A Touch-Sensitive Computer Terminal
(Hardware and Software Aspects)
A Touch-Sensitive Computer Terminal
(Hardware and Software Aspects)

1. Introduction

This note describes an implementation of the touch-sensitive screen originally developed for the CERN SPS accelerator control system (ref. 1). The device consists of a set of capacitors etched into a film of gold deposited on a sheet of glass. Each capacitor is constructed so that when the glass is touched by a finger its capacitance changes. This change can be detected and used as described below.

This implementation was designed for use with the CERN Omega Spectrometer computer systems. Whereas the SPS method uses CAMAC interfacing, the touch screen controller at Omega is interfaced to the computers in a more generalised way as a computer terminal with the touch-sensitive areas emulating a keyboard and the television monitor behind the touch-sensitive screen acting as the terminal's screen. Thus the computer sees, from a hardware point of view, a normal VDU terminal with a standard CCITT V24 interface (line speed up to 9600 baud). In addition, the computer's operating system may treat the device as just another terminal and no special software driver is required, although a small subroutine library exists to assist in programming for the device.

2. Hardware Description

The touch-sensitive screen, with its 16 sensitive areas (or 'buttons') is connected to a detection system (ref. 1). This provides 16 outputs together with a validating strobe signal.

The outputs from this card are applied to an encoder card (DD4385) which encodes the 16 lines into the octal values 0 to 17. Provision is made for bit 6 to be permanently true at the input to the encoder which therefore outputs the octal values 100 to 117 - which correspond to the ASCII representation of the 'visible' characters @, A, B, C, ... M, N, O. The encoder's outputs together with the strobe are applied to the keyboard input of the logic card of a popular make of display terminal which provides the line serialisation and CCITT V24 interface for the computer. Hence, on touching 'button' 1 the computer will receive the ASCII character @, button 2 will give A, button 3 will give B, and so on. In addition a 'keyclick' sound is produced when any such button is touched.

In a similar way, the computer's terminal port serial output is applied to the CCITT V24 interface input, is deserialised and treated by the character generation and memory circuits of the same display terminal logic card. The video output of this card is then used to drive a nine inch television monitor placed behind the touch-sensitive screen. The sensitive areas may therefore be 'labelled' by writing text on the TV monitor in the appropriate screen positions.

Provision is made in the controller for level conversion to the balanced line transmission standard used at Omega for the connection of terminals to the computers (DD3892).
The controller's console also carries 8 normal pushbuttons (fig. 3) whose outputs may also be applied to the character encoder in such a way as to emulate the ASCII characters P, Q, R, etc. A flexible patching system also allows the output of these 8 buttons to provide instead NIM or TTL pulses or relay contact closures. Thus, one or more of these pushbuttons could provide an independent interrupt source for the computer.

3. Fault Finding Facilities

The easiest test to perform is simply to place the terminal logic card in its 'Local' mode where the characters generated by the touch-sensitive screen and subsequent encoding, being 'visible' ASCII characters, should be written on the screen. A more powerful test is to loop the serial data from output to input, thus, in addition, testing the line transmission and reception circuits.

4. Physical Layout

The controller is designed as a free-standing unit (fig. 1). Since it is normally located alongside a Tektronix T4010 graphic display, the screen was set at the same height as the T4010's keyboard and at the same inclination. Otherwise, the console's dimensions were dictated by the desire to use standard components (logic cards, power supplies, etc.).

5. Reliability

In some two years of service, no major problems have been encountered. Good ventilation of the detection circuits and the terminal logic card has been shown to be essential and some redesign of the airflow system was necessary.

The stability of the touch screen and its detection system has been proved to be very good, only requiring infrequent (approximately once per year) adjustment of its sensitivity.

Care must be taken to avoid stray magnetic fields with their inevitable distortion of the TV raster display and the consequent loss of correlation between touch sensitive buttons and their labels as written on the TV monitor.

6. Software

As mentioned above, no special software driver is required as the computer's normal terminal driver is sufficient. However, it is useful to provide at least simple subroutines which translate for a user programmer button number into X-Y position on the screen when writing the labels, and on input decode the ASCII character received into the number of the button pushed.

At Omega, PDP 11 computers are used with the RSX-11M operating system (ref. 2). Most user programmes are written in FORTRAN IV and two packages are provided to allow the FORTRAN programmer to use the touch screen controller. The simpler of these packages allows the user to specify up to 16 button labels at a time, to alter one of these at a time or all 16, to give a header line to a 'page' of 16 buttons and to pick up the number of the button selected. The second package has additional options to help in automatic
button label layout, label storage, multi-line labels and so on.

One great advantage of having the touch screen look like any other terminal is that any other terminal can take its place. For example, suppose that a programme which normally uses the touch screen has to be run on a computer with no touchscreen, or at a remote console where no touch screen is available. Since output is via the system's terminal driver, the button labels appear in the usual 4x4 format at the console (fig.3). Instead of pushing a touch-sensitive button, the user types the character which represents that button (€ for button 1, A for button 2, B for button 3 and so on). Thus, any other display (e.g. a T4010) or even a teletype-like device can be used in place of the touch screen if necessary.

7. Usage

A typical way of using the device is to select options on a tree-structured list. For example, the first page of 16 buttons may be a list of possible programmes; selecting one of them produces a list of that programme's options; selecting one of these options may then produce a list of that option's sub-parameters and so on. Where a level has more than 16 possible choices one button is reserved to invoke the next 16 options at the same level rather than a page of sub-parameters (see figs.4-6). Some of the 8 normal buttons can be assigned special meanings; for example, one could mean 'go back up one level', another could mean 'go to the top level', and so on.

Another common use is as an ON/OFF switch where selecting the button changes its state and the button label reflects its current state (see fig.7). Thus, it can be seen that we have available a 'button box' with an infinite supply of buttons.

8. Future Developments

One interesting possibility would be to add a microprocessor to the unit in order to give it some local intelligence. The sort of tasks imaginable for this microprocessor would include button legend layout, automatic flipping of the legend of an ON/OFF switch when it is selected, checking that the button selected was within the given range of that menu, and so on. (We note that these are the sort of options which must at the moment be provided by software).

Acknowledgements

We would like to express our thanks to B. Stumpe of SPS who gave invaluable assistance in the adaptation of the device to this implementation and also to the DD workshop people who built the DD models.

References

1. CERN Yellow Report 73-6 (Beck and Stumpe)
2. Introduction to RSX-11M (DEC-11-OMIEA-B-D)
Figures:

1. General view of the device
2. Internal layout (rear view)
3. View showing the screen and the eight push buttons
4. Example of a task menu - the top level, showing the main options of the task
5. Example of a task menu - the next level of the above task; having selected option E of the previous menu, this level of options allows the user to display the histograms, display their limits or clear them
6. Example of a task menu - the next level of the task; the user has selected to display the histograms and this menu allows him to choose which particular histograms to display
7. Example of a task menu - this is the menu of another task showing the use of ON/OFF switches
8. Block diagram of electronics
Fig. 5

Fig. 6
Fig. 7
Fig. 8
Block Diagram of Electronics

General strobe touch-sensitive and pushbuttons

Television monitor

Video signal

CRT

Touch-sensitive screen

Detection system

Priority encoder card

Display terminal logic card

Converter card CCITT/balanced line

Converter card TTL/NIM

Button filter and 'keyclick' generation

Patching card

Strobe

Loudspeaker

Balanced line input, output to computer

To device of choice (e.g. independent computer interrupt, etc...)

8 TTL signal output

Relay card