# A laboratory set-up for the remote access to the TMS320C31 SDK

By

Mounir Boukadoum, Department of Computer Science

University of Quebec at Montreal (Canada)

Email: boukadoum.mounir@uqam.ca

## Introduction

This paper describes a laboratory set-up that exists at the University of Quebec at Montreal and that allows access to a TMS320C31 EVM board either locally or from a remote location.

The Electronics and DSP Laboratory of the Computer Science Department is composed of 15 work benches, each equipped with an oscilloscope, a voltmeter, a signal generator, a PC running Windows 95 and connected to the school intranet, and a TMS320C31 EVM that connects to the parallel port of the PC. As the laboratory is used for different courses, physical access to it for DSP practice by the students is limited to a few hours a week. In addition, the laboratory is closed after office hours further reducing access to the equipment by our students. To solve this problem, we have implemented a network set-up that allows access to the C31 EVMs from any location inside and outside the university. The heart of our solution is a software package called SIAD-D (a French acronym for “Integrated Service for Remote Access to DSPs”) that allows communication with the C31 EVM boards both locally and from a distant location, either from home or from another location outside of the DSP laboratory.

SIAD-D evolved from an earlier project, SIAD, that provided the students with either of a local or remote connection to any of a set of standalone development cards, each connected to the serial port of a NCD x terminal. In a typical set up, the x terminals are located in a laboratory classroom, along with the standalone cards, and the remote terminals correspond to the students’ home computers connected to school via a modem. SIAD relies on a client-server architecture to perform its operation. Both of the local and remote terminals connect to a Unix server station via the university’s intranet. Once the server station receives a connection request from a terminal, either local or distant, it identifies the origin of the request and then polls the status of each X terminal to determine the ones that are not being used. Once a free x terminal is found, a communication session is established between the standalone card that is connected to the X terminal and the user. During the session, the user may issue system commands to the Unix server to edit and download programs into the standalone card.

SIAD also has the ability to test the operation of any card by trying to download and run a test program in it. The results are then analyzed and a state is assigned to the card: available, disconnected or defective. SIAD has been successfully used with several EVM boards that have on-board monitor programs and serial communication ports. However, problems arose when trying to use it for communication with the TMS320C31 EVM: the latter doesn't have a built in monitor program; it uses a MS-DOS program, dsk3d, to interface with the user; and it uses the PC's parallel port for communication, all features that prevent the use of SIAD.

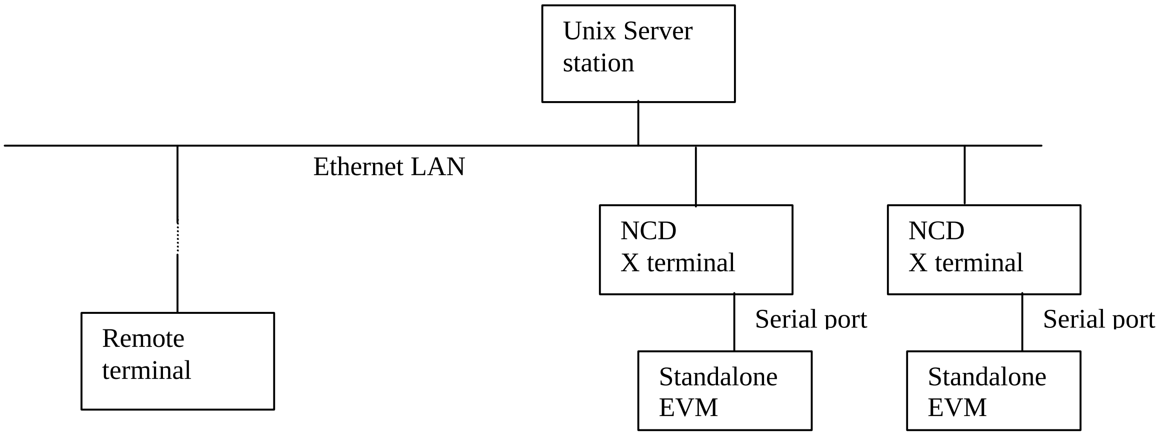

**Figure 1**. Architecture of SIAD

In order to address these problems, a modified version of SIAD, SIAD-D, was developed to work with the C31 EVM. SIAD-D also uses a X terminal, via an emulator program,  in order to connect the PCs within the DSP laboratory to the Unix server station, but it also allows DOS windows to exchange data with the server when needed. The user connects to the Unix workstation for program editing, compiling, assembling,  etc., and to the PC for executing the dsk3d program that comes with the SDK, thus enabling downloading, execution and debugging of user programs. In order to accomplish this, SIAD-D uses a two tier client-server architecture whereby one server program handles both the communication between the user and the Sun workstation and between the user and the PC (hence the C31 EVM). The server talks to the user on one port and switches between two other ports in order to transfer information to the x terminal or to the DOS window.

The server either receives connection requests locally, from the x terminals, or remotely from the outside world. Then , following the example of SIAD, it polls the status of each PC to determine the ones that are available to connect to a C31 EVM. Once an available PC is found, the server creates a session with the user and the latter may either talk to the server or to the PC, depending on the operation to be accomplished (see Figure 2).

## SIAD-D Components

SIAD-D uses four components in its architecture, it has a request server that sits in the Unix workstation, one client application that also sits in the Unix workstation and that handles the communication with a X terminal of the PC, the one whose EVM the user is connected to, and two client application that reside on the PC and that allow the user to either interact with the dsk3d program or to start a program to test the EVM that the user is connected to.

Both local and remote requests from the user go to the server, which routes them either to an application server within the Sun Workstation, or to a DOS window, depending on context. The Unix workstation is used for program development while the DOS windows allows interaction with the 320C31 EVM. The server constantly maintains a list of available boards and of the users connected to them. It also keeps the addresses of the DSP laboratory PCs in order to distinguish between local and remote calls and in order to identify the C31 EVM that is being used by a particular user. As mentioned above, the user may execute two kinds of client programs, one for logging into the DSK boards and the other to test and find out which DSK boards are functional.

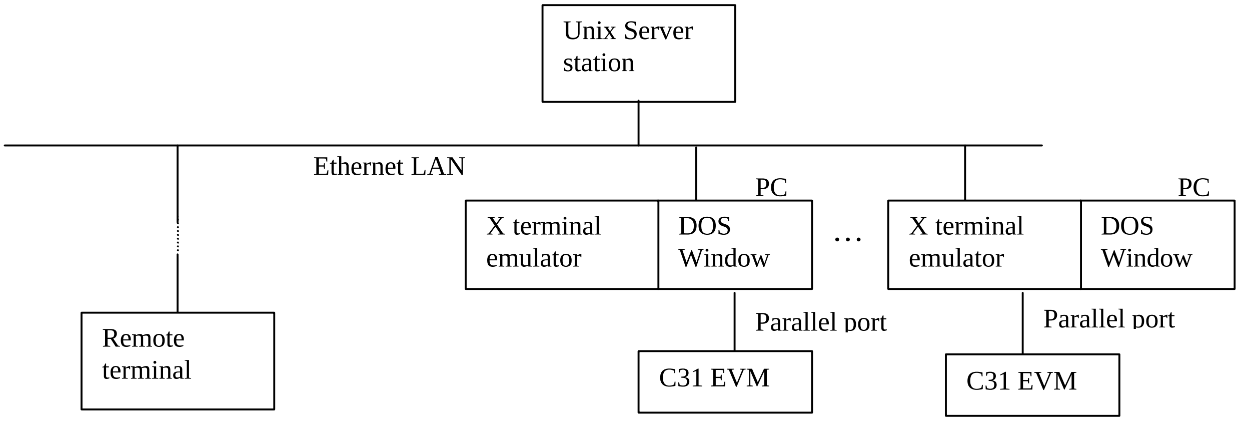

**Figure 2**. Architecture of SIAD-D

## Using SIAD-D

Four different hook up configurations are possible between the user and either of the Unix application server and the C31 EVM :

- Connection between a PC and the attached C31 EVM via the server
- Connection between a PC and the Unix application server
- Connection between a remote terminal and the Unix application server
- Connection between a remote terminal and a C31 EVM via a PC’s DOS windows

The very first time a user uses SIAD-D, he/she executes two shell scripts that adds SIAD-D’s path to the system paths of the Unix station and the PC. Then, after the server program is launched, a user may connect to a board by starting the client program within the Unix workstation from his/her local or remote terminal. Regardless of the connection mode, the same command is issued to start the client program. At this point, SIAD-D has the functionality of SIAD: it may connect any standalone card that is attached to a chosen PC’s to the user. However, since the interaction with the C31 EVM has to pass through the dsk3d program, an additional step has to be performed, namely the re-routing and re-formatting of the data meant for the x terminal to the DOS Window. This is accomplished by switching the port number that the server uses to relay user data to a new one and by starting a TSR client program in the DOS Window that listens to the new port and that creates a pipe between the dsk3d program and the server.

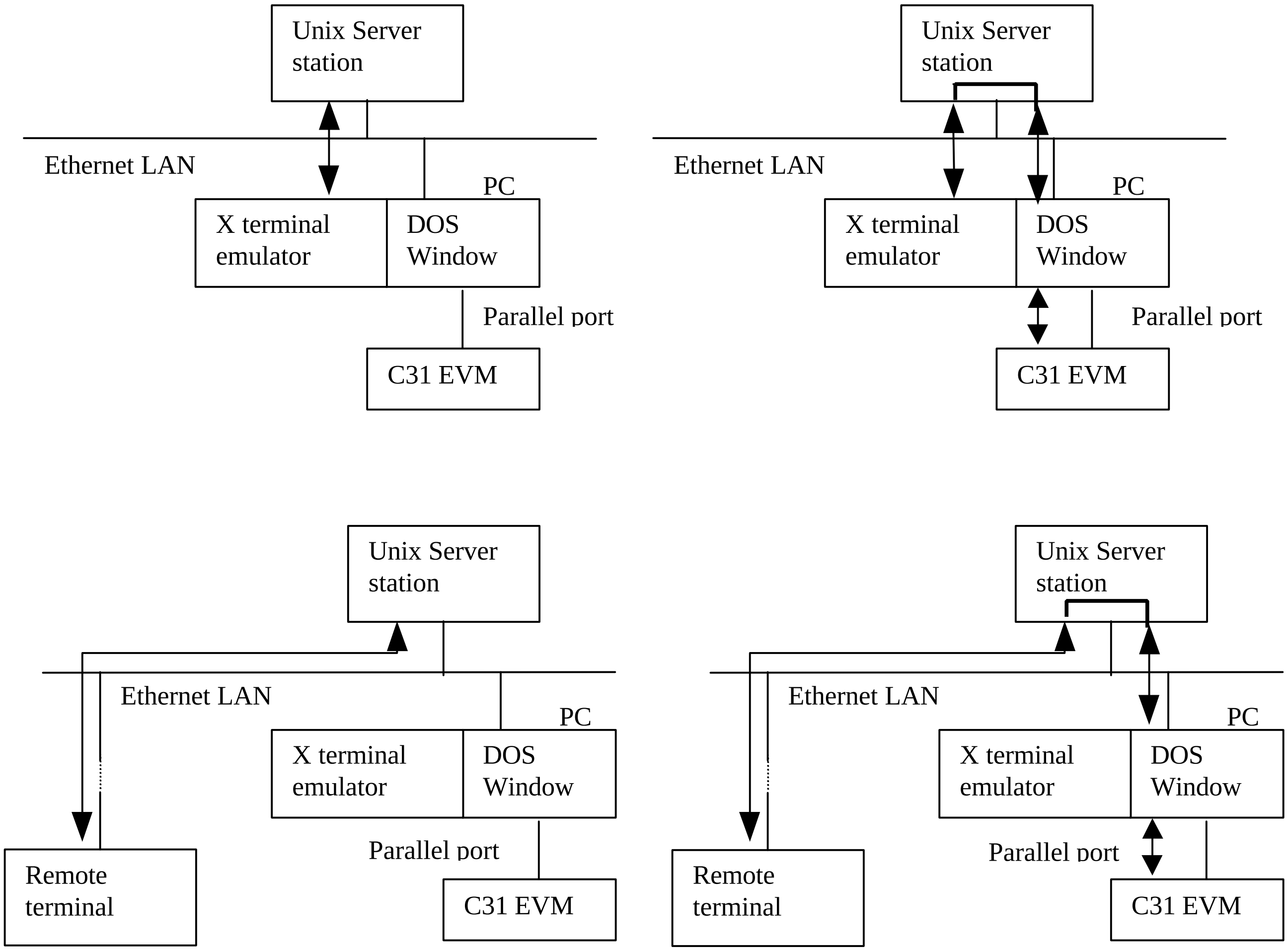

**Figure 2**. Different operation modes of SIAD-D

After the client program is started, it issues a login message to the server port, and, if a communication link between the two is established, the server responds by an acknowledge message that is displayed on the user screen (remote or local terminal). The user may then issue a command to display the server menu. The connection to the server may be terminated at any time by typing <ctrl-d>.

The server's menu choices are as follows:

* Perform a reset of the DSK board (either hardware or software)
* Execute a Unix command from the user terminal
* Communicate with the user PC via a console window
* Switch local and remote directories
* Get help

The ability to execute system commands while communicating with the board was developed so that the users who operate from a single console window may also use edit and assemble their programs without having to leave SIAD-D.

### Access priorities

The distinction between local and distant accesses is only relevant to the server program, which uses this information to identify the C31 board that is attached to the user's PC, in the case of a local connection.  The server uses this knowledge to assign this specific board to the user if possible. The server assigns boards to the users in to the following order:

* If the connection request comes from a terminal that is located in the DSP laboratory, and that is hooked to a EVM board, the server check the availability and functionality of the board. If the check is positive, a connection is established between the user the DSK board; otherwise, an error message is displayed on the user terminal, and the user is asked if he/she wishes to connect to a different board than the one his/her terminal is attached to.

* In all other cases, the user may either request a specific or the first available DSK board, and the server will attempt to fill the request. Here also, a check of the availability and the correct operation of any found DSK is made before connecting it to the user.

If a user has a circuit set up that is connected to a DSK board, then his/her request for connection should be made by specifying the network address of the PC that is attached to the board. This is the default assumption when the user starts the communication program from one of the DSP laboratory terminals.

### Getting board status information

The user may at anytime know the status of all the DSK boards in the DSP lab. Two programs are available. The first one polls the different links to the boards, and then reports busy, available or defective for each board, depending on whether it can connect to the board and, if yes, whether the board passes a test program. The second program not only displays the current status of the boards, but also alerts the user of any subsequent change of status. Each time that a board changes state, its new status is displayed and a beep is generated. This option allows users to be notified when a board becomes available in case all operational boards are used when they attempt a connection.

## SIAD limitations

SIAD transmits data as ASCII strings delimited by a carriage return. For instance a command is only send to the DSK after the user hits the <*Return*> key. As a result, commands that do not end with a carriage return key cannot be processed.

At the present, only telnet connections to the boards are possible. A provision for internet access would be an elegant alternative, albeit slower and less functional, as it would provide a simple solution to the Unix server-DOS window communication problem, by using the http protocol along with the html/Java languages. A truly distributed architecture, using CORBA, would be an even better solution as it adds the benefit of using any platform and any programming language for the development of the client side of SIAD. It is this solution that is being implemented for the next version of SIAD.