WDOOCS::PORTING DOOCS TO WINDOWS PC

V.Kocharyan, YerPhI, Yerevan, Armenia
K. Rehlich, DESY, Hamburg, Germany

Abstract

The Distributed Object Oriented Control System (DOOCS) is used at the TTF VUV-FEL as a multiplatform control system that integrates the contributions of the collaboration. The communication in DOOCS is based on Open Network Computing/Remote Procedure Calls (ONC/RPC) with eXternal Data Representation (XDR) network format from SUN. This communication protocol was ported to Windows operating system. This paper describes the Windows version of DOOCS (WDOOCS) in detail and presents new hardware resources, which were also employed within control system in the current implementation. Integration of the MATLAB and LABVIEW software into the TTF VUV-FEL control system requires an interface to DOOCS that has been developed for Windows, UNIX and Linux operating systems. Several DOOCS servers for VISA oscilloscopes, CAN-Bus PCI boards and etc. are utilizing WDOOCS.

INTRODUCTION

DOOCS is a distributed control system that primordial was designed to run mainly on Solaris, SunOS and PC's with the LINUX operating system [1]. The whole system is written in the programming language C++ and consists of two parts, namely client and server libraries.

Different and incompatible versions of RPC exist for both UNIX and Microsoft Windows operating systems. On UNIX, it is common to have both ONC RPC and DCE (Distributed Computing Environment) RPC. Although these RPCs perform similar functions, their implementations and network marshalling specifications are different. ONC RPC uses eXternal Data Representation (XDR) for network marshalling, whereas DCE RPC uses Network Data Representation (NDR) for marshalling. On Windows, there is natively implemented Microsoft RPC, which also uses NDR.

WDOOCS STUCTURE

Sun Microsystems's ONC/RPC library C source code is freely distributed for the user development. This communication protocol has been ported into Microsoft Windows operating system as MS Visual C++ dynamic linked library project. The core of the project is *wsock32.lib* library methods. The *sunrpc.lib* reference library is used for other MS Visual C++ application projects. The *sunrpc.dll* is located in Winows\system32 directory.

Two client API libraries *EqClient.dll* and *DoocsAPI.dll* were adapted on MS Visual C++ code level. The first one supports DOOCS protocol only and is usually used for server applications. The second one supports DOOCS and TINE protocols and is used to create client MS Visual C++ projects. DOOCS client library contains three main classes (*EqCall*, *EqAdr*, *EqData*) and internal extra frames to collect the RPC requests in a local database – (*EqServices*). The *EqCall* class is the client interface to the RPC communication. The authorization problem for *set* requests from the non-UNIX environment, in current version, has been solved via a special authorization request to the UNIX name server. The request contains host and user name strings. The server returns an integer array with UNIX authorization information. The server response information can be also configured by means of the server configuration file, which allows establishing correspondence with the Window and UNIX users account.

In contrast to UNIX shared server library the Windows *serverlib.lib* is static library, but it's fully identical to the DOOCS data types and the server library methods [2]. One of components of SUN ONC RPC server is a *portmapper*, which has been done as Windows services (background application). The *portmapper* starts automatically on the computers where DOOCS servers are supposed to run.

The flexibility of the C++ DOOCS libraries source codes allows compiling them with the MS Embedded Visual C++. They have been tested in addition on Windows CE 3.0 platform in the Pocket PC.

WINDOWS DOOCS CLIENTS

Above [3,4], on the UNIX client application side the TTF VUV-FEL control system provides a set of generic and specialized programs and incorporates commercial products. All of these programs use the same client API calls to access device data. Some of them have been ported to Windows operating system.

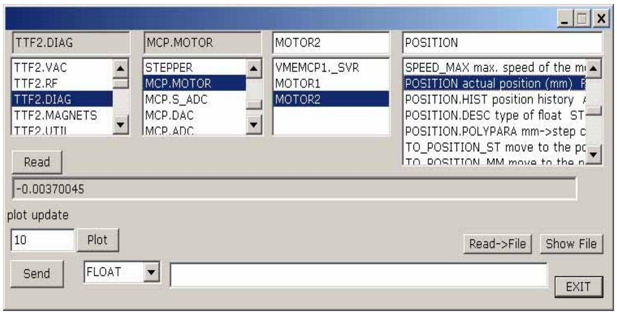
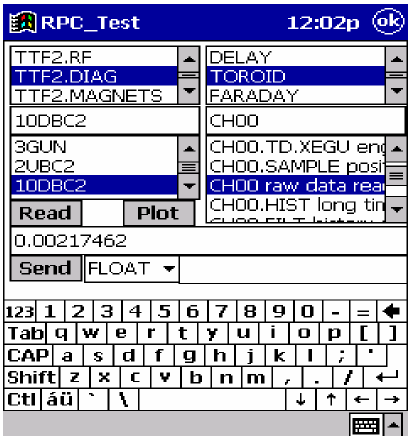

TTF2.DIAG
MCP.MOTOR
MOTOR2
POSITION
TTF2.VAC
TTF2.RF
TTF2.DIAG
TTF2.MAGNETS
TTF2.UTIL
STEPPER
MCP.MOTOR
MCP.S_ADC
MCP.DAC
MCP.ADC
VMEMCP1._SVR
MOTOR1
MOTOR2
SPEED_MAX max. speed of the m
POSITION actual position (mm) F
POSITION.HIST position history
POSITION.DESC type of float ST
POSITION.POLYPARA mm->step c
TO_POSITION_ST move to the pc
TO_POSITION_MM move to the n
Read
-0.00370045
plot update
10
Plot
Read->File
Show File
Send
FLOAT
EXIT

Figure 1: RPC_test application for Windows PC.

a) RPC_test - is a generic test program to display and modify all available device data from the TTF VUV-FEL control system. It reads the actual list of device servers from a name server and the available properties of a certain server from the server process itself.



RPC_Test 12:02p ok

TTF2.RF
TTF2.DIAG
TTF2.MAGNETS

DELAY
TOROID
FARADAY

10DBC2

CH00

3GUN
2UBC2
10DBC2

CH00.TD.XEGU en
CH00.SAMPLE posi
CH00 raw data rea
CH00.HIST long tin

Read | Plot

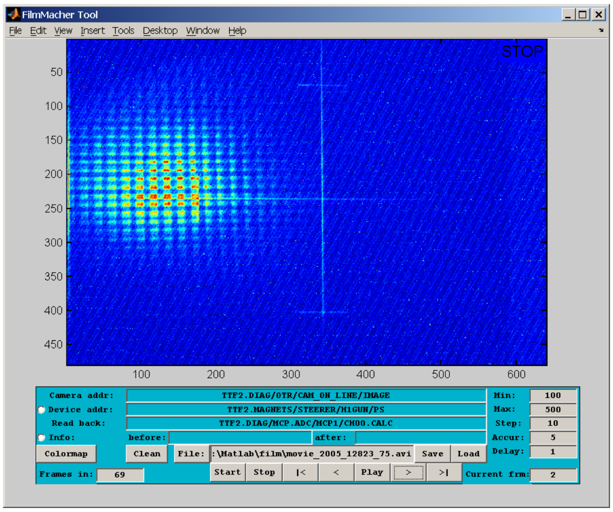
0.00217462

Send | FLOAT

123 1 2 3 4 5 6 7 8 9 0 - =
Tab q w e r t y u i o p [ ]
CAP a s d f g h j k l ; '
Shift z x c v b n m , . /
Ctl áü ` \

Figure 2: RPC_test application for Pocket PC.

b) Many diagnostics are control by Matlab applications with the of use *ttfr()* and *ttfw()* Matlab external interface functions. *ttfr()* - allows to read values from available devices, *ttfw()* – to modify data in desired devices. All these functions assist for data exchange between Matlab applications and linear accelerator control system. They were ported from UNIX to Windows Matlab environment. The *getimage()* external function was developed on Windows and then ported to UNIX. The *getimage()* – provides Matlab applications with different formats of image bitmap which come from DOOCS image transmit servers [5].

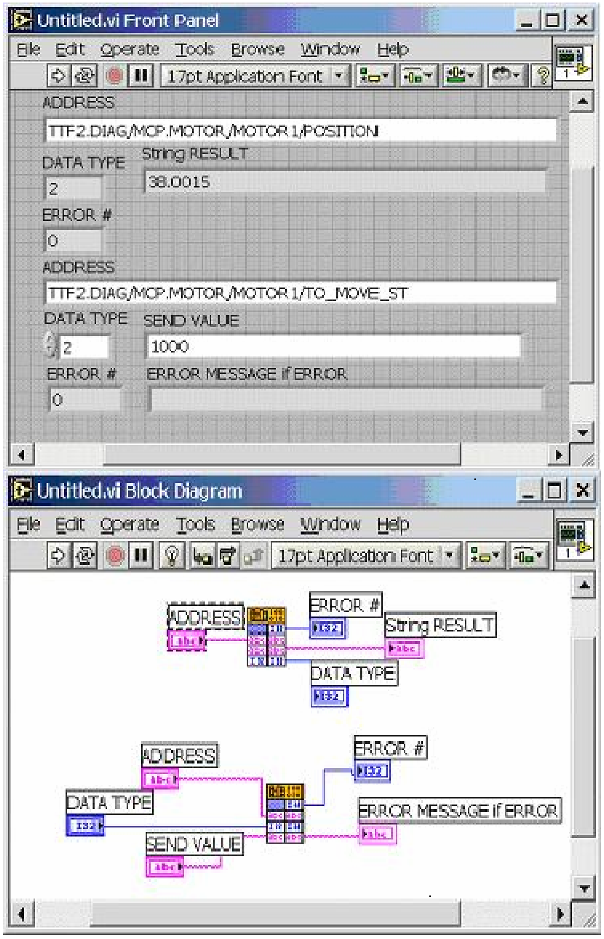


FilmMacher Tool

Camera addr: TTF2.DIAG/OTR/CAM_ON_LINE/IMAGE
○ Device addr: TTF2.MAGNETS/STEERER/H1GUN/PS
Read back: TTF2.DIAG/MCP.ADC/MCP1/CH00.CALC
○ Info: before: after:
Colormap | Clean | File: :\Matlab\film\movie_2005_12823_75.avi | Save | Load
Frames in: 69
Start | Stop | |< | < | Play | > | >|
Min: 100
Max: 500
Step: 10
Accur: 5
Delay: 1
Current frm: 2

Figure 3: Film Maker Matlab application.

A "Film Maker" Matlab application is based on *getimage()* external function. It support scans of linear accelerator device (like a magnet or an RF – phase) and simultaneously creates online movie for the following frames analyses by Matlab. The first photon beam 32nm wavelength of the SASE lasing presentation movie has been done by the "Film Maker" from the photon diagnostic camera directed to a Ce:Yag crystal.



Untitled.vi Front Panel

ADDRESS: TTF2.DIAG/MCP.MOTOR/MOTOR 1/POSITION
DATA TYPE: 2
String RESULT: 38.0015
ERROR #: 0
ADDRESS: TTF2.DIAG/MCP.MOTOR/MOTOR 1/TO_MOVE_ST
DATA TYPE: 2
SEND VALUE: 1000
ERROR #: 0
ERROR MESSAGE if ERROR:

Untitled.vi Block Diagram

Figure 4: A simple LabVIEW vi providing data exchange with DOOCS devices server.

c) In the LabVIEW package VIs send or receive data to/from DOOCS devices via Call Library Function node. This node calls specified shared library (*DOOCSapi.so* for UNIX) or dynamic linked library (*lv_client.dll* for Windows) function directly.

WINDOWS DOOCS SERVERS

Several TDS5000 (Windows ME) and TDS7000 (Windows 2000) Series Oscilloscopes are used to control and adjust linac electronic hardware. These are Windows-

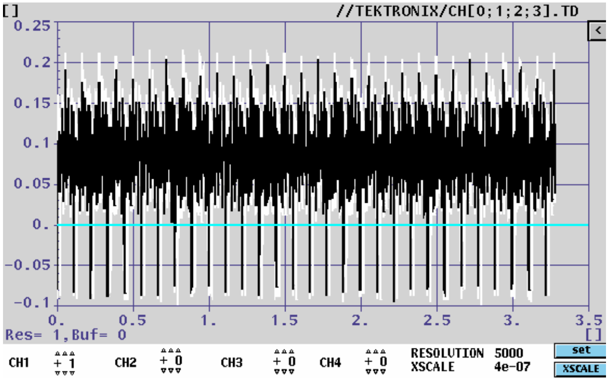


Figure 5: Tektronix oscilloscope DDD component.

based instruments and include a specialized type of GPIB resource called *virtual GPIB*. A DOOCS server running on oscilloscopes has access the embedded instrument software by using a virtual GPIB software connection. The server main goal is oscilloscopes control and data compatibility with DOOCS clients' software.

Lots of VME crates are used in the TTF VUV-FEL control systems. Sometimes it is necessary, to switch crates on/off, to generate system reset or to check the current status of crate like temperatures, states, power channels voltages and currents, errors. The connection between VME crates is done via CANBus. PCAN-PCI card is equipped with the CAN controller SJA1000 and the driver 82C251. It has opto-isolated dual channel and it gives the possibility to have two CAN-buses, which allows to control up to **252** VME crate. The VME crate server *vmecrate.exe* employs *pcan_pci.sys* Windows driver and *pcan_pci.dll* library methods. It has been developed as a Windows services in PITZ at Zeuthen and runs at the VUV-FEL in Hamburg too.

CONCLUSION

The porting of DOOCS protocol to Windows based PCs has permitted a data exchange between Windows and UNIX platforms. New hardware resources were employed within TTF VUV-FEL control system. The WDOOCS based FireWire camera servers are described in [5]. The linear accelerator control software developments are continued on both platforms.

REFERENCES

[1] O.Hensler, K.Rehlich, et al " DOOCS : A Distributed Object-Oriented Control System on PC]s and Workstations ", PCaPAC conference, 1996

[2] O.Hensler, K.Rehlich. DOOCS Distributed Object Oriented Control System . DOOCS - manual.

[3] K.Rehlich, et.al. "DOOCS: an Object Oriented Control System as the Integrating Part of the TTF Linac", Proceedings of ICALEPS97, Beijing China)

[4] K.Rehlich, et.al "An Object-Oriented Data Display for the TESLA Test Facility", Proceedings ICALEPCS '97, Beijing China

[5] V.Kocharyan, K.Rehlich. "WDOOCS:: FireWire cameras support for DOOCS", PCaPAC 2005, Hayama, Japan.