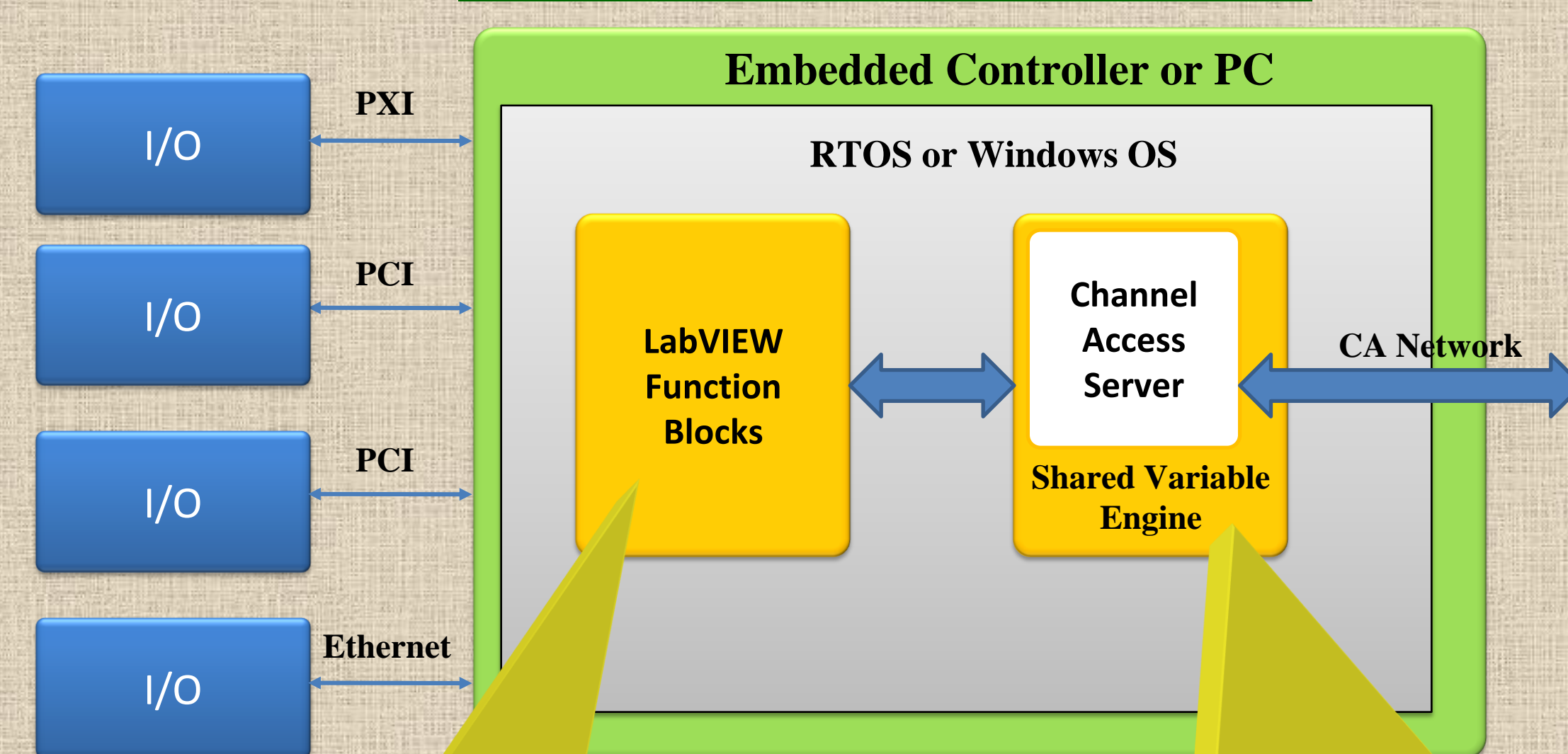


PXI Chassis with Timing and Synchronization

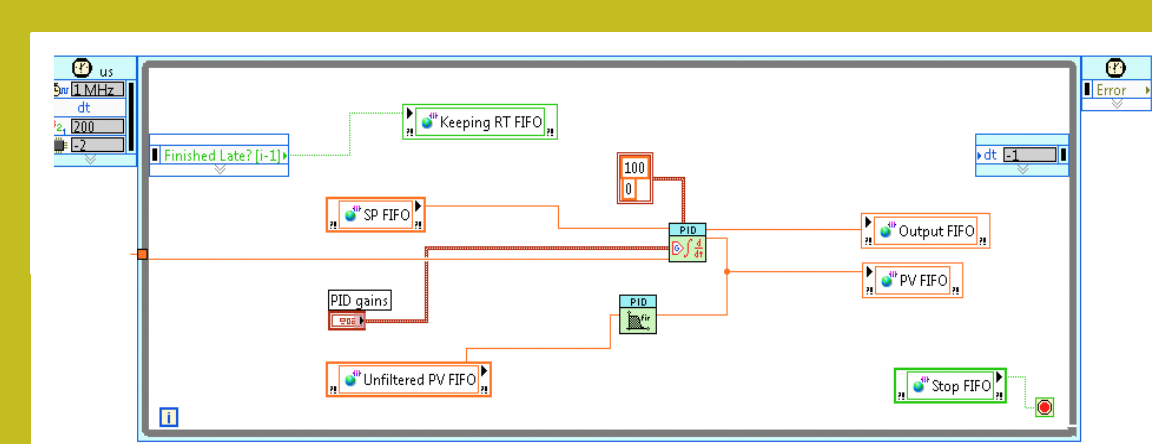
Embedded Controller with Windows, RTOS or Linux

I/O Cards Plug Into PXI Backplane

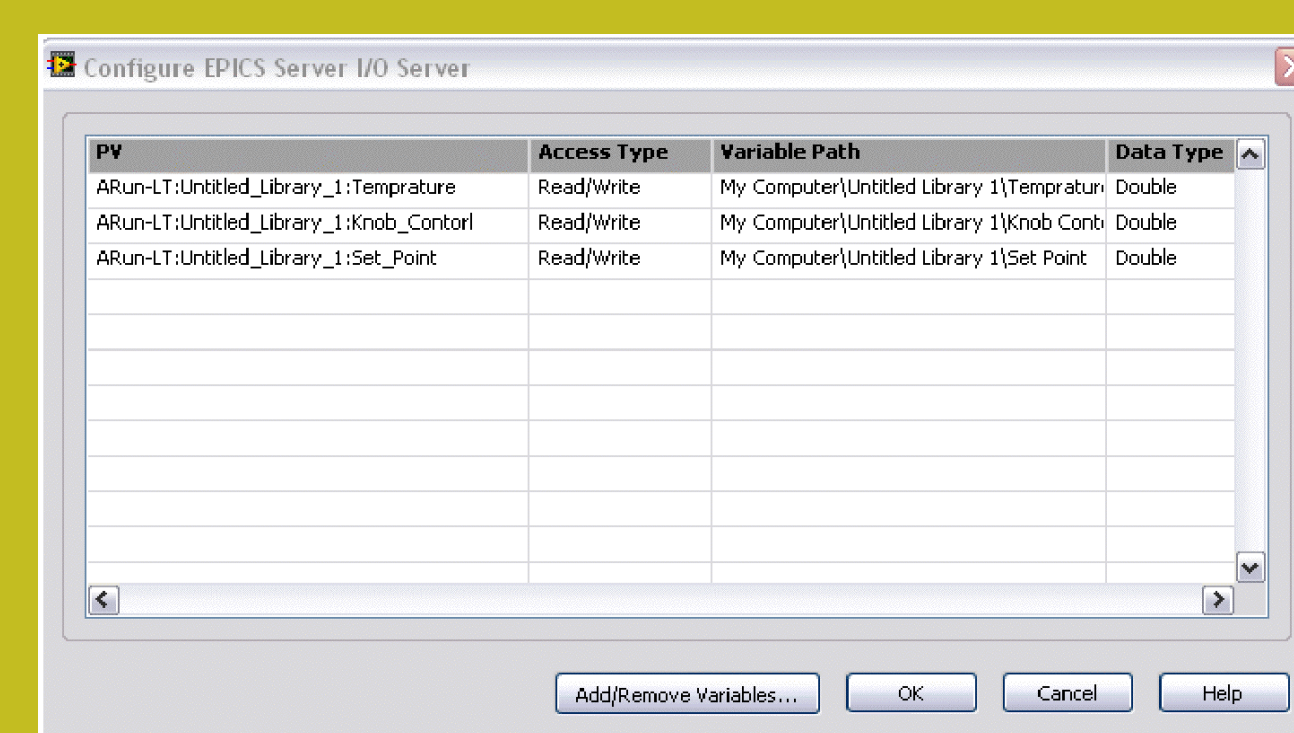
Software Architecture



LabVIEW



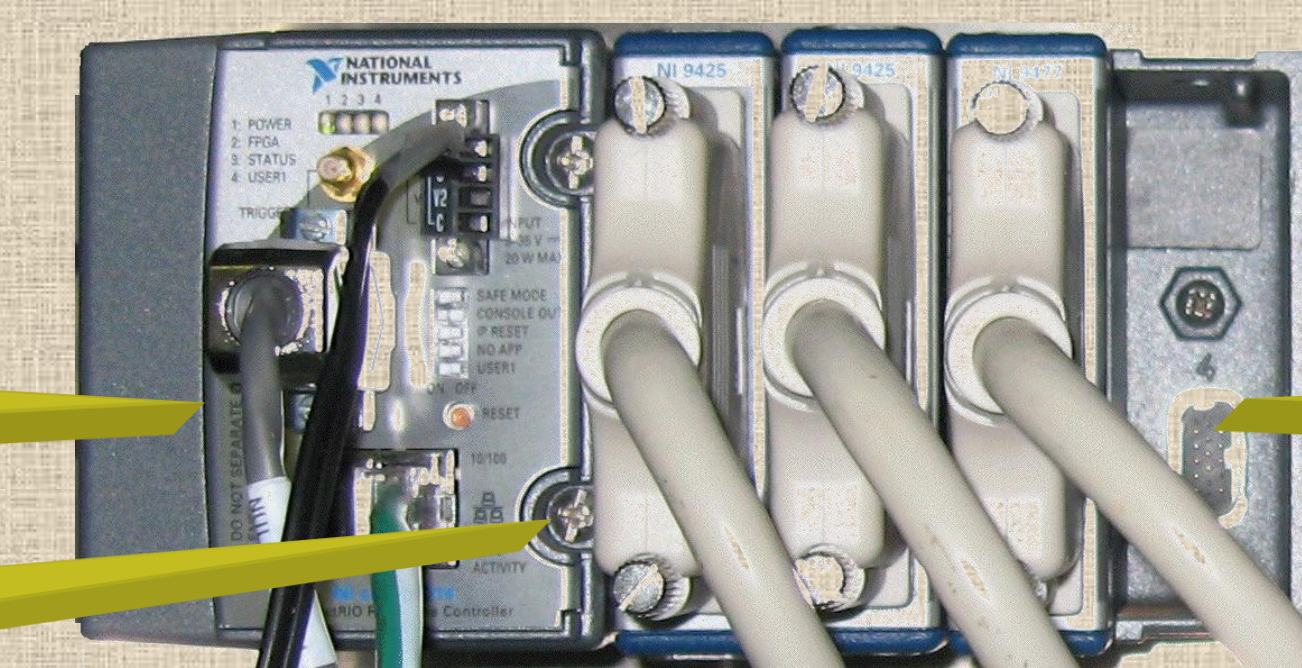
CA Server in LabVIEW 2009



EPICS On CompactRIO

- Full EPICS Record Processing
- Full EPICS Utilities
- Available on CompactRIO
- More Complex to Set Up
- Save/Restore
- All Record Fields Are Available
- Sequencer

Hardware Architecture

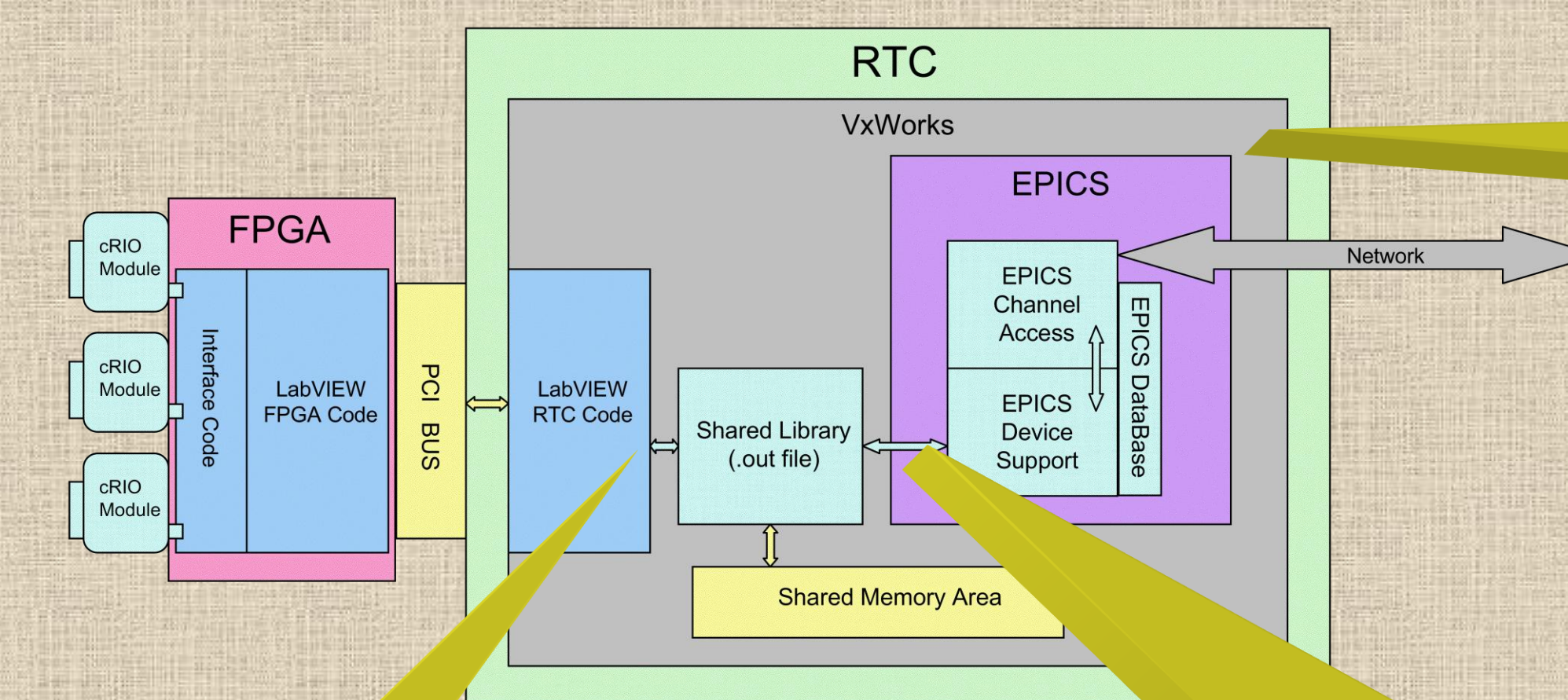


Power-PC Based Real Time Controller

PCI Bus Interfaces RTC and FPGA

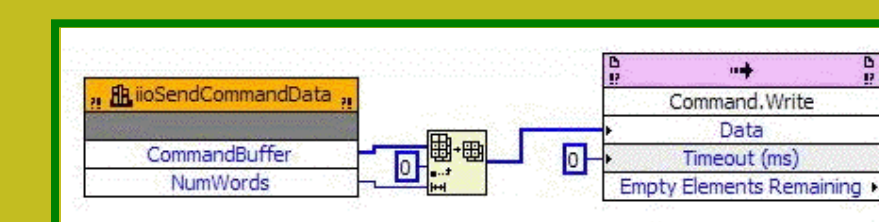
I/O Cards Plug Into FPGA Backplane

Software Architecture



EPICS-Enabled Board Support Package

LabVIEW Interface Call Library Function Node VI



EPICS Interface EPICS Device Support

```

LOCAL_RTC
epicsStatus aIRead (aIRecord *pRec)
{
    /* Local variables */
    /* Pointer to device private struct for this record */
    iio_AIDevPvt *pDevPvt; /* Local status variable */
    epicsStatus status; /* Calibrated voltage (not used) */
    epicsFloat64 Volts;

    /* Get the device private area, then get the latest value from the FPGA VI */
    pDevPvt = iio_AIDevPvt (*pRec->dpvt);
    status = iioReadCalibratedAnalog (
        pDevPvt->pSlot, /* Pointer to slot structure */
        pRec->inp.value.vmeio.signal, /* Signal number */
        &Volts, /* Calibrated voltage */
        &pRec->rval); /* Calibrated raw value */
}
    
```