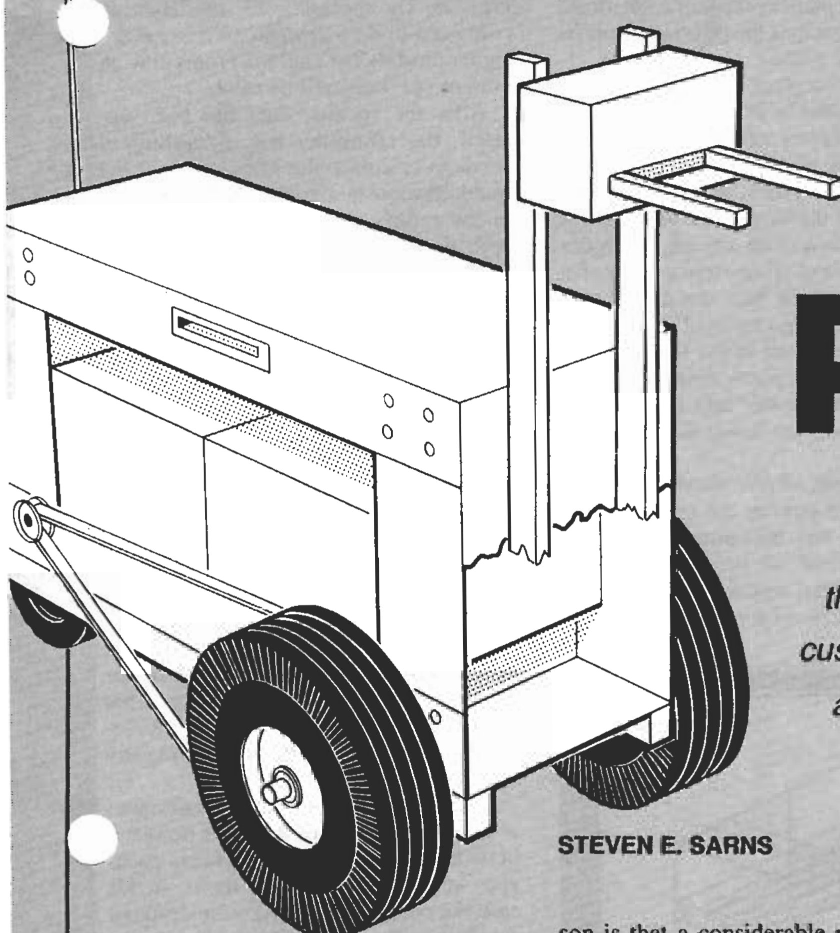

R-E ROBOT

You needn't be satisfied with a robot that looks and acts like all the rest. You can customize our robot to your heart's content—and share your designs with other readers!

STEVEN E. SARNS

THE BIG ONE FINALLY CAME THIS AUthor's way: a job providing both financial reward and a fascinating challenge for his company, Vesta Technology. The project: To design a robot, including a control computer, an arm, and additional subsystems for motion control, navigation, and operator input/output. While designing the robot, we discovered much about the personal robot industry. For one, it appears to be dominated by expensive robots with limited capabilities. We felt that a new approach could make a home robot more affordable and more exciting.

Designing a robot requires expertise in a number of areas, including mechanics, electronics, and computer hardware and software. In order to augment Vesta's limited abilities in the field of mechanical engineering, we enlisted Stock Drive Products to aid our development effort. That company is the major supplier of mechanical components to the industrial robotics market. See the Sources box for their address.

The cost of a robot

Stop for a moment and consider why personal robots are so expensive. One reason is that a considerable markup takes place at each point in the distribution chain. A manufacturer's purchasing department must have a secure supply of parts, so it may be willing to pay higher prices to attain that security. The hobbyist, however, has the advantage of being able to buy from less-expensive sources of parts. He can, for example, take advantage of surplus outlets, thereby eliminating middlemen; the result is a substantial savings over manufacturers' prices.

As for the controller, we designed a complete low-cost single-board computer that is highly compatible with the IBM-*PC*. Our approach emphasizes the use of flexible electronics that allow you to customize your robot with available mechanical parts.

By providing the electronic-control system and minimizing mechanical costs, we believe that building a personal robot can be both entertaining and affordable. In the upcoming series of articles, we will show you how you can adapt our designs to your problems.

The main components of our system are the single-board development system, a control/sensing board, and control software. Because the electronics systems are efficient and adaptable, you are free to interface them with whatever mechanical system meets your needs. The systems software that we have developed (and are still developing) is quite sophisticated, but the applications programming is left to you.

The bottom line is that we are not offering a kit for the type of ready-to-assemble robot that so many other companies offer; rather, we are suggesting that you can build the robot that you really want or need by integrating our control system with your mechanical design.

Overview

As we discuss the specifications of the **R-E** Robot, keep in mind that you can build your robot with other components, and in other configurations.

Our robot is powered by two 12-volt lead-acid batteries; it has a top speed of five miles per hour. Although we used utility batteries, we could have used auto or motorcycle batteries. Circuitry that indicates when power is low is included onboard, as is a 117-volt AC battery charger.

The robot's drive system consists of two independent 10.5-inch pneumatic tires that are connected to two toothed belt drives and to two 1/20-horsepower DC torque motors. A caster mounted at the rear provides lateral stability and ease of movement.

The robot is equipped with sensors for measuring temperature, light, and sound.

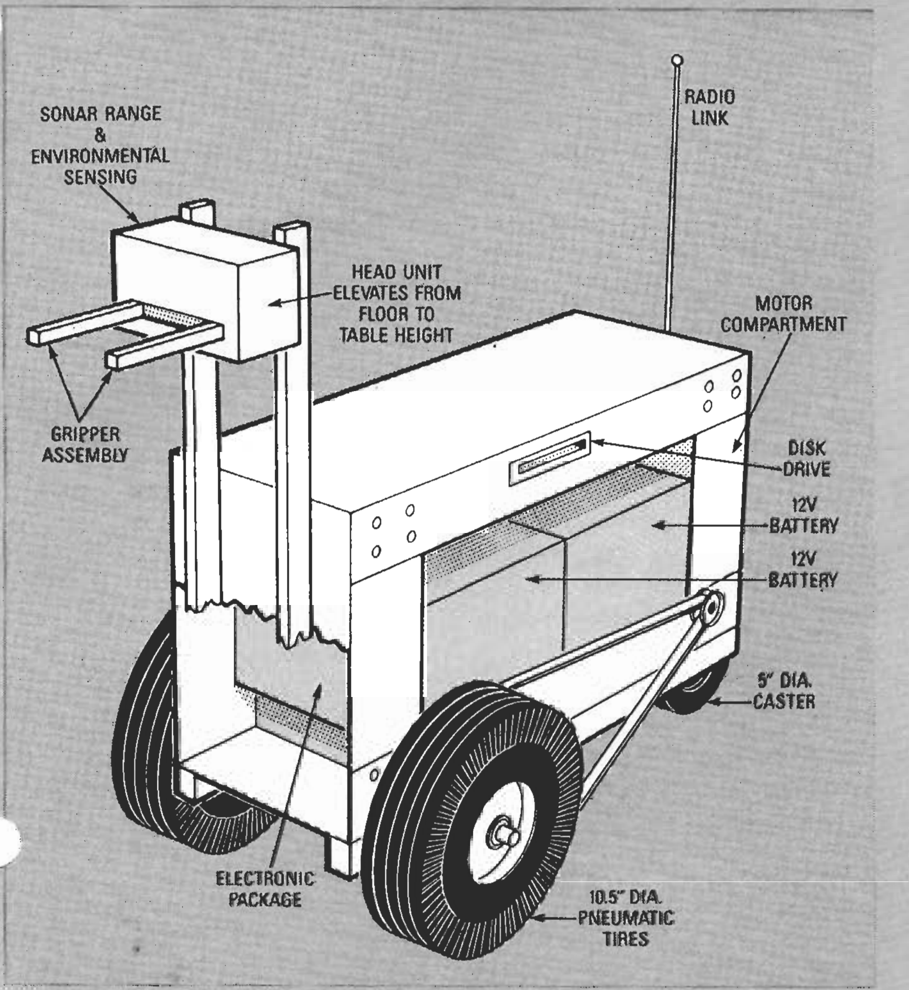

FIG. 1—R-E'S PERSONAL ROBOT has a unique mechanical configuration. This artist's conception shows the overall structure of our prototype. The robot can be modified to suit just about any application.

Microswitch collision detectors and sonar ranging, usable at distances as great as 20 feet, are also provided.

The robot lacks a traditional robot arm. Instead, it features a powerful gripper that rides a vertical track at the front of the unit. The arm, as shown in Fig. 1, somewhat resembles an industrial forklift. While some flexibility is sacrificed using that approach, some important advantages are gained. For one thing, the mechanical design is greatly simplified. That means that greater lifting capacity could be provided without greatly increasing cost. The gripper is capable of vertical travel from floor level to about table height.

In addition, the robot has options for an RF link, a speech synthesizer, and even a speech-recognition system.

e RPC

The hardware that makes it all possible is the RPC (*R*obotic *P*ersonal *C*omputer). The heart of the RPC is a highly integrated Intel 80188 microprocessor; it significantly reduces costs by including—in the IC package—many support devices that are external to the 8088 (used in a true-blue IBM-*PC*). The entire computer occupies a PC board that is less than eight inches on a side.

The interface between the RPC and the I/O unit is an adaptation of the IBM-*PC* bus. Signals from that bus are available at a 60-conductor IDC connector. That bus allows prototype circuits to be built without using special prototype cards.

Control boards

The RPC controls three custom boards. Board 1, shown in Fig. 2, contains most of the basic control circuits; with it and the RPC, the robot is capable of unsupervised operation. Board 1 controls the two torque (drive) motors. Each PWM (*P*ulse-*W*idth *M*odulation) drive-motor controller can deliver as much as 500 watts per wheel. A feedback encoder allows the RPC to keep track of speed and position. Torque load is also monitored.

Other Board-1 functions include gripper-motor control, and control of the sonar ranging system. Based on the Texas Instruments and Polaroid sonar systems, the ranging system can be used for collision avoidance, navigation, and security. Board 1 also contains several miscellaneous systems, including the battery charger, the DC-DC converter, and a beeper alarm. The environmental sensing systems (temperature, light, and sound detectors) are also on Board 1. Last, the collision detector outputs are processed on that board.

The robot differs from many projects presented in **Radio-Electronics** in that it is an evolving project. Many of its circuits are still in the design or testing phase. As a result, some of the final details may differ from those presented here. The other two control boards fall into the still-being-designed category. As of now, Board 2 will contain the speech synthesis and recognition hardware, and that Board 3 will house the RF data link.

Software

Complementing the hardware is a flexible programming environment that allows the programmer to choose his favorite and most productive language. The RPC may be programmed in two different ways.

One approach involves use of either of the onboard languages: BASIC and FORTH. Each language is a combination operating system, development system, and high-level language. Each includes debugging support, inherent ROMability, and access to mass storage, and each supports interrupt programming, integrated procedures, and multitasking. An onboard EPROM programmer allows software to be written, tested, and burned into EPROM for dedicated use.

The other approach makes use of the RPC's IBM-*PC* compatibility. The RPC boots most operating systems designed for the *PC*, thereby allowing the programmer to choose his favorite language. Assembler, Fortran, Pascal, C, BASIC, Compiled BASIC, and many others are all available. Programs in those languages can also be burned into ROM, if the compiler used generates ROMable code. Program code can also remain stored in battery backed-up static RAM for power-on execution. In addition, programs and data can also be stored on floppy disks. The RPC can accommodate any mixture of as many as four 3.5- and 5.25-inch floppy-disk drives.

RCL

Although the robot's software is not yet as extensive as we would like, modules have been written to test each of the robot's capabilities. The next step is an extremely sophisticated *R*obotic *C*ontrol

SOURCES

Can you imagine what a robot we could build with a staff of 250,000 (the entire readership of **Radio-Electronics**)? One key to the success of the **R-E** Robot is the collective development capability of that readership. In an effort to encourage the exchange of programs, sources of parts, hardware enhancements, and any other items of general interest, **Radio-Electronics**, Stock Drive Products, and Vesta Technology are each offering special support.

Radio-Electronics will open a special section of its new remote bulletin board system (RE-BBS) to builders of the **R-E** robot. You can reach the bulletin board by calling 516-293-2283.

Stock Drive Products (55 S. Denton Ave., New Hyde Park, NY 11040 516-328-0200) has agreed to supply a kit of parts for the drive sub-system, including two 10-inch pulleys and two 2-inch pulleys. Part number 2Z6-RL11862 is available for $32.00.

To simplify the mechanical aspects of building a robot, Vesta will sell, for a limited time, an aluminum chassis (resembling the one in Fig. 1) at cost, approximately $45. The fully-populated RPC will be available for $294, including 16K of RAM and the FORTH operating system. The Board-1 PC board is available as a bare board for $41, or fully assembled for $289. All source code for testing the robot and implementing RCL is available on a 5.25-inch disk for $2.00. All Vesta products are covered by a 15-day return policy. MasterCard or Visa accepted; no purchase orders or terms available. Please add $8.00 for shipping and handling for the computer board. Vesta Technology, Inc., 7100 W. 44th Avenue, Suite 101, Wheatridge, CO 80033, 303-422-8088.

Additional sources for various parts and sub-systems will be listed in future installments of this article. **R-E**

TABLE 1— SPECIFICATIONS COMPARISON

	RB-5X Robot	HERO 2000	R-E Robot
Dimensions	13" diameter × 23" high	16.5" wide × 22.5" long × 32.4" high	19" long × 18" wide × 20" high
Weight	24 pounds with arm	78 pounds with arm	55 pounds with arm
Speed	.23 mph	1 mph	5 mph
Arm	4 axes and gripper	4 axes and gripper	1 axis and gripper
	12 oz payload	16 oz payload	10 pound payload
	1"/second	6"/second tactile feedback	8"/second
Language	Tiny BASIC	Interpreted BASIC with specialized robotic commands	Mutlitasking FORTH with user alteraable Robot Control Language (RCL) overlay
Subsystems	Two RS-232 ports	Two RS-232 ports	Two RS-232 ports
	8 I/O lines	Cassette I/O	Disk drives (4)
	Sonar system	Sonar system	Sonar system
	8 perimeter bumper panels	Environmental sensing	Environmental sensing
	Speech synthesis	Speech synthesis	Speech synthesis
		Motor speed control Real-time clock	Motor speed control Real-time clock
		Keyboard/LCD	Collection sensor
Sleep Mode	No	Up to 6 days	Months
Battery	6 VDC 90 WH Sealed	12 VDC 288 WH Sealed	2 × 12 VDC 480 WH
Operator	Terminal	Teach pendant	Terminal
Interface		Keyboard with special function keys and keypad LCD on robot	Direct or remote connection
Remote Control	None	RF link, 100′ range built in to teach pendant	RF link attaches to user supplied RF transceivers
Microprocessor	8073, 4MHz	8088, 5MHz	80188, 8MHz
Mass Storage	2K ROM	Cassette tape, disk optional in future	Disk, optional
Memory	8K/16K RAM 2K ROM	24K/576K RAM 64K ROM	16K/768K 48K ROM
Bus	None	Proprietary 12 slot back plane, based on S-100	Modified IBM "PC" bus using flex cable and simple "ROBUS" expansion bus
Wheels	2" casters 4" solid wheels	3" casters 6" solid wheels	5" casters 10.5" pneumatic tires.
Cost (Basic unit)	$2,500 assembled	$2,500 kit	$850 components

Language (RCL). The inclusion of RCL on-board is possible only because of the power of the RPC.

The onboard RCL puts our robot a step ahead of almost all other home robots. Most robots are controlled with obscure software commands. A typical motion function could be programmed as follows:

```
OUT (1,1):REM Turn on drive motors
DELAY 1000:REM For one second
OUT (1,0):REM Then turn off motors
```

RCL allows the operator to program the same function as:

```
10 FEET FORWARD
```

Choosing a language in which to implement RCL was not an easy task. Because RCL was to be interpreted, we had to implement it in a language that executes quickly. To ease development and to allow people to customize RCL for their own purposes, it had to be written in a high-level language. We also wanted to minimize the cost of the hardware required for developing the RCL interpreter.

After considering BASIC, C, FORTH, and Pascal, we decided that FORTH met our requirements best. It runs much faster than interpreted BASIC, but it allows interactive program development, testing, and debugging. In addition, that language promotes the writing of modular, structured programs (as do Pascal and C), but it does not require a disk-based development system.

Another benefit is that FORTH is extensible, which means that the code we write becomes a part of FORTH. Because most

continued on page 94

R-E ROBOT

continued from page 56

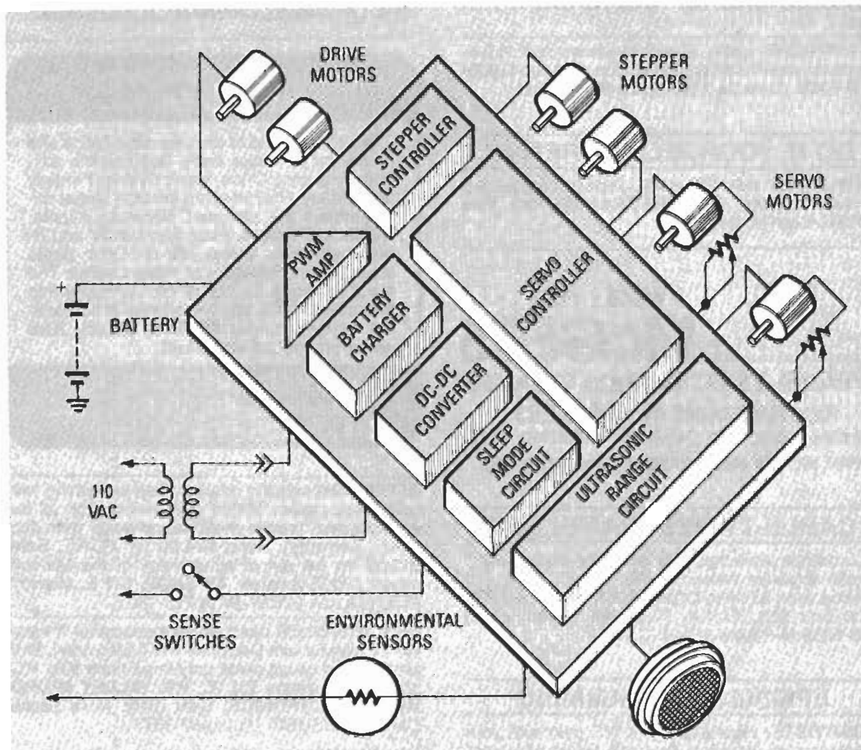

FIG. 2—MOST OF THE BASIC CONTROL CIRCUITS are located on a single custom board, designated Board 1. That board is shown here in block-diagram form.

of FORTH is written in FORTH, the internal routines are available to the programmer, and we can use parts of the FORTH interpreter in the RCL interpreter. The end result is an interactive RCL that can be modified by the user, and that can be used without a disk drive. We will be looking at RCL in depth in a future installment.

Operator interface

Most applications of robots are limited not by the ability of the hardware to perform a given task, but by the amount of time it takes to "teach" the robot. RCL significantly improves programming productivity.

Several methods of operator interface were considered. One was the "teach pendant" approach wherein motion sequences are learned and stored for recall and execution at a later time. However, after both utility and expense were considered, all other methods of interface were abandoned in favor of using a serial terminal. The reason is that almost everyone who considers building a robot has a terminal or a personal computer that can emulate a terminal. And if a more sophisticated method of control is desired, the required modifications are easy to perform. However, modifications of that sort are left to the ingenuity of the reader.

It is important to be able to operate the robot from a remote location. What we need is a way to transport our commands from the control terminal to the robot. That is done using an RF link. The link interfaces to the terminal via the terminal's serial port.

Table 1 shows how our robot stacks up against several of the leaders in the personal-robot market: the *RB-5X* from RB Robots (14618 W. Sixth St., Golden, CO 80401) and the Heath (Benton Harbor, MI 49022) *HERO 2000*. When you compare the capabilities of those robots to ours, we think you'll find that our inexpensive, build-it-yourself robot more than holds its own against the competition.

That concludes our overview of the **R-E** robot. In the coming months we will analyze each of the robot's subsystems in detail and show you how you can adapt our design to your applications.

As part of the design process, a special section of RE-BBS, **Radio-Electronics'** new computer bulletin-board service, will be dedicated for use by robot builders. We invite readers who devise interesting applications, programs, and experiments, or who discover sources of parts, or who have questions, answers, or any other information of general interest to share that information with others by posting it on the bulletin board. In addition, the author and the editors will be posting circuit modifications, design updates, and supplier information there for your convenience. By sharing information in that way, we hope to develop the kind of personal robot that has long been promised but never produced. **R-E**