Introduction

Real Time Systems

It has become a tradition to have one issue of the *Journal* each year devoted to the theme of that year's Rochester Forth Conference, which I chair. Last year's Conference dealt with robotics as did Volume 1 Number 1 of the *Journal*. This year's Conference focused on real time systems, and three of these papers were presented there.

By presenting papers at the Conference which are published in the *Journal* we serve several purposes. First, the Conference allows us to draw the best authors doing the latest work in many fields. In addition, the authors benefit immediately from their audience interactions. Finally, Conference participants are rewarded with quality presentations of peer-reviewed material. We plan to strengthen this relationship between Conferences and the *Journal* in the future, and we look forward to our continued authors' and readers' enthusiasm for what we are doing. If you have comments, please address a letter to the editor care of the managing editor, Mr. Jonathan Ross.

To further broaden the scope of our efforts we welcome Dr. Hans Nieuwenhuÿzen as the European Editor. Dr. Nieuwenhuÿzen is an astronomer at the University of Utrecht, in Utrecht, Holland, where he has been active in Forth for several years as an author, lecturer, programmer and teacher. The two technical notes in this issue, "Separate Headers" and "Identifiers" were written by him and an associate from Utrecht.

The other articles in this issue deal with real-time systems. But what are real-time systems and what is their relationship to Forth? One of the invited speakers this year, Dr. Michael Starling, in his talk "Of Widgets and Clock Ticks" suggested that we must make a distinction between "relevant time programming", which is limited by the response time of the user, and real-time programming, where, "we are actually *connected* to the real world." This real world connection occasionally operates on millisecond or microsecond timescales, and may require the synchronization of tens or hundreds of devices.

Thus, the ideas of limited time, and also, limited resources, even if only in "critical regions" of code, become important. As the *Conference Program* noted:

> Forth's first application was telescope control requiring precise alignment, in real-time, of large optical systems. This difficult application coupled with insufficient computer resources forced Charles Moore to find a different approach to real-time programming. Today, even with the availability of 32 bit microcomputers and soon, 256K RAMs, there are more applications than ever taxing computer resources and programmers' talents. [1]

Starling also recognized the problems of real time programming when he stated:

> The implementation of complex real-time systems involves the successful marriage of hardware and software for the purpose of solving a real problem. It requires a programmer versed in many aspects of the overall problem. Implementation is aided greatly by tools having extraordinary power while allowing access to all levels of the system. [2].

He then states the the advantage of such tools is that one may "discover how the hardware and the application REALLY work."

These qualities of real time synchronization, limited resources, programmer discovery, and an implied project evolution, permeate the four papers in this issue. Two of the papers deal with robotics in the classroom, which implies limited resources and the need for student self-discovery. Cotterman, *et al.* from Wright State University built a low cost robot arm for the classroom, and developed a sophisticated undergraduate laboratory around it. Formisani, *et al.*, then at Stanford, designed and built an intelligent parts carrier in only 9 months for their masters theses. Both papers were first presented at the Conference.

The paper by Anthony, from General Electric, describes an elegant laser beveling application and presents codes for manipulating the XY positioner and rotary table similar to Cotterman's robot arm codes. His time constraints were such that a single Forth task sufficed. On the otherhand, McGuire, previously with the Kitt Peak National Observatory, gives an overview of one of the first multi-tasking Forth systems used for real-time control and data acquisition. This system has been used for several years with its most impressive application beaing a CCD camera used at observatories in the US and Chile. McGuire also presented at the Conference.

In summary, these papers capture the essence of programming in real-time with Forth. Perhaps what is most noteworthy is not the difficulty of the applications, so much as the clarity of their Forth concept. Yet, these papers do not have unreserved praise for Forth. As Cotterman, *et al.* note, Forth needs a better packaging of existing and additional functions to be more widely useful. Nonetheless, these papers present diverse and continuing uses of Forth.

Thanks

I have been very pleased with the help and support that the authors and reviewers have given me, and I appreciate the superb efforts of the *Journal* staff. I would like to thank our many sponsors, without whom these papers would not have made it to print. I urge our readers to further participate in the *Journal* and join their colleagues on the title page as authors or on the masthead as reviewers and sponsors.

Lawrence P. Forsley
University of Rochester

References

[1] Forsley, Lawrence, "Invited Speakers", *1984 Rochester Forth Applications Conference Program.*Institute for Applied Forth Research, Inc. Rochester, New York: June, 1984.

[2] Starling, Michael, "Of Widgits and Clock Ticks", *Proceedings of the 1984 Rochester Forth Applications Conference.* Institute for Applied Forth Research, Inc. Rochester, New York: November, 1984.