machine intell TA LS..qxd 4/28/2009 11:02 AM Page 57

at my site at www.machineinteltech.com. Let me also say that I prefer to refer to the synthetic brain as an "android" and not a robot, because it will emulate human linguistic interaction in a visual world.

Many think we should be running robots/androids from networked PCs. My opinion is that remote processing has major drawbacks. The processing system should be encapsulated in the device and not be a remote system because of transmission bandwidth, radio frequency licensing and interference issues, etc. For practical reasons, it is better on board.

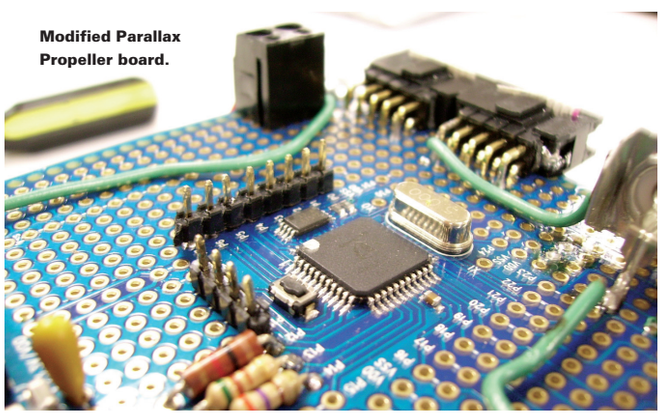
The product components on my website are set up so that you can interconnect processors and they can work in concert. Four Propellers give you 32 totally independent simultaneous processing streams (not time-shares). The PC has a problem: it is a time-slice-based system (even with multiple cores in the system), and it is not a true multiprocessing stream design. Even a multi-core PC has a massive operating system overhead to parcel the running software between the cores. Also, the chips and supporting hardware quickly get out of the price range of robotics researchers.

My children and I are working together to create a platform with a good price/performance value for android intelligence. At some point, I will release a PC program in some form as a conceptual model for others to pick apart to see how it works. We are building a platform that's more powerful than a PC in terms of its parallel processing and is powerful enough for me to port this new technology to. When complete, I will impart this to the hobby public.

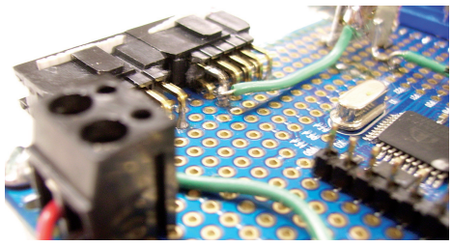
The brain does not store information as we think of it in a computer. It stores logic about the information and that's why it's so inexact. I believe we store logic packets that can be connected to other logic packets. As you learn, it takes less space to describe something new because you can use what you already have to further interconnect.

ROBOT: How does your technology work?

DR. JIM: On a disk, at the end of a sector or track, you have an error correcting code [ECC] so that if you lose bits in that data stream in that sector, you can reconstruct it. I own a patent that performs a similar function to encode logic streams (Patent no. 5267328). The concept is about how error-correcting codes are generated. Error-correcting codes describe information to which they are attached in sufficient detail such that if some information is lost, it can be reconstructed and filled back in. My software takes that concept and uses it to combine all this abstract information in logic streams as opposed to data streams. I create several ECCs that point to tables in memory that point to tables that point to tables. It bounces all around from a technical standpoint. It uses something designed for error-correcting codes on media. I take just the ECC and I store it. There are only maybe, 50 bytes in the ECC, and I store pointers between these ECC packets. Any reader can learn more about error-correcting code concepts by Googling "error correcting codes."



Modified Parallax Propeller board.



Partial view of a modified Parallax Propeller board used in the android project.

This unique approach allows me to put information into an abstract form. The ECC codes are, in a sense, analogous to the logic packets in the human brain. You can look at a tree but cannot tell me the shape of every leaf, but you did see it. You looked at the abstract part of it. I am able to compress the information and move it into the abstract realm digitally using an approach resembling an ECC.

My goal is to bring this out within the year. It is building on my past work and will move the hobby forward. If you look at patent number 5267328, you'll get a sense of my background. That patent shows how to encode geometric shapes into abstract form.

ROBOT: Have you tested the synthetic brain in a verbal interaction?

DR. JIM: Yes; on a PC, but I really want it on the Parallax Propeller.

ROBOT: How far have you proceeded with the Propeller?

DR. JIM: At least four Propeller chips are needed, but they are relatively inexpensive at less than $10 each, and you would need $39 for a prototyping Propeller board. That is incredibly affordable. They are reduced instruction set machines, and that makes them fast. Most instructions occur in 50 nanoseconds on each Cog.

The Propeller is the exact solution to mate with this technology. I have come up with an almost limitless interconnection capability; check out the details on memory expansion at www.machineinteltech.com. The minimum is four, but it is set up so that you can put together an almost unlimited number of Propellers. I'm writing an operating system for it that is already running in skeletal form. I created a servo driver for the platform as well. It can drive 20 servos per board with no external logic. You could have 100 servos out there—or solenoids. Hobbyists can incrementally build as massively parallel a computer as they wish. We will have the ability to interface with USB peripheral devices for non-volatile mass storage. Everything will be on board the robot.

My goal is to provide a research platform that can be used by serious hobbyists and researchers. It is also in the same price range as computerized toys. I am bringing the concepts I first learned with mainframes to the android research arena.

ROBOT: We look forward to sharing more of your android project with our readers. Thank you for this interview.

DR. JIM: My pleasure.

Editors note: A "build manual" for constructing the initial hardware platform for the android project is posted at www.machineinteltech.com and www.botmag.com/issue17. ◎

Links

Machine Intelligence Technologies, www.machineinteltech.com

For further information, please see our source guide on page 81.