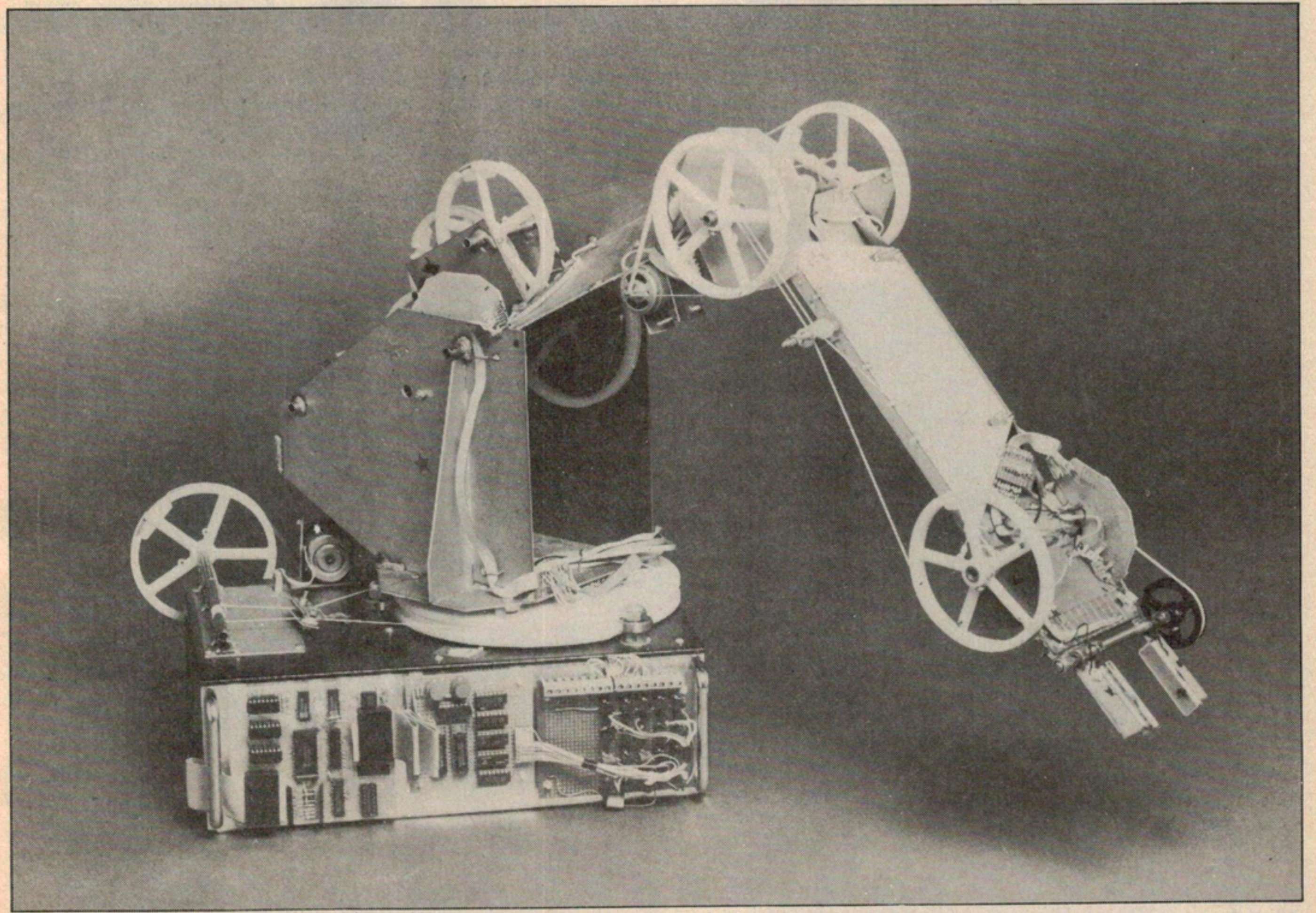
Algernon

a low cost robot for any 8-bit computer Pt.3

Here's the third and final article on Algernon, the low cost robot fabricated from copper clad phenolic board. In this, we wrap up details of the circuit assembly and review the control program.

The development of this project by the designer, Ray Pakalns, together with the sponsoring company, Jaycar Electronics, has proved to be a much larger undertaking than was initially foreseen. It is only now that the amount of work necessary to complete the project has been realised.



This photo shows the original prototype with the PCBs mounted on the outside of the base for access.

Algernon

Finally though, the development work on Algernon is complete, at least as far as the mechanicals and the interface details for the Microbee are concerned. But it is clear that the project has lots of potential for further development to allow it to be interfaced with other 8-bit computers. There is also a lot of scope for software development, to suit it to various computers and to produce a variety of operating routines.

At the time of writing this final article, Ray Pakalns has just completed the comprehensive assembly manual for the project. It is a very creditable production running to over 80 foolscap pages in which every facet of the project is discussed. This manual will be available on its own from Jaycar stores at the very reasonable price of $10.00. This gives all the information needed to build the robot except for the copper patterns of the sensor discs and circuit boards.

Included in the manual are the full cutting and assembly details for the various copper clad sections of the robot and we can well imagine some constructors buying just the manual, the essential PC boards and some of the hardware such as the dial drive pulleys and motors and then building the rest of the robot completely from scratch.

We certainly wish that we could have had access to the manual when we assembled our prototype because it answers so many questions that we had to resolve the hard way, by trial and error.

Ray Pakalns has also written a preliminary program in Basic to put Algernon through a series of simple movements, controlled via the Microbee. Ultimately, Jaycar will be supplying a more refined version of this program, written largely in machine code. This will greatly increase the operating speed of the robot which otherwise tends to be on the leisurely side.

With the simple routines on the supplied program it will be possible for constructors to develop their own operating programs to put Algernon through complex routines.

A facility to control the speed of each motor is also included in the supplied program, with speeds 0 to 9 available. The lowest setting means that the motor will only creep in a series of short pulses, while the speed of the highest setting is determined by the load and gear ratio.

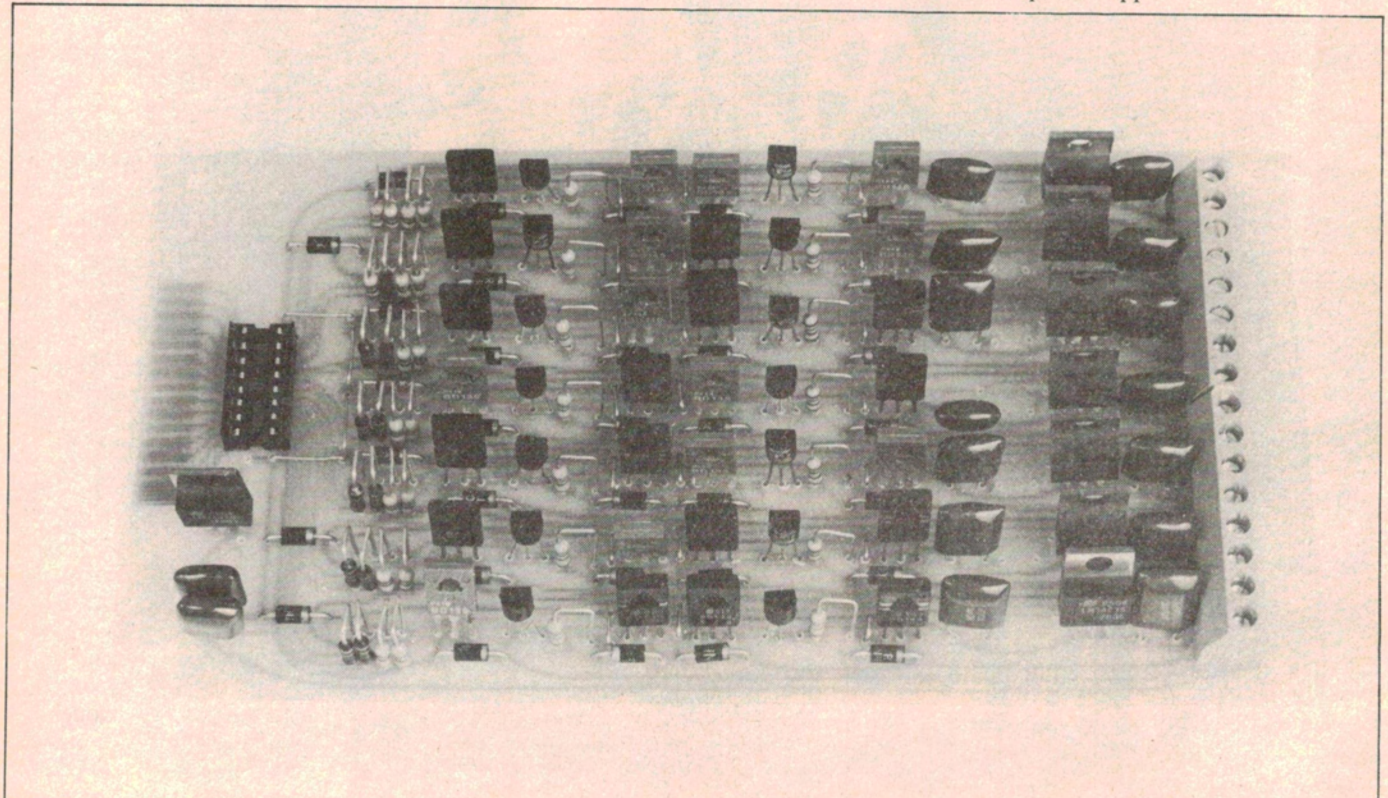
In setting a sequence of movements, the programmer must manually operate the robot to every required position in the sequence. This is done via the Microbee keyboard.

At each desired position, operating the "S" (store) key of the keyboard memorizes the position. What is memorized is the coded data from each of the sensor discs. As the data for most of the discs is picked up through three pin contacts, it is possible that one of the pins may be between pads on the disc. This would lead to an invalid input. The program will not permit this, causing the joint to shuffle minutely until a proper code is generated.

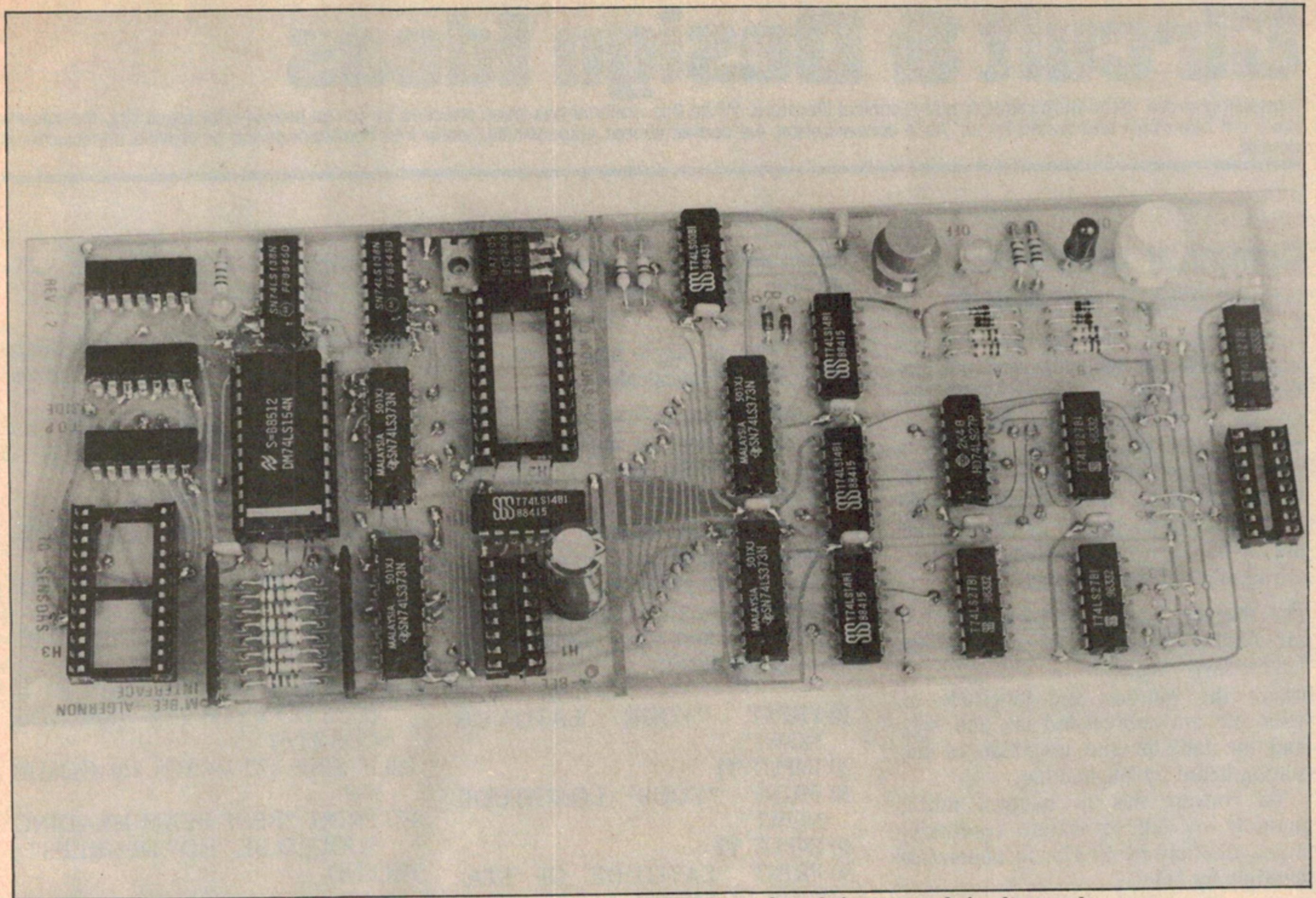
An initial program written to make Algernon pick up a small box brought to light an interesting problem. The eight-bit gripper sensor did not have sufficient resolution to grasp the box properly. The robot attempted to close the grippers too tightly, so that eventually, the rubber band drive acted as a clutch, with the motor continuing to spin.

The problem here was due to the fact that the program did not have any conditional statement. It could look at either the gripper position or the tactile sensor output, but it could not combine data from both. The net result was that it attempted to close the grippers too tightly.

With the final program, this problem will be solved with a control sequence which will close the grippers to the specified position unless the tactile sensor output is tripped.



All of the connections to the motor control board are made through a 16-pin header and a 16-way terminal block.



The interface board includes interlock circuitry to prevent forward and reverse from being engaged simultaneously.

Interface and control circuits

The interface and control circuits are reasonably straightforward. Since the Microbee performs all the decoding and sequencing, the Algernon electronics are only needed for buffering, interlock and power.

There are actually two PCBs which must be built up for the robot — one is the power board with the motor drive transistors and the other is the interface circuit. The interface board is the more complicated of the two, being a double-sided board measuring 224 x 90mm. Two main functions are performed by this board, multiplexing the coded sensor information and latching motor control.

Without multiplexing, the amount of wiring and circuitry required would otherwise get out of hand. As it is, an 8-bit data bus is shared by five sensors.

Multiplexing is performed by a 74LS154 4-line to 16-line decoder. Depending on the 4-bit input code, one of 16 outputs will be enabled. Only 13 of the 16 outputs are used in the standard robot: three outputs are connected to each of the waist, shoulder and elbow sensors; two are connected to the wrist; and one each to the gripper and touch sensors.

It is advisable to wire the robot as per the manual to avoid erroneous multiplexing signals. With the correct wiring, the computer, via the 74LS54, sequentially enables all of the sensor wipers.

Data is fed via eight diodes on each sensor disc to the data bus. This is connected directly to the Microbee parallel input bus. The program analyses the input data and performs the necessary calculations to determine each sensor position.

The Microbee control signals are latched by two 74LS373 octal Tristate latches. Two of these are needed to latch the 16 control lines.

To prevent forward and reverse from being engaged simultaneously for each motor, the latched outputs are fed through an interlock circuit. This consists of three 74LS14 hex Schmitt triggers in conjunction with three 74LS27 triple 3-input NOR gates.

The low speed options on the board are provided by two oscillators each built around 74LS14 Schmitt triggers. By tailoring the component values the particular duty cycle can be selected to suit any requirement. Each of the low speed options can be used with up to three of the motors. Keyboard controls are used to select the required low speed mode.

The motor control board (180 x 95mm) is single sided. It has 42 transistors and eight regulators, not to mention a few other parts. The board is rather neat, with all of the components set out in rows. Further, connections are made either through a 16-pin header or 16-way terminal block.

Our version of the Algernon had the interface board on the outside of the metal box to allow access. The recommended method though is to mount the boards inside the box — it's a lot neater.

The circuits of the interface and control boards are fully detailed in the manual described above. Although a low cost robot, the design has certainly meant a big investment in time and money. We think it will be very popular with computer enthusiasts.