# Australian Forth Symposium.

# Workshops:

# Project Management & Forth.

The purpose of this workshop session to consider whether Forth is a suitable language for large projects. The success of a large project depends partly on selecting the right resources and on good project management. This workshop session shall examine four engineering and commercial projects where Forth was used as the major programming language. Participants shall describe their project and consider the following:

* Forth version used.
* Size of the program.
* Time it took to:
    * Identify requirements.
    * Specify details.
    * Code software..
    * Debug software
    * Document the project.
    * Commission the project.
* The number of people working on the project.
* The success of others especially maintenance staff reading the software.
* Project management techniques employed such as
    * Project tracking with PERT charts etc.
* Employment of re-useable code.
* Type of system eg Real Time control, graphics interface etc.
* Interface to other systems including machine code etc.

**What is are the limitations of using Forth on large projects?**

This question shall be considered at this workshop

An example of one of the four projects to be presented is the following project by Nigel Lovell from the University of NSW.
If you are interested in developing large projects in Forth then come to this workshop.

Philip Mallon.

# ASYST PROGRAMMING FOR BIOMEDICAL ENGINEERING APPLICATIONS*

NIGEL LOVELL and BRANKO CELLER
Biomedical Engineering Laboratory
Department of System and Control
School of Electrical Engineering and Computer Science
University of NSW, Kensington 2033.

## Abstract

The ASYST** language is a Forth based language designed to provide a full range of mathematical, statistical and interface functions for the scientific user. Integral to the language are functions for computer interfacing via serial (RS232/422) devices, functions for interfacing using the IEEE-488 (GPIB) bus, and drivers for interfacing using plug in data acquisition cards.

Features of the ASYST language are described with reference to biomedical engineering application areas. Particular reference will be drawn to software developed in our laboratory for acquisition and analysis of ECG data. We have named this suite of ASYST programs CARDIOSYS. CARDIOSYS was the first application program to be written in the ASYST language in Australia.

The advantages and disadvantages of using ASYST over conventional languages will also be discussed.

## Keywords

ASYST, Forth, Integrated Software, Data Acquisition, Turnkey Systems, ECG Analysis, Blood Pressure Analysis, IBM XT/AT

* Presented at Australian Forth Symposium, University of Technology, Sydney, May 19-20, 1988.

** ASYST is a registered trademark of ASYST Software Technologies.

## Introduction

Preceding the cost reduction in laboratory computers, the largest proportion of a researchers budget was devoted to the purchase of the computer and dedicated data acquisition cards. The purchase of software was given a lower priority. In addition, software products on the market at the time were limited in terms of versatility and speed. The result was that the researcher would develop his/her own suite of computer programs to acquire data, analyse data and produce hardcopy. Typically the programs would be written in Fortran or C, and would take man months or even man years to develop and debug.

With the low cost of the microcomputer and its associated hardware, the researcher/clinician has at his disposal an important workstation for acquisition, analysis and display of physiological data. Equally as significantly is the attitude that it is no longer an extravagance to consider the purchase of a software package which integrates all major aspects of scientific computing. Indeed it is more cost effective to adopt this approach than to develop a suite of programs from scratch. In our laboratory, a low cost IBM/AT compatible computer with data acquisition facilities (Data Translation DT2801A data acquisition card) was coupled with ASYST, a high level programming language.

This paper will present various biomedical engineering application areas, with particular reference to work conducted in our laboratory, to illustrate the advantages the ASYST language offers over conventional programming techniques.

ASYST is a Forth based language. It utilises the concept of threaded in line code, and is heavily stack oriented. The language is different from standard Forth in that the language addresses the needs of the scientific community. ASYST functions include, data acquisition, GPIB control of instruments, graphics and windowing support, file storage and retrieval, and an interactive/compiled programming environment. Analysis functions include matrix reduction, data smoothing, local maxima/minima detection, complex FFT's and inverse FFT's, convolutions, and nonlinear curve fitting. Statistical functions include ANOVA's, histograms, hypothesis testing and regression analysis.

## Application Programming

ASYST was chosen as the developmental language in our laboratory due to its ability to generate graphics, support data acquisition and data analysis. In addition, ASYST can be run in an interpretative or a compiled mode. A program may be tested by typing it in interactively. Once debugged it can then be compiled. This method has the advantages of an interactive language, that is, ease of code development and debugging, as well as the benefits of a compiled language, namely, fast execution speed.

ECG acquisition and detection routines were written in ASYST, to provide custom functions specific to the detection and display of heart interval data. These functions we have called CARDIOSYS. In our laboratory, the heart interval data is used in the study of the dynamics of vagal and sympathetic control of cardiac rhythm (Ref.2). Multiple channels of ECG data are sampled and stored to disk. The raw data can then be plotted or scrolled through in order to determine areas of interest. A post processing menu allows the data to be decoded so as to determine PP and PR heart intervals. CARDIOSYS is transparent to the ASYST work environment, that is, all the normal ASYST functions are still operative except for the use of the function keys **<F1>** - **<F10>** and **ALT <F10>**. The user may type in any ASYST command, or load any ASYST program while running CARDIOSYS. This is useful if the user wishes to experiment with different methods of display or analysis of the ECG data. CARDIOSYS is a menu driven package. There are seven menus from which the operator may choose. These seven menus are all accessed from another menu, the MAIN MENU, by using the following function keys.

**<F1>** HELP MENU

**<F2>** SETUP MENU

**<F3>** ACQUISITION MENU

**<F4>** ANALYSIS MENU

**<F5>** PLOT MENU

**<F6>** SCROLLER MENU

**<F7>** UTILITIES MENU

Although some of the features of CARDIOSYS are specific to ECG analysis, and in particular to heart period analysis, it is envisaged that the salient features of CARDIOSYS will apply to many other applications. These features include a menu driven data acquisition program allowing the user to select such parameters as trigger source, number of samples, number of channels, A/D gain and sampling frequency. CARDIOSYS will sample data continuously to the hard disk at a sampling rate of 27 kHz on an unfragmented hard disk. This figure is an order of magnitude greater than any other commercially available subroutine for data streaming on an IBM/XT. The high speed disking routine was written in assembly language and called from ASYST. If similar disking algorithms were to be written purely in ASYST code the disking throughput would be reduced to 5 kHz. Another novel feature of CARDIOSYS is the data scrolling facility. This feature allows the user to scroll through previously acquired data in an interactive fashion. The user may change the physical location and number of channels to be displayed, the channel gains and offsets, and the rate and direction of scrolling. Scrolling is performed using a special waveform scroller card (WFS200PC by Dataq Instruments). This allows waveforms to be acquired and displayed at up to 8 kHz. This is two orders of magnitude greater than comparable scrolling rates using the standard bit mapped graphics card.

Routines based on CARDIOSYS have been implemented at St. Vincent's Hospital, Sydney, for use in acquiring, analysing and displaying blood pressure waveforms. Likewise an ASYST system has been installed at the National Drug and Alcohol Research Centre, Prince of Wales Hospital, Sydney, for use in EMG studies. This suite of programs utilises the waveform scroller card extensively, to display up to eight channels of data as it is simultaneously being stored to disk. Many other ASYST packages are in use in hospitals, universities and industry throughout Australia. Indeed, with the release of ASYST version 2.0 and support for the high performance data acquisition cards, which can acquire data at up to 250 kHz (DT2821-G from Data Translation), new application areas are rapidly unfolding for the general purpose microcomputer.

## Discussion

The philosophy behind CARDIOSYS and indeed the other applications described, of using a high level software package, such as ASYST, on a low cost garden variety computer, certainly suffers from several deficiencies. One limitation is the slow speed of graphics generation when using standard bit mapped displays. CARDIOSYS minimized this problem by displaying waveforms using a hardware scroller board which is supported by the ASYST language. Another limitation is due to the nature of Forth code, in that there is no optimization stage when compiling. This means that ASYST routines will generally execute slower than similar routines written in other compiled languages. ASYST to some extent circumvents the speed problem by incorporating code written in assembler for time critical events such as direct memory access (DMA) and programmed input/output performed by data acquisition cards. Also, Forth code is array orientated, therefore considerable time savings can be achieved if data is analysed using operations which manipulate the whole array at one time.

In the future, with the NOVIX processor on the market, it should certainly be possible to have a version of ASYST which is NOVIX compatible, and thus overcome the minor speed deficiencies of the language.

Another limitation is that one must learn the ASYST programming language. As ASYST is such an extensive package, a considerable amount of time is required to learn its independent language. If the user is familiar with Forth, however, a considerable time saving will be gained. Once learnt though, most programming tasks can be coded more concisely and with greater speed using ASYST, than a lower level language such as Fortran or Pascal. In addition, the user need only learn one software product for his/her complete scientific requirements, as ASYST is an integrated scientific language.

Using ASYST the researcher can benefit from a low cost, and expedient solution to integrating a computer into his/her laboratory. The computer can quickly become a powerful scientific workhorse for data acquisition, manipulation, reduction and display.

## References

(1) Lovell, N.H., Celler, B.G., *CARDIOSYS - A Low Cost System for the Acquisition, Analysis and Display of ECG Data*, A'asian Phys. Eng. Sci. Med. (1987). **10**, 4.

(2) Parker, P., Celler, B.G., Potter, E.K. and McClosky, D.I., *Vagal Stimulation and Cardiac Slowing, J. Automatic Nervous System. (1984).* **11**, 226-231.