

RICHARD PAWSON

ROBOT PROJECTS

A PRACTICAL GUIDE TO MAKING HOME ROBOTS

Full constructional and programming details inside

Contains 8 low-cost projects

Build your own fun and educational robots

Interfacing robot devices to a home computer



AS SEEN ON TV
£1.99

CONTENTS

Introduction	1
Construction kit guide	2
Basic electrical skills	4
Key to wiring diagrams	6
Buggies in Fischertechnik and Lego	10
Harmonic drive simulation	15
Motor control	16
Toy android	18
Draughts-playing robot	24
Lego robot arm	28
Card-dealing robot	34
Free-roving robot	40
Useful addresses	48

Senior Editor
Project Editor
Art Director

Lionel Bender
Sarah Mitchell
Bernard Higton

Main photography
Instructional photography
Typesetting
Reproduction

Ian McKinnell
Henley Photographics Ltd
Facet Filmsetting
Hong Kong Graphic Arts Centre

Printed in Yugoslavia

Robot Projects was conceived, edited and designed by
Frances Lincoln Limited,
Apollo Works, 5 Charlton Kings Road,
London NW5 2SB

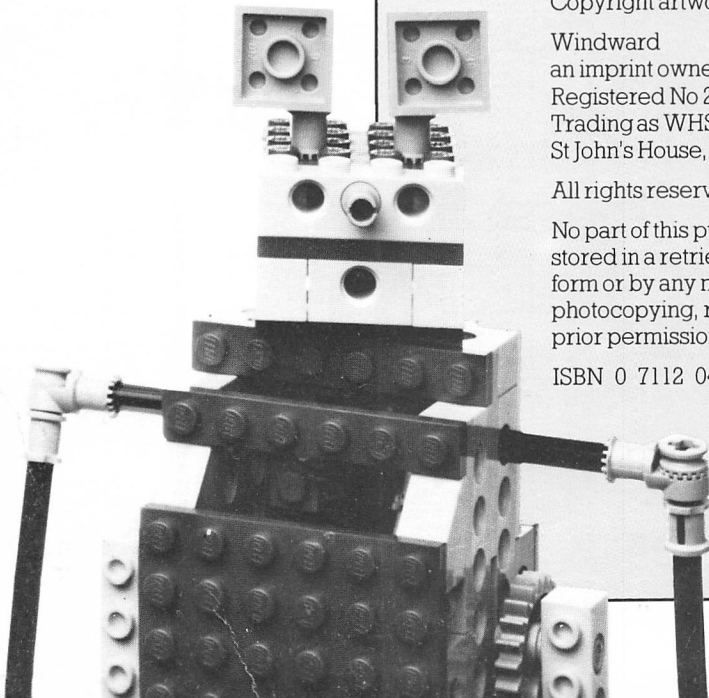
Copyright © Frances Lincoln Limited 1985
Copyright text © Richard Pawson 1985
Copyright artwork © Frances Lincoln Limited 1985

Windward
an imprint owned by W H Smith & Son Limited
Registered No 237811 England
Trading as WHS Distributors,
St John's House, East Street, Leicester, LE1 6NE

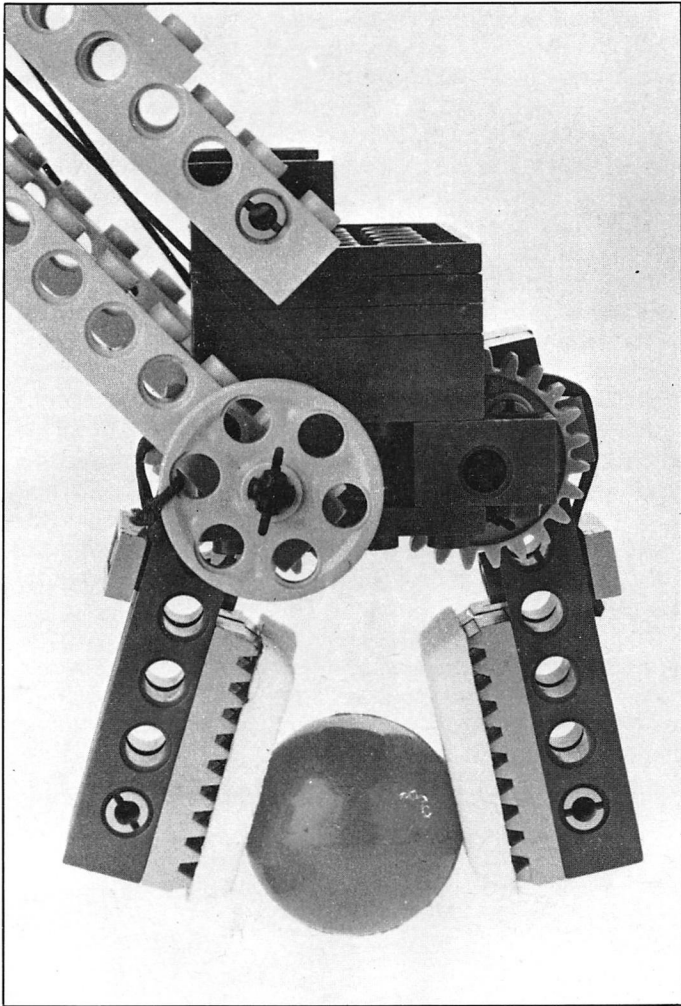
All rights reserved.

No part of this publication may be reproduced,
stored in a retrieval system, or transmitted, in any
form or by any means, electronic, mechanical,
photocopying, recording or otherwise without the
prior permission of the publishers.

ISBN 0 7112 0419 5



INTRODUCTION



THIS magazine comprises a series of practical robotics projects you can try out for yourself, at home or at school. Some of the robot designs can perform nominally useful tasks such as dealing a hand of cards or playing draughts, but the purpose of the projects is to demonstrate the principles from which real robots are constructed. After following a few of our designs, you should be able to combine the features into a robot of your own specification.

Construction materials

All projects have been designed with standard educational construction kits, either Fischertechnik or LEGO®. This eliminates the need for any specialist tools or techniques such as machining or gluing. However, on some occasions the standard kit components can work out more expensive than, say, balsa wood, plastic or sheet metal. If you are familiar with working

these materials then you should certainly apply them to your robotics experiments.

The cost of the projects has been kept as low as possible, though most will still require two or four electric motors. The more Fischertechnik or LEGO that you have available, the greater the range of projects you can attempt, so it is a good idea to club together with other enthusiasts or to use your school's or college's facilities. On pages 2-3 some advice is given as to which Fischertechnik and LEGO components are the most useful to the robotics experimenter and how to buy them in the most economical way.

Control systems

To qualify for true robot status, mechanisms should be programmable. Most of our robot projects can be linked, or interfaced, to a home computer, and either operated live from the computer's keyboard or left to execute a program that has been previously typed in and stored in the computer's memory. All the interfacing is done through a relay block, which can be purchased for most home computers for between £20 and £50. By using one of these blocks (the functioning of the devices we used is explained on pages 8-9) you will not need to buy any electronic components, nor in most cases will you even need a soldering iron, since the circuits simply plug together. It is assumed that you already know how to program your own computer and that you are fully conversant with the commands required to operate your relay block should it differ from the one that we use.

Each of the projects includes a schematic wiring diagram showing how the motors and other components are connected to the relay block. There is also a sample program to demonstrate the operation of the model. The program has been carefully written and documented so that it will be easy to modify it for your own system.

If you don't have access to a home computer, or haven't yet grasped the essentials of programming, then it is possible to operate most of our projects using the conventional battery boxes and switches supplied by Fischertechnik and LEGO. The models, however, are less exciting with the programmable element removed.

These projects are intended to give you the flavour of robotics experimenting. Robot building can be a fascinating hobby, combining the skills of the programmer with those of the model engineer, and it has considerable educational value, too.

CONSTRUCTION KIT GUIDE

Here we compare Lego and Fischertechnik, highlighting their relative merits as robot construction kits. We also give advice on buying the components you will need for our projects.

Lego

Lego started out as a more sophisticated replacement for conventional building block sets, which could be used by young children for constructing model houses. The patented system of knobs and holes for holding components together was actually based on a feature from ancient Greek architecture. Over the years, the original components have been supplemented by wheels, lights, motors and gears to increase the range of models that can be built. Lego now accounts for around 80 per cent of the construction set market.

Technical Lego – now called Lego Technic by the manufacturers – took this a stage further. Intended for use by older children, it was designed with mechanisms rather than just static objects in mind. It has to a large extent filled the niche vacated by Meccano, who fell into difficulties in the 1970s but who had previously dominated that market.

Lego (for which read Lego Technic) is thus highly suitable for constructing your own robots. At the time of writing, Lego have yet to introduce their first robot-related set, but that cannot be far away. Hopefully, the range of technical components will be expanded in the future to incorporate sensors and other electrical devices.

Buying Lego

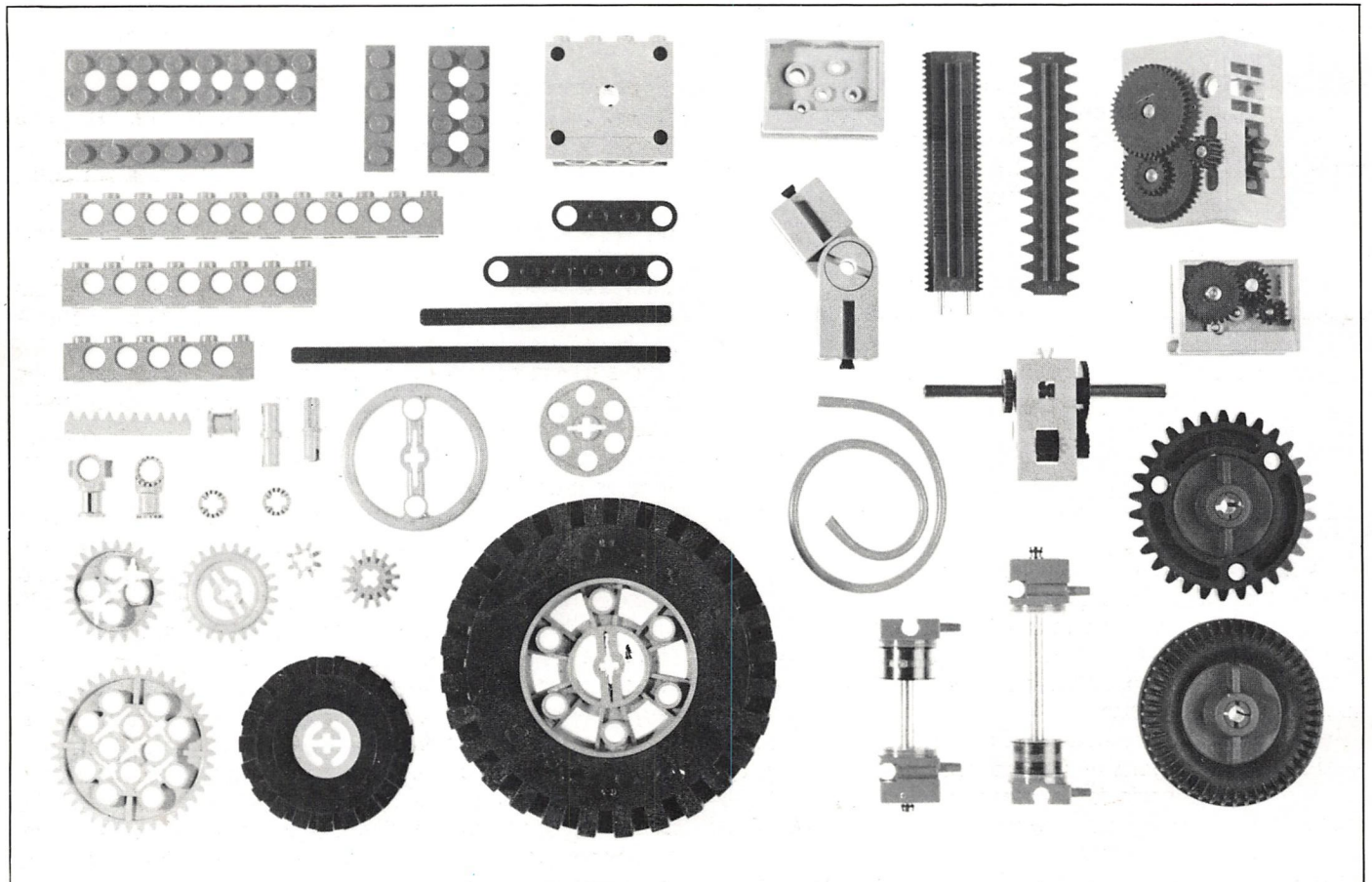
Lego Technic is sold primarily through toyshops and department stores, and most people start out with one of the larger sets, based on a specific model such as a tractor or motor car chassis. These provide a good stock of components with which to start experimenting but are not an economical way of expanding your set.

The 8710 Supplementary Set offers very good value for money – most of the Lego projects we describe could be made from two such sets alone. Lego also run a spares service, through which you can purchase individual packs of axles, beams, joints, etc. If you don't have the tractor set, or similar, then this is the best way to acquire the large wheels and tyres. Several projects make use of the 40-tooth gear wheels which do not feature in many sets but are available in a spares pack.

Many schools are unaware that they can purchase special educational packs of spares at a substantially reduced price, so it's worth suggesting these robot projects to your teachers!

There's no avoiding the fact that robot projects need plenty of motors – some can be built from one or two, but several require four. Lego motors come in two varieties: 4.5 volt and 12 volt. All the Lego projects in this booklet feature the former type, but should work perfectly well using the latter. It is best to start with a couple of the 8700 motor sets, which include the battery boxes and switches, but thereafter individual motor units can be purchased for half the price of a motor set through the spares service.

The 8720 gearbox set is also very useful, supplying two types of gearbox and a chain drive as well. The robot arm on pages 28–33 uses one such set – the rover on pages 40–47 uses two. They are quieter in operation and more compact than a whole train of individual gears.



Lego v. Fischertechnik

A comparison of strengths and weaknesses

Lego

All measurements must be multiples of the standard unit spacings.

Easy and fast to assemble.

Fixing system relatively weak.

Fixing system based on original building blocks and therefore very 'directional'. It is possible to align any component any way, but only with difficulty.

Easy to pivot one beam or structure on another with connecting pins.

Beams long and slender.

Useful range of gear wheels with fine teeth.

Gears and wheels fit positively on splined axles. However, wheels can't be made to spin freely.

Complete lack of electrical sensors for feedback or control purposes.

Fischertechnik

Sliding fittings allow fine adjustments.

Order in which components must be put together not always obvious.

Fixing system very strong.

No directionality to the fixing system and therefore very good for building three-dimensional objects, particularly the supporting structures around mechanisms.

Difficult to make a pivot joint, except with the bulky elbow bricks.

Building blocks rather fat and cumbersome for some applications.

Limited number of gears, each rather bulky.

Gears and wheels rely on friction to grip axles and therefore can't transmit very high torque. Can be made to freewheel, though.

Wide range of electrical sensors and other components suitable for robotics.

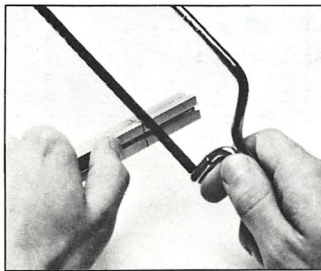
Fischertechnik

Fischertechnik is manufactured in West Germany, and though it is a rival system to Lego Technic, it has different origins. In the first place, Fischertechnik was designed originally for technical applications (both statics and mechanics), where Lego grew out of an educational toy. Secondly, it is aimed firmly at the educational and even industrial marketplace. As a family product, in this country at least, it has a low profile, largely due to its higher relative cost and low availability in toyshops, though this may change with the growing interest in home robotics.

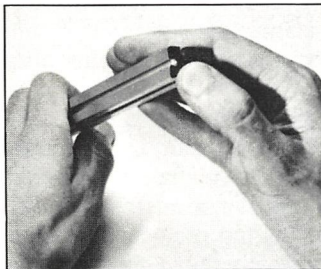
In schools and universities, Fischertechnik is used for teaching maths, physics and engineering, while in industry it has formed the basis for engineering prototypes of new products and even to simulate the operation of large plants such as car production lines.

Though in some ways more difficult to assemble than Lego,

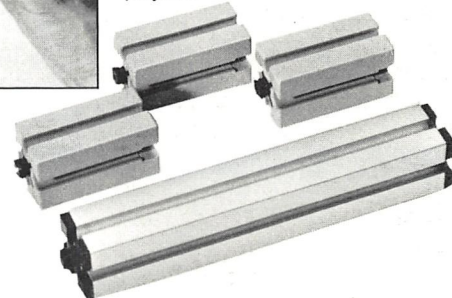
FISCHERTECHNIK ALUMINIUM EXTRUSION



For long beams, aluminium extrusion is neater and stronger than lengths of standard bricks. It is purchased in lengths of 3m (10ft) and can be cut with a junior hacksaw. Mark around the extrusion with a pencil to make sure that the end is sawn off straight.



Black plastic endpieces are purchased separately and inserted. You are advised to cut the extrusion only to standard block lengths (allowing for the endpieces), so that your long beams can be used in a wide variety of projects.



robots constructed from Fischertechnik more closely resemble their real life counterparts, both visually and functionally. The wide range of sophisticated components, particularly electrical, reduces the need to incorporate non-Fischertechnik parts, which would be difficult to mount.

Buying Fischertechnik

All Fischertechnik sets are general purpose rather than geared towards specific models, and this works to the advantage of the robot experimenter. The UT1 is the best startpoint, preferably supplemented with a UT2 motor set. The minimotors are more useful for robot building than the main motors, though you will need at least one of the latter. Minimotors can be purchased in special sets, complete with a useful range of gears, racking and electrical circuit components, or individually through the spares service.

It is in fact possible to order any component through the spares service, and this is one of Fischertechnik's great assets. The range of electrical components is the best of any construction system, with microswitches, electromagnets and light sensors to name but a few.

The aluminium extrusion is also invaluable for our purposes, and was interestingly developed specifically for the BBC Buggy. Sawn to the desired length and fitted with end pieces, it provides a neater and stronger structure than individual blocks and is a first-class engineering material.

Two more recently introduced sets of value in robotics are the pneumatics and computing kits. We have used these in only one project, the Card-dealing robot shown on pages 34-39, though if you have them they could be built into more. Again, the individual components are available as spares. The statics set looks as though it could be useful because the beams are much thinner than the conventional blocks. They are, however, much more pliable, and therefore not to be recommended for robots.

BASIC ELECTRICAL SKILLS

The projects described in the following pages, and other robots that you may build, can be run 'live' from a series of battery boxes fitted with on/off or reversing switches. Interfacing a working model to a computer, however, is far more satisfying. It not only opens up the possibility of programming a sequence of movements into the computer's memory, but it is a very good way to learn about electrical circuits and electronics in general.

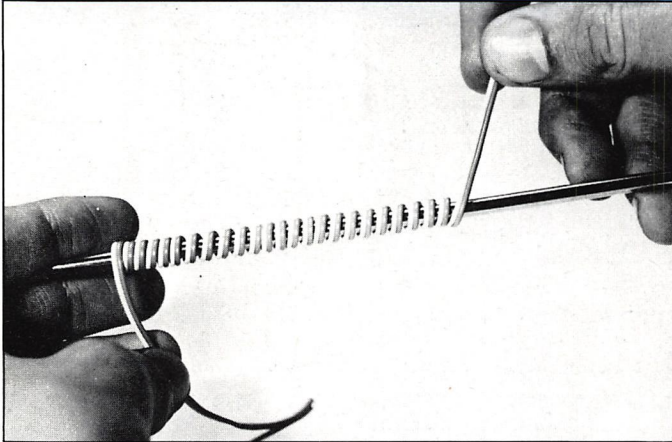
Accordingly, the electrics contained in this booklet are designed to be followed by someone with no previous experience of electronics. Those who have such experience can certainly improve on the performance of the models, using specially designed circuits. But the projects as shown do not require the purchase of any electronic components, and with the exception of the one or two that entail solder-

ing, can be wired together with a small screwdriver.

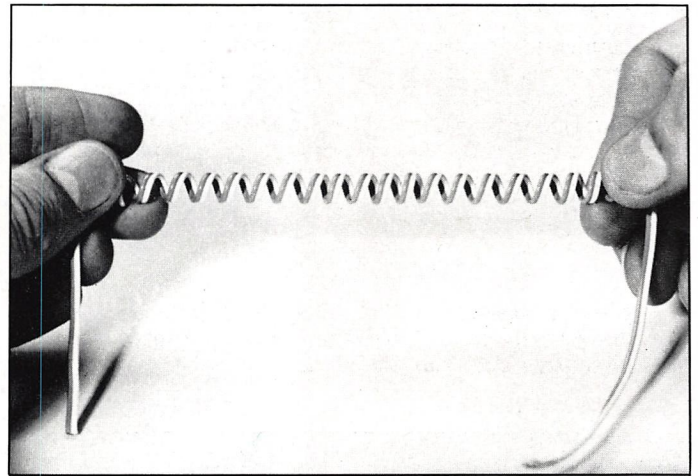
We need to start off with three necessary cautionings. First, none of these projects involve mains electricity. Controlling household appliances from a computer is becoming a popular application, and would be particularly appealing to the robotics experimenter. There are standard and quite safe interfaces available from electronics shops for this sort of thing, but unless you are very experienced in electronics, you are advised not to try interfacing mains devices to a computer.

The second hint is for your equipment's safety, as well as your own: before wiring up a circuit or making modifications, make sure that all equipment is switched off, batteries removed and the computer unplugged from the mains. Third, elsewhere you may find robot projects that are modifications to electronic toys or off-the-shelf robots. The modifications are usually all reversible. They would nevertheless invalidate the original warranty, and there is always the risk that the product could be accidentally damaged.

Coiling cable



Coiled cables add a professional touch to a finished model. Start by winding the wire around a thin rod or axle, ensuring that it lies flat as shown.



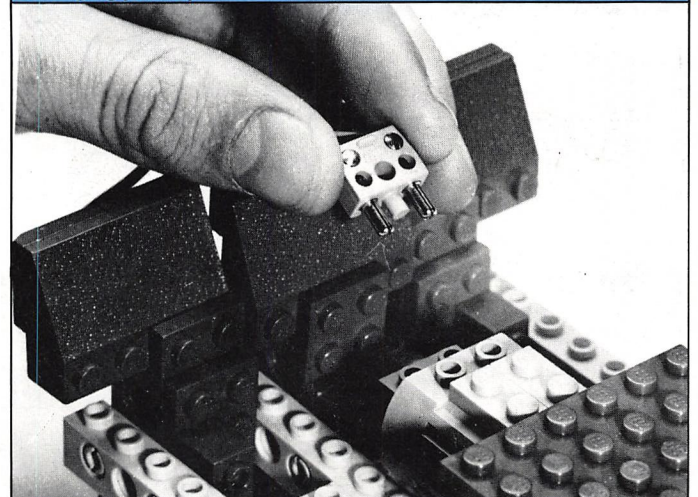
Slide the axle out and stretch the coil slightly. Coiled cable is slightly elastic, which keeps the wiring neat, even when there is movement between the ends.

Wire stripping



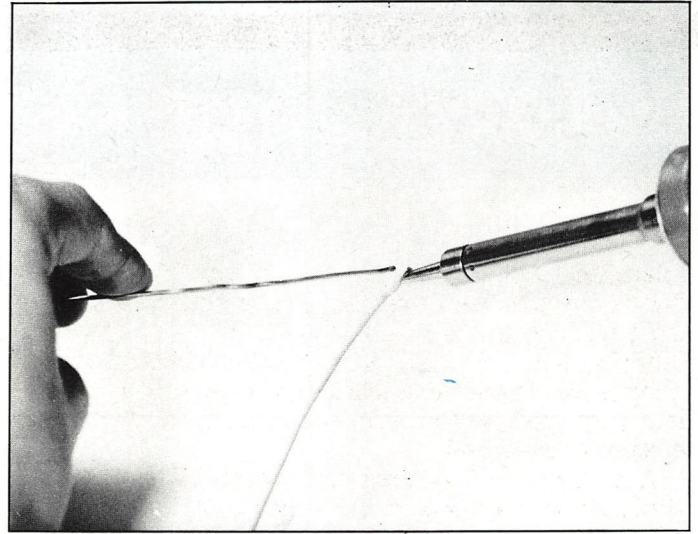
If you don't have proper wire-strippers, a pair of long-nosed pliers and sidecutters can come in very useful for stripping the ends of short lengths of wire.

Changing polarity



If, having assembled a model and connected it to a computer, parts of the robot move in the opposite direction to that expected, the polarity on one or more of the motors simply needs to be reversed. That entails swopping over the two wires leading to the motor, or in the case of Lego, turning around the motor plug.

Soldering

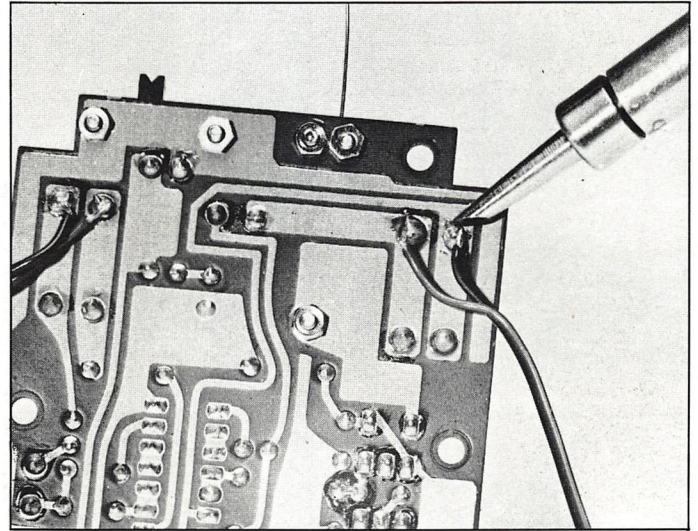


None of our projects require any soldering, but solder joints provide the best electrical connections so are recommended. Soldering is not difficult. You will need a soldering iron with a fine tip – preferably a fairly modern design that is intended for electronics work – and some multicore solder. If you don't have access to a vice, Blu-tack is useful for holding the workpiece at the right angle.

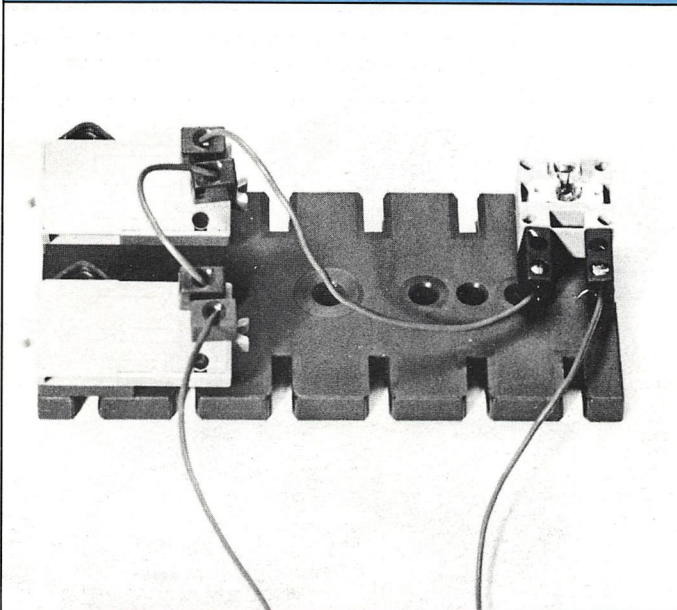
The Golden Rules of Soldering: (1) Disconnect all batteries and power supplies from whatever you are soldering. (2) 'Tin' (coat with solder) all components before attempting to join them. (3) Never apply solder to the soldering iron and then try to transfer it to the joint. (4) When soldering an electronic component or a printed circuit board (PCB), don't leave the soldering iron on the joint for more than a few seconds or the electronics could be damaged.

Tinning the ends of all wires helps considerably. Hold the tip of the iron against the bare wire for a few seconds, then add the solder until a little runs onto the wire. Remove iron and solder and wait to cool.

A tinned piece of wire should affix to a joint on a PCB without additional solder. Place the wire on the joint, then apply the soldering iron for a few seconds. A little solder can be added if the joint is weak.

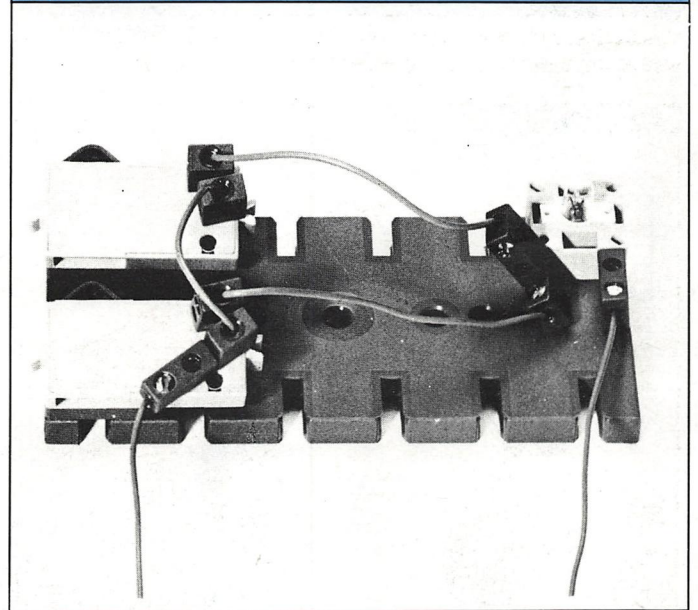


A series circuit



These two microswitches are connected in series and they behave like an AND function in a computer. The circuit is completed (in this case to light the bulb) only when both switches are pressed.

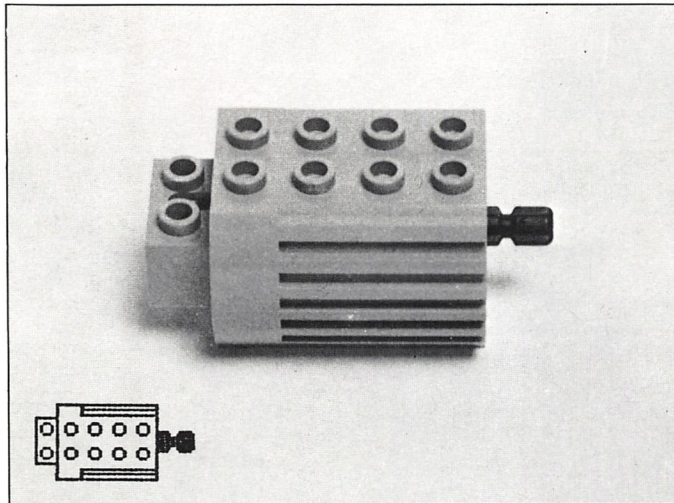
A parallel circuit



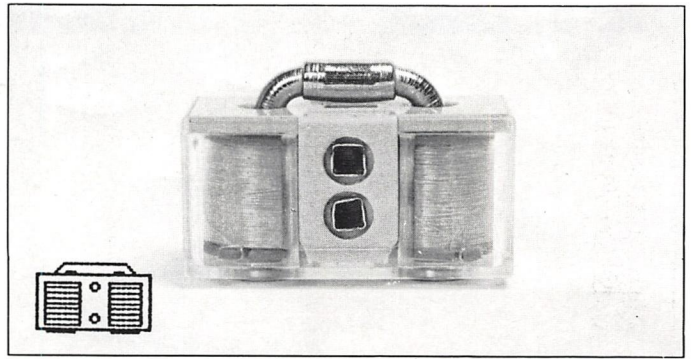
These two microswitches are connected in parallel and they behave like an OR function in a computer. The circuit will be completed by pressing either (or both) of the switches.

KEY TO WIRING DIAGRAMS

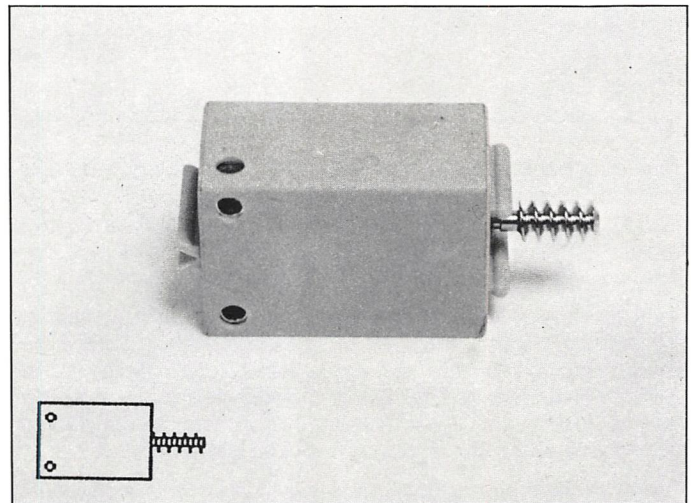
Each of our projects that can be connected to a home computer features a schematic wiring diagram, which shows the interconnection of components but not the actual lengths of wire to be used. Shown here is an explanation of each of the symbols used in these diagrams, together with some of the common circuit arrangements.



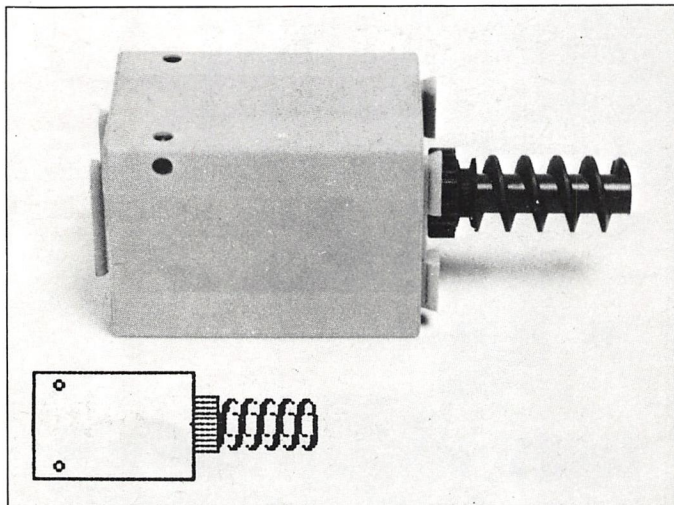
Lego 4.5v motor



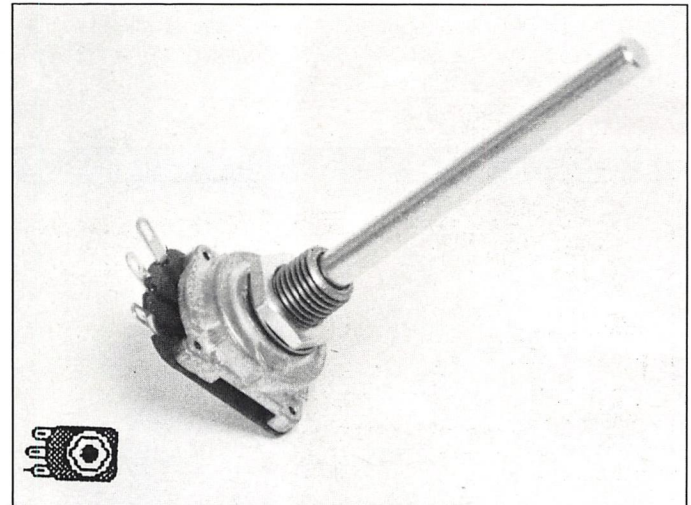
Fischertechnik electromagnet



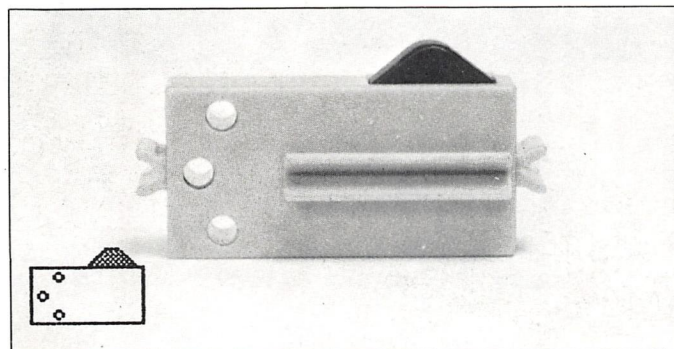
Fischertechnik minimotor



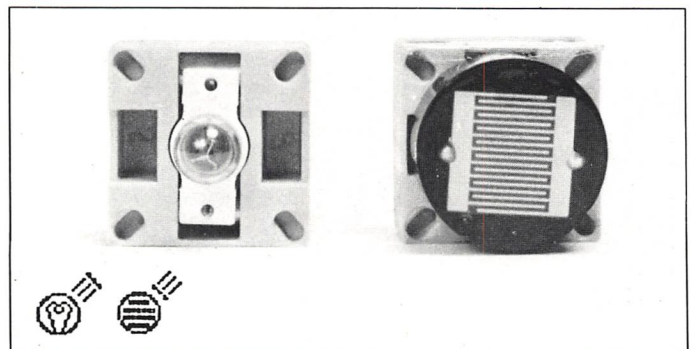
Fischertechnik motor



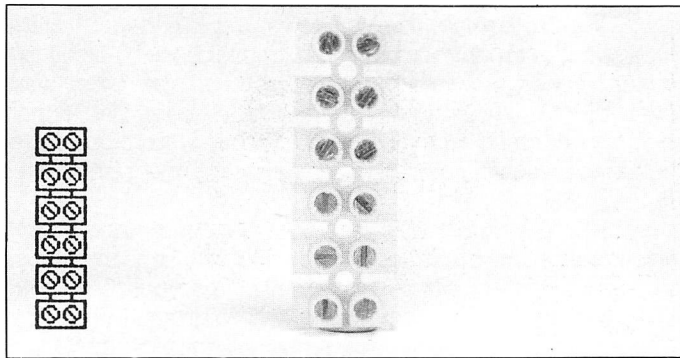
Fischertechnik potentiometer (computing kit)



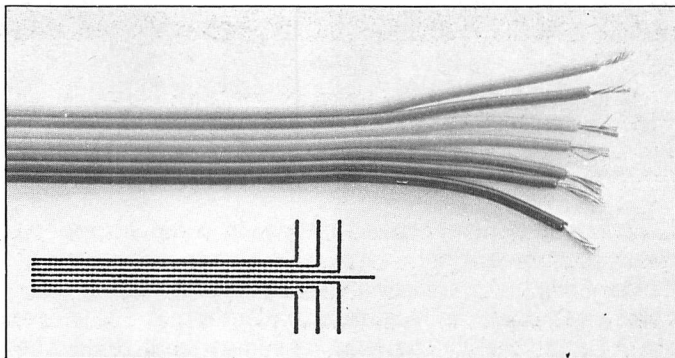
Fischertechnik microswitch



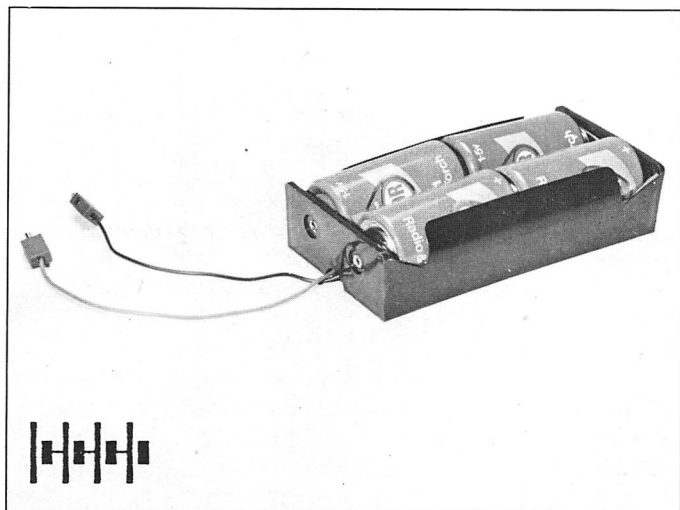
Fischertechnik light bulb and light-sensitive cell



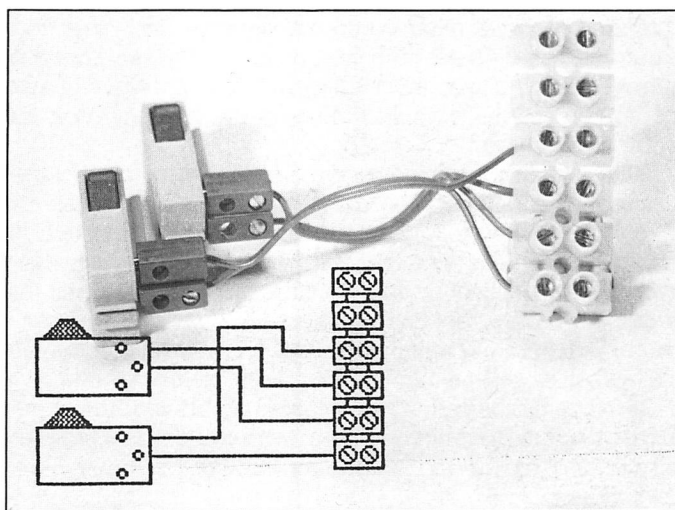
Connector block



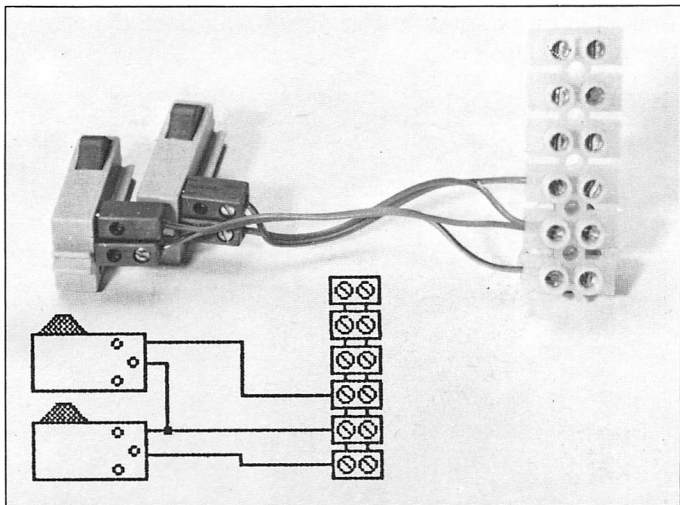
Ribbon cable



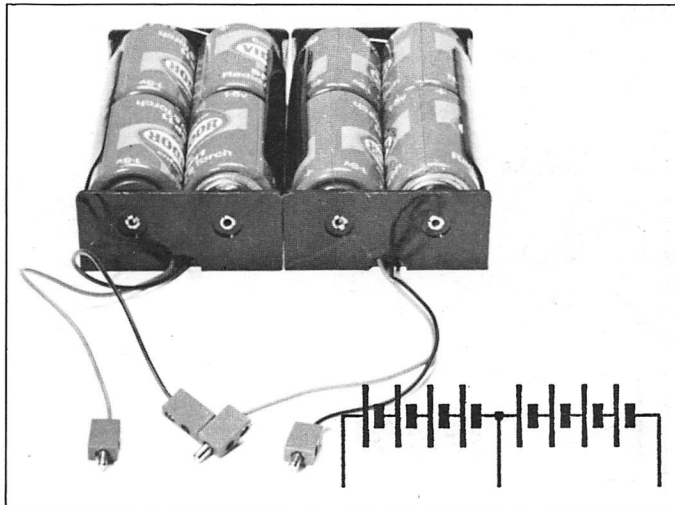
6v battery box for use with Fischertechnik



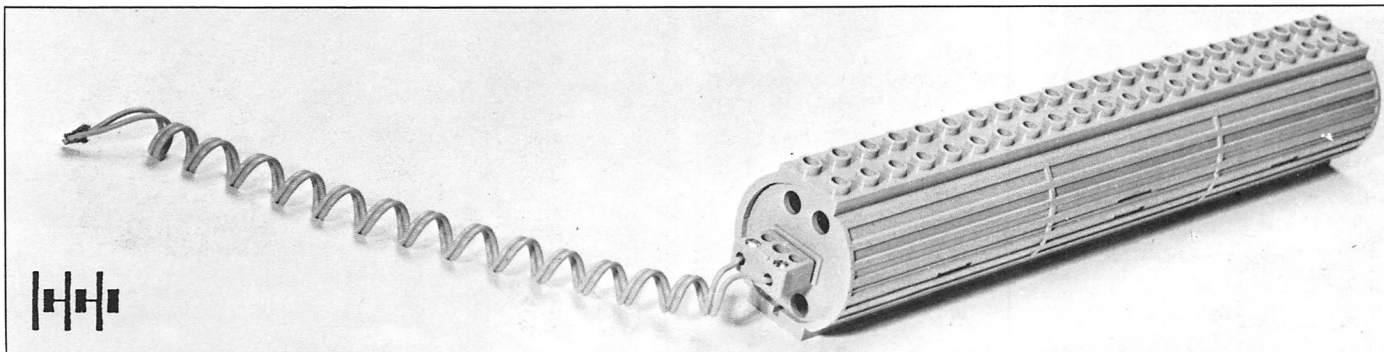
Two wires crossing, without joining



Two wires joining (join usually made at nearest junction)



Double battery box



Lego 4.5v battery box

INTERFACING

Despite their apparent speed and power, computers operate on tiny electronic signals – neither the current nor the voltages inside a computer are large. Devices like motors and electromagnets, however, require much larger currents to operate. A Lego motor, for example, consumes about $\frac{1}{2}$ amp when running freely, and upwards of 2 amps when turning a heavy load. The output ports on a home computer are primarily intended to drive other computer devices, and their maximum current is measured in milliamps (thousandths of an amp). Were you to connect a motor to the output port of most computers, not only would it not turn but it might 'cook' the computer's chips.

The solution is to drive the motors from their normal sources (batteries) and use the computer to make and break the circuit, but without letting the large currents leak back into the computer. The easiest technique for achieving this is to use a relay. A relay has two separate circuits: an input circuit that works on low current, and an output circuit that can handle larger currents. Current in the input activates an electromagnet, which closes the output circuit. Most motor cars employ a relay (though it is called the solenoid) to operate the starter motor, because it would be dangerous to have the 100 amps or so needed by the motor running through the ignition key switch.

A relay is not the only solution to the problem. There are purely electronic devices that employ no mechanics at all, but the relay is an inexpensive device and more difficult to overload and burn out. A relay interface unit, or relay block, is available for most home computers, and with some you will get the choice of several different types. They all work on the same principles, but the method of programming them from the computer varies, as does the nature of their output connections. Here we have insufficient space to print instructions for all possible computers working with all possible relay blocks, and in practice we have restricted ourselves to working with the three widely available systems illustrated: the Commodore Vic-20 with the VicRel (the latter will also work with the Commodore 64), the Sinclair Spectrum with the InterSpec, and the BBC Model B with the InterBeeb. The suppliers of the relay units are listed at the back of the booklet.

We have tried, however, to make the wiring diagrams sufficiently clear and the programs sufficiently well structured and documented to make it easy to convert them to work with another system, assuming that you have read the instruction manuals of your computer and interface units.

Which relay?

The VicRel has been used in the majority of projects because its six independent relays offer the most flexibility. The InterSpec and InterBeeb (the units are effectively the same) feature only four relays, and one terminal from each of the four is wired together. Most of the projects will transfer easily across – the example on this page shows an equivalent circuit wired up for the two different systems. For the sophisticated projects that do require more than four relays, the only alternative is to implement some of the functions manually – such as using a reversing switch to change motor rotation direction whenever prompted by the computer.

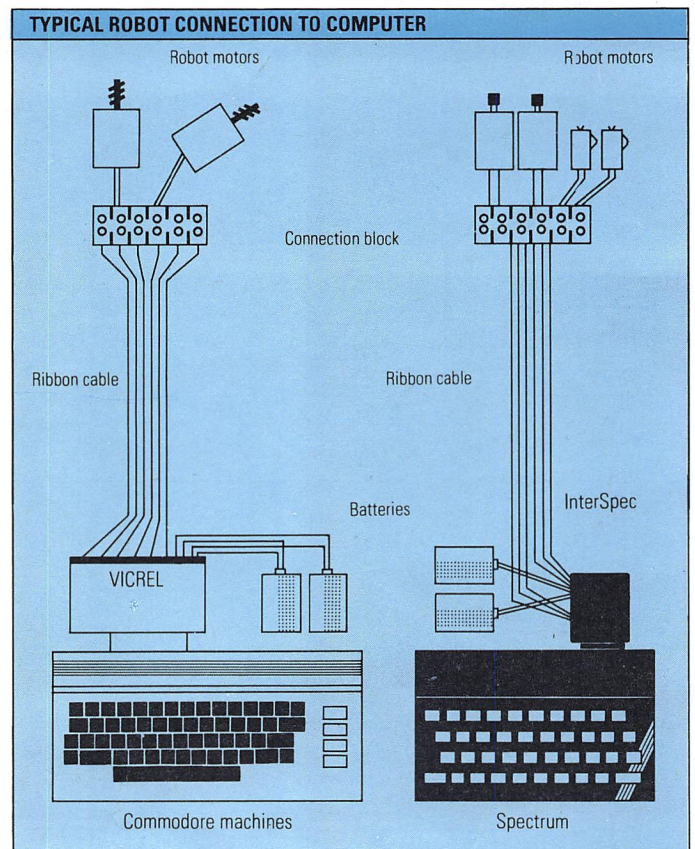
Having all the relays combined together also means that it is only possible to control motors in both directions by using two battery boxes. Accordingly, this approach has been used even with the VicRel, to make conversion easier. The technique of controlling a motor in both directions from one set of batteries is shown in the Lego Toy Android project (see pages 18–23).

Both the interface blocks used have facility for inputs as well as outputs so that the robot can be fitted with sensors such as microswitches or light-sensitive cells. The VicRel has two switch inputs, and the InterSpec and InterBeeb units four. Though constructed from solid state electronics, they act a little like relays in reverse – they protect you from accidentally damaging your computer by inputting too large a current.

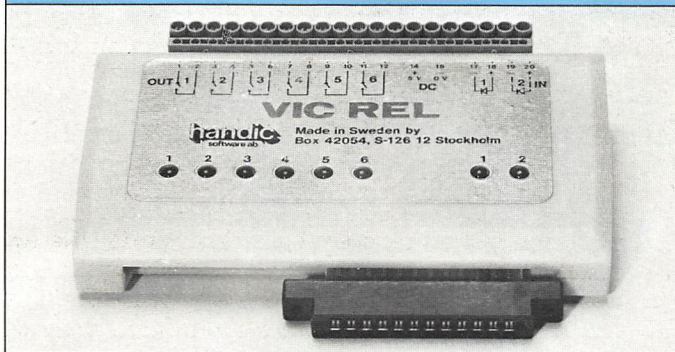
Where the InterSpec and InterBeeb really score over VicRel is with their eight analogue inputs, which can be used with analogue sensors such as microphones and potentiometers. The only project that uses these here is the polar arm, but you can add analogue sensors to many of the others if desired. Some computers have their own analogue input ports specifically for experimentation, and others have inputs for analogue joysticks that could be adapted with appropriate referral to the manual.

Computer control

Programming the relays to open or close and to read the switch or analogue inputs is explained in each system's manual. Programs written on the computer in BASIC will generally access the interface unit through PEEK and POKE statements or their equivalent. We have printed here a very brief summary of the operation of our three choice units, again to assist conversion between them. Lines of BASIC in our printed programs that refer to the interface are few in number – the bulk of each program is concerned with deciding what to control rather than how.



VICREL



Works with Commodore Vic-20 or 64. Provides six independent relays (with two terminals each), two opto-couplers (for linking to switch inputs), 5v and Ground terminals.

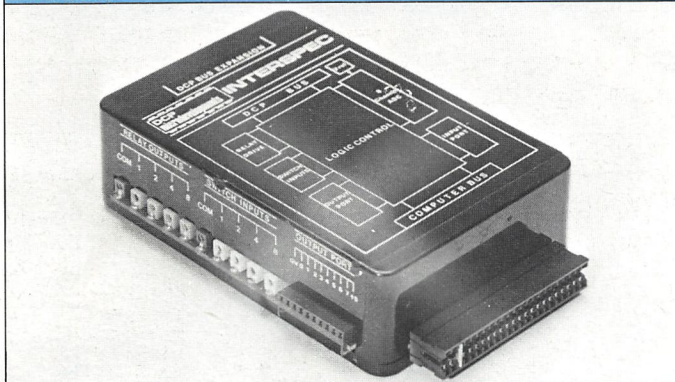
VicRel plugs into the user port and is initialized by POKE 37138,63 (Vic) and POKE 56579,63 (Commodore 64). Relays are controlled and switch inputs read through location 37136 (56577 on the '64) – the lowest six bits controlling the relays, the highest two bits reading the switches. Thus:

POKE 37136,53 will close four of the six relays, because 53 is 110101 in binary.

PRINT PEEK(37136)AND64 will print 0 if the first switch is closed and 64 if it is open.

PRINT PEEK(37136)AND128 will print 0 if the second switch is closed and 128 if it is open.

INTERSPEC



Has same specification as the InterBeeb, except that it takes its power from the Spectrum computer.

To close relays: OUT 63,n where n is in the range 0 to 15. Set the appropriate bit to 0 to open the relay again.

To read switch inputs: PRINT (IN 63)–240 will return a value in the range 0 to 15.

Output port: OUT 95,n where n is in the range 0 to 255.

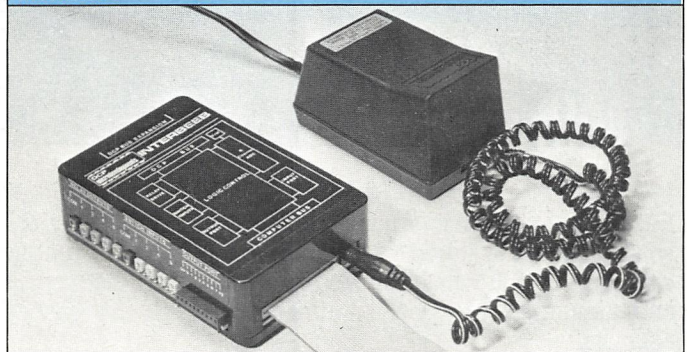
Input port: PRINT IN 95 returns a value in the range 0 to 255.

To read an analogue input: OUT 31,n where n is in the range 0 to 7 corresponding to the channel required. Then PRINT IN 31 returns a value in the range 0 to 255.

IMPORTANT NOTE:

After switching on, all relays must be set to open before any batteries are connected. When the relay block is first switched on any of the relays may be closed – in some circuits this could short-circuit the batteries and cause possible damage. The listings in this booklet contain an initialization routine, so the safest policy is to load and run the program first.

INTERBEEB



Works with BBC Micro and comes with an external transformer. Provides four commoned relays (a total of five terminals), four commoned switch inputs, 8-bit input port, 8-bit output port (these function like a user port), eight analogue inputs which can measure any voltage between 0 and 2.45 volts, plus 5v and Ground terminals.

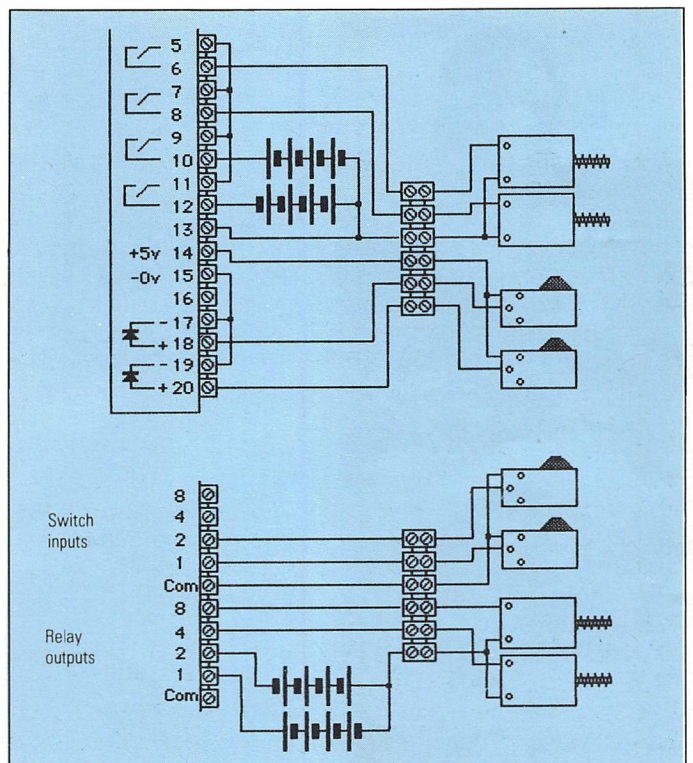
To close relays: ?&FCC1=n, where n is in the range 0 to 15; e.g. 0 opens all relays, 11 closes relays 1, 2 and 8.

To read switch inputs: PRINT ?&FCC1 returns a number in the range 0 to 15, corresponding to the closed switches. 0 means all are open.

Output port: ?&FCC2=n (n is in range 0 to 255).

Input port: PRINT ?&FCC2 returns a value in range 0 to 255.

To read an analogue input, ?&FCC0=n, where n is in the range 0 to 7 corresponding to the input required. PRINT ?&FCC0 will then return a value in the range 0 to 255 proportional to the voltage being read.



Wiring. The diagrams show an identical circuit (the one for the buggy built from Fischertechnik) wired to the two types of interface block. On the VicRel, the four relays are connected together with an external common wire. On the InterSpec/InterBeeb the relays are commoned internally. The same is true for the switch inputs. Note also that the switch inputs on the VicRel must be driven from the 5v supply, whereas the other units merely require a connection between the input line and the common line.

BUGGIES IN FISCHER-TECHNIK AND LEGO

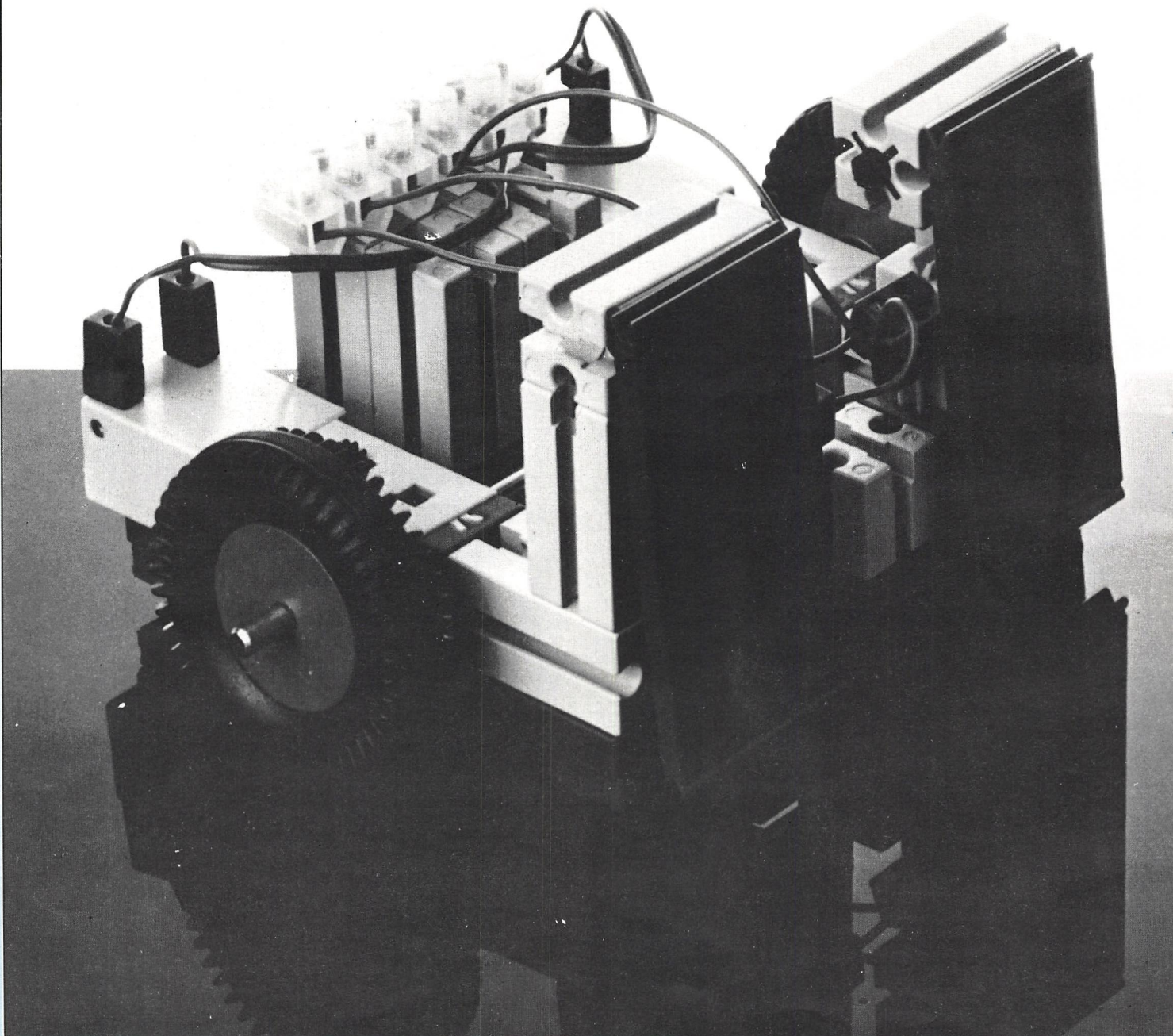
Floor buggies are the simplest type of robot device to design, build and interface to a home computer. A floor buggy is a small mobile unit linked to a computer by means of an umbilical cord through which the motors receive their instructions and their electrical power – unless batteries are carried onboard.

Most buggies feature two main wheels, independently

driven by motors, and the chassis is balanced with either a third, castor, wheel or a simple peg that drags along the ground. Buggies are designed to move around on smooth surfaces, and few will function correctly on a carpet.

The buggies can move in a straight line forwards or backwards, or they can be made to turn on their axis by driving the two wheels in opposite directions. The movement can be controlled in live mode, by pressing keys on the keyboard, or a movement routine can be built up in the computer's memory and then executed at the touch of a button.

Collision detectors mounted on the front or on all four corners of the buggy can send signals back to the computer whenever the vehicle bumps into an obstacle. In this way, the buggy can be programmed to explore a room and build up an



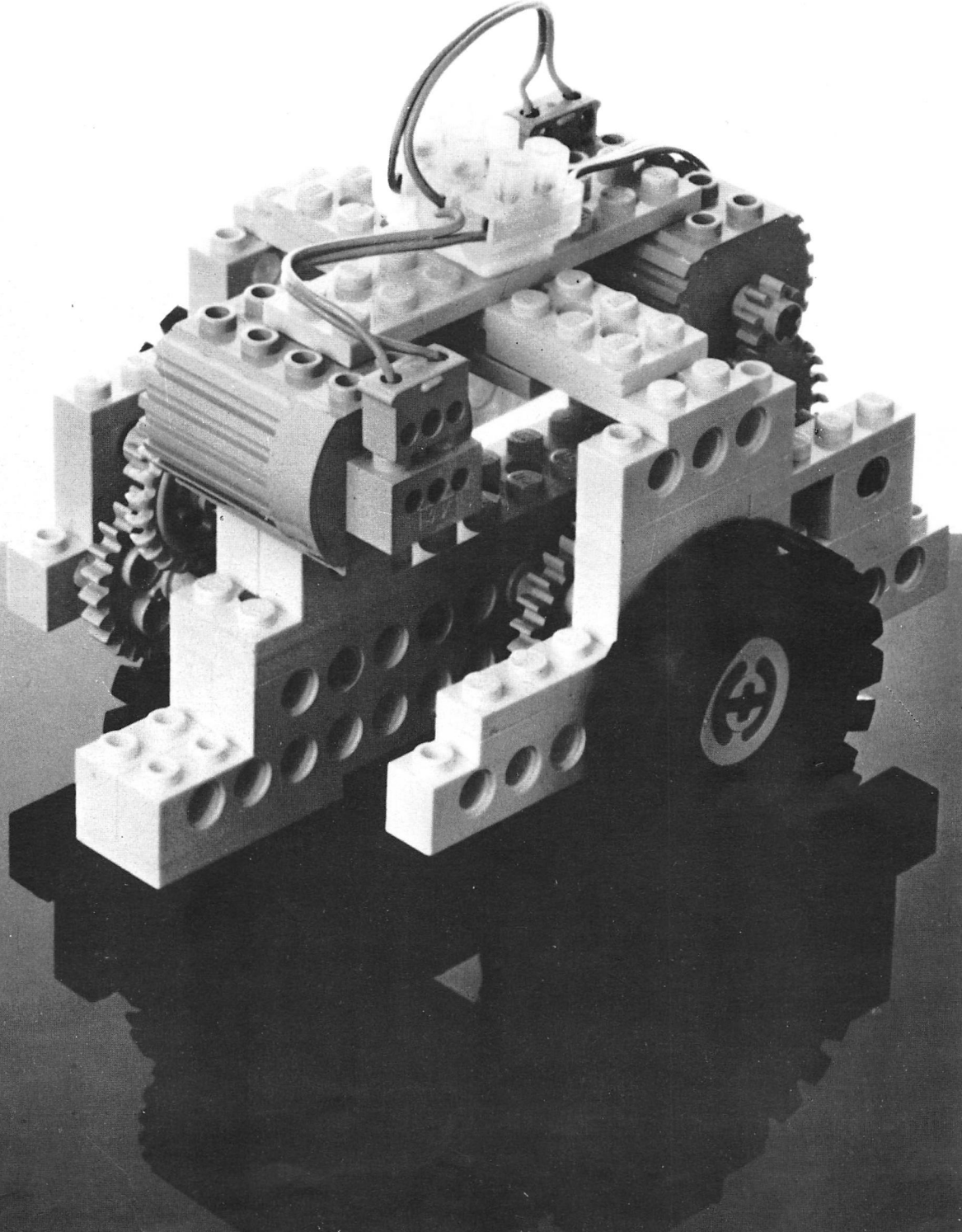
appropriate map in the computer's memory. Other types of sensor that can be fitted to the chassis include ultrasonic rangefinding units for detecting obstacles before the vehicle hits them, and light detectors that enable the buggy to follow lines drawn on the floor.

This project shows how to construct a very simple floor buggy from either Fischertechnik or Lego. The two designs can be driven using the same interface and program, though the Fischertechnik buggy is also equipped with two collision detectors made from microswitches. Using our simple program as a starting point, you can write your own applications for the buggies.

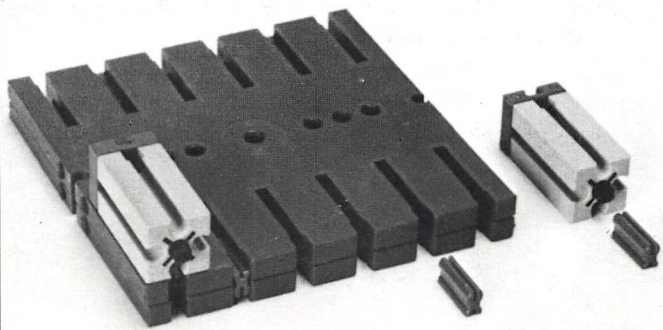
Be warned, however, that these buggies do not move as accurately as commercially available floor robots. Unless you

are lucky, the two motors will not run at identical speed, so the buggy won't move in a true straight line. Furthermore, the only way to control the distance these buggies move is by activating the motors for a known time. Floor robots costing £100 or more employ electronic feedback to obtain precise movements – it is possible for these buggies to draw fine patterns on the floor by fitting a pen between their wheels.

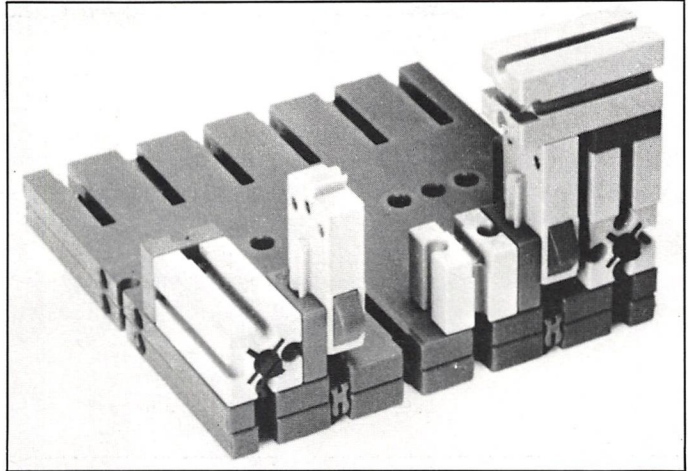
The Motor Control project on pages 16–17 shows how you can incorporate simple feedback into Fischertechnik models and also control the speed at which the motors run. You could therefore go on to design a more sophisticated buggy incorporating both these principles. But for the moment, the two models shown here can be built very cheaply and provide a lot of fun.



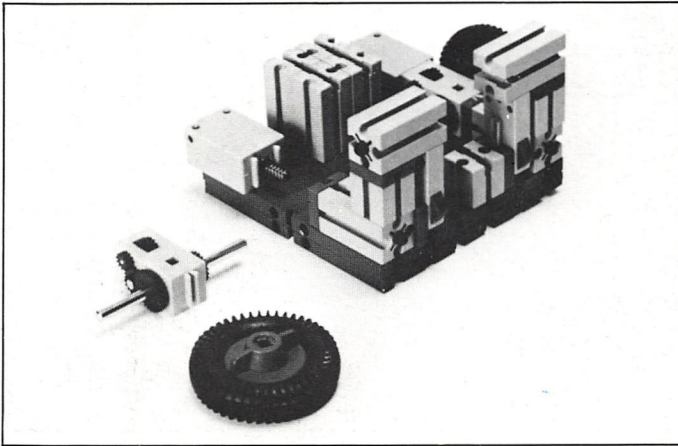
FISCHERTECHNIK BUGGY CONSTRUCTION



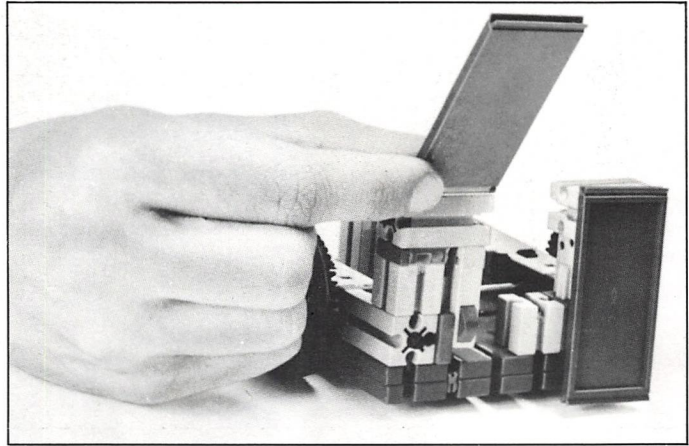
1 Two of the square, red baseboards, one atop the other, form the chassis of this buggy; they are pinned together at the front and back. This gives the vehicle more weight and minimizes the ground clearance so that the buggy doesn't wobble when it changes direction.



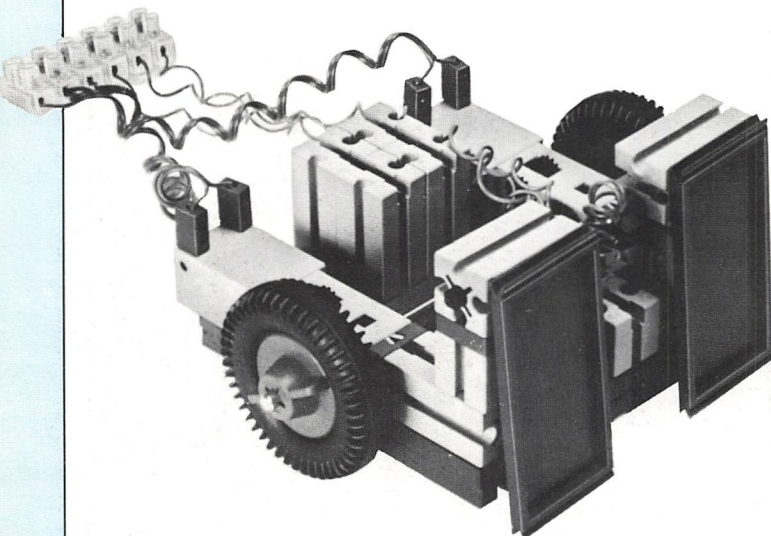
2 Microswitches are used as collision detectors since they require very little force to close the contacts. Because the microswitches feature the Fischertechnik fitting on one side only, the left and right detectors are mounted in different ways.



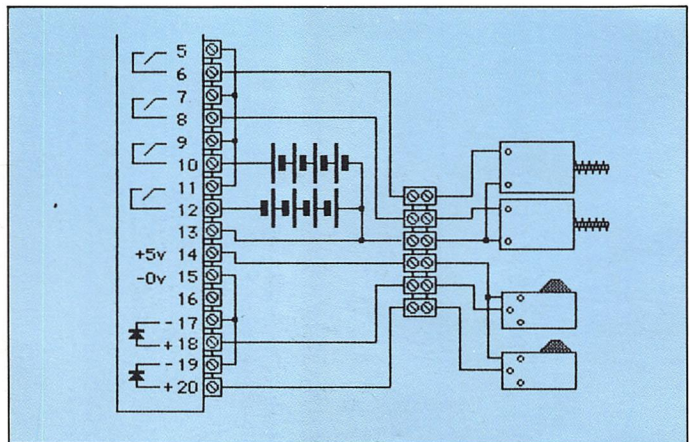
3 Four standard blocks are fitted to weight down the rear. The minimotors slide into the baseboard from the rear, and then each gearbox can be mounted on the motor shaft from the top. The driving wheels need the rubber rings fitted to them for extra grip.



4 Two red plates hang in front of the microswitches to act as bumpers. The most convenient way to mount them is with a piece of sticky tape, acting as a hinge. Check that the switches close when the bumpers are pressed.

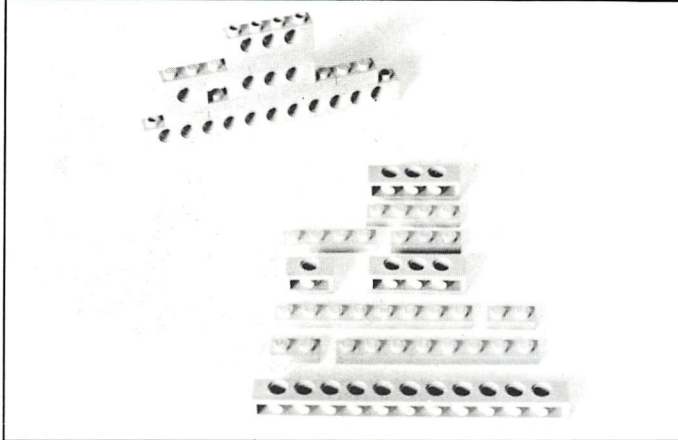


5 The two motors and two switches are wired to an electrical block, which can then be stuck onto the chassis. Ribbon cable or multicore wire, preferably at least 3m (10ft) in length, can be used for the umbilical link to the computer.

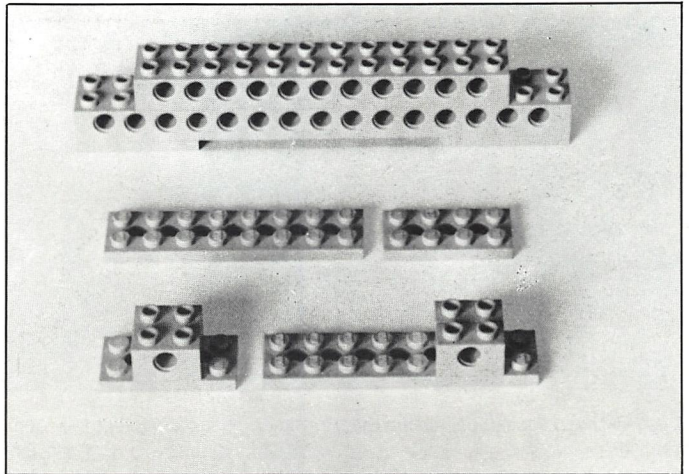


Fischertechnik wiring. Using the VicRel's six relays, it is possible to run the two motors in both directions using just a single set of batteries. Two sets have been used in this circuit so that the buggy can be driven using other interface blocks that feature just four relays. If when connected up the buggy moves forwards or backwards when it should turn, or vice versa, swap over the two wires on one or both of the motors to make them spin in the correct direction.

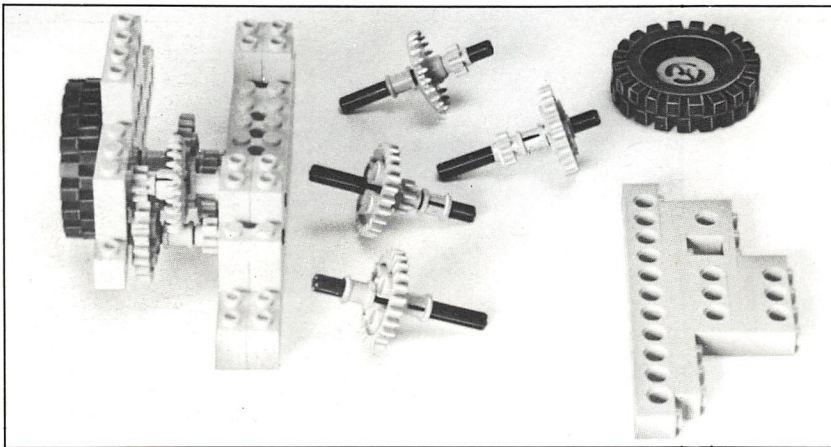
LEGO BUGGY CONSTRUCTION



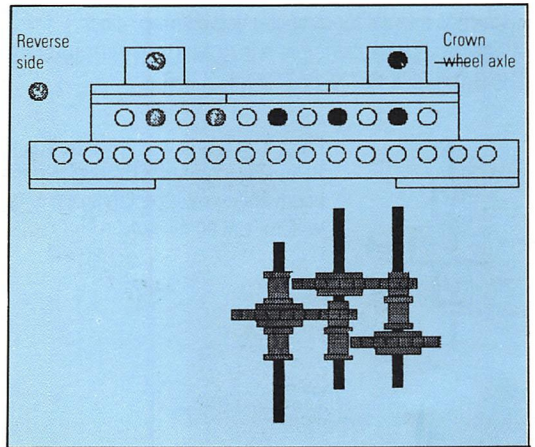
1 This buggy is symmetrical in design. Two identical side supports are constructed first, using single-thickness beams and plates.



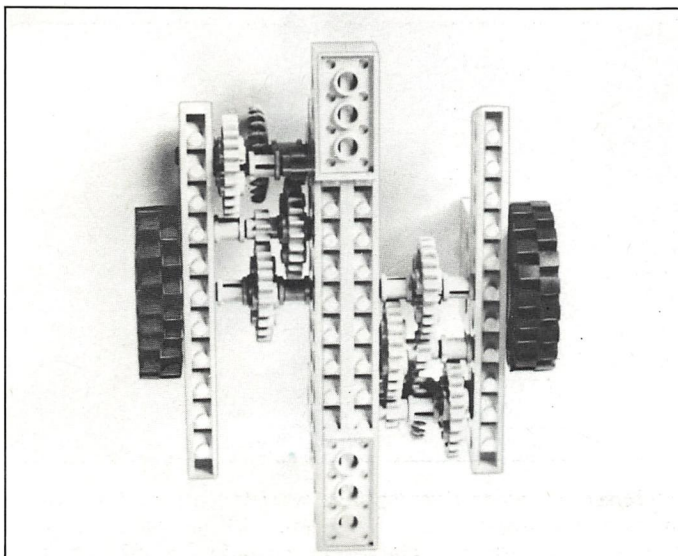
2 The central support is built in three stages. The two thin plates on the bottom of the support are to reduce ground clearance.



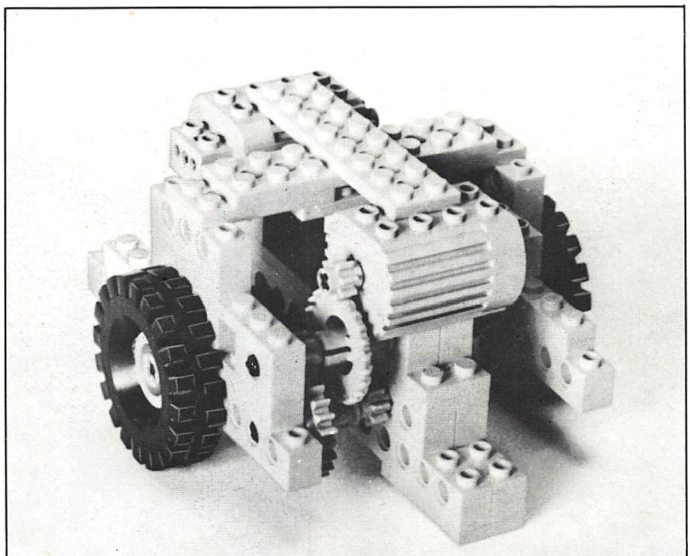
3 Three of the four axles on each side of the buggy fit into holes in the bottom of the side supports and in the middle row of the centre support. The axle with the crown wheel is located in the next row up.



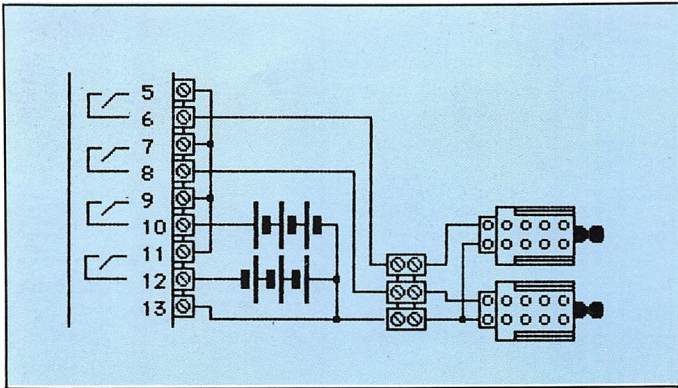
4 This diagram shows the location of three of the axles on each side of the centre support, in the 11-hole beam. The fourth axle sits on the row above and contains a crown wheel to mesh with the motor.



5 The chassis complete and wheels fitted to the central axles, from below. The crown wheels are therefore partially hidden.



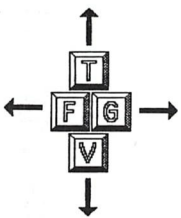
6 The motors, mounted fore and aft, engage on the crown wheels, and two cross-members then hold the structure together.



Lego wiring. Lego motors require three batteries per set instead of four, and there are no microswitches in the circuit. You could construct your own collision detectors using simple push-button switches, or even two pieces of wire that when bumped make contact with one another.

Program. This was written for the Commodore Vic-20 with the VicRel interface unit. It will run on the Commodore 64, too, using the modifications shown. For other computers, re-write each of the subroutines with reference to the instruction manual for computer and interface block.

WARNING: LOAD AND RUN THE COMPUTER PROGRAM BEFORE INSERTING THE BATTERIES IN THE BATTERY HOLDER.

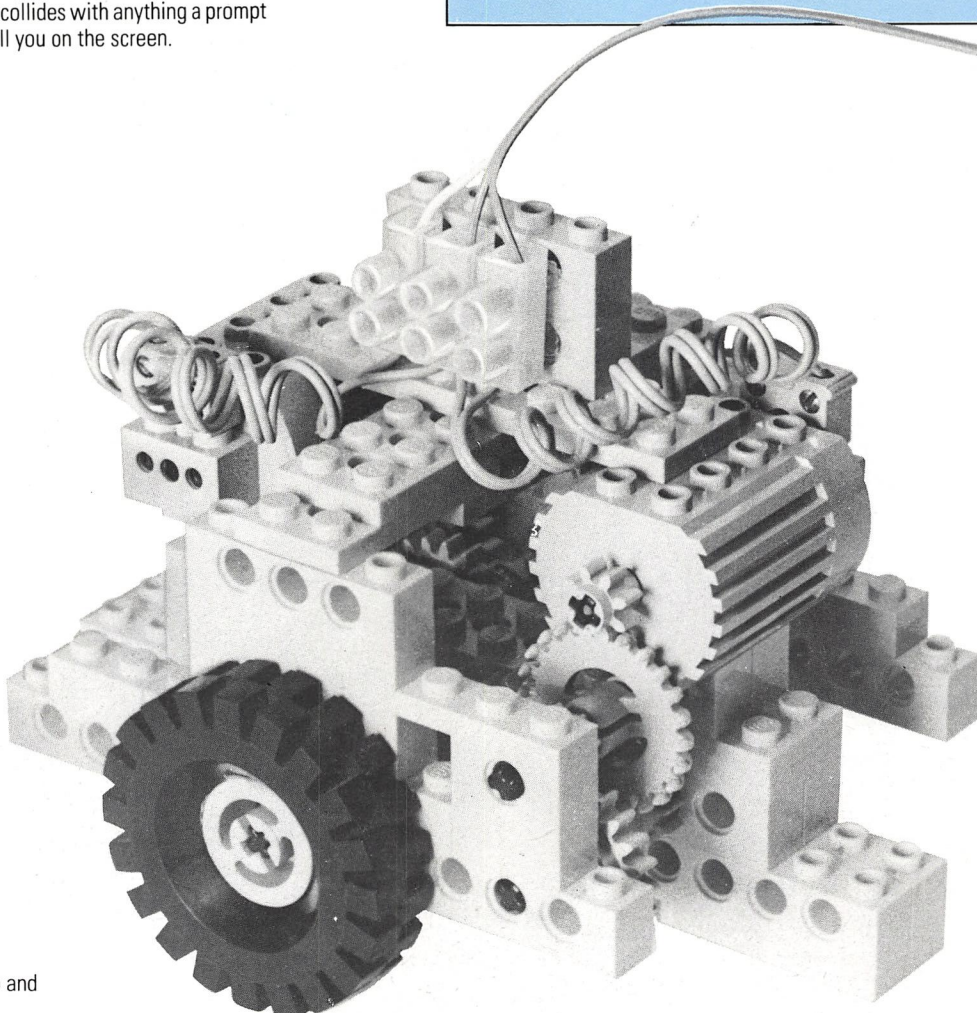


Press T, F, G and V to drive the buggy. Releasing a key stops the buggy. If the buggy collides with anything a prompt will tell you on the screen.

```

10 REM**BUGGY CONTROLLER**
20 REM**INITIALIZE**
30 CN=37136:POKECN+2,63
40 PRINT"INSERT BATTERIES"
50 REM**KEYBOARD VALUES**
60 KA=197:FK=50:BK=27:LK=42:RK=19
70 REM**CONTROL VALUES**
80 FW=28:RV=44:LF=20:LR=36:RF=24:RR=40
90 LB=64:RB=128
92 LH$="LEFT HAND COLLISION"
93 RH$="RIGHT HAND COLLISION"
100 REM**KEYBOARD CONTROL**
110 A=PEEK(KA)
120 IFA=FKTHENGOSUB200
130 IFA=BKTHENGOSUB300
140 IFA=RKTHENGOSUB400
150 IFA=LKTHENGOSUB500
160 IFA=255 THENPOKECN,0
170 GOTO110
200 REM**DRIVE FORWARDS**
210 POKECN,FW:GOSUB600
220 RETURN
300 REM**DRIVE BACKWARDS**
310 POKECN,BW:GOSUB600
320 RETURN
400 REM**TURN RIGHT**
410 POKECN,RR:GOSUB600
420 POKECN,LF:GOSUB600:POKECN,0:RETURN
500 REM**TURN LEFT**
510 POKECN,LR:GOSUB600
520 POKECN,RF:GOSUB600:POKECN,0:RETURN
600 REM**TEST BUMPERS**
610 IF(PEEK(CN)ANDLB)=0 THENPRINTLH$:GOSUB700
620 IF(PEEK(CN)ANDRB)=0 THENPRINTRH$:GOSUB700
630 RETURN
700 REM**STOP+SLIGHT DELAY**
710 POKECN,0:FORI=0TO50:NEXTI:RETURN

```



The complete buggy, wired up and ready to go.

HARMONIC DRIVE SIMULATION

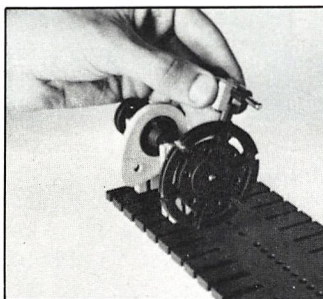
Electric motors used in industrial robots are geared down by a ratio of several hundred to one, to both reduce the speed of rotation and increase the torque (turning force) available to do useful work. The gearing mechanism can be achieved using several stages of conventional gears, but each stage introduces an element of free play, called backlash, which is very undesirable for precise control.

Conceivably the gearing down could be done in just one stage, using one very large cog and one very small one. However, the drive then becomes asymmetrical: the output shaft is not in line with the input shaft. This is awkward for many robot structures. Also, at the join between the two cogs only a few teeth mesh, which reduces the maximum force that can be transmitted.

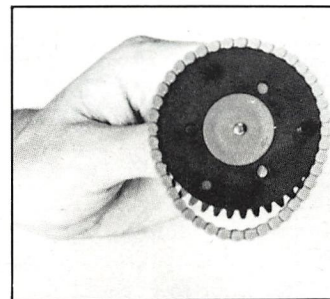
The harmonic drive is an elegant piece of design that employs just three components – two moving, one static – to gear down by as much as 300 to 1. It introduces very little backlash and, because many teeth are meshed together, it can bear very high loading. Four harmonic drives were used on the independently powered wheels of the Apollo lunar rover, and currently they form an integral part of most electrically powered robots.

The drive is precision engineered, and cannot be emulated by the home constructor. The project shown here, therefore, only demonstrates the broad principles on which the harmonic drive works. It uses Fischertechnik.

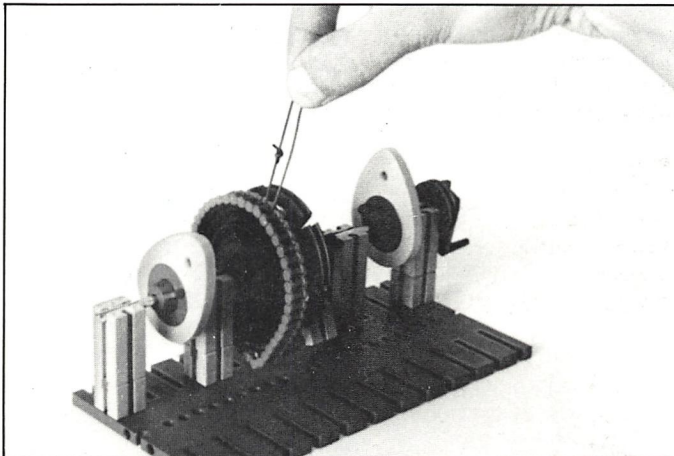
The three components of a harmonic drive are a circular spline, a flexible spline and a wave generator. In our model, the flexible spline (the chain) has 42 links, and is prevented from rotating by the loop of string. The circular spline has 40 teeth and is affixed to the output shaft. As the input shaft is turned, the wave generator moves round between the circular and flexible splines. Now as the flexible spline can't rotate, the



1 The stub axle is mounted on the large red wheel in such a way that it can be adjusted to take up the slack in the chain.

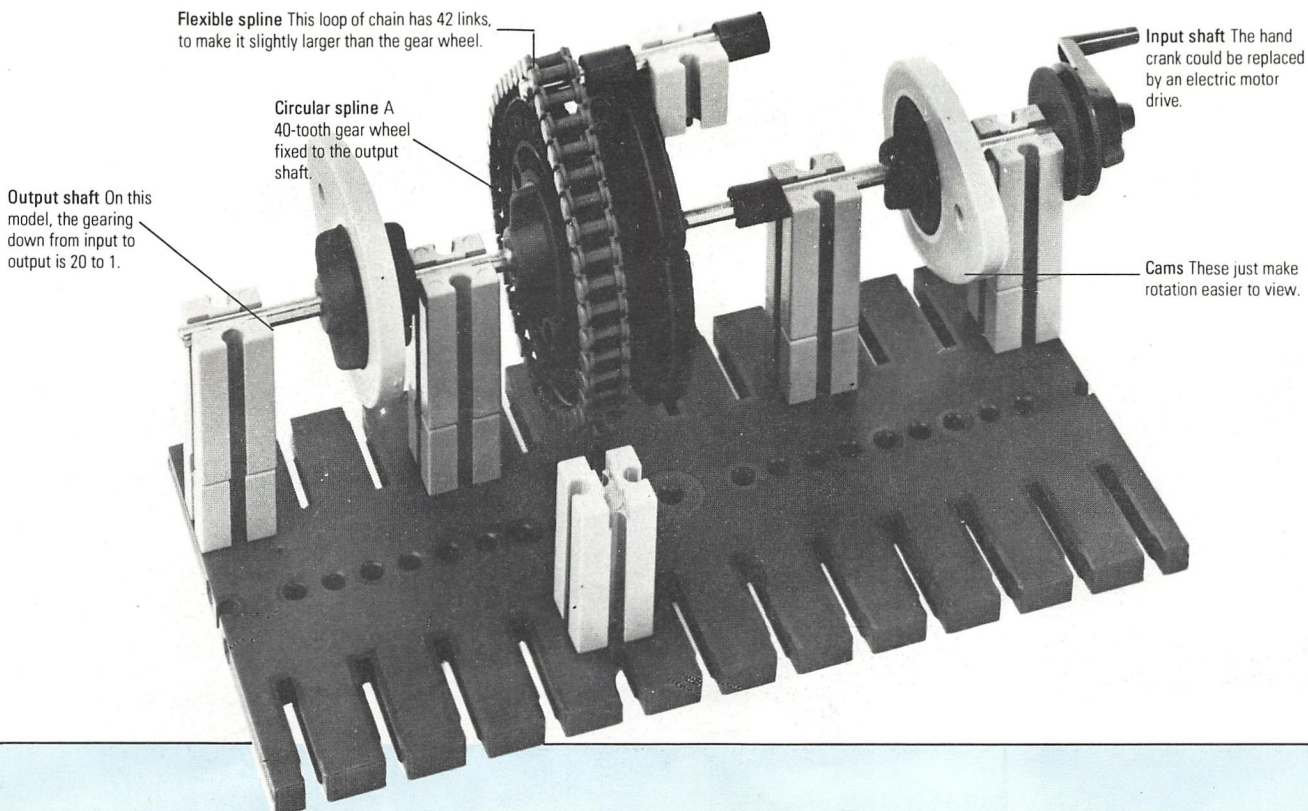


2 Mounted on the gear wheel, the chain should hang away from the lower teeth, forming an egg shape.



3 A loop of string is tied through one link of the chain, then secured to a block on the baseboard. Adjust the tension by sliding the block so that the chain is not pulled away from the gear wheel.

circular spline is forced to turn slowly backwards – to keep up with the number of teeth. When the input shaft has turned through one revolution clockwise, the output shaft will have turned anticlockwise by two teeth (42–40).



Flexible spline This loop of chain has 42 links, to make it slightly larger than the gear wheel.

Circular spline A 40-tooth gear wheel fixed to the output shaft.

Output shaft On this model, the gearing down from input to output is 20 to 1.

Input shaft The hand crank could be replaced by an electric motor drive.

Cams These just make rotation easier to view.

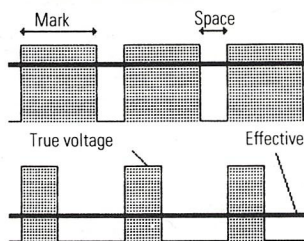
MOTOR CONTROL

Though a small electric motor can be linked to a home computer using a relay, and switched on and off by a single BASIC command, building useful robot devices requires far more sophisticated control. The robot engineer must be able to control both the speed at which the motor turns and the position where it stops. This can be achieved using dedicated electronic circuitry, but this project demonstrates that both types of control are possible with a standard interface block and some clever programming.

The photographs show how to build a simple winch mechanism. The objective is to control the rate at which string is wound in and to stop the drum at any desired position – verified by the movement of a weight hanging on the end of the string. The winch itself could be built directly into a model crane, but the principles demonstrated can be incorporated into any small robot device, and indeed form the basis of the more sophisticated projects in this book.

Fischertechnik has been chosen for the design because the kit has its own microswitches. The speed control part of this project will work equally well with a Lego or any other small electric motor. Then the only change needed in the wiring diagram is to fit the correct number of batteries for the motor. To achieve position control with Lego, you will need to incorporate a non-Lego microswitch, several types of which are readily available from radio spares suppliers.

The chosen method of speed control is called pulse width modulation, or voltage chopping. This means that the voltage supplied to the motor is switched on and off many times a second so that the power arrives as a rapid chain of pulses.



Pulse width modulation. The diagram shows how the effective voltage supplied to the motor can be increased or decreased by modifying the length of time the power is left on and off. Long pulses of power at short intervals (above) produce the highest effective voltage.

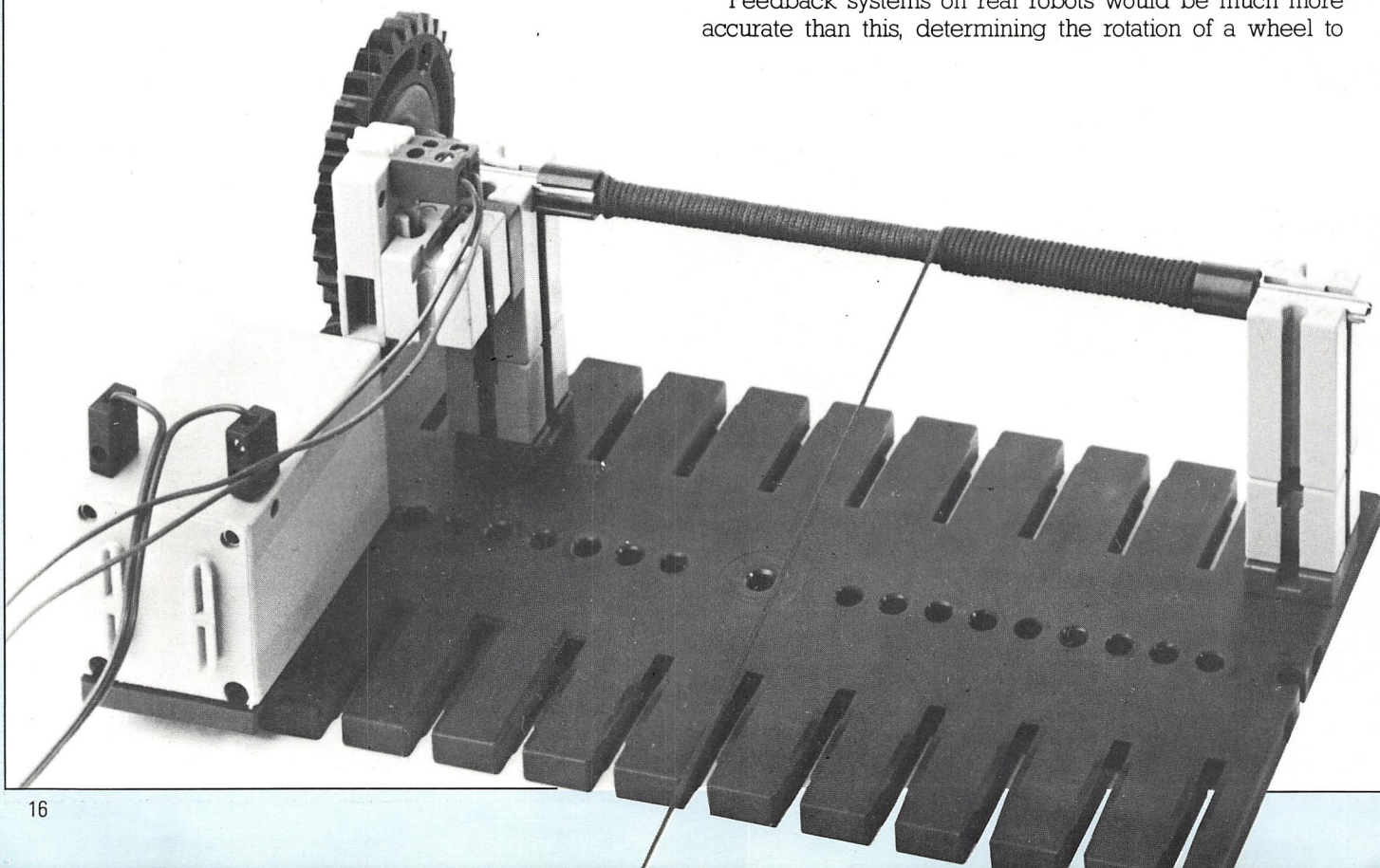
The speed of the motor will be determined by the ratio of the time for which the power is switched on to the time it is switched off in each pulse. In technical terms this is called the mark/space ratio, but it is best understood by looking at the diagram.

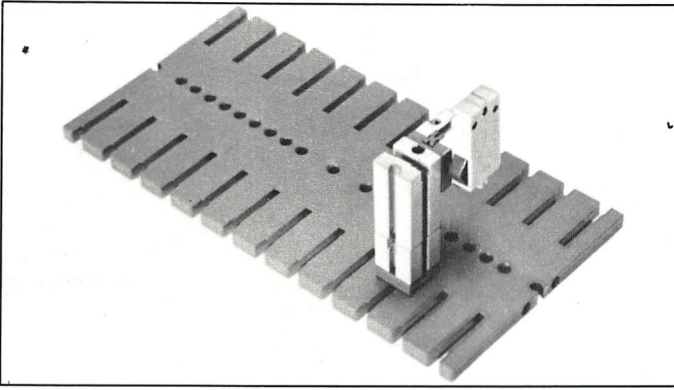
Voltage chopping has its limits – if you try to run the motor very slowly it will simply stall. Performance can be improved by placing a large capacitor in parallel with the motor (assuming you are familiar with elementary electronics). Most relay blocks are quite capable of switching on and off as fast as this experiment requires, but it is a good idea to consult the instructions first.

Programming on a home computer to achieve voltage chopping is a simple matter, as our sample program shows. The pulsing is controlled by a main loop containing two delay loops that determine the duration for which the relay is kept closed and then held open. Try running the speed control program and see what range of speeds you can achieve for winding the string in and out again.

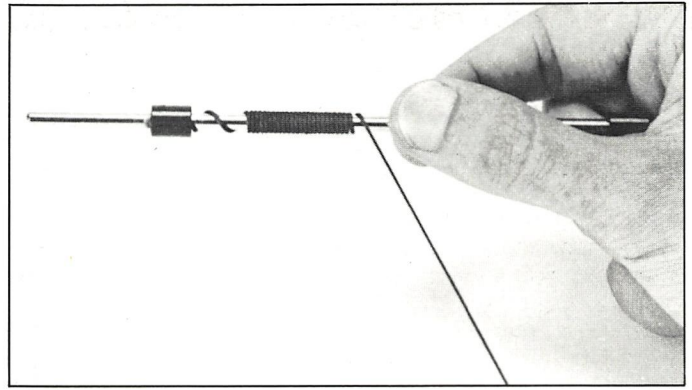
The position at which an electric motor halts can be accurately determined only by using feedback, which means sending signals back from the winch to the computer. This requires some form of electronic sensor mounted on the winch, and our design uses a microswitch. As the winch rotates, the wing-nut on the driving cog depresses the microswitch every half revolution. Each closure of the switch corresponds to the string advancing about 7.5mm ($\frac{3}{16}$ in). The computer simply counts the pulses and compares the distance moved with the distance required.

Feedback systems on real robots would be much more accurate than this, determining the rotation of a wheel to

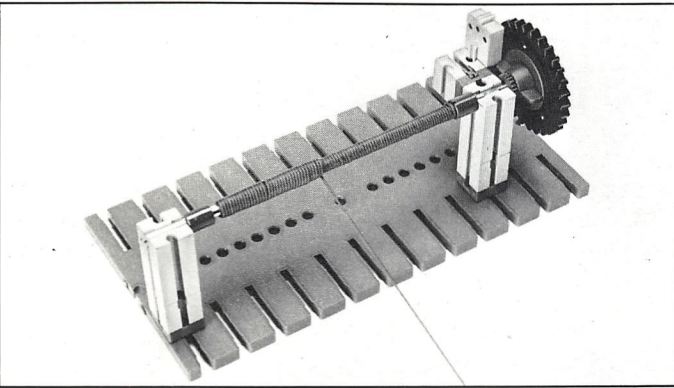




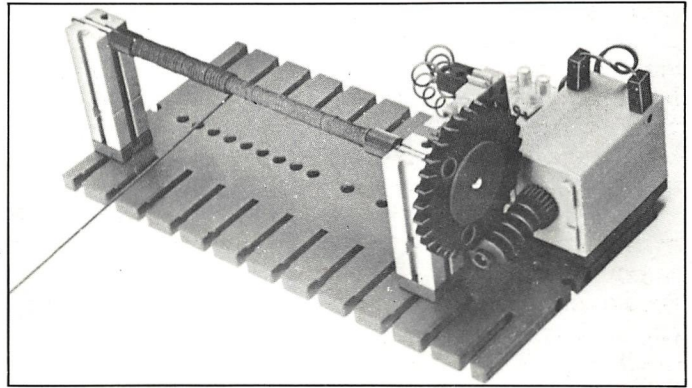
1 Mount the microswitch on the baseboard atop a support made from one and a half standard blocks plus two thin red fillers. When the model is complete the position of the microswitch can be finely adjusted.



2 Secure the string onto the axle with a single red collar, then neatly wind on a further 1m (3ft) of string. Note that if a second layer of string is formed, the effective diameter of the drum changes.



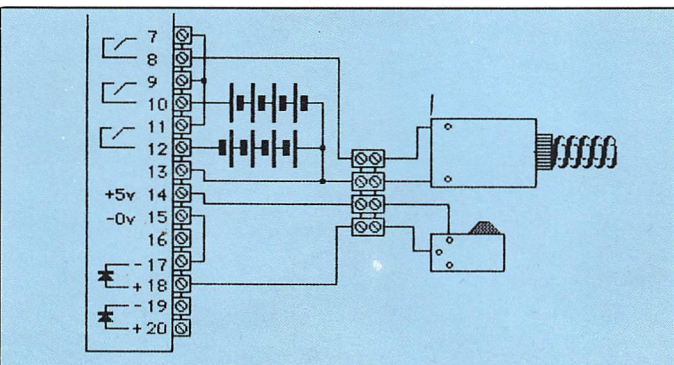
3 Fasten the axle into its two bearings, then affix the driving gear. Make sure that the gear grips the axle very firmly and that when the gear wheel rotates the microswitch is depressed each half-turn.



4 Wire up the model as in the wiring diagram, using coiled cable and a connection block for neatness. Attach a weight (such as an apple) to the free end of the string with a secure knot.

within a single degree or less. In place of our mechanical switch would be a light beam and light detector that counts the passing of holes or teeth on the wheel, a mechanism that eliminates wear and tear.

Run the feedback program and see how accurately it can stop the winch. If you try writing a simple program that activates the motor for a specified time, but that doesn't employ the feedback switch, you will find that this is a lot less accurate. As a final exercise, combine the two themes of this project and write a program that gradually accelerates the motor from rest and then slows it down automatically as the weight approaches the desired position.



Wiring diagram. Two sets of batteries are used to allow the motor to be run in both directions. With the six-relay VicRel it would be possible to achieve the same effect using just one set, but the circuit shown here will work with most relay units. For a Lego motor use only three battery cells per set.

WARNING: LOAD AND RUN THE COMPUTER PROGRAM BEFORE INSERTING THE BATTERIES INTO THE BATTERY HOLDER.

Program 1

```

10 REM**SPEED CONTROL**
20 REM**INITIALIZATION**
30 CN=37136:POKECN+2,63:F=1
40 PRINT"INSERT BATTERIES"
50 PRINT"PRESS 0-9 FOR SPEED CONTROL"
60 PRINT"AND SPACE TO REVERSE"
70 REM**CONTROL**
80 GETA$:IF A$="" THEN 120
90 IF A$=" " THEN F=-F:GOTO 110
100 T1=VAL(A$):T2=9-T1
110 DR=32+F*8
120 REM**PULSE WIDTH CONTROL**
130 IF T1=0 THEN 80
140 POKECN,DR:FORDL=1TOT1:NEXT
150 POKECN,0:FORDL=1TOT2:NEXT:GOTO 80

```

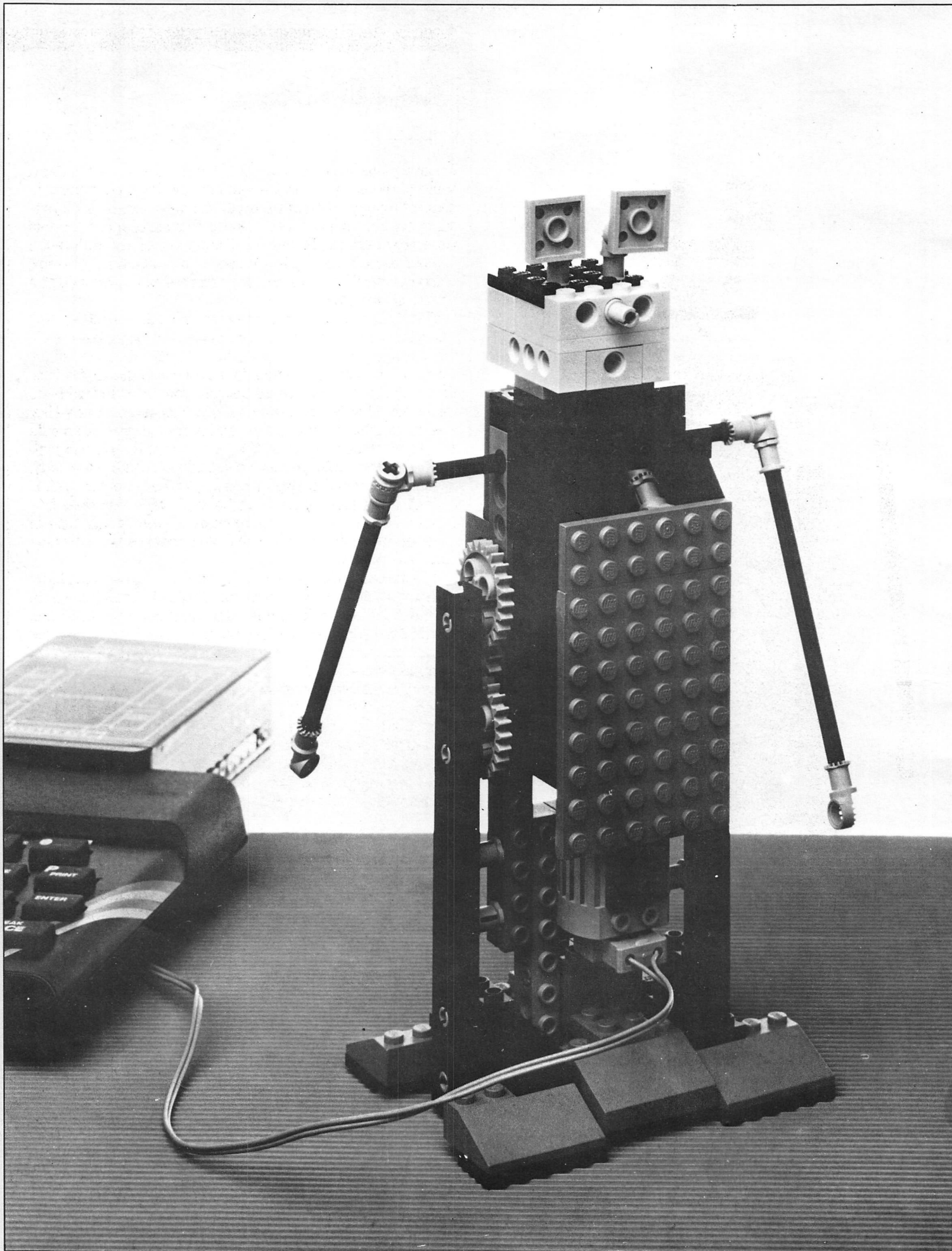
Program 2

```

10 REM**FEEDBACK CONTROLLER**
20 REM**INITIALIZE**
30 CN=37136:POKECN+2,63
40 PRINT"INSERT BATTERIES"
50 DW=40:UP=24
60 INPUT"HOW MANY STEPS";SP
70 IF SP=0 THEN END
80 REM**SET DIRECTION OF MOTION**
90 IF SGN(SP)=-1 THEN POKECN,DW:GOTO 110
100 POKECN,UP
110 REM**COUNT PULSES/STEPS**
120 SP=ABS(SP)
130 IF (PEEK(CN) AND 128)=128 THEN 130
140 IF (PEEK(CN) AND 128)=0 THEN 140
150 SP=SP-1:IF SP>0 THEN 130
160 REM**TURN OFF MOTOR**
170 POKECN,0:GOTO 60

```

Program. Both these programs are written for the Vic-20, but will work with a Commodore 64 by changing the value of CN in line 30 to 56577.



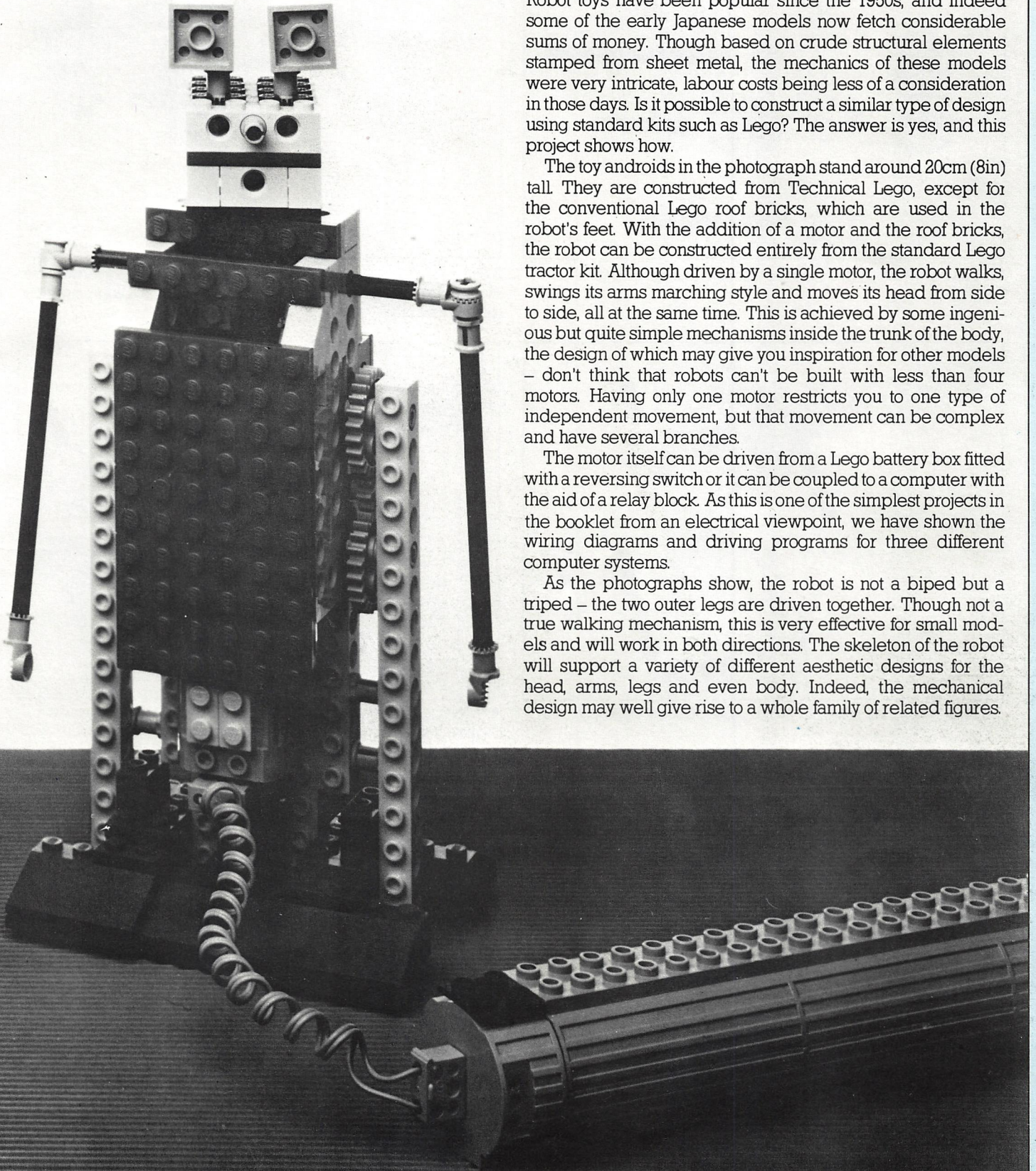
TOY ANDROID

Robot toys have been popular since the 1950s, and indeed some of the early Japanese models now fetch considerable sums of money. Though based on crude structural elements stamped from sheet metal, the mechanics of these models were very intricate, labour costs being less of a consideration in those days. Is it possible to construct a similar type of design using standard kits such as Lego? The answer is yes, and this project shows how.

The toy androids in the photograph stand around 20cm (8in) tall. They are constructed from Technical Lego, except for the conventional Lego roof bricks, which are used in the robot's feet. With the addition of a motor and the roof bricks, the robot can be constructed entirely from the standard Lego tractor kit. Although driven by a single motor, the robot walks, swings its arms marching style and moves its head from side to side, all at the same time. This is achieved by some ingenious but quite simple mechanisms inside the trunk of the body, the design of which may give you inspiration for other models – don't think that robots can't be built with less than four motors. Having only one motor restricts you to one type of independent movement, but that movement can be complex and have several branches.

The motor itself can be driven from a Lego battery box fitted with a reversing switch or it can be coupled to a computer with the aid of a relay block. As this is one of the simplest projects in the booklet from an electrical viewpoint, we have shown the wiring diagrams and driving programs for three different computer systems.

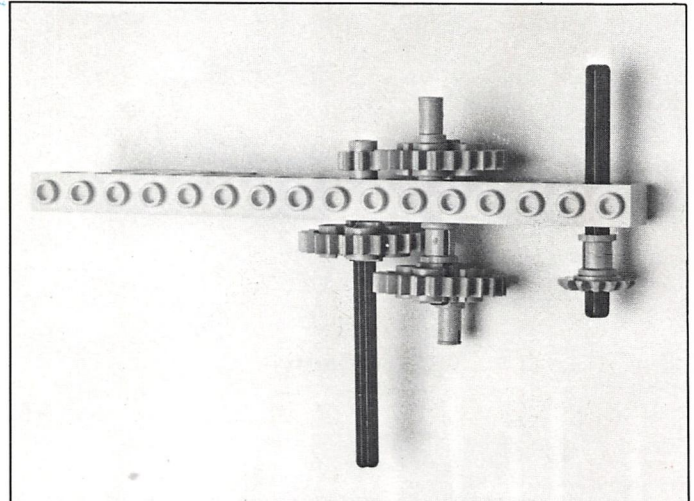
As the photographs show, the robot is not a biped but a tripod – the two outer legs are driven together. Though not a true walking mechanism, this is very effective for small models and will work in both directions. The skeleton of the robot will support a variety of different aesthetic designs for the head, arms, legs and even body. Indeed, the mechanical design may well give rise to a whole family of related figures.



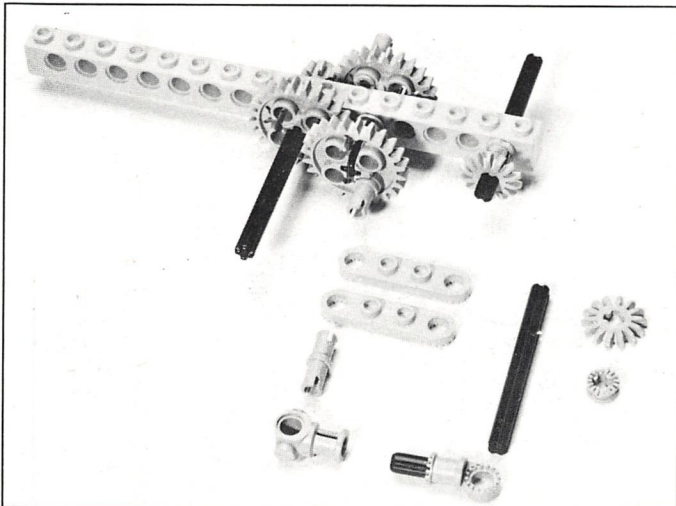
CONSTRUCTION



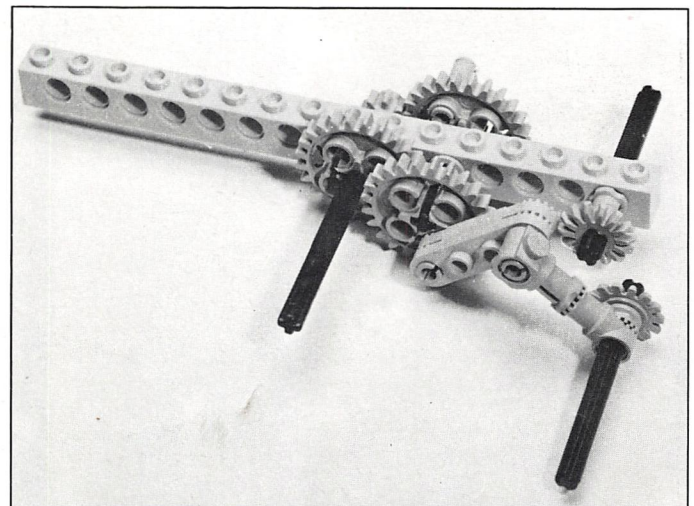
1 Components for the first stage laid out so as to reflect their order of assembly.



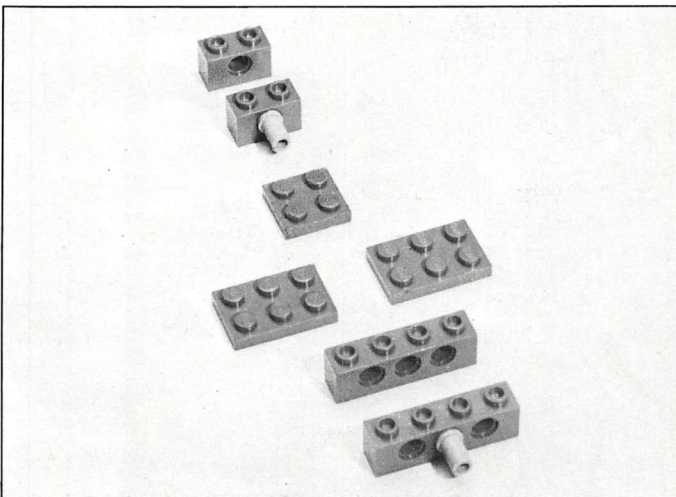
2 Slot the 4-, 6- and 8-unit-long axles into the 16-unit holed beam. Two of the 24-tooth gears have pins in them and are fitted so that the pins are diametrically opposite each other.



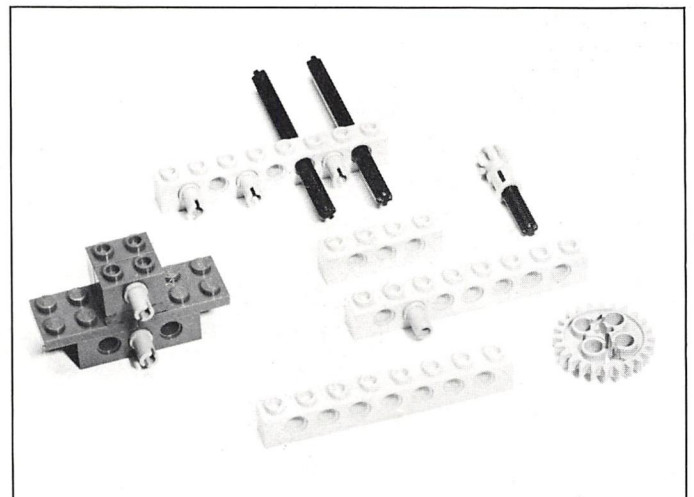
3 Parts normally used by Lego for steering gear are added to the assembly.



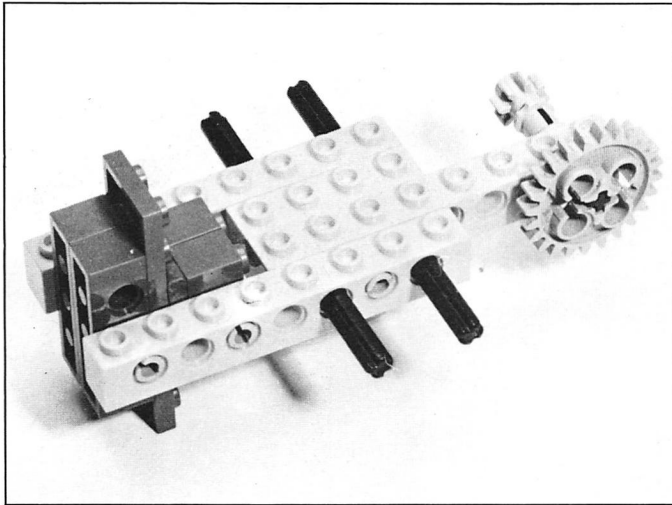
4 When the 24-tooth gear rotates, the new axle with the bevel gear will oscillate. This will move the head and arms.



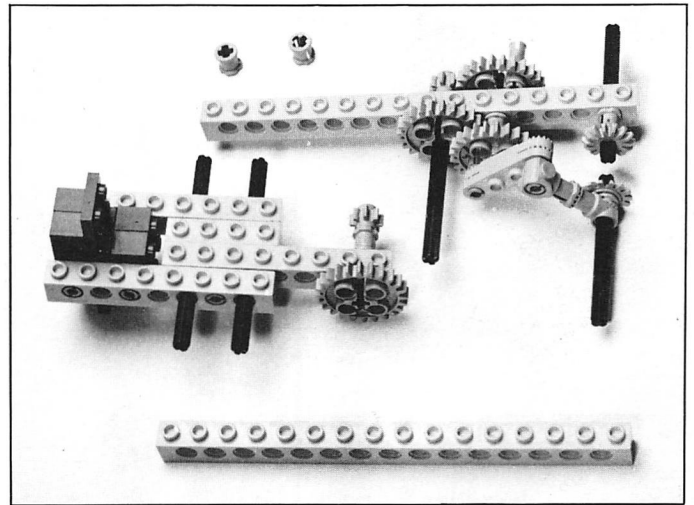
5 The top of the central foot is made from 2- and 4-unit bricks and plates and held in place by pins on either side.



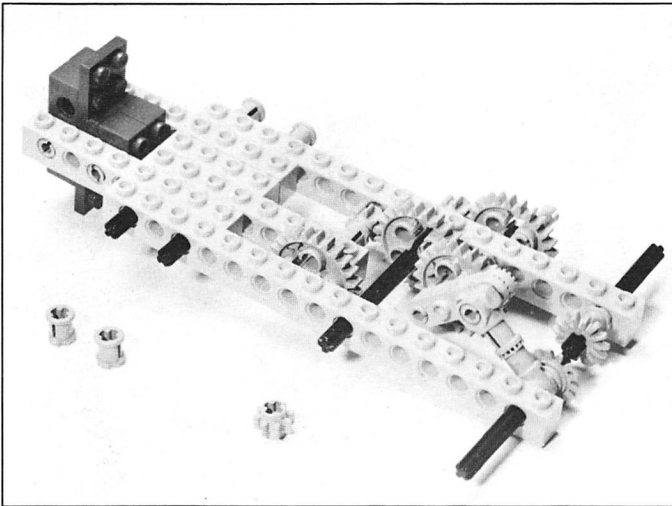
6 The foot lies between 8-unit beams. Two of the axles shown are 8 units long and the one with the two gear wheels is 4 units.



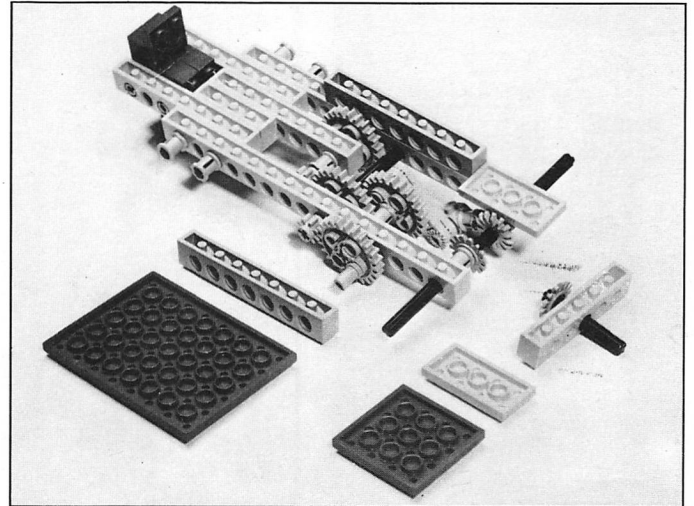
7 Ensure that all the pins shown have been inserted as these are needed for rigidity.



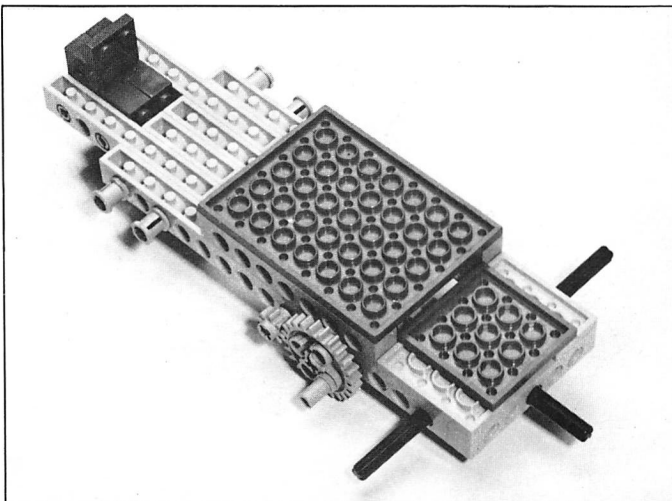
8 The central leg and the oscillating mechanism constructed previously. A second 16-unit beam and two more collars are added next.



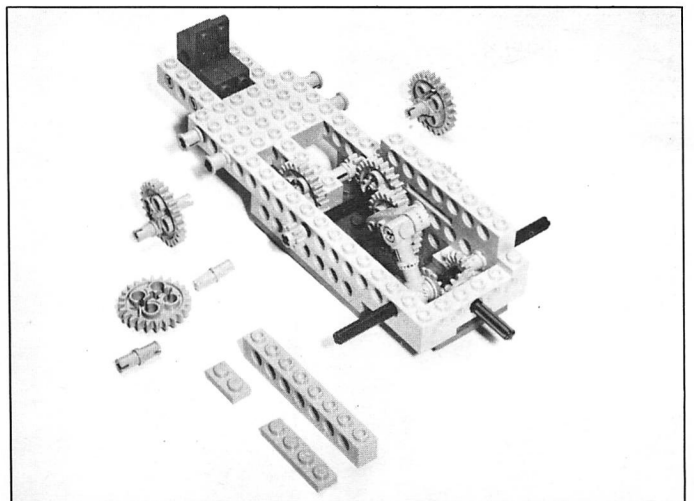
9 The leg and mechanism are connected, the final collars fitted, and an 8-tooth gear added.



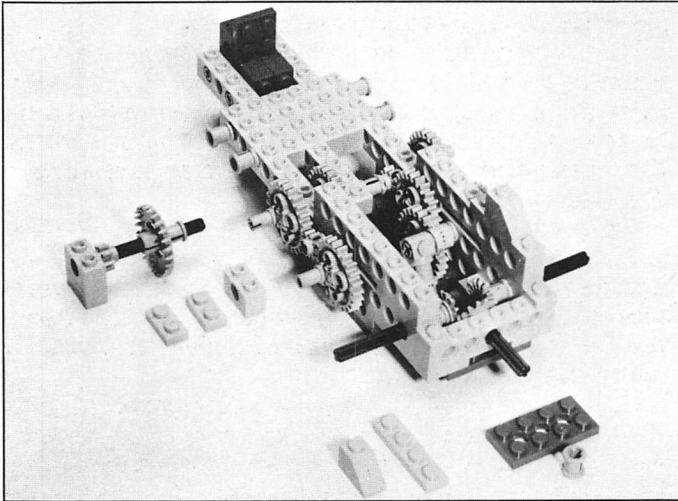
10 Turning the body over, the back is constructed from two 8-unit beams and a 6x8 plate. A 6-unit beam, two 4x2 plates and one 4x4 plate make the shoulders, with a 4-unit axle for the neck.



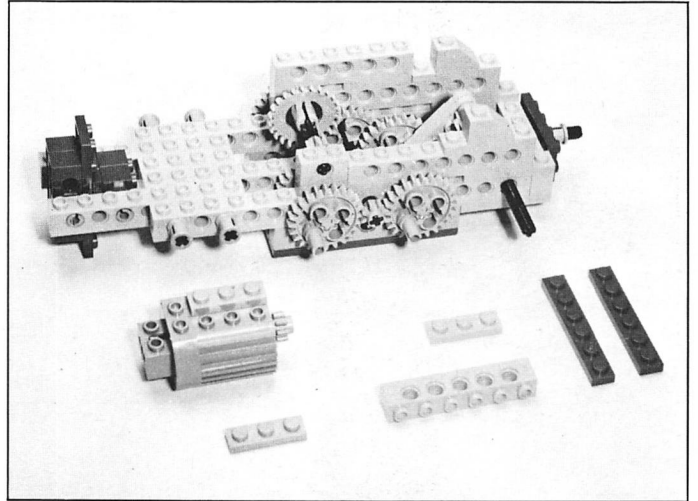
11 The back, shoulder and neck assemblies in place.



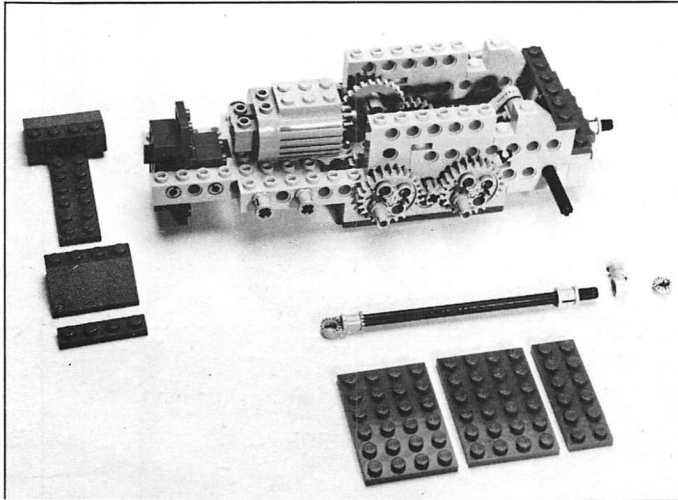
12 Turning back over, thin plates and an 8-unit beam are added to each side, plus the gear wheels shown.



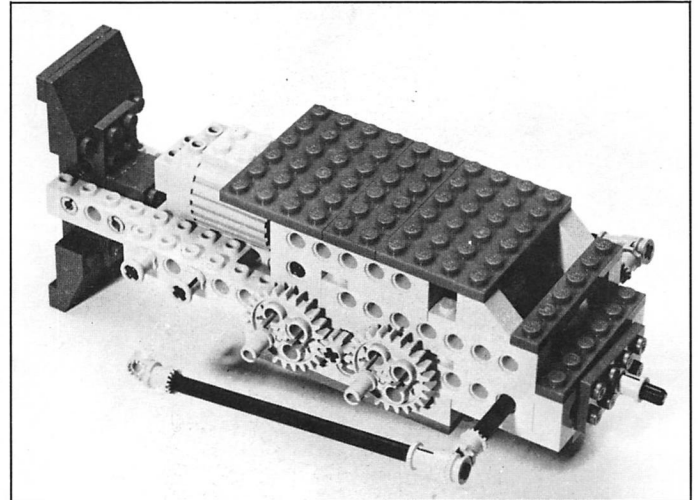
13 The crown wheel sits on a 6-unit axle, supported at each end on a 2×1 block and two plates. The assembly is mounted on the waist. Chest pieces, neck plate and collar are added.



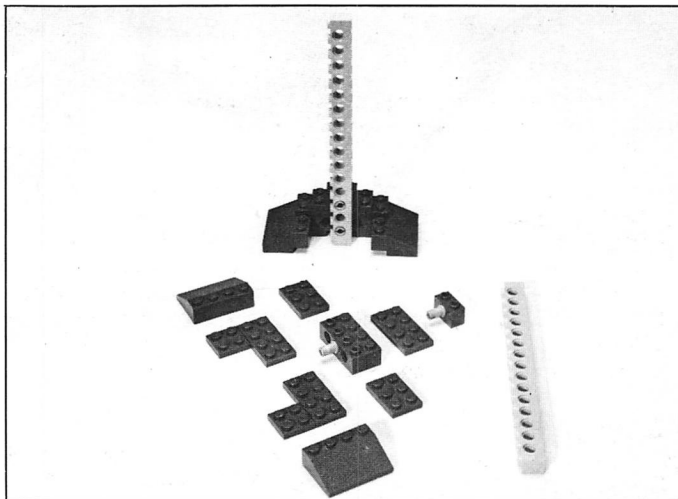
14 The motor features thin plates on top and an 8-tooth gear that engages with the crown wheel. Plates and beams complete the sides.



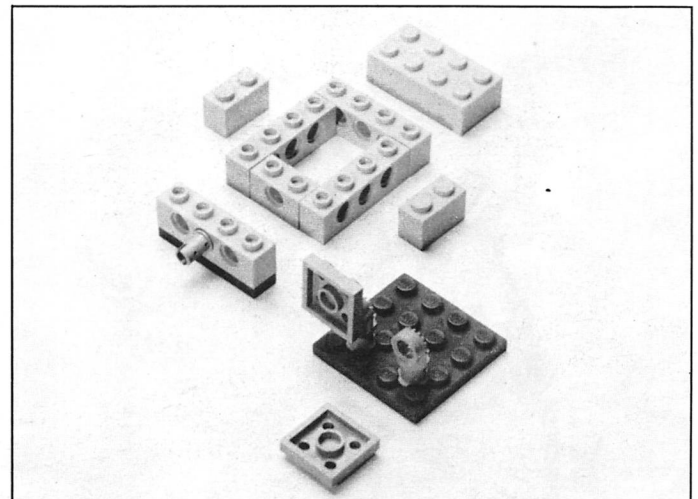
15 The sole of the central foot is made from Lego roof bricks, the arms are 12-unit axles fitted with hinges, and the front of the body comprises three large plates.



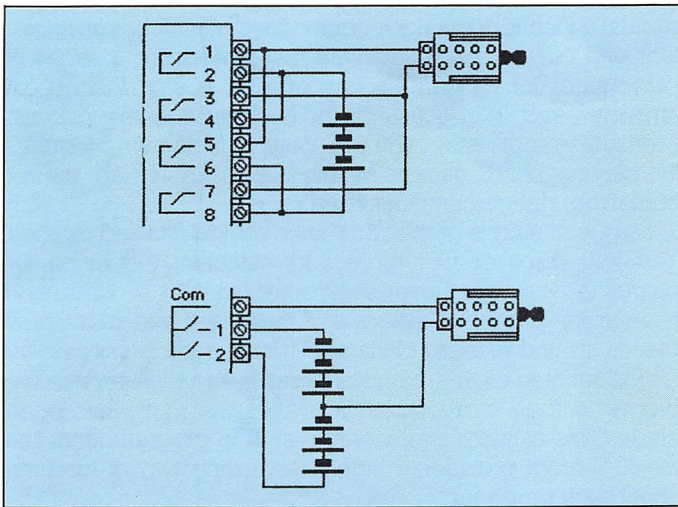
16 The pins on the exterior gears should line up and be in the same position on either side of the body.



17 The two outer feet are constructed in the same way, each from a 16-unit beam, bricks and plates, and are fitted to the legs with two pins.



18 The head can follow more-or-less any design, as long as it clips onto the 4×2 plate on the neck.



Programs and wiring

Shown left are two different ways of wiring the toy android to a computer. You can use these techniques to drive any other Lego model fitted with a single motor.

The first wiring diagram and program are for the Vic-20 computer fitted with a VicRel relay block. Because the relays on the VicRel are all independent, four of them can be wired as shown so that the motor can be driven in either direction. The model is controlled from the keyboard by means of the F (Forward) and B (Back) keys.

The second wiring diagram works for both the InterSpec and InterBeeb units, and it is therefore accompanied by listings for both Spectrum and BBC computers. However, because the four relays on these interfaces are all commoned together, the only way to drive a motor in both directions is to use two sets of batteries, as shown.

All the programs listed here are for simple keyboard operation, but there is no reason why they cannot be expanded to memorize a sequence of moves in an array and then repeat them. Note that all three programs start with a routine that initializes the relays and *then* requests the batteries to be inserted. This is essential to avoid burning out batteries when the computer is switched on.

VIC-20 PROGRAM

```

10 REM**WALKING ROBOT CONTROLLER**
20 PRT=37136:REM**PORT ADDRESS**
30 POKEPRT+2,63:PRINT"INSERT BATTERIES"
40 FW=5:    REM**FORWARD**
50 BW=10:   REM**BACKWARD**
60 SP=0:    REM**STOP**
70 IFPEEK(197)=42THENPOKEPRT,FW:GOTO70
80 IFPEEK(197)=35THENPOKEPRT,BW:GOTO70
90 POKEPRT,SP:GOTO70

```

SPECTRUM PROGRAM

```

10 REM **WALKING ROBOT CONTROLLER**
20 LET prt=63: REM **PORT ADDRESS**
30 OUT prt,0: PRINT "Insert Batteries"
40 LET fw=1:   REM **FORWARD**
50 LET bw=2:   REM **BACKWARD**
60 LET sp=0:   REM **STOP**
70 IF INKEY$="f" THEN OUT prt,fw
80 IF INKEY$="b" THEN OUT prt,bw
90 IF INKEY$="" THEN OUT prt,sp
100 GO TO 70

```

BBC MICRO PROGRAM

```

10 REM**WALKING ROBOT CONTROLLER**
20 PRT=&FCC1:REM**PORT ADDRESS**
30 ?PRT=0 : PRINT"INSERT BATTERIES"
40 FW=1 : REM**FORWARD**
50 BW=2 : REM**BACKWARD**
60 SP=0 : REM**STOP**
70 IF INKEY(-68) THEN ?PRT=FW : GOTO70
80 IF INKEY(-101) THEN ?PRT=BW : GOTO70
90 ?PRT=SP : GOTO70

```

The control program is printed here for the Spectrum, BBC and Vic-20. Once loaded and run, the program prompts you to insert the batteries into the battery holder. Movement is then controlled by pressing F for forwards, B for backwards. The robot stops when the keys are released.

19 On the completed figure, the outer legs should line up and move in synchronization. The motor can be driven from a battery box, as shown here, or from a computer, as shown in the wiring diagrams.

X-Y (DRAUGHTS-PLAYING) ROBOT

Robots constructed around the Cartesian system of coordinates (sometimes therefore called X-Y-Z robots) are less common than the jointed-arm type. They may not fulfil everyone's expectation of what a robot ought to look like, but for many industrial applications they are the most suitable design. The electronics industry uses small-scale X-Y robots (i.e. with only two axes of movement) for inserting the silicon chips into printed circuit boards, and much larger versions looking more like overhead gantry cranes form the basis of fully automated warehouses.

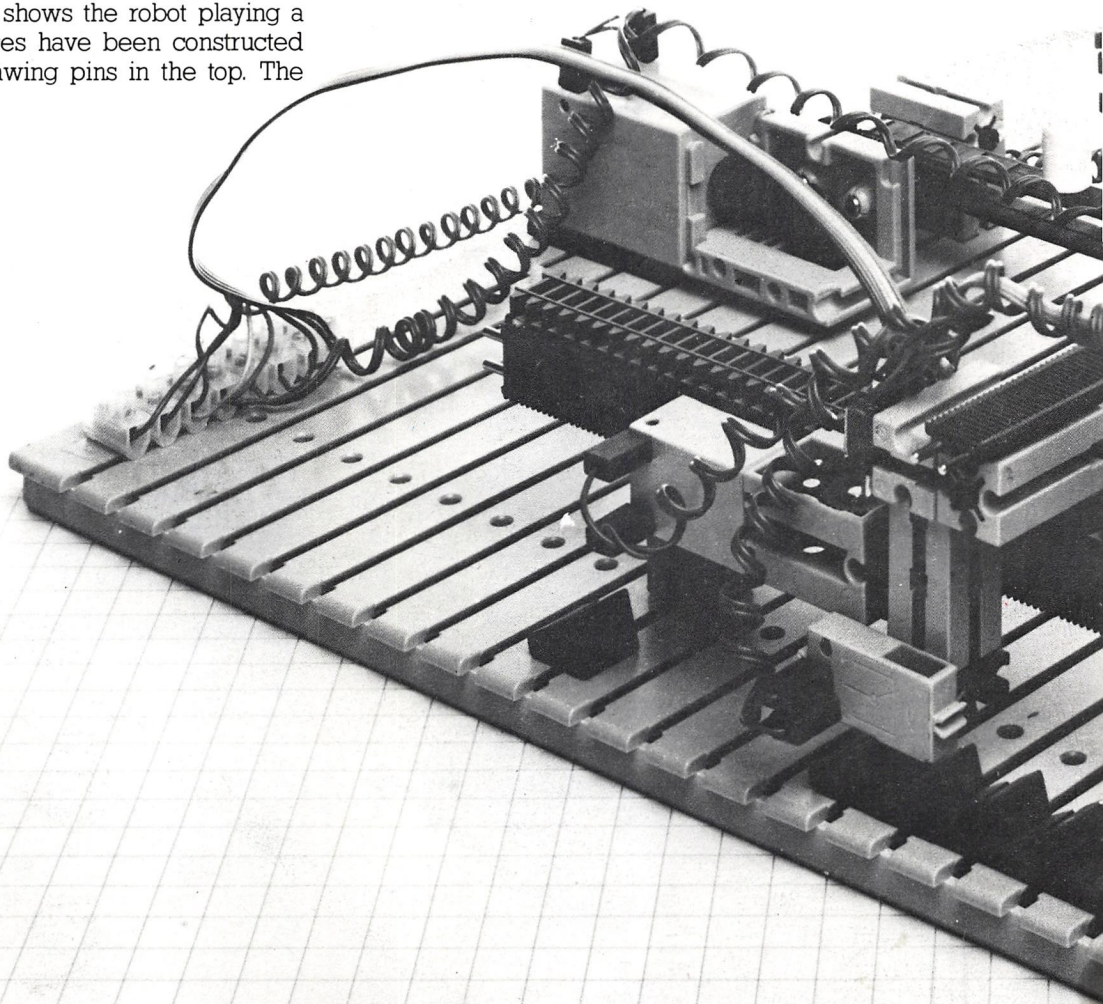
This project shows you how to build an X-Y robot from Fischertechnik and control its movement from a micro-computer. The model shown here is really a chassis upon which a great many experiments and further projects could be based. The techniques explained should also enable you to develop quite different robots based on Cartesian coordinates.

The photograph on this page shows the robot playing a board game. The draughts pieces have been constructed from wooden doweling with drawing pins in the top. The

standard Fischertechnik electromagnet then makes an excellent end effector. Supplied with this project are a series of subroutines for moving the end effector to any desired coordinates and then activating or de-activating the magnet. From these subroutines you can construct your own programs for playing board games, setting the computer up against human opponents or against itself.

Start with simple games first, such as noughts and crosses. You will discover that there isn't sufficient clearance for pieces to be jumped over each other, so your program will have to keep track of the state of the board and move new pieces around existing obstacles. It may even be necessary for the robot to clear a path temporarily and then restore the pieces to their correct positions after the move has been made. This calls for considerable skill in programming, but the X-Y robot provides a highly rewarding way of learning about such problems.

Game playing is not the only application. A miniature conveyor belt has been included in the design so that a warehousing robot can be simulated. The subroutine supplied will advance the conveyor belt until the light beam at the end is broken. That means that the next item is ready for collection by the robot. You can try labelling the draughts pieces to



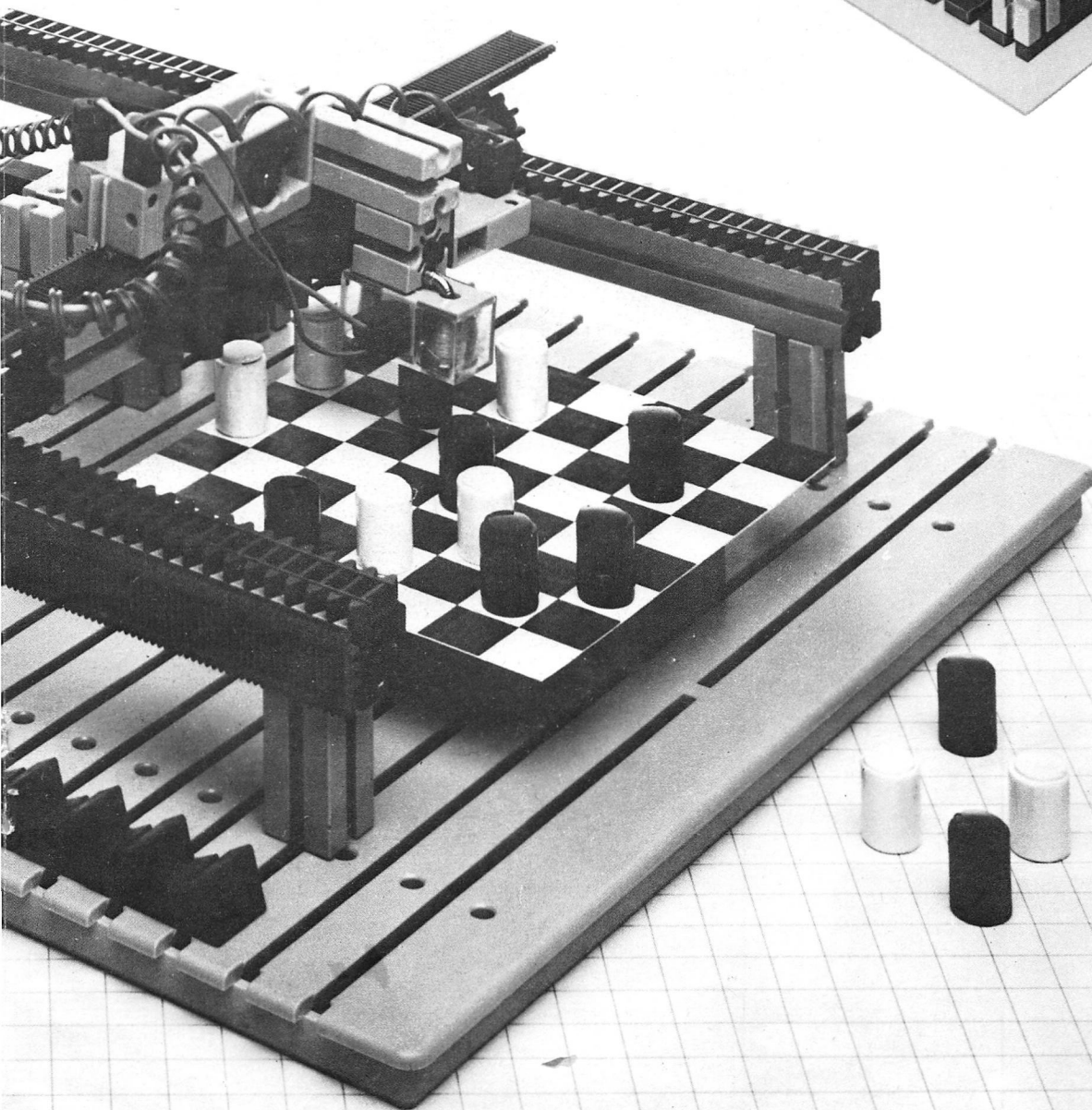
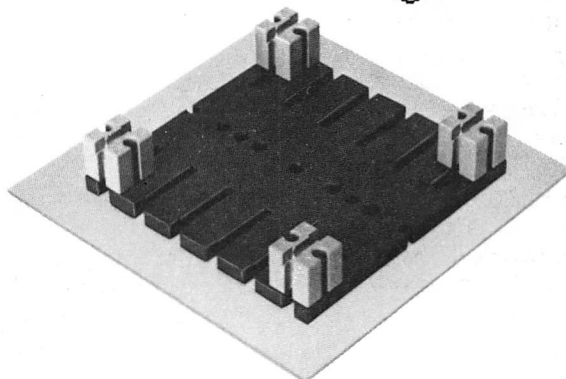
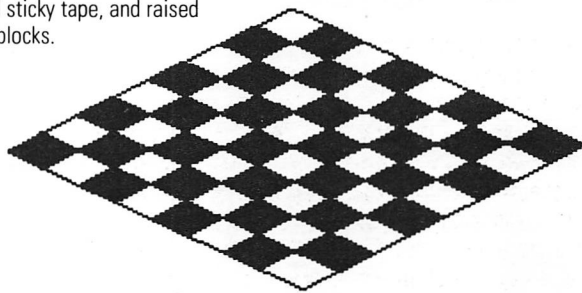
represent different stock items, then write a program that accepts an item from the conveyor, together with a label typed in on the keyboard, and stores the item somewhere on its floorspace. On request from the keyboard, the conveyor is reversed and the robot retrieves the appropriate item.

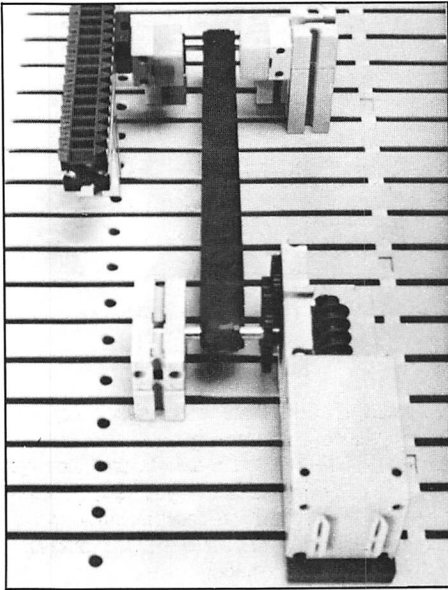
Fundamental to the design of this robot is the positional feedback employed on both axes, without which it would be impossible to position the end effector accurately. A micro-switch passes over a line of angled bricks and the computer keeps track of position by counting the closures of the switch. You can alter the spacing of the angled bricks to suit the application and fit as many or few as you need. When the computer is instructed to move to co-ordinates 4,7, it will move the end effector to the fourth angled brick along and the seventh one across, so it doesn't even matter if their spacing is unequal.

Once you've exhausted these possibilities, you can add to the basic chassis. If you have a third minimotor, then a Z axis to raise and lower the end effector would be a boon. Alternatively, you could replace the electromagnet with a hook or a mechanical end effector. Finally, you can extend the warehouse concept by adding another conveyor. Individual items arrive on one, the robot stacks them into neat piles, and then ships the loaded pallets out the other end.

Making the chessboard. Mark out the board on a piece of stiff card, with the size of the squares identical to the size of a standard building block (15mm). The board is then mounted onto a square baseplate, using double-sided sticky tape, and raised on four half-blocks.

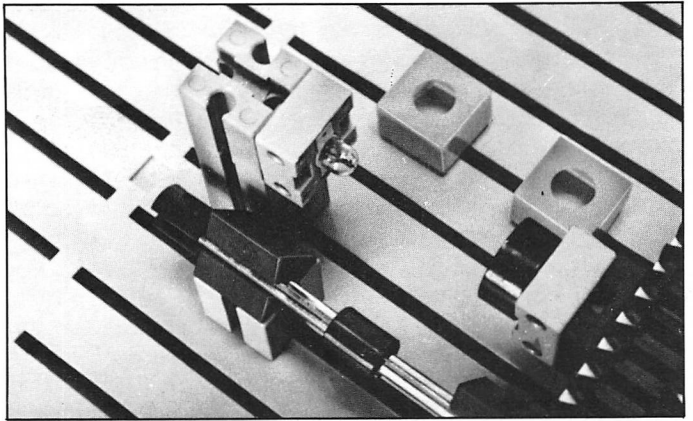
Playing pieces. A rod of 1cm ($\frac{3}{16}$ in) thick doweling is sawn into pieces of equal length. A drawing pin (of the kind that can be picked up by magnet) is inserted into each, and the group of pieces painted or labelled as required.



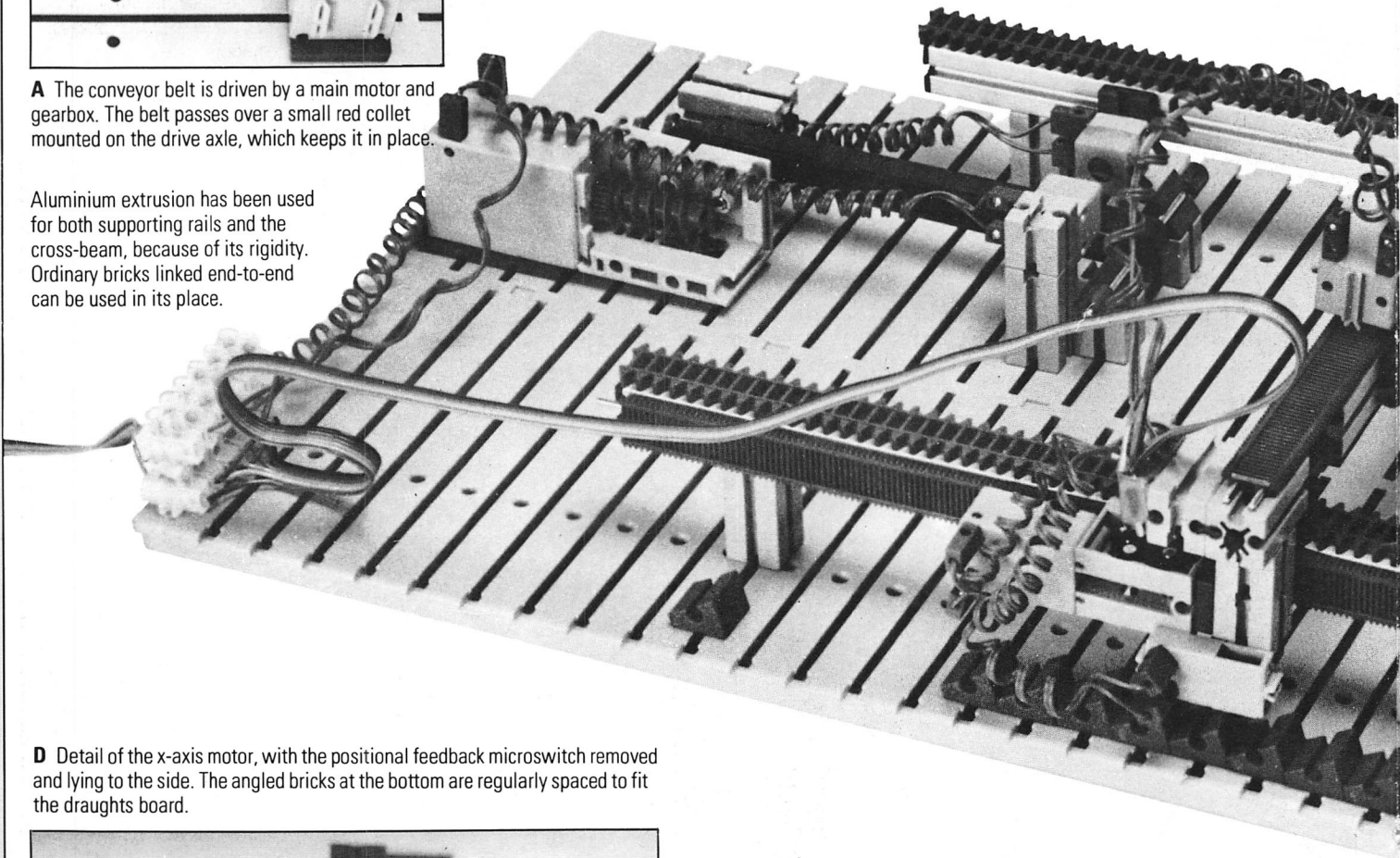


A The conveyor belt is driven by a main motor and gearbox. The belt passes over a small red collet mounted on the drive axle, which keeps it in place.

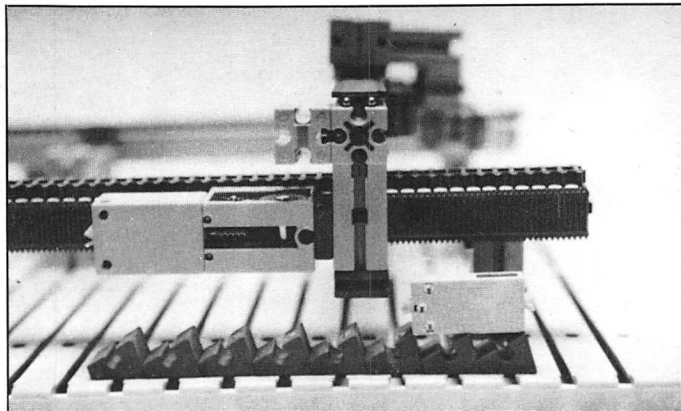
Aluminium extrusion has been used for both supporting rails and the cross-beam, because of its rigidity. Ordinary bricks linked end-to-end can be used in its place.



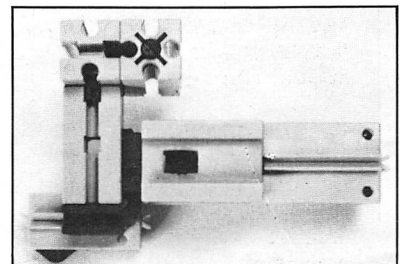
B The caps supplied must be fitted over the light bulb and light detector. The system may still be prone to extraneous light, in which an additional cap, made from cardboard and with a small hole in the end, should be constructed and placed over the light detector.

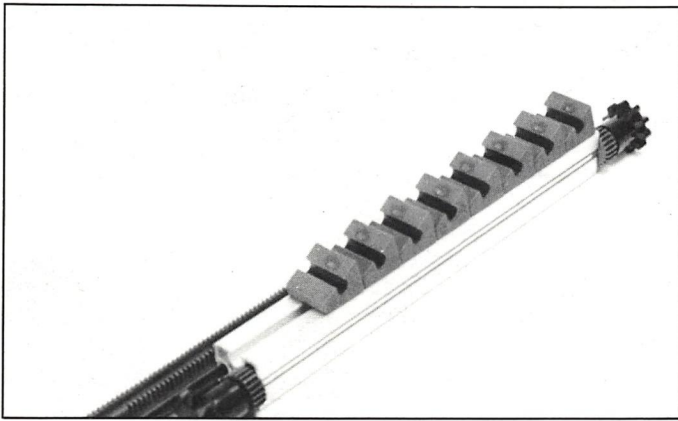


D Detail of the x-axis motor, with the positional feedback microswitch removed and lying to the side. The angled bricks at the bottom are regularly spaced to fit the draughts board.

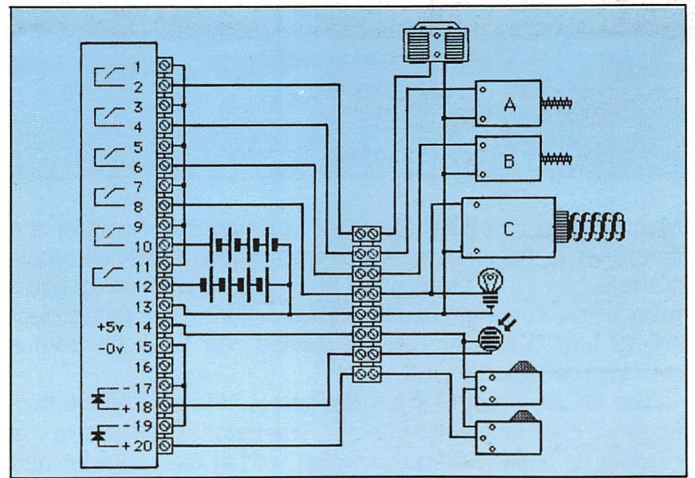


E The x-axis motor and microswitch removed, shown from the reverse side.





C The two black pinions at either end of the cross-beam are locked tight onto an axle that passes through the length of the beam. They engage with the racking on the top surface of the two supporting rails, and their function is to ensure that the cross-beam remains perpendicular to the rails.



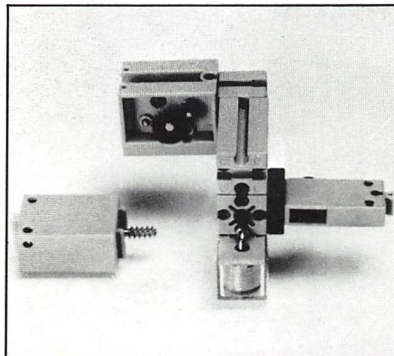
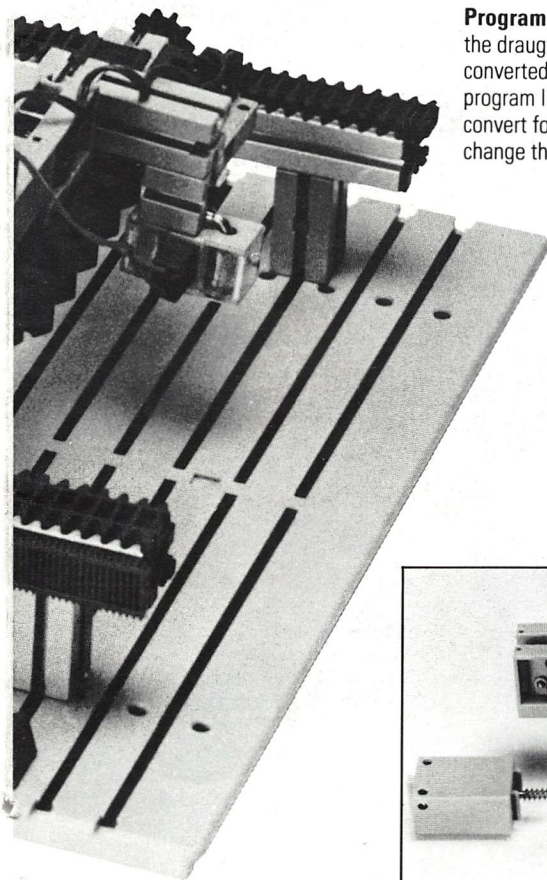
Wiring diagram. The two microswitches are wired up in series so that they can be read by a single input line (the other is for the light cell). So one of the switches must be held closed for the other one to be read. Our program stops one axis with the switch closed before moving the other. If your relay block has more lines available, or you omit the conveyor belt, feed each microswitch to a separate input. Note, too, that the light bulb is in parallel with the conveyor motor, because the two are always activated together.

Program. This automatically sets out the draughts board. It can be easily converted by rewriting or adding more program lines between 150-1000. To convert for use on Commodore 64 change the value of PRT to 56577.

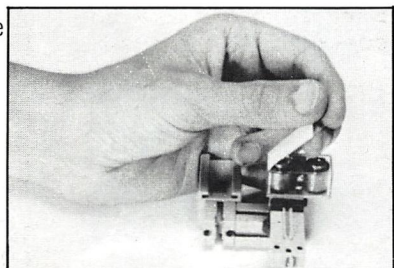
```

10 REM**DRAUGHTS ROBOT CONTROLLER**
20 REM**INITIALIZE**
30 PRT=37136:POKEPRT+2,63
40 PRINT"INSERT BATTERIES"
50 REM VARIABLES
51 MG=0 :REM MAGNET FLAG
52 MX=2 :REM MOTOR X
53 MY=4 :REM MOTOR Y
54 M=8 :REM CONVEYOR MOTOR
55 D1=16 :REM DIRECTION 1
56 D2=32 :REM DIRECTION 2
57 OFF=0 :REM TURN OFF MOTORS
60 REM**USER SETS TO 1,1**
70 PRINT"USE CURSOR KEYS TO SET TO 1,1"
80 PRINT"AND THEN PRESS RETURN"
90 GETAS:IFA$=""THENX=2:Y1=1
100 IFA$=""THENX=2:X1=1:GOSUB1000
110 IFA$=""THENY=2:Y1=1:GOSUB1100
120 IFA$<>CHR$(13)THEN90
130 X=1:Y=1
150 REM**SET UP BOARD**
160 FORI=1TO24:READXP,YP:GOSUB1200
170 X1=1:Y1=0:GOSUB1000:GOSUB1100
180 GOSUB1300
190 X1=XP:Y1=YP:GOSUB1100:GOSUB1000
200 GOSUB1350
210 NEXTI:END
1000 REM**DRIVE X TO X1**
1010 IFX1>8ORX1<1THEN1080
1020 DX=SGN(X1-X)
1030 IFDX=-1THENPOKEPRT,MX+D2+MG:GOTO1050
1040 POKEPRT,MX+D1+MG
1050 IF(PEEK(PRT)AND64)=0THEN1050
1060 IF(PEEK(PRT)AND64)=64THEN1060
1070 X=X+DX:IFX<>X1THEN1050
1080 POKEPRT,OFF+MG+D1:RETURN
1100 REM**DRIVE Y TO Y1**
1110 IFY1>8ORY1<0THEN1180
1120 DY=SGN(Y1-Y)
1130 IFDY=-1THENPOKEPRT,MY+D2+MG:GOTO1150
1140 POKEPRT,MY+D1+MG
1150 IF(PEEK(PRT)AND64)=0THEN1150
1160 IF(PEEK(PRT)AND64)=64THEN1160
1170 Y=Y+DY:IFY<>Y1THEN1150
1180 POKEPRT,OFF+MG+D1:RETURN
1200 REM**FEED CONVEYOR**
1210 POKEPRT,M+D1+MG:GOSUB1400
1220 IF(PEEK(PRT)AND128)=0THEN1220
1230 POKEPRT,OFF+MG+D1:RETURN
1300 REM**MAGNET+FLAG ON**
1310 GOSUB1400
1320 MG=1:POKEPRT,MG+D1:GOSUB1400:RETURN
1350 REM**MAGNET+FLAG OFF**
1360 GOSUB1400
1370 MG=0:POKEPRT,MG+D1:GOSUB1400:RETURN
1400 REM**SLIGHT PAUSE**
1410 FORTD=0TOL00:NEXTTD:RETURN
10000 DATA8,8,6,8,4,8,2,8,7,7,5,7,3,7,1,7
10010 DATA8,6,6,6,4,6,2,6,7,3,5,3,3,3,1,3
10020 DATA8,2,6,2,4,2,2,2,7,1,5,1,3,1,1,1

```



F Detail of the end effector, the y-axis motor and positional feedback microswitch, which can be finely adjusted for position when mounted on the cross-beam.



G A small piece of paper is stuck on the bottom surface of the electromagnet with adhesive tape. Without this, the drawing pins are likely to become permanently magnetized.

LEGO ROBOT ARM

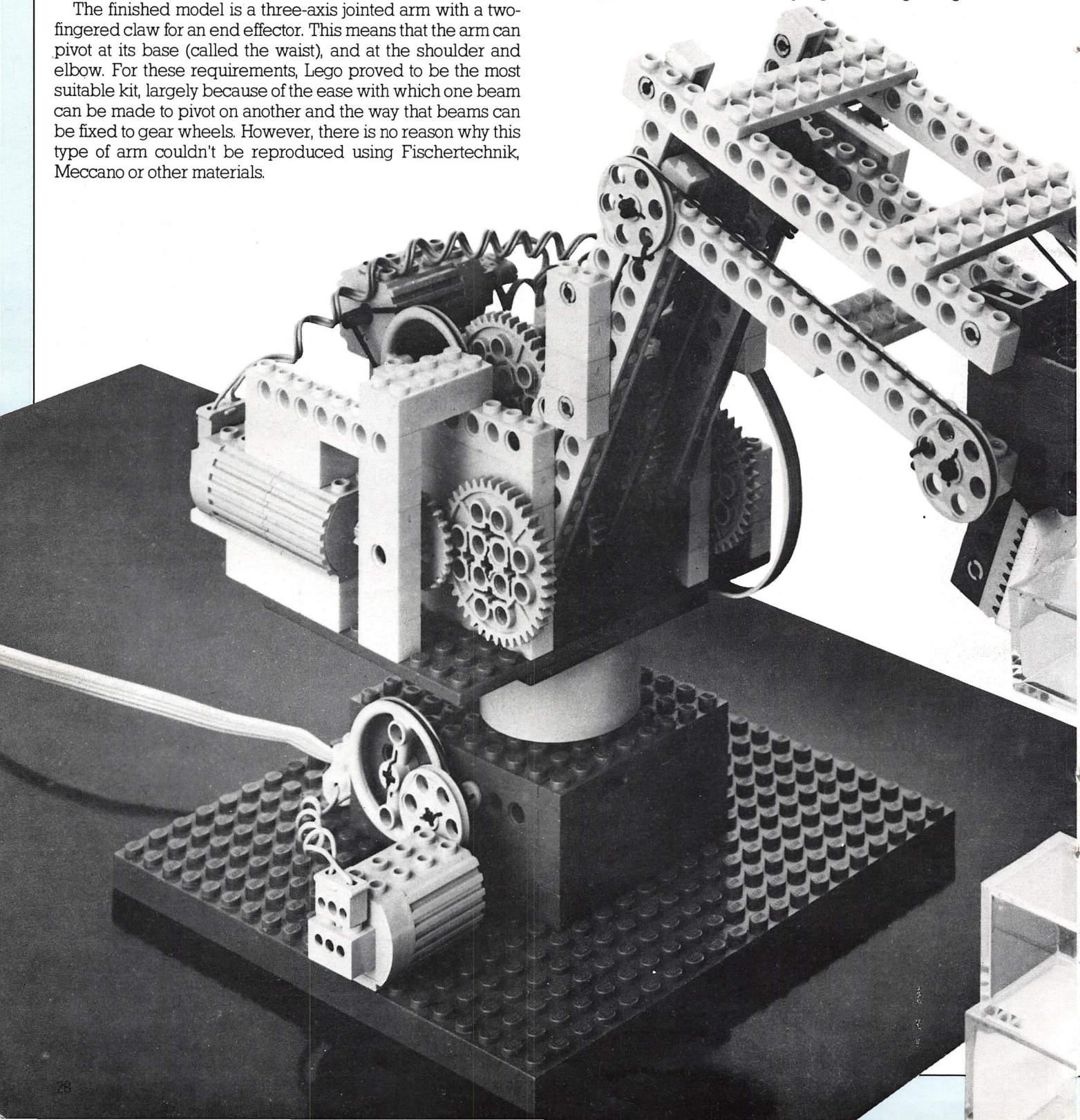
Manipulating a small-scale robot arm under the control of a computer is the ideal way of learning how industrial robots operate, as well as being great fun. Purpose-built educational robot arms, however, can cost £500 or more, but this project shows how a similar type of device can be built from a standard construction kit.

The finished model is a three-axis jointed arm with a two-fingered claw for an end effector. This means that the arm can pivot at its base (called the waist), and at the shoulder and elbow. For these requirements, Lego proved to be the most suitable kit, largely because of the ease with which one beam can be made to pivot on another and the way that beams can be fixed to gear wheels. However, there is no reason why this type of arm couldn't be reproduced using Fischertechnik, Meccano or other materials.

The arm can be driven through battery boxes and switches, or interfaced to a computer, as shown overleaf. It must be stated, though, that its operation falls somewhat short of the more expensive off-the-shelf products. Specifically, the working envelope of this robot – the volume of space that its end effector can reach – is very restricted. It can lift only very light loads and, most important, it lacks any form of feedback, so precise control is difficult.

Nevertheless, this project demonstrates several very important principles in the construction of robot arms, and these will help you to design your own models, employing more axes of freedom or using a quite different geometry.

Firstly, it demonstrates the very high level of gearing down



needed to obtain useful torque and controllable speed. Using a combination of Lego gearboxes plus trains of conventional gear wheels, the motor powering the end effector is geared down by 300:1, the shoulder by 250:1, and the elbow by 500:1. These arrangements would be far too clumsy for an industrial robot, which is why they employ the harmonic drive system (see page 15).

Secondly, the base unit highlights the need for proper bearings that can provide relatively friction-free rotation and, at the same time, structural support – the weight of the extended arm exerts a considerable bending moment (turning force) on the axis of rotation. Our model uses a large roadwheel to bear the load, but a professional design would employ ball bearings or a proper turntable.

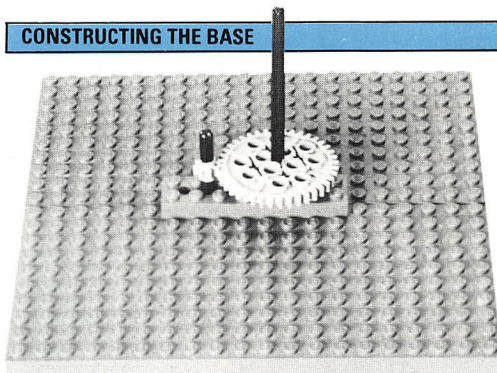
A third feature of importance is the way that power is transmitted through an arm that changes in shape. At both the shoulder and elbow, the power drive (consisting of chains and

string belts) passes through the axis of the joint so that rotating one limb will not cause movement in another.

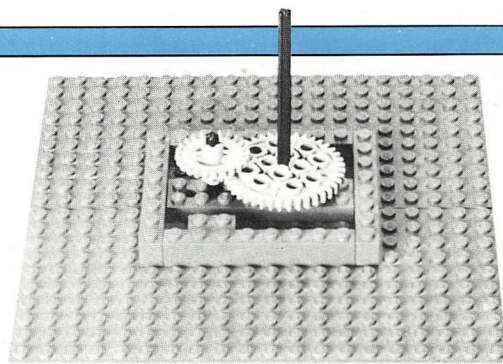
Finally, note the way that rubber bands have been used on both the forearm and upper arm. They counteract the weight of the robot's own limbs, without adding further mass as a traditional counterbalance would. Without some form of counterbalance, almost all of the motor's effort would be taken up with lifting the arm itself, rather than any useful load at the end effector.

As stated before this project is more about construction techniques than about control systems. You could write a computer program that memorized a sequence of movements and then played them back, but it wouldn't be very accurate. The program printed allows the arm to be controlled from the computer's keyboard in live mode, and you can set yourself various exercises, such as stacking building blocks or playing noughts and crosses.

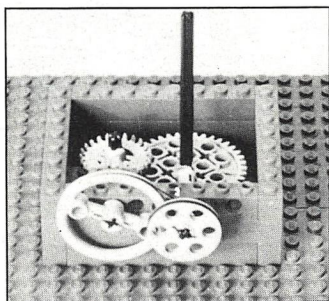
CONSTRUCTING THE BASE



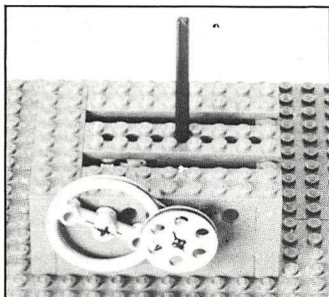
1 Two rectangular baseboards are held together using two 8×2 holed plates. The 40-tooth gear wheel is mounted on a 12-unit-long axle, the 8-tooth gear on a 4-unit-long axle.



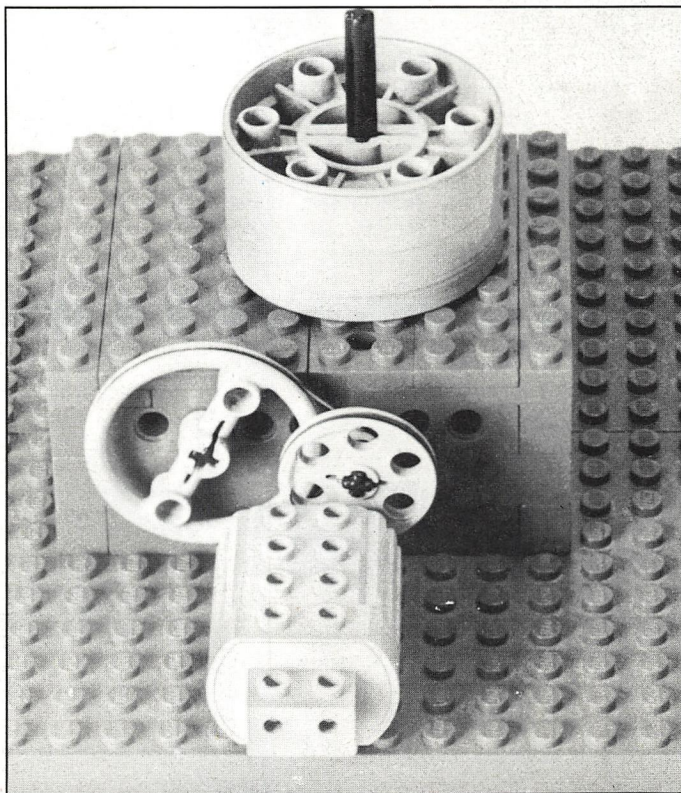
2 A 24-tooth crown wheel is fitted onto the shorter axle, and the first layer of the base housing is added. Note the location of the 2×1 brick inside the casing.



3 Two pulley wheels are mounted on 4-unit axles. An 8-tooth gear wheel on the left-hand axle meshes with the crown wheel. The other axle has a bush either side of the wall.

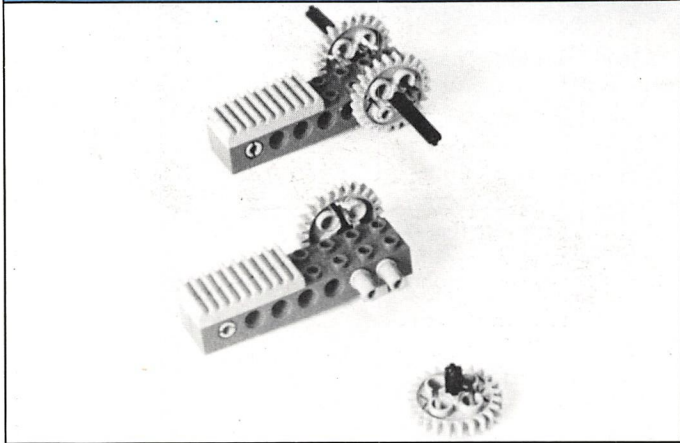


4 The first layer of 8×2 top plates is added. Rubber belts are fitted to both pulleys, the left-hand one passing over the bush on the right-hand axle.

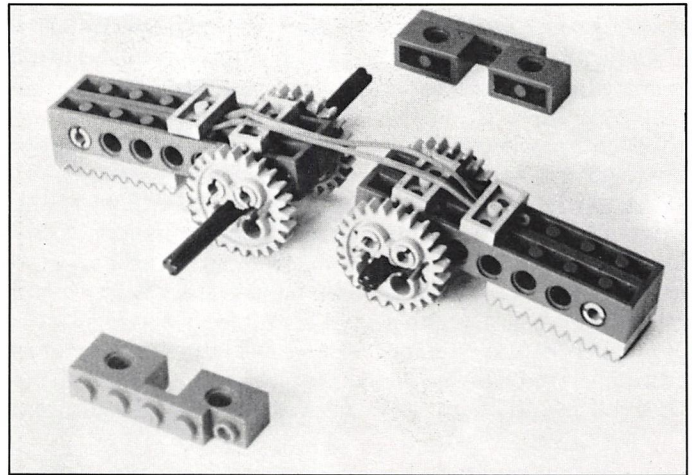


5 A second layer of plates is added, followed by the inner part of a large tractor-type wheel to act as a bearing. The second rubber belt links motor (in foreground) and smaller pulley.

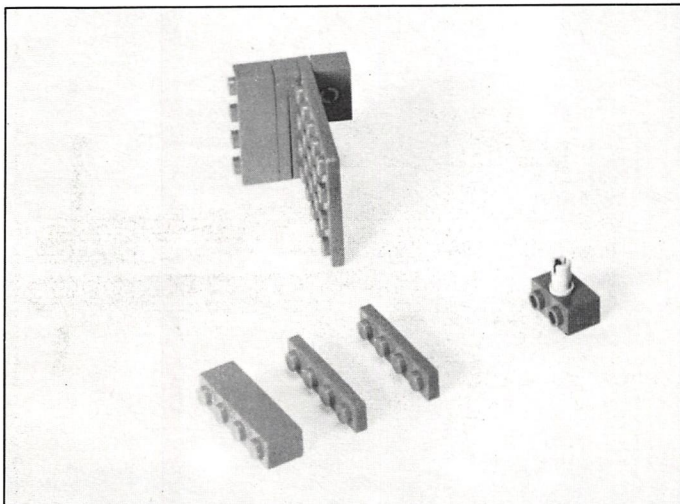
CONSTRUCTING THE END EFFECTOR



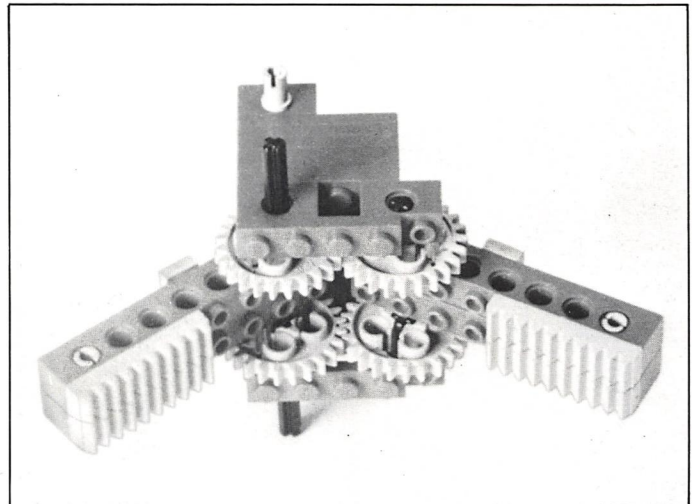
6 The two fingers of the end effector are identical, except that one (below in picture) features 2-unit-long axles, where the other has 4-unit-long axles. All four gear wheels have 24 teeth.



7 Two 2x1 plates hold the halves of each finger together, the outermost plate also securing a thin elastic band, the purpose of which is to spring the two fingers apart. On each side two 2x1 holed bricks and a 4x1 cross plate are used to link the fingers.

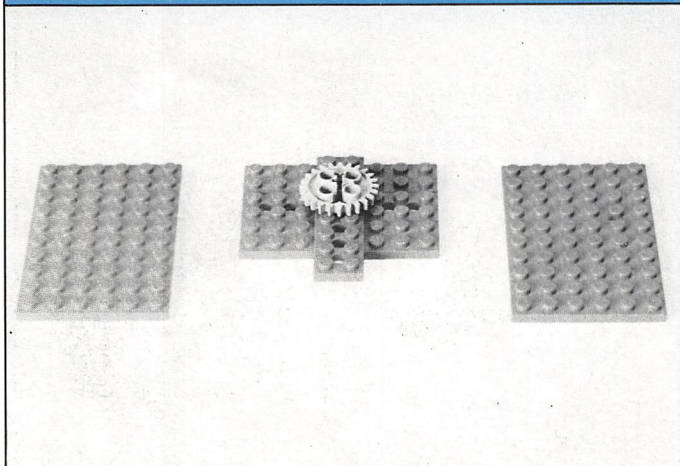


8 The two sides of the end effector housing, completed using 4x1 plates and bricks, fit onto a 6x4 plate. Connector pins are fitted into the two 2x1 holed bricks on the top surface to join onto the arm.

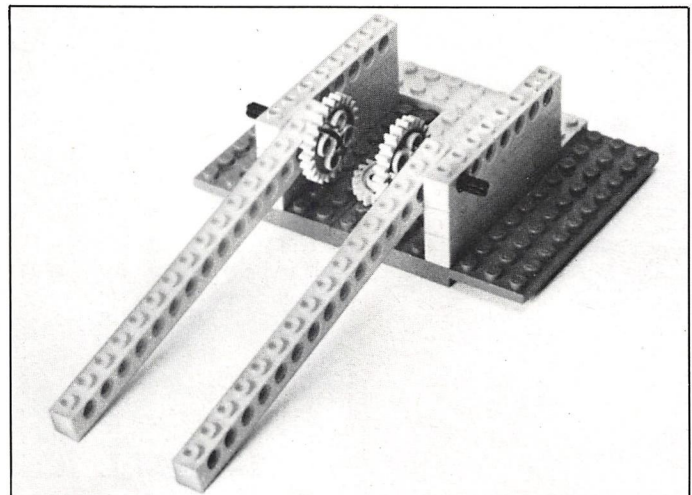


9 The finished end effector. Check that the jaws open and close smoothly, and that the elastic band runs across the top of the jaws, providing moderate tension when closed.

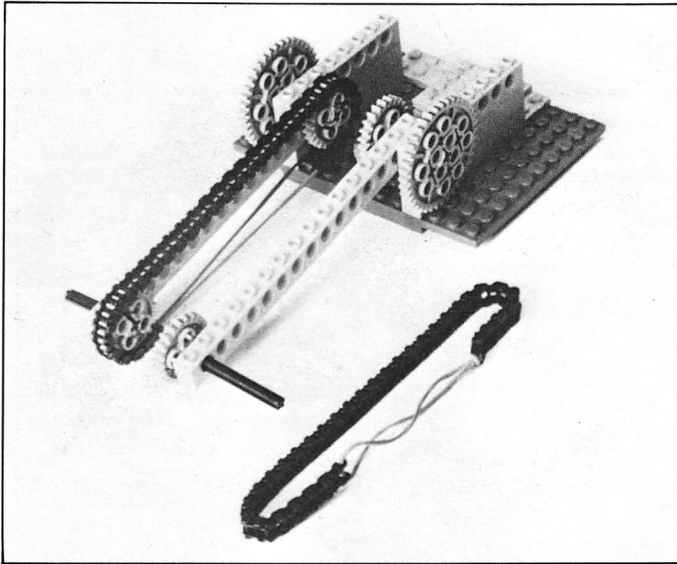
CONSTRUCTING THE ARM



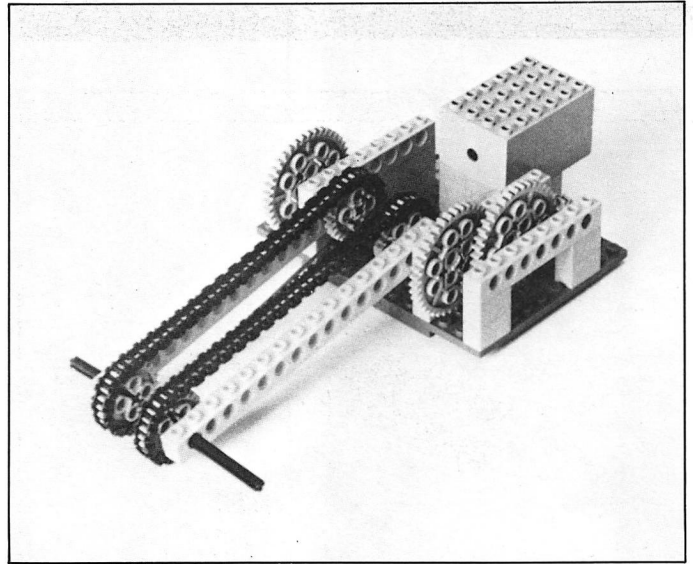
10 The baseplate consists of two 6x10 plates and four 2x8 plates, two of which have holes. A 24-tooth gear wheel fits over the common hole, to take the turntable axle.



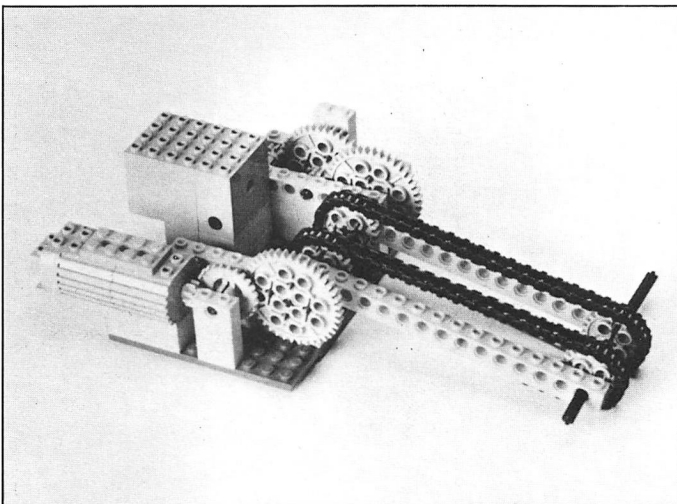
11 The side walls are three layers high. A 4-unit axle, bearing a 24-tooth gear wheel and a 16-unit-long holed beam, is fitted to the front of each side. A single 8x2 plate goes behind the two sides.



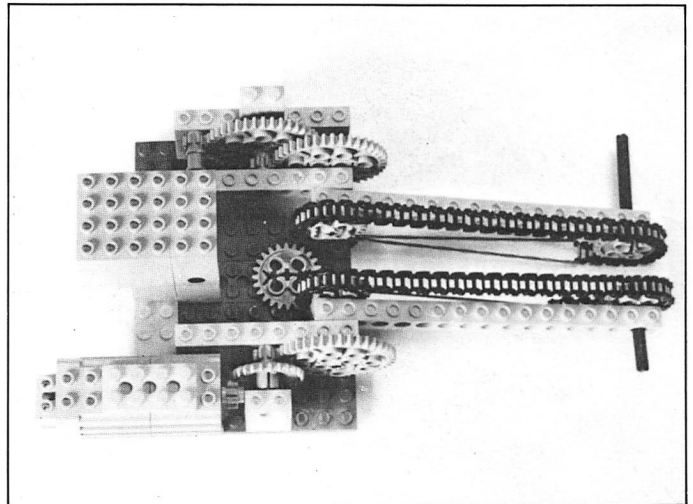
12 A 4-unit axle and a 6-unit axle, both fitted with 24-tooth gears, form the elbow joint. The drive chains are lengthened with rubber bands, which also provide tension. A 4×2 and a 4×4 plate are added to support the gearbox.



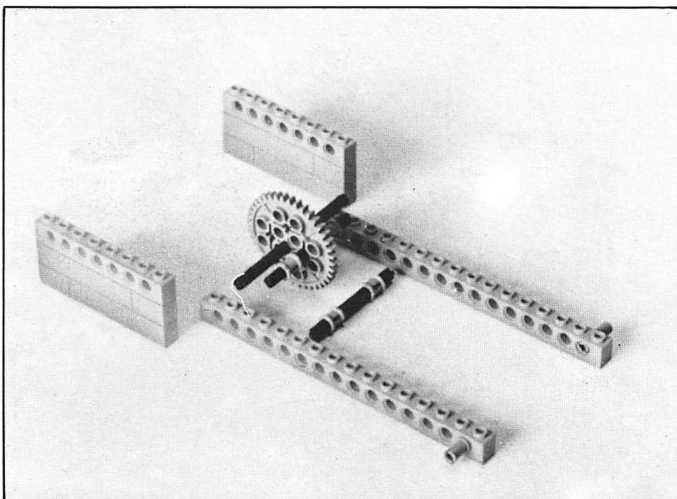
13 The gearbox is mounted on a total of three sets of plates and one course of bricks. A 6-unit axle passes from the box through the side into a train of 8- and 40-tooth gears supported by a raised 8×1 holed beam.



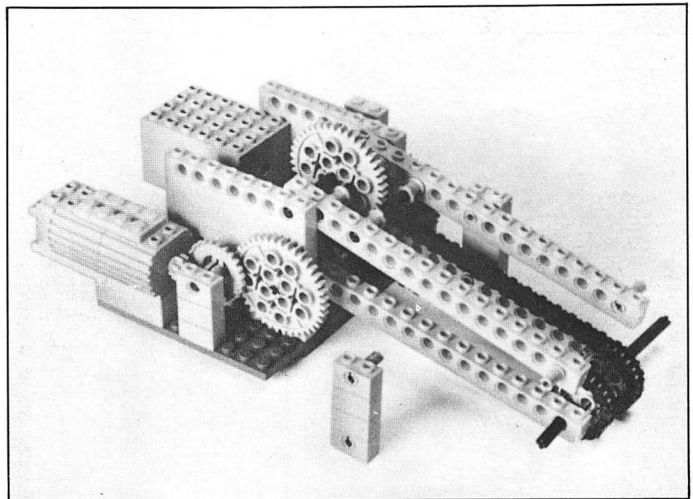
14 The second motor used in the project also drives through a 20:1 gearbox to an 8-tooth gear wheel. This meshes with a crown wheel, mounted on a 4-unit axle along with another 8-tooth gear (not visible in this picture).



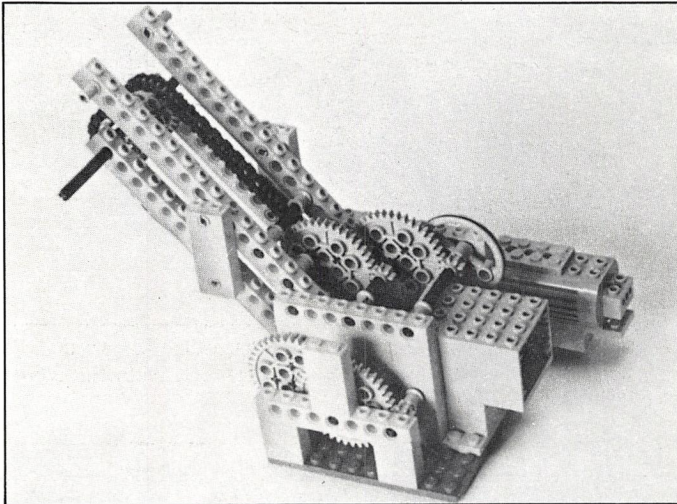
15 This shows the detail of the gear trains on both sides of the main housing. Note that once the gearboxes are mounted, neither of the sets of gears should be forced.



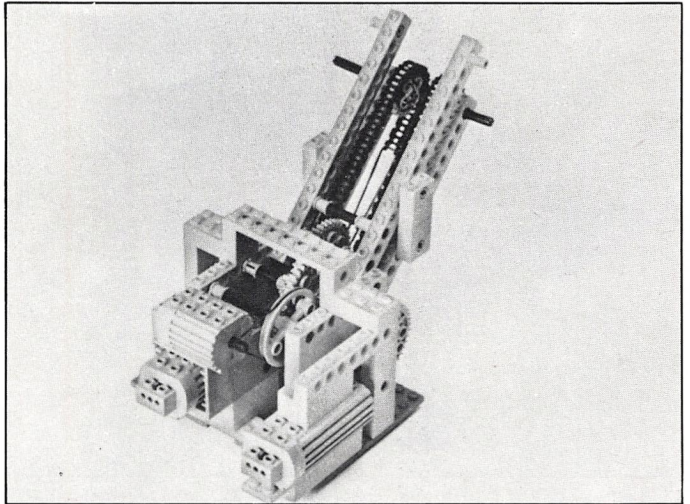
16 The upper body and arm are constructed from 8-unit- and 16-unit-long beams respectively. Two 6-unit axles bridge the arm, while one 8-unit axle bearing a 40-tooth gear locates the arm in the two sides.



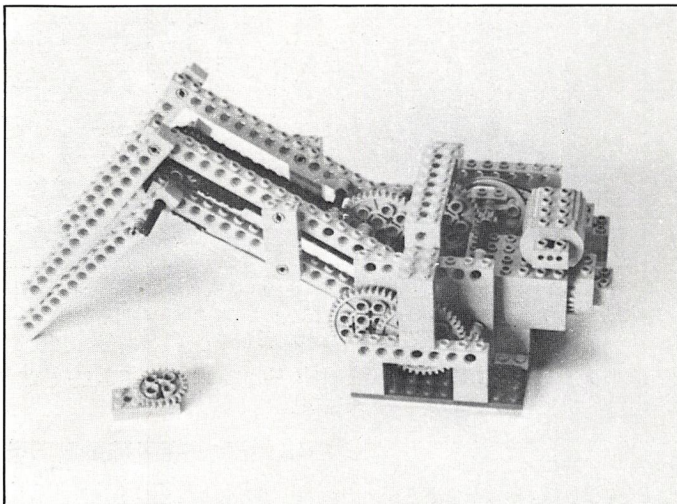
17 After mounting the new structure on the main body, two side pieces are fitted, to keep the main beams parallel. Each consists of four 2×1 bricks with two connecting pins.



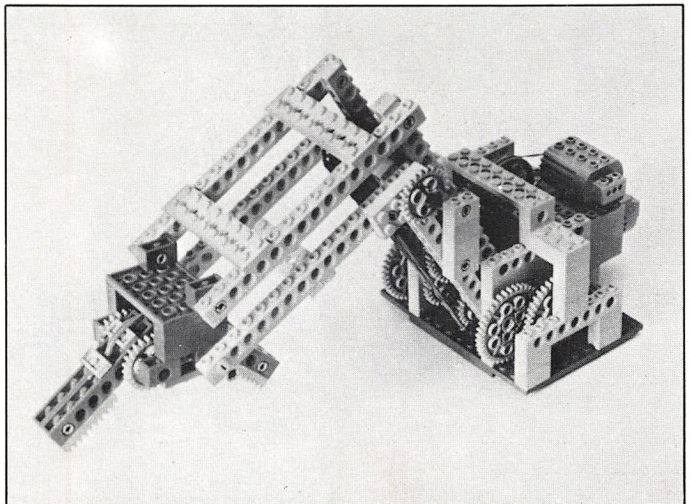
18 Two more 8-unit axles, one above the main gearbox and one centrally, bridge the main body, with a train of 8- and 40-tooth gears mounted on them. The large pulley has already been fitted with a rubber belt.



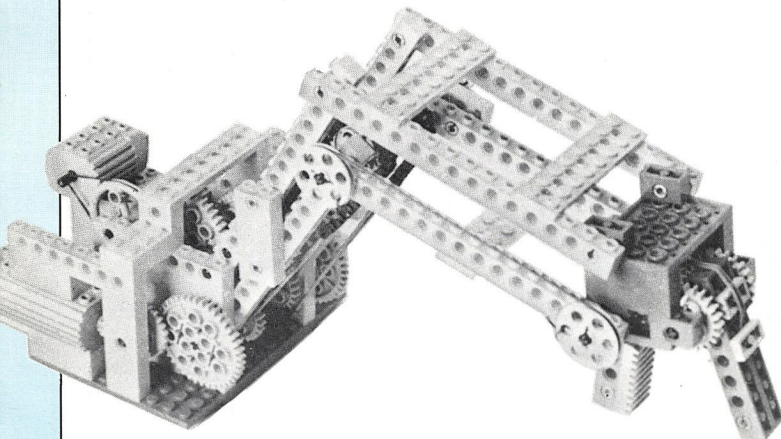
19 The top motor, placed above the main gearbox, is now engaged with the rubber belt, and the remaining motor inserted into its gearbox. A supporting bridge of plates and bricks strengthens the structure.



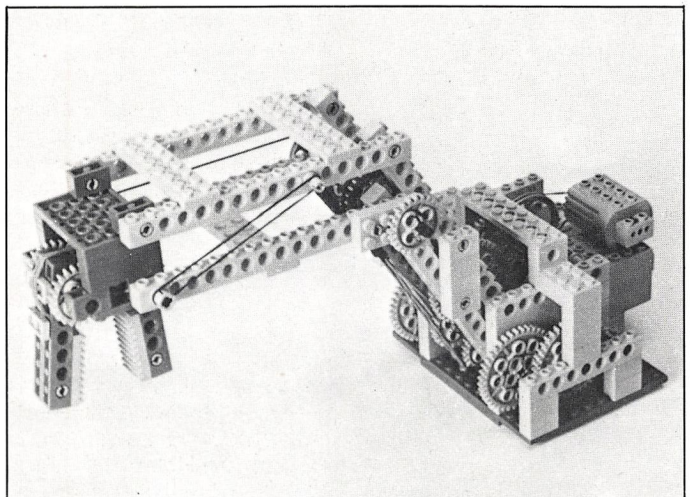
20 Four 16-unit-long beams make up the forearm. An 8x2 holed plate bearing a 24-tooth gear is fitted over the longer of the protruding axles and secured via the peg.



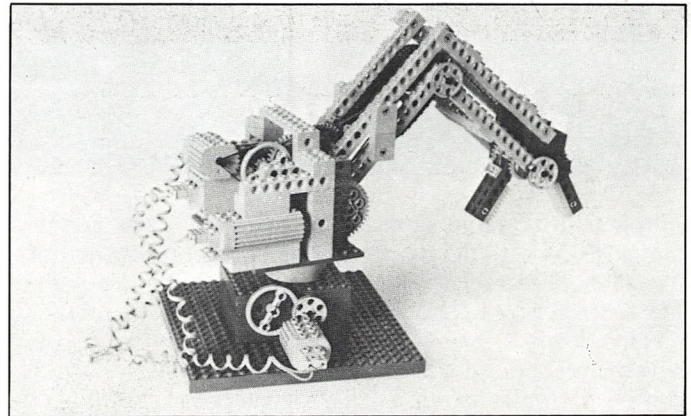
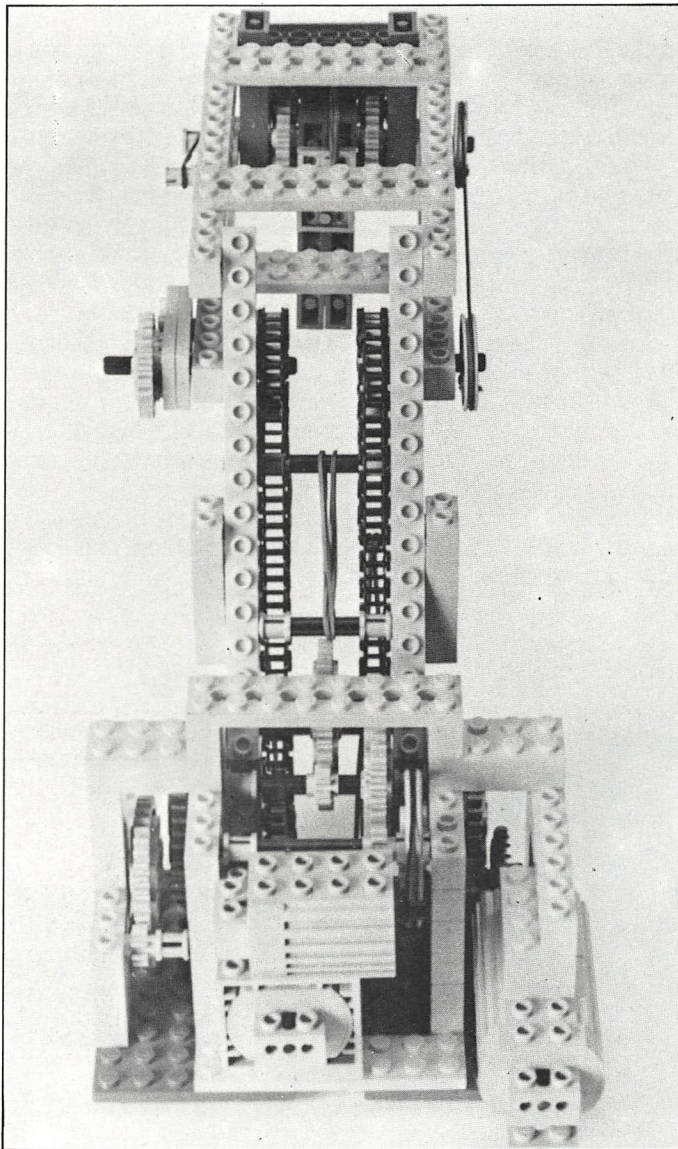
21 The end effector can now be fixed to the forearm and secured with a bush one side and a pulley wheel the other. Three 8x2 plates placed transversely make the arm more rigid.



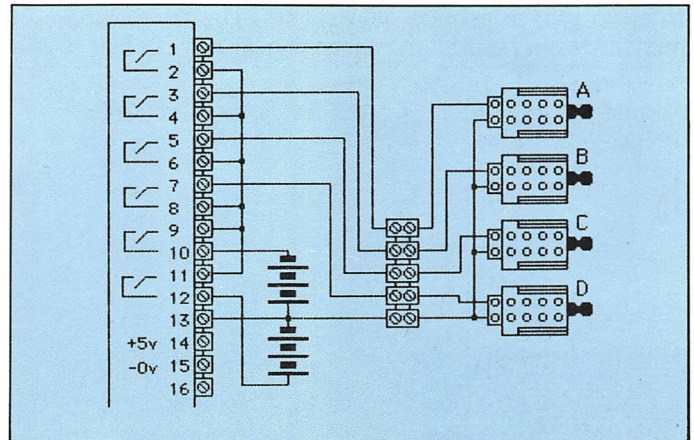
22 A piece of string is tied between two pulley wheels, one placed on the elbow, one on the wrist. Experiment with the length of the string so that the motor can open and close the jaws fully.



23 A rubber band is fitted to the forearm using an additional pin. It should bear the weight of the arm, but not force it upwards so that the drive chain becomes slack.

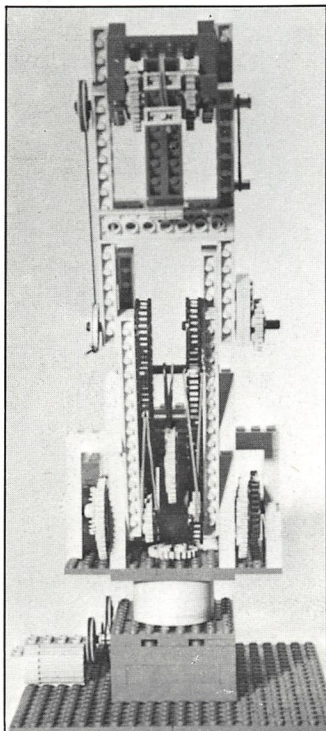


26 The arm is now ready for wiring up, to either four battery boxes and switches or a computer. The wires can be coiled and screwed into a connection block for neatness.



Wiring diagram. One wire is common to the four motors, and they are driven from four relays. Using two sets of batteries and two further relays, it then becomes possible to drive each motor in either direction.

WARNING: LOAD AND RUN THE COMPUTER PROGRAM BEFORE INSERTING THE BATTERIES INTO THE BATTERY HOLDER.



24 The complete arm, seen from behind. A second rubber band, this one doubled up, and placed centrally, counterbalances the weight of the upper arm. The new, 6-unit axle, is fitted between the two lower beams in the upper arm, near to the elbow joint.

25 The arm is now fitted to the base unit. The protruding axle passes through the baseplate and engages with the gear wheel. The body must be pushed down as far as possible.

Program. After initializing the interface block, the movements of the arm are controlled with eight keys: 1 and Q rotate the base in either direction, 2 and W the shoulder, 3 and E the elbow, 4 and R the gripper. Written for the Vic-20, the program can easily be modified for the Commodore 64.

```

10 REM**ARM CONTROLLER**
20 REM**INITIALIZE**
30 PRT=37136:POKEPRT+2,63
40 PRINT"INSERT BATTERIES"
50 DIMKEY(8):FORI=1TO8:READKEY(8):NEXTI
60 MA=1:MB=2:MC=4:MD=8:D1=16:D2=32:OFF=0
100 REM**CONTROL**
110 SCAN=PEEK(197):T=9:FORI=1TO8
120 IFSCAN=KEY(I)THEN T=I:I=9
130 NEXTI:IFT=9THENPOKEPRT,OFF:GOTO110
140 ONTOSUB200,250,300,350,400,450,500,550
150 POKEPRT,OFF:GOTO110
200 REM**SHOULDER UP**
210 POKEPRT,MA+D1:GOSUB1000:RETURN
250 REM**SHOULDER DOWN**
260 POKEPRT,MA+D2:GOSUB1000:RETURN
300 REM**FOREARM UP**
310 POKEPRT,MB+D1:GOSUB1000:RETURN
350 REM**FOREARM DOWN**
360 POKEPRT,MB+D2:GOSUB1000:RETURN
400 REM**BASE CLOCKWISE**
410 POKEPRT,MC+D1:GOSUB1000:RETURN
450 REM**BASE ANTICLOCKWISE**
460 POKEPRT,MC+D2:GOSUB1000:RETURN
500 REM**OPEN GRIPPER**
510 POKEPRT,MD+D1:GOSUB1000:RETURN
550 REM**CLOSE GRIPPER**
560 POKEPRT,MD+D2:GOSUB1000:RETURN
1000 REM**DELAY**
1010 FORTD=0TO20:NEXTD:RETURN
10000 DATA0,56,1,57,48,9,49,10

```

CARD-DEALING ROBOT

Problem: design a robot that can deal cards. So simple for a human to perform, this function nevertheless represents quite a challenge for a robot. The model pictured here, made from Fischertechnik, can deal cards from a shuffled pack, face-up or face-down, to any of five players, and with reasonable reliability. For those who like to be one up on the Joneses, this device could be worth a lot of merit points, but the real purpose of the project is to demonstrate how the robot designer goes about tackling an awkward application.

One of the keys to solving this problem, for example, is knowing when to mimic the human method of working and when to pursue a quite different solution. So, looking at a human dealing cards, the first thing to note is that it is a

two-handed operation. The pack is held in the right hand, say, and cards are pushed off the top (or bottom if the dealer is unscrupulous) one at a time. The left hand grasps the extended card and pivots around to the appropriate player before releasing it, or turning it face up and then releasing it.

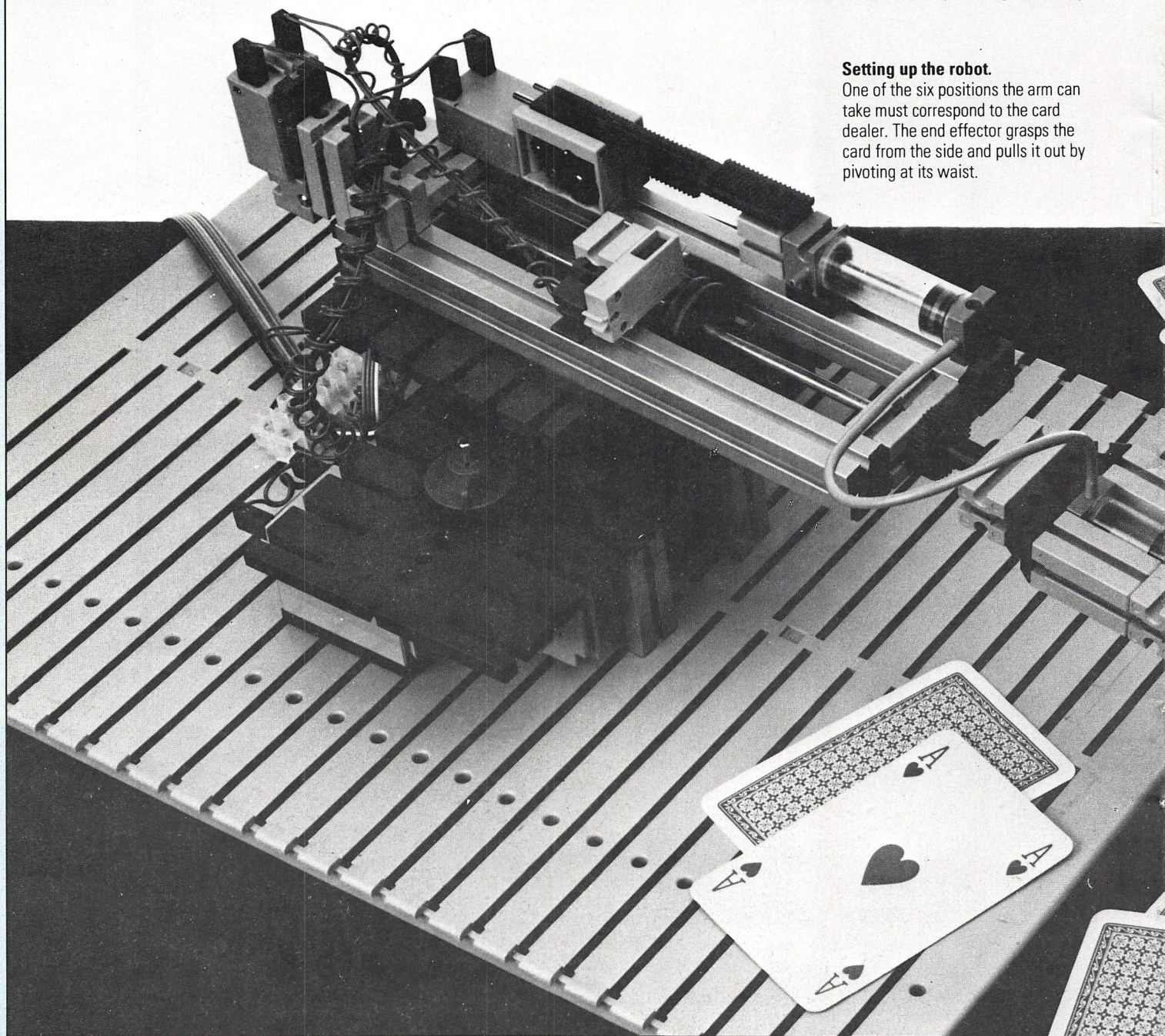
The job of the left hand should be well within the capabilities of a conventional robot arm. It need only have two degrees of freedom: the main pivot and the ability to rotate the end effector. It will need positional feedback to locate each of the players consistently, but this can be achieved with a microswitch.

Two flat fingers will suffice for the end effector, but the grip must be tight enough to prevent the card from falling out when the wrist rotates. The grip of plastic fingers can be improved by spreading on a little rubber-solution adhesive and letting it dry, but the best technique is to employ pneumatics because air is so compressible.

The job of the right hand is far more tricky to emulate using construction kit technology, though the problem has been solved in the form of sheet feeders in the printing industry.

Setting up the robot.

One of the six positions the arm can take must correspond to the card dealer. The end effector grasps the card from the side and pulls it out by pivoting at its waist.



Humans employ friction to slide off the top card from the pack, with pressure from the thumb. For a robot, a rubber tyre on a rotating wheel is the nearest equivalent. As each card is dealt the pack gets smaller, so the driving wheels will have to be pivoted on a supporting structure, to keep a constant pressure on the pack.

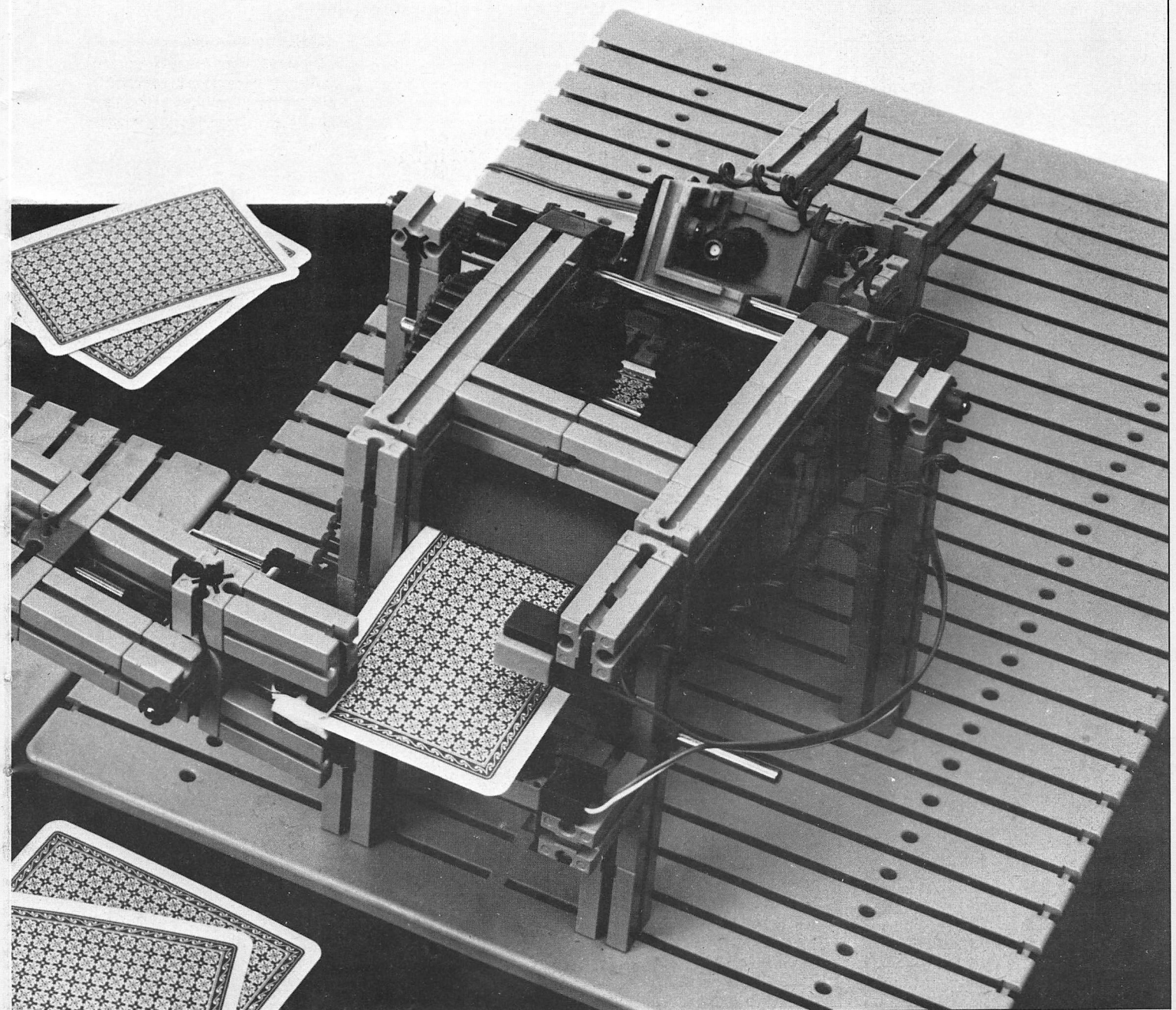
However, the cards will then come off the pack at different heights, which will cause a problem unless the robot arm is equipped with a vertical axis of movement. The solution to this problem demonstrates the technique of inverting the design: fix the driving wheels and pivot the pack of cards!

How is the delivery of cards off the pack to be co-ordinated with the movement of the robot arm, and indeed with the requirements of the players, which are presumably typed in on a computer keyboard? The supply of components to an industrial machine is often regulated using light beams and, fortunately, Fischertechnik can supply both light bulbs and light detectors. The motor can then drive a card off the pack until its front edge interrupts the beam, starting again when that card has been removed by the arm.

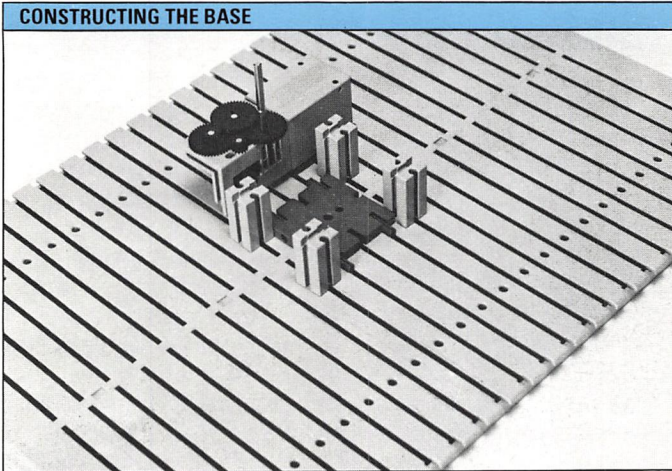
Based on this information you could proceed to build a device for dealing cards off a pack, but you would soon encounter the biggest problem of all: the stickiness of most playing cards will result in them coming off the pack in twos and threes. So how do humans cope with this? Well, while the thumb is pushing a card off the top, the four fingers are acting as a brake on the underneath – if more than one card is delivered, the movement of the fingers pushes the bottom ones back. We can simulate this on the robot with another rubber-rimmed wheel and a pair of deflector plates to guide the advancing cards over it.

When you build this project, you will find that several parts of the model must be finely adjusted to make it work – you may even have to try a couple of different packs of cards. We don't claim that the dealing works 100 per cent of the time, but building the robot is a very rewarding exercise.

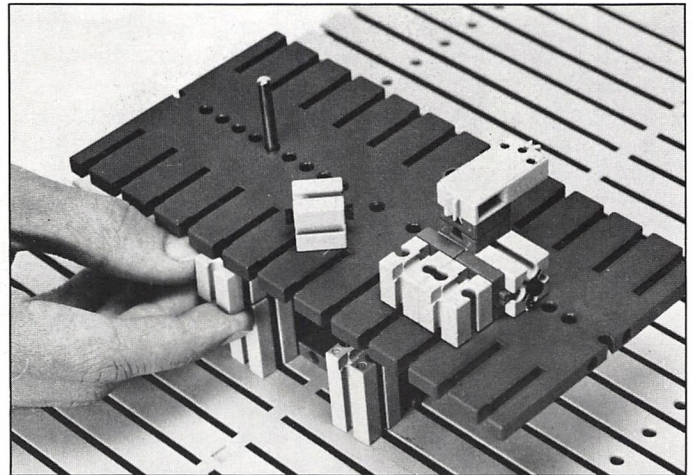
The program supplied consists of a series of simple sub-routines for dealing a card, operating the end effector and moving the arm. From these you can write your own applications programs, for card games such as pontoon, where each player can buy or twist cards from the pack.



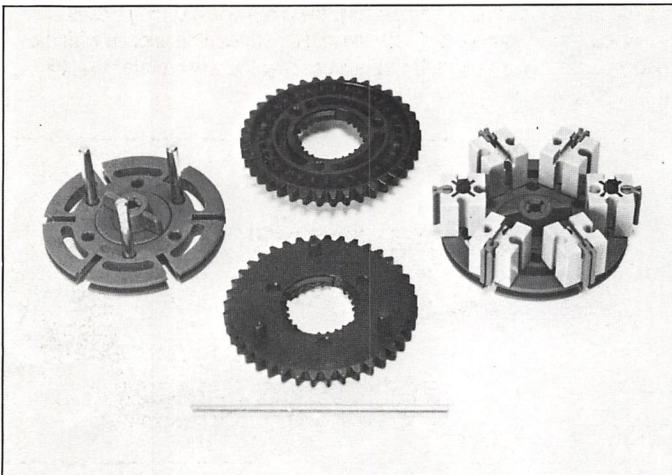
CONSTRUCTING THE BASE



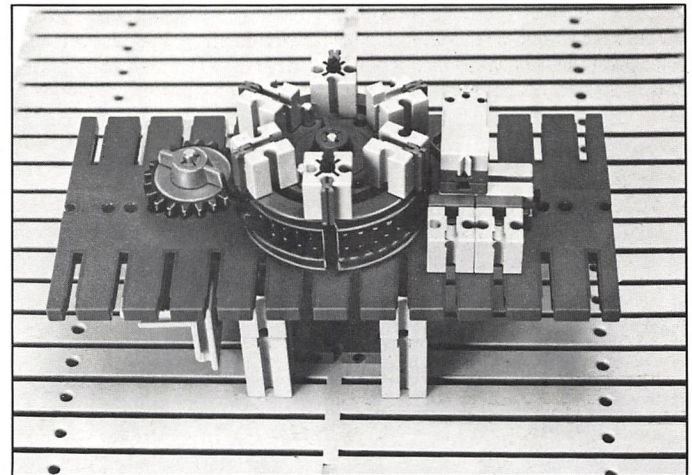
1 The motor and gearbox are mounted directly onto the large baseboard. The small red plate in the centre, between four legs, is to anchor the main turntable shaft. (If you don't have such a plate, use an arrangement of bricks.)



2 The large rectangular plate is fitted over the gearbox axle and fixed to the legs with four half-bricks. Next, the supporting structure for the microswitch is added. The switch measures rotation of the arm.

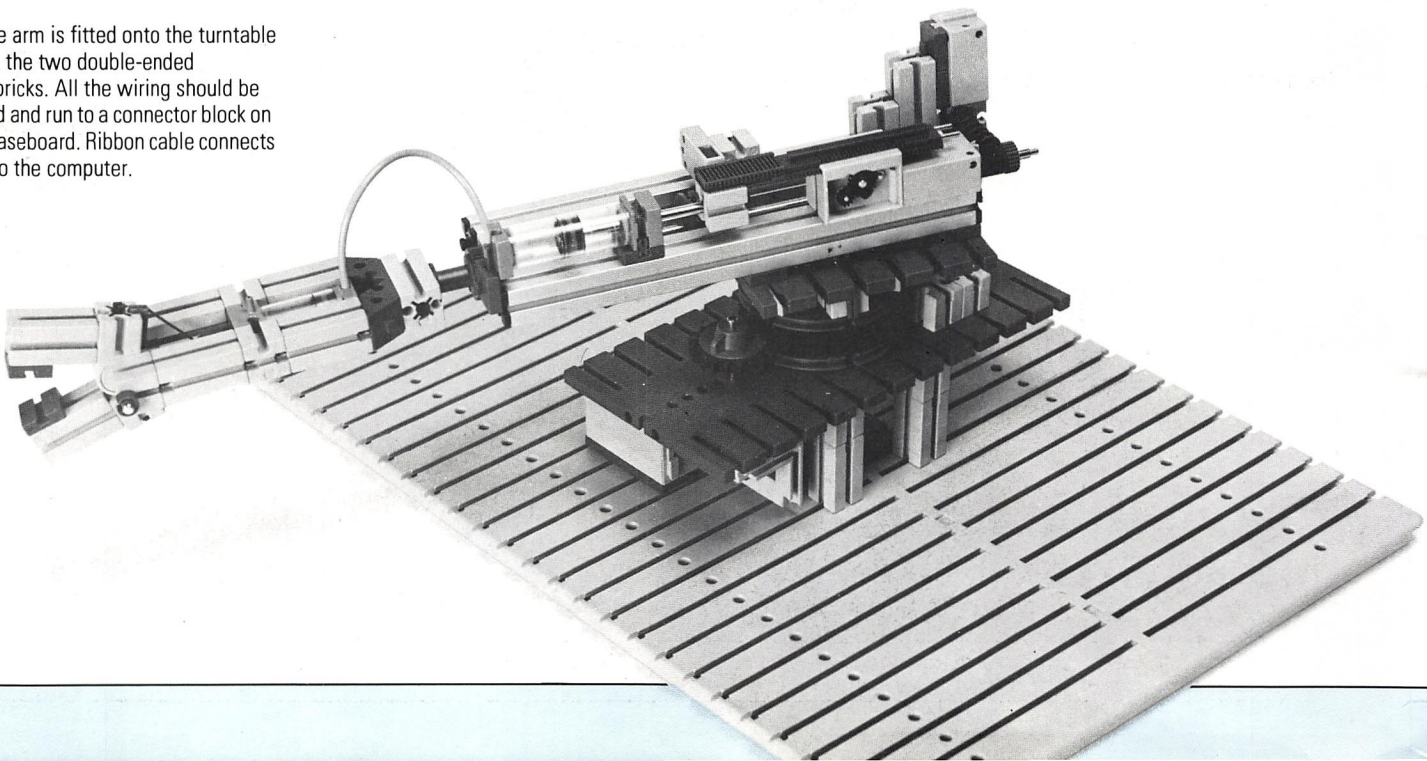


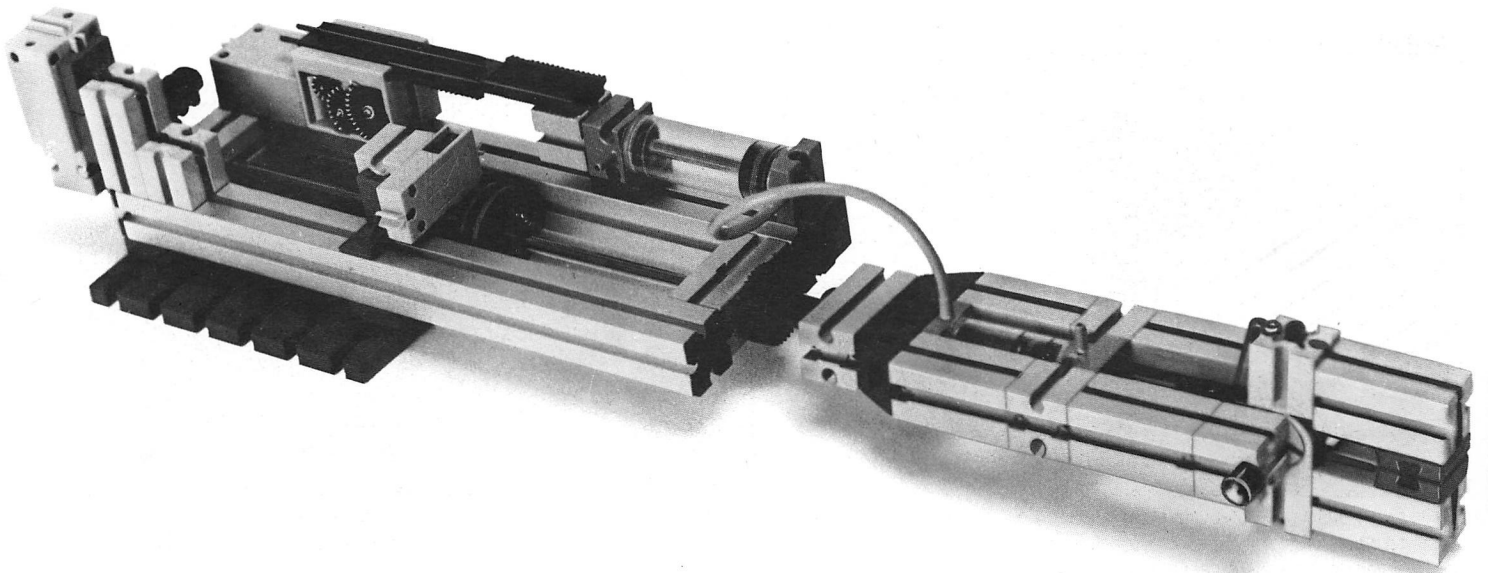
3 This rather complex structure for the turntable is needed for rigidity. If you have the Fischertechnik computer kit, a special turntable component is included. Two of the six half-bricks are double-ended.



4 The microswitch is adjusted so that the six red connectors in the half-bricks close the switch as they pass by. Now the driving gear can be fitted to the axle shaft, so that it engages with the turntable.

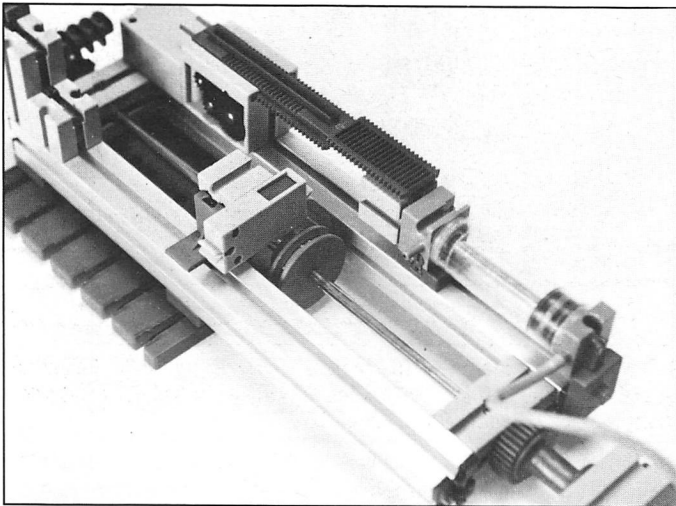
5 The arm is fitted onto the turntable using the two double-ended half-bricks. All the wiring should be coiled and run to a connector block on the baseboard. Ribbon cable connects this to the computer.



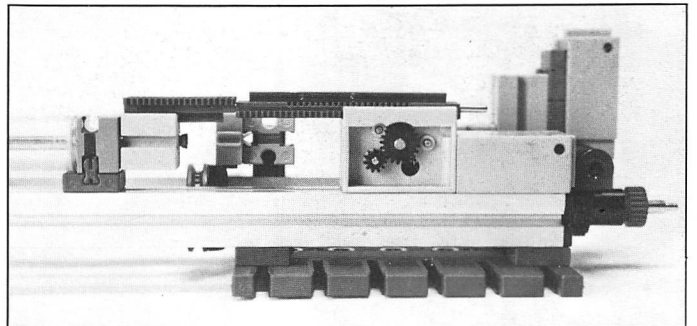


6 Two pieces of aluminium extrusion form the chassis of the arm – a line of six blocks would do the same job but with less rigidity. They are supported on the square baseplate with red quarter-bricks and red connectors, then all the other components slide between them from one end or are mounted on the top. The wrist is rotated by a minimotor at the rear, so you will need a long axle, or two

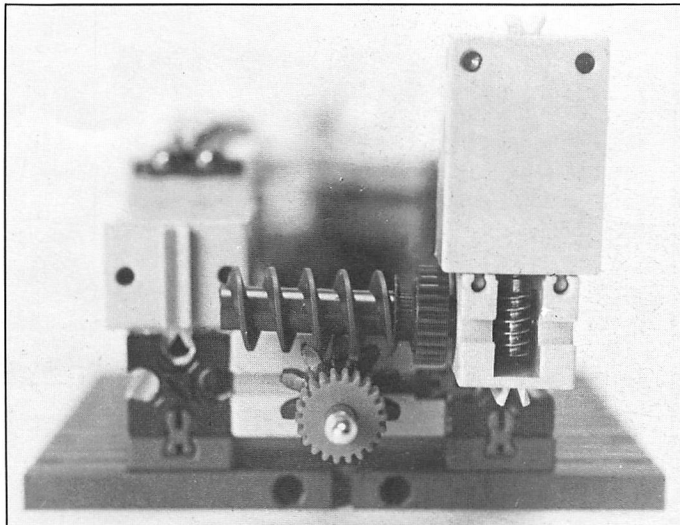
axles securely joined. The end effector is connected by string to the slave cylinder to open the two fingers equally, the hinge being sprung closed. A microswitch is mounted half-way along the extrusion, to engage with the wing-nut on a standard wheel. This ensures that the wrist rotates exactly half a turn.



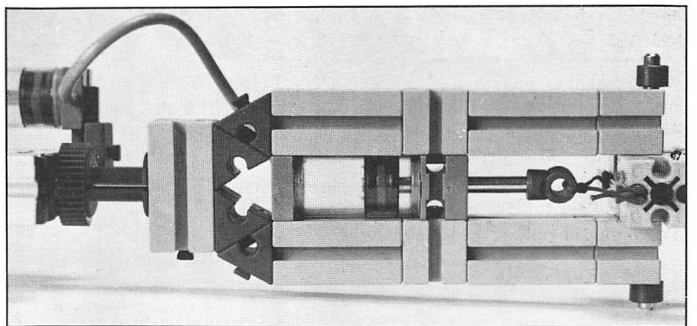
The flange on the central pulley wheel activates the microswitch.



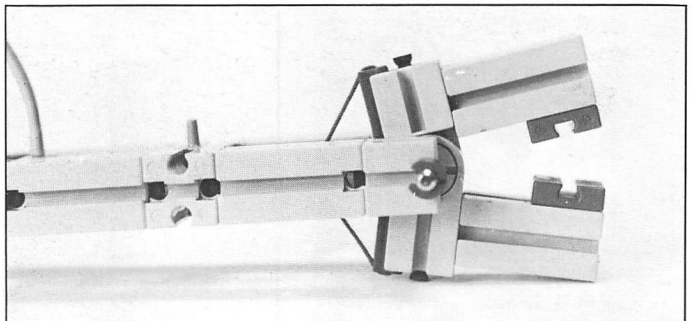
The rack and pinion drive powers the master cylinders.



The wrist motor is mounted at the back of the arm.

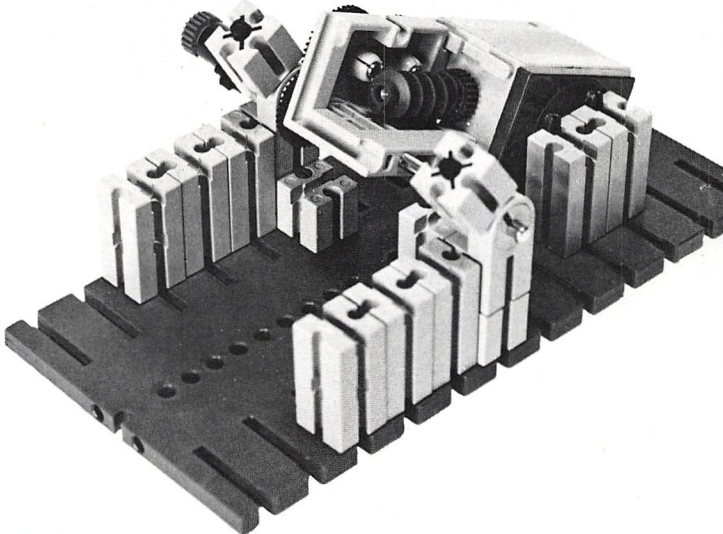


Detail of the wrist joint and slave cylinder.

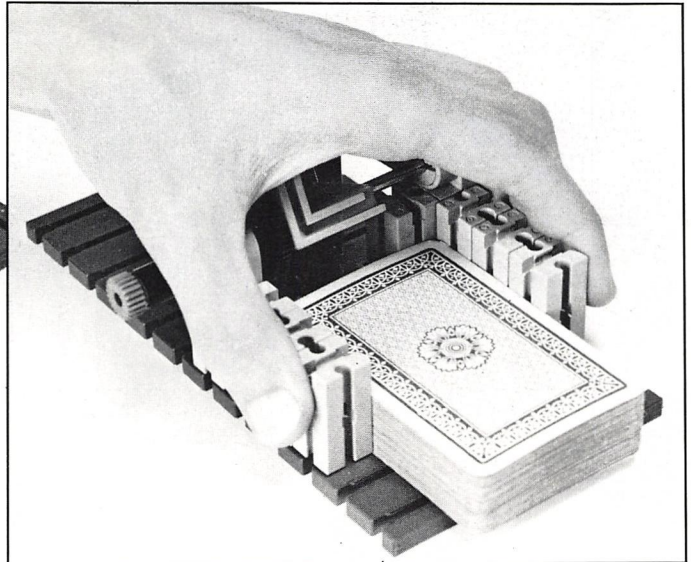


The hinge joint is sprung to close the gripper.

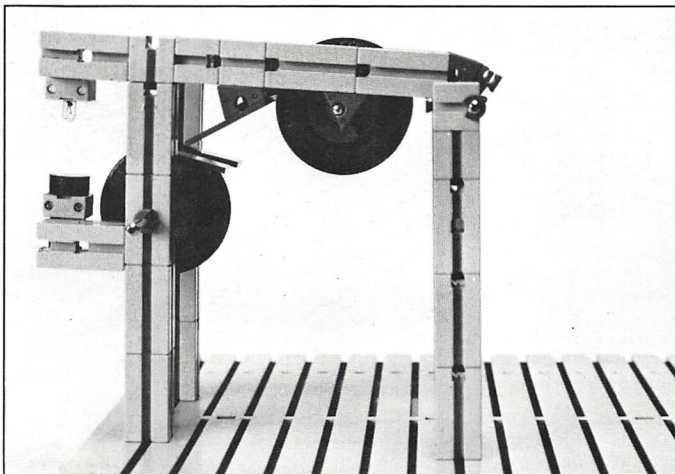
CONSTRUCTING THE DEALER



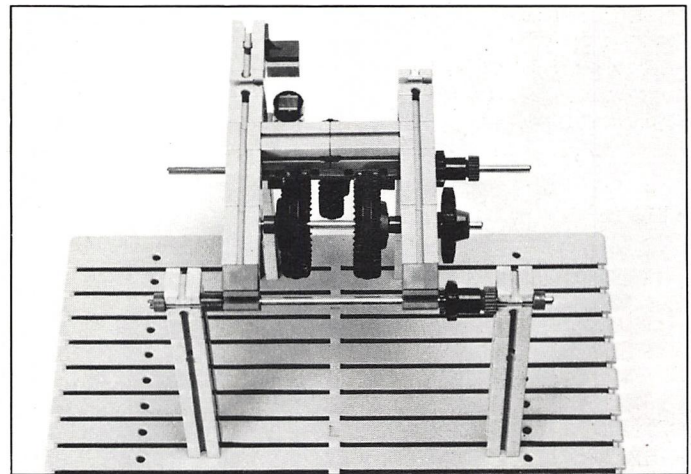
7 The rectangular plate holds the pack of cards and the driving motor. A long axle is used as the output from the gearbox, and this passes through the two hinges to a small black gear on the far side.



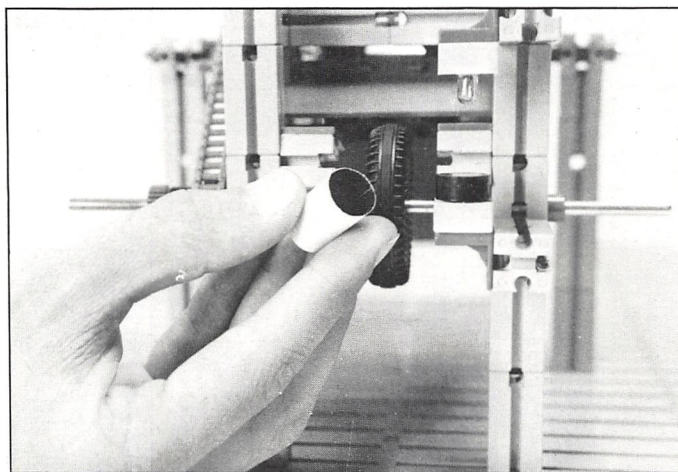
8 The blocks on either side should be squeezed together to provide the maximum grip on the pack of cards when it is in place. The less sticky the cards are, the better this robot will work.



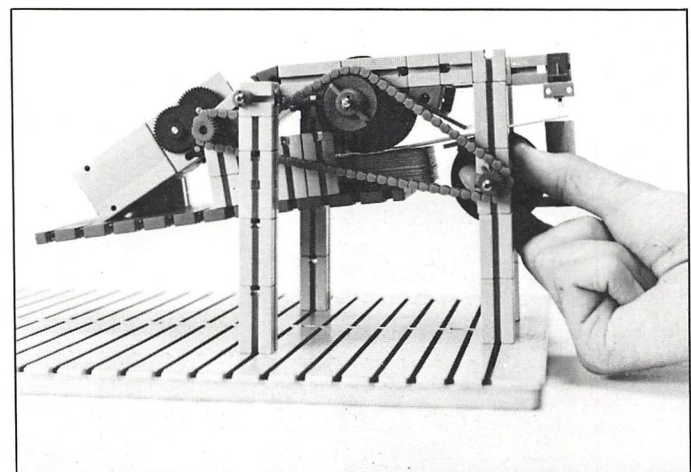
9 The supporting frame is shown here and in 10 from two angles. Note the mounting of the light bulb and detector, and the deflector plates for guiding the card over the frontmost wheel.



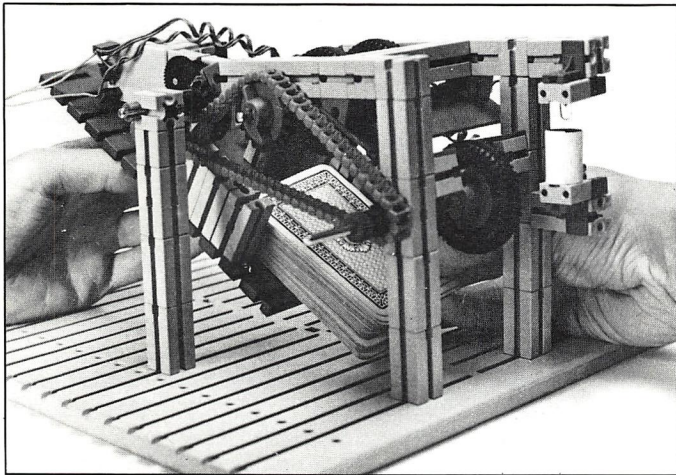
10 The top plate is fixed by two angled blocks, while the two smaller bottom plates are supported on two of the half-bricks that can rotate on their mountings (the ones with red protrusions). All wheels have rubber tyres fitted.



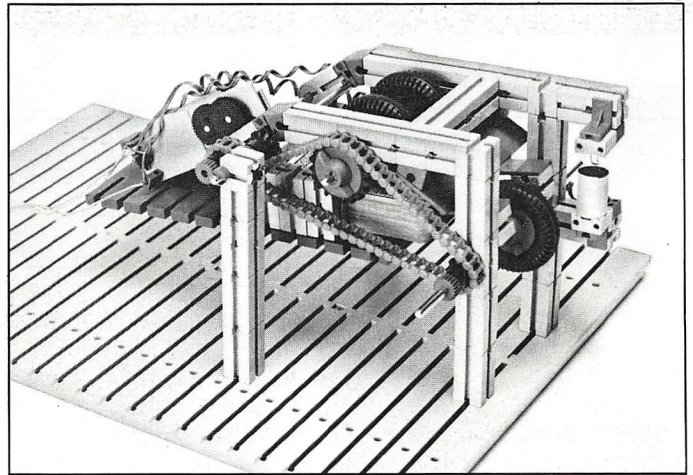
11 A small tube 2cm (3/8in) in length, made from rolled-up paper or card and coloured black on the inside, is mounted on the light detector. This makes the system less vulnerable to light from other sources.



12 The chain is now mounted as shown, passing under the topmost black gear, which acts as a tensioner. The bottom deflector plates can be adjusted using finger and thumb, to guide the cards properly.



13 The pack of cards is inserted by pivoting the holder upright. You will probably find it necessary to weight down the back of the holder using Plasticine to provide enough pressure on the driving wheels.



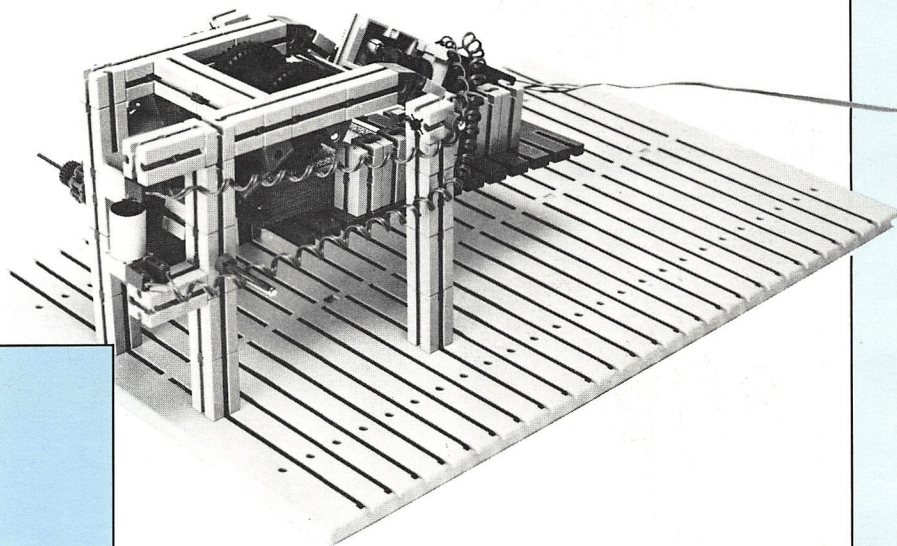
14 The finished dealer is wired up and its operation tested by hand before it is interfaced with the computer and the robot arm. The dealer can be mounted on a smaller baseplate if you have one.

Program. This was written for the Vic-20 but will work with the Commodore 64 by changing the value of PRT in line 30 to 56577. After initializing, the program prompts: PLAYER (1-5), TWIST (Y/N)? Responding with 3,Y will result in a card being dealt face upwards to player number three.

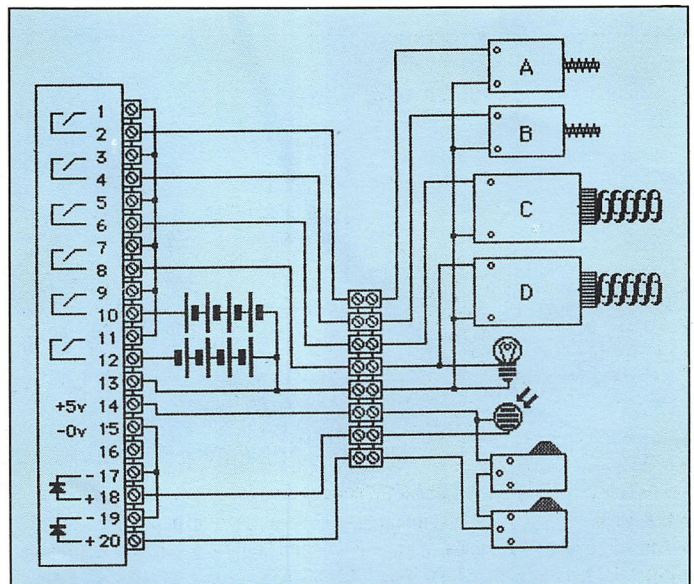
```

10 REM**CARD DEALER**
20 REM**INITIALIZATION**
30 PRT=37136:POKEPRT+2,63
40 PRINT"INSERT BATTERIES"
50 D1=16:D2=32:OFF=0
60 MA=1:MB=2:MC=4:MD=8
100 REM**BASIC PLAY**
110 CP=0:GOSUB1200:GOSUB1400
120 INPUT"PLAYER (1-5),TWIST (Y/N)";NP,T$
130 GOSUB1300:GOSUB1000
140 IFT$="Y"THENGOSUB1100
150 GOSUB1200:GOSUB1300
160 IFT$="Y"THENGOSUB1100
170 NP=0:GOSUB1000:GOTO110
1000 REM**ROTATE ARM TO POS NP**
1010 DR=SGN(NP-CP):IFDR=0THEN1070
1020 IFDR=-1THENPOKEPRT,MC+D1:GOTO1040
1030 POKEPRT,MC+D2
1040 IF(PEEK(PRT)AND64)=0THEN1040
1050 IF(PEEK(PRT)AND64)=64THEN1050
1060 CP=CP+DR:IFCP<>NPTHEN1040
1070 POKEPRT,OFF:RETURN
1100 REM**SPIN HAND**
1110 IFHP=1THENPOKEPRT,MB+D2:HP=0:GOTO1130
1120 POKEPRT,MB+D1:HP=1
1130 IF(PEEK(PRT)AND64)=0THEN1130
1140 IF(PEEK(PRT)AND64)=64THEN1140
1150 POKEPRT,OFF:RETURN
1200 REM**OPEN HAND**
1210 POKEPRT,MA+D2
1220 DL=1200:GOSUB2000
1230 POKEPRT,OFF:RETURN
1300 REM**CLOSE HAND**
1310 POKEPRT,MA+D1
1320 DL=800:GOSUB2000
1330 POKEPRT,OFF:RETURN
1400 REM**FEED CARD**
1410 POKEPRT,MD+D1:DL=150:GOSUB2000
1420 IF(PEEK(PRT)AND128)=0THEN1420
1430 POKEPRT,OFF:RETURN
2000 REM**DELAY DL STEPS**
2010 FORTD=0TODL:NEXTD:RETURN

```



Wiring. A rather elaborate wiring system so take care to link up the various units correctly. Note the wires that join with one another. The reason for the two microswitches being wired in series is given on page 5.



FREE-ROVING ROBOT

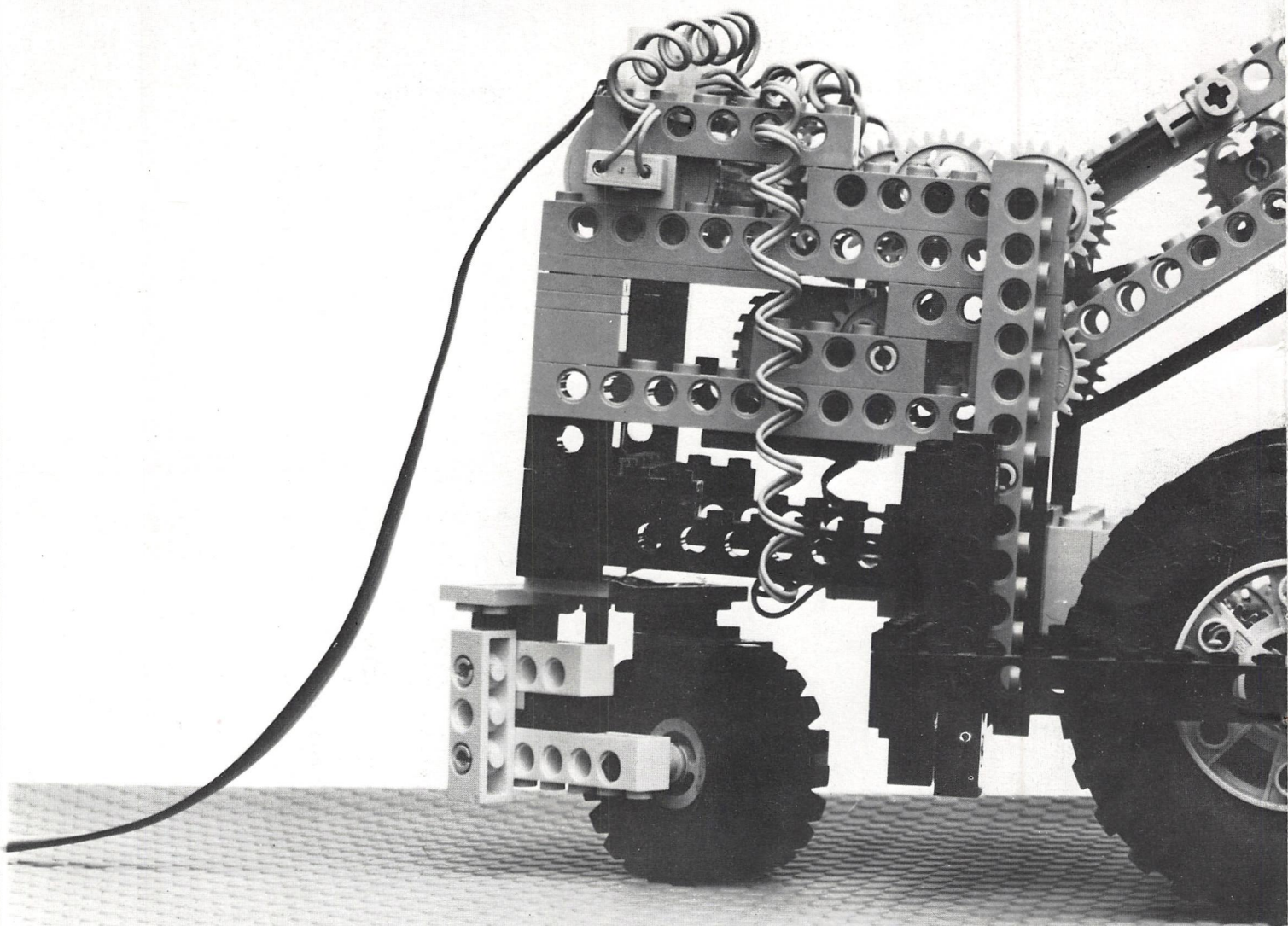
The intention of this project is to produce a general-purpose roving floor robot using Lego that can act as the base unit for a variety of robot superstructures. The first half of the project therefore shows how to construct the base unit, and this is followed by instructions for an elegant design of grabber arm that can be fitted to the base. However, the design of the base lends itself to many more possibilities, including a jointed arm, an android type of body and a vertical lifting arm like a fork-lift truck, and you can experiment with these ideas for yourself.

A general-purpose roving base has to be stable, controllable, rigid and easy to build onto. If it is to negotiate small steps, such as that at the edge of a carpet, then large rubber-

tyred wheels and high ground clearance are mandatory. The design shown is constructed largely from the Lego tractor kit, with a few additional pieces that include four motors. The two large wheels are independently driven to permit forward, reverse and turning motions. The jockey-wheel at the rear provides stability and allows the robot to turn on its own axis. It is important to keep the centre of gravity of any superstructure well behind the axis joining the two main wheels.

A gear reduction of 180 to 1 was found to be the optimum for the driving wheels; higher reduction would give a greater torque and load-bearing capacity but would make the movement frustratingly slow. To reduce the size of the vehicle and the backlash that results from long trains of gears, the motors are fitted into standard Lego 20:1 gearboxes. Gearbox sets feature two different units so two sets will be needed, but the other two gearboxes can be used to very good effect on the grabber arm.

Rigidity is the other major consideration for the base. The chassis shown can withstand a fair battering, and the base can



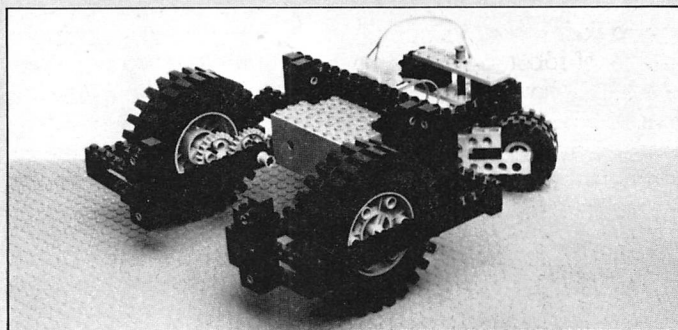
be grasped at any point without falling apart. This has been achieved by pinning anchor beams across the chassis at several points, a technique that can be extended to any Lego robot design.

Though unusual, the shape of the resulting base unit is very versatile. Electronics enthusiasts can mount collision detectors or other sensors on the front bumpers, and the well between the wheels allows an arm to reach down to and operate at floor level.

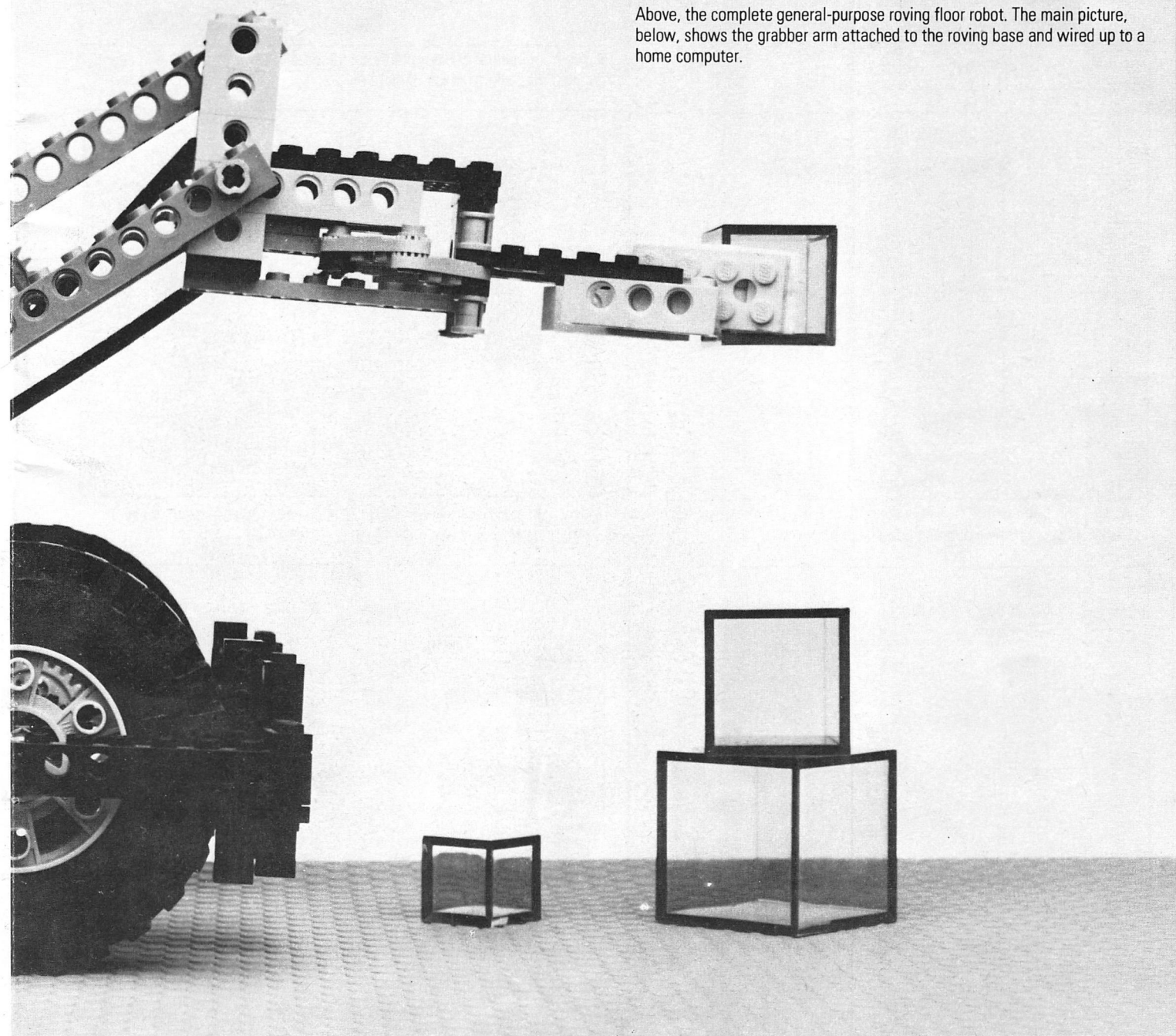
The arm design shown here is based on parallelogram geometry, so that the end effector remains level as the arm is raised. Raising the arm requires a considerable amount of torque, and in fact the motor is geared down by 540 to 1 with the partial help of a gearbox. The gripper itself follows very closely the design of many industrial end effectors. Power is transmitted through a tensioned chain and thence via a rack and pinion. The gripper is built from standard Lego pieces that form the steering mechanism on the larger vehicle kits.

Once constructed, the roving robot with arm can be pro-

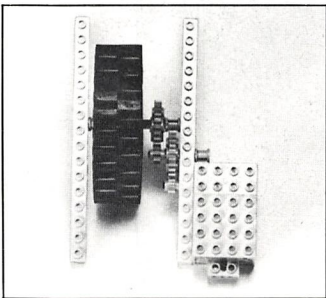
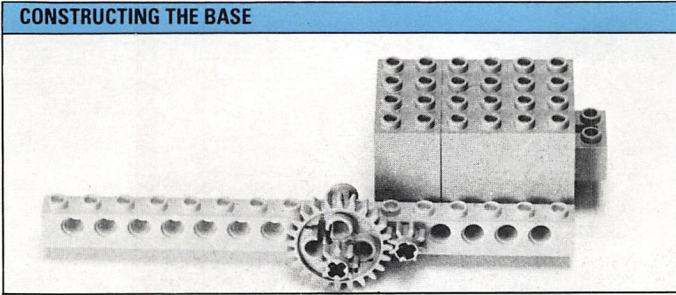
grammed to retrieve objects from anywhere in the room. Its main limitation is that it lacks any sensors or feedback, so precise movement cannot be achieved. The program listed is for controlling the robot live from the computer keyboard, but the subroutines for driving the motors could be built into a larger program that memorizes and repeats a sequence of moves of a complexity of your choice.



Above, the complete general-purpose roving floor robot. The main picture, below, shows the grabber arm attached to the roving base and wired up to a home computer.

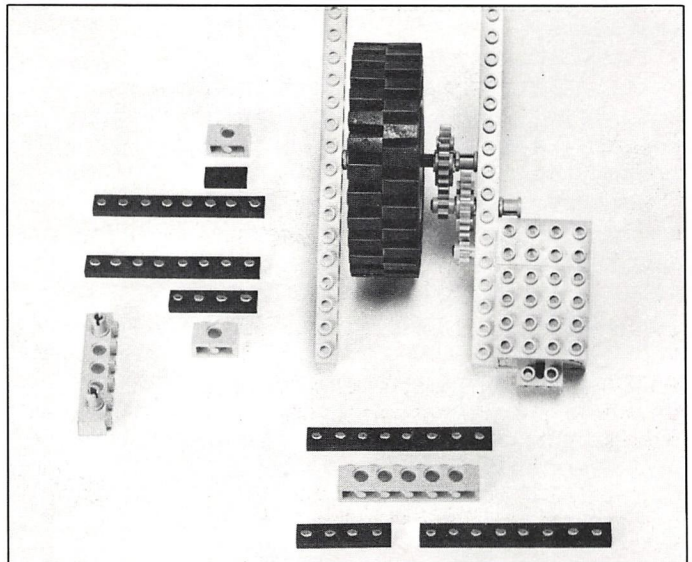


CONSTRUCTING THE BASE

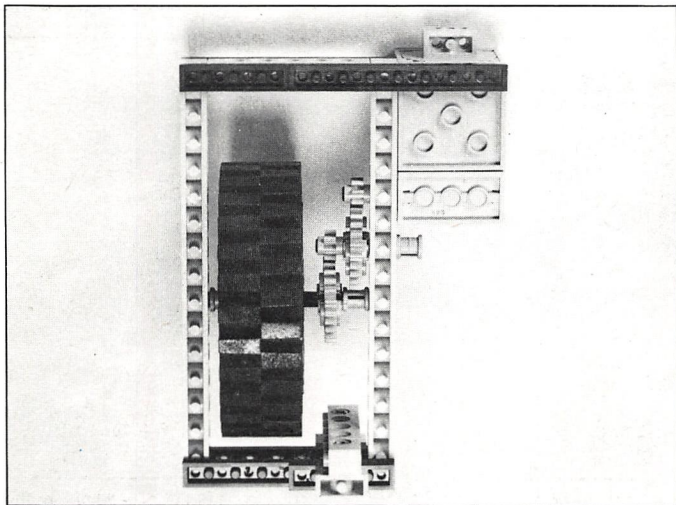


1 Above, a 6-unit-long axle fits into a square gearbox and motor and thence through a 16-unit-long beam. Gear wheels – 24- and 8-tooth – and bushes are added to axles as shown.

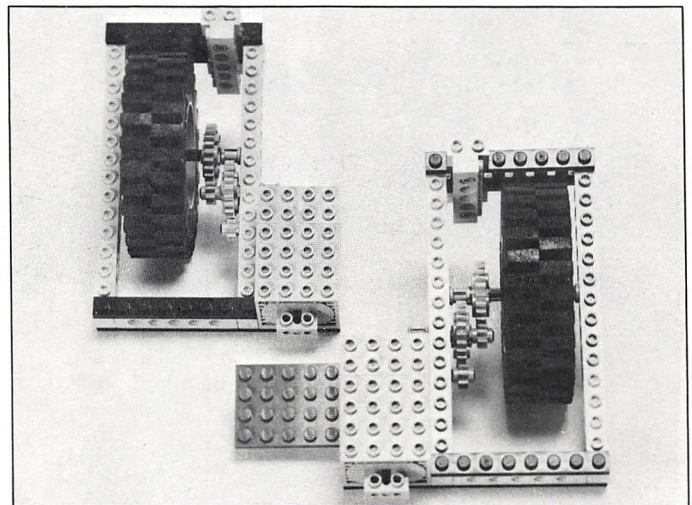
2 Left, the large tractor wheel sits on an 8-unit axle. The total gear reduction is 180 to 1.



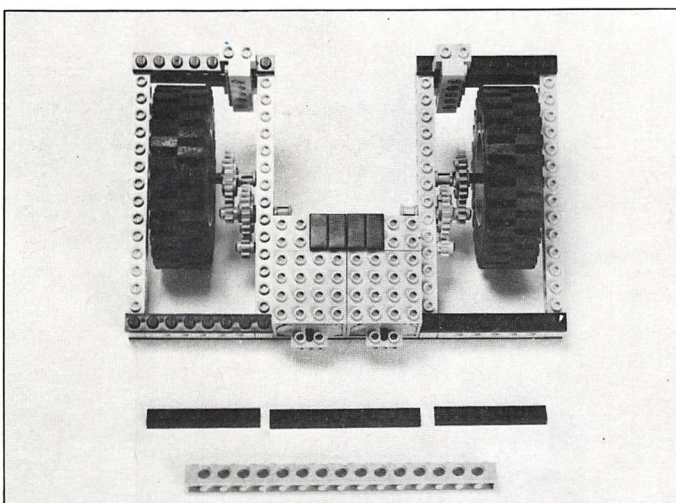
3 The front bumper of black plates is strengthened by a vertical 6-unit beam and two pins, shown in left of picture.



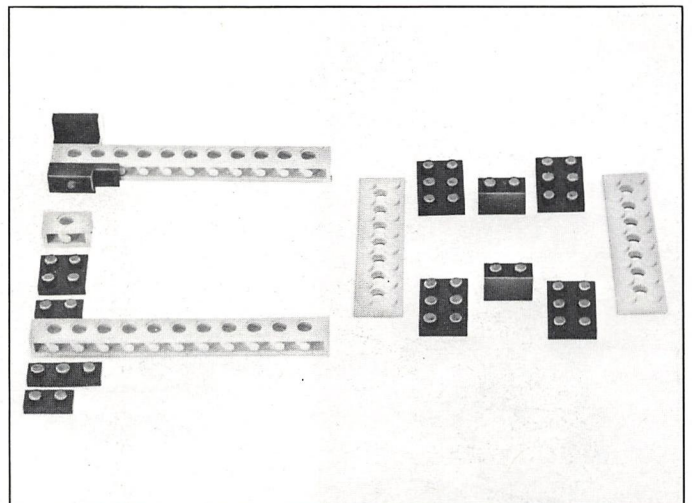
4 At the rear of the wheel assembly the bottom line of thin plates extends sideways to support the motor and gearbox unit.



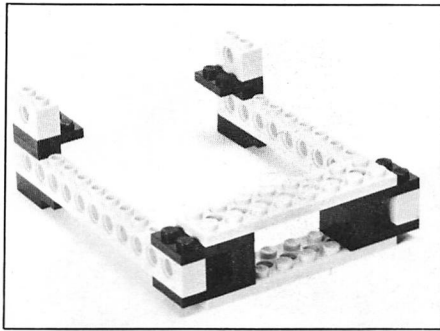
5 The right-hand drive wheel assembly is constructed as a mirror image. A 4x10 plate holds the two units together.



6 Long beams and plates fit underneath the motors. Small plates can be added on top to take a wiring block.

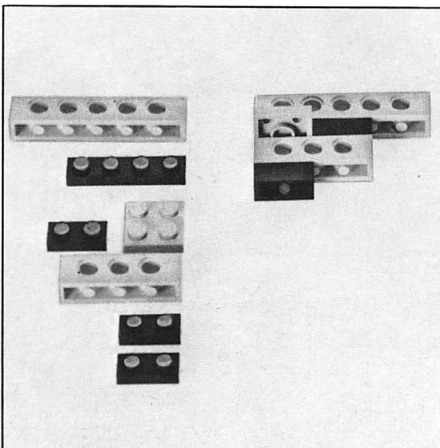
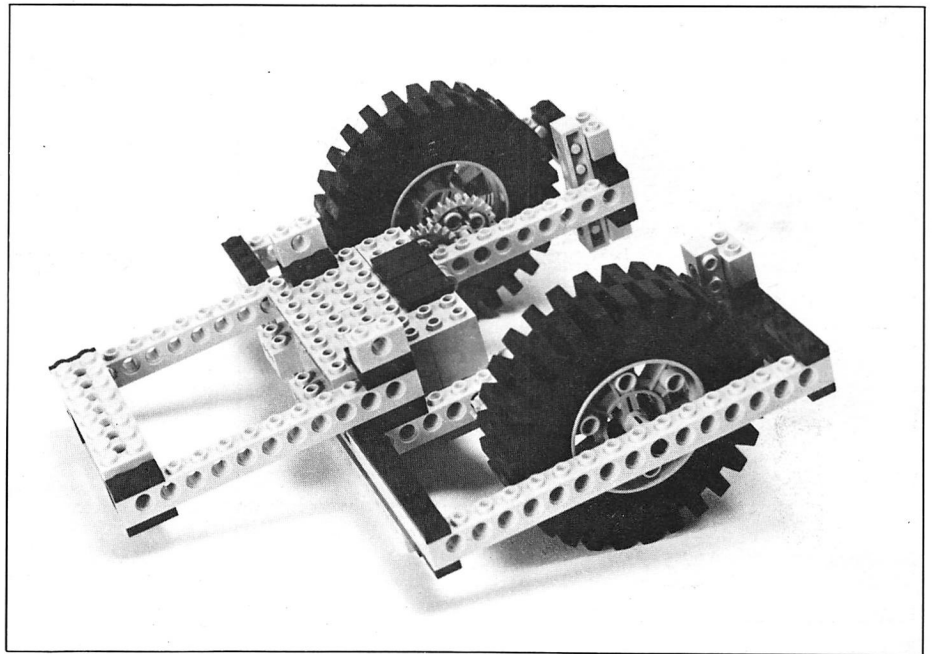


7 The rear chassis is built from two 12-unit beams, four 2x1 blocks and plates as follows: two 2x8, four 2x3, two 2x2, four 2x1 and two 3x1.

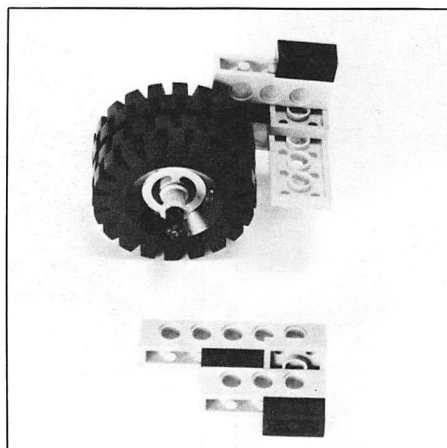


8 The long beams form the sides of the rear chassis, the blocks and plates the cross-beam and side supports.

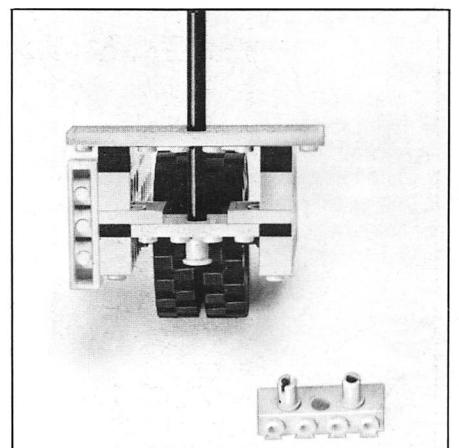
9 The rear chassis then fits onto the main chassis, either side of the gearboxes. Further strengthening will be added later.



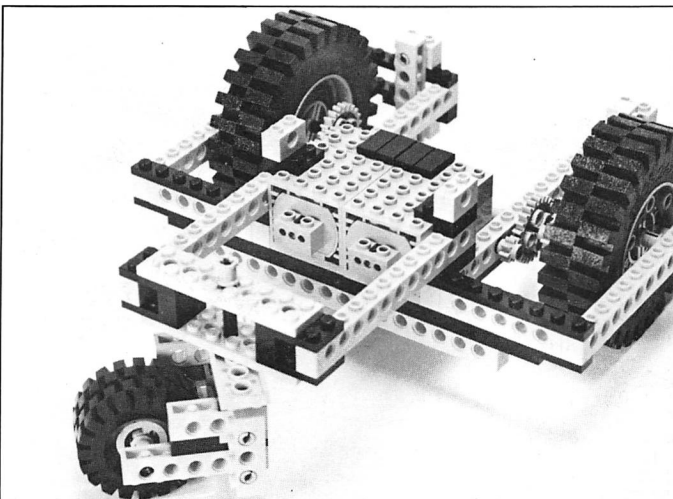
10 The two sides of the jockey-wheel support are constructed as mirror images, using beams and plates.



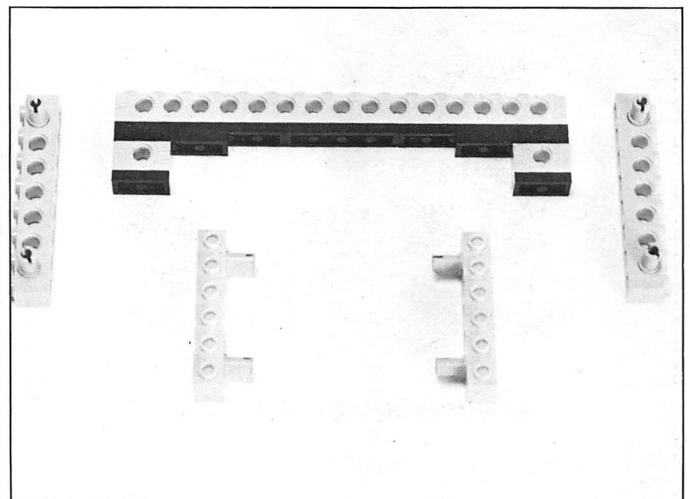
11 The two 44mm-diameter wheels fit on a 6-unit axle with two collars. A 4x2 plate holds the side supports together.



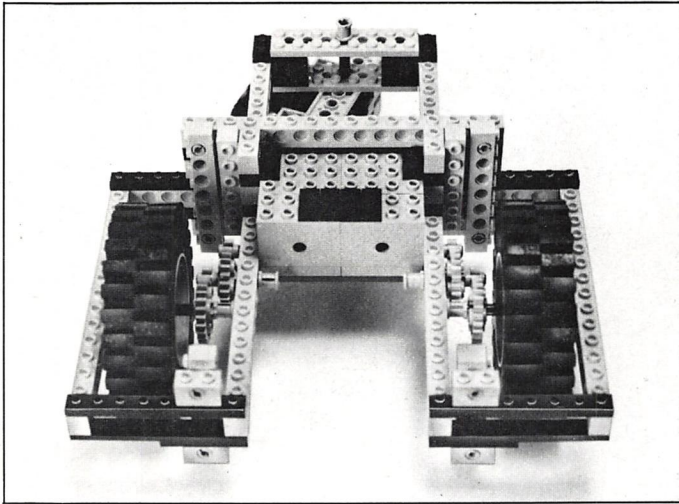
12 An 8x2 plate and an 8-unit axle complete the axle with two collars. Two 4x1 strengthening beams are pinned to the sides.



13 The jockey-wheel unit is fitted to the rear chassis and the axle secured with a collar, top and bottom.



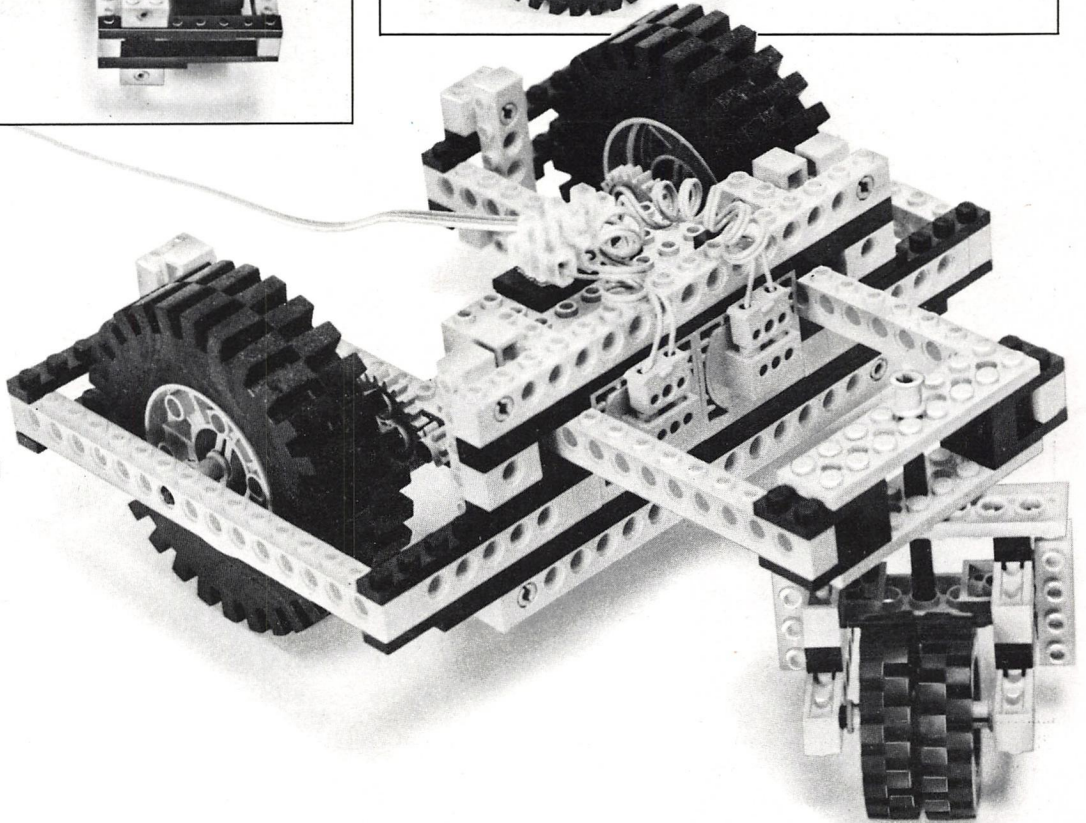
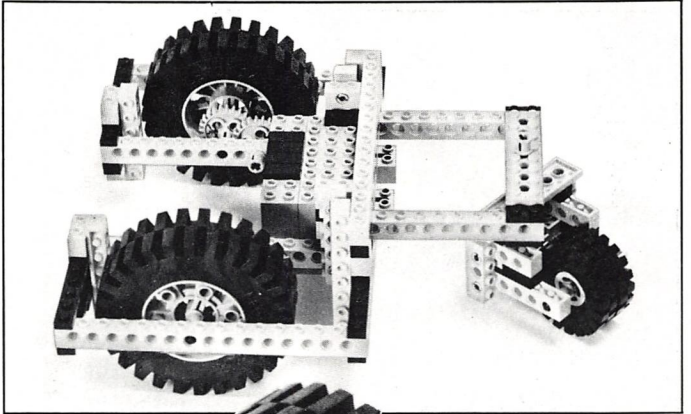
14 A bridge across the drive unit is formed from a 16-unit beam and several layers of plates and bricks as shown. The two vertical beams are pinned on either side to strengthen the structure.



15 Above, the bridge in position, attached to the motor units and side supports of the rear chassis (see step 8).

16 Above right, at this point the roving base, here seen from the side, is finished.

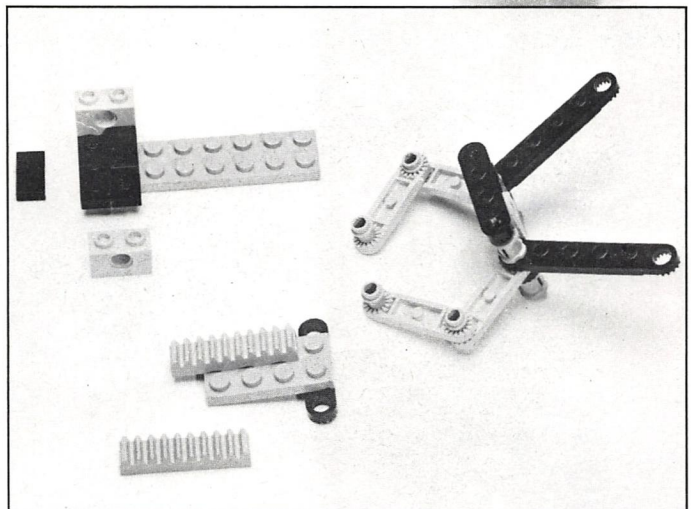
17 Right, the two driving motors can be wired up and the base used as a buggy. Alternatively, you can proceed to build on either a superstructure or the grabber arm shown in the following steps.



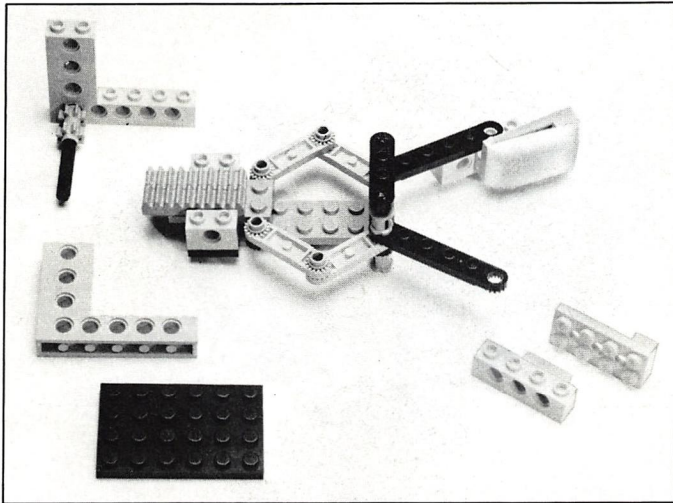
CONSTRUCTING THE END EFFECTOR



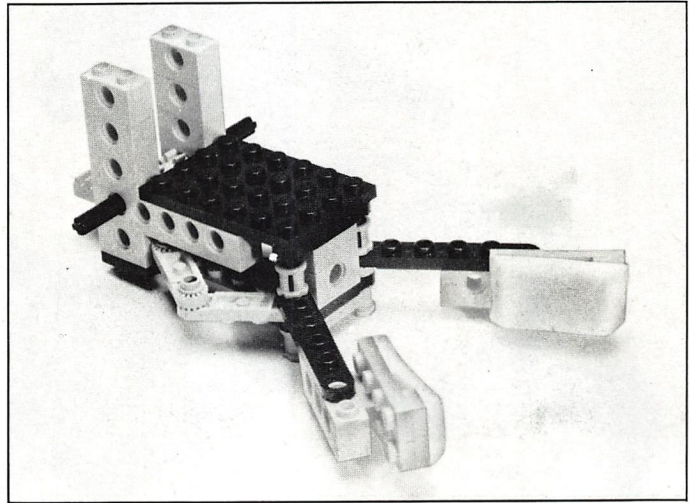
18 The fingers are made from steering links, which must all face in the directions shown here. One finger is shown made up, the other finger and the linkages as separate components.



19 Two smooth 2x1 plates sit on top of the black 4x2 plate for the racking to slide over. The bushes on the ends of the fingers push into holes in the 4x1 rounded plate.

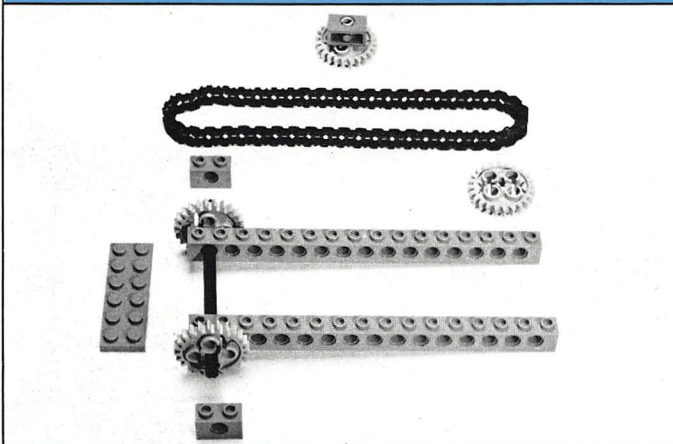


20 An 8-unit axle supports two 8-tooth gears and is held by a 6-unit beam on either side. The beams bear three 2×1 bricks. Grippers, bearing draught-excluder tape, are added to the fingers.

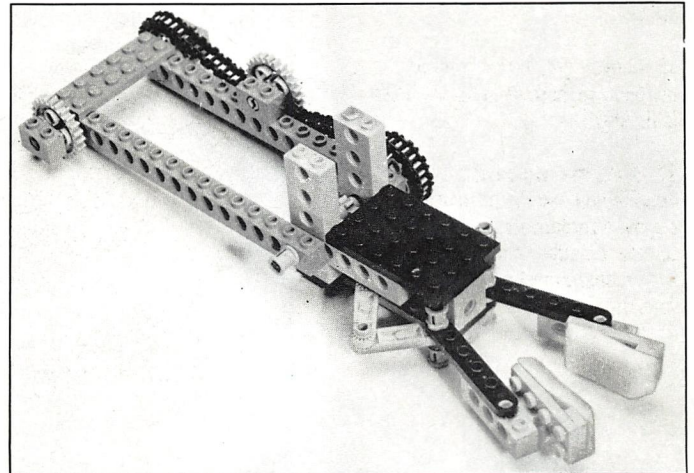


21 The complete end effector. A 6×4 plate holds the elements together.

BUILDING THE ARM

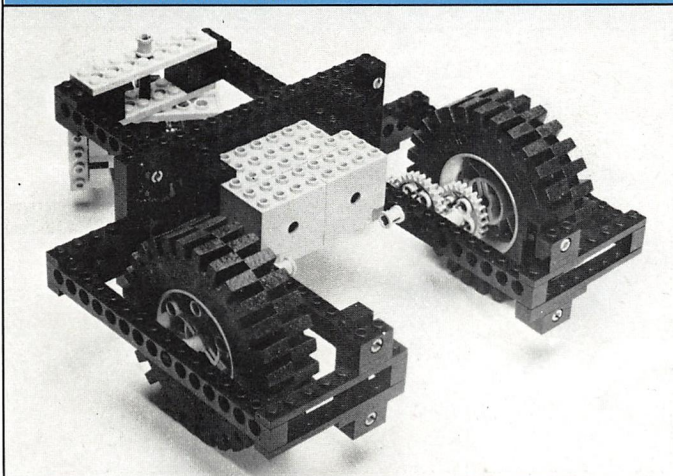


22 A 10-unit-long axle spans two 16-unit beams and is fitted at each end with a 24-tooth gearwheel and 2×1 brick. A 6×2 plate is placed above the axle.

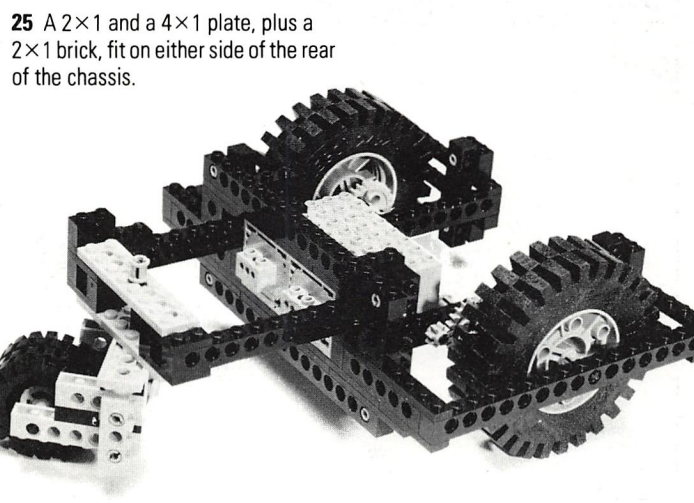


23 The end effector is fitted and a 48-link chain placed over the gearwheels. The middle 24-tooth gear is to tension the chain. The arm is secured to the gripper with a collar.

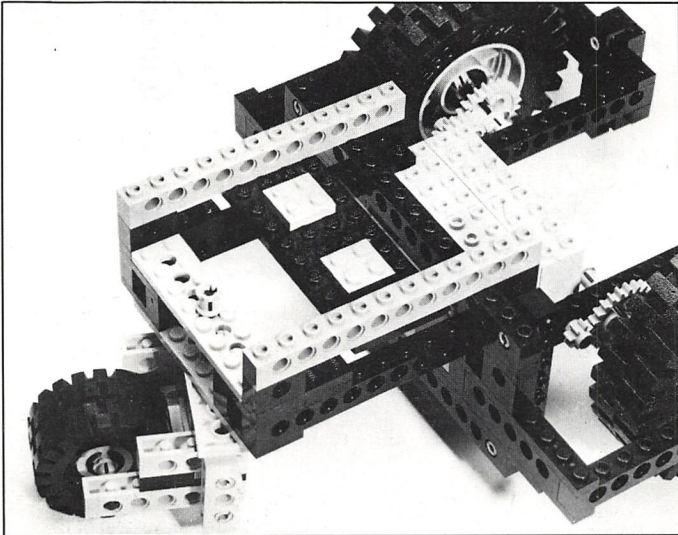
FITTING THE ARM TO THE BASE



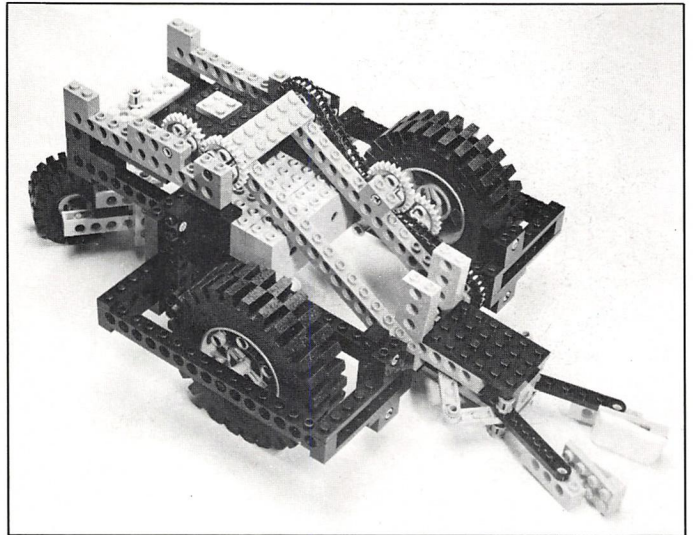
24 The two inner vertical pinning beams (see step 14) must be removed first, to be replaced by longer ones later on. (We have rebuilt the base in black for clarity.)



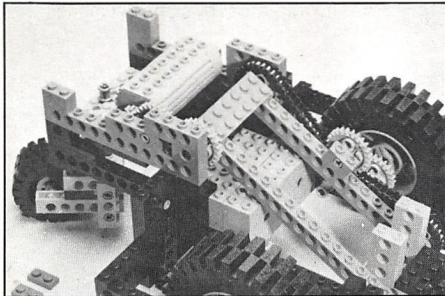
25 A 2×1 and a 4×1 plate, plus a 2×1 brick, fit on either side of the rear of the chassis.



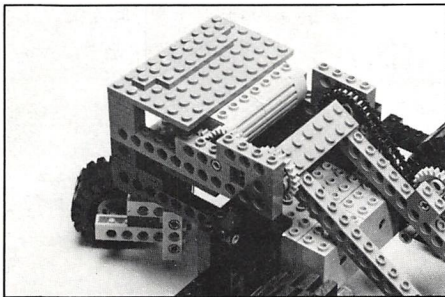
26 Two 12-unit beams form the base for the arm.



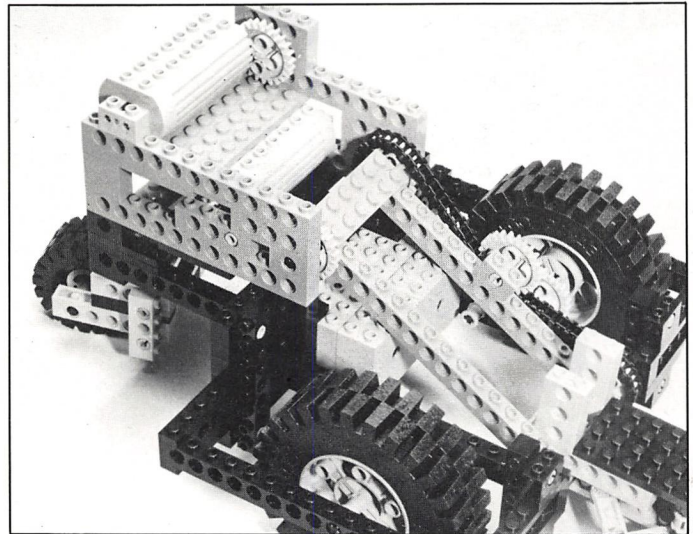
27 After fitting the arm, additional bricks and beams and a 24-tooth gear are fitted on the same level.



28 Left, the motor and gearbox feature an 8-tooth gear. Two 2×1 plates fit on the 2×1 bricks each side at the rear.

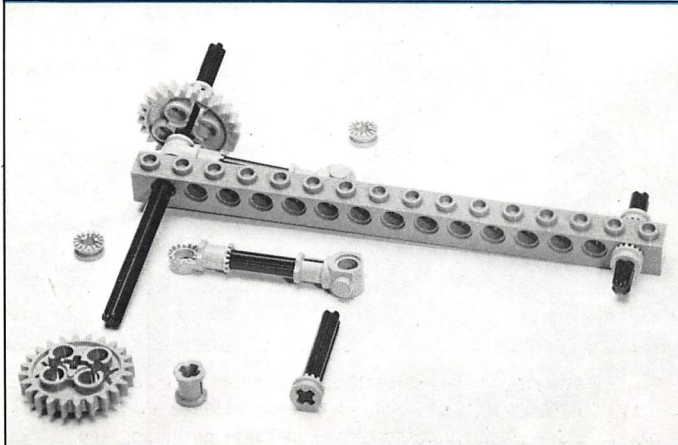


29 Left, plates totalling 10×6 units sit on top of the motor, followed by another ridge of plates at the rear.

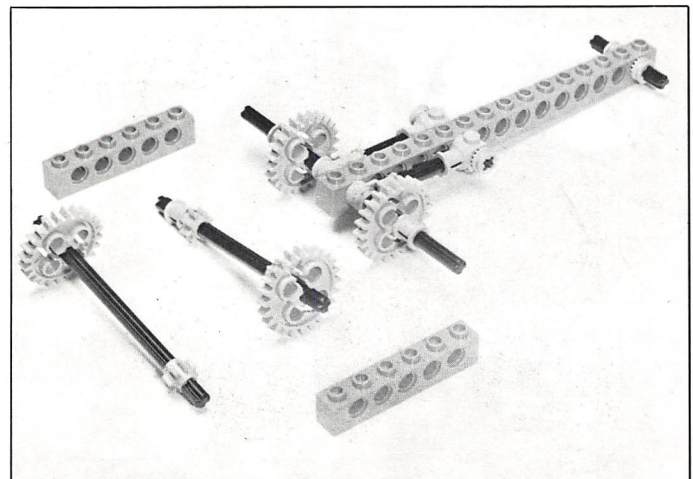


30 The second motor and gearbox features an 8-tooth gear meshing with the 24-tooth gear shown.

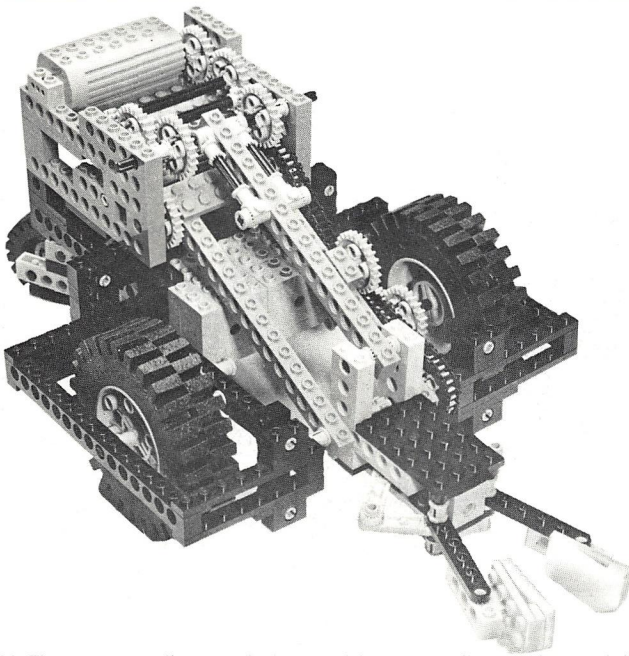
CONSTRUCTING THE UPPER ARM



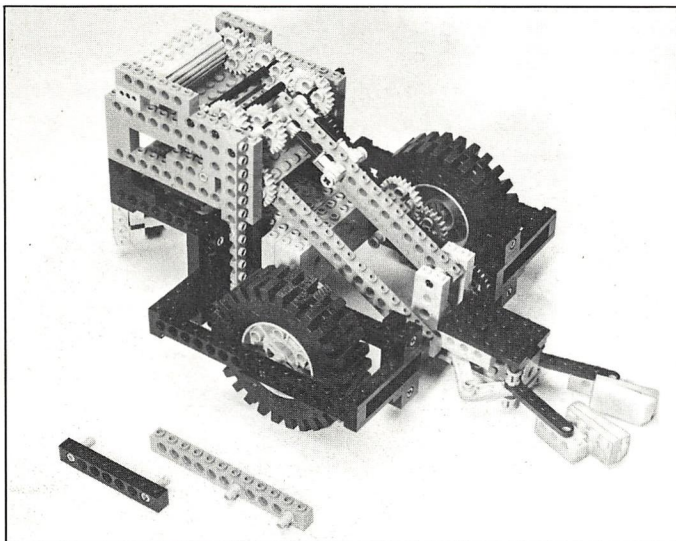
31 This is constructed symmetrically around a 16-unit holed beam, using four 4-unit and one 10-unit axles fitted with gears and bushes.



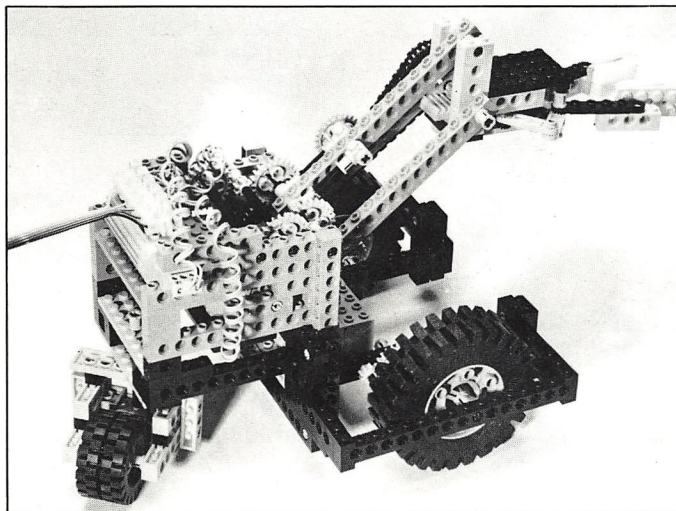
32 Two further 10-unit axles, fitted with 8- and 24-tooth gears and spanning 6×1 beams, complete the gearing system.



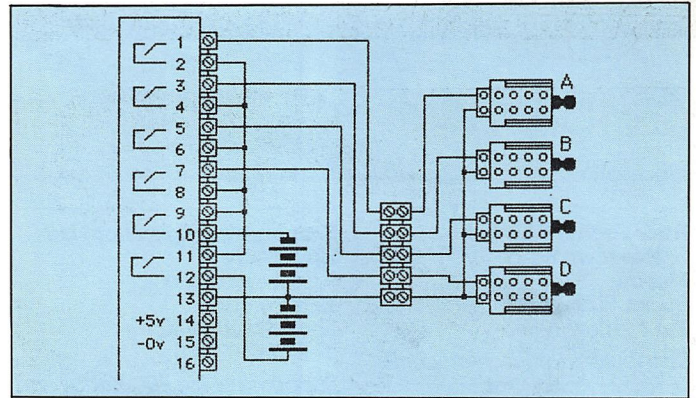
33 The upper arm clips onto the base and, by temporarily removing two bricks, onto the gripper.



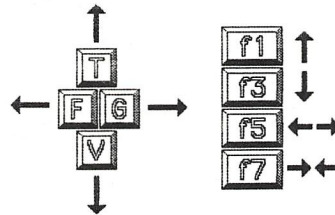
34 The 8-unit vertical beams are temporarily removed and 12-unit beams are pinned vertically either side (see steps 14, 24).



35 The completed base and arm are ready for wiring up. The rover can be driven from four battery boxes or linked to a computer as shown in the wiring diagram.



Wiring diagram. Motors A and B drive the right- and left-hand wheels respectively, C is the gripper motor, and D raises and lowers the arm. THE PROGRAM SHOULD BE LOADED AND RUN BEFORE THE BATTERIES ARE INSERTED INTO THE BATTERY HOLDER.



Driving the robot. The T, F, G and V keys control the movement of the buggy as shown, with the arm being controlled by the four function keys. F5 and F7 open and close the gripper. Releasing a key stops the motion.

```

10 REM**ROVING ROBOT CONTROLLER**
20 REM**INITIALIZE**
30 PRT=37136:POKEPRT+2,63
40 PRINT"INSERT BATTERIES"
50 DIMKEY(8):FORI=1TO8:READKEY(I):NEXTI
60 MA=1:MB=2:MC=4:MD=8:D1=16:D2=32:OFF=0
100 REM**CONTROL**
110 SCAN=PEEK(197):T=9:FORI=1TO8
120 IFSCAN=KEY(I)THEN T=I:I=9
130 NEXTI:IFT=9THENPOKEPRT,OFF:GOTO110
140 ONTOSUB200,250,300,350,400,450,500,550
150 GOTO110
200 REM**DRIVE FORWARDS**
210 POKEPRT,MA+MB+D1:RETURN
250 REM**DRIVE BACKWARDS**
260 POKEPRT,MA+MB+D2:RETURN
300 REM**TURN RIGHT**
310 POKEPRT,MA+D2:GOSUB1000
320 POKEPRT,MB+D1:GOSUB1000
330 POKEPRT,OFF:RETURN
350 REM**TURN LEFT**
360 POKEPRT,MB+D2:GOSUB1000
370 POKEPRT,MA+D1:GOSUB1000
380 POKEPRT,OFF:RETURN
400 REM**RAISE ARM**
410 POKEPRT,MC+D1:GOSUB1000
420 POKEPRT,OFF:RETURN
450 REM**LOWER ARM**
460 POKEPRT,MC+D2:GOSUB1000
470 POKEPRT,OFF:RETURN
500 REM**OPEN GRIPPER**
510 POKEPRT,MD+D1:GOSUB1000
520 POKEPRT,OFF:RETURN
550 REM**CLOSE GRIPPER**
560 POKEPRT,MD+D2:GOSUB1000
570 POKEPRT,OFF:RETURN
1000 REM**DELAY**
1010 FORTD=0TO20:NEXTD:RETURN
10000 DATA50,27,19,42,39,47,55,63

```

This listing is for a Vic-20 computer with VicRel relay unit. In line 60, MA TO MD correspond to the control bits for motors A to D in the wiring diagram. PEEK(197) in line 110 returns a value corresponding to the key being pressed. The following lines check this value against the codes for the allowable keys, which have been read into KEY(I) from the DATA statement. Other computers can emulate this function using GET or INKEY.

USEFUL ADDRESSES

Artur Fischer (UK) Ltd

25 Newtown Road
Marlow
Bucks. SL7 1JY
Tel: Marlow 72882

Distributors for Fischertechnik

British Robot Association

35-39 High Street
Kempston
Beds. MK42 7BT
Tel: 0234 854477

*Publish useful newsletters,
particularly for industry*

Commoton

241 Green Street
Enfield
Middx. EN3 7SJ
Tel: 01-804 1378

*Suppliers for Beasty interface,
Snap vision system for BBC
Micro and catalogue of robot
building materials*

Colne Robotics Ltd

Beaufort Road
Off Richmond Road
East Twickenham TW1 2PQ
Tel: 01-892 8197

*Armdroid - educational robot
arm*

Cyber Robotics Ltd

61 Ditton Walk
Cambridge CB5 8QD
Tel: 0223 212612

*Cyber 310 - educational robot
arm, Votrax speech synthesizers*

The Droid Factory

Unit 14
70 Wootton Road
Abingdon OX14 1LD
Tel: 0235 30663

Promotional androids

D.C.P. Micro Developments

2 Station Close
Lingwood
Norwich NR13 4AX
Tel: 0223 833902

*InterSpec and InterBeeb relay
units*

Economats (Education) Ltd

Epic House
9 Orgreave Road
Handsworth
Sheffield S13 9LQ
Tel: 0742 690801

*Suppliers of the BBC Buggy and
educational distributors of
construction kits such as
Fischertechnik, Meccano and
Lego*

Feedback Instruments

Park Road
Crowborough
East Sussex TN6 2QR
Tel: 08926 3322

*Educational robot arms and
school experiment kits*

Handic Software

5 Albert Road
Crowthorne
Berks. RG11 7LT
Tel: 0344 778800

VicRel - relay unit

IFS (Publications) Ltd

35-39 High Street
Kempston
Bedford MK42 7BT
Tel: 0234 853605

*Wide range of technical robot
publications and conference
proceedings*

Jessop Microelectronics

Unit 5
7 Long Street
London E2
Tel: 01-729 1851

Jessop Turtle

Lego (UK) Ltd

Wrexham
Clwyd LL13 7TQ
Tel: 0978 266949

Maplin Electronics

P.O. Box 3
Rayleigh
Essex SS6 8LR
Tel: 0702 552911-
Hero 1 - kit form

Market Shooters

7 Hazlewell Road
London SW15
Tel: 01-788 0056

*Promotional robots and
androids*

Microrobotic Systems

Tel: 01-370 0726

Fischertechnik Robot Kit suppliers

Milton Bradley

Spencer House
23 Sheen Road, Richmond
Surrey TW9 1AL
Tel: 01-940 6069

Manufacturers of Big Trak

The Original Android Co. Ltd

The Royal Victoria Patriotic
Building
Trinity Road
London SW18
Tel: 01-870 6221

Promotional robots

Practical Robotics

E.C.C. Publications
196-200 Balls Pond Road
London N1 4AQ
Tel: 01-359 3525

*Newsstand distributed
magazine*

Prism Robots

Prism House
18-29 Mora Street
London EC1
Tel: 01-253 2277

*UK distributors for Androbot
and Movits*

Probotics

April Cottage
Post Office Square
Siddington, Cirencester
Tel: 0285 61579

Promotional robots

Promotional Engineering

214 Ash Grove
Heston, Middx.
Tel: 01-570 4889

Promotional robots

RH Electronics

Chesterton Mill
French's Road
Cambridge CB4 3NP
Tel: 0223 311290

*Vision system for BBC micro
using conventional TV camera*

Rhino Robots Inc.

3402 N. Mattis Avenue
P.O. Box 4010
Champaign
IL 61820
USA
Tel: (217) 352-8485

*Educational arms and floor
robots*

Robotics Age Inc.

Strand Building
174 Concord Street
Peterborough
NH 03458
USA
Tel: (603) 924-7136

*Robotics Age Magazine
(monthly)*

Systems Control

Perry Avenue
Teesside Industrial Estate
Thornaby TS17 9LN
Tel: 0642 764823

Educational robot arms

Tomy (UK) Ltd

Wells House
231 High Street
Sutton
Surrey SM1 1LD
Tel: 01-661 1547

*Tomy toys - ROBO 1,
Verbot etc.*

Valiant Designs Ltd

Park House
140 Battersea Park Road
London SW11 4NB
Tel: 01-720 3947

Valiant Turtle

World of Robots

55 East Parade
Harrogate
North Yorks. HG1 5LQ
Tel: 0423 57864

Promotional Robots

Your Robot

2nd Floor
155 Farringdon Road
London EC1R 3AD
Tel: 01-833 0531

Magazine

Zenith Data Systems Ltd

Bristol Road
Gloucester GL2 6EE
Tel: 0452 29451

Hero 1 - assembled form

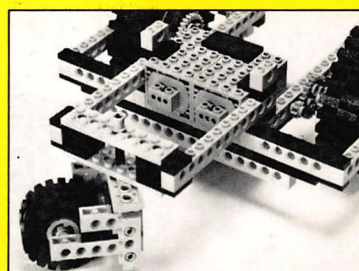
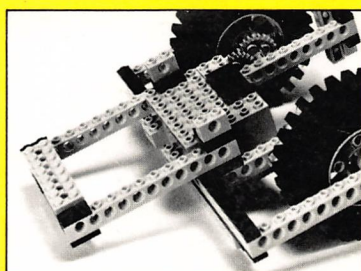
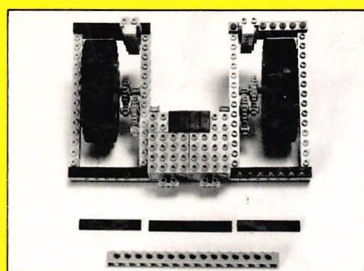
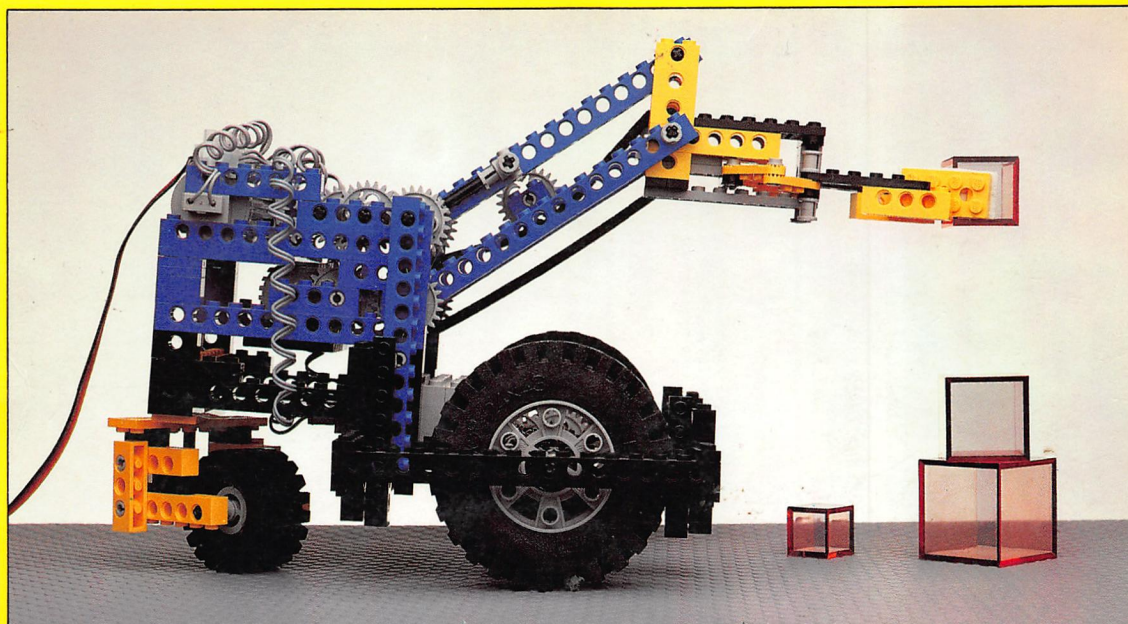
Copyright © Frances Lincoln Limited 1985
Copyright text © Richard Pawson 1985
Copyright artwork © Frances Lincoln Limited 1985

Windward
an imprint owned by W H Smith & Son Limited
Registered No 237811 England
Trading as WHS Distributors,
St John's House, East Street,
Leicester LE1 6NE

ISBN 0 7112 0419 5
Printed and bound in Yugoslavia by
Mladinska knjiga, Ljubljana

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission of the publishers.



ROBOT PROJECTS

- is a comprehensive, step-by-step guide to practical robotics.
- contains full constructional and programming details for projects that introduce basic mechanics, construction techniques and circuit design for robot devices.
- uses widely available educational construction kits, Fischertechnik and LEGO®.
- includes robot devices that are programmable and can be linked to any popular home computer.
- includes full wiring details and demonstration programs for projects.
- does not require any special tools or skills.
- presents a fascinating educational hobby, combining the skills of a computer programmer with those of a model engineer.

ISBN 0-7112-0419-5



9 780711 204195

THE ROBOT BOOK

Robot Projects is a compilation of selected projects from *The Robot Book* by Richard Pawson.

The Robot Book is concerned with understanding these fascinating machines. It provides a beginner's guide to the whole subject of robots and robotics, from toy and educational robots to the ones used in industry and in space exploration. It examines the origins of robots as well as providing an informed look at where robots are going in the future.

The Robot Book also contains details of home robot projects you can try out for yourself. Robots you can build. Robots you can buy and use. Robot devices you can link to a home computer.

The robot revolution is a fact. Understanding robots is important – for you and for your children.

Contents

Part 1 History of Robots

The origins and development of robots in fact and fiction

Part 2 Robots in Action

The types and applications of robots in use today

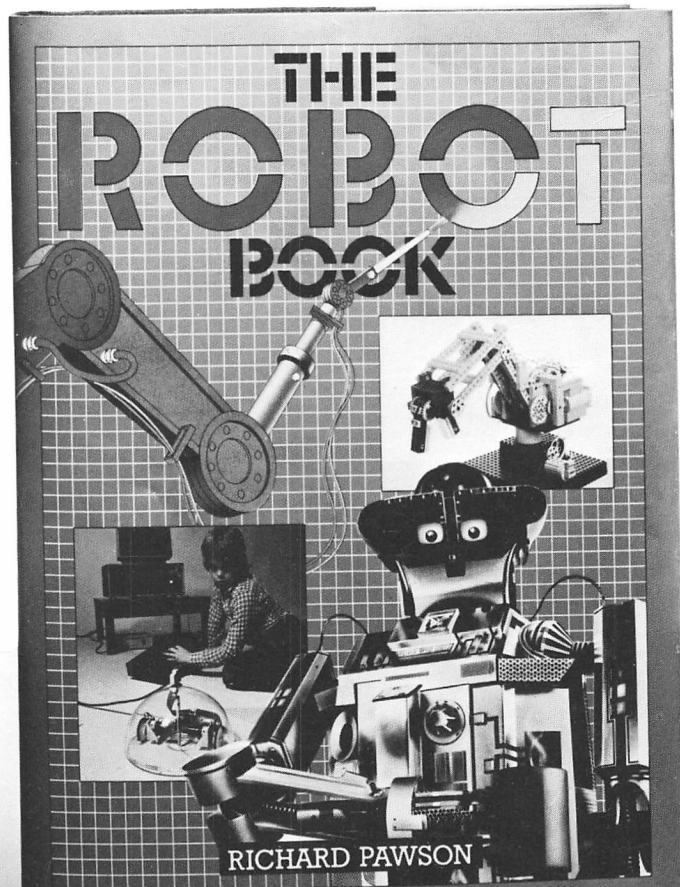
Part 3 How Robots Work

Robot theory and technology for the home robot user or experimenter

Part 4 Practical Projects

Robot devices to build and to control with a home computer

Glossary of robotics terms



The Robot Book is:
published by Windward

ISBN 0 7112 0413 6 (Hardback)
0 7112 0414 4 (Softback)

Price £12.95 Hardback
£7.95 Softback