A Forth Controlled Oceanographic Instrument

Everett F. Carter and H. T. Rossby
University of Rhode Island

Our research group has developed a microprocessor-controlled neutrally bouyant float for making measurements of ocean currents, such as the Gulf Stream in the Atlantic Ocean. The float freely drifts at a predefined density level (which is typically at about 800 meters depth), measuring temperature and depth and recording acoustic navigation signals (emitted at precisely known times from sound sources that were placed in the ocean in advance. These sound sources can be as much as 1200 km from the float). After a programmed amount of time (typically 30 to 45 days) the float surfaces and transmits the recorded data to shore via satellite.

The first generation floats were programmed in assembly language. This was reasonable since all the floats followed essentially the same sampling strategy. The new generation of floats are to be much more flexible. Different floats may have different sensors attached, or may have to sample in different ways. In order to allow greater flexibility in the software a Forth interpreter was developed for the floats.

The Forth interpreter is modeled after the 1979 standard, but is customized for the floats in several ways: (1) The memory space for the microprocessor (the MC146805) is only 8k and the data as well as code must reside in this space. Hence the memory restrictions are severe. (2) The Forth interpreter must be ROM based since no mass storage is available. (3) The limited instruction set of the processor forced us to use a direct threaded interpreter instead of the more traditional indirect threaded interpreter.

The RAFOS float system

We have developed a new type of instrument for the measurement of ocean currents [4],[5]. The float (which is called a RAFOS float) housed in a Pyrex pipe that is about 5 feet long and 4 inches in diameter (see Figure 1). Inside the tube are the electronics which consists of a radio, an acoustic receiver, and the microprocessor board (Figure 2).

The float accurately follows a predetermined density level in the ocean (typically at about 800 meters depth). As the float drifts in the current it acquires navigation data, measures the pressure, temperature and potentially other environmental parameters. The navigation information is in the form of the time of arrival of sound pulses that were previously placed in the ocean. These sound sources can be as much as 1200 kilometers from the float and still be heard due to the presence of a sound conducting channel (which acts like a wave guide) at the level were the floats typically drift. By triangulating on two or more sound sources, we can determine the position of the float at a given time. After a programmed amount of time in the ocean, which is typically 30 to 45 days, the float surfaces and transmits its data to our laboratory on shore via satellite. Because of the high cost of ship time compared to the cost of a float, we do not recover the floats; they are considered expendable.

As the last of the first generation floats transmitted their data in the summer of 1984, we began thinking about how the next generation float would be utilized. These floats will be used as before, but with adaptive sampling strategies or for shorter intervals interactively with a ship. An interactive scenario could involve for example, putting a few floats in for a few days, recovering them from the same ship which then puts in more floats in a way that depends upon the data received from the first floats.

The original RAFOS floats were all programmed in assembly language, all the floats had the same program. When using the newer floats, different floats may require different sampling programs or the sampling scheme may not be able to be chosen until just before the floats are deployed, in either case assembly code for these floats would be a impractacle. Forth was the obvious solution to the problem of how to efficienty get the sampling program into the floats.

The MC146805 processor

The whole system is controlled by a CMOS microprocessor, the Motorola MC146805. The MC146805 is a CMOS 6805 which is essentially a modification of a 6800. The modifications to the 6800 were made by Motorola in order to make it especially useful for process control applications. Unfortunately some of these modifications (such as throwing out half of the registers and reducing the instruction set) make it harder to design a Forth interpreter.

There are several variants of the 6805. The one that we use in the floats has a memory space of only 8k bytes. This is a severe restriction upon the system since there is no mass storage so that the data as well as the float control program must reside in that space. We have partitioned the system so that half of it consists of ROM that contains the Forth interpreter and the other half is RAM. The RAM portion contains two things: (1) application code (in Forth) that is downloaded to the float before it is used, and (2) the data that the float gathers when it takes its measurements.

RAFOS Forth

Although Forth was the obvious choice for a new software system for the floats, the hardware and system constraints mentioned above prevented the use of a commerically available Forth interpreter. This being the case, we designed a Forth interpreter for the floats. The implementation details of the interpreter are described in [1].

The interpreter is a direct threaded interpreter that follows the 1979 standard [2] as much as possible. The deviations from the 1979 standard are due to the fact that there is no mass storage and all the code must reside in ROM. Memory constraints are also important, since everything the managing software and the data itself must all fit within 8k. The tight memory constraints prevent the implementation of the full language, instead a major subset (approximately 90%) of the language is in the ROM. The use of a direct threaded interpreter [3], and implicitly setting the Forth word WIDTH to 3, helps to keep the headers small so that memory usage is minimized.

Some of the Forth interpreter is in RAM. The inner interpreter is in the base page (0000 to 00FF hex) because this part of the memory space is actually within the processor so that access to it is much faster than the external RAM. The instructions referring to the base page are also fast because fewer bytes are used in those instructions. Also the reduced instruction set of the 6805 required the use of some self-modifying code, which must reside in RAM in order to work.

When the system is booted up, reset, or upon execution of COLD the initial values of the user and system variables, the inner interpreter, and the self-modifying code are all copies from ROM to their executable locations in RAM.

Conclusion

The Forth controlled floats have yet to actually be used at sea. So far they have been used on the lab bench to evaluate the new sensors that will be on board the floats which are to go into the water this coming fall. The usefulness of Forth in these floats has prompted us to consider upgrading the CPU board on them to use a version of the 6805 that has an address space of 64k bytes. When we do this, then there will be plenty of room for a complete Forth system on the floats.

The lab bench floats are not in their glass tubes so communicating with the CPU is just a matter of plugging into the serial I/O Port. With the seagoing floats the details of the communication link is still an open issue. It is envisioned to either use a watertight connector that goes through the glass wall or orthogonally mounted (in order to avoid interference) optical links.

The Forth controlled lab bench floats floats are also serving as a useful teaching aid for teaching the rest of the research group the Forth language. Although Forth has a long history in process control and data acquisition, its use in oceanography is relatively novel. Having RAFOS float CPUs with Forth ROMs readily available has proved invaluble as a teaching aid for learning Forth.

Bibliography

[1] Carter, E.F., Forth Goes to Sea, Dr. Dobb's Journal, in press 1986.

[2] Forth Standars Team, Forth-1979 Standard, Forth Interest Group, 1980.

[3] Ritter, T. Varieties of Threaded Code for Language Implementation, BYTE, Sept. 1980, pp. 206-227.

[4] Rossby, H.T., A.S. Bower, and P-T. Shaw, Particle Pathways in the Gulf Stream, Bulletin of the American Meteorological Society, V. 66, No. 9, Sept 1985, pp. 1106-1110.

[5] Rossby, H.T., D. Dorson, and J. Fontaine, The RAFOS System, Journal of Atmospheric and Oceanic Technology, in press 1986.

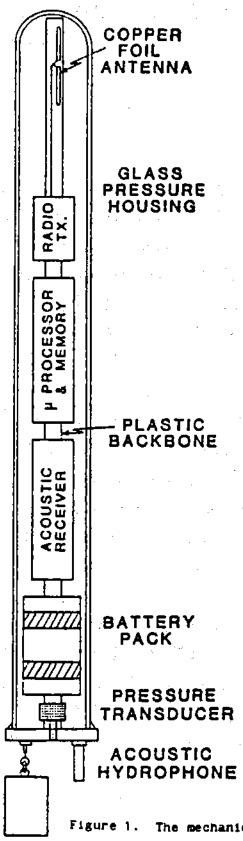

Figure 1. The mechanical arrangement of the RAFOS float. A glass pipe, rounded at the upper end, is closed at the lower end with an aluminum endplate. All of the components, antenna, radio transmitter, microprocessor, acoustic receiver and battery, are mounted on a PVC spar prior to insertion in the pipe. FROM [5]

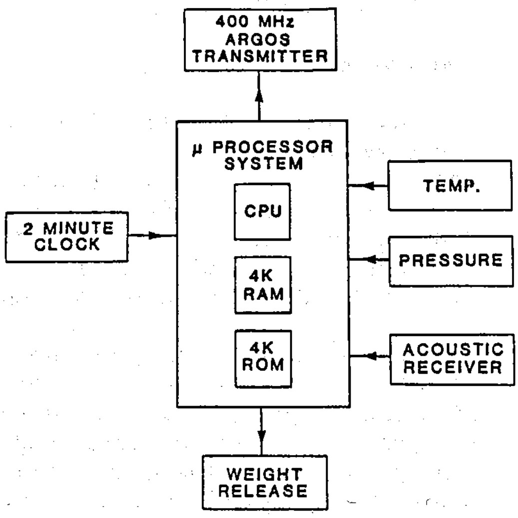

Figure 2. Functional block diagram of the RAFOS float electronics. Under software control and the precision 2-minute clock the microprocessor (Motorola 6805) controls the timing of all functions. FROM [5]