Hands-Free Music Tablet

Steven Tomer	steve@wasatch.com
Nate Decker	milamberoftheassembly@yahoo.com
Group Website:	http://www.cs.utah.edu/~ndecker/CE3992/

Abstract

The typical musician handles a great deal of sheet music, especially in situations such as a concert or recital. When a musician does not play from memory in these instances, it is necessary for the user to be able to turn pages. Common solutions involve having another person turn the page (such as the interior stand-partner in a symphony) or stopping the music momentarily to turn the page. The sheer volume of material involved also can make disorganization inevitable. We propose an electronic solution to this problem—a touch-screen LCD display, with a remote control that can be operated by the user’s foot, eliminating the need to remove hands from the instrument in order to turn pages.

Introduction and Motivation

In this project we will build a tablet device that will be used by musicians to view sheet music for performances without the mess and confusion of reams of paper. The device we have envisioned will be a display with a foot pedal attached. The musician will be able to play music by reading from the display and based on inputs to the pedal either turn pages forward or backward without removing his hands from the instrument. The final demonstration for this project will be straightforward. It will be simple to show how the pedal affects the displayed pages and how to load and organize sheet music in the software.

Although this design will not ultimately be implemented, we are interested in it because it is something that does not exist in today's current market. Any person who has ever been involved in music can recognize the advantage to having a paperless and hands-free display. The reason this device doesn't exist is fairly obvious. The market for it is relatively small compared to other devices. Additionally, the cost seems excessive, a deterrent for musicians who are often viewed as destitute. However, this particular design seems tailor-made for a computer-engineering project. It has elements of programming as well as interfacing with physical hardware. We are interested in it, not so much for the marketing value, but for the comprehensive engineering aspects it represents. It also seems to be a genuinely useful project.

Project Tasks

- Hardware
  - Locate vendors for tablet PCs and purchase a cheap model. We only need basic functionality so the more simplistic the hardware, the better. Ebay would probably be a good choice.
  - Design and build a simple foot pedal interface. There are some available through various companies, but the device itself is too simple to justify the cost of buying an existing design.
  - Interface the foot pedal with the ports of the PC.
- Software
  - Write software to react to the interface that the foot pedal provides.

- Write a TWAIN interface to a scanner to allow the user to scan sheet music into the software.
- Write software to store these images in a convenient form, likely TIFF.
- Write a simple database to maintain the works that are available in the system and keeping pages from individual works together.
- Write the main user interface to display the images of the sheet music and provide simple interactive user controls.

Specific Task Interfaces

Since the display interface is built-in to the tablet itself and has internal interfaces, there is little risk of complications in that area. Our most sensitive design element is the pedal itself and how it interacts with our program. The pedal must perform three tasks. Go forward a page, go back a page, or return to the beginning of the piece. This translates into four states.

1. The musician is not touching the pedal and the current page is displayed, this is the wait state.
2. The musician is "turning the page" forward by pushing one of the pedals.
3. The musician is "turning the page" back by pushing a second pedal.
4. The musician is returning to the beginning of the piece by pushing both pedals. There could be a master pedal above two smaller ones that will simultaneously send both signals.

Four states translates into two bit encodings of data. The interface is one way, the pedal communicates to the tablet, but does not care what happens to the data it sends.

If we build the foot pedal from scratch, we could use a serial port or if possible, two separate parallel ports to provide this data. The foot pedal would be active high. It would have three wires (under the parallel implementation) running along its cord. The first wire would go from the tablet's VCC to the pedal to allow the ports to be pulled high. Pressing a button would complete the circuit and carry these high signals along the remaining two wires.

A much simpler method of building the foot pedal would be to use a pre-built component, a mouse, inside an enclosure. This has a known and simple interface (USB), and an pre-defined software interface. A three-button mouse could encode the four states without the requirement for the user to press two buttons at once. A two-button mouse could be used, however.

Testing and Integration Strategy

To reduce complexity and simplify debugging, we'd like to divide up our project into as many small pieces as possible, debug them individually, and add the debugged pieces to each other one by one. There are two obvious pieces that are immediately apparent, the hardware and software.

First, we can test the functionality of our foot pedal itself. If the pedal is made using a mouse, we can test the mouse before and after we build it into the enclosure.

If the pedal internals are created through simple electronics, we can provide power to the completed apparatus and probe the outputs to verify that they are in fact outputting the specified signals. Next, we can write a simple program that will verify it is detecting these signals. At this point we could catch possible problems like debouncing, bad encoding, or erroneous signals caused by glitches. Once we have verified that the pedal is functional and interfacing properly with a basic software application, we can proceed.

Our second portion that would be easy to isolate and test would be our ability to display stored images. Once we can show a page of sheet music on the screen, we will know that this portion is successful and is ready to interface with our pedal components.

The final phase of testing would include the integration of the hardware and software, and to determine if the user could indeed navigate the music.

Hardware

The hardware portion of this project will be done by Nate Decker. Our initial plan for the foot pedal interface was based on the fact that only four states are needed to be represented and thus only two ports were needed. We concluded that any two pins could be utilized in parallel to transmit the state encoding. However, we decided that the USB protocol would be the easiest to interface with the tablet itself through software. This would require the foot pedal to transmit a serial signal which complicates the design significantly. From the beginning, our intention to design our own foot pedal was to avoid the expensive foot pedals available commercially. Their prices can range anywhere from 70 to 200 dollars or more which was excessive considering the application we had in mind.

The engineering approach to the typical problem is to use existing components rather than reinventing the wheel. It occurred to us that a common USB mouse already has the functionality we are looking for. It has two buttons: left click and right click. These two buttons can be directly mapped to the two foot pedal functions. Mouse prices can vary a great deal, but the more expensive models often attribute their prices to increased precision and tracking. For our purposes, the mouse will not be transmitting positional data so we can use the cheapest USB capable mouse on the market. We found some available starting at around five dollars.

Obviously it's impractical to try and use mouse buttons with one's feet. We'd like to build some kind of housing unit for the mouse which would be a low tech interface between pedals and buttons. A more sophisticated design would likely use the on board mouse circuit and try to connect to the pedals directly. However, in the interest of simplicity, the casing will be physically attached to the actual mouse buttons themselves. The mouse will be the core of the housing unit and attached in some permanent fashion to the structure. The pedals will have resistance provided by springs so that a reasonable degree of force is required to use a button. Additionally, the contact between pedal and button will be separated by a spring so that excessive pressure won't break the mouse circuitry.

Alternative Hardware Sources

In the unlikely event that we are unable to make a mouse work with our system, we have alternative sources for the hardware. https://secure.nch.com.au/cgi-bin/hw.exe?class=FP

Software

The software portion of the project will be done by Steve Tomer. The major portion of the software will be written in MFC. We will build a dialog to load and display the images to the screen, and a separate dialog to display and let the user build a database to group individual scores.

The main dialog, which displays the image, will have handlers for events from the pedal. If a three-button mouse were used, three event handlers would be needed. If a two-button mouse were used, the software would need latency and state to de-bounce the state where both buttons must be used.

Individual pages of each score will be stored in a simple format, preferably TIFF. TIFF has a simple and fast library, LIBTIFF, which is freely usable. This library, originally written by Sam Lefler, is available from the following site: http://www.libtiff.org.

For convenience to the user, we will include a TWAIN interface to allow the users to scan-in music scores without the need for an additional program. The interface to TWAIN will have to be written based on the specification available at the following site: http://www.twain.org.

Individual images will need to be grouped together into scores. The scores will be kept in a simple database, which we will need to write. This database will allow the user to easily reference and find music scores. The database will have the feature to group scores together so that an entire recital or concert could be played without any user interaction other than through the pedal.

Group Communication Plan

We plan to continue to meet weekly during our class sessions. In addition we are also partners in our embedded systems class and have opportunities to discuss issues during those times as well. Since there are only two members of our group, communication problems should be minimized and all meetings will have 100% attendance. We have been currently communicating regularly via e-mail and will continue to do so throughout the course of the project.

Schedule and Milestones

February 2007 – Initial project proposal

March 2007 – Finalize design ideas and prepare final project proposal

April 2007 – Project presentation

Summer 2007 – Obtain required hardware to start building the project

August - September 2007 – Build the hardware and write the software

October 2007 – Debug and polish the project

November 2007 – Completed project

Risk Assessment

This project has little inherent risk, as the idea and implementation is quite simple. The largest risk is in building and interfacing the foot pedal. Even this is minimal if our plan to use a mouse as the basis of the pedal is successful. Writing the software seems very straightforward.

Bill of Materials

Would need to purchase:

1 Tablet PC

Individual part list for building the foot pedal TBD, but likely it would include a mouse, and a special enclosure.

Already have:

A scanner, for loading the music

Visual C++ for writing the software

Vendor List

Tablet PC – a low cost provider or possibly Ebay

Foot Pedal – CompUSA for the mouse, and probably Home Depot, Standard Electric Supply

Conclusion

This project has two main components—the control software and the pedal itself. It seems as though this would be the easy place to split the project, since there are two of us. The point of interdependency would be the interfacing between the hardware and software, and to receive the signal from the hardware.

The ability to store and move through music hands-free and paper-free makes this device genuinely useful, and a great project.