Intelligent Keyboard (ikbd) Protocol
Revised: 02/26/85

1. Introduction

The Atari Corp. Intelligent Keyboard (ikbd) is a general purpose keyboard controller that is flexible enough that it can be used in a variety of products without modification. The keyboard, with its microcontroller, provides a convenient connection point for a mouse and switch-type joysticks. The ikbd processor also maintains a time-of-day clock with one second resolution.

The ikbd has been designed to be general enough that it can be used with a variety of new computer products. Product variations in number of keyswitches, mouse resolution, etc. can be accommodated.

The ikbd communicates with the main processor over a high speed bi-directional serial interface. It can function in a variety of modes to facilitate different applications of the keyboard, joysticks, or mouse. Limited use of the controller is possible in applications in which only a unidirectional communications medium is available by carefully designing the default modes.

2. Related Documents

Atari Corp. RBP/GHU/SD Keyboard Schematic (9/14/84).

Atari Corp. RBP/GHU/SD Keyboard Layout (unnumbered, no date)

3. Keyboard

The keyboard always returns key make/break scan codes. The ikbd generates keyboard scan codes for each key press and release. The key scan make (key closure) codes start at 1, and are defined in Appendix A. For example, the ISO key position in the scan code table should exist even if no keyswitch exists in that position on a particular keyboard. The break code for each key is obtained by ORing 0x80 with the make code.

The special codes 0xF6 through 0xFF are reserved for use as follows:

0xF6	status report
0xF7	absolute mouse position record
0xF8-0xFB	relative mouse position records (lsbs determined by mouse button states)
0xFC	time-of-day
0xFD	joystick report (both sticks)
0xFE	joystick 0 event
0xFF	joystick 1 event

The two shift keys return different scan codes in this mode. The ENTER key and the RETurn key are also distinct.

4. Mouse

The mouse port should be capable of supporting a mouse with resolution of approximately 200 counts (phase changes or "clicks") per inch of travel. The mouse should be scanned at a rate that will permit accurate tracking at velocities up to 10 inches per second.

The ikbd can report mouse motion in three distinctly different ways. It can report relative motion, absolute motion in a coordinate system maintained within the ikbd, or by converting mouse motion into keyboard cursor control key equivalents.

The mouse buttons can be treated as part of the mouse or as additional keyboard keys.

4.1. Relative Position Reporting

In relative position mode, the ikbd will return relative mouse position records whenever a mouse event occurs. A mouse event consists of a mouse button being pressed or released, or motion in either axis exceeding a settable threshold of motion. Regardless of the threshold, all bits of resolution are returned to the host computer.

Note that the ikbd may return mouse relative position reports with significantly more than the threshold delta x or y. This may happen since no relative mouse motion events will be generated: (a) while the keyboard has been "paused" (the event will be stored until keyboard communications is resumed) (b) while any event is being transmitted.

The relative mouse position record is a three byte record of the form (regardless of keyboard mode):

```
%111110xx       ;mouse position record flag
       ||
       |------- ;right button state
       -------- ;left button state
X       ;delta x as a twos complement integer
Y       ;delta y as a twos complement integer
```

Note that the value of the button state bits should be valid even if the MOUSE BUTTON ACTION has set the buttons to act like part of the keyboard.

If the accumulated motion before the report packet is generated exceeds the +127..-128 range, the motion is broken into multiple packets.

Note that the sign of the delta y reported is a function of the Y origin selected.

4.2. Absolute Position Reporting

The ikbd can also maintain absolute mouse position. Commands exist for reseting the mouse position, setting X/Y scaling, and interrogating the current mouse position.

4.3. Mouse Cursor Key Mode

The ikbd can translate mouse motion into the equivalent cursor keystrokes. The number of mouse clicks per keystroke is independently programable in each axis. The ikbd internally maintains mouse motion information to the highest resolution available, and merely generates a a pair of cursor key events for each multiple of the scale factor.

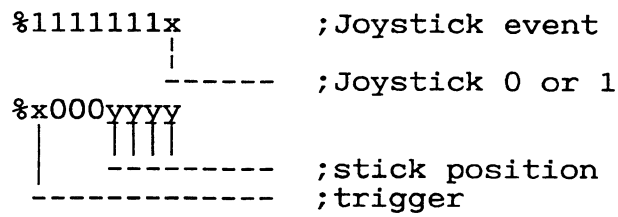
Mouse motion produces the cursor key make code immediately followed by the break code for the appropriate cursor key. The mouse buttons produce scan codes above those normally assigned for the largest envisioned keyboard (i.e. LEFT=0x74 & RIGHT=0x75).

5. Joystick

5.1. Joystick Event Reporting

In this mode, the ikbd generates a record whenever the joystick position is changed (i.e. for each opening or closing of a joystick switch or trigger).

The joystick event record is two bytes of the form:



5.2. Joystick Interrogation

The current state of the joystick ports may be interrogated at any time in this mode by sending an "Interrogate Joystick" command to the ikbd.

The ikbd response to joystick interrogation is a three byte report of the form:

```
0xFD            ;joystick report header
%x000yyyy       ;Joystick 0
%x000yyyy       ;Joystick 1
                ; where x is the trigger
                ; and yyyy is the stick
                ;  position
```

5.3. Joystick Monitoring

A mode is available that devotes nearly all of the keyboard communications time to reporting the state of the joystick ports at a user specifiable rate. It remains in this mode until reset or commanded into another mode. The PAUSE command in this mode not only stops the output but also temporarily stops scanning the joysticks (samples are not queued).

5.4. Fire Button Monitoring

A mode is provided to permit monitoring a single input bit at a high rate. In this mode the ikbd monitors the state of the Joystick 1 fire button at the maximum rate permitted by the serial communications channel. The data is packed 8 bits per byte for transmission to the host. The ikbd remains in this mode until reset or commanded into another mode. The PAUSE command in this mode not only stops the output but also temporarily stops scanning the button (samples are not queued).

5.5. Joystick Key Code Mode

The ikbd may be commanded to translate the use of

either joystick into the equivalent cursor control keystroke(s). The ikbd provides a single breakpoint velocity joystick cursor.

Joystick events produce the make code, immediately followed by the break code for the appropriate cursor motion keys. The trigger or fire buttons of the joysticks produce pseudo key scan codes above those used by the largest key matrix envisioned (i.e. JOYSTICK0=0x74, JOYSTICK1=0x75).

6. Time-of-Day Clock

The ikbd also maintains a time-of-day clock for the system. Commands are available to set and interrogate the time-of-day clock. Time-keeping is maintained down to a resolution of one second.

7. Status Inquiries

The current state of ikbd modes and parameters may be found by sending status inquiry commands that correspond to the ikbd set commands.

8. Power-Up Mode

The keyboard controller will perform a simple self-test on power-up to detect major controller faults (ROM checksum and RAM test) and such things as stuck keys. Any keys down at power-up are presumed to be stuck, and their BREAK (sic) code is returned (which without the preceding MAKE code is a flag for a keyboard error). If the controller self-test completes without error, the code 0xF0 is returned. (This code will be used to indicate the version/release of the ikbd controller. The first release of the ikbd is version 0xF0, should there be a second release it will be 0xF1, and so on.)

The ikbd defaults to mouse position reporting with threshold of 1 unit in either axis and the Y=0 origin at the top of the screen, and joystick event reporting mode for joystick 1, with both buttons being logically assigned to the mouse. After any joystick command, the ikbd assumes that joysticks are connected to both Joystick0 and Joystick1. Any mouse command (except MOUSE DISABLE) then causes port 0 to again be scanned as if it were a mouse, and both buttons are logically connected to it. If a mouse disable command is received while port 0 is presumed to be a mouse, the button is logically assigned to Joystick1 (until the mouse is reenabled by another mouse command).

9. ikbd Command Set

This section contains a list of commands that can be sent to the ikbd. Command codes (such as 0x00) which are not specified should perform no operation (NOPs).

9.1. RESET

0x80
0x01

N.B. The RESET command is the only two byte command understood by the ikbd. Any byte following an 0x80 command byte other than 0x01 is ignored (and causes the 0x80 to be ignored).

A reset may also be caused by sending a break lasting at least 200 mS to the to the ikbd.

Executing the RESET command returns the keyboard to its default (power-up) mode and parameter settings. It does not affect the time-of-day clock.

The RESET command or function causes the ikbd to perform a simple self-test. If the test is successful, the ikbd will send the code of 0xF0 within 300 mS of receipt of the RESET command (or the end of the break, or power-up). The ikbd will then scan the key matrix for any stuck (closed) keys. Any keys found closed will cause the break scan code to be generated (the break code arriving without being preceded by the make code is a flag for a key matrix error).

9.2. SET MOUSE BUTTON ACTION

```
        0x07
        %00000mss        ;mouse button action
              |||        ; (m is presumed = 1 when in
              |||        ;  MOUSE KEYCODE mode)
              ------     ;0wx   mouse button press or release
                         ; ||   causes mouse position report
                         ; ||   (wx only relevant if in
                         ; ||    absolute mouse positioning mode)
                         ; |--  1 -> mouse key press causes
                         ; |          absolute position report
                         ; ---  1 -> mouse key release causes
                         ;          absolute position report
                         ;100 = mouse buttons act like keys
                         ;
```

This command sets how the ikbd should treat the buttons on the mouse. The default mouse button action mode is %00000000, the buttons are treated as part of the mouse logically.

9.3. SET RELATIVE MOUSE POSITION REPORTING

```
        0x08
```

Set relative mouse position reporting. (DEFAULT) Mouse position packets are generated asynchronously by the ikbd whenever motion exceeds the setable threshold in either axis (see SET MOUSE THRESHOLD). Depending upon the mouse key mode, mouse position reports may also be generated when either mouse button is pressed or released. Otherwise the mouse buttons behave as if they were keyboard keys.

9.4. SET ABSOLUTE MOUSE POSITIONING

```
0x09
XMSB    ;X maximum (in scaled mouse clicks)
XLSB
YMSB    ;Y maximum (in scaled mouse clicks)
YLSB
```

Set absolute mouse position maintenance. Resets the ikbd maintained X and Y coordinates.

In this mode, the value of the internally maintained coordinates does NOT wrap between 0 and large positive numbers. Excess motion below 0 is ignored. The command sets the maximum positive value that can be attained in the scaled coordinate system. Motion beyond that value is also ignored.

9.5. SET MOUSE KEYCODE MODE

```
0x0A
deltax          ;distance in X clicks
                ;  to return {LEFT} or {RIGHT}
deltay          ;distance in Y clicks
                ;   to return {UP} or {DOWN}
```

Set mouse monitoring routines to return cursor motion keycodes instead of either RELATIVE or ABSOLUTE motion records. The ikbd returns the appropriate cursor keycode after mouse travel exceeding the user specified deltas in either axis. When the keyboard is in key scan code mode, mouse motion will cause the make code immediately followed by the break code. Note that this command is not affected by the mouse motion origin.

9.6. SET MOUSE THRESHOLD

```
0x0B
X       ;x threshold in mouse ticks
        ; (positive integers)
Y       ;y threshold in mouse ticks
        ; (positive integers)
```

This command sets the threshold before a mouse event is generated. Note that it does NOT affect the resolution of the data returned to the host. This command is valid only in RELATIVE MOUSE POSITIONING mode. The thresholds default to 1 at RESET (or power-up).

9.7. SET MOUSE SCALE

```
0x0C
X       ;horizontal mouse ticks per internal X
Y       ;vertical mouse ticks per internal Y
```

This command sets the scale factor for the ABSOLUTE MOUSE POSTIONING mode. In this mode, the specified number of mouse phase changes ("clicks") must occur before the internally maintained coordinate is changed by one (independently scaled for each axis). Remember that the mouse position information is available only by interrogating the ikbd in the ABSOLUTE MOUSE POSITIONING mode unless the ikbd has been commanded to report on button press or release (see SET MOUSE BUTTON ACTION).

9.8. INTERROGATE MOUSE POSITION

```
0x0D
        Returns:
                0xF7    ;absolute mouse position header
                BUTTONS
                        0000xxxx
                            ||||
                            |||--- right button down
                            |||     since last
                            |||     interrogation
                            ||---- right button up
                            ||      since last report
                            |----- left button down
                            |       since last report
                            ------ left button up
                                    since last report
                XMSB    ;X coordinate
                XLSB
                YMSB    ;Y coordinate
                YLSB
```

The INTERROGATE MOUSE POSITION command is valid when in the ABSOLUTE MOUSE POSITIONING mode, regardless of the setting of the MOUSE BUTTON ACTION.

9.9. LOAD MOUSE POSITION

```
0x0E
0x00    ;filler
XMSB    ;X coordinate
XLSB    ; (in scaled coordinate system)
YMSB    ;Y coordinate
YLSB
```

This command allows the user to preset the internally maintained absolute mouse position.

9.10. SET Y=0 AT BOTTOM

```
0x0F
```

This command makes the origin of the Y axis to be at the bottom of the logical coordinate system internal to the ikbd for all relative or absolute mouse motion. This causes mouse motion toward the user to be negative in sign and away from the user to be positive.

9.11. SET Y=0 AT TOP

```
0x10
```

Makes the origin of the Y axis to be at the top of the logical coordinate system within the ikbd for all relative or absolute mouse motion. (DEFAULT) This causes mouse motion toward the user to be positive in sign and away from the user to be negative.

9.12. RESUME

```
0x11
```

Resume sending data to the host. Since any command received by the ikbd after its output has been paused also causes an implicit RESUME this command can be thought of as a NO OPERATION command. If this command is received by the ikbd and it is not PAUSEd, it is simply ignored.

9.13. DISABLE MOUSE

0x12

All mouse event reporting is disabled (and scanning may be internally disabled). Any valid mouse mode command resumes mouse motion monitoring. (The valid mouse mode commands are SET RELATIVE MOUSE POSITION REPORTING, SET ABSOLUTE MOUSE POSITIONING, and SET MOUSE KEYCODE MODE.)

N.B. If the mouse buttons have been commanded to act like keyboard keys, this command DOES affect their actions.

9.14. PAUSE OUTPUT

0x13

Stop sending data to the host until another valid command is received. Key matrix activity is still monitored and scan codes or ASCII characters enqueued (up to the maximum supported by the microcontroller) to be sent when the host allows the output to be resumed. If in the JOYSTICK EVENT REPORTING mode, joystick events are also queued.

Mouse motion should be accumulated while the output is paused. If the ikbd is in RELATIVE MOUSE POSITION REPORTING mode, motion is accumulated beyond the normal threshold limits to produce the minimum number of packets necessary for transmission when output is resumed. Pressing or releasing either mouse button causes any accumulated motion to be immediately queued as packets, if the mouse is in RELATIVE MOUSE POSITION REPORTING mode.

Because of the limitations of the microcontroller memory this command should be used sparingly, and the output should not be shut off for more than <tbd> milliseconds at a time.

The output is stopped only at the end of the current "event". If the PAUSE OUTPUT command is received in the middle of a multiple byte report, the packet will still be transmitted to conclusion and then the PAUSE will take effect.

When the ikbd is in either the JOYSTICK MONITORING mode or the FIRE BUTTON MONITORING mode, the PAUSE OUTPUT command also temporarily stops the monitoring process (i.e. the samples are not enqueued for transmission).

9.15. SET JOYSTICK EVENT REPORTING

0x14

Enter JOYSTICK EVENT REPORTING mode (DEFAULT). Each opening or closure of a joystick switch or trigger causes a joystick event record to be generated.

9.16. SET JOYSTICK INTERROGATION MODE

0x15

Disables JOYSTICK EVENT REPORTING. Host must send individual JOYSTICK INTERROGATE commands to sense joystick state.

9.17. JOYSTICK INTERROGATION

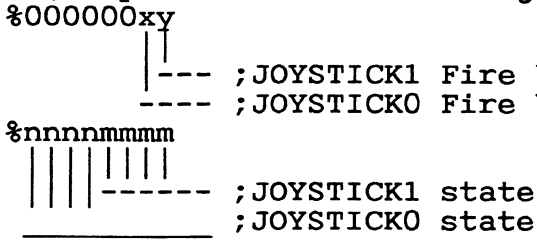
0x16

Return a record indicating the current state of the joysticks. This command is valid in either the JOYSTICK EVENT REPORTING mode or the JOYSTICK INTERROGATION MODE.

9.18. SET JOYSTICK MONITORING

```
0x17
rate    ;time between samples in hundredths of a second
```

Returns: (in packets of two as long as in mode)



Sets the ikbd to do nothing but monitor the serial command line, maintain the time-of-day clock, and monitor the joystick. The rate sets the interval between joystick samples.

N.B. The user should not set the rate higher than the serial communications channel will allow the 2 byte packets to be transmitted.

9.19. SET FIRE BUTTON MONITORING

```
0x18

        Returns:        (as long as in mode)
                %bbbbbbbb   ;state of the JOYSTICK1 fire
                            ; button packed 8 bits per byte,
                            ; the first sample is the MSB
```

Sets the ikbd to do nothing but monitor the serial command line, maintain the time-of-day clock, and monitor the fire button on Joystick 1. The fire button is scanned at a rate that causes 8 samples to be made in the time it takes for the previous byte to be sent to the host (i.e. scan_rate = 8/10 * baud_rate). The sample interval should be as constant as possible.

9.20. SET JOYSTICK KEYCODE MODE

```
0x19
RX      ;length of time (in tenths of seconds) until
        ; horizontal velocity breakpoint is reached
RY      ;length of time (in tenths of seconds) until
        ; vertical velocity breakpoint is reached
TX      ;length (in tenths of seconds) of joystick closure
        ; until horizontal cursor key is generated before
        ; RX has elapsed
TY      ;length (in tenths of seconds) of joystick closure
        ; until vertical cursor key is generated before
        ; RY has elapsed
VX      ;length (in tenths of seconds) of joystick closure
        ; until horizontal cursor keystrokes are generated
        ; after RX has elapsed
VY      ;length (in tenths of seconds) of joystick closure
        ; until vertical cursor keystrokes are generated
        ; after RY has elapsed
```

In this mode, joystick 0 is scanned in a way that simulates cursor keystrokes. On initial closure, a keystroke pair (make/break) is generated. Then up to Rn tenths of seconds later, keystroke pairs are generated every Tn tenths of seconds. After the Rn breakpoint is reached, keystroke pairs are generated every Vn tenths of seconds. This provides a velocity (auto-repeat) breakpoint feature.

Note that by setting RX and/or RY to zero, the velocity feature can be disabled. The values of TX and TY then become meaningless, and the generation of cursor "keystrokes" is set by VX and VY.

9.21. DISABLE JOYSTICKS

0x1A

Disable the generation of any joystick events (and scanning may be internally disabled). Any valid joystick mode command resumes joystick monitoring. (The joystick mode commands are SET JOYSTICK EVENT REPORTING, SET JOYSTICK INTERROGATION MODE, SET JOYSTICK MONITORING, SET FIRE BUTTON MONITORING, and SET JOYSTICK KEYCODE MODE.)

9.22. TIME-OF-DAY CLOCK SET

```
0x1B
YY      ;year (2 least significant digits)
MM      ;month
DD      ;day
hh      ;hour
mm      ;minute
ss      ;second
```

All time-of-day data should be sent to the ikbd in packed BCD format.

Any digit that is not a valid BCD digit should be treated as a "don't care" and not alter that particular field of the date or time. This permits setting only some subfields of the time-of-day clock.

9.23. INTERROGATE TIME-OF-DAY CLOCK

```
0x1C
        Returns:
                0xFC    ;time-of-day event header
                YY      ;year (2 least significant
                        ; digits)
                MM      ;month
                DD      ;day
                hh      ;hour
                mm      ;minute
                ss      ;second
```

All time-of-day data is sent in packed BCD format.

9.24. MEMORY LOAD

```
0x20
ADRMSB  ;address in controller
ADRLSB  ; memory to be loaded
NUM     ;number of bytes (0-128)
{data}
```

This command permits the host to load arbitrary values into the ikbd controller memory. The time between data bytes must be less than 20 mS.

9.25. MEMORY READ

```
0x21
ADRMSB  ;address in controller
ADRLSB  ; memory to be read
        Returns:
                0xF6    ;status header
                0x20    ;memory access
                data    ;6 data bytes
                data    ; starting at ADR
                data
                data
                data
                data
```

This command permits the host to read from the ikbd controller memory.

9.26. CONTROLLER EXECUTE

```
0x22
ADRMSB  ;address of subroutine in
ADRLSB  ; controller memory to be
        ; called
```

This command allows the host to command the execution of a subroutine in the ikbd controller memory.

9.27. STATUS INQUIRIES

status commands are formed by
inclusively ORing 0x80 with
the relevant SET command

Example:

```
0x88 (or 0x89 or 0x8A) ;request mouse mode
        Returns:
                0xF6    ;status response header
                mode    ;0x08 if RELATIVE
                        ;0x09 if ABSOLUTE
                        ;0x0A if KEYCODE
                param1  ;0 if RELATIVE
                        ;XMSB X maximum if
                        ;  ABSOLUTE
                        ;DELTAX if KEYCODE
                param2  ;0 if RELATIVE
                        ;XLSB if ABSOLUTE
                        ;DELTAY if KEYCODE
                param3  ;0 if RELATIVE
                        ;  or KEYCODE
                        ;YMSB if ABSOLUTE
                param4  ;0 if RELATIVE
                        ;  or KEYCODE
                        ;YLSB if ABSOLUTE
                0       ;pad
                0
```

The STATUS INQUIRY commands request the ikbd to return either the current mode or the parameters associated with a given command. All status reports are padded to form 8 byte long return packets. The responses to the status requests are designed so that the host may store them away (after stripping off the status report header byte) and later send them back as commands to the ikbd to restore its state. The 0 pad bytes will be treated as NOPs by the ikbd.

Valid STATUS INQUIRY commands are:

0x87	mouse button action
0x88 0x89 0x8A	mouse mode
0x8B	mouse threshold
0x8C	mouse scale
0x8F 0x90	mouse vertical coordinates (returns 0x0F Y=0 at bottom 0x10 Y=0 at top)
0x92	mouse enable/disable (returns 0x00 enabled 0x12 disabled)
0x94 0x95 0x99	joystick mode
0x9A	joystick enable/disable (returns 0x00 enabled 0x1A disabled)

It is the (host) programmer's responsibility to have only one unanswered inquiry in process at a time.

STATUS INQUIRY commands are not valid if the ikbd is in JOYSTICK MONITORING mode or FIRE BUTTON MONITORING mode.

10. Appendix A -- Scan Codes

The key scan codes returned by the ikbd are chosen to simplify the implementation of GSX.

GSX Standard Keyboard Mapping

Hex	Keytop
01	Esc
02	1
03	2
04	3
05	4
06	5
07	6
08	7
09	8
0A	9
0B	0
0C	-
0D	==
0E	BS
0F	TAB
10	Q
11	W
12	E
13	R
14	T
15	Y
16	U
17	I
18	O
19	P
1A	[
1B	]
1C	RET
1D	CNTL
1E	A
1F	S
20	D
21	F
22	G
23	H
24	J
25	K
26	L
27	;
28	'
29	`
2A	(LEFT) SHIFT
2B	\
2C	Z
2D	X
2E	C

2F	V
30	B
31	N
32	M
33	,
34	.
35	/
36	(RIGHT) SHIFT
37	{NOT USED}
38	ALT
39	SPACE BAR
3A	CAPS LOCK
3B	F1
3C	F2
3D	F3
3E	F4
3F	F5
40	F6
41	F7
42	F8
43	F9
44	F10
45	{NOT USED}
46	{NOT USED}
47	HOME
48	UP ARROW
49	{NOT USED}
4A	KEYPAD -
4B	LEFT ARROW
4C	{NOT USED}
4D	RIGHT ARROW
4E	KEYPAD +
4F	{NOT USED}
50	DOWN ARROW
51	{NOT USED}
52	INSERT
53	DEL
54	{NOT USED}
5F	{NOT USED}
60	ISO KEY
61	UNDO
62	HELP
63	KEYPAD (
64	KEYPAD)
65	KEYPAD /
66	KEYPAD *
67	KEYPAD 7
68	KEYPAD 8
69	KEYPAD 9
6A	KEYPAD 4
6B	KEYPAD 5
6C	KEYPAD 6
6D	KEYPAD 1
6E	KEYPAD 2

```
6F      KEYPAD 3
70      KEYPAD 0
71      KEYPAD .
72      KEYPAD ENTER
```