Cont 7/27/76 /

LA-UR -76-125

TITLE: A DEVELOPMENTAL SYSTEM FOR MICROCOMPUTER BASED INTELLIGENT
GRAPHICS TERMINALS

AUTHOR(S): Richard Kellner
Lynn D. Maas

SUBMITTED TO: GRAPHICS '76

By acceptance of this article for publication, the publisher recognizes the Government's (license) rights in any copyright and the Government and its authorized representatives have unrestricted right to reproduce in whole or in part said article under any copyright secured by the publisher.

The Los Alamos Scientific Laboratory requests that the publisher identify this article as work performed under the auspices of the USERDA.

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

An Affirmative Action/Equal Opportunity Employer

MASTER

TITLE: "A Developmental System for Microcomputer Based
Intelligent Graphics Terminals"

AUTHORS: Richard Kellner
Lynn D. Maas

TO BE PRESENTED AT: GRAPHICS '76, July 14-16

ABSTRACT

A low cost intelligent graphics terminal for developmental work consisting of a Tektronix storage tube terminal and an ALTAIR 8800 microcomputer is described. The microcomputer system is used to develop and debug customized intelligent terminal configurations and features before a production model is finalized and fabricated. This allows flexibility in terminal design and allows changes to be tried easily before final versions are built. The customized terminal features that are described are the abilities to process a segmented display file, to provide refresh of up to 130 characters or vectors, to control accessory devices such as a flexible disc and a graphics tablet, to provide data compression of over 60% on large pictures, and to provide a local debugging utility.

Introduction

At the Los Alamos Scientific Laboratory there are two time-sharing systems available to users at remote sites through a system of dedicated data communication lines. The KRONOS system is available on three CDC 6600-class machines and LTSS, the Livermore Time-Sharing System, is available on two CDC 7600 machines. The data lines from the various locations in the Laboratory are connected to the computers through a network of concentrators and switching machines. One of the KRONOS machines is also available on a telephone dial-up basis.

The majority of the data lines support a data rate of 300 baud, but there are also some 110 and 1200 baud lines. The graphics terminals used with the time-sharing systems are Tektronix storage tube terminals. Besides their ability to store alphanumeric and graphics data on the screen of the CRT, these terminals have the advantages of low cost and software support by the manufacturer.

Although the Tektronix graphics terminals provide a powerful interactive graphics capability by themselves, several features that significantly improve their performance in our environment can be provided by the addition of a small amount of local processing power. Several of these features are aimed at alleviating the problems caused by the low data rates between the host computers and the terminals. We have interfaced a MITS ALTAIR 8800 microcomputer to a Tektronix 4015 graphics terminal in order to develop some of these features.

The custom features that we have experimented with on our developmental terminal are data compression, menu refresh, control of Tektronix accessories, handling of segmented display files, and debugging. A custom terminal for a specific user application can be constructed that would have one or more of these features incorporated into it.

A terminal that could handle similar functions was previously described in [3].

Custom Terminal Features

Because of the low data communication speeds involved, complex pictures take a relatively large amount of time to paint on the CRT of a terminal. A draw takes from two to five characters to transmit, depending on the length of the draw and the resolution being used, and a move takes from three to six characters. Since they are very special, we are neglecting the cases of short horizontal draws or moves, which take one or two characters, respectively.

At 300 baud, 15 vectors at most (including dark vectors) could be sent each second. However, the terminal does not receive

BLANK PAGE

data continuously because of limited size system output buffers, time-share delays, and various delays in the data concentrators. Therefore, in reality, the effective data communication rates are lower than the actual line rates being used. For medium size pictures of about a thousand vectors, 1200 baud is a usable data rate with our time-sharing systems. For large pictures, the drawing time becomes excessive at 1200 baud. Because of the various system delays mentioned above, normally we cannot achieve higher than 1200 baud effective data rates with the LTSS system even if a terminal is run at its maximum speed of 9600 baud. Therefore, data compression is very desirable, especially for those applications that involve large pictures.

Typically, our pictures are composed of sequences of vectors that have the property that endpoints of successive vectors in a given sequence do not change by very much. This property can be used as the basis for a data compression scheme in conjunction with the Tektronix graphics terminals.

The stand-alone Tektronix graphics terminals are addressable in terms of absolute screen coordinates only. For the typical case of 1024 x 1024 resolution, this implies that it takes 20 bits of information to address a given point on the screen of the terminal. Normally these 20 bits are sent to the terminal as four ASCII characters of eight bits each, with three bits of each character being redundant information. In the cases where the low or high order five bits of Y data or the high order five bits of X data have not changed from those of the last point, they need not be sent again because they are remembered by the terminal. In this sense, two of the three bits mentioned above are not redundant because they are used to indicate whether a character represents X or Y data, and low or high order bits. Thus, a stand-alone Tektronix graphics terminal has a certain amount of data compression built into it.

However, for pictures that have the property mentioned above, it is desirable to be able to use relative coordinates for addressing points. If endpoints of successive vectors in a sequence have not changed much, it follows that it does not take many bits of information to represent the change; but, some information must also be present that tells how many bits of data correspond to a given point.

There is more than one way to implement the above ideas. The amount of data compression attainable with a given scheme is picture dependent so that an adaptive scheme would seem desirable. However, a fixed scheme can be implemented that works well for a variety of pictures, yet still gives very good data compression for pictures composed mainly of very short vectors.

The method operates on sequences of vectors such that in a given sequence, the endpoints of successive vectors do not change by more than a fixed amount n (which can vary from sequence to sequence) in absolute screen coordinates. The data is packed as

n-bit relative coordinate information into 8-bit characters for transmission to a terminal. These sequences, with their corresponding n, are recognized by the terminal by means of certain control characters.

For a given picture, n should be chosen as small as possible in a given sequence; otherwise, a loss of data compression would result. Also, the number of sequences in a picture should be minimized because of the overhead of the control characters that must be sent to signify a sequence. These two conditions are not always compatible, so that a compromise must be reached.

A fine mesh is an example of the type of picture for which the method works well. Meshes typically arise when partial differential equations are numerically solved by a finite element or a finite difference scheme. Usually we can take $n = 5$ or 6 for these pictures and obtain up to 60% data compression compared to the usual Tektronix data format.

A disadvantage of the above data compression scheme is that if there is a transmission error in one of the compressed sequences, the picture can become greatly distorted. If one of the packed data characters is received incorrectly, the portion of the picture represented by the rest of that sequence will be translated from the correct version. This does not happen with absolute coordinate data. If a control character is missed, the result is much worse. Thus, an error detection scheme might be a worthwhile addition to the data compression scheme, depending upon the quality of the data lines between the host computer and the microcomputer.

A capability that is very desirable in certain applications is the ability to refresh selected information on the screen of the terminal. For example, there might be a large amount of stored picture information on the screen while only a menu is being refreshed. If the menu were stored and the screen erased every time the menu area filled up, a large amount of time would be spent needlessly in repainting the picture information.

In addition to storage capability, the 4015 has a limited refresh capability. For hardware characters, the refresh limit is 4000 characters per second, which is about 130 characters at a 30 Hertz refresh rate. Vector refresh is limited by the line rate between the microcomputer and the terminal, but in no case can it exceed 5000 inches per second. Tektronix sells a custom product "display buffer" that can be used for refresh purposes independently of a microcomputer [see 5]. This buffer can drive the terminal at the full refresh rate. However, the display buffer is almost as expensive as the complete microcomputer system. On the other hand, the display buffer plugs into the terminal's minibus, making a more compact arrangement.

Therefore, depending upon the application involved, it might be desirable to use the display buffer alone, the microcomputer

alone, or both the display buffer and the microcomputer. If the microcomputer is used alone for refresh, a high speed interface (TTY-port interface) must be used instead of a RS232 interface to take full advantage of the terminal's speed. The display buffer might be used alone for those applications where the added capabilities of the microcomputer are not needed. The microcomputer might be used alone for those applications requiring several custom features. Both the display buffer and the microcomputer might be used when it is inconvenient to have a TTY interface on the Tektronix terminal or when the microcomputer would be overtaxed without the display buffer.

The microcomputer is well-suited to use as a controller for a graphics tablet or a flexible disc. Our time-sharing systems will accept input data strings only if they are less than a certain length and conform to a certain format. However, it is desirable to be able to send large amounts of data from a tablet or disc to the host computers. Consider the case where one wants to draw curves on the tablet and have the coordinates sent to a host machine. This is useful, for example, in interactively solving boundary value problems for partial differential equations.

A Tektronix tablet has a mode in which it continuously samples the pen position and generates coded coordinates. Our time-sharing systems will not accept these continuous strings of data, however. But, by having the microcomputer store the data, block it into suitably sized and formatted blocks, and ship the data to the host one block at a time upon prompts by the host, the continuous mode of the tablet becomes usable. Communication between the tablet and the microcomputer is accomplished by means of certain fixed control characters that the tablet is equipped to generate for just these purposes.

The Tektronix flexible discs are equipped to send data to a host in a prompted block mode if the data on the disk has previously been assembled into blocks of the appropriate size, separated by carriage return characters. Clearly, there are cases when the data will not have this format, and the microcomputer can be used to send the data in these cases as outlined above.

The microcomputer also makes feasible the use of a flexible disc or a graphics tablet of another manufacturer to obtain special features or lower cost in a specific application. The only requirement is that the microcomputer be equipped with an I/O port with the appropriate kind of interface to the disc or tablet. Much of the same software in the host computer that is used for the Tektronix disc or tablet can be used for another manufacturer's disc or tablet. The microcomputer handles the conversion between the various control sequences, block sizes, and, in the case of the tablet, coordinate codes.

The ability to support a segmented display file [see 2] that is stored locally at the terminal can be useful in those applications where only part of a picture changes between

viewings. Only those segments of a picture that change between viewings are retransmitted from the host. In storage mode, the 4015 is capable of generating 1000 hardware characters per second. The vector writing rate is the same as in refresh mode. Therefore, great savings are possible in data communications time and thus also in picture repaint time. The segmented display file can be stored on a flexible disc or in the microcomputer's memory. The Tektronix flexible disc interfaces directly to the terminal's minibus and can transmit data to the terminal at a rate of 6000 baud. If the memory of the microcomputer is used for storing the display file, then DMA to the terminal would be a desirable feature.

Segmentation is handled with a minimal amount of software in the host computer, in addition to the Tektronix software. These additions are in the form of a library of FORTRAN subroutines. These subroutines include open, close, append, delete, paint, display, and undisplay. "Open" and "close" are used in conjunction with each other to define a segment and to cause it to be stored. Segments are only stored in the memory of the microcomputer. "Append" is used to add to the contents of a segment and "delete" is used to delete a segment. A display flag is associated with each segment. When the "paint" subroutine is called, all segments that have their display flag set are drawn on the terminal. The display flags can be set and reset by the "display" and "undisplay" subroutines, respectively. The memory of the microcomputer is managed by a linked list scheme.

A related function is storing display subroutines [see 2] locally at the terminal. This has obvious application to logic or electronic circuit design. The display subroutines could be descriptions of the commonly used circuit symbols in relative coordinates. Then a circuit design or analysis program in the host computer could display a symbol by transmitting to the terminal a symbol identifier and positioning data, not a detailed description of that symbol in absolute coordinates. Any vectors that lie partially or totally outside the viewing area after positioning are forced to the boundary. Clipping will be provided in later versions.

The microcomputer can serve as a very useful debugging utility. For example, character sequences that produce very strange behavior may be sent to the terminal. When this happens because of some software or hardware error, it can be very difficult to locate. Such things have happened to us because of system errors on both time-sharing systems. The microcomputer can track such things down by triggering on specified character sequences and by storing the transmitted data bordering on the trigger points. The stored data can then be sent one character at a time to the terminal, thus trapping the offending character(s) without manually searching through and decoding a paper tape full of hundreds of characters.

The microcomputer can also be used as a diagnostic tool to check out graphics terminals. A program written to sequentially exercise all of the various modes of the terminal can aid greatly in locating hardware problems.

System Block Diagram

Hardware and Software of the Developmental System

The technology for implementing the functions we have described with a microprocessor has existed for several years. However, it is only recently that hardware prices have dropped to the point where we could consider putting such a system on every Tektronix graphics terminal at the Los Alamos Scientific Laboratory that could potentially benefit from it. Our philosophy has been to purchase, rather than design, hardware whenever possible. Furthermore, we program in a high level language, PL/M [see 1], to keep software costs to a minimum, except when the executable code is too slow.

Our developmental system consists of a MITS ALTAIR 8800 microcomputer and a Tektronix 4015 storage tube graphics terminal. The ALTAIR microcomputer uses an INTEL 8080 8-bit microprocessor, which executes all instructions in 2-9 microseconds and can directly address up to 65K of memory. A bus structure is used on which the I/O lines, the memory address lines, control signals, power supply voltages for TTL and MOS circuits, etc. are available. A large number of accessory boards that plug into the bus are available from MITS and other manufacturers. We have 9K of 8-bit word volatile semiconductor memory, 4K of EROM nonvolatile memory, two I/O channels with RS232C interface, and two I/O channels that can interface to either RS232C, TTL levels, or TTY 20 milliampere current loop standards. The complete microcomputer costs under \$1600. Both volatile and nonvolatile memory is available for less than 2/3 cents per bit on 4K memory boards. A real time clock, vectored interrupt capability, and DMA would enhance our abilities to perform the functions that we have described, and these will be added to the system as they become commercially available.

The I/O boards can accept any of the standard ASCII characters with or without a parity bit. However, the boards had to be modified to enable them to transmit and receive a "break" control signal, which is used by both of our time-sharing systems.

Our Tektronix 4015 is equipped with a 4921 flexible disc, a 4093 graphics tablet, and a Tektronix special product display buffer. The 4015 also has an enhanced graphics module that provides features useful in some applications, such as 4096 x 4096 point addressability and hardware dashed and dotted lines.

Source code for the microcomputer is written in the PL/M language. PL/M is similar to, but simpler and more systems-oriented than, PL/I and can be learned in several days by anyone who is familiar with a high-level programming language. We have a PL/M cross-compiler implemented on both the KRONOS and LTSS time-sharing systems. There is also an 8080 simulator that can be used, among other things, for timing measurements on portions of code. A cross-assembler is available that we use for optimizing code that is too slow from the compiler and for writing heavily used subroutines. All of the above software packages are INTEL products that are written in FORTRAN.

Once the executable code has been generated on one of the time-sharing systems, we "down line load" it via the data communication lines into the microcomputer by means of a loader permanently stored in EROM. The code can then be tested and changes can be made either by keying them in directly at the microcomputer or by changing the source code on the time-sharing system. When a code is completely debugged, it can be put in EROM for use in a dedicated microcomputer system. Control character sequences are used to switch between the various modes of the microcomputer system.

The down-line loader has simple checksum error checking and handshaking with the software in the host computer. It requires about 600 eight-bit words of storage for the program itself. Programs for refresh, segmentation, controlling peripherals, and debugging all require similar amounts of program storage. Programs for data compression and subroutines are longer, but the actual lengths can vary greatly depending upon the exact implementation.

Conclusion

An ALTAIR 8800 microcomputer connected to a Tektronix graphics terminal provides a low cost intelligent graphics terminal that can handle the functions described in this paper. Software costs are minimal because the programming for the microcomputer can be done in a high-level language. Also, we have only implemented functions that involve a minimum of software changes in the host computer to the previously existing software.

There are several functions that would be useful but would involve major software additions and changes in the host computer. For example, conic sections could be sent by the host computer in parametric form and decoded by the microcomputer. However, applications software would have to be written that would take advantage of this capability. This is a nontrivial task if general purpose codes are desired. A data compression scheme for text, such as described in [4] is another example.

Several versions of BASIC are available from MITS and other manufacturers for the ALTAIR 8800. These enable the terminal to be used in a stand-alone mode for problems that do not require the computational power, software libraries, and data bases of large time-shared machines.

BIBLIOGRAPHY

1. INTEL - "A Guide to PL/M Programming," Rev. 1, Intel Corp., Santa Clara, Calif., September 1973.
2. W. Newman and R. Sproull - "Principles of Interactive Computer Graphics," McGraw Hill, 1973.
3. S. Pardee, F. Rosenfeld, and P. Dowd, "GC101 - A Remote Time-Share Terminal with Graphical Output Capabilities," IEEE Transactions on Computers, V. C-20,8, pp. 878-881, August 1971.
4. J. Sullivan - "Text and Data Compression," Union Carbide Corp., Report UCCND-CSD-INF-51, August 1975.
5. TEKTRONIX - Display Buffer Instruction Manual, Tektronix, Inc., Beaverton, Oregon, 1975.