Speech Control for Virtual Instruments Using the TMS320C30 DSP

Authors: D. Santos, J. Costa, R. Vasconcelos, S. Rodrigues, D. Freitas, C. Espain

ESIEE, Paris

September 1996
SPRA317

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI™) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Figures

Speech Control for Virtual Instruments Using the TMS320C30 DSP

Abstract

This application report describes a speech recognition interface for a Microsoft Visual Basic (VB) application. The VB application is a virtual instrument control system that can be used in various environments. The speech recognition interface enhances the ability of the operator to control various system components without manually navigating the graphical user interface (GUI) screens of the VB application.

Several options were considered during the analysis but only one option proved to be optimal. The solution described in this paper uses the Texas Instruments (TI™) TMS32C30 DSP to handle all speech recognition calculations.

The speech recognition component using the TI TMS320C30 DSP was implemented as a co-processor residing on the PC. The VB control system runs on the main PC processor. The controlled instruments are attached to the system through an IEEE 488 parallel bus.

This report includes a system level diagram and references.

This document was part of the first European DSP Education and Research Conference that took place September 26 and 27, 1996 in Paris. For information on how TI encourages students from around the world to find innovative ways to use DSPs, see TI's World Wide Web site at www.ti.com.

Product Support on the World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

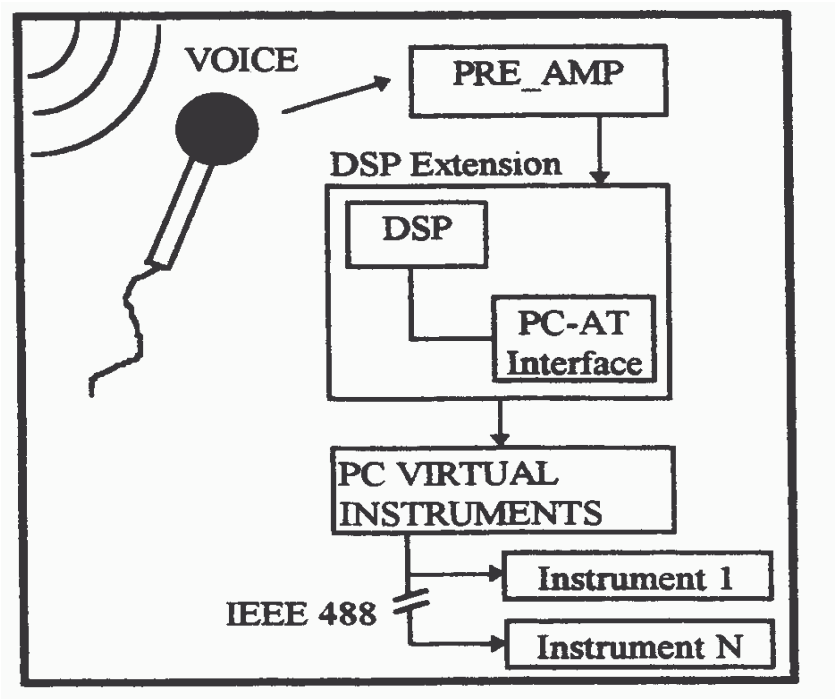
Introduction

This paper presents the development of a speech controlled virtual instrumentation system. Voice recognition includes applications in areas such as instrumentation, speech processing and industrial control. This project is a culmination of efforts in an Electrotechnical Engineering Degree program.

Because PC performance was a critical consideration in the speech control system, it was decided to design a co-processor in the form of a PC bus extension for the purpose of offloading the speech signal processing calculations. The extension ensures that the PC applications would not be affected by the intense computational requirements of voice recognition. The core consists of two boards and are based on the computing power of the TI TMS320C30 DSP.

Figure 1 illustrates the architecture of the developed system.

Figure 1. System Architecture



Description of the Core

The DSP extension was developed for the 16 bit PC-AT bus. It has 64 Kbytes of base memory, allows easy expansion, and is shared between the DSP and the PC in a way that a program can be loaded without the use of EPROM's. Initially, the intention was to use a Test Bus Controller however the bus controller was not yet widely available. The DMA solution was not used because the project did not require the transfer of large amounts of data between the TMS320C30 and the PC. Only a small amount of data and a few control signals were required.

The voice input is digitized by the analog interface circuit (AIC) resident on the DSP board.

Communications between the DSP and the PC are based upon an interrupt protocol. This protocol is supported by several routines in the DSP *operating system* (OS). The DSP operating system activity is centered around data acquisition by the AIC.

When the data arrives, the OS launches a set of routines that initiate the speech recognition program executing in an isolated-word recognition mode. The functions and routines that establish communication with the PC-AT bus are very important and are incorporated in the kernel of the DSP OS.

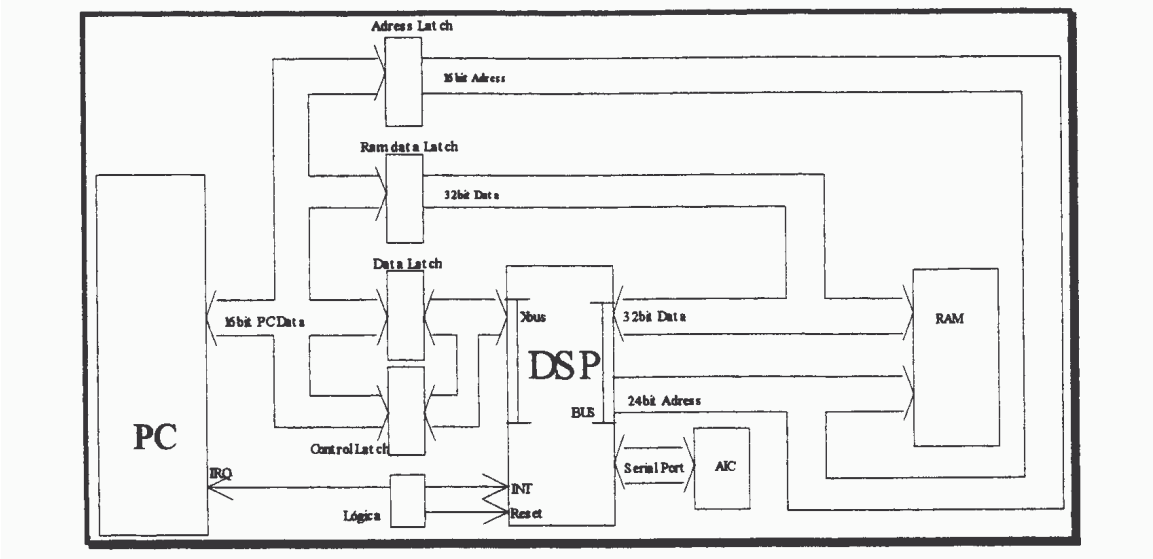
Hardware

The DSP core is a typical circuit with a TMS320C30 running at 33Mhz, with 64K Words of static RAM. The upgradeable serial AIC has a sampling rate of 8Khz and14 bits of resolution.

The DSP has 2 ports: the primary port, which controls the data and program memory, and the expansion port that controls the communication memory and peripheral port. On the primary port the address map is divided in 2 blocks of 8M: on the lower block resides 64K of ram and on the upper block resides a 1M block of static ram. The expansion port has 2 functions: data communications with the PC, which is implemented as an addressable block of memory, and the peripheral port that is addressed as an Input/Output (I/O) port.

Figure 2 shows a block diagram of the circuit:

Figure 2. Core Circuit Diagram



Two aspects of the core system require a detailed explanation:

- Bootloader
- Communications protocol

The bootloader has the majority of the EPROM emulator programs: the DSP memory is shared with 2 ports on the PC, a I6bit port to generate the addresses (Address Latch), and a 32bit port (Ram data Latch) for data.

The DSP is placed in the reset state during the initial program loading so that the data and address latches on the DSP will have a high impedance. This allows the PC to control the shared RAM during program loading. After the program is loaded, the PC's latches are set to high impedance and the DSP is taken out of the reset state. Once the DSP is out of reset, it begins running the program that was previously loaded into RAM.

The communications protocol is interrupt-driven: the sender writes to the Data Latch when it wants to send data. The write generates an interrupt at the receiver. The receiver, via the interrupt routine, reads the Data Latch and writes a response to the same latch in order to generate an interrupt at the sender. This completes the data transfer and notifies the sender that the value has been read. Since the interrupt is the same for both sender and receiver, the interrupt routines must keep track of the phase they are in: waiting for confirmation or waiting for data.

Operating System

The OS of the DSP consists of a collection of functions and interrupt service routines. The OS runs monitoring functions in the background to ensure the proper operation of the system and to quickly react to real-time events.

The main program executes the initialization routines for the different parts of the system when it is first started. After startup, the main program executes in an infinite cycle processing data from the AIC through OS functions. A system error or communications from the PC will interrupt the data processing cycle. An example of a communications interrupt would be the system mode definition such as command recognition or system training.

A watchdog timer was implemented to allow the DSP OS to recover from errors that may occur in the code downloaded from the PC.

Recognition Software

The main procedure will accomplish the recognition of isolated words previously defined by the user.

The basis of the recognition is the use of the *Hidden Markov Model* (HMM).

First of all, the system must go through a phase of apprenticeship for every word specified (10 or more samples are necessary for each word). Normalized *WAV* files were used for system testing.

1) For each file we attempt to determine the beginning and end of each word, following these steps:
   a) Division of the signal in 256-point frames, with 100-point overlaps.
   b) Calculation of the signal's energies and number of zero crossings.
   c) Based on those values the word is encountered.
   d) Calculation of the signal's noise energy.
2) In the second phase, the *Mel-cepstral* coefficients and delta-energies are calculated.
3) Then we realize the vector codification of the frames using *K-means* algorithms.
4) Next step is the recognition or system training using *Viterbi* algorithms.

5) Finally, we use the HMMs to obtain a final vector of the characteristics of the whole word. This vector is compared with the initially trained table of word vectors.

This operation is divided into two modes: recognition or training. Training must be defined at the beginning of the session. Both modes follow the same procedure, except for the training phase when the HMMs matrices must be changed. In the final analysis, recognition consists only of the comparison between the current sample under inspection and the results of the training phase.

Both the operating system and the main procedure were written in C language and then compiled using the TMS320 floating-point DSP optimising C compiler from Texas Instruments.

Virtual Instrumentation System

The main system also contains a graphical user interface implemented in Microsoft's Visual Basic. The VB application provides access to different electronic virtual instruments through the VB drivers. The actual instruments are physically connected to the PC by an IEEE 488 bus. This approach facilitates the rapid development and implementation of a virtual instrumentation environment.

The virtual instruments selected for this project were modeled after Hewlett Packard function generator and the Keithley multimeter. Almost all the functions of the actual instruments were implemented.

The virtual instrument program operation consists of receiving a voice command from the DSP. The PC VB program is interrupted by the DSP according to the communications protocol discussed in the hardware section of this report. The VB program decodes the voice command and executes the order.

The virtual instrument program can also operate independently of the voice command system. For example, in this mode the program could execute a two-channel analysis.

Conclusion

This system exhibits a very empowering functionality regarding the human-machine interface. It provides the capability of supporting the work of people with disabilities as well as relieving an operator from the tedium of manual commands through a simple but effective voice interface.

Educational advantages were derived from developing the TI DSP system. The modular design approach and the joint development of hardware/software provided an excellent opportunity to learn without compromise and to simultaneously produce a very competitive product for real-world commercial use.

References

TM5320C30 Users Guide

The Programmer's PC Sourcebook, ThomHogan

Visual Basic 3.0 for Windows, Curso Completo, Nuno Nina

Windows System Programming, Peter Wilken and Dirk Honekamp