Hardward Memo #9 March 29, 1972

Bit Box

John Roe

(Alias Basic Box, New Demobox, Semi-Universal Logo Device Device, Thornton Box, "Controller") is about to have the formats of its input controller, turtle controller, and music box frozen in wire. This memo describes these; if you'd like it some other way, find me soon.

Micro-controller: already wired. Has a control state loaded in two ways:

6-bit: Send the magic character $0 \leq C \leq 177$ octal (This character is wired in, can be changed fairly easily. Suggest some.); the next non-magic character is put into control register.

7-bit: Send a character ≥ 140 octal; b_1-b_5 of that character are put into control register. The two modes are chosen by a jumper on the controller card. The controller behaves as follows:

$$\text{CSR:}\quad \underbrace{b_1\ \ b_2}_{A}\ \ \underbrace{b_3\ \ b_4}_{B}\ \ b_5$$

Device pointer (1-4) is always alternated between A and B, A first after CSR load. Currently b_5 is ignored: it could (1) if =1 cause no devices to get data; or (2) if =1 cause all four devices to be multiplexed, 1 first. (1) is easy. (2) requires annoying changes to controller and would be slow.

Turtle: about to be wired. Commands will be same as demobox turtle, except of course for device pointer.

Music box: design far along. Notes will be same as now. Control state is taken as any character following a #, 43 octal.

CS:	b_1	2	3	4	5	6
	# of voices to update 1-4		play voice Ø staccato (st.)	1 st.	2 st.	3 st.

Voices will be loaded as now. If you want box to shut up, you must fill it with silent notes!

A voice not being updated will continue to play the note it contains rather than switching between it and the previous, as now. b_{3-6} could be used in some other way; if you have ideas discuss with me soon.

Space (40_8) loaded as a note will be silent.

Percussion: the two characters 41 (!) and 42 (") will be two percussion sounds, as yet undetermined. Suggest some, I will reject any hard to make. It has yet to be determined whether a percussion sound will be started: (1) whenever its character appears at the music box (and is loaded as a silent note into the player in whose turn it falls); (2) same as (1) except that the character does not advance the voice sequence pointer, so that it need not disable a voice temporarily; may be annoying to time; (3) same as (1) except that sound is started when that batch of voices is loaded into player; (4) same as (2) except that sound is started when the batch of voices being loaded when 41 or 42 comes ~~are~~ is loaded into the player. If (3) I may require that the character in that voice be changed before that same sound may be started again. Please note: if lack of consensus or overwhelming reasoning, I will choose easiest for me, viz. (1). But (4) may be nearly as easy.

Other devices envisioned are:

Terminal interface: input to system comes from terminal, output to that device # goes to terminal. Uses up a device slot, but may save you a TTY port on system. Could have a different flavor similar to demobox and exec. Output to any device also goes to terminal unless the terminal device is commanded to shut up.

Relay box: like present one. b_1-b_6 control relays which possibly control outlets.

D to A device: b_1-b_6 are converted to analog voltage for your pleasure. Could have a scheme for sequentially loading serveral DACs or a higher resolution one. Might include a DC servo and power amp, or have external one.

Cassette device: generates feeps for and possibly controls and/or gets input from a cassette recorder to generate canned demos, etc. Could also be used as cheap keypunch recorder. May not be useful at all.

Input device: controlled by output to it. This version of basic box can input to system from only one device, so any input multiplexing must be done within one device. Possible modes (either analog or digital for each):

1) spew)
2) one character) as now
3) send when input changes
4) multiplex amongst inputs as 1) or 2)
5) as 4) and 3), and send tags too
6) send every time t
7) when some other device tells it to.

These latter devices will not be seriously started until first three are winning; comments about them and suggestions for more would be most welcome.

JSR

HARDWARE MEMO 9B

Terminal Device

May 29, 1972

John Roe

Modes:

a) terminal gets all output unless shut off
b) terminal gets only output addressed to its device

Wakes up in a), not shut off. Further, system gets all terminal input which has priority over all other input. Terminal input sent with B_8=0.

When terminal device wants to send, it grabs priority line when next high; does not raise it until out of characters to send.

Output to terminal device:

22_8 DC 2 implies b)

23_8 DC 3 implies a), on

24_8 DC 4 implies a), off

Buffers input and output such that they can change data rates.

HARDWARE MEMO 9C

6/14/72

THORNTON BOX INPUT DEVICE

John Roe

Sends data from turtle touch sensors, feedback pots, etc. back to system as TTY characters. Sends all characters with $b_8 = 1$.

Controlled by output to it:

b_6	b_5	=> Input Source:	
1	1	Serial Ganglion	Strobed in by ext. device
1	0	Parallel Ganglion	
0	1	Analog	
0	0	Internal Data Bus	

Deselected sources are completely ignored.

b_4	b_3	=> Input Timing
1	1	Send only one word, now
1	0	Send any new data, i.e. when changes
0	1	Send at maximum rate
0	0	Send at 1/4 maximum rate

The last loaded by the source is the data sent. b_2 and b_1 are transferred to source device for its pleasure; could be used to select subdevice, etc.

Input data is restricted to values between 40 and 177 octal, inclusive. Values less than that will not be sent even if requested by source device. Lower case (≥140) characters are sent and cause the source device to pre-empt the terminal device for the next character; if the source device has not presented that character before the TTY transmitter is emptied, that pre-emption disappears. Thus iff desired, sources can tag their data by preceding it with some lower-case character.

Internal Data Bus mode exists in order that devices may relay data to the input device for synchronization and transmission by it. Since the bus is shared with the microcontroller it is expected that only command mode (equivalent to $b_4 = b_3 = 1$ mode) or "send after you get a character" mode will be sweet.