Introduction

Tell me three times...

This issue marks the beginning of the *Journal of Forth Application and Research* as a multidiscipline medium focused on Forth. These first papers document the usage of Forth in industry (GE, TRW, Rockwell), at universities (McMaster University and the University of Massachusetts), and by government agencies (NBS and the Department of Agriculture). Why Forth? Our experiences in computer applications indicate the need for an alternative software and hardware technology than that traditionally practiced by programmers and engineers. Forth is an alternative by being qualitatively and quantitatively different from other programming languages. First, it is a user definable environment, and second, it allows tight control over that environment.

These programming qualities are important to me at the Laboratory for Laser Energetics. They also served as my impetus to chair the Rochester Forth Conferences for the past three years. For the last two years we have had invited speakers addressing special topics, and this year's theme was robotics. Three of our invited speakers, Johnson, Slater and Rippey, have papers in this issue of the *Journal*. Three other authors from that Conference, Franklin, Dumse and Thompson, also have their presentations in the *Journal*.

Given a brief definition of Forth, what is a robot? One of our invited speakers, Don Davenport, in his talk on "The Industrial Robot", used the Robot Institute of America's definition:

> "[A robot is] a programmable, multifunction manipulator designed to move material parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks."[1]

Another of our invited speakers, Charles Moore, in his presentation of "They, Robots", preferred a different view:

> "Let us agree that a robot can act upon its environment. This rules out business systems, image systems, signal-processors. There remain telescopes, vehicles, almost anything with moving parts. It is axiomatic that a robot act in real time."[2]

For this *Journal* issue we will take the more expansive view.

Robotics presents a cutting edge to many disciplines. As a mechanized device a robot has elements of mechanical, electrical and possibly hydraulic engineering. It may require sensory input. To have the wherewithal to operate autonomously, it may take instructions from a paper tape reader, as in a simple numerical control (NC) application, or from an adaptive multiprocessor network. Whereas NC presents a minimal software burden, multiprocessor networks and fully concurrent compilers are not well understood.

An awareness of this software bottleneck is growing. There have been 15 years of research into a variety of robotic programming systems. Some systems behave like programming languages, while others are presented as CAD/CAM packages for directing motion. All of them tend to be slow or cumbersome for users, implementors, or researchers. Furthermore, it is unreasonable to dedicate a mainframe computer, with its megabytes of memory, to each axis of a robot, regardless of how inexpensive memory is. Obviously each of these groups needs appropriate, resource-conserving tools, but how?

Forth was designed for the real-time control of moving parts [2, 3], to paraphrase its developer, Charles Moore. The papers collected in this issue provide examples of Forth's applicability to the entire range of robotics. Herbert Johnson and Piero Bonissone's paper, "Expert System for Diesel Electric Locomotive Repair", explores the realm of artificial intelligence and offers the promise of robots understanding their environments. Although begun in LISP, their project was completed in Forth. Their paper is followed by Dan Slater's, "A State Space Approach to Robotics" which draws examples from precision servo control systems for movie special effects and spacecraft antenna checkout. His premise is that state space methods are applicable to both robotics and general programming.

The next three papers deal more closely with Don Davenport's definition of industrial robots. "Control of a Cartesian Robot", by Judy Franklin *et al.*, describes a closed-loop digital servo system used to control an axis of a stationary robot. Randy Dumse's paper, "A Robotic Application for Contamination-Free Assembly", makes use of related ideas in building a cleaning workstation. He also alludes to the power of a Forth-based language. This theme is echoed in Douglas Thompson's paper, "Improvement of a Human/Robot Interface through the Use of FORTH".

All of these ideas are being brought together at the National Bureau of Standards as described by William Rippey in "FORTH and Automation Research at the National Bureau of Standards". Rippey's is one of several groups at NBS working towards the totally automated, integrated, factory of the future.

William Caskey's paper, "Dynamic Processing of Analog Data Using an Apple Computer" is a departure from the robotics theme. He discusses the inexpensive automation of a gas chromatograph used for soil analysis. A technical note by Nicholas Solntseff on "Vectored Versions of Forth Compiling Words", and the abstracts from the 1983 Rochester Forth Applications Conference round out the issue.

Unlike Lewis Carroll's map in *Hunting of the Snark*, we began with an "absolute blank" and then spent a year filling in the "North Poles and Equators". I wish to thank all of these authors and the reviewers for their efforts and the production staff of the Institute for Applied Forth Research, Inc. for bringing the *Journal* to life, and this issue to print.

If you thought Forth applications were challenging before, you haven't seen anything yet. What I tell you three times is true!

Lawrence P. Forsley
University of Rochester
September 21, 1983

[1] Davenport, Don, "The Industrial Robot", *Proc. of the 1983 Rochester Forth Applications Conference*. Institute for Applied Forth Research, Inc., Rochester, NY; p 5.

[2] Moore, Charles, "They, Robots", *Proc. of the 1983 Rochester Forth Applications Conference*. Institute for Applied Forth Research, Inc., Rochester, NY; p 10.

[3] Moore, Charles, "Forth: A New Way to Program a Mini Computer" *Journal of Astronomy and Astrophysics*, Supplement 15, September, 1974, AAAS, New York; pp 497-511.