

An intelligent experimental station controller for human experimental psychology

K. J. BURKHARDT and J. PALMER
University of Washington, Seattle, Washington 98195

and

W. WADDINGTON
IKON Corporation, Seattle, Washington 98195

The purpose of this paper is to describe the intelligent experimental station controller used in the extensible multiprogramming system for experimental psychology (EMPP) at the University of Washington. The EMPP system is an integrated hardware and software system designed to simplify the development of on-line psychological experiments. An intelligent graphics control system is one of the most important aspects of the system. The controller oversees the operation of eight independent experimental stations, each containing a cathode ray tube (CRT) terminal and a keyboard console. In addition to conventional computer interface functions, the controller contains an extensible hardware character generator which allows dynamic selection of the display character set. This paper describes both the hardware and software features of the system.

Display requirements for human experimental psychology are often significantly different than those encountered in more conventional data processing environments. In addition to conventional requirements for strings of alphanumeric characters and for text, psychology often requires geometric forms, dot patterns, and other graphics-oriented features. Commercial graphics-oriented display terminals are often not appropriate because many experimenters are concerned with controlling display onset and offset times to within a few milliseconds. The solution in many cases has been to select a general-purpose CRT and then, by software and hardware means, customize the device to the specific application.

This paper describes a technique for interfacing a set of general-purpose CRT terminals to an on-line laboratory computer. The first section of the paper describes several of the design aspects of the system. The second section describes the operation of the system hardware, and the third section describes the software that has been developed to support the terminals.

DESIGN CONSIDERATIONS

There are principally three factors that must be considered when configuring a CRT system for experimental psychology: (1) the characteristics of the cath-

Development of the EMPP system was supported by the National Institute of Mental Health, Grant MH 21795, Earl Hunt, principal investigator, and the National Institute of Education, Grant NIE-6-74-0104, Clifford Lunneborg, principal investigator, to the University of Washington. K. J. Burkhardt's present address is Department of Electrical Engineering, University Heights Campus, Rutgers University, New Brunswick, New Jersey 08903.

ode ray tube, (2) the method of character generation, (3) the method of interface to the host computer.

CRT Characteristics

A visual display on a CRT consists of a set of illuminated points on a phosphor-coated screen. The persistence of these points is a function of the decay rate ("speed") of the phosphor. The "faster" the phosphor, the greater the refresh rate required to maintain an illusion of a steady picture. Phosphor decay times range from several milliseconds for low-persistence phosphors, such as the P-11 (generally available as an option from CRT vendors), to virtually as long as required for storage tubes. In the latter case, internal control logic maintains a display until externally cleared.

Sperling (1971) and Van Gelder (1972) have pointed out that high-speed phosphors should be used in applications where millisecond timing accuracy is required. Slow phosphors make a system unsuitable for brief presentations because short display times are impossible. Many storage tubes are also inappropriate in this case because erasing a storage tube often causes a flash to persist on the screen for as long as ½ sec.

Other criteria that must be evaluated when selecting a CRT are grid size (number of phosphor points on the screen), screen size, and the efficiency of the phosphor (Sperling, 1971).

EMPP experimental stations contain a Tektronix 604 monitor scope equipped with a P-11 phosphor that is refreshed at a rate of either 50 or 100 times/sec.

The 604 was selected for several reasons: (1) It is a standard "off the shelf" product. It had a relatively fast delivery time (3 months) and has proved to be easy to maintain. (2) The scope can be equipped with a bright

and fast P-11 phosphor. (3) Resolution is excellent for both graphic and alphanumeric displays. (4) The scope can easily be interfaced with standard D/A converters. (5) It is relatively inexpensive. (Cost is approximately \$650.)

Method of Character Generation

Character generation refers to the procedure used to convert internally coded data to a set of coordinate points corresponding to the desired visual stimuli. The method of character generation is primarily a function of environment. Hardware character generation is generally significantly faster than software techniques. Software, however, tends to be less expensive and more flexible than hardware. In situations where all characters are to be the same size and the same type (e.g., standard ASCII), the best solution is often to use a hardware character generator. If graphics or special characters are needed, a better solution might be to implement the character conversion in software. Gillman and Buckley (1973) have surveyed several techniques for software character generation, although their arguments with respect to the limitations of hardware character generation are somewhat outdated. With recent advances in integrated circuit technology, in particular the introduction of microcomputers and the decreased cost of semiconductor memories, it is feasible to take a hybrid approach to solving the character generation problem. The EMPP system used both special-purpose hardware and software to obtain the speed of hardware with the flexibility of software.

Computer Interface

The method of communications between the selected CRT terminal and the host computer is one of the most important features of any display system. If communications facilities are not properly designed, it may be impossible to operate the terminals to the desired specifications. If the required refresh rate cannot be maintained, the quality of visual displays can be seriously degraded. Communications requirements are strongly affected by the type and number of terminals selected for a system. Communications between CRT terminals and the host computer generally take one of three forms: serial communications (via RS232C or current loop), parallel direct input/output (DIO), or parallel direct memory access (DMA).

Serial interfaces are useful when speed of presentation is not critical and when terminals perform the character generation. DIO interfaces can also be used in these situations and when sending coordinate data to one or two displays. In situations where speed is important, direct memory access is the best alternative. Data (either coded or in coordinate form) are fetched directly from the host computer's memory and transmitted to the terminals. A potential problem, even with DMA schemes, results when there is an attempt to hold refresh data in the host memory. For example,

a system with eight displays, each displaying 2,000 points with a refresh rate of 50 Hz, would require data transmission at a rate of 800 KHz. This is greater than the bandwidth of many minicomputers.

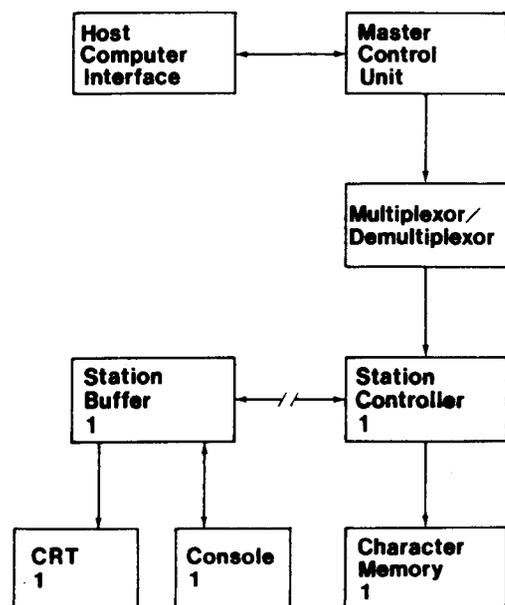
The technique used in the EMPP system is to transmit coded data via the DMA channel and store the refresh data in memories local to the displays. Data transmission occurs only when a new display is initiated. Using this technique, data transmissions are in short bursts (less than 8 msec), at a rate of 250 KHz, which is well within the bandwidth of the NOVA.

HARDWARE DESCRIPTION

Figure 1 is a block diagram of the experimental station control system. Communications between the controller and the host computer (a Data General Nova 820) are via a computer interface functional unit. Data transfers between the host and the controller are via two data paths. Direct input/output (DIO) is used for transmitting commands to the auxiliary system, while the direct memory access (DMA) channel is used primarily for the transmission of data.

Central control for the experimental station controller is by the master control unit. The master control unit decodes input commands, performs hardware character generation when required, and selects specific stations for requested input/output operations. To a large extent, the master control unit can be viewed as a programmable central processing unit.

The multiplexor/demultiplexor is used for selecting appropriate data paths between the master control unit and the individual station controllers. It is purely a pas-



Experimental Station Controller

Figure 1. Experimental station controller hardware.

sive component and is controlled by the master control unit.

The station controllers are used for loading the local display memory buffers and for controlling refresh operations at each of the stations. The station controllers transmit x and y coordinate pairs to the station buffers, which contain the digital to analog (D/A) converters and other control logic required to interface with the analog CRTs and the consoles.

From a conceptual standpoint, the experimental station controller can be viewed as a hierarchical network of nine computers; one central control processor (the master control unit) and eight independent slave processors (the station controllers). The control processor transmits commands and data to the slave units which then perform the requested operations independently. Interference between the station controllers occurs only in situations where there are simultaneous requests to reload display memory buffers. The worst case delay is 8 msec/conflict.

Operational Characteristics

From a functional standpoint, the system performs four basic operations; (1) keyboard input, (2) relay output, (3) CRT update and refresh, and (4) hardware character generation.

Keyboard input. The system has the ability to monitor the status of eight control switches located on each of the input consoles. Input may be accomplished on either a polled (by program) or an interrupt basis. Each keyboard is capable of interrupting the host central processing unit (CPU) when any key changes its state. A resident program can then determine which keyboard caused the interrupt and which key was pressed (released). Responses from voice keys, telegraph keys, or other special devices connected to the keyboard system can also be recognized.

Relay output. Eight digital outputs are available at each station for latching relays or solid state switches to perform functions such as controlling random access slide projectors or turning on tape recorders. Relay output is accomplished through standard host I/O instructions by first selecting the appropriate output station and then transmitting an 8-bit data word. The actual data transfer uses the same data path inside the station controller as the CRT refresh data. The transfer is performed on a "cycle stealing" basis that is transparent to the host machine. Relay data is latched in the station buffer and remains unchanged until the next relay is encountered.

CRT update and refresh. CRT update and refresh are the functions that involve the bulk of the control system hardware. In order to minimize the processing burden on the host computer, all refresh operations are performed by the station controllers. Each station is assigned a dedicated 2K by 16-bit refresh memory. The data contained in these memories correspond to the x and y coordinates of the desired displays. Re-

fresh operations are at a rate of either 50 or 100 Hz (switch selectable). Prior to any refresh cycle, the host can elect to: (1) continue refresh for a given station, (2) turn off any or all stations, or (3) load a memory with new data.

Memory load (update) data are taken from the host computer memory via the host DMA channel. Data may consist of coded x and y coordinates, ASCII characters, or special-purpose graphics characters. In the former case, data are transmitted directly to the appropriate station controller via the master control unit. In the case of ASCII or special graphics characters, input is transformed into x and y coordinates by a hardware character generator within the master control unit.

Hardware character generation. One of the most powerful features of the experimental station control system is its ability to define an arbitrary character set and to convert character codes transmitted from the host computer to x and y coordinates for display operations. This feature greatly reduces burden on the host, as well as increasing the accuracy of timing measurements.

To understand the capabilities supplied by the hardware character generation facilities, it is useful to describe the operation of the master control unit (Figure 2) in detail. The master control unit consists of five components: (1) instruction control logic, (2) the base register, (3) an adder, (4) the output register, and (5) control memory. The instruction control logic generates the control signals required for the operation of the master control unit and for communications between the host computer and the station controllers. The instruction control logic recognizes commands originating in the host computer, in the station controllers, and in the control memory.

The base register is a 16-bit holding register that contains a copy of the present position of a pseudocursor. Together with the adder and the control memory, the base register is used to calculate x and y coordinates for individual characters.

The output register is a 16-bit holding register that contains the x and y coordinates to be sent to the selected station.

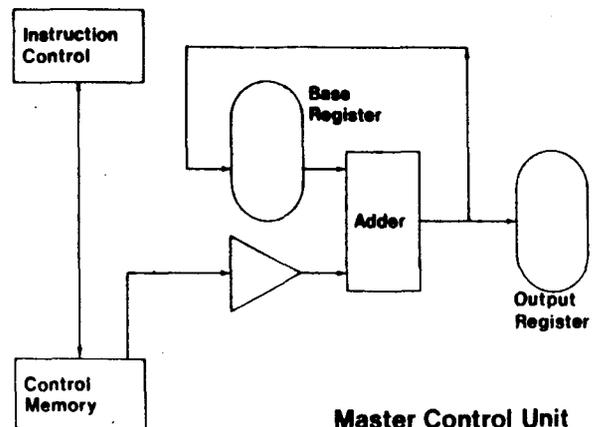


Figure 2. The master control unit.

The control memory is a 2K by 16-bit semiconductor memory that contains the instructions that define the character set recognized by the hardware character generator. Instructions exist for initializing the base register, for directly setting the output register, and for adding or subtracting 4- or 8-bit increments to the base register and output register for specifying cursor movement on either the x or y access.

The technique used for the generation of characters is very similar to standard techniques used for software character generation. An 8-bit character is used as a branch address in the control memory, and then a sequence of instructions defining that particular character is executed. The primary advantage over standard software character generation is that the master control unit has been specifically designed with character generation in mind, and thus the instruction set is significantly faster than a standard minicomputer instruction set. Instruction execution time is 4 microsec, with an average character definition time of approximately 80 microsec (assuming 20 dots are required for an individual character). A similar algorithm implemented in software on the NOVA 820 requires approximately 1.2 msec. Another advantage is that character generation is external to the computer and the only host overhead is the time required to perform a direct memory access for a single character.

There are few limitations on the type of characters that can be defined using the hardware character generator. The only constraints are that the instructions required to define the desired character set fit into the 2K (expandable to 4K) control memory and that there are fewer than 512 elements in the character set. The standard EMPP character set consists of upper- and lowercase ASCII in two sizes, graphics control characters (explained later in this paper), and approximately 10 other special characters.

Characters can be added to an existing character set simply by appending the appropriate instructions to a control memory definition file and reloading the control memory. Although only 8 msec are required to reload the control memory, there are presently no software facilities for redefining the character set while the system is on-line. Experimenters who have a need for special characters add them to the standard character set by means of an off-line utility program.

SOFTWARE DESCRIPTION

To fully utilize the hardware facilities supplied by the experimental station controller, it was necessary to develop a series of powerful, yet easy to use, input/output routines. The next section of this paper describes the operation of the station controller from the experimenters standpoint.

Programs in the EMPP system are written in the higher level language, PL/E. PL/E is an algebraic lan-

guage designed primarily for human experimental psychology. The main features of the language are the ability to process string or numeric variables, to control the external equipment in the system, and to control the operation of a large number of subjects simultaneously. Experimental programs consist of command statements, assignment statements, and conditional statements. A command statement is a natural language like directive such as ADD C to Y STORE in Z. An assignment statement is an algebraic representation of one or more command statements. Conditional statements are two-tailed IF-THEN-ELSE statements. For a more detailed description of the types of commands available in the EMPP system, see (Burkhardt, 1976).

One of the major differences between PL/E and conventional programming languages is that the EMPP system is capable of simultaneously controlling multiple experimental stations, where the same experiment may be performed asynchronously at each of the stations or where stations may be totally independent in the sense that as many as eight unrelated experiments can be performed simultaneously. PL/E programs are written as if only a single subject could participate. At run time the EMPP operating system determines which stations have subjects and causes a copy of the experimental program to be executed at each of the appropriate stations. Through the use of FORK and JOIN instructions, the programmer can specify whether a program segment will be executed once for all subjects or once by each subject. The "global" mode of execution is used primarily for initializing the data base and for operator communications. The actual experiment generally takes place in the "local" mode.

PL/E command syntax is the same whether a command is issued in the global or local mode. All synchronization, scheduling, etc. is totally transparent to the programmer. The same experimental program can be used to perform an experiment with a single subject or with as many as eight subjects. The only difference between the two cases is in operator communication with the program load routing.

Formatted Scope Output in PL/E

PL/E input/output commands allow I/O to any device to be rigidly defined or relatively free of fixed format constraints. Scope commands have the following format:

WRITSCOPE FORMAT-LABEL. ARGUMENT-LIST

The format label refers to a nonexecutable description of the desired operation. The format referenced by the label may be predefined or it may be passed in a string variable. The argument list corresponds to the parameters or variables to be output to the scope. Each argument corresponds to a descriptor contained in the format string. Numeric values are automatically con-

verted to ASCII before output. The selection of which scope will receive the display is determined by the run-time operating system.

Format statements. Format statements consist of a string of format commands. A format command consists of a format character and a format argument. The format character defines the operation to be performed and the format argument defines the scope of the operation. For example, "X<5>" informs the scope control software that five spaces should be added to the output string.

In defining the various format commands, an attempt was made to keep them compatible with FORTRAN. Aside from the special commands required for scope control, the primary difference between FORTRAN and PL/E format commands is that PL/E data fields may be of variable length. For example:

```
WRITSCOPE FORM1. X
FORM1. FORMAT (A <0 >)
```

would cause the character string X to be printed in its entirety. If X was a numeric variable and the format command were "I<0>," X would be converted to ASCII and only the required number of characters would be output.

Figure 3 is a table of the format commands, their required arguments, and the variable types associated with each command. Commands exist for the output of decimal, octal, binary, and double precision integers, for fixed and variable length character strings, and for vectors of any data type.

Special commands for scope control. Special commands exist for specifying the size of output characters, for positioning the display, and for loading and clearing the display memories.

Under normal operation, PL/E supports two sizes of ASCII characters. The large size divides the screen into 128 by 128 grid points, allowing approximately 14 lines of display, with a maximum of 14 characters per line. The smaller size uses a 256 by 256 organization which halves the size of display characters and quadruples the capacity. Character size is set through the use of the "Z" command. Character sizes can be intermixed in the same format statement.

Position of display is set as a function of characters previously output. Location is relative to the upper left-hand corner of the screen. A display can be positioned as desired through the use of the L and X commands or by output of special graphics control characters. Actual displacement from the origin is generally a function of display size.

The scope memories are normally loaded at the completion of format statements. Commands exist, however, for selectively loading and starting or stopping display operations. The Unload exit (U) command terminates scope format processing without loading the

COMMAND	DESCRIPTION	ARGUMENT	ASSOCIATED VARIABLE
A	ALPHA (ASCII)	FIELD LENGTH	STRING VARIABLE
B	BINARY	FIELD LENGTH	NUMERIC VARIABLE
D	DOUBLE PRECISION	FIELD LENGTH	NUMERIC VARIABLE
E	ERASE MESSAGE	MESSAGE NUMBER	NONE
H	HOLLERITH	NUMBER OF CHARACTERS	NONE
I	INTEGER (BASE 10)	FIELD LENGTH	NUMERIC VARIABLE
J	JAM LOAD	NONE	NONE
K	CLEAR SCREEN	NONE	NONE
L	LINES (CRLF)	NUMBER DESIRED	NONE
M	MESSAGE ID	ID NUMBER	NONE
N	OUTPUT SUBJECT NUMBER	NONE	NONE
O	OCTAL	FIELD LENGTH	NUMERIC VARIABLE
P	PURGE ALL MESSAGES	NONE	NONE
S	START DISPLAY	NONE	NONE
U	UNLOADED EXIT	NONE	NONE
V	VECTOR NEXT	DIMENSION	NONE
W	WRITE LITERAL	DATA FIELD	NONE
X	FILL SPACES	NUMBER OF SPACES	NONE
Z	SET CHARACTER SIZE	SIZE	NONE

Figure 3. PL/E formatted scope control commands.

memory or without display operations. The Jam load (J) command loads the buffer memory without starting display. The Start display (S) command assumes the scope memory has been loaded and immediately starts the refresh cycle. The S and U commands are used primarily in situations where timing is extremely critical. By using S and U selectively, the potential effects of delay caused in hardware character generation (worst case, 10 msec for 2,000 points) are minimized. In terms of controlling onset times, it is possible to control display to within 1 msec using this technique.

Displays are turned off via the clear screen (K) command, which inhibits refresh operations and thus blanks the screen. Scope messages are erased by the purge (P) command, which deletes all previously encountered messages for the appropriate scope. Messages can be selectively erased by using the M and E commands. M associates a message identification number with a particular format statement and E allows that particular number to be erased.

Graphics

The formatted I/O routines are designed primarily to deal with ASCII input and output. There are, however, many situations in which an experimenter may want the capability to present graphics on the station displays. In situations where graphics are limited in scope (e.g., a small number of stick figures), special graphics characters can be defined for the hardware character generator and these figures processed as normal string variables. In other situations, however, more involved techniques are required.

PL/E contains a GRAPH command that converts end points corresponding to graphic display coordinates into special graphic characters that allow complex figures to be drawn on the displays. These special graphic commands correspond to angular rotations relative to the present position of the display. The display always starts at the origin (upper left-hand corner) and consists of a sequence of bright and dark (invisible) lines. The general form of the GRAPH command is:

GRAPH INPUT-VECTOR,SIZE,DISPLAY-SIZE,
OUTPUT-VARIABLE,BUFFER,COUNT.

The input vector is an array that contains the end points of the lines to be plotted. Size is the number of end points to be displayed. The display size refers to the grid spacing previously mentioned. The output variable and the buffer are the string variable and its corresponding buffer that will contain the special characters output by the GRAPH routine. Count is the number of graphics characters required to draw the figure described by the input vector. To illustrate the use of the GRAPH command, examine the following command sequence:

GRAPH PTS(0),10,0,PIX,PIXBUF(0),COUNT.

This sequence will take 10 points from the vector PTS which are considered to be lying in a 128 by 128 grid and insert their graphic character description in PIX.

Graphic data description. An input vector describes a sequence of line segments over the positive quadrant of a Cartesian coordinate system. The origin (0,0) is in the upper left corner, with positive x coordinates increasing to the right and positive y coordinates increasing downward. The input data consists of alternating x and y coordinates. For example, to outline the display screen one could use the following:

GRAPH BOX(0),4,0,OUTLINE,BUFF(0),COUNT

Where BOX(0)=127

BOX(1)=0
BOX(2)=127
BOX(3)=127
BOX(4)=0
BOX(5)=127
BOX(6)=0
BOX(7)=0

The cursor can be moved without physically drawing a line by using negative arguments.

Presentation of complex figures. Graphic data may be output using either the formatted scope routine, WRITSCOPE, or by the special-purpose DISPLAY command. WRITSCOPE is limited to processing 128 characters in a single format statement for purposes of memory efficiency. Graphic displays, however, may require as many as 1,500 characters, depending on complexity. The DISPLAY command simply adds the graphic characters to the list of messages that have previously been output to the appropriate scope. The WRITSCOPE J and S commands must be issued to physically start the display operation.

Control of Consoles

There are PL/E commands for monitoring the status of input consoles and for selectively switching the digital outputs. The WAITR command suspends a subject until either a time limit expires or one of a set of keys is pressed (released). The WAITK command suspends a subject until the desired response. RECDK monitors the status of the keyboard without suspending the subject. All three commands record the actual response as well as the time elapsed since the command was issued. Delays caused by simultaneous requests at multiple stations amount to less than 50 microsec, which is well within the 1-msec accuracy of the system.

CONCLUSION

This paper describes a technique for efficiently interfacing a set of general-purpose CRT terminals to a central laboratory computer. The system can be used to present alphanumeric text, dot patterns, and complex geometric figures. A series of higher level software routines simplify the method of presentation and accurately control display onset and offset times.

REFERENCES

- BURKHARDT, K. J. EMPP: An extensible multi-programming system for Experimental Psychology. *Behavior Research Methods & Instrumentation*, 1976, 8, 239-244.
- GILLMAN, C. B., & BUCKLEY, P. Software character generation. *Behavior Research Methods & Instrumentation*, 1973, 5, 228-230.
- SPERLING, G. Flicker in computer-generated visual displays: Selecting a CRO phosphor and other problems. *Behavior Research Methods & Instrumentation*, 1971, 3, 151-153.
- VAN GELDER, P. CRT displays in the experimental psychology laboratory. *Behavior Research Methods & Instrumentation*, 1972, 4, 102-103.

(Received for publication April 19, 1976;
revision accepted June 4, 1976.)