October 31, 2002

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, B.C., V5A 1S6

Re: ENSC340 Project Design Specification for SmileyBaby Mobile

Dear Dr. Rawicz,

The following document, Design Specifications for SmileyBaby Mobile, delineates the components and subsystems used to achieve the project goals. It will describe the function and use of the DSP board, the software composer and the algorithms that we plan to use for the project.

The purpose of this design specification is to clarify and solidify for the design team the theories and methods necessary to maximize the characteristics of the DSP board and the software to make this project successful.

LittleFellows is comprised of four fun-loving but focused energetic 4th year engineering science students: Shona Huang, Marjan Houshmand, Farnam Mohasseb, and Farhood Hashmi*. If you have any questions concerning our proposal, please feel free to contact us by email at ensc340-group@sfu.ca*

Sincerely,

Farnam Mohasseb
Chief Executive Officer
LittleFellows Inc.

Enclosure: Functional Specification for SmileyBaby Mobile

Design Specification for SmileyBaby Mobile

Team Project: *Farnam Mohasseb*
Marjan Houshmand
Shona Huang
Farhud Hashemian

Submitted to: *Dr. Andrew Rawicz*
Steve Whitmore
School of engineering science

Issued date: *October 31, 2002*

Executive Summary

Design Specification of SmileyBaby Mobile
LittleFellows Inc.
October 31, 2002

LittleFellows Inc. mission is to create a friendly and comforting atmosphere for babies and to assist their parents with ever increasing challenges of raising a child in today's busy working environment.

LittleFellows Inc. is a new and small company with unlimited potential to lead technological-base and innovative baby product of tomorrow's market. Four engineering students from Simon Fraser University have incorporated this company in fall 2002. The key element of success of LittleFellows lies in the company's diverse background in both technical and non-technical areas. LittleFellows' ambitious culture distinguishes the company from the other competitors.

Our first product, SmileyBaby Mobile, is entering the market in spring 2003. This smart gadget will distinguish babies crying pattern from all other surrounding sounds, and it will play babies' favorite music to put them back to sleep. In future models, this gadget will be interfaced with other toys and mobiles to create a comforting environment for babies.

Parents like to see their child smiling, and at LittleFellows we do our best to bring comfort, warmth and happiness to both parents and their little fellows. LittleFellows' highest priority is to address baby's needs, and we do realize parents are extremely concerned with their child's safety and well being. LittleFellows Inc. paves the road to success and achieving the highest level of customer satisfaction by paying close attention to customer needs.

Table of Contents

Table of Figures

Introduction

SmileyBaby Mobile is a device that turns on the mobile and plays soft music automatically when the baby cries. Researchers in the field of psychology have discovered that babies use different patterns of crying to show their feelings. Most individuals, especially parents, have a hard time ignoring baby's cry and will attempt to attend to their baby. However, not all the cries require the parents' immediate attendance. SmileyBaby is a tool intended to decrease the amount of work required for parents to complete. In addition, SmileyBaby furnishes the baby with warmth and comfort. In other words, this gadget detects that the baby is crying, perhaps due to boredom, fear or loneliness, and runs a mobile or plays a selection of music. A functional unit of SmileyBaby will be ready by December of 2002. Further developments for commercial purposes will occur following that date.

Scope

This document delineates the components and subsystems used to achieve the project goals. It will describe the function and use of the DSP board, the software composer and the algorithms that we plan to use for the project.

The purpose of this design specification is to clarify and solidify, for the design team, the theories and methods necessary to maximize the characteristics of the DSP board and the software for the success of this project.

Referenced Documents:

- http://www.ensc.sfu.ca/users/whitmore/public_html/courses/305/305.htm, Specifications and Post-Mortems.ppt, Example Functional Spec.pdf, Steve Whitmore, September 14, 2002.

- http://www.parenthood.com/articles/phw877.htm, Why Do Babies Cry?, Fiona Marshall, May 30, 2002.

- http://www.blakehall.demon.co.uk/ExampleIntroLesson.htm, The Development of Social Behavior,
- Home Based Study Ltd, May 30, 2002.

- http://www.todaysparent.com/baby/article.jsp?cId=6518, How Babies Communicate, Teresa Pitman, May 30, 2002.

- http://www.slumbersounds.com/baby-crying-info.htm, Crying Out Loud - Deciphering and Soothing Your Baby's Cries, Dr. Charles E. Schaefer, September 25, 2002

- http:// www.fisherprice.com, Fisher-Price homepage: www.fisherprice.com, Oct.1, 2002.
- http:// www.cppsc.gov, U.S. Consumer Product Safety Commission homepage: www.cppsc.gov, Oct1, 2002.

- http://www.epinions.com/Tiny_Love_Maya_Symphony_in_Motion_3D_Toys, Oct. 3, 2002.

- http://www.fisherprice.com, Fisher-Price homepage, Oct.1, 2002

- http:// www.cppsc.gov , U.S. Consumer Product Safety Commission homepage, Oct1, 2002

Intended Audience

The design team of the company will be using this document to develop detailed strategies to make the project successful. The project manager will keep all design team members on task and schedule by referring to this document.

System Overview

This section outlines an overview of the SmileyBaby system. As demonstrated in, SimleyBaby consists of several sections. Figure 1 shows a microphone receiving sound in the room and converting analog audio to a digital signal. The signal will be stored in a FPGA (a type of memory) from which the signal will be fed to a DSP (Digital Signal Processor) for analyzing. The device will decide if the existing sound contains the baby's cry. Finally, the outcome from the Digital to Analog converter turns on or off the mobile determined by the DSP.

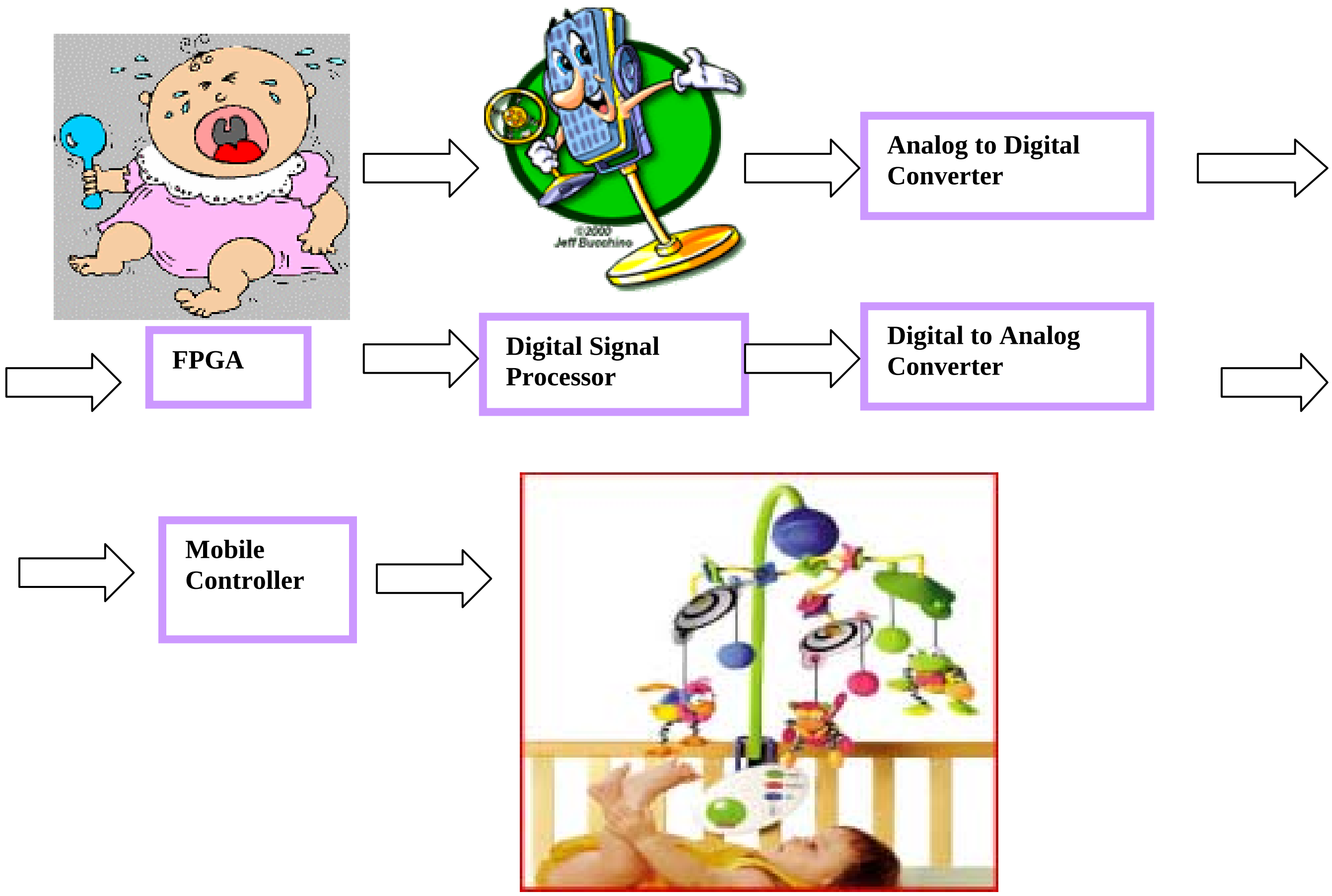

Figure 1, System Overview of SmileyBaby Mobile

System Hardware

The design team has chosen to use the DSP chip TMS320C6201 on the Evaluation Module (EVM) produced by Texas Instruments. This complete system is composed of a C6x series chip, SBSRAM (synchronous burst static

RAM), SDRAM (synchronous dynamic RAM), analog to digital conversion capabilities, and a PCI bus for PC interfacing. The functional diagram is shown in

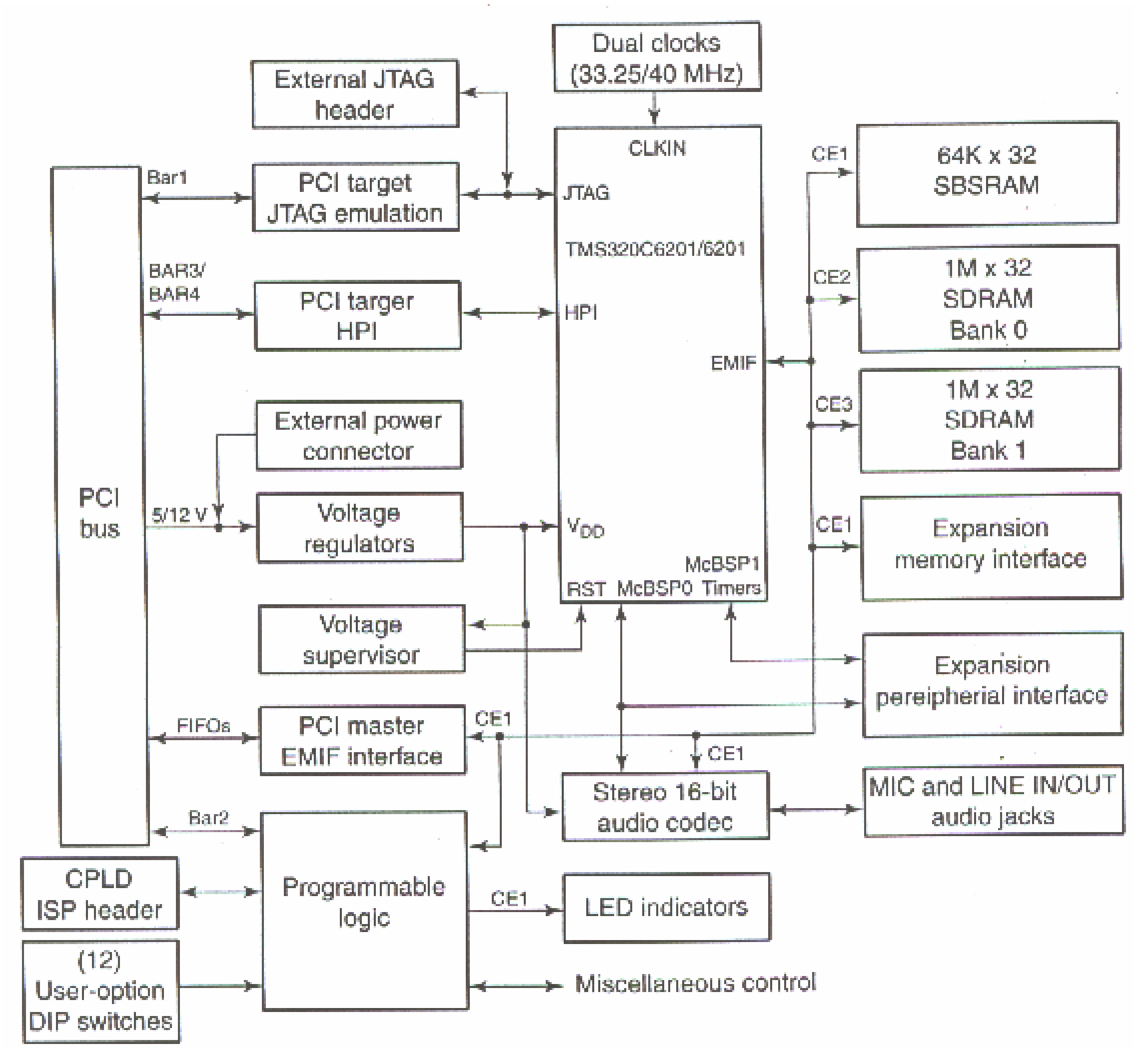

Figure 2, EVM Functional Diagram

Analog to digital conversion uses 16-bit codec format whose frequency can be sampled from 5.5 to 48 kHz. Resistant memory consists of four 1M x 16

SDRAM and one 64K x 32 SBSRAM. This type of RAM may be an advantage in our project due to its faster speed. The EVM board has on-board regulators, which provides 1.8V or 2.5V for the C6x chip core, 3.3V for the memory and 5V for audio components.

System Interface

Supply Voltage

The DSP will receive power from the +5V delivered by the computer's switching power supply. This voltage and current is regulated and is capable of delivering to the load all the power required. The selected mobile's supply voltage will not be altered when used with the SmileyBaby and will continue to be powered by batteries.

System to Mobile

The relay contacts at the output of the DSP board will be connected through hard wire to bypass at will the on/off switch within the mobile. Since we are using relays, there is no electrical connection between the actual mobile and the DSP controller.

System Software

Software Composer

The software utility that comes with our DSP chip is the Code Composer Studio (CCS). CCS provides an easy to use graphical user environment (GUI) for interfacing and debugging purposes with real-time capabilities. Programmers can easily build and debug their application programs from within a user-friendly GUI.

CCS includes the following features:

- An integrated code editor to edit C and assembly source codes
- An easy-to-use Project Manager window to build applications
- Breakpoints, watch variables, view memory/registers/stack, probe points to stream data to and from the DSP chip; all for debugging purposes
- Graphing utilities to plot signal/data
- Capability for viewing disassembled and C instructions as they execute on the chip
- General Extension Language (GEL) to create custom functions for commonly performed tasks
- Real-time capabilities: Plug-ins and API (as explained later)
- Real-time data exchanges (RTDX) between the DSP chip and the host

The CCS real-time plug-ins provide the ability to probe, trace, and monitor a program without disrupting its real-time behavior. The API utility incorporates modules that allow the user to configure global run-time parameters and properties for objects such as software interrupts, I/O pipes, and event logs. The API configuration files simplify the tasks of memory mapping and ISR vector mapping. In addition, the RTDX enables the user to exchange data between a host and the chip without disrupting the real-time operation of the program. Nonetheless, CCS includes the complier, assembler, linker, and debugger/simulator for our DSP chip. It is not only capable of converting a C or assembly file into a DSP executable file, but it is also able to understand linear assembly code.

Figure 3 illustrates the steps CCS takes to convert a source file (.c extension for C .asm for assembly, and .sa for linear assembly) into an executable file (.out extension). Compiler and assembler optimizer provide the translation of, C source files and linear assembly source files, respectively, into an object file. The linker combines object files as instructed by the command file into an executable file. The simulator can then be used to simulate the execution of the executable file on our C6201 processor.

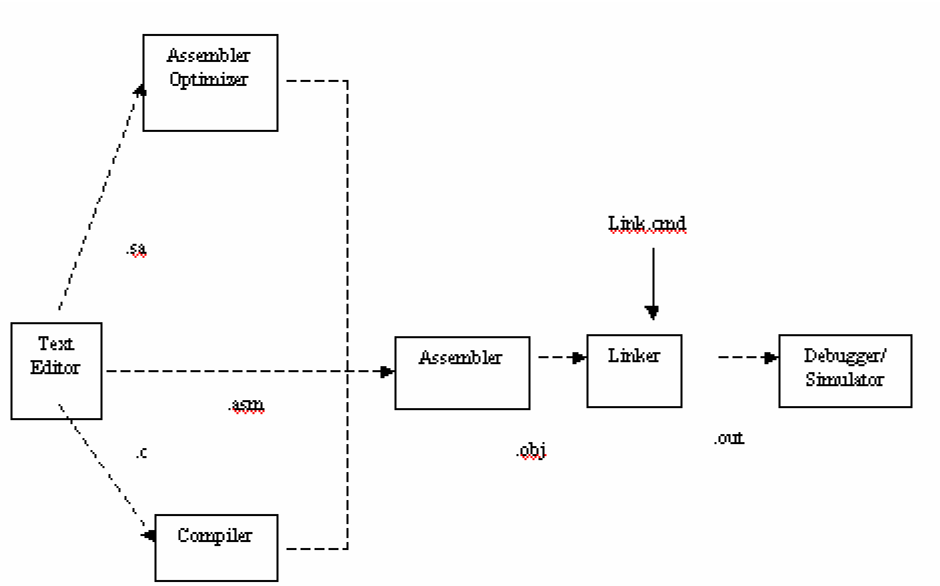

Figure 3 Code Composer Studio for C6201 DSP chip

The extensions shown in the figure above are as follows:

.c = C source file

.sa = linear assembly source file

.asm = assembly source file

.obj = object file

.out = executable file

.cmd = linker command file

The designers of LittleFellows intend to write and analyze the code in C. This is because programs in C require less effort to write, debug and analyze in comparison to programs in assembly language. However, since assembly programs are much more efficient than C programs, if our required processing rate is not met by using C language, we will switch to assembly language, and replace the time-consuming portion of our code with its equal version in assembly.

Software Flowcharts

SmileyBaby is designed to work when the baby is crying, and both the mobile and DSP are powered up. When the DSP is on, the sounds will be recorded for five seconds through the microphone for processing and analyzing. (For further information on how the DSP communicates with outside world, please refer to Figure 5.) The processor then decides whether the incoming sounds contain the content of a baby's cry. If the baby's cry is recognized, the mobile will be turned on for three minutes. Finally, the algorithm will be repeated. Refer to the flowchart Figure 4 below for a more detailed description:

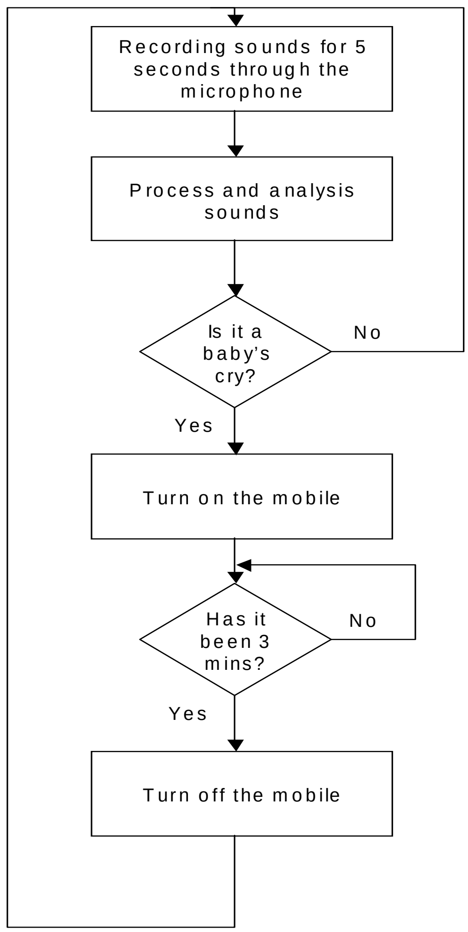

Figure 4, Description of the DSP Algorithm

The details on analyzing and processing the signal are given in the Software Implementation Plan. Please be aware that this is implemented through time and frequency domains. The on/off switch function with the mobile is accomplished through relays. Interfacing audio with the mobile represents some more sophisticated electronics. Figure 5 demonstrates how audio will be processed through the DSP. For the SmileyBaby design, only the audio input sampling section is required through the serial port, which alerts the DSP that viable information is present and is to be acted upon.

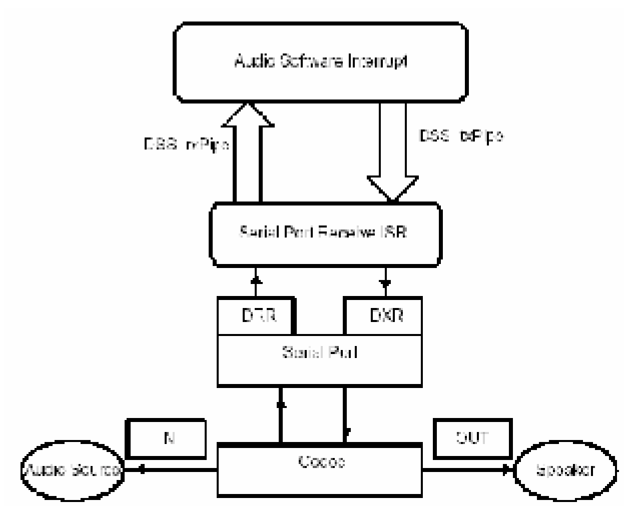

Figure 5, Diagram of the Audio Example

Software Implementation Plan

The most important section of the software for SmileyBaby is the implementation plans for discriminating the baby's cry from any other surrounding noises. Time and frequency domains illustrate distinguishable patterns of the baby's cry and that demonstrates the ability of identifying the cry, once the patterns are seen and matched by the DSP.

Time Domain

By looking carefully and analyzing the baby's cry signals in time domain, it can be shown that the amplitude of the cry signal is higher relative to other sounds. In addition, due to normal intake breath periods between each scream of the baby, an interesting pattern is formed consisting of two bell shaped signals. Note that these two are close to each other but separated by a low amplitude interval, that is the baby sucking air in. The longer interval precedes the shorter bell punctuated by a space; the period of time of the longer interval is at least four times the elapse time of the shorter bell. An example of this time domain signal is shown below:

Figure 6, A 17 Day Old Boy Crying

As illustrated in the example above, the maximum amplitude of both signals are high for better recognition of the baby's cry. The software will be written to determine a threshold voltage in order to separate high amplitude from low amplitude.

Frequency Domain

The frequency domain is more difficult to deal with because the patterns are less uniform and regular. However, the energy spectrum, (which is frequency versus time), for the baby's cry, contains valuable information. In all the cry samples, there is higher energy concentrated at high frequencies, which is not observed in other noises. For a better understanding please look at the Figure 7 as shown below:

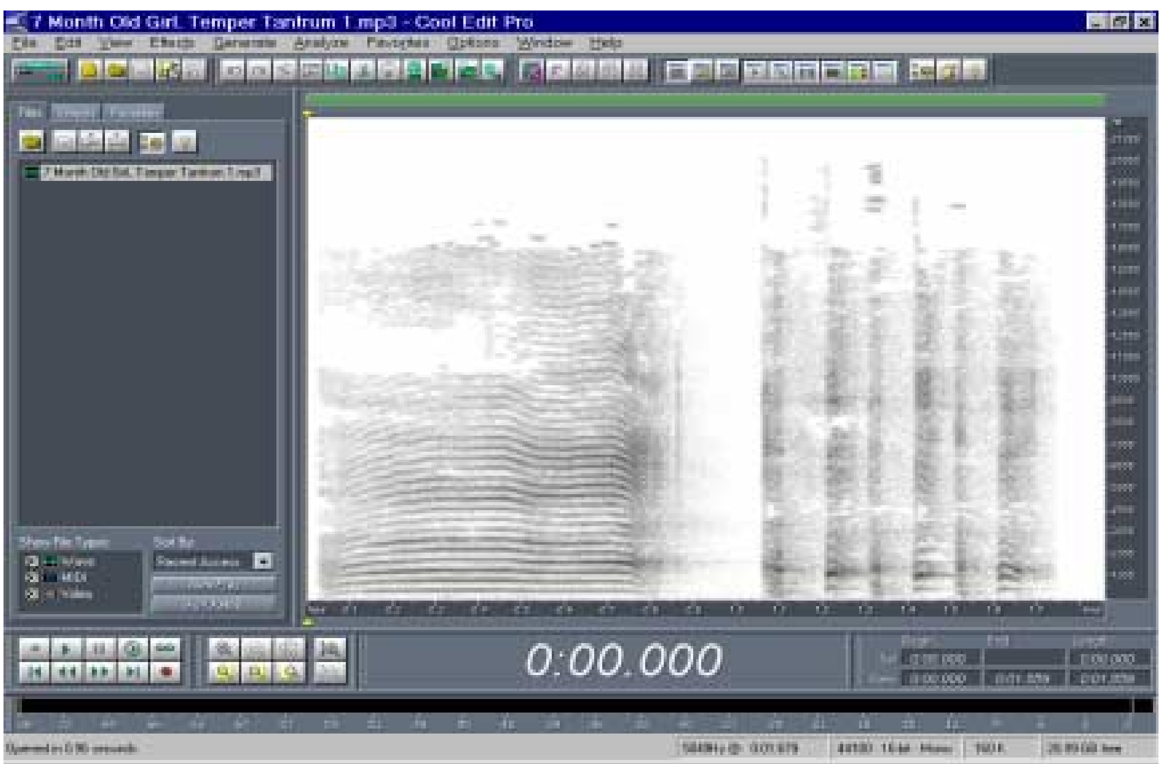

Figure 7, Energy Spectrum of a Baby's Cry

The main objective of the SmileyBaby is differentiating and classifying the baby'cry by combining the patterns contained in both time and frequency domains. The wise interpretation of the data within the domains will make this product more functional and saleable.

Test Plan

All major stages in our design process will be thoroughly and independently tested. Once we conclude that all sections are working fine independently, we will initiate the real run.

Hardware Testplan

To guarantee the functionality of our TMS320C6201 DSP chip, we need to perform some hardware testing and verifications.

Component Connection Verification

The following tasks need to be carried out by a test engineer:

1. Verify the DSP chip is properly connected to the PCI slot of the computer.

2. Verify the connection match the schematic shown in user's manual.

3. Verify the jumpers on the EVB board are selected as required in the user's manual.

These two verifications need to be done before proceeding to the second section.

DC Power Verification

The following tasks need to be carried out by a test engineer:

1. Turn on the computer, and verify the green LED of the DSP chip is on and the red LED is off.

Please note that if the red LED is on, one must turn off the computer and go back to the first stage of verification.

EVB SoftwareVerfication

In this stage of testing, we need to make sure that the driver for EVB board (TMS320C6201) is properly installed, the code composer studio works fine and the DSP and the computer can communicate (in other words, the DSP works properly).

PC Control Software

TMS320C6201 is provided with a driver that is compatible with all Windows platforms, and can be easily installed by the user. The driver facilitates the communication between the PCI bus and the EVB board. In order to make sure the driver is properly installed a test engineer must carry out the following tasks:

1. Right click on My Computer, select Properties→ Hardware→Device Manager→right click on the TMS320C6201→Properties and confirm that its driver status box reads the following: "This device is working properly."

2. Open a command window. Go to the \ti\c6000\evm6x\bin directory and type in evm6xtest. Once the testing program finishes execution, confirm that its output matches the one shown in the user's manual of the EVB board. Moreover, verify that the output indicates that the JTAG is internal. If JTAG is external, reconfigure the jumpers and redo this test.

3. Run CCS and carry out Project Management tutorial. Verify the expected output. This program confirms that CCS is properly installed and works properly.

DSP/BIOS Verification

In order to make sure that the DSP is working fine, a test engineer must perform the following:

1. Connect an Audio player (e.g. CD player) to the line-in jack, and a speaker to the headphone jack of the chip.

2. Open the audio project found in \ti\examples\evm6201\bios\audio. Run the project, and verify that you hear what your audio player is playing.

By performing this task, the test engineer verifies the fact that the stream of audio to the DSP matches the one processed and sent out by the DSP. In other words, he/she verifies that the DSP is working fine.

Software Testplan

Low Level Software Testing

Initially, a test project will be implemented to read in an audio signal and output the echoed version of the signal. It should be noted that reading, echoing and writing to output buffers are three separate modules. Each of these modules works independently and must be test separately first (before integrating them together in the test project). The figure below represents the test project block diagram.

Engineerings must test each module, construct the project and verify that the output is an echoed version of the input.

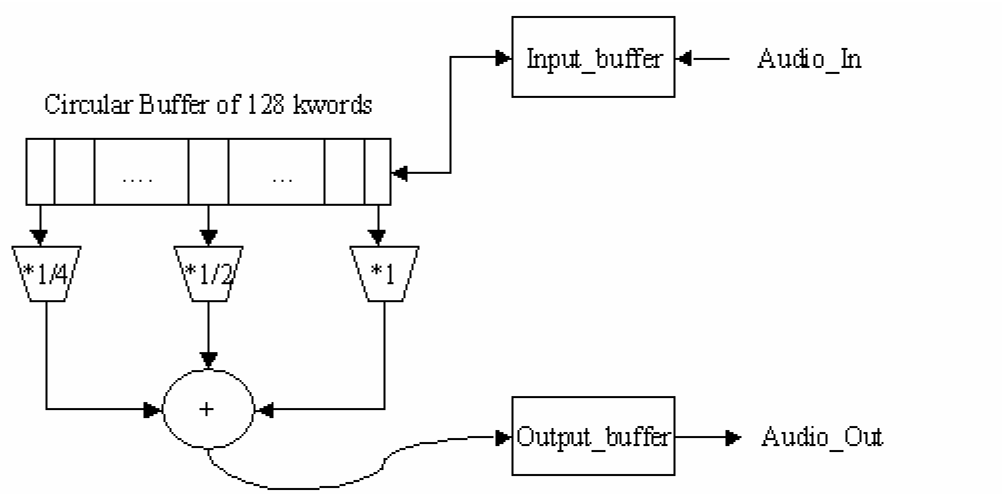

Figure 8, Test Project Block Diagram

High Level Software Testing

The block diagram below illustrates a rough high-level software design for our project. Please note that new modules may have to be added as the design process proceeds. Each block represents a separate module and must be tested prior to its use in the main project. A test engineer will have to perform the testing and verification.

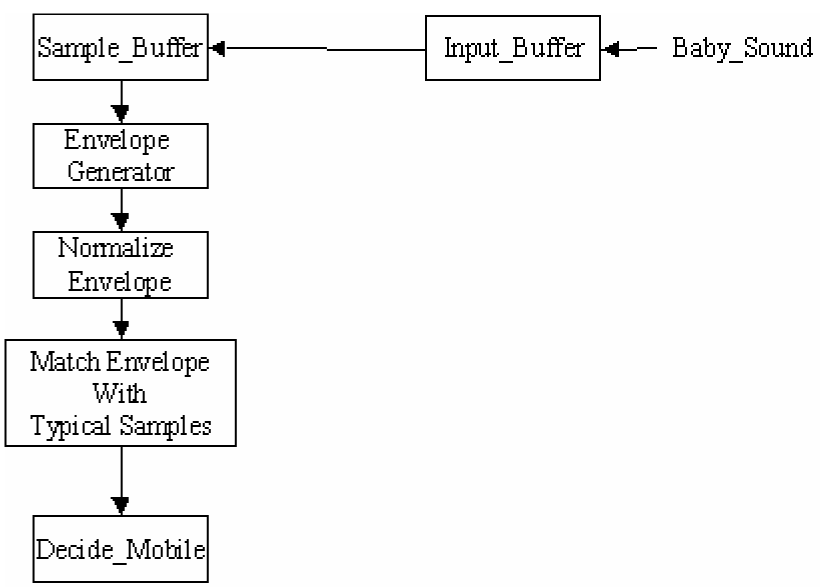

Figure 9, High-level diagram for our main project

Conclusion

SmileyBaby product incorporates a unique method of analyzing the time and frequency domains in order to control through logic external original manufactured devices in the consumer market place. This product can be creatively utilized to sample and synthesize data in a variety of industrial and commercial environments to control conditions and events. The prototyping board will use a dedicated computer connected to the DSP board, which in term will connect to the selected mobile. This system will of course be used only during the development stages. A dedicated CPU will control the DSP in a final standalone marketable product.

Appendix

Glossary

A/D converter Converts analog signal to digital signals

API Application Program Interface

Bus A set of signal lines through which the processor of a computer (or a microprocessor) communicates with memory and I/O devices.

CPU central processing unit (CPU) The combination of the register file, the control unit, and the ALU.

Control unit The part of the processor that decodes and monitors the execution of instructions. It arbitrates the use of computer resources and makes sure that all computer operations are performed in a proper order.

D/A converter Converts digital signals to analog signals

DSP digital signal processing unit (DSP) A circuit that improves the accuracy and reliability of digital communications. A DSP chip is able to differentiate between human-made signals, which are orderly, and noise, which is inherently chaotic.

GFCI Ground fault circuit interrupters are electronic devices that protect people from serious injury due to electric shock. They monitor the electricity flowing in a circuit, and if the amount flowing into the circuit differs from the amount returning, the GFCI will shut off the current. Although they prevent electrocution, there is still a risk of electric shock.

GUI Graphical User Interface

Integrated Circuit A microcontroller contains everything that is in a microprocessor and may contain memories, an I/O device interface, a timer circuit, and A/D converter, and so on.

ISR Interrupt Service Routine

Memory Storage for software and information

Microcontroller A computer system implemented on a single, very large-scale

Microprocessor A CPU packaged in a single integrated circuit

PCI PC interfacing

Program A set of instructions that the computer hardware can execute.

Polarized Plug A plug with one large or wide prong and one narrow one, it ensures that the plug is inserted correctly in a socket and reduces the risk of injury by electric shock.

RAM random-access memory (RAM) allows read and write access to every location inside the memory chip.

RTDX Real Time Data Exchange

Three Prong Plug This plug on a three-wire cord set provides a path to ground for electricity that is straying or leaking from a product. It helps to protect the equipment and can prevent electric shock.

SBSRAM synchronous burst static RAM

SDRAM synchronous dynamic RAM