

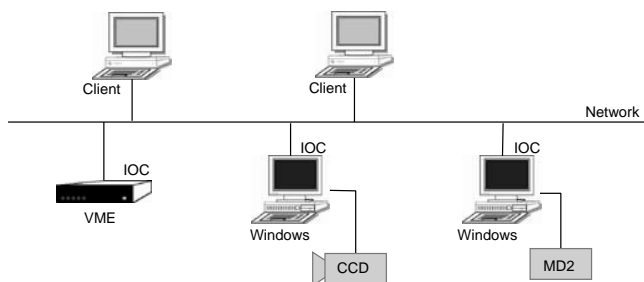
## USING WINDOWS PC AS EPICS IOC

R. Krempaská, A. Bertrand, M. Grunder, D. Vermeulen

*The beamline scientists often require special instrumentation to be controlled remotely and very often controlled along with standard beamline components. This article describes basic ideas and rules how to build EPICS Input Output Controller (IOC) applications for controlling special instrumentation hardware running under Windows PC.*

### INTRODUCTION

A number of instruments used at SLS, especially at the beamlines, is controlled by software running on Windows PC. These can be CCD (Charged Coupled Device) cameras, microdiffractometers, electron analysers, etc... Because the SLS control system software is based on EPICS distributed architecture, the need to control these devices within EPICS is obvious. Since EPICS release 3.14, it is possible to run an EPICS server, called Input/Output COntroller (IOC), on many platforms like Win32 or Linux [1] (earlier EPICS releases are available only under the VxWorks operating system running on VME based systems). For the fact the EPICS communication protocol (called Channel Access, or CA protocol) is platform independent, it is possible to use EPICS for heterogenous client-server systems as schematically depicted in Fig.1.



**Fig. 1:** A heterogenous client-server distributed architecture which uses the EPICS CA protocol for remote control of the beamline components (controlled via VME IOC) and special instrumentation near the experimental stations controlled with Windows PC

### DEVICE SUPPORT DEVELOPMENT ON WINDOWS

In order to integrate an instrument with a programming interface running on Windows, a new EPICS module called device support has to be implemented. A device support is the interface between specific hardware (driver) and EPICS records – basic bricks of an EPICS control system. One can say that EPICS records, conveniently grouped in template files, separate hardware specifications from functionality issues (for example an analog output record called SET\_TIME with device type (DTYP) AO\_DEV shall set an integration time of a microdiffractometer).

The device support is not included in the EPICS base software (so called iocCore), but is kept separately in Windows Dynamically Linked Libraries (DLL). When a

Windows EPICS iocCore starts, it loads the required device support DLLs. Much effort has been spent to simplify and unify implementation of Windows based EPICS devices. A fixed directory structure containing sources, Makefiles and libraries is provided. Typically a new device support is made by adding functions to initialize, read and write values to specific hardware.

### IOC ON WINDOWS

At the end a Windows EPICS IOC application is a single executable file. It performs initialization by executing statements residing in common startup scripts. The developer only has to write EPICS template and substitution files in which some of the EPICS records will have device type (DTYP) fields referring for example AO\_DEV. No further compilation is needed. In this context a new device type implementation is very similar to VME based systems, where an EPICS release is decoupled from device support. The next section illustrates a Windows based IOC application.

### MICRODIFFRACTOMETER

In order to control a microdiffractometer delivered by a private company [2] on the Protein Crystallography beamline, we implemented an IOC application using microdiffractometer device support. A user controls this instrument by using EPICS process variables which subsequently call functions of an underlying software supplied by a private company.

### CONCLUSIONS

Many special instrumentation software used at beamlines is running under Windows PC. It turns out, that the beamline scientists need to control these devices along with other beamline components (monochromator, valves, motor motions etc...) being controlled via EPICS CA protocol. The application program interface to implement a Windows based EPICS device presented here is an optimal solution to this problem. Once a device is running under EPICS, it can benefit from remote control and also from a wide variety of EPICS extension applications like archiving, graphical user interfaces and alarm handling.

### REFERENCES

- [1] M.Kraimer et al.: EPICS: Input/Output Controller Application Developer's Guide Release 3.14.1
- [2] <http://www.maatel.fr/>