

# Remote Web Server of Digital Signal Processors Cards and its Applications in Education

Ibarra Picó, Francisco, Fuster Guilló, Andrés, Colom López, José

DTIC. Departamento de Tecnología Informática y Computación. Campus de San Vicente. Universidad de Alicante. Apartado 99, 03080 Alicante. SPAIN

ibarra@dtic.ua.es

## Abstract

*A system is presented for the remote execution and planning of processes on digital signal processor cards. The hardware and software designed permits the sharing of processors and remote execution on the part of the users. We can distinguish three major modules: the Host, which acts as a signal processor hardware server, the customers, who send processes (typically programs written in C) and receive the compilation and results of the execution, and lastly, the host/customer connection (the link is carried out on a logic level, and thus independently of the type of net connection: internet, ethernet, token...). This situation is adequate for permitting access to high cost hardware platforms on the part of multiple users, facilitating the sharing of resources in research groups and the utilization in the classroom (at accessible costs) of high technology (where, until now, only the use of low cost processors has been envisaged).*

*Index terms-- Multimedia, educational, applications, remote access, Digital Signal Processors*

## 1. Introduction

In engineering studies the use of DSPs in laboratories is interesting, but their high cost make it difficult. The use of Web multimedia environments for the shared access to this kind of resources is the object of this work. As it is known, systems based on digital signal processors (DSPs) are widespread and with many diverse applications [1]. One of the problems with high calculation power DSPs is their high cost per unit. This prohibits, in the majority of cases, their use in teaching laboratories, where the number of working terminals necessary causes the total costs to escalate hugely. This confines the utilization of these resources to research laboratories and/or industrial applications; making it impossible, in many cases, to establish classrooms with access to and use of high technology. The standard nature and the low cost of the equipment necessary to set up a laboratory based on the DSP server presented here allows the establishment of a laboratory with multiple points of access, with a saving of over 70% in many cases.

The development of tools and methodologies allowing the sharing of resources is a dynamic process in computer science, which has been spreading progressively in the different parts which make up a computer (peripherals, multitasks and CPUs, memories,...) Moreover the appearance of computer nets introduces strategies of remote process execution, distributed process execution, etc [2].

We find ourselves before a new integrating step in the development process, where, using computer nets, we gain access to a shared resource: the signal processing card.

As we have indicated, there exists a wide range of DSPs with diverse applications and costs. In particular, those of the highest range, orientated towards image processing, communications and concentrators (TMS320C4x, TMS320C6x, TMS320C8x, SHARC, ...) [3][4][5] in isolated form and, above all, in development and application cards (for example, a card based on the C40 with memory and frame grabber for video input) reach very high costs [6]. This confines their use to specialized groups, and, logically, in applications which require their high computing speeds of hundreds of MFLOPS. This fact prohibits their generalized use in the classroom, because of the high cost associated with the introduction of tens of units in laboratory groups. In this way practical sessions are habitually carried out using lower power-performance (and cost) technologies, with the use of more specific equipment limited to very small groups of students (at best). The system described here allows the shared massive and instrumental use of high cost hardware platforms.

Other utilities of this type of system are the shared use of platforms by research groups, and the remote change of programs in execution time. From the point of view of a research group, the DSP server system permits the shared use of resources from remote locations, without having to move physically to the operative system or the machine in use. Moreover, the interchange of software in execution on a remote platform allows the adaptation of process algorithms to changing local conditions (and the execution at any given moment of the most adequate algorithm).

## 2. System architecture

The customers who enter in the proposed system can be on any platform, the only requisite is to be able to execute a web navigator. And to have total control, the automatic generation of a list can be configured with all the entries which take place. The Customer places an order via a web page. The web server receives the order by means of a dynamic link library (Vdsp.dll).

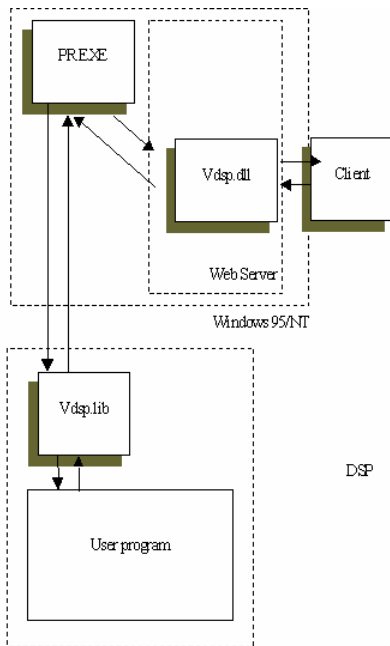


Figure 1. DSP server architecture

The latter carries out the appropriate security checks and sends the order to the PR.EXE program. This program interacts directly with the hardware installed at the server. The above-mentioned program loads the customer code in the DSP and initiates the execution. To establish communication with the DSP, a functions library (Vdsp.lib) is used which links with the customer code and executes in the DSP itself. Vdsp.lib receives the exits and entries of the code and sends them or elicits from the PR.EXE program. This controls the permitted execution time and prepares the response page for the customer. When the code finishes the execution, it informs the PR.EXE program. The response is shown to the customer via the web server [7]. The system can be seen graphically in figure 1.

## 3. Integrated functions

The customers of the system can registrate therein, remotely compile and link programs in C (Texas Instruments Compiler), and carry out executions upon the DSP server.

The following buttons in figure 2 appear on the principal panel to provide access to the functions which the user may require from the system. They are grouped in two sections: development and access.

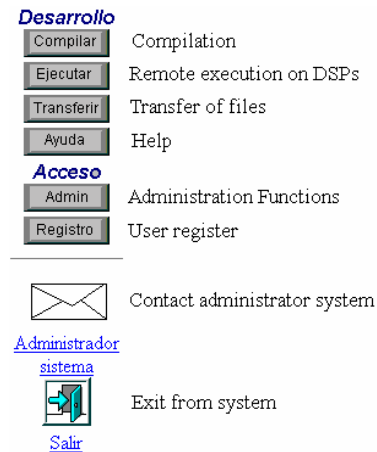


Figure 2. System functions

### 3.1. Development

This refers to the services related to the development of programs for DSPs.

#### 3.1.1 Compile.

Using the Compile button, we gain access to the compilation form. On first entry, the name and access code, as introduced into the registration form, will be requested.

In the File field, we must put the name of the file we wish to compile. This file has to be found locally and can be localized using the button which appears on the right of the field. There appears underneath a field which indicates which processor must execute our code. The Compile button will carry out the compilation at the server and will show us the exit from the compiler. Some parameters of the compiler can be specified by marking the boxes in Compiler Options.

Fichero fuente a compilar	<input type="text"/>	<input type="button" value="Examinar..."/>
Procesador	A	
	A	
	B	
	C	
<input type="button" value="Compila"/>	<input type="button" value="Limpiar"/>	

---

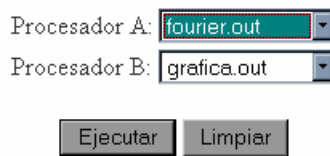
Opciones del compilador:	Compiler options
<input checked="" type="checkbox"/> Genera listado ensamblador	Generate assembler list
<input type="checkbox"/> Interlist de C y ensamblador	Interlist of C and assembler
<input type="checkbox"/> Compatibilidad K&R	K&R compatibility
<input checked="" type="checkbox"/> Relajar comprobación de prototipos	Relax prototype check
<input type="checkbox"/> Relajar combinación de punteros	Relax pointer combination
<input type="checkbox"/> Optimizar código	Optimize code

Figure 3. Compilation

### 3.1.2 Execute.

The execution page shows several unfolding lists. Each one of them corresponds to a processor of the PCIC44/QPCC40 board, and contains executable files which are found in our directory. We will see, in the section on the configuration of the board, the possibility of configuring up to 4 processors, in which case four unfolding lists would appear. The particular case reflected in figure 4 refers to a configuration with two C44s.

Any combination can be adopted for the execution, only having to bear in mind that each executable was compiled to be executed with a certain processor.



Procesador A:

Procesador B:

Figure 4. Execution

On pressing the Execute button, the execution is carried out in the DSPs, and the exit is shown on the results page. The results which imply the transmission of a file are shown as hyperlinks which, when pressed, discharge onto the local disc. The exit from multiple processors is done on the same results page, offering the programmer the possibility of distinguishing the origin of the messages or not.

### 3.1.3 Transfer.

The following service allows the user to send as many files as he considers necessary to be read by his programs, using the data entry functions which we will detail in the section on the programmer. The files will be stored in the directory which the user has at the server.



Fichero

Figure 5. File transfer

Figure 5 shows the option which is presented for sending a file. Loading the page as many times as he considers necessary, the user can send as many files as he needs. With the 'Browse' button (the label depends on the navigator) the user is able to navigate through the hierarchy of files in the local machine.

If we so desire, we can compile programs locally and send only the executable code to the server.

### 3.1.4 Help.

A help section is provided which contains web pages with indications relative to the compilation, execution and transfer processes, as well as the functions available to the programmer.

## 3.2. Access.

This section comprises two types of task. On the one hand, those relative to user registration applications, and on the other hand, those concerning the administrator, which will be expanded upon in the section dedicated to administration. We concentrate, for the moment, on the registration option. The system introduces an access control, permitting entry to registered users, and distinguishing between types of users with differing capacities.

### 3.2.1 Types of users:

#### 'Anonymous' user

Can carry out the following activities:

1. To obtain information on digital signal processors in general, and on our system in particular. This information is obtained by the simple act of connecting with the corresponding URL. The user is shown how to register, and is offered images of the different boards and configurations.
2. To send a registration application, in order to become a user of the system, if the administrator considers it appropriate.

#### 'Guest' user

Once the guest user's login and password are introduced, he will be able to see some user pages, but will only be able to compile and execute a series of pre-established test programs.

#### 'Normal' user

Will receive the following services:

1. Compilation of source programs for d.s.p., written in C and using a library which the system provides.
2. Execution of such programs concurrently in d.s.p.
3. Sending of files via internet to be used for the programs.

#### 'Administrator' user

This type of user has access to the following operations:

1. User registrations.
2. Management of users and applications: consultations, drop-outs and user modifications, elimination of applications made by anonymous users, or registrations of users stemming from such applications.
3. Configuration of d.s.p.: type of board and memory addresses.

### 3.2.2 Admin.

This button provides access to the administration functions, which are grouped on the following way:

- [User registrations](#)
- [Management of users and applications](#)
- [DSP configuration](#)

Figure 6. Administration options

We will detail these functions in the section dedicated to the administrator.

### 3.2.3 Register.

In order to registrate as a user it is necessary to fill in a small form, to which we can gain access via the Register button on the Navigator (figure 7). Once the form is filled, the administrator will pick up the application and give the relevant permission.

Apellidos:	<input type="text"/>	Surname
Nombre:	<input type="text"/>	First Name
E-mail	<input type="text"/>	E-Mail
Login	<input type="text"/>	Login
Clave:	<input type="text"/>	Password
Confirma Clave:	<input type="text"/>	Confirm Password
<input type="button" value="Aceptar"/>		

Figure 7. User register

## 4. Programming the DSPs

In order to program and gain access to the system, previous knowledge beyond sufficient notions of C language is not necessary.

For the system to interact correctly it is necessary to include a series of lines in our code. The minimum program skeleton can be seen in the following list:

```
1 void main(void) {
2
3     /*-- Wait for semaphore--*/
4     esperaInicio();
5
6     salidaCadena( "Hello from A.\n" );
7
8     /*-- Inform that you have finished --*/
9     marcaFin();
10 }
```

Figure 8. Minimum program

The above program shows the exit "Hello from A.". In line 4 we wait for the PC to give us the signal to begin execution. In this way, the beginnings of the programs in each DSP are synchronized. As the very name indicates,

esperaInicio(), wait for the corresponding semaphore to open, according to the processor on which we are executing.

Once the program execution is finished, it is necessary to indicate to the PC that it is finished. This is done with the call marcaFin().

Memory management. Although direct memory management is possible, it is better to leave the administration thereof to the programming libraries. Assigning memory as is done in the standard way from C (malloc), there should be no problem.

Data interchange functions. As is logical, in our programs we wish to obtain a result, and that this result should be explanatory to the maximum degree possible. In our case, we have a wide range of formats in which we can give the results, depending on their type.

The code sent by the user to be compiled is C standard, and will use routines of a library provided by the d.s.p. server. These routines emit results in different formats: graphs, images, texts, numbers...The standard libraries incorporated by the compiler may also be used (stdlib.h, math.h..., but not stdio.h).

The types of data carried by the DSP server correspond to those of the Texas Instruments compiler. The types of data and the library functions which the system can handle are listed in the user's guide.

## 5. Administration tasks

To gain access to the administration of the DSP Virtual Laboratory, it is necessary to have been designated an Admin access level (2).

The functions explained below are accessible by pressing the "Admin" button on the principal menu. Once a correct administrator login and password have been introduced, access is gained to a page from which the following operations can be carried out:

- [User registrations](#)
- [Management of users and applications](#)
- [DSP configuration](#)

Figure 9. Administration options

## 6. Implementation and tests

This environment has been used to carry out the sharing of boards of type PCI/C44S TMS320C44 Carrier Board. A system with a PCI board with four C44 and 128K shared memory processors and QPC/C40S with C40 processors.

At this moment, the system developed permits access to cards of different types, and there are several operative drivers. The co-existence is also possible of different cards upon the host, and the construction of a hybrid system adapted to specific applications. For example, systems with 8 DSPs C4x and line frame grabber.

The boards were located in a PC acting as host (Pentium 150 Mhz, or superior, 32 MB RAM, 1GB HD, Net Card 2000 NET or standard, Windows NT server or Windows 95 with Internet Information server or ISAPI). The customers were pentium and 486 type PCs with 8 MB RAM, minimum of 0.22 GB HD, Net Card 2000 Net or standard, net Navigator, Windows NT Operating System 3.5, WIN95 or Windows 3.11.

The compilation page allows us to search for the source file on the local disc, and to indicate the file execution processor. We can select certain compilation options.

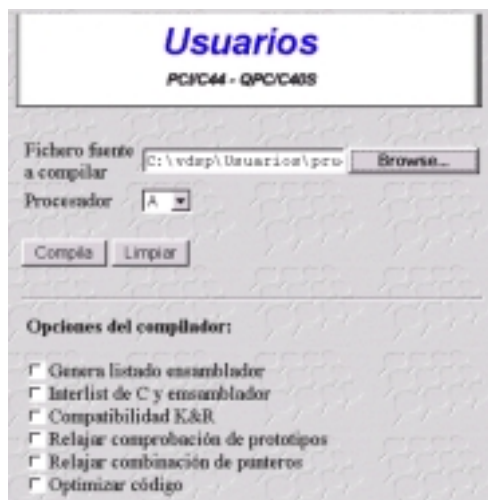


Figure 10. Compilation pages

The result of the compilation is shown in figure 11.

### Compilación

```

Compilando y Linkando...
[prueba]
TMS320C3x/4x ANSI C Compiler Version 4.70
Copyright (c) 1987-1996 Texas Instruments Incorporated
"proca\prueba.c" ==> main
TMS320C3x/4x ANSI C Optimizer Version 4.70
Copyright (c) 1987-1996 Texas Instruments Incorporated
"proca\prueba.c" ==> main
TMS320C3x/4x C Code Generator Version 4.70
Copyright (c) 1987-1996 Texas Instruments Incorporated
"proca\prueba.c" ==> main
TMS320C3x/4x COFF Assembler Version 4.70
Copyright (c) 1987-1996 Texas Instruments Incorporated
PASS 1
PASS 2

No Errors, No Warnings

TMS320C3x/4x COFF Linker Version 4.70
Copyright (c) 1987-1996 Texas Instruments Incorporated

```

Laboratorio Virtual de DSP 1997

Figure 11. Compilation pages

Our system was configured with two C44 processors. This is why it shows two unfolding pages to select the file to be executed.

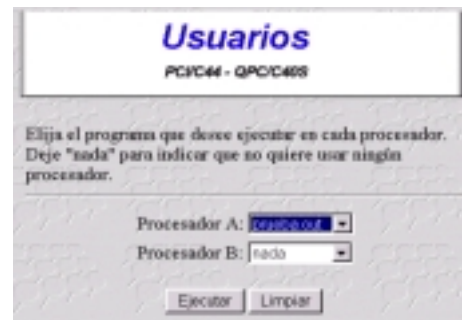


Figure 12. Execution page

We also see, in figure 13, the result of the execution of a program which reflects a matrix of calculated values.

### Ejecución

Hola mundo

120	120	120	120	120
120	120	120	120	120
120	120	120	120	120
120	120	120	120	120
120	120	120	120	120

Laboratorio Virtual de DSP 1997  
Universidad de Alicante

Figure 13. Execution page

## 7. Conclusions

The practical benefit has been stated of this type of system in the classroom and in other situations (research groups, dynamic process situations which require changes of Kernels software, etc) allowing access to high technology at reduced costs, thanks to the "transparent" sharing, regarding the user of this situation: in real terms there is a perception of using the board in a monoset mode. Moreover, access to the DSP resources is carried out in a real manner from a web multimedia environment, with no kind of simulations. The execution and compilation is carried out by the DSP itself in real time.

## 8. References

- [1] Texas Instruments. TMS320 DSP Solutions CD-ROM. Technical Documentation. Texas Instruments. 1997.
- [2] Hwang, K. Advanced computer architecture. Mc Grwa-Hill. 1993.
- [3] Texas Instruments. TMS320c4x User's guide Texas Instruments. 1992.
- [4] Sonitech International, Inc. DSP Solution Guide. Sonitech. 1994.
- [5] Ifeachor, E, Digital Signal Processing. Adisson-Wesley. 1993.
- [6] Loughbourgh Sound Images. High Performance Image Processing Systems. LSI. 1996.
- [7] Borland. User's manual of Borland C++ and Database tools. Borland. 1995.