AUDIBLE COMPUTING, Part 2 --- Masa Kasahara

What would happen if we only relied on our hearing when using a computer? Masa will continue his experiment based on the feedbacks he received in the previous talk. He will discuss **Prototype 1** in detail and also touch upon his view of human and computer languages inspired by Ting's papers on computers and Zen.

What is Audible Computing?

- My (tentative) Definition: To perform any kind of computing tasks based on an audible interface rather than by visual or other physical means.

Why Forth?

- Theoretically, any language can be used for this project as long as it can be represented in alphanumerical format. – Most of the languages, however, are not designed with audible computing in mind.
- Any characters can also be used, theoretically. – Chinese characters, for example, are too complicated to implement. – Is there a Chinese Morse Code?

John Rible's Input on Open Firmware:

- It didn't happen. The way Open Firmware initializes devices is the complete opposite of what Microsoft does.
- Without Forth in Firmware, the idea of using Morse Code as an interface with the motherboard won't work. This idea has been abandoned.

Prototype 1:

- Instead of building a Forth system with a Morse Code interface built-in, how about a terminal program written in Forth with a Morse Code interface? This terminal program can be connected to another Forth system. This approach is probably easier since the Morse Code interface is running as an application though this separation may not be apparent in Forth programming.

Serial Terminal with a Morse Code Interface: What is needed?

1. Find a good Forth implementation. It should run on a regular PC. The source should be available and well documented. It would also be nice if the system can run from a floppy.
2. Find a cheap PS/2 mouse to hook up to a squeeze key (Morse Code keyer). Eventually, the left and right mouse switches will be connected to the contacts of a squeeze key.

Good Forth Implementations:

- Kevin’s Input: FPC
- Ting’s Input: eForth under MS-DOS

Alan's Input on the Interface

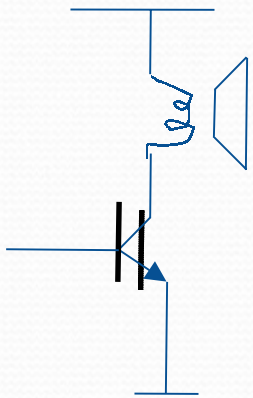
- Printer Port is better: PS/2 ports do not exist on many newer notebooks. - The PS/2 mouse idea has been dropped.
- One issue with a printer port: It is not accessible under Windows 2K or XP. The spooler blocks access to the port.
- Is USB a better choice? It is widely available on PCs today and you can access it under any OS.
- Does a regular mouse driver with USB legacy mouse support in BIOS cover USB mouse? This is important if you want use a USB mouse under MS-DOS.

Requirements:

- For clarity there will be two tones for Morse Code: one for incoming, the other for outgoing. The frequencies should be adjustable to suit personal preferences.
- Any unrecognized characters in input stream will be simply discarded. – No complicated terminal emulations, but a simple teletype emulation (TTY).
- Any unrecognized character in keying stream will be interrupted with a correction symbol for an immediate re-transmission, which makes it easier to use.

Problem with a PC Speaker

- No dual tones possible with the standard PC speaker implementation.



Solution: PWM Digital Linear Amplifier

- Turns an on/off speaker control into PWM Digital Linear Amplifier utilizing CPU and PIC (Programmable Interrupt Controller).
- No additional hardware is required.
- It only burns CPU cycles. The question is how much computer overhead can be allocated for this.

20KHz Sampling

Oscillator Implementation:

- All sample-based.
- Each oscillator has a few different samples for frequency multiplied by a few different volume levels.
- The output from two oscillators are simply added at each sampling interval.

Prototype 2:

- Prototype 1 is focused on the feasibility of the concept while Prototype 2 is focused on actual implementation of the concept.
- Find Open Firmware for PCs. In the worst case, use my SPARCstation 4. Open Boot is a Forth system.
- The target Forth system should be modified for Audible Computing, especially in editing (or human interface) area. Preferably, implement a switch to change the mode of operation so that the original interface is kept intact. Or, if a keyboard and/or video interface is not present, it automatically switches into Morse Code mode.

Ting's Input:

- Every PC has a sound card and speakers.
- If we use Windows, we don't have to worry about implementing the sound system. We can use a MIDI implementation with two oscillators.

Prototype 1 (Modified):

- A terminal program runs under Windows with the Morse Code Interface. This terminal program can be connected to another Forth system.
- I found a sample TTY program for Windows at Microsoft web site:

  http://msdn2.microsoft.com/en-us/library/ms810467(d=printer).aspx

  It is a fairly complete program with source code and detailed explanation of the code.

Status of SPARCstation 4 (SS4)

- The target machine doesn't have to be a SS4, but it contains Open Boot, a sister of Open Firmware.
- The source code for a similar UltraSPARC Station is available, which could be helpful if I ever want to play with it.
- 3 perfect SCSI hard disks for experimenting were purchased at the HSC parking lot sale for $1 each.
- Fedora 3 port exists under the Aurora Project.