48783

Radio-Electronics®

SUPERCONDUCTIVITY: THE LATEST NEWS

$2.25 FEB. 1988
IN CANADA $2.75

TECHNOLOGY - VIDEO - STEREO - COMPUTERS - SERVICE

BUILD AN AUDIO/VIDEO SWITCHER

PUT YOUR A/V SIGNALS ON THE RIGHT TRACK

MAKE YOUR OWN PC BOARDS

NEW STEP BY STEP GUIDE EXPLAINS HOW IT'S DONE

THE FUTURE OF SHORTWAVE RADIO

NEWS FROM THE 1987 HIGH-FREQUENCY BROADCAST CONFERENCE

PROGRAMMABLE LOGIC DEVICES

AN ATTRACTIVE ALTERNATIVE TO STANDARD TTL PARTS

COMPUTERDIGEST

DESKTOP PUBLISHING SYSTEMS FROM SIMPLE TO SUPER

BUILD THE ULTIMATE CONTROL SYSTEM

THE REACTS 7000 ADVANCED MODULAR CONTROL SYSTEM

02
0 71896 48783 1

A GERNSBACK PUBLICATION

Radio-Electronics ADVANCED CONTROL SYSTEM

A complete, sophisticated control computer that is capable of operating almost every appliance or system in your home, and more!

H. EDWARD ROBERTS, M.D.

THIS IS THE FIRST IN A SERIES OF ARTICLES IN WHICH WE WILL introduce REACTS, the *R*adio-*E*lectronics *A*dvanced *C*on*T*rol *S*ystem. Over the next few months, we will build a control/robotics computer called the REACTS 7000, which is based upon the DataBlocks, Inc. *Altair II* system, a complete line of modular control elements currently available for personal and industrial-control computers. We'll also build all the peripherals the computer will need in order to perform a wide variety of control functions. In addition, we will show you how to use your computer in *real* applications.

Control computers

Control computers differ from conventional computers in several important ways. One of those differences, the ability to easily interface with a huge variety of external devices, gives the control computer the potential, over the next few years, to revolutionize our lives in more ways than anything yet produced by mankind.

In a typical home, there are applications for dozens of control computers. Some simple applications include controlling appliances; adjusting the heating, air conditioning, and/or humidity; minimizing power consumption; and ultimately running a robot lawnmower or vacuum cleaner so you won't have to.

REACTS 7000 is designed to be completely modular, with each module containing the circuitry to perform one or more complete functions. It also uses conventional programming languages (BASIC, C , assembly, etc.). Each subunit or module in the computer plugs into every other module. For example, the first module we will build is the central processor/computer module. That module is a complete stand alone system that includes its own memory, serial port, disk, vectored interrupt, real-time clock, system clock, memory-expansion hardware and all necessary buffers. In other words, it is a complete personal computer. Following that, we will build modules that contain semiconductor disk systems along with integral PROM programmers, complete CRT terminals, A/D converter modules, stepping-motor modules, etc.

The goal of this series of articles is to make *you* the system designer. You do the designing by simply selecting what modules you need for your application. Those modules are then stacked together in any order to create a custom control computer for any application. You need only to pick a language and write the program. Surprisingly, most control applications require simple programs.

Central to REACTS is the soft-hardware concept. Soft-hardware is simply hardware that can be changed or altered as easily as software. The author first developed the concept at MITS while designing the Altair computer, but only recently has it been possible to exploit it in a practical way. Over the next few months you will see how we achieve soft-hardware using straightforward, well-established principles. Indeed, the main requirement for soft-hardware is to make sure that each module's operation doesn't interfere with the operation of any other module. Additionally, each module must be self-sufficient mechanically and electrically, Further, each module should provide shielding to meet FCC requirements, and each module must provide its own motherboard.

CPU module

The REACTS CPU module consists of a microprocessor, 64K static RAM, 32K EPROM disk system, serial I/O port, vectored interrupt system, completely buffered bus drivers, crystal-controlled clock, 1-megabyte memory-expansion subsystem, and internal sense switches. If you aren't familiar with any of those terms, don't give up! Stick with us as we are going to discuss each item individually and will explore the engineering philosophy used in each decision.

Cost and reliability are important considerations with any computer system, but to a large extent they determine where and when it is practical to use a control computer. For instance, it is not practical to use a $20,000 dedicated computer to provide security and environmental control for your $100,000 home. On the other hand, if the same job could be done by a computer system that sold for $5, nobody could afford to be without one. Similarly, a system that would automatically drive your car to work but was only 99.5% reliable would not be very interesting.

Sources

The following items are available from DataBlocks, Inc., 579 Snowhill Road, Glenwood, GA 30428. Or call (800) 652-1336; in Georgia call (912) 568-7101: DP-CPU—design package of schematics and instructions, $10.00; PC-CPU—PC board for CPU module (includes DP-CPU design package,$37.00; PC-CLK—PC Board for clock, $18.00; SYS-PROM—the REACTS operating system (enhanced SB-80) installed on a 32K UV-erasable PROM (includes operating system documentation), $44.00; REC-CPU—complete kit of parts, PC boards, IC's, connectors, for CPU module (does not include clock or system PROM), $147.00; REC-CLK—complete clock subsystem including all parts, PC boards, NiCd battery, and connectors, $43.00; and REC-SYS—All of the above, $218.00. An Elpac power supply is also available for $49.00. Please add $10.00 postage and handling per order. GA residents must add appropriate sales tax.

Our goal is to design a machine that is affordably priced and as close as possible to being 100% reliable. That is accomplished by eliminating mechanical subsystems and using special connectors. Cost is controlled, to a large extent, by the soft-hardware concept; you only include what is required for the particular application.

There are some problems along the way with developing an "ultimate" control computer. First, most of us don't have a $250,000 microprocessor-development lab. We have to be able to develop the hardware and software we need on the target system (a target system is the final product of a development project). But, that requires the development of both hardware and software that may only be needed for development and then scrapped when finished.

Fortunately, the soft-hardware approach allows us to use our target system as the development system. We do that by including the development modules (CRT controllers, PROM burners, etc.) during the development stage and then removing the unneeded development modules when finally we install the target system. That approach is especially economical for those who will build multiple systems.

Most board-level computers are designed to work with custom software because that is the easiest to implement. The approach we will take is to use a disk operating system even on the most minimal system. That allows us to use a disk-operating system with all its inherent power. There are a number of disk operating systems from which to choose, but a CP/M-like system is familiar to both MS-DOS users as well as CP/M users. For that reason, we are using the DataBlocks disk-operating system which is a superset of CP/M; indeed, all standard CP/M software will be compatible with our system.

Included in the operating system are all the utilities needed for development, such as drivers for the PROM programmer, drivers for printers, debug routines, etc. Those special utilities can be left out of the final target system if desired, to minimize system size and cost.

Design principles

The following discussion will give you an idea of the design concepts used in engineering REACTS. It will also give you a feel for the tradeoffs made in the design of the system.

The microprocessor used in the CPU module is a version of the Z-80. That is probably the most popular and widely used microprocessor ever made. It has become the standard in the control industry, and more personal computers have been built using it than any other microprocessor. It isn't the fastest of the microprocessors—indeed some of the newer high-performance microprocessor systems will out-perform it by a factor of 20 to 1. But of the modern microprocessors it is the easiest to understand, and it does not have any unpredictable or "funny" interfacing quirks. That is especially important to the non-expert designer. As we proceed, you will see how we get around the processor-speed problem by using the principle of distributed processing; that is the use of multiple processors in a single system. The multiprocessor concept is based on the assumption that computers are free. It turns out that the principle is reasonably valid in practice, since the CMOS Z-80 is less expensive than a number of other IC's in REACTS. We will also use a number of other techniques that will greatly expand the power and capability of the system.

REACTS CPU

To get an idea of how fast the REACTS CPU module is in terms of control functions, let's look at a simple example. Assume we needed to turn on or off 2000 switches in a predefined manner. The high-speed version of REACTS would be capable of doing that at a rate of 500 times a second, or approximately 1-million switch operations per second.

The microprocessor used is CMOS. Indeed, all components in the module are CMOS. That increases cost, but it makes battery power, either emergency or continuous, easy. In addition, the noise immunity is improved, and less heat is generated so no fans are needed. Further, less-expensive power supplies can be used and the system can be used in relatively confined quarters without the problem of overheating.

The minimum system memory is 64K of static CMOS RAM, but the system will address a total of 1 megabyte. Static RAM adds to the cost of the system, but it does allow for easy battery powering and makes the system more predictable in a multiprocessor and/or control environment. The addressing scheme is straightforward with the exception of the expanded memory. The expanded memory is based on a paging scheme that allows the computer to switch pages of memory 32K bytes at a time. We will discuss some specific uses of that memory in future articles. It is interesting to note that the page-switching system allows the system to operate at effective direct-memory-access speeds of 10 gigabytes per second. As a comparison, if we could read a 40 megabyte disk that fast, it would only take 4 milliseconds to read the whole disk!

A conventional RS-232 port is included in the basic CPU module. That port can be used to connect to a terminal, modem, or any other standard RS-232 device. The baud rate and signal characteristics are under software control and can be modified from within the program.

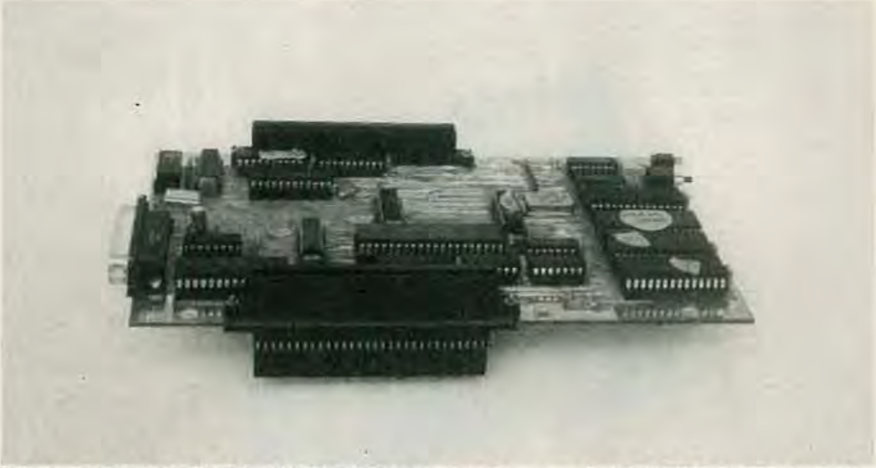


FIG. 1—EACH MODULE'S MOTHERBOARD interfaces to the REACTS system bus via feed-through Molex connectors.

The CPU module supports 9-levels of interrupts: one *N*on-*M*askable *I*nterrupt, or NMI, and 8 vectored levels. An interrupt-driven system is especially useful in process control.

There is somewhat of an aura of intimidation associated with interrupts. But actually, they make programming simpler and much faster in many applications. To use an interrupt, you simply pull the line low. That halts the program that is currently running and causes a jump to a special subprogram in memory; its much like a GOSUB command in BASIC. The computer executes the interrupt program and then returns to the original program. Interrupts provide two advantages to the system designer. The first is that the interrupting device can be serviced at random; that is, the program doesn't have to keep checking to see if a service is needed. Second, the interrupt can be serviced instantly; it doesn't have to wait for the main program. Vectoring simply means that the interrupts have levels of priority and that a high-priority interrupt can interrupt a lower priority one. Indeed, it is possible to have a number of interrupts waiting for service in a busy system. In later articles, we will see some detailed examples of practical uses of the interrupt system.

The basic CPU module includes sense switches. Those are simply switches that can be set and read by the program. They are actually a "poor man's" keyboard. An example of where they are useful is in the development vs. target system. In those systems, the computer checks to see the setting of the sense switch on power up, and from that determines whether it should look for a terminal or start executing a program. In a real target system, the sense switches would be used to select which program is loaded at auto start-up.

REACTS disk

In order to meet our design criteria, even the most minimum system will contain a disk. That is achieved in our system by using a UV-erasable PROM disk. That disk is seen as a disk by both the external hardware and software. It has all the attributes of a write-protected magnetic disk, save one: it operates at blinding speeds. In later articles, we will build larger semiconductor disk systems as well as a PROM programmer that allows you to burn your own PROM disk system for the computer.

REACTS makes extensive use of RAM and PROM disks. Semiconductor disks are significantly more reliable than conventional magnetic disk drives since there are no moving parts. They also consume much less power and are smaller. If large amounts of data need to be stored, then a magnetic disk becomes more attractive. In later articles, we will build a miniature floppy disk that's appropriate for mass storage and that will be compatible with our system. The overwhelming advantage of the semiconductor disk is speed.

The random-access time of a modern high-speed hard-disk system is approximately 25 milliseconds. The random-access time of our disk is approximately 10 microseconds, or approximately 2500 times faster. That is one of the reasons why our system is capable of outperforming some of the bigger and more expensive pro-

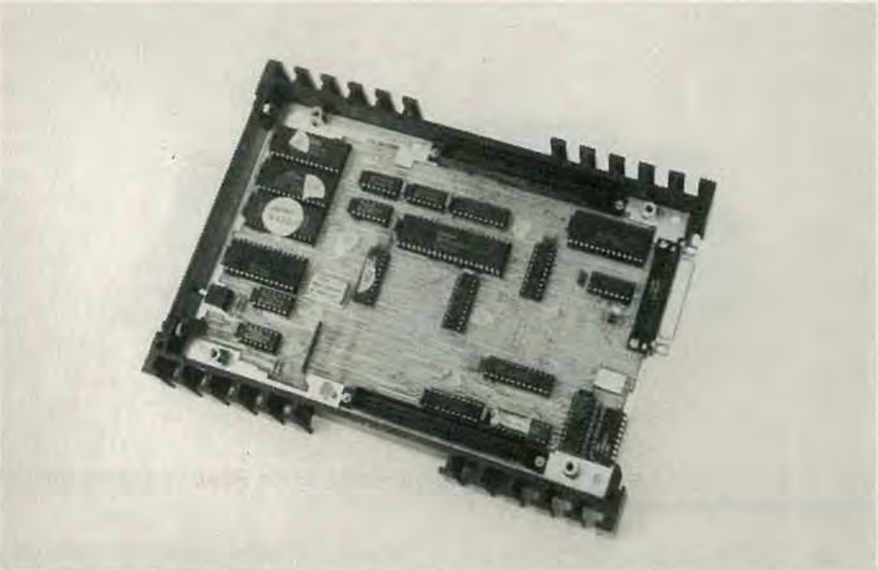
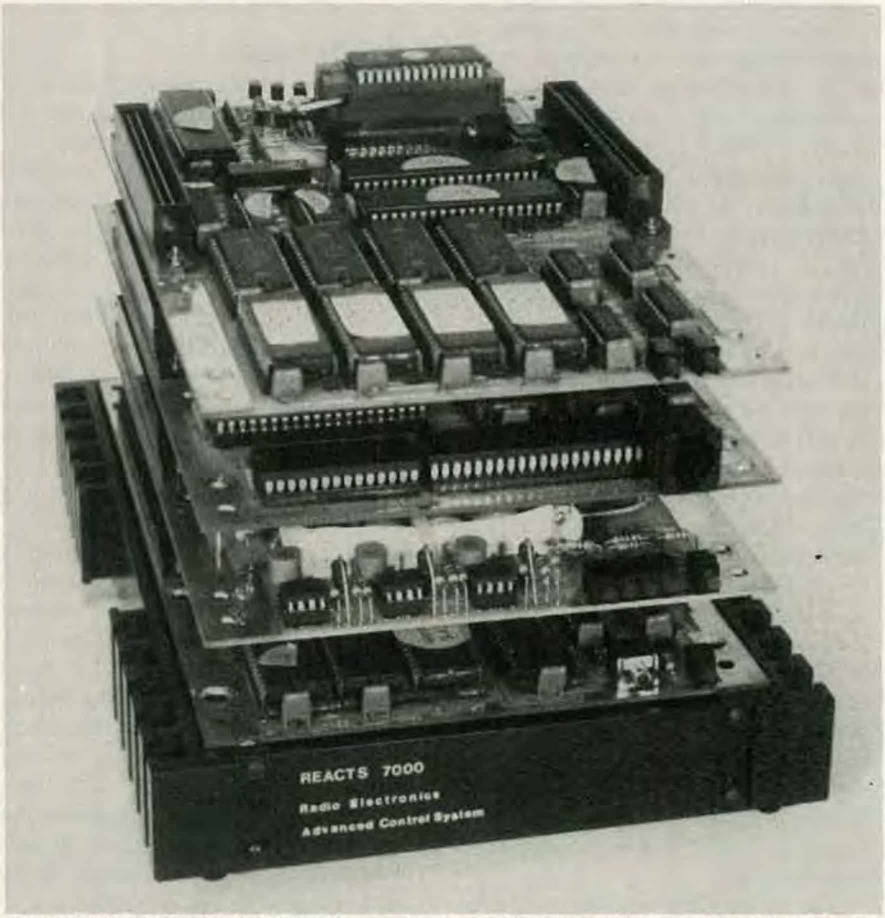


FIG. 2—THE EXTRUDED ALUMINUM CASE is designed to comply with FCC shielding requirements.



THE REACTS 7000 control/robotics computer system consists of series of stackable modules. Here, a four-module system consisting of the CPU, a power supply, a CRT/printer interface and a PROM programmer is shown. The modules are shown out of their shielded cases.

cessors. Finally, semiconductor disk systems are much easier to understand and much easier to use when designing custom software.

In order to maintain the soft-hardware concept, the REACTS bus is driven by CMOS drivers. No more than 1 or 2 CMOS loads are ever placed on the bus by any one module. That buffer system also allows for disconnecting the processor from the bus. That is a necessary condition in multiprocessor schemes. Buffers are not the most exciting topic, but their proper use is critical in a multi-processor, soft-hardware system so they will be addressed as we procede.

The standard module card is 8 inches by 5.3 inches and connects to the 120-line system bus using special feed-through Molex connectors. See Fig. 1. Those two connectors allow the modules to be stacked together in any order. The bus that we are using is the Altair-II bus developed by DataBlocks. Our system is designed to be fully compatible with all the existing DataBlocks modules and software. At the present time, there are literally dozens of different modules available that use the DataBlocks Altair-II bus.

REACTS case

Each module can be provided with its own shielded case that meets FCC standards. If you desire, you can mount the finished, unenclosed assembly inside a conductive case, which also meets FCC requirements. Figure 2 shows a module in its extruded aluminum case; note that the rear panel has been removed for clarity. While that case is unique to REACTS, it is compatible with the standard Altair-II system or equivalents.

A word about FCC standards is appropriate at this time. Each builder is responsible to make sure he meets the FCC requirements. All modules in this series of articles are tested to the most stringent FCC requirements using self-contained cases. Nevertheless, it is your responsibility to verify that your system doesn't interfere with any other service.

The real-time clock is plugged into the CPU-module board. That is done to keep cost at a minimum; the clock need only be installed if an application requires it. An on-board NiCd battery is used to provide backup power for the clock. The clock provides time, date, month, year, and can be used to generate interrupts to the main system. We will make extensive use of the clock in some of our future articles.

That's all we have room for this time. When we next meet we continue our look at REACTS and show you how to build the first of our modules, the CPU. **R-E**