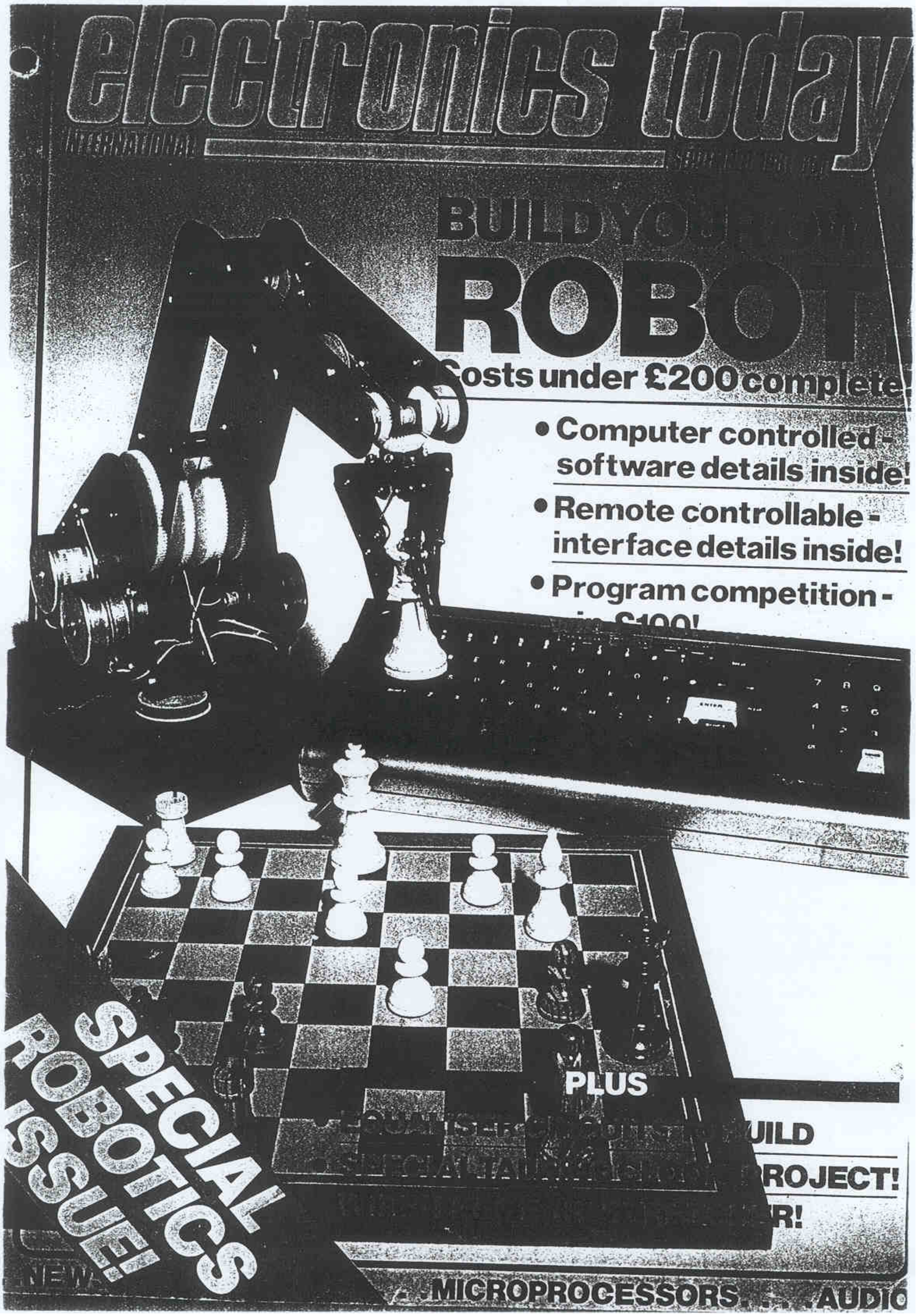
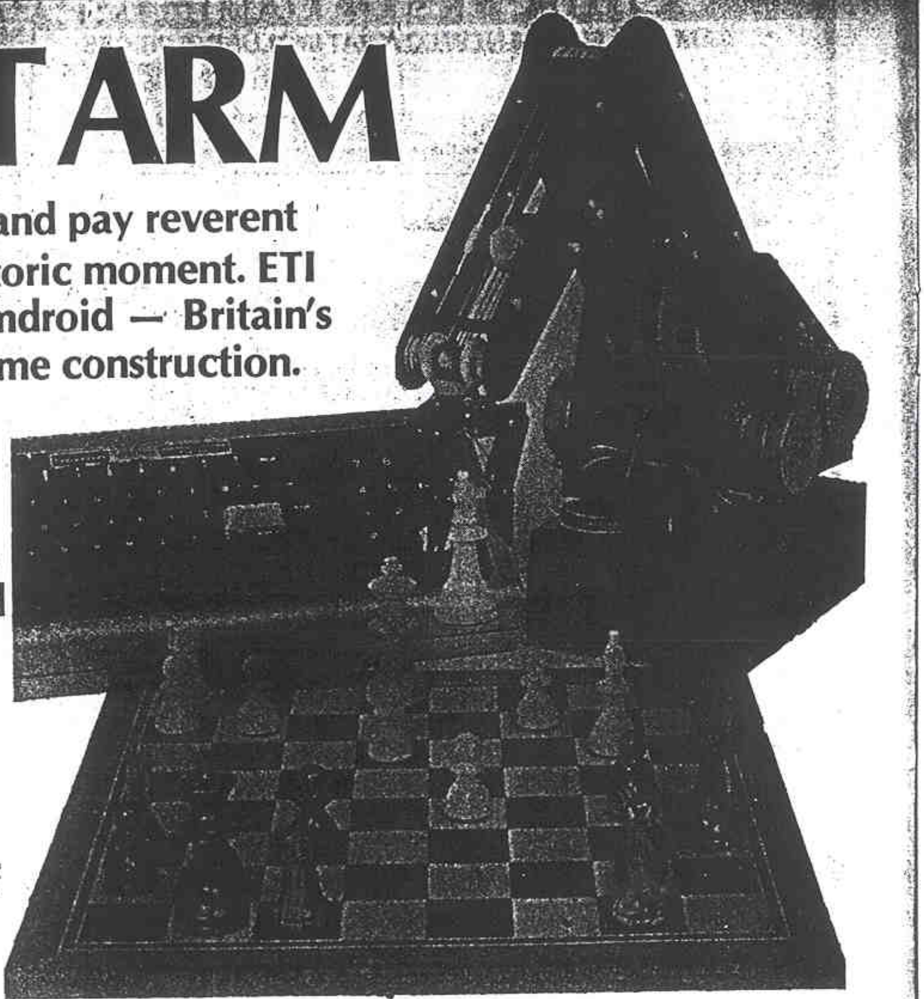

electronics today
INTERNATIONAL
BUILD YOU
ROBO
osts under £200 complete
• Computer controlled - software details inside!
• Remote controllable - interface details inside!
• Program competition -
PLUS
UILD
ROJECT!
R!
SPECIAL ROBOTICS ISSUE!
NEW
MICROPROCESSORS . . . AUDIO

ROBOT ARM

op what you're doing and pay reverent ttention — this is a historic moment. ETI roudly presents the Armdroid — Britain's irst 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.

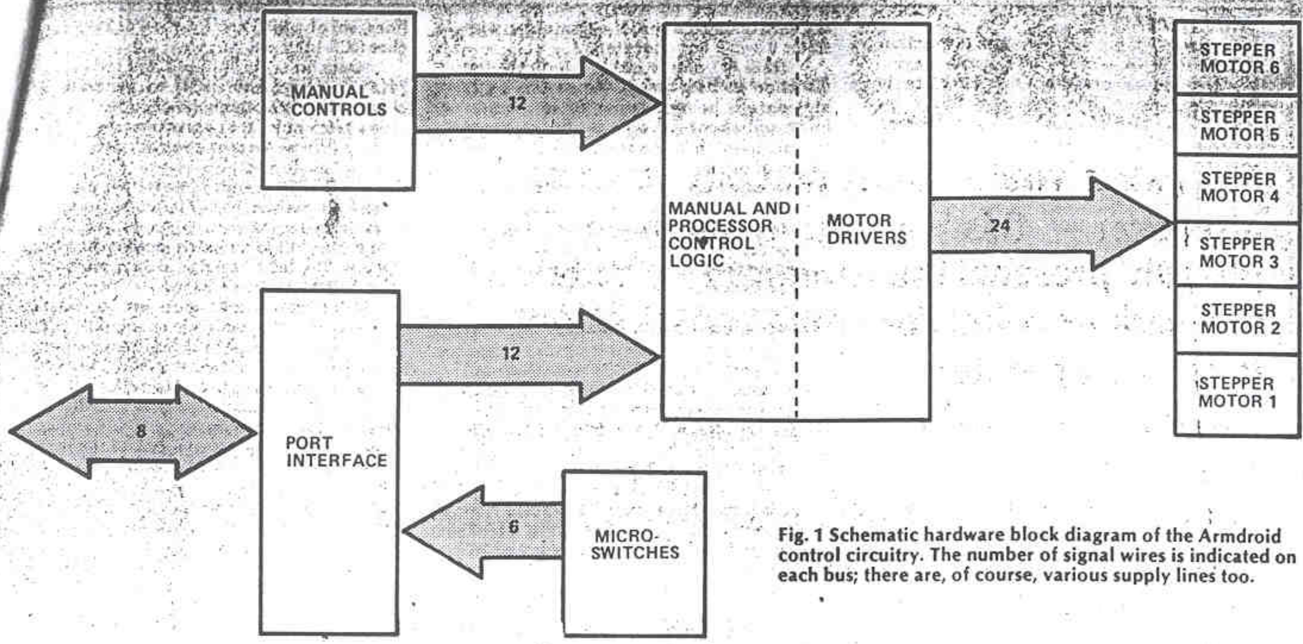


Fig. 1 Schematic hardware block diagram of the Armdroid control circuitry. The number of signal wires is indicated on each bus; there are, of course, various supply lines too.

OUTPUT FROM MICRO		INPUT TO MICRO
X2	D8	MS6
X1	D7	MS5
CCLK	D6	MS4
CDIR	D5	MS3
A3	D4	MS2
A2	D3	MS1
A1	D2	U1
$\overline{OUT}$	D1	IN

OUTPUT BITS
X1,X2: Unused (could be used with CCLK, CDIR for direct stepper drive)
CCLK: Clocks driver circuitry
CDIR: Gives motor direction
A1,A2,A3: Motor address
$\overline{OUT}$: Low indicates data out from microprocessor
INPUT BITS
MS1-MS6: Microswitch inputs (if used)
U1: For personal use
IN: High indicates data in to microprocessor

Fig. 2 Interface port specification. This will help readers who wish to write their own software routines.

Software

A program tape containing the 'tutor' program, to enable the Armdroid 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 though 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 4921-4926 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 (46C5) for a delay of about 0.01 s.

Manual Metalwork

In order to make construction of the Armdroid 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 de-bug each channel

TERFACE

...to enable the robot arm to function as broad a range of microprocessor ...ment as possible, the interface is ...ned around a standard eight bit, bi-...ctional port. This may be a latched or ...latched port; in a non-latched situation ...interface circuitry will normally be in an ...put-to-micro' configuration.

The port is configured as follows. The eight output lines are defined as four data bits (D8-D5), three address bits (D4-D2) and one bit (D1) which defines the direction of data travel on the port. Four data lines are provided for the user, who at a later stage may wish to directly control the stepper motor coils from the computer instead of via the stepper control logic.

When the motors are being operated via the stepper control logic, only two of the data lines are used. Data bit six is used as a clock bit to step the motor; the delay on this bit will relate to the speed of the step, and hence the speed and flow of movement. Data bit five is used to indicate the direction the motor will step in.

The address bits are used to channel the selected clock and direction bit to the appropriate motor output. The three address bits can define eight states; states 1-6 are used to select any of the six motors, with states 0 and 7 not allocated.

Data bit one is used to indicate the direction of data travel. When this bit is low, data is being transferred to the arm joints and when high, data can be read from the microswitches, if installed. On the transistion of bit one from high to low a pulse is generated which causes the data to be latched into the addressed output latch.

In the input mode the lines D8-D3 are used to read any of the six microswitches installed on the arm. These are in the form of reed switches and magnets, and indicate one specific position of each joint and hence enable the arm to be reset before any learning sequence takes place.

A spare input bit (D2) exists which can be buffered and used for some extra input sensor. For example, this spare input could be used to connect a 'home brew' transducer to the system.

The interface circuitry consists of 12 TTL components which decode and route the data out to the appropriate motor control/driver logic. Two 74LS125 ICs (IC1,2) buffer the data out to the decoder and latches. A 74LS138 (IC6) decodes the three input address bits to provide eight select lines, six of which are for the 74LS175 latches (IC7-12).

Data bit D1 is buffered and fed into a 74LS123 monostable (IC4) to generate a clock pulse. This causes the decoder to provide a latch pulse for approximately 500 ns to the addressed motor control latch. Data bit D1 is tied to a pull-up resistor (R1) so that this line is high, except when data is output from the processor. The 74LS125 buffers are enabled by the buffered output of bit 1 so data is only fed to the latch inputs when bit 1 is low. The bit 1 buffer is always enabled because its enable is tied low.

The microswitch inputs are buffered by a 74LS366 (IC5) which is enabled by the complemented output of bit 1, hence when bit 1 is high the 74LS366 is enabled. Thus for a large portion of the time this buffer will be enabled and the contents of the microswitches will be input to the micro. This allows users to operate the arm under bit interrupt control, allowing instant response to a microswitch change and avoiding the 'polling' of the microswitches. The six microswitch inputs are pulled up, hence the switches can be connected to the arm using only one return lead per switch and the arm chassis as ground.

Fig. 3 Hex dump of the TRS-80 Model I tutor program. The memory following this program is designated ARST, the ARm STorage area that holds the numerical data defining a learned sequence of moves.

```
4400  CD D6 46 21 6B 49 CD A7 28 CD C5 46 21 84 49 CD
4410  A7 28 CD 49 00 CD 33 00 CD 5D 49 FE 4C CA 51 44
4420  FE 52 CA 85 44 FE 57 CA 35 45 FE 43 CA D7 44 FE
4430  47 CA 8C 45 FE 44 CA 26 46 FE 42 CA 00 44 FE 4D
4440  CA 9F 46 FE 51 CA B1 46 21 DE 49 CD A7 28 C3 09
4450  44 21 E5 49 CD A7 28 CD 49 00 CD 33 00 CD 5D 49
4460  FE 53 20 05 CD D6 46 18 0C FE 43 20 E4 2A 8A 4A
4470  7D B4 CA 38 46 AF 32 8E 4A CD 53 47 47 78 B7 C2
4480  79 44 C3 09 44 21 C4 49 CD A7 28 CD 49 00 CD 5D
4490  49 AF CD 12 02 CD BD 46 CD 96 02 CD 35 02 47 CD
44A0  35 02 4F B0 CA 38 46 ED 43 8A 4A 21 AE 4A C5 1E
44B0  00 06 06 CD 35 02 77 83 5F 23 10 F7 C1 CD 35 02
44C0  BB 20 0B 0B 78 B1 20 E6 CD F8 01 C3 2C 45 21 75
44D0  4A CD A7 28 C3 09 44 ED 4B 8A 4A 78 B1 CA 38 46
44E0  21 C4 49 CD A7 28 CD 49 00 CD 5D 49 AF CD 12 02
44F0  CD 96 02 ED 4B 8A 4A CD 35 02 B8 20 D1 CD 35 02
4500  B9 20 CB B0 CA 38 46 21 AE 4A C5 1E 00 06 06 CD
4510  35 02 BE C2 CE 44 83 5F 23 10 F4 C1 CD 35 02 BB
4520  C2 CE 44 0B 78 B1 C2 0A 45 CD F8 01 21 81 4A CD
4530  A7 28 C3 09 44 ED 4B 8A 4A 78 B1 CA 38 46 21 C4
4540  49 CD A7 28 CD 49 00 CD 5D 49 AF CD 12 02 CD C5
4550  46 CD 87 02 CD C5 46 ED 4B 8A 4A 78 CD 64 02 79
4560  CD C5 46 CD 64 02 21 AE 4A C5 1E 00 06 06 7E CD
4570  BD 46 CD 64 02 CD DD 46 83 5F 23 10 F1 CD 64 02
4580  C1 0B 78 B1 20 E3 CD F8 01 C3 09 44 CD 5D 49 AF
4590  32 91 4A 21 B2 49 CD A7 28 CD 49 00 CD 33 00 CD
45A0  5D 49 FE 4F 28 09 FE 46 20 E2 3E 01 32 91 4A 3E
45B0  2E CD 33 00 CD C6 45 3A 91 4A B7 28 F2 21 61 4A
45C0  CD A7 28 C3 09 44 ED 4B 8A 4A 78 B1 28 19 21 AE
45D0  4A 11 9C 4A C5 01 06 00 ED B0 E5 CD F2 45 CD 65
45E0  49 E1 C1 FE 53 20 05 AF 32 91 4A C9 0B 78 B1 20
45F0  E6 C3 0E 00 06 06 11 AD 4A 21 A1 4A 7E FE 00 28
4600  0F FA 00 46 3E 03 12 35 18 08 3E 01 12 34 18 02
4610  AF 12 1B 2B 10 E6 3E 01 32 90 4A CD 89 48 0D C2
4620  F4 45 C9 C3 09 44 21 09 4A CD A7 28 21 AE 4A ED
4630  4B 8A 4A 78 B1 C2 41 46 21 21 4A CD A7 28 C3 09
4640  44 01 00 00 C5 E5 60 69 23 DD 21 92 4A CD 00 47
4650  21 92 4A CD A7 28 3E 3A CD 33 00 E1 06 06 7E E5
4660  C5 CB 7F 28 04 26 FF 18 02 26 00 6F DD 21 92 4A
4670  CD 00 47 21 92 4A CD A7 28 3A 10 38 CB 47 28 03
4680  CD 49 00 C1 E1 23 CD 55 49 10 D3 CD 5D 49 C1 03
4690  3A 8A 4A B9 20 AE 3A 8B 4A B8 20 A8 C3 09 44 3E
46A0  01 32 8E 4A CD 53 47 C2 A4 46 AF 32 8E 4A C3 09
46B0  44 CD 5D 49 21 7F 49 CD A7 28 C3 86 43 C5 06 14
46C0  CD CD 46 C1 C9 C5 06 00 CD CD 46 C1 C9 C5 C5 00
46D0  00 C1 10 FA C1 C9 21 00 00 22 8A 4A AF 32 8E 4A
46E0  21 AE 4A 22 8C 4A CD EA 46 C9 21 9C 4A 11 9D 4A
46F0  36 00 01 11 00 ED B0 C9 F5 D5 CD 33 00 D1 F1 C9
4700  F5 E5 D5 CB 7C 28 10 7C 2F 67 7D 2F 6F 23 3E 2D
4710  DD 77 00 DD 23 18 04 3E 20 18 F5 FD E5 FD 21 49
4720  47 3E 30 FD 5E 00 FD 56 01 B7 ED 52 DA 32 47 3C
4730  18 F7 19 DD 77 00 DD 23 FD 23 FD 23 1D 20 E2 FD
4740  E1 D1 E1 AF DD 77 00 F1 C9 10 27 E8 03 64 00 0A
4750  00 01 00 CD 41 49 3A 40 38 CB 7F 28 06 CD C5 46
4760  CD C5 46 AF 32 90 4A 3A 10 38 CB 47 CA 73 47 C3
4770  F2 47 01 00 00 CB 4F CA 7D 47 CD FC 47 03 CB 57
4780  CA 86 47 CD FC 47 03 CB 5F CA 8F 47 CD FC 47 03
4790  CB 67 CA 98 47 CD FC 47 03 CB 6F CA A1 47 CD FC
47A0  47 03 CB 77 CA AA 47 CD FC 47 01 00 00 3A 04 38
47B0  CB 4F CA B3 47 CD 00 48 03 CB 7F CA C1 47 CD 00
47C0  48 03 00 01 3A CB 6F 28 03 CD 00 48 03 3A 04 38
47D0  CB 57 CA D8 47 CD 00 48 03 CB 67 CA F1 47 CD 00
47E0  48 3A 00 00 03 CB 4F CA FD 47 CD 00 48 CD 17 48
47F0  F6 01 C9 3A 3E 40 B7 C0 AF 48 AF C9 1E 03 18 02
4800  1E 01 21 A8 4A 09 F5 7E B7 28 04 AF 77 F1 C9 73
4810  3E 01 32 90 4A F1 C9 3A 3E 4A B7 C2 93 48 32 8F
4820  4A 06 06 DD 21 A8 4A FD 21 AF 4A 21 A2 4A FD 2B
4830  DD 2B 3D FD 7E 00 B7 28 1F FE 01 28 27 DD 7E 00
4840  FE 01 20 09 CD 81 48 FD ?? 00 00 18 0B 34 7E FE
4850  ?? ?? 01 18 DD ?? 00 93 10 D4 CD ?? 48 3A ?F 4A
4860  B7 20 1C C9 DD 7E 00 FE 02 20 09 CD 81 48 FD 36
4870  00 00 18 E4 ?? ?? FE 00 ?? ?1 48 DD 36 00 01 10
4880  D7 F5 3E 01 32 0F 4A F1 C9 F5 E5 C5 21 A8 4A 06
4890  06 7E B7 20 ?? ?? 10 F9 C1 F1 F1 C9 C1 E1 3E 01
48A0  ?? 7C 49 CD C5 46 9F CD FC 48 CD C5 46 F1 C9 C5
48B0  ?? ?? ?? 00 00 2A 0A 4A 2? 3? FE 01 D2 D8 49 ??
48C0  ?? ?? CD 5B 3C 4A 21 9C 4A 01 06 00 ED B0 ED 53
48D0  8C 4A ?? CA 46 F1 ?1 C? ?? 3? 4A CD 97 28 CD 49
48E0  80 ?? ?? 00 ?? 5D 49 FE 11 CA F3 48 FE 53 CA D5
48F0  48 18 E5 CD ?C 46 C1 C1 ?? C3 09 44 F5 C5 E5 4F
4900  06 06 21 AD 4A 7E F6 0? 5? 20 02 10 29 CB 27 CB
4910  27 CB 27 E6 F0 C5 CB 20 ?? CB 41 38 20 C? EF CB
4920  87 32 E3 37 FC F0 ?? E3 3? CD BD 4C CD BD 46 CD
4930  B? 4? ?? ?? 1? C1 2B 10 CC ?? C1 F1 C? CB AF 1?
4940  DE E5 D5 C5 21 0? 49 11 0? 4A 01 05 00 ?6 00 ED
4950  B0 C1 D1 E1 C9 F5 3E ?? CD F8 46 F1 C? F5 3E 0D
4960  CD F8 46 F1 C9 D5 CD 3B 00 D1 C9 41 52 4D 20 4?
4970  4F 4E 54 ?? 4? 4? 4? 4? ?? 3? 4D 48 31 00 00 4?
4980  ?? 45 0D 00 4C 4F 41 ?? 4? 2? 5? 45 41 44 2C 43
4990  4C 45 41 52 2C 57 52 49 54 45 2C 47 4F 2C 44 49
49A0  73 70 20 4? 4? 4F 54 2? 4? 41 4? 2? 51 55 49 54
49B0  0C 00 46 4F 52 45 5? 4? 5? 3? 4F 5? 20 4F 4E 43
49C0  4? 2? 20 00 54 59 5? 45 20 53 50 41 43 45 20 42
49D0  41 52 ?? 57 4? 15 4? 2? 5? 45 41 44 59 00 53 4F
49E0  53 52 50 0D 00 ?? 54 41 52 54 20 41 47 41 49 4E
49F0  ?? 20 4F 52 20 43 4F 4E 54 20 46 52 4F 4D 20 43
4A00  55 52 52 20 50 4F 53 0D 00 4D 4F 56 45 4D 45 4E
4A10  54 ?? 41 52 52 41 59 2? 44 4? 5? 5? 4C 41 59 0D
4A20  00 0D 4E 4F 20 53 45 51 55 45 4E 43 45 20 49 4E
4A30  ?? ?? 54 4F 52 45 0D 00 4E 4F 20 4D 4F 52 45 20
4A40  41 52 4? 2? 53 54 4F 52 45 ?? 4? 45 4? 54 2C 44
4A50  4? 4C 4? 54 ?? 2? 4F 52 20 53 41 56 45 3F 20 0D
4A60  00 0D 53 45 51 55 4? 4E 4? 45 20 43 4F 4D 50 4C
4A70  45 54 45 0D 00 52 45 41 44 20 45 52 52 4F 52 0D
4A80  00 54 41 50 45 20 4F 4B 0? 00 00 00 AE 4A 00 00
4A90  00 00 20 30 30 30 ?? 31 00 01 4? 7? 00 00 00 00
4AA0  30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

[sep]arately since if you have all six in [op]eration at any given moment, [h]orrendously complex gyrations of the [r]obot arm are possible and it will not [a]lways be easy to see exactly what the [f]aults 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!

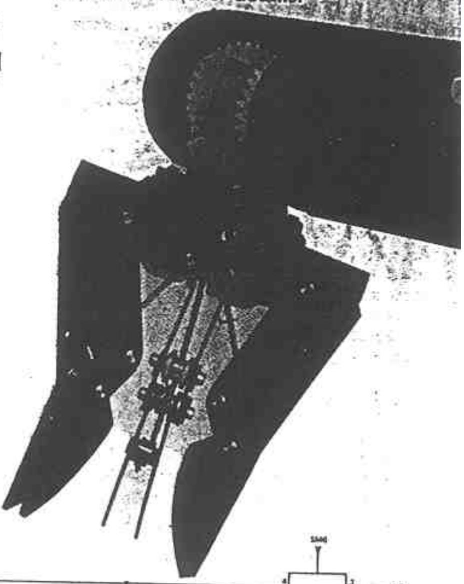
Refer to the component overlays and circuit diagrams provided during construction at each stage. *Do not* simply 'knock the whole thing together' and then start checking! IT WILL FAIL. While the interface and/or control box is not particularly expensive, there is no point in throwing money away by merrily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



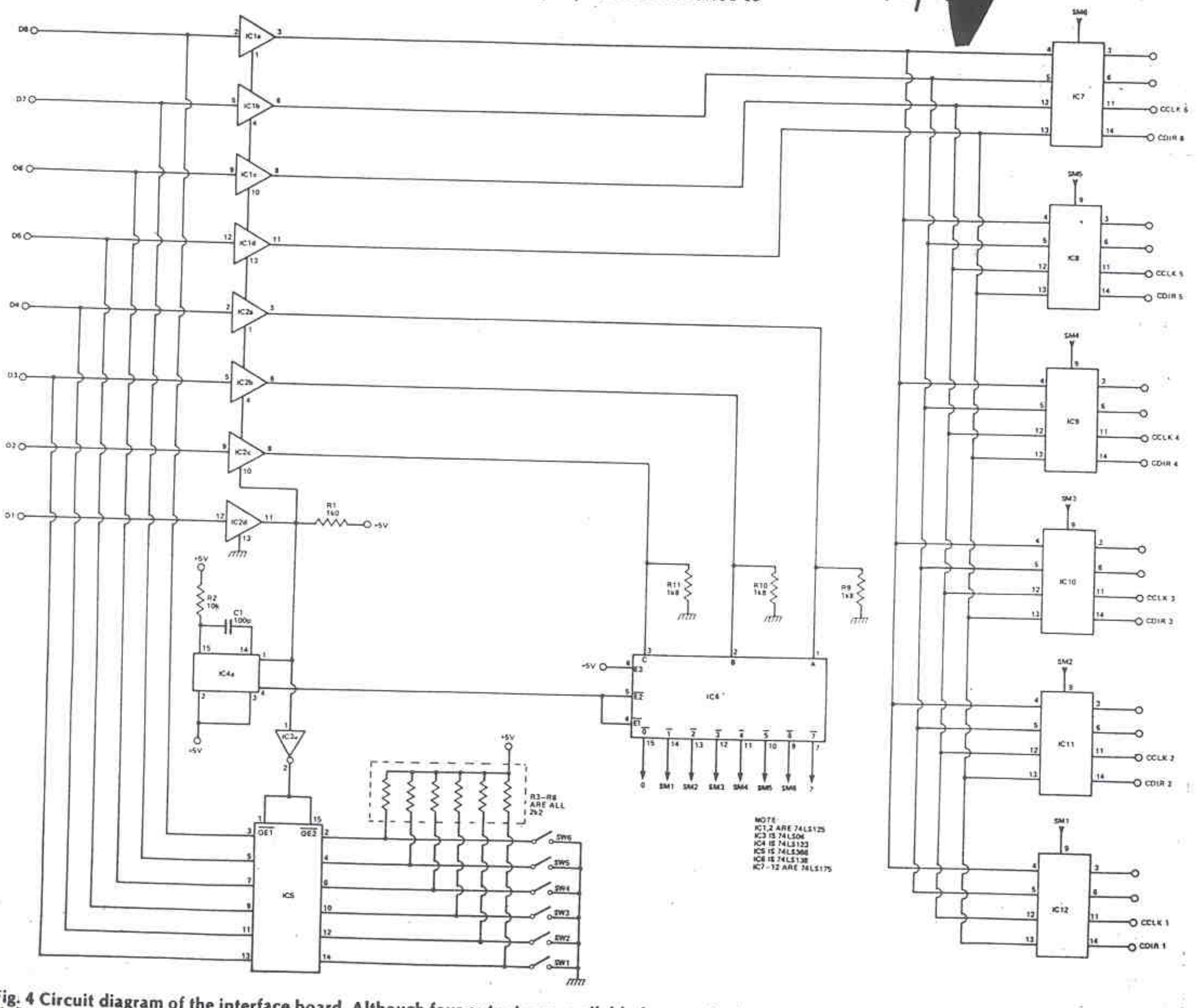
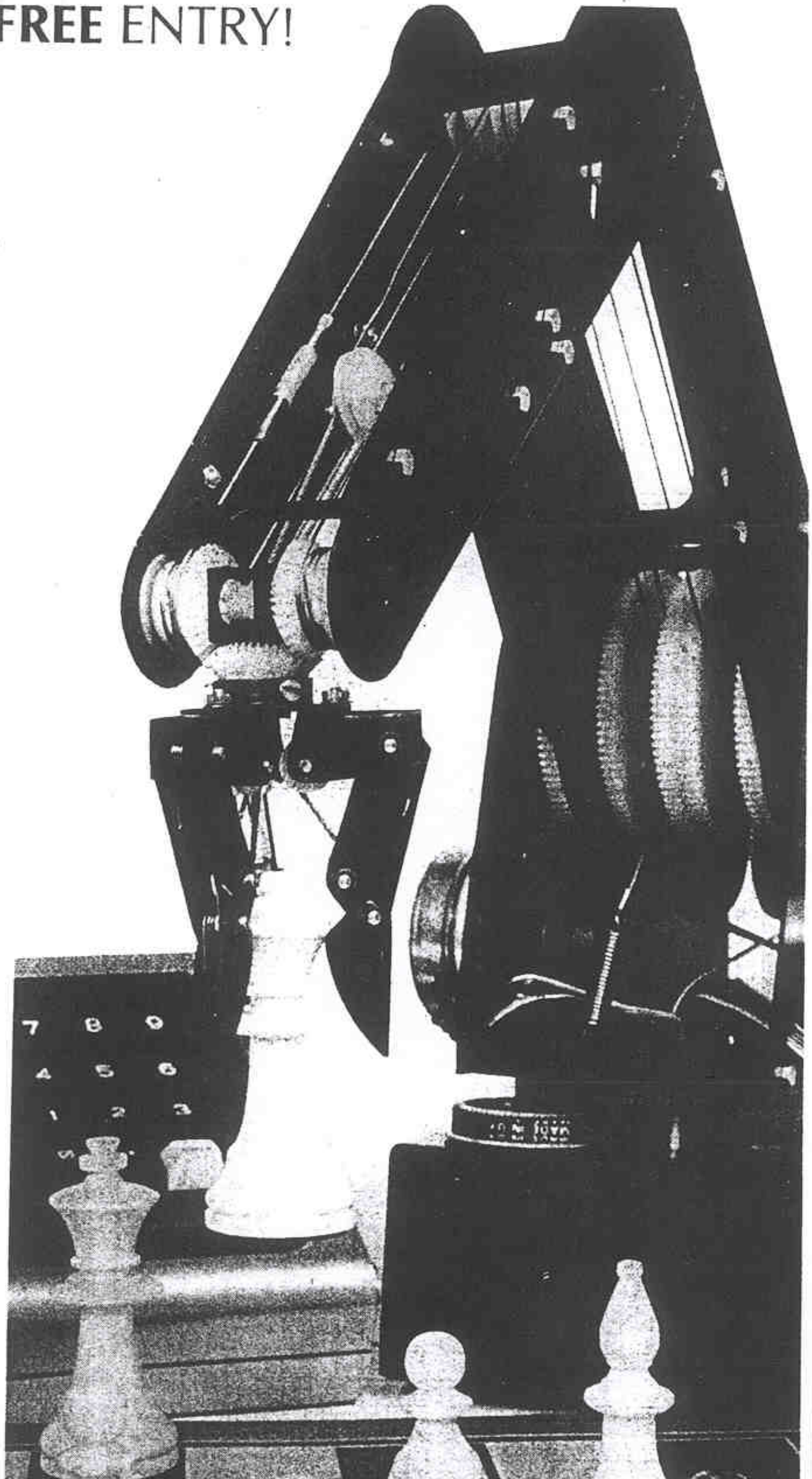


Fig. 4 Circuit diagram of the interface board. Although four outputs are available from each of the six latches, only the two labelled outputs (CCLK and CDIR) are used in this application. This particular section of the design is very versatile; for example, driving triacs from the latch outputs gives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

£100!
IZE £25
RIZE £10

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 Genie
- Apple/ITT 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: Armdroid Competition, Electronics Today International, 145 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 I 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 Modmags 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.

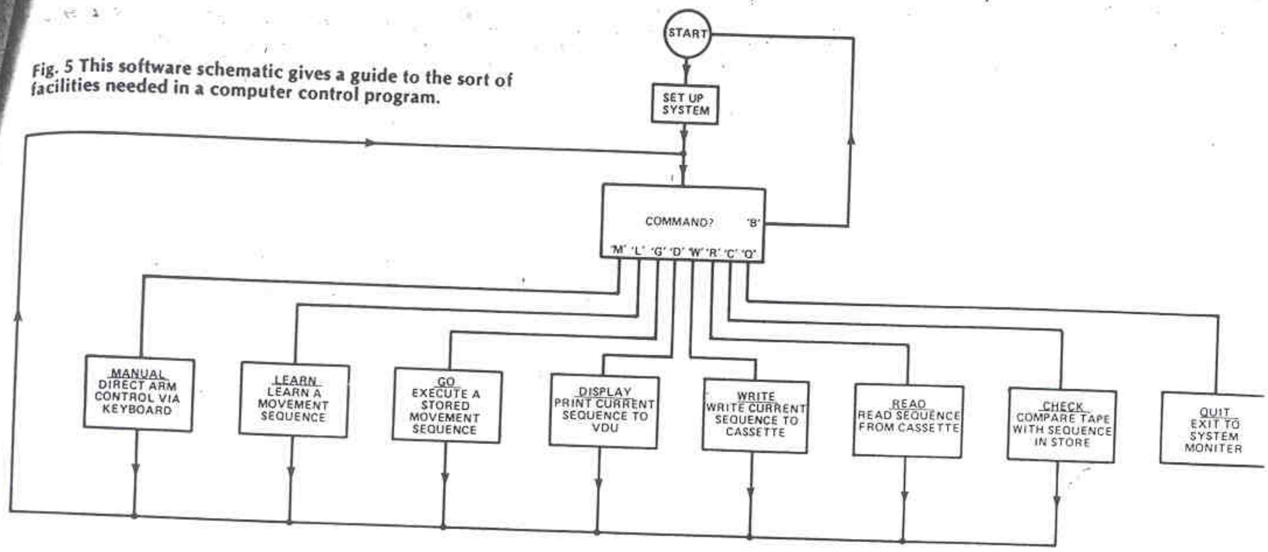
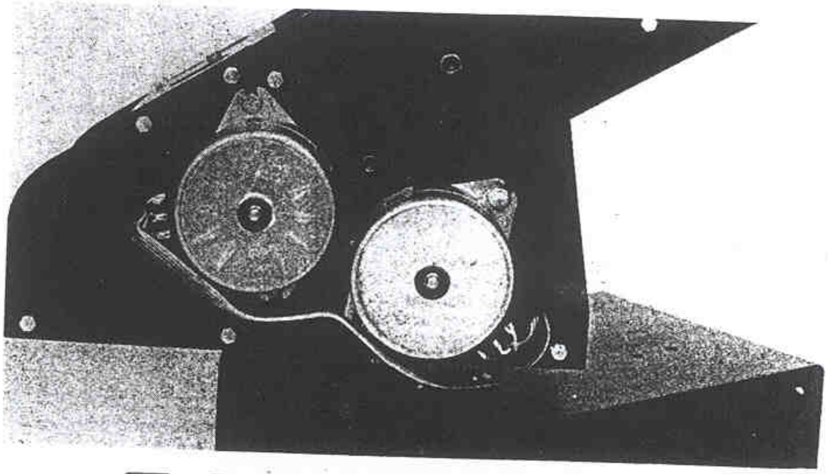


Fig. 5 This software schematic gives a guide to the sort of facilities needed in a computer control program.

CLOCKWISE ROTATION ↑ / COUNTERCLOCKWISE ROTATION ↓

STEP	Q_A	Q_B	Q_C	Q_D
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

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



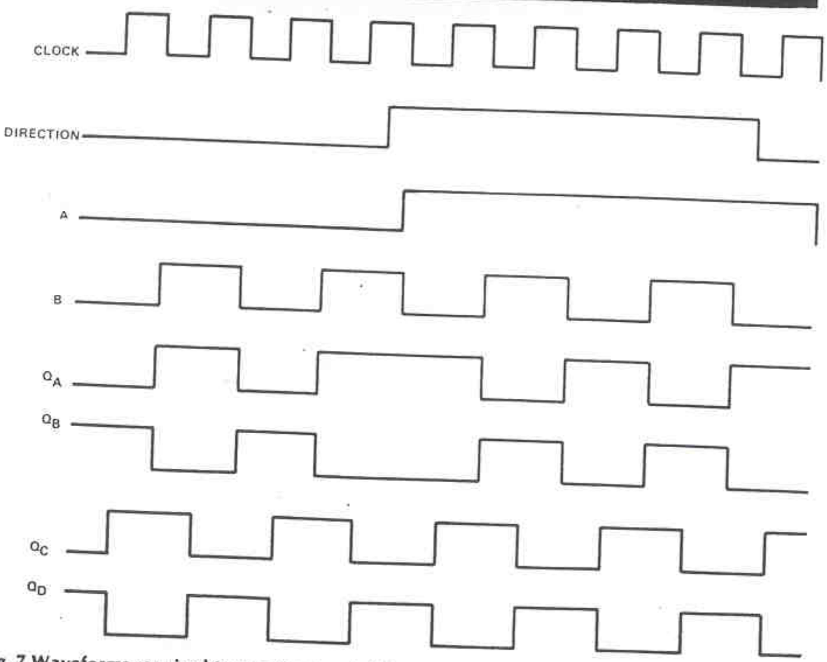


Fig. 7 Waveforms required to step the Armdroid motors correctly. These can be generated using fairly simple circuitry.

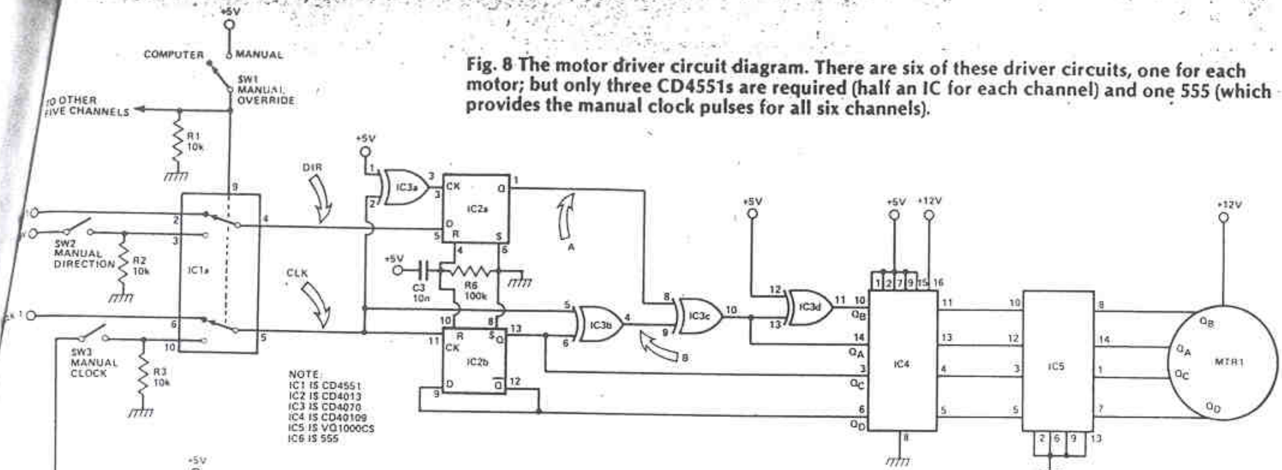
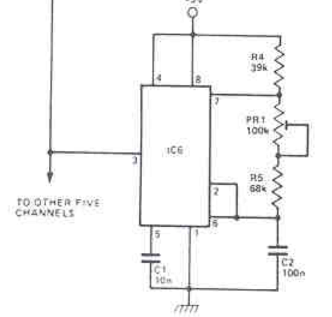



Fig. 8 The motor driver circuit diagram. There are six of these driver circuits, one for each motor; but only three CD4551s are required (half an IC for each channel) and one 555 (which provides the manual clock pulses for all six channels).

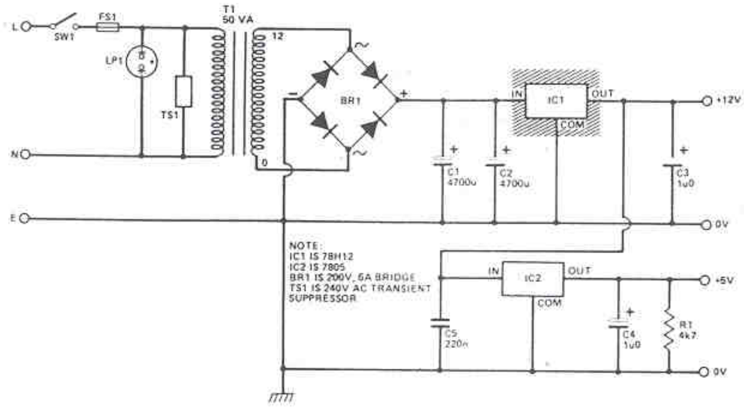


Fig. 9 Circiut diagram of the power supply for this project, which is capable of driving the boards and all six motors.

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 half stepped by the processor. If the above is carried out, then ICs 1, 2 and 3 will be redundant in each motor drive logic section.

The circuitry 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.

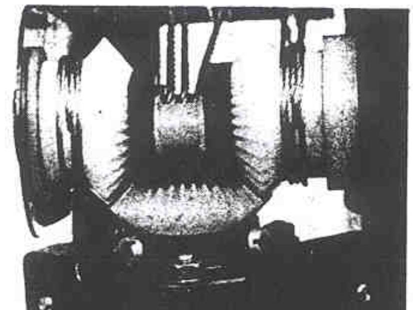
The six motor driver stages need two power supplies to function; 12 V at about 3A5 and 5 V at 150 mA.

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 CDIR and CCLK signals will be selected and placed on the DIR and CLK lines. In manual mode, clock pulses are fed into the system from IC6 (a simple 555 astable) via SW3, 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 CLK 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 CLK. The $\overline{Q}$ 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 CLK (waveform Q_C in Fig. 7). The waveform Q_D is simply the inverse of Q_C and is taken from the $\overline{Q}$ output of IC2b. A capacitor/resistor network (C3/R6) resets the flip-flops on power-up.

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

The four waveforms Q_A-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 four 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 Armdroid project with the Parts Lists and component overlays.