GEMINI GALAXY 1 with EXTENDED KEYBOARD

This Galaxy has been supplied with the new GM827 extended keyboard. In order to support the extra facilities provided by the keyboard the system has been fitted with a new EPROM, 'IVC-MON V2.0'. The following outlines the additional facilities that are provided over the original 'IVC-MON V1.0' described in the enclosed 'G812-IVC SOFTWARE' manual.

1 Nested ESCape sequencies have been modified so that, for example, "ESC ESC % %" no longer causes the IVC to 'lock-up'.

2 'Clear to End of Screen' (ESC %) has been modified to include the last character on the screen.

3 'Delete Line and Scroll Up' (^K) has been modified to clear the bottom line if the cursor is positioned there.

4 The Cursor Addressing (ESC = R C), Set (ESC S X Y), Reset (ESC R X Y), and Test (ESC T X Y) sequencies have now been modified to accept nested ESCape sequencies during the command string.

This is made possible as the screen coordinates in the above sequencies are always offset by 20H and so a new ESCape sequence can be detected. This means that certain CP/M packages that do keyboard polling during cursor addressing will no longer cause a 'lock-up'.

5 The scroll routine now buffers characters from the host/IVC interface as per the IVC-MON manual.

6 The screen scroll buffer has been reduced from 128 bytes to 64 bytes.

7 The stack area has been moved and can now accomodate 64 bytes.

8 The IVC-MON V2.0 monitor has been extended to handle the extended keyboard with the programmable function keys. As a result the monitor now resides in a 4K EPROM (2732). The extra 512 bytes of workspace mentioned in the G812 manual (Appendix 1) have now been used. LK2 is now used to define whether the GM821 or GM827 keyboard is in use. (In = GM821, Out = GM827.)

9 An additional ESC sequence (ESC f) has been added to define the function keys.

David

The enclosed EPROM needs to be fitted in the IVC to support the extra function keys, numeric pad etc.

Two link changes need making

(1) Move link 1 from B-C (2716) to B-A (2732)

(2) Remove link 2 (selects 827 extended kbd in place of 821 kbd)

Paul

ROTEC KEYBOARD

The Rotec keyboard includes additional keys that may be programmed (via the IVC software) to return one or more characters to the host computer. In order that the IVC can distinguish these special keys the Rotec keyboard returns unique double-byte codes from these keys. The IVC software replaces each double-byte code by a single character or string of characters from an internal table. This table is held in the workspace ram of the IVC, and may be modified at any time, either by program, (using the ESC f sequence), or directly from the keyboard. On Reset an initial table is copied out of the IVCMON EPROM into the ram. The necessary information is given for those able to program 2732 type EPROMs to change the default strings.

The programmable keys are the row of function keys along the top of the keyboard (labelled F0-F9 & EDIT), the four cursor control keys, and all the keys in the numeric pad at the right-hand end of the keyboard.

The shift key may also be used in conjunction with these keys to produce another set of unique codes.

Each of these keys, with the exception of shift/EDIT can be redefined to produce any character or string of characters required by the user. For example F0 could be set up to hold the string "pip a:=b:*.*[v]<CR>". The key definitions may be set up in two ways:- a) By the User at the keyboard, and b) By program using an escape sequence.

Defining keys from the keyboard.
Typing shift/EDIT on the keyboard will draw the response

*** List/Edit a Function key ***

If a function key is now pressed, the current definition of that key is listed on the screen. All control codes in the string are displayed in the expanded form of ^<character> (eg a carriage return would appear as ^M). This is followed by the message:

*** List/Edit complete ***

The IVC monitor has put this information directly onto the screen, NOTHING HAS BEEN SENT TO THE HOST COMPUTER and it is totally unaware of what has heppened.

If instead of hitting a function key shift/EDIT is pressed again the following string will appear:

*** Press the function key to be defined, then type in a string ***
*** followed by any function key ***

At this point you can select the function key you wish to redefine. Type it followed by the string you wish to enter. As you type in the string it will be echoed to the screen, once again with control characters being expanded to the form ^<character>. NOTE it is assumed that any character typed is to be part of the string, thus if you hit "backspace" ^H will appear on the screen and be control/H will be entered into the

string. If you make a mistake you will have to start again.

The entry of a new definition is ended when any function key is pressed. (No recursive definitions are allowed!). At this point the following messages will appear:

```
*** New defintion entered ***
***  List/Edit complete   ***
```

If no string was entered the function key will no longer return any characters and if the key is "listed" the following message will appear:

```
*** Function key undefined ***
***  List/Edit complete   ***
```

As above, the Host computer is totally unaware of what is happening.

Defining keys by software.
A key may be redefined by software using the following escape sequence:

ESC f <code> <string> <byte with msb set> .....

where <code> is the unique code identifying the string. This is equivalent to the code produced by the key with 80H added. (ie The IVC code for F0 is 81H, for shift/F0 is 91H, and so on). <string> is the string of characters to be returned every time the key is pressed. The new definition is terminated when a byte with the msb set is encountered. If this byte is a legal keycode (81H-0BDH excluding 90H and 9BH) then a new definition is started, if it is illegal (ie > 0BDH) then the escape sequence is terminated.

Two additional features are included in the escape sequence:
<ESC> <f> <d> or <ESC> <f> <D> will reset the key defintions to their default (or power-up) state.
<ESC> <f> <?> will cause the IVC to send to the Host the table of the current function key definitions. The table is terminated with the byte 0FFH.

NB It is perfectly possible to enter a null string for a key definition and effectively disable it. (It will be ignored until redefined).

If you get too carried away with your defintions you will see the message:

*** IVC internal error - table overflow ***

It shouldn't really happen as the table can use up to 512 bytes which gives an average of about 8 characters per key (assuming you redefine the numeric pad as well).

The simple routine SAVEKEYS is supplied. This is for setting up COM files holding particular sets of key definitions. Thus it is possible to easily set up files such as KPEN.COM and KWS.COM that could be executed before running programs such as PEN or WORDSTAR to customise the key settings appropriately.

IVC FUNCTION KEY CODES

Shown here are the hexadecimal codes associated with the various programmable keys. The codes are in the range 81H-0BDH. Note that the following codes do not occur and treated as illegal by the "ESC f" sequence: 90H and 9BH

The function keys:

shifted	91	92	93	94	95	96	97	98	99	9A	XX
normal	81	82	83	84	85	86	87	88	89	8A	8B
KEY	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	EDIT

The Cursor control keys:

shifted	9C	9D	9E	9F
normal	8C	8D	8E	8F
KEY	<-	->	/\	\/

The Numeric pad:

shifted	B0	B1	B2	B3
normal	A0	A1	A2	A3
KEY	7	8	9	+
shifted	B4	B5	B6	B7
normal	A4	A5	A6	A7
KEY	4	5	6	-
shifted	B8	B9	BA	AF
normal	A8	A9	AA	AE
KEY	1	2	3	ENTER
shifted	BB	BC	BD	
normal	AB	AC	AD	
KEY	,	0	.	

CHANGING THE DEFAULT DEFINITIONS

If required the table of default key definitions in the EPROM can be changed. In order to do this you must be able to re-program a 2732 type EPROM. The existing EPROM should be copied to the memory of the programmer, and then the end of the program in the EPROM should be located. Currently this is around address 0C00H. Searching backwards from this point the copyright message "(c) dci software 1982" should be located. The default table starts immediately following this message.

The first four bytes of the table are:-

80 1B 90 1B ..... (In hexadecimal)

ON NO ACCOUNT MUST THESE BE CHANGED otherwise you will find that you have redefined the ESCAPE key (normal and shifted).

The new strings can be entered in a similar manner to those already there. The format is identical to that of the "ESC f..." sequence.