KeyStick Project Proposal

Etec 471

Robert Birch

09/11/2004

Project Proposal

Prof. Todd Morton
Introduction

After years of using computers, I started to question the fact that I had to remove my hands from the keyboard in order to operate my standard mouse on the mouse pad. I also questioned my hand position because I am prone to carpel tunnel syndrome. I will develop a new kind of input device that solves both these issues.

I have a plan to consolidate the keyboard and the functions of a mouse into one device that I can use continually without removing my hands. In this document I will refer to this device by the name KeyStick.

Description

The KeyStick (Fig. 1a) will be the same size as an average full size keyboard. It will consist of a base and two hand grip columns. The columns or “sticks” will be able to tilt in eight directions. On the far side of the columns there will be four keys (Fig. 1b) similar in look and feel to keys on an average keyboard. The computer user will use these eight keys in combination with the two stick’s positions to generate enough states to
emulate both the keyboard and mouse. When the sticks are not being held they are naturally in the “neutral” position. In the neutral position the keys on the left hand correspond to “asdf” and the keys on the right stick correspond to “jkl;”. By tilting the left stick forward just slightly (around 2 degrees) the keys now correspond to “qwer”.

There will be a mouse mode that uses the sticks themselves as the positioning input device similar to a joystick. Because it will take a user time to become proficient at using this device, I believe they should have access to their old keyboard and mouse while learning. Using the USB will allow the KeyStick to act as another input option and will not interfere with the existing input devices.

The PC will receive a key code from the KeyStick and the software on the PC will decode the meaning of the key code. This will allow the user to develop any matrix of key input and will allow for macro expansion of keys as well as useful key combinations like Ctrl-Alt-Del.

![Figure 2: Information flow](image)

**Benefits**

The KeyStick will allow people with limited arm movements to input data into the computer. I hope that this design will lessen the stress on the carpel tunnel and allow me to input data all day long without any discomfort. Another benefit will be that the user can hold the sticks and emulate the keyboard and mouse simultaneously. This is
useful when surfing the web for example because there is almost constant switching between keyboard and mouse.

Comparison

The saying about building a better mouse trap can be applied to building a better input device. There are a lot of input device companies working hard to design a better input device. SafeType (http://www.safetype.com/) sells what they claim is the “Worlds Best-selling Vertical Keyboard.” This keyboard (Fig. 3) is basically a regular keyboard cut down the middle and each hand’s half positioned vertically. The benefit of their design is that the hands are left in their prone position while typing. The prone position is the position your hand is usually in when you walk and shake hands with someone. My design also leaves the hands in the prone position. The OrbiTouch (http://www.keybowl.com) sells an input device that is referred to as a “chording
keyboard.” This means that each keystroke is simulated by one or more keys, or in this case, one or more movements. Their claim is “Typing Without Finger or Wrist Motion.” Because the KeyStick is almost a cross between these two award winning keyboards, I feel like it is a very effective design.

I have also researched nine other input manufacturers, as well as a lengthy patent search, and have come up with no input devices similar to the KeyStick.

Project Development

When developing projects I always run concurrent sub-projects. This project will be no different. The sub-projects will be as follows:

- USB connectivity
- Physical prototype
- Microcontroller programming
- Windows application for changing key maps

**USB connectivity** includes: Picking out the chip set and development system required for USB product development, learning all I can about the USB standard interface, and learning how to program the Windows driver that communicates with the USB host adapter.
**Physical Prototype** includes: Building the base that holds the cpu and switches, building the key sticks--including the input keys transferred from a junked keyboard, and designing how to gather positional data from the “key sticks”.

**Microcontroller programming** includes: Picking out a kernel scheme (Time Slice or uC,) gathering data from the physical prototype and communicating with the USB registers (endpoints.) This will also include picking out a software development system for the microcontroller, if I choose a chipset not supported in our EET lab.

**Windows application** includes: Developing a user interface, a help menu, a key map modification dialog box, a tutorial, and including the USB connectivity portion windows driver.

**Demonstrations**

My demonstration will be the final product in a box with the KeyStick, the User’s manual and the CD with the device driver on it. This product will work like most USB products. The consumer will plug in the device, Windows will prompt for the CD, and the intuitive installation program will do the work.