

TSS DISPLAY DRIVER

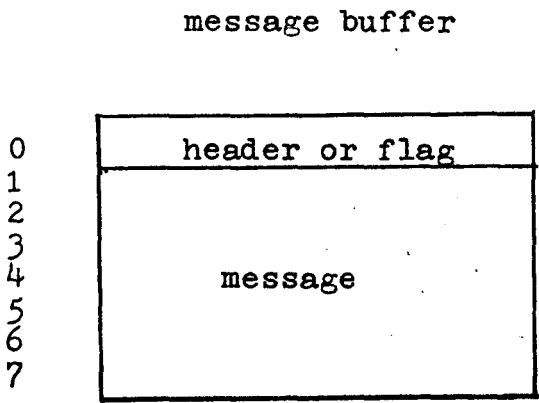
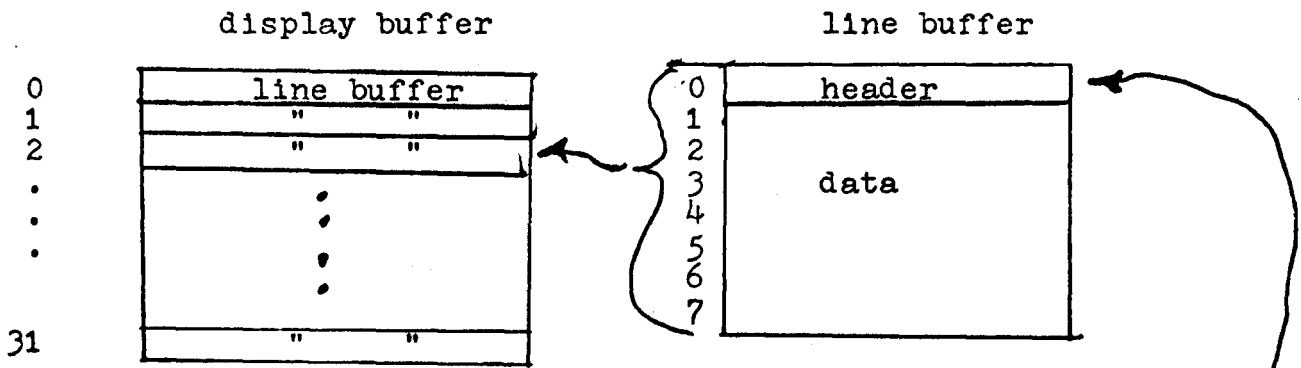
The purpose of the display driver is two fold;

- 1) To allow users and the system to communicate with the operator through the display console.
- 2) To provide access to system information, core, and ECS from the console as a debugging aid.

Displays are made up of logical displays which may be selected by the operator for viewing on either screen. Communication is accomplished through a logical display known as a user display. A user display is a display buffer (an ECS file) which may be written into by the user and will be copied onto the screen by the PPU when selected by the operator. In addition, there are core/ECS displays and PPU composed displays (clocks, etc.).

User Screens

Each user screen consists of a display buffer an Event channel and a message buffer. The display buffer consists of 32 eight word line buffers. Each line buffer consists of a one word header and seven data words.



Brite is the relative brightness of the line buffer on the screen. It ranges from 1 to n, where n will probably be about 5. The brightness is proportional to n_j and $n = 0$ is equivalent to $n = 1$.

Fcn function is half of the function code to be sent to the device. Only the lower 4 bits are significant. $Fcn = XY$; where X is the mode and Y is the character size as follows:

X = 0	dot mode
1	character mode
Y = 0	64 characters/line (small)
1	32 characters/line (medium)
2	16 characters/line (large)

Ct is the length of the data to be output in bytes.
if $Ct = 0$, nothing is output.

Data will be output verbatim, beginning with the high order byte of the word following the header word and continuing until Ct is exhausted. If $Ct = 0$, no data in the corresponding line buffer will be output. Data is assumed to be positioning information and display code (or dot co-ordinates). See peripheral manuals for additional information.

Keyboard Messages

Operator Type-ins.

Entries made by the operator will be displayed in the lower left corner of the left screen.

Messages beginning with a slash ("/") will be interpreted by the PPU as a comand. Other messages will be sent by the message buffer of the display currently on the right screen. There will be a PPU command to enter a message into an arbitrary message buffer. Messages in the buffer will end with a console (cr) character (60B).

The Message Buffer

The message buffer is eight words long. It has one header word followed by seven data words. The header is a busy flag.. When the operator pushed (cr), the flag will be checked. If the buffer is not busy (ie; flag = 0), the flag will be set, the message copied into the buffer, and an event sent on the corresponding event channel.. If, however, when an attempt is made to send the message, the buffer is busy (flag = 0), the message "BUFFER X BUSY" (where X is screen name) will appear on the screen. The operator may then either try again (cr), blank the message (blank). or force the message to be stored in the buffer (force)..

Since there exists one message buffer and one event channel for each user screen, there should be no problems. However, if several users are sharing a screen (by agreeing to write only in certain line buffers and only at certain screen co-ordinates) they must sort the messages themselves.

NOTE: The messages in display code, ten characters per word, left justified, and beginning with the word following the header. Each message ends with a console (cr) (60B). No guarantees are made about zero fill. It is also hoped that the user will zero the busy flag after receiving a message.

New proposal for messages

There now exist logical keyboards which take the place of message buffers and their event channels. These logical keyboards are independent of logical screens. A logical keyboard may be selected by the operator for receipt of implied messages (those not beginning with "/"). Messages may be sent to an arbitrary keyboard on command. ~~A message will consist of a string~~ A logical keyboard will consist of an event channel. Currently, messages will be limited to 10 chans. until message channels are implemented. All other statements about ^{the} message buffers format will hold for message channels (except there is no header word).