BUILD YOUR OWN ROBOT!
Costs under £200 complete!

- Computer controlled - software details inside!
- Remote controllable - interface details inside!
- Program competition - win £100!

PLUS
- EQUALISER CIRCUITS TO BUILD
- SPECIAL TALKING CLOCK PROJECT!
- WIRE-LESS EAR STRETCHER!

SPECIAL ROBOTICS ISSUE!
ROBOT ARM

Stop what you're doing and pay reverent attention — this is a historic moment. ETI proudly presents the Armdroid — Britain's first serious robot for home construction. It can be operated manually or taught movement sequences under computer control, and is really only limited in its applications by your imagination. System concept by Ron Harris. Realisation and development by Agit Channe, Nick Ouroussof and Andrew Lennard.

Welcome to the Robot Age. With the publication of this project ETI shepherds in a new era in our hobby. Robotics is the logical extension of electronics and modern manufacturing methods. We already have all the necessary technology to produce viable robots: cheap memory; cheap processing; mass produced computers; comprehensive I/O electronics; accurate and versatile metalwork machinery — automated, of course.

Any civilised country wishing to survive as an economic power in the 1990s and beyond will have to have a large and operative robot population in its industries. Read the article elsewhere in this issue for an assessment of Britain's chances, based on today's figures.

Know The Robot

One of the greatest obstacles to industrial robots is the lack of freely available information on the subject for the engineer and technician who will be expected to use and control the dreaded 'mechanical men'.

The Armdroid is the first in a line of ETI robotic projects, all of which can be built and used by anyone who can solder! The arm can lift loads in excess of any commercial equivalent we know of up to £1,000 in price.

As such it is designed to fulfill the needs of the small industrial user who is searching for a small programmable manipulative machine; the educational establishment interested in research and adaptation; and finally the hobbyist at home who just wants to build a good project.

We hope that it will stimulate interest in the field and serve to illustrate the accessibility of this new branch of technology. Although originally configured to run from a Tandy TRS-80 Model I home computer, the bus structure is such that it can be instantly set up to run from any other machine with this (standard) input.

In order to encourage this level of involvement we are offering a £100 prize to the author(s) of the most ingenious piece of software to run the Armdroid — on any machine except the Tandy! (See end of article for details.)

Establishments who do not yet own a computer need not despair, as a control box is available to operate the arm without recourse to a processor. The circuit details are given herein.

Capabilities

Built along the lines of the prototype described here, and with a Tandy computer, the Armdroid can be used under direct keyboard control or 'taught' a sequence of actions, which it will then repeat either once or forever (in theory!) to a very high degree of accuracy.

It is a 'continuous path' robot, which means that more than one motor can be operating at any given time, making possible very complex motions. Many commercial machines are what is termed 'point-to-point', in that each motor/driver operates in sequence, moving the robot from one point to another in a series of steps.

The 'claw' or 'grabber' on the Armdroid is of a totally new design and is the subject of patent applications.
Software
A program tape containing the 'tutor' program, to enable the Amroid to be programmed for repetitive actions, is available for the TRS-80. At a later date, routines for the other major machines may well become available if the demand warrants it.

The interface port specification is given in Fig. 2; to enable programmers to write routines to drive the machinery in the meantime.

A block diagram of the required program is also given, though not a full flowchart. A full (machine code) listing of the TRS-80 tutor program is available from our Charing Cross Road offices, in exchange for an SAE for us to send it in! As it runs to some 700 lines we thought it inappropriate to publish it all here. However, we've included a hex dump for those people who want to be able to load and use the program without necessarily understanding it.

Anyone rewriting the software for other machines should take note of the following points. First, the TRS-80 uses the Z-80 microprocessor so any machine with a different micro will require a complete rewrite of the machine code. Line 46AE contains a jump address which in the published listing (Fig. 3) simply points back to the start of the program. If you want the facility to quit the tutor program and jump back into the system monitor, this is the address to change (what you change it to naturally depends on your machine). Lines 4921-4925 contain the port address, which should also be changed to suit.

The next few lines contain calls to a delay routine; this sets the torque of the motors by controlling the clock delay. Two delay routines are provided in the program, DELS (46BD) which gives a delay of about 0.001 s and DELT (46CS) for a delay of about 0.01 s.

Manual Metalwork
In order to make construction of the Amroid possible for the home constructor, we have arranged for a kit of parts to be supplied, somewhat in the manner of a Meccano set! All the drilling and cutting is done for you; all you have to do is slot it all together.

And to make that easy, a comprehensive assembly manual is to be supplied free with each kit.

The arm is also to be made available in fully assembled form, albeit at a higher price, for those users who wish only to experiment with the finished item.

Because of the existence of the excellent metalwork manual, we are not going to deal with the building of that side of the project here at all. It would simply duplicate information which is being supplied anyway and we do not have the space to do it thoroughly.

Have a good look at the detailed photographs within the article if you’re in any doubt as to its assembly. Follow the manual through carefully and no problems should occur.

Construction
Anyone who has ever built an ETI project before — or even one from the other, lesser, electronics magazines will be quite capable of wiring up the interface and PSU required. Follow the basic rules — and check everything at each stage before proceeding any further.

Build and test the PSU first, and make sure you obtain the correct voltages of 12 V and 5 V before connecting circuits to the PSU output. Assemble and test each motor drive circuit individually. It will be much simpler to debug each channel
HOW IT WORKS

The interface between the microcontroller and the arm is designed to provide a range of functions. The microcontroller is responsible for controlling the arm, while the interface translates the microcontroller's commands into the appropriate motor and sensor inputs.

The interface consists of several key components:

1. **Motor Control** - Each motor is driven by a dedicated control circuit that receives commands from the microcontroller.
2. **Sensor Inputs** - Various sensors (e.g., position, force, temperature) are connected to the interface to monitor the arm's status.
3. **Power Supply** - A stable power supply is necessary to ensure consistent operation of the arm.
4. **Communication** - A protocol (e.g., Ethernet, USB) is used to transmit data between the microcontroller and the interface.

The interface is designed to be modular, allowing for easy expansion or modification as the arm's requirements evolve.

The microcontroller sends commands to the interface, which then translates these commands into appropriate motor and sensor inputs. This process ensures smooth and efficient operation of the arm.

For example, when the arm is in the process of moving, the microcontroller sends a position command to the interface. The interface then interprets this command and sends the appropriate signals to the motors, allowing them to adjust their positions accordingly.

The interface also includes a communication module that facilitates data exchange between the microcontroller and other systems, such as a computer or a user interface.
separately since if you have all six in operation at any given moment, horrendously complex gyrations of the robot arm are possible and it will not always be easy to see exactly what the faults are or even in which channel they lie.

Note that the parts list and overlay for the drive board are a little peculiar, with some parts apparently labelled the same. This is because the drive circuit is repeated six times, but with a few exceptions; some parts appear six times, some three times and some only once. But it does make sense if you study it carefully in conjunction with the circuit diagram — honest!

Refer to the component overlays and circuit diagrams provided during construction at each stage. Do not simply 'knock the whole thing together' and then start checking! IT WILL FAIL. While the interface and/or control box is not particularly expensive, there is no point in throwing money away by merely destroying ICs wired in reverse.

The only setting up procedure involves PR1 — this component should be used to adjust the motor speed in manual mode so that the motors do not slip when stepping.

**In Use**

Normally we can give a pretty good indication to our readers as to which applications a project is best suited to — in this case, however, you will have to tell us! There will be such a diversity of use that your particular application is likely to be of great interest to other readers.

To this end we will publish — and pay for — applications reports from users of the Armadillo in future issues of ETI. For schools, colleges and so on this obviously represents a chance to recoup some of the cost. Contact the Editor for further details.

---

**Fig. 4 Circuit diagram of the interface board.** Although four outputs are available from each of the six latches, only the two labelled outputs (CCLK and CDIR) are used in this application. This particular section of the design is very versatile; for example, driving triacs from the latch outputs gives a computer-controlled disco lighting console.
ROBOT PROGRAM COMPETITION

FREE ENTRY!

Here's a chance for all you would-be robotists and programmers to earn yourselves £100!

In order to encourage the wider use of our first robot, ETI is offering a cash prize of one hundred pounds for the best program submitted which enables the Armdroid to be used with any of the following computers:

- Commodore PET — any model
- Tangerine MICRON
- Sharp MZ-80K
- Tandy TRS-80 Model III
- Superboard (expanded)
- Video Game
- Apple II
- NASCOM
- Acorn Atom

Any memory size may be used, but we would suggest that a minimum of 8K is accepted. (The routine takes nothing like this amount of space incidentally.)

Entries must include a full listing and tapes are ONLY acceptable for PET and Sharp. Any other undocumented entries supplied on tape will be disqualified.

Group entries are quite acceptable, but it must be made clear who is to receive the cheque in the event of success, be it school, club or individual.

Handwritten entries will be considered, although presentation will play a part in deciding the competition, so typed entries (and printed listings) are to be preferred.

Closing date is October 31st 1981 and the winners will be announced as soon as possible after that date.

Send to: Armdroid Competition, Electronics Today International, 143 Charing Cross Road, London WC2H 0EE.

RULES

1. The entry must allow the Armdroid to be both keyboard-controlled and to execute a series of actions under program control.
2. Programs for the TRS-80 Model III are not eligible for consideration, although they may be submitted to ETI and Computing Today as articles in the normal manner.
3. Entries arriving with a postmark of November 1st or later will not be eligible for judging.
4. All entries become the property of Modums Ltd upon submission and no correspondence concerning the competition will be entered into. Any entries published in the magazine will be paid for at the usual rates.
5. The judges' decision shall be considered final by the contestants.
Fig. 5 This software schematic gives a guide to the sort of facilities needed in a computer control program.

Fig. 6 Coil stepping sequence for the stepper motors used in the Armroid. Compare this table with the waveforms in Fig. 7 to see how the control signals generate movement.

BUYLINES

Colne Robotics can supply either a complete kit of parts or assembled units for the Armroid.

Armroid — Kit (including Manual): £199
Armroid — Assembled (including Manual): £270

Interface/Driver/Power Supply and cassettes of software:

Kit £45
Assembled £55

Manual Control Box:

Kit £20
Assembled £25

All prices are exclusive of VAT (15%) and postage and packing. Add £2.50 p & p for the Armroid (either kit or assembled), and £1.50 for all other items.

Colne Robotics Co Ltd, 1 Station Road, Twickenham, Middlesex, TW1 4LL. Telephone: 01-892 7044. Telex: 8814086 GCIC.

Fig. 7 Waveforms required to step the Armroid motors correctly. These can be generated using fairly simple circuitry.
Fig. 8 The motor driver circuit diagram. There are six of these driver circuits, one for each motor, but only three CD4551s are required (half an IC for each channel) and one 555 (which provides the manual clock pulses for all six channels).

Fig. 9 Circuit diagram of the power supply for this project, which is capable of driving the boards and all six motors.

**HOW IT WORKS**

**THE ARM DRIVERS**

The arm motor driver logic has been designed so that it can be driven from a manual control box, or from the output of the computer interface circuitry. If the arm is to be controlled only from the CPU interface, then a large portion of the driver circuitry can be ignored.

The four outputs from the CPU interface logic can be connected to the four inputs of IC4 and the processor must then produce its own drive signals as shown in Fig. 7. This will also enable the motors to be halved by the processor. If the above is carried out, then ICs 1, 2, and 3 will be redundant in each arm motor drive section.

The circuit described has a manual override so that if for some reason the arm is doing something that you dislike, then it is possible to stop it using the manual controls.

The six motor driver stages need two power supplies to function; 12 V at about 3A and 5 V at 150 mA.

At the front end of the circuitry is a CMOS switch (IC1a). This is used to select the clock signal and the direction signal which are to be fed into the system. When the processor is controlling the motor driver, the CD4551 and CCLK signals will be selected and placed on the DDIR and DCLK lines. In manual mode, clock pulses are fed into the system from IC6 (a simple 555 astable) via SW1 and SW2 controls the motor direction. To move a joint one way press SW3; to move it the other way press SW3 and SW2 simultaneously. Pull-down resistors R1, 2, 3 are needed to prevent the inputs of IC1a floating when the switches are open (CMOS doesn’t like this).

The CLKB signal is fed to the clock input of IC2b, a D-type flip-flop, so the data on the D input is latched on the rising edge of CLKB. The Q output is coupled back to the input so that each latched input is the inverse of the previous one; thus the Q output is a signal with half the frequency of CLKB. The waveform of Q,D in Fig. 7. The waveform Q,D is simply the inverse of Q,D and is taken from the Q output of IC2b. A capacitor/resistor network (C5/R6) resets the flip-flops on power-up.

The Q,D and Q,D clock pulses are derived from CLKB, the Q output of IC2b and the R output of IC2a. The Q output of IC2a is selected when the selected direction input DDIR is latched by the falling edge of CLKB, since the clock signal is inverted by IC3a. CLKB is XORed with the Q,D clock signal in IC3b so when Q,D is identical to CLKB point B will be low otherwise it will be high. The output of IC3b in Fig. 7 is XORed with the Q output of IC2a to produce Q,D and Q,D is obtained by inverting Q,D in IC3d, an XOR gate with its input tied high.

The four waveforms Q,D, Q,D are then fed into IC4, a level shifter. Here the 5 V inputs are converted into 12 V outputs, and then fed into IC5. This is a high current quad VFET which provides the high current outputs for the motor coils. The driving current level for the motor coils is about 300 mA at 12 V.

Next month we conclude the Arm project with the Parts List and component overlays.
ROBOT ARM

Part 2 of the Armdroid project gives the complete constructional details of the electronics. System concept by Ron Harris. Realisation and development by Ajit Channe, Nick Ouroussoff and Andrew Lennard.

This month we give the Parts Lists and overlays for the Armdroid, plus a few things we have to point out. A large number of supply decoupling capacitors are required on the interface board — for clarity these were not shown on last month’s circuit diagram, but they appear as C2-14 in Fig. 3. Figure 3 also includes two extra resistors which experience has shown to be necessary.

On the motor driver board, there are four spare pads to the left of each IC4. These may be used to directly connect control signals from the outputs of the latches on the interface board, if direct computer control of the stepper coils is desired as mentioned last month.

Finally, Colne Robotics are continuing development work on the gripper, and invite suggestions for alternative designs. If you think you’ve got a good idea (some possibilities are mentioned in Robotics Today this month) get in touch with Colne at the address given in the column last month. If they like your proposal they’re prepared to do the development and engineering work and pay the inventor a royalty. Over to you!

---

Fig. 1 Motor wiring diagram. Note that both the centre terminals are connected to 12 V.

---

Fig. 2 Overlay for the PSU. Note that IC1 requires a heatsink.

---

PARTS LIST

<table>
<thead>
<tr>
<th>Parts List</th>
<th>BRT1</th>
<th>T51</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER SUPPLY</td>
<td>300 V, 6 A bridge rectifier</td>
<td>240 V AC transient suppressor</td>
</tr>
<tr>
<td>Resistors (all 1/4 W, 5%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>4K7</td>
<td></td>
</tr>
<tr>
<td>T51</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacitors</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1,2</td>
<td>4700u, 25 V axial electrolytic</td>
</tr>
<tr>
<td>C3,4</td>
<td>1uH 35 V tantalum</td>
</tr>
<tr>
<td>C5</td>
<td>220n polyester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semiconductors</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1</td>
<td>78H12</td>
</tr>
<tr>
<td>IC2</td>
<td>7805</td>
</tr>
</tbody>
</table>

---

ETI OCTOBER 1981
ROBOT PROGRAM COMPETITION
FREE ENTRY!

Here's a chance for all you would-be robotists and programmers to earn yourselves £100!

In order to encourage the wider use of our first robot, ETI is offering a cash prize of one hundred pounds for the best program submitted which enables the Amdroid to be used with any of the following computers:

- Commodore PET — any model
- Tangerine MICRON
- Sharp M2-60K
- Tandy TRS-80 Model 111
- Superboard (expanded)
- Video Genie
- Apple/TT 2020
- NASCOM
- Acorn Atom

Any memory size may be used, but we would suggest that a minimum of 8K is accepted. (The routine takes nothing like this amount of space incidentally!)

Entries must include a full listing and tapes are ONLY acceptable for PET and Sharp. Any other undocumented entries supplied on tape will be disqualified.

Group entries are quite acceptable, but it must be made clear who is to receive the cheque in the event of success, be it school, club or individual.

Handwritten entries will be considered, although presentation will play a part in deciding the competition, so typed entries (and printed listings) are to be preferred.

Closing date is October 31st 1981 and the winners will be announced as soon as possible after that date.

Send to: Amdroid Competition, Electronics Today International, 145 Charing Cross Road, London WC2H 0EE.

RULES

1. The entry must allow the Amdroid to be both keyboard controlled and to execute a series of actions under program control.
2. Programs for the TRS-80 Model 1 and are not eligible for consideration, although they may be submitted to ETI and Computing Today as articles in the normal manner.
3. Entries arriving with a postmark of November 1st or later will not be eligible for judging.
4. All entries become the property of Electronics Ltd upon submission and no correspondence concerning the competition will be entertained. Any entries published in the magazine will be paid for at the usual rates.
5. The judges' decision shall be considered final by the contestants.
Fig. 3 Interface board overlay. The connection to the computer port will depend on your particular machine.
**PARTS LIST**

**MOTOR DRIVER**
(For one channel only — six sets required except where stated)
- Resistors (all 1/4 W, 5%)
  - R1: 10k (one off)
  - R2: 10k
  - R3: 39k (one off)
  - R4: 68k (one off)
  - R6: 100k

- Potentiometer
  - PR1: 100k miniature vertical preset (one off)

- Capacitors
  - C1: 10n ceramic (one off)
  - C2: 100n ceramic (one off)

**Semiconductors**
- IC1: CD4551 (three off)
- IC2: CD4013
- IC3: CD4070
- IC4: CD4017
- IC5: VQ1600CS
- IC6: 555 (one off)

**Miscellaneous**
- SW1: SPDT toggle (one off)
- SW2, 3: SPST push-button
- MTR: 12 V stepper motor
- 0.1" 16-way edge connector (one off); four-way PCB plug and socket connectors (three off); 25-way terminal block (one off)

---

Fig. 4 Overlay for the motor driver board. Points A, A, B, B, ..., F-F are linked by lengths of insulated wire; normally points G-G-G are also linked so that the manual override switch (SW1) controls the CMOS switches in all six channels. However, it is required to provide a separate override for each pair of channels, pads are provided at pins 9 of each IC1 for the extra pull-down resistors.