Colne Robotics Armdroid
The Small-Systems Robot

Steven W. Leininger
5402 Summit Ridge Trail
Arlington, TX 76017

If you think you've explored all the possible hardware options for your small-computer system and are looking for some excitement, you might be interested in Armdroid, a new computer-controlled robot arm. The bright orange mechanical arm is available from Colne Robotics in kit or assembled form, complete with power supply and interface electronics. The kit form, besides being less expensive, "enables the person assembling the device to understand the principles of the robot," according to the manufacturer. The robot can be used for a variety of experimental and educational applications. It has 6 degrees of motion and a lift capacity of 10 ounces. I received both a kit and an assembled Armdroid for my evaluation, along with a "preliminary" manual.

Mechanical Description
The Armdroid has five major mechanical components: the base, the shoulder, the upper arm, the forearm, and the wrist and hand assembly. Each section is connected to its neighbor by a pivoting or rotating joint. The stationary base sits on the tabletop and provides support for the rest of the arm. The base, which also serves as the enclosure for the stepper-motor-drive electronics, contains the motor which rotates the arm about a vertical axis through the base.

At a Glance

<table>
<thead>
<tr>
<th>Name</th>
<th>Armdroid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use</td>
<td>Robotic arm</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Colne Robotics</td>
</tr>
<tr>
<td></td>
<td>207 NE 33rd St.</td>
</tr>
<tr>
<td></td>
<td>Fort Lauderdale, FL 33334</td>
</tr>
<tr>
<td>Dimensions</td>
<td>At shoulder: 18 by 18 by 20 cm (7 by 7 by 11.5 in)</td>
</tr>
<tr>
<td></td>
<td>Shoulder pivot height: 25 cm (10 in)</td>
</tr>
<tr>
<td></td>
<td>Arm length at maximum extension from shoulder pivot to finger tip: 48 cm (19 in)</td>
</tr>
<tr>
<td>Price</td>
<td>Kit: $595</td>
</tr>
<tr>
<td></td>
<td>Assembled: $695</td>
</tr>
<tr>
<td>Features</td>
<td>6 degrees of motion; menu-driven control software; 10-ounce load capacity</td>
</tr>
<tr>
<td>Additional Hardware Needed</td>
<td>TRS-80 Model I Level II (other microcomputers will be supported in the future)</td>
</tr>
<tr>
<td>Additional Software Needed</td>
<td>Learn, an interactive menu-driven control program (included)</td>
</tr>
<tr>
<td>Hardware Option</td>
<td>Zero-position sense switches</td>
</tr>
<tr>
<td>Documentation</td>
<td>Construction and Operation Manual, 87 pages</td>
</tr>
<tr>
<td>Audience</td>
<td>Experimenters, students, and professionals interested in robotics</td>
</tr>
</tbody>
</table>

About the Author
Steven W. Leininger was the design engineer for the original Radio Shack TRS-80 Model I microcomputer. He is now an independent computer consultant.
The shoulder rotates on the main bearing, a fairly heavy-duty ball-bearing assembly at the top of the base. Five stepper motors and associated reduction gears and drive belts are mounted on the shoulder and provide motion control to the arm, wrist, and hand.

The upper arm connects to the shoulder with a horizontal pivot and is rotated on that pivot by one of the stepper motors in the shoulder. If you move the upper arm vertically, the hand is raised and brought closer to the base. Cable-driving gears transmit motion to the forearm and the hand and wrist assembly; these are mounted in the shoulder end of the upper arm.

The forearm fastens to the upper arm with a horizontal pivot and is rotated about that point with one of the motors in the shoulder. The primary response to pivoting the forearm is the raising or lowering of the hand with respect to the tabletop.

The hand and wrist assembly attaches to the end of the forearm with a combination horizontal pivot and bevel gear assembly. The operator uses two motors in the shoulder to either rotate the hand about the pivot (an upward-down motion) or twist the hand about its axis. The remaining motor in the shoulder opens and closes the hand's three rubber-tipped metal fingers.

You can move any section independently without affecting the orientation of the other sections because of the Armadroid's parallelogram-type construction. This independence of control permits the angle of the hand to remain constant with respect to the workbench while the rest of the arm is manipulated to position the hand in the desired location.

Interface Electronics

The Armadroid I tested came with an I/O (input/output) adapter for the Radio Shack TRS-80 Model I. This adapter, a nolatched parallel port, plugs into the expansion port on the TRS-80. A cable from the adapter plugs into the base of the Armadroid.

Colne Robotics has mounted two printed-circuit cards within the base of the Armadroid: the interface board and the motor-drive board. The interface board accepts signals from the TRS-80, conditions them, and converts them to pulses of the duration and shape suitable for controlling the arm's motors. The motor-drive board amplifies the signals to provide the voltage and current levels required to drive the motors' coils.

You can set the Armadroid's internal electronics for external computer control or operation via manual switches by making the selection on the two printed-circuit boards inside the Armadroid's base.

Building the Kit

Being a disciple of Erector Set and Heathkit, I had no fears about venturing out into the frontiers of robot kit building. To get a feel for the scope of the project, I laid all the parts out and familiarized myself with the construction section of the manual.

The manual I received was a preliminary version. The entire mechanical assembly instructions were on just six pages! Undaunted, I forged ahead. About halfway through the first paragraph, I was instructed to glue magnets onto some of the gears. Apparently, the magnets are optional (at least they weren't included in the kit), but no mention was made of that fact. The system uses the magnets and their respective reed switches to sense the home position of the gears.

The instructions rambled on, sometimes with several steps in a sentence. The manual specified part numbers (usually) but didn't refer to the drawing numbers.

I knew the next part was going to be tricky because the instructions said that an assistant would be helpful. The task at hand was to assemble a dual-race ball-bearing assembly from scratch. Using refrigerated petroleum jelly
as per the instructions, I greased the bearing track and imbedded 24 ball bearings in the goo. After carefully inserting the base-support column into the bearing and turning the assembly upright, I attempted to repeat the job on the upper bearing track.

Darn. While tightening the adjusting ring, three balls hopped out of the lower bearing and huddled in a mound of petroleum jelly. Back to the beginning; twice more the same thing happened. Arrghh! Finally, success! But wait, why was the shoulder pan rubbing on the shoulder-drive gear? And, wasn’t that ball-bearing assembly just a little bit off parallel? At this point, I decided to cheat and look at the factory-assembled Armadroid. It appeared that the bearing-support column was too short. I described my problem to the gentlemen at Colne Robotics over the phone and was told that I probably had the bearing ring—an almost but not quite symmetrical part—on upside down.

I tried it again: I disassembled the bearing, inverted the bearing ring, and carefully placed the steel balls in the petroleum-jelly-coated track (I’m pretty good at this by now). Continuing as before, I installed the adjusting ring and beheld a smoothly operating shoulder bearing.

The instructions continued: put this motor here, put these gears there, and see the drawing. Well, I looked at the drawing. (The drawings are good up to a point, but they lack fine detail or close-ups in some areas.) I cheated a couple more times by looking at the assembled arm to verify my understanding of the drawings and text.

Assembly continued on the upper arm and forearm. The wrist posed no major problems. Then disaster struck! The fingers are held together with a large number of “circlips” (split rings that fit around the outside of a shaft). The circlips allow you to slide a rod through a hole, then prevent the rod from sliding back again. A special pair of circlip pliers is an absolute necessity to proceed beyond this point. I tried to make do with what I had (needle-nose pliers, screwdrivers, etc.) and realized I definitely needed the proper tools. It would have been nice if the appropriate pliers came in the kit or were at least available as an option.

The final assembly of the hand progressed easily after I purchased the circlip pliers. The instructions said to connect the arm assembly to the shoulder and base assembly. The cable threading came next. In the helpful hints section, the instructions said that this operation is greatly simplified by threading the arm before attaching it to the shoulder. So I started over again.

The actual cable threading progressed well, except for a clearance problem on one of the wrist cables. After checking the preassembled arm, I decided that cable clearance in the wrist is an assembly problem that Colne Robotics had experienced and corrected but had not updated in the manual. Ten minutes later, the offending cable had been restrung and worked smoothly.

The two printed-circuit boards went together just about as well as one would expect. No part numbers or reference designators were silk-screened on the boards, so I had to rely on the drawings in the manual for parts placement. Mounting the interface and motor-driver printed-circuit boards into the base of the Armadroid and connecting the stepper-motor wires to the driver board completed the assembly operation.

Using the Armadroid

A machine-language cassette for the TRS-80 Model I Level II microcomputer comes with the Armadroid. The menu-driven program, named Learn, allows you to familiarize yourself with the operation of the robot arm and to create, modify, and save motion sequences.

The manual suggests reading through the software description quickly and proceeding to the “Introductory Demonstration Sequence” section, which tells you to load Learn and enter the learn mode by typing an ‘L’.
**INTRODUCING PKASO™**

The master printer interface at a very low cost

For the first time ever a truly affordable Apple interface offers all the most sophisticated text and graphics capabilities on Epson®, Okidata®, Centronics®, and IDS® printers. With the easy to use PKASO interface you simply slip it into your Apple Computer, attach the cable to your printer, and enjoy all these features:

- Broader range of text printing using your software
- High resolution graphics with up to 40 creative options
- LoRes and HighTone graphics in 16 levels of grey
- SuperRes plotting with up to 2160 × 960 points per page
- User created or software defined characters and symbols
- Full text and graphics dump of absolutely any screen image.

At Interactive Structures we've built our reputation on innovation, quality and service, and we're doing it again with the new PKASO series. The PKASO Interface will bring out the best in your Apple Computer, your data printer and your program. It will perform with all popular languages such as BASIC and ASSEMBLER. It will print both text and graphics with PASCAL. And it's the first and only Apple Interface to offer all this plus support for the Apple Z-80 CP/M System and for full Apple // operation.

Don't settle for less. And don't pay more. Call us now for the name of the PKASO dealer near you. Circle 210 on inquiry card.

---

**Photo 5:** The hand and wrist assembly has three fingers. The fingers are opened and closed in unison under program control. The wrist allows both rotation and up-and-down motion of the hand.

This mode lets you manually operate the robot while programming it to follow the same motions automatically.

The program asks you if you want to start again, continue from the present position, or exit the program. Type “S” to clear the memory and free the arm. The arm is free when no torque is applied to the stepper motors. This allows you to initialize the Armadillo's position by hand using the large gears in the shoulder. When you are satisfied with the starting position, press the space bar. The program applies torque to the arm, effectively stiffening and locking the arm in place.

You can now move the arm using the Q, W, E, R, T, Y, and 1 through 6 keys to manually control the movement of the different parts of the arm. If you're like me, it will take a couple of tries to predictively move the arm, rotate the wrist, and open and close the hand using manual control. Type a “0” to get out of the learn mode.

Now the miracle of life! Press “G” for go, and the Armadillo takes the shortest path to your initial starting position. The program then asks “O” (once) or “F” (forever). Forever seems like a long time for something you haven't tried yet, so type “O”.

Wow! The arm is doing just what you taught it to do! And without the long pauses for head scratching and note taking! You are returned to the menu.

To look at the sequence of commands that were sent to the stepper motors, type “D” for display. A table appears on the screen showing the stepper increment values stored in memory.

To extend the sequence of movements, simply reenter the learn mode, and type “C” for continue. You can add additional motions by using the manual-control keys. Once again, you must type “0” to return to the menu mode.

After testing the new sequence, you may decide that some of the motions need to be fine tuned. This can be done using the edit mode.
Three cassette-tape commands allow you to save your Armdroid sequence for a rainy day. “W” (write) saves the sequence in memory on the tape, “R” (read) retrieves the sequence from tape, and “C” (check) verifies that the data on the tape is the same as that in memory.

Colne Robotics has graciously included the source listing for the Armdroid control software in the manual. The Z80 assembly-language source is well documented and should prove to be a valuable learning tool for the student of robot technology. The source code is also useful to those who wish to modify the control software for a specific application.

I understand that Colne Robotics is developing similar software for other microcomputers, such as the Commodore PET, the Apple II, and the Sinclair ZX81. Watch their advertisements for further details.

Documentation

The 87-page manual is broken down into four sections. The introduction section is nine pages long and strays from the purpose of an experimental robot arm. Discussions on the economic and social impact of industrial robots, complete with tables and formulas, seem more like padding than useful information.

The second section deals with the mechanical assembly of the Armdroid. As noted above, some deficiencies and inaccuracies in the instructions exist. A hand-holding, step-by-step approach would benefit the novice builder.

The next section details the electronics of the Armdroid. This section was not too bad, but again a step-by-step approach would be helpful.

The final section describes the software package included with the arm. This chapter of the manual was the easiest to use, due in part to the quality of the Learn program itself. And I applaud the inclusion of the program listing as an aid to understanding the ins and outs of microprocessor-controlled robotics.

It should be noted that my review is based on a “preliminary” manual for the Armdroid. I have been assured that the manual will be revised to eliminate some of the limitations that I have noted above.

Conclusions

• The Armdroid is a low-cost manipulator with good dexterity and maneuverability.
• The software delivered with the arm is easy to use and serves as a powerful tool in understanding robot operation.
• The Armdroid kit is not for the inexperienced builder, unless the manual is improved.
• I feel I have learned a lot about the mechanics, electronics, and software of robots, thanks to the people at Colne Robotics.

---

Apple Logo

by Harold Ableson

The name Logo describes not only the evolving family of computer-language detailed in this book, but also a philosophy of education that makes full and innovative use of the teaching potential of modern computers. Apple Logo presents the Apple II user with a complete guide to the applications of this unique system and also includes a description of TI Logo for users of the Texas Instruments 99/4 computer.

The designers' vision of an unlimited educational tool becomes reality for the Apple II user who begins to work with this procedural language. Logo enables even young children to control the computer in self-directed ways (rather than merely responding to it), yet it also offers sophisticated users a general programming system of considerable power.

---

Apple Logo actually teaches programming techniques through “Turtle Geometry”—fascinating exercises involving both Logo programming and geometric concepts. Later chapters illustrate more advanced projects such as an “INSTANT” program for preschool children and the famous “DOCTOR” program with its simulated “psychotherapist.”

ISBN 0-07-00425-0
240 pages
Softcover, spiral-bound
$14.95

Call Toll-Free 800/258-5420

BYTE Books 70 Main Street
Peterborough, N.H. 03458

B5