THE PRESSURE SENSITIVE TOUCH-PAD

James P. Williams
Gregory F. Welch

April 30, 1985

E.E.T. 454

Prof. Tom Shultz

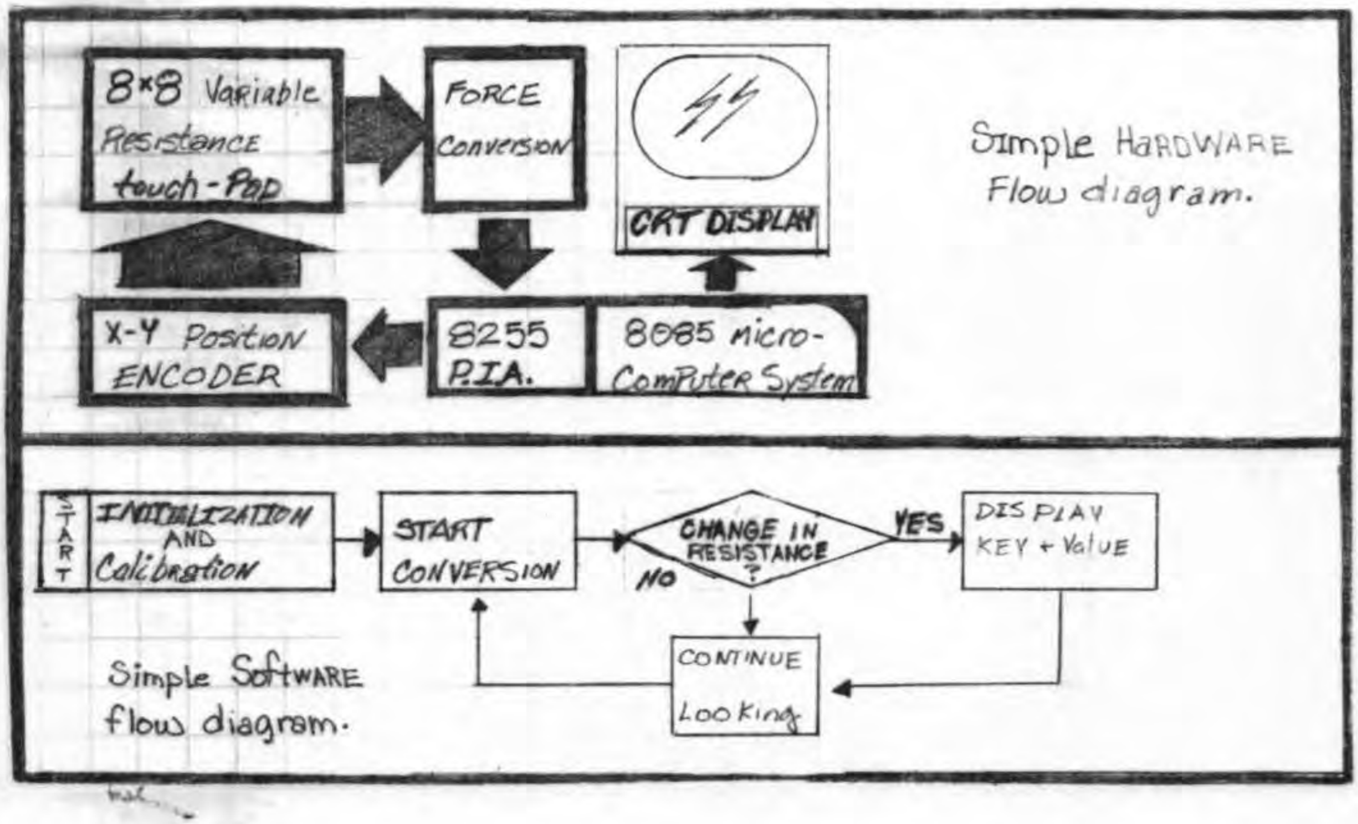
System Plan:

This project, which was designed by James P. Williams and Gregory F. Welch, will use an original design for a pressure sensitive touch-pad used to allow input to a microprocessor system.

The touch-pad will use a matrix of conductive foam squares which behave as variable resistors. Using some special scan circuitry, the software will continue to scan the touch-pad for a large change in resistance from one scan to another.

The problem faced is that of designing the hardware and the software required to scan the pressure sensitive touch-pad, and to determine (by software) whether or not a specific area of the touch-pad was being pressed.

The touch-pad should be such that a small child could use it to control a small vehicle such as a wheelchair. The positional data could represent the direction in which to move, and eventually, the amount of pressure could determine the speed at which the wheelchair moves.


8×8 Variable Resistance touch-Pad
FORCE Conversion
CRT DISPLAY
Simple HARDWARE Flow diagram.
X-Y Position ENCODER
8255 P.I.A.
8085 micro-Computer System
START
INITIALIZATION AND Calibration
START CONVERSION
CHANGE IN RESISTANCE ?
YES
NO
DISPLAY KEY + Value
CONTINUE LooKing
Simple SoftWARE flow diagram.

Introduction:

The pressure sensitive touch-pad system was designed to eliminate the need for a positive (definite) depression of a specific key to perform a task. However, the touch-pad is still set-up in a matrix fashion, so that the area of the most pressure could be determined. The purpose of this is to facilitate control of a small vehicle by children with underdeveloped fine motor skills. Such children might have problems with pressing one specific key among many, and they might not be able to apply direct force on a key.

With this current system, the user would apply pressure to a certain area of the touch-pad, and the system would recognize the key (or area), and display a number on the video display terminal which corresponds to that area.

Software:

There are several special features in the touch-pad software which require further explaination.

The first area of interest is a typed subroutine called Value. The purpose of this subroutine is to allow the software to determine the amount of resistance (relatively) that is characteristic of a certain key or area. It begins by placing the count of the key OR'd with data to initialize the hardware on an output port. This selects the key to be read, and places a high on both the address latch enable (ALE) and the start conversion (SC). The values are OR'd together because the control is all processed through the same port.

Next, the routine places the count value on the output port twice again. Both times, the count is OR'd with the data required to control the hardware. The first time, a low is placed on the ALE line, and then a low on the SC line. This begins the process of conversion.

Finally, the routine loops, checking for an end of conversion (EOC) signal. When the data is ready, it is returned through the typed subroutine.

The next area of software interest is the main section of code. The main section of code is comprised of two loops, one inside the other. The outer loop repeats until either a break is encountered, or some other means such as a reset is used to halt the microprocessor. It begins by zeroing out (so to speak) several variables to be used. It then scans the keyboard and looks for the largest difference in resistance between the present and the last scan. Then, if this difference is greater than a threshold value, the number corresponding to the area pressed is displayed on the vdt.

The inner loop repeats sixty-four times, every time it is encountered. It begins by calling the Value subroutine which returns a value between zero and 128, representing the current value of resistance for that key. It then determines if the current scan value is less than the permanent value, then the permanent value is updated to the current value. Next, it determines the difference between the permanent value and the current value. Throughout the looping, it checks to see if that difference is the greatest found yet. If so, it stores the key number and the value, otherwise, it continues on to the next key.

Hardware:

The hardware consists mainly of two parts; the touch-pad, and the scan circuitry.

The touch-pad consists of 64 conductive foam squares which are sandwiched between eight conductive strips positioned in one direction below, and eight others in a perpendicular direction above. This forms an eight by eight matrix which allows which allows the scan circuitry to determine the current resistance of any square of foam.

The scan circuitry uses an eight channel digital demultiplexer, and an eight channel analog multiplexer. The software delivers a six bit word to this circuitry, three of which are used to place a high on one of eight lines using the demultiplexer, the other three which are used to select one of the eight channels of the analog multiplexer A/D converter. This chip then converts the analog signal on the selected line to an eight bit digital word which is then made available to the microprocessor. This, although it is an actual voltage reading, is used to represent the current resistance of the selected square. The larger the number, the greater the force applied to the square, the less the resistance.

Analysis:

There were two problems encountered during the development of the touch-pad which were considered to be major.

The first problem was that of selectivity. Originally, the touch-pad consisted of one large sheet of conductive foam with a matrix of wires above and below it. The problem was that pressure in one spot might appear as pressure in several spots. This problem was solved almost completely by seperating the sheet into 64 seperate squares of foam which did not come in contact with eachother. Then to further eliminate the problem, used flexible copper strips instead of the stiff wire originally used.

The second problem was that of determining with software, what actually represented a key being pressed. The first attempt kept the values read during each scan in an array which was updated with each scan. The problem was that because the array was being updated so frequently, there was never really time to determine if the change was due to an actual depression of the pad, or some other outside event. This problem was solved by updating the array only once at the begining of the program. Later scans are compared to this scan only, and the array is only updated if the movement (resistance change) seems to be opposite to that of a depression, in other words, settling of the foam.

In the future, one might seperate the conductive foam further with strips of insulating foam. This would tend to physically support the keys surrounding the one being pressed, but still still allow the flexibility of the original design.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.