SECTION IV

THE INTELLIGENT KEYBOARD

IKB-1

4 - 1 KEYBOARD SYSTEM OVERVIEW

The IMSAI Intelligent Keyboard (IKB-1) is a unique, programmable keyboard integrated into the VDP-40. A number of keyboard functions may be programmed by the user through simple keyboard commands, or under direct control of the system CPU. The IKB-1 contains a full complement of on-board devices including an Intel 8035 microprocessor.

The IMSAI Intelligent Keyboard utilizes 63 alphanumeric, contact keys. All keys are fully debounced, and the keyboard provides full N-KEY ROLLOVER. Continuous depression of a key will invoke an auto repeat function, causing a character to be output continuously until the key is released. Standard alphanumeric and control keys are arranged in a typewriter type layout.

The keyboard may be configured to operate as a parallel OR serial input device. The parallel mode supports a full 8 bit parallel format with RDY and ACK handshaking. The serial mode supports both EIA and TTL levels. Serial baud rates are selectable by keyboard commands and allow for the selection of seven separate rates from 110 to 2400 baud.

NOTE: Serial mode requires several jumper changes on the IKB-1 and is not normally used on the VDP-40.

CPU programming allows the keyboard to operate in encoded or unencoded modes. Selection of the encoded mode allows the keyboard to output standard ASCII code, while the unencoded mode gives the user a raw map of the keyboard array allowing selected key depressions to be used for games, music and system development. Any individual key may be reprogrammed to output any given ASCII character. The IKB-1 keyboard may be programmed to operate in upper case only or in both upper and lower case use. The IKB-1 also contains a small, high quality sound transducer.

4 - 2 KEYBOARD HARDWARE

The IMSAI keyboard uses an on-board 8035 processor to handle all keyboard control functions. The keyboard control firmware is resident in the 4751 ROM at U7.

The control sequence is completely dependent on the firmware program, described in Section 4 - 5. The 8212 at U6 serves as an address latch during instruction fetch cycles from the 4751 ROM.

The two 74154, 4-line to 16-line decoders, at locations U1 and U2 allow the processor to scan the keyboard array and optional external keypad. Keyboard data is output to the parallel interface through the 8212 latch at U4.

4 - 2.1 Instruction Fetch

The 8035 processor executes those instructions stored in the 4751 PROM at U7. During an instruction fetch cycle, the 8035 outputs the 12 bit address of the next instruction to be executed on its Bus Lines (D0 - D7), and Port 2 (bits 0 - 3). The high order address bits A8 - A11 are internally latched and remain stable on the output lines of Port 2 (bits 0 - 3). The low order address bits are latched into the 8212 at U6 using the ALE strobe from the 8035 processor.

Once the address is present and stable at the address inputs of the 4751 ROM, the processor will drive its /PSEN strobe active low to enable the outputs of the 4751 ROM. Data is then input to the processor through its Bus lines D0 - D7 on the rising edge of /PSEN.

4 - 2.2 Keyboard Scan

The keyboard array consists of a diode matrix, 16 columns wide and 4 key positions (rows) to a column. To identify a key closure, the 8035 processor will output a column number to the 74154 at U1 through port P1, bits 0 - 3. The column number is decoded by the 74154 and will drive one of its 16 output lines active to place a low logic level along the selected column line of the keyboard array.

The 8035 processor then reads the state of the 4 row lines through port 2 bits 4 - 7. Any key closure in the selected column will appear as a "0" in one of the four bit positions (rows).

By successively scanning each column in this manner, the 8035 processor can key closures in the entire keyboard array.

If an external keypad is used, the same scanning technique is used with the 74154 at U2.

4 – 2.3 Speaker

An audible speaker beep is created by the 8035 processor by successively outputting a series of 1's and 0's to the speaker through output port one, bit 5. The pitch or frequency of the tone is determined by the rate of repetition.

4 – 2.4 Character Output

Characters to be output from the keyboard are latched into the 8212 at U4 from the 8035's Bus Lines on the rising edge of the processor's /WR strobe. The outputs of the 8212 directly interface to the external input port when the keyboard is operating in parallel mode. Parallel handshaking is achieved using the 8035 port 1, bits 6 and 7, as /ACK and /RDY. The 8035 outputs a "1" to bit 7 of port 1 when a character is ready to be input to the computer. Bit 6 of Port 1 is then polled to determine when the computer has read the character.

When the keyboard is used in the serial mode, the serial data appears on bit D0 of the 8035's Bus and is latched into the 8212 at U4. The DO0 output of the 8212 is then used to provide the TTL serial data line out to the interface. The DO0 line of the 8212 2ll also drives the base of Q1 so that EIA level serial data appears at the collector of Q1 (J1 pin 2).

4 – 2.5 Program Lines

When an output port is used to program the keyboard, the data appears at the T0 input of the 8035. T0 is a general purpose test input and in this case allows the 8035 to input the sequence of tests which comprises the command strings.

Handshaking is achieved by using the T1 input of the 8035 as a ready line from the interface. An acknowledge signal is generated with an output line from the 74154 at U2. The processor will place a IFH on Port 1 bits 0 – 4, then, on the next cycle, will place a 00 H on these same bits. This causes U2 – 17 to toggle resulting in the desired acknowledge signal (J1 – 11).

4 – 3 KEYBOARD OPERATION

The keyboard will operate in two basic modes of operation: program mode and data entry mode. The program mode allows the keyboard options to be programmed either directly from keyboard commands or from an output port. The data entry mode is the standard data input mode of operation and allows data to be entered to the computer. Keyboard operation in the data entry mode is determined by the option selected in the program mode.

4 - 3.1 Program Mode (from Keyboard)

The keyboard options may be programmed directly from keyboard commands. Programming is initiated by depressing the "PROG" key located in the upper right corner. The keyboard will verify entry into the program mode with an audible beep and the PROG LED lit.

Once the keyboard is in the PROGRAM mode, a valid command typed on the keyboard will select a particular keyboard option. A number of commands may be typed in sequence to select more than one keyboard option where applicable. The last character of each valid command will be confirmed with an audible beep.

When all features have been selected, the program made is terminated by pressing the "END MODE" key, located to the left of the "PROG" key.

Valid commands are as follows:

RQ	places the keyboard in parallel mode
R1	selects 110 baud serial output
R2	selects 150 baud serial output
R3	selects 300 baud serial output
R4	selects 600 baud serial output
R5	Selects 1200 baud serial output
R6	Selects 1800 baud serial output
R7	Selects 2400 baud serial output
D	selects DEFAULT MODE ... Parallel outputs upper case only, encoded outputs
U	selects UPPER CASE ONLY mode. Operation is similar to TTY keyboard. All alpha character output in upper case. Shift key controls upper case for all non-alpha characters.
L	enables LOWER CASE characters. Operation is similar to typewriter keyboard. Shift key control upper case for all characters. SHIFTLOCK key will lock all alpha characters to upper case. SHIFTLOCK may be released by depressing the SHIFT key. SAB substitutes any ASCII character, B, for any other ASCII character, A. The keyboard will output the character, B, whenever the key, A, is typed. The resulting character, B, is unaffected by "CTRL", "SHIFT" or "FLAG". To cancel the current substitution, type "S" followed by two blank spaces.
E	sets the keyboard in normal ENCODED mode. In this mode, the keyboard is scanned using a N-key Rollover algorithm and characters are output in Standard ASCII format. The se of the flag key is encoded as the high order bit of any character sent.

V places the keyboard in verbatim or unencoded mode. When the END-MODE key is depressed, the keyboard will start outputting a continuous stream of bytes indicating the complete state of the keyboard. This may be used to transmit multiple key depression information for music or games. For details of the unencoded format reference.

4 – 3.2 Data Entry Mode

Once the keyboard options have been selected and the END MODE key has been depressed, the keyboard will operate in the Data ENTRY MODE.

ENCODED: If the "E", encoded, option is selected, any key or sequence of keys typed on the keyboard or external keypad (optional) will cause the corresponding ASCII code to be sent to the computer.

In the encoded mode, continuous depression of a key will invoke the AUTO REPEAT function, causing the character to be output continuously until the key is released.

Selection of the "U", upper case only option, will cause the SHIFTLOCK LED to light.

UNENCODED: If th "V" unencoded option is selected, the keyboard will output a continuous series of bytes which will provide the user with a continuously updated map, indicating the state of the keyboard array. (The state of the external keypad is excluded from the map.)

Each byte which is output from the keyboard is in the following form:

D7 D6 D5 D4 D3 D2 D1 D0

Row State Row Number

The low order four bits contain a binary encoded row number (0 – 15 decimal), and the high order four bits indicate the state of that row.

A "0" in any of the bit positions D7 – D4 indicate that the corresponding key(s) of that row are depressed. Likewise, a "1" indicates that the key(s) is in the up position.

The following table lists the keyboard array by row number and bit position (D7 – D4).

The keyboard will successively output the state of each row in the above format as long as the "V" unencoded mode is selected.

Unencoded Mode

Row #	Row State D7	D6	D5	D4
0	RSHFT	CTL	TAB	ESC
1	FLAG1	SHFTLK	BRK	LSHFT
2	2	A	Q	!
3	R	S	W	"
4	C	D	E	#
5	V	F	R	$
6	B	G	t	%
7	N	HI	Y	&
8	M	J	U	'
9	<	K	I	(
10	>	L	O	)
11	?	+	P	0
12	=	†	§	.
13	SPACE	BS	?	LF
14		DEL	CR	
15		MODE		ENDENTRY

4 - 3.3 IKB-1 INTERFACE

For use in the VDP-40, the IKB-1 keyboard is configured for parallel output. Serial output is available, but several jumpers will need to be installed. The following table lists the signal lines used in the parallel mode. All signals are available at the 24 pin J1 connector are the rear of the keyboard PC board.

Parallel Mode Signals (J1 Connector)

Pin #	Signal Name	Description
22	DO7	Data bit 7
21	DO6	Data bit 6
20	DO5	Data bit 5
19	DO4	Data bit 4
18	DO3	Data bit 3
17	D02	Data bit 2
16	DO1	Data bit 1
15	DO0	Data bit 0
24	/RDY	Ready (active low)
23	/ACK	Acknowledge (active low)

4 - 4 KEYBOARD SOFTWARE

4 - 4.1 INITIALIZATION

On startup of the 8035, all interrupts are disabled and the chip is started at location 0. The program then clears most of RAM, initializes the timer, turns on the timer interrupt, and enters a wait loop which performs various functions, such as turning on the LED's, while waiting for a signal from the interrupt routine to initiate keyboard scans. I/O routines handling communication with the computer are called from the main loop.

4 - 4.2 TIMER

The timer interrupt is based on a resettable 134.1 microsecond timer clock which is derived from the 3.58 MHz system crystal. Three processes are timed by the interrupt routine: (1) keyboard scans, which are initiated every 10 to 14 milliseconds after the interrupt routine sets the byte SCANFLAG; (2) serial output, which is handled directly by the interrupt routine; and (3) the loudspeaker beeps, which are also handled directly by the interrupt routine. The principal function of the interrupt routine is to time serial output accurately; for this reason, each interrupt begins with a delay loop (FUGLUP) which precisely times the resetting of the timer clock. The length of the delay is looked up in a table CKFUDGE which is indexed by the baud rate index RATE. Likewise, the timer is set to a value looked up in the table CKDIV, again indexed by RATE. This sets up the basic interrupt rate. Serial output is initiated by the HANDSHAKE routine by putting a data byte in RSERCHAR and setting RBITNUMBER to 11. The interrupt routine, upon seeing a non-zero value in RBITNUMBER, starts counting down from the main interrupt rate to the baud rate (by a value in CKBIT). Each time the counter goes to zero, the rightmost bit of RSERCHAR, shifted right, or a stop or start bit, is placed on the BUS outputs of the 8035, RBITNUMBER is decremented, and the clock divider is re-initialized.

4 - 4.3 SPEAKER

The loudspeaker beep is initialized from several places in the program by setting RBEEP to some nonzero value. Each time the initial timer routine sees a nonzero value of RBEEP, it complements bit 5 of P1 and decrements RBEEP. Since the basic timing rate varies for higher baud rates, the beep is higher in pitch when the baud rate is 1800 or 2400 baud, and slightly lower when for parallel output.

4 - 4.4 MAIN LOOP

The main loop of the program alternately checks various conditions which turn the indicator lights on, and calls a series of routines which handle keyboard encoding and data communications. It should be noted that whenever a LED is flashed, the program enters a wait loop so that the light is long enough that it can be seen. This has an effect on the timing of non-interrupt driven processes, such as unencoded scans and RDY-ACK

handshaking with the computer. The other processes in the main loop are HANDSHAKE, which initiates transmission of a byte to the computer, CHECKT1, which handles control (programming) information coming from the computer, CHECKPROG, which looks at the PROG/END-ENTRY keys to set or clear PROGFLAG, and CHECKSCAN, which will initiate a scan of the keyboard when the interrupt routine has set SCANFLAG.

4 - 4.5 HANDSHAKE

HANDSHAKE functions entirely differently in serial and parallel mode. In serial mode, it looks for the FULL flag, which is set by SCAN or DOUNENC to indicated that a byte to be transmitted is in BUFF. If FULL is set, it clears FULL, transfers the data byte from FUFF to RSERCHAR, and sets RBITNUMBER to 11 to signal the interrupt routine to start transmitting the character. In parallel mode, on the other hand, the routine places the data byte on the BUS, sets P1 bit 7, which is the RDY line going to the computer, and waits for P1 bit 6, the ACK line, to become true. When ACK becomes true, RDY is turned off. In the current routine, the situation is actually a little more complicated; after a timeout delay, RDY is turned off whether ACK appears or not. Also, an extra check for ACK going away immediately after RDY is turned off speeds up handshaking if the computer responds quickly or if RDY is jumpered to ACK.

4 - 4.6 CHECKT1

CHECKT1 checks for programming signals coming from the computer. The T1 input to the 8035 chip is used as a RDY input, T0 is used as a data bit, and an acknowledge is generated using one of the multiplexer outputs. The program works by counting 1's on the T0 line when T1 goes true. It exits without doing anything if too many (more than 3) 1's are seen in a row or if more than about 3 milliseconds elapse between 1's. A normal control message ends with 0. The meanings of the various legal sequences are defined in the programming page of the manual.

4 - 4.7 CHECKPROG

CHECKPROG sets PROGFLAG to 1 when the PROG key is depressed, if PROGFLAG is currently 0. The effect of this is to cause a branch from the scan routine to PROGRAMMIT, which uses input characters to setup other values for PROGFLAG. On subsequent entries to PROGRAMMIT, a 4-way branch on the value of PROGFLAG will be used to interpret subsequent characters in a command string. Note that at the point of entry to PROGRAMMIT, a character is freshly encoded, that is, shift, control and flag do not effect its value. If PROGFLAG is 4, the current mode is "enter character which will be substituted for some key", which has to be handled by a special routine later in SCAN. When a particular command sequence is completed, PROGFLAG is reset to 1 and the speaker is beeped. Note that some of the programming routines are also used by the external-programming routine CHECKT1, which calls REMOTESEL. In this

case, if PROGFLAG is already 0, it is not set to 1.

4 - 4.8 CHECKSCAN

CHECKSCAN works by evoking the N-key rollover encoding routine SCAN whenever SCANFLAG is set and VERBFLAG is false. SCAN works as follows: A map of keys which are currently believed to be depressed is maintained in the array OLDKEYS. Scanning consists of reading in succession all 22 4-bit scan rows of the keyboard matrix (including the external keypad), packing the nibbles into 8-bit bytes, and comparing the input bytes with the corresponding entry in OLDKEYS. If a transition is seen, scanning stops, the row of the matrix, the bit number of the rightmost different bit, and whether the change is up or down are recorded. On the next scan, the resulting flags ONBIT and OFFBIT are checked. If the corresponding key is STILL down, or still up, respectively, the transition is considered debounced. In the case of a downward transition, the bit for the corresponding key is cleared (down keys are 0's) and encoding begins. In the case of an upward transition, the bit for the corresponding key is set in OLDKEYS. In this way, only transitions in the keyboard matrix initiate encoding and transmission of characters — any number of keys can be down, and a new depression will transmit a new character.

4 - 4.9 ENCODING

Encoding is straightforward. An index to an ASCII table ASCTABLE on page 3 of the program is computed, and the basic ASCII code corresponding to the key is looked up. At this point PROGFLAG is checked to see if the character is to be input to the programming routine rather than to be transmitted. After that, if it is to be transmitted, the shift key and the shiftlock bit of MODE (set or cleared earlier) are checked to determine if the character is to be shifted. Likewise, the control key is checked if appropriate to determine if the character is to be turned into a control character. At this point, if PROGFLAG is equal to 4, the character is saved as a substitute character, and not transmitted. Otherwise, the character is placed in BUFF, FULL is set to tell HANDSHAKE to send the character, and REPCOUNT is set to -50 to time the delay to the first repeat of the character. Subsequent scans check to see if the last character to be encoded has its corresponding key still down. If so, REPCOUNT is incremented; if it goes to 0, the character is re-transmitted and REPCOUNT set to -5. This results in a delay of about .8 second before the first repeat and about .08 seconds between subsequent repeats.

4 – 4.10 UNENCODED OUTPUT

Unencoded output is handled as follows: When the CHECKSCAN routine sees that VERBFLAG is set, it calls the routine DOUNENC. This routine reads a row (4 bits in the left half of the input byte) of the keyboard matrix, packs the row number into the right half of the byte, and transmits the byte to the computer, incrementing the row number for the next call to DOUNENC. Note that (1) 1 means a key is UP, 0 means the key is DOWN, and (2) only the keys on the basic keyboard can be read in this mode — the external keypad is ignored, because only 16 rows can be encoded in this manner.

4 – 5 PARALLEL HANDSHAKING

When the keyboard is configured for parallel operation, the /RDY and /ACK lines are dedicated for handshaking.

The keyboard will assert /RDY (active low) when data is stable and is ready to be latched by the interface port. If it is desired to use /RDY, install a jumper in location JF.

If it is desired to use an external acknowledge signal /ACK from the parallel input port, also install a jumper at location JD. In this configuration, once /RDY is asserted, the interface should respond by driving /ACK active low to indicate the reception of data.

If it is desired to jumper the keyboard's /RDY output to the /ACK input, install a jumper at location JE. This configuration will result in a short /RDY pulse and does not require an external acknowledge signal from the interface.

4 – 6 CONTROL (OUTPUT) DATA AND HANDSHAKING

If a parallel output port is used to program the keyboard, the data and handshake lines need to be configured for the type of output port used.

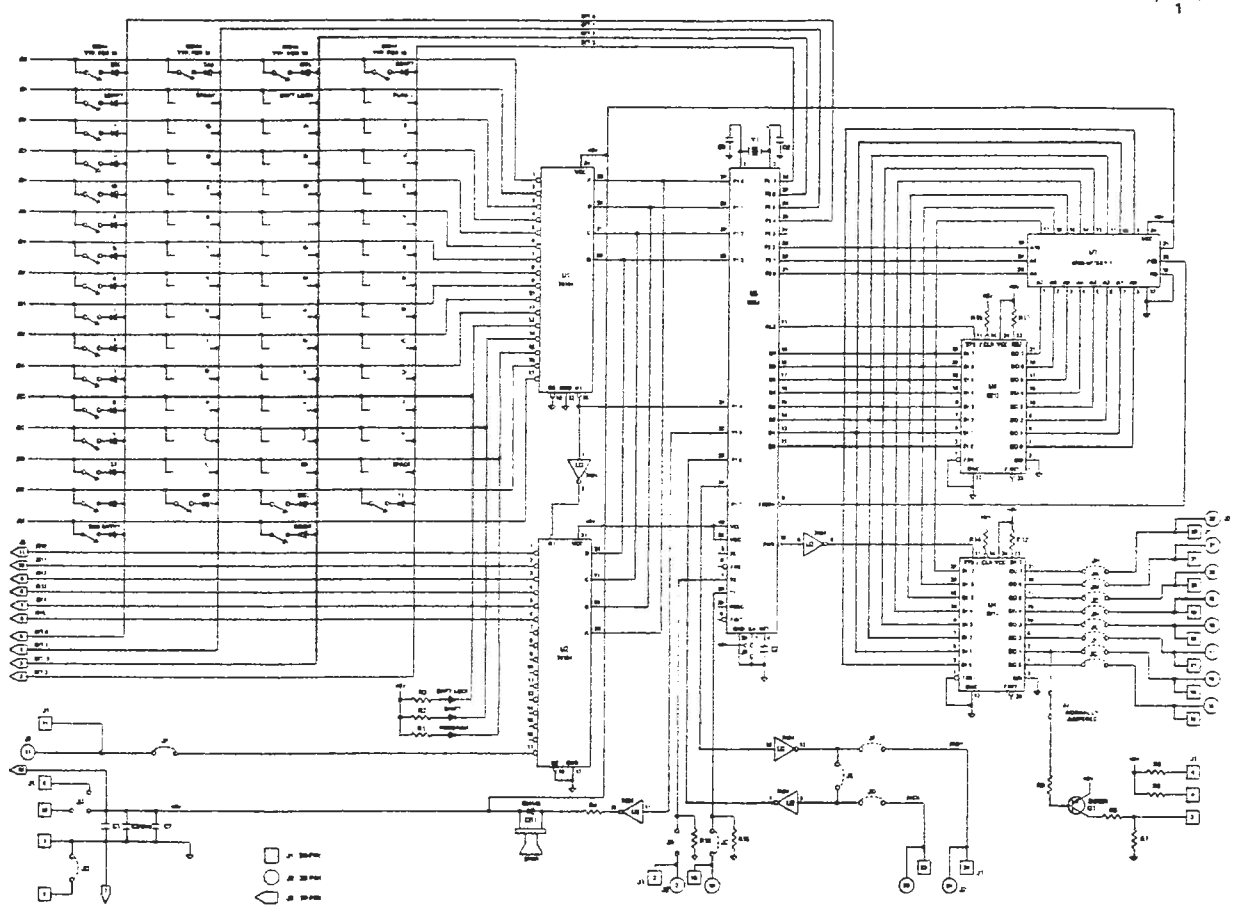
A jumper should be installed in location JB to enable the use of the data input line (J1, pin 2). This line will be driven by one bit of the output port used for keyboard programming.

Jumpers JC and JP allow the READY and ACKNOWLEDGE handshake lines to be used.

When it is desired to use a READY output from the interface, install a jumper in location JC. The interface should assert READY (J1, pin 10) when data is stable and is ready to be READ by the keyboard processor.

When it is desired to use an ACKNOWLEDGE output from the keyboard, install a jumper in location JP. The keyboard will assert ACKNOWLEDGE (J1, pin 11) following the receipt of the READY signal.

The READY and ACKNOWLEDGE handshaking is suitable for use with an output port having hardware handshake line., e.g. 8212 type ports. The use of software implemented handshake lines is not recommended due to timing constraints.



REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
0	ORIGINAL REV. 1	6/77	
1	REV. 1.3	8/77	

U1, U2	74154
U3	7404
U4, U6	8212
U5	8035
U7	4700 WITH "KEY 1 REV. 1"
C1	33 uF
C2	2.2 uF
C3 THRU C7	.1 uF
C8, C9	20 pF
CR1, 62 DIODES	1N4148
J1	25 PIN CONNECTOR
J2	26 PIN CONNECTOR (NOT PROVIDED)
J3	15 PIN CONNECTOR
Q1	2N3906
R1 THRU R3	82 OHM ¼W
R4	100 OHM ¼W
R5 THRU R7	1K ¼W
R9	4.7K ¼W
R8	150 OHM ¼W
R10	47 OHM ¼W
R11, R12, R15, R18	10K ¼W
R13, R14, R16, R17	1K ¼W

TOLERANCES UNLESS OTHERWISE SPECIFIED
FRACTIONS DEC. ANGLES
± ± ±

APPROVALS	DATE
DRAWN	
CHECKED	

IMSAI SYSTEM
IKB – 1 REV. 1 6/77
SCHEMATIC DIAGRAM

SCALE | SIZE B | DRAWING NO.

DO NOT SCALE DRAWING | SHEET