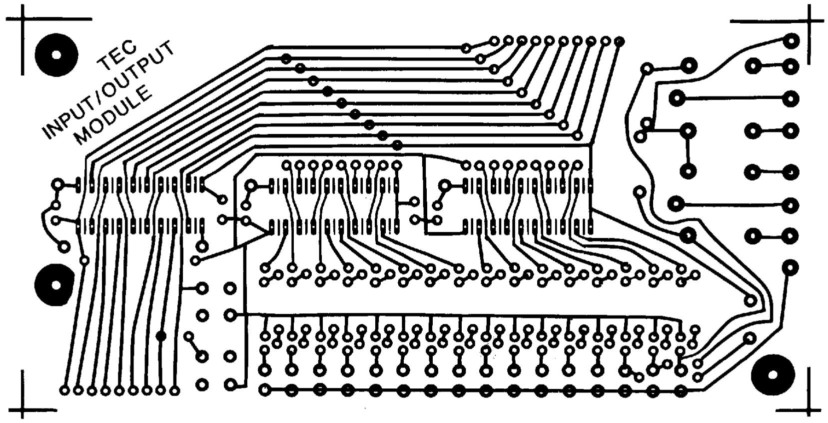
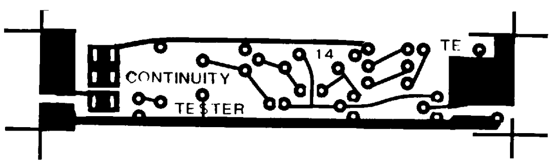
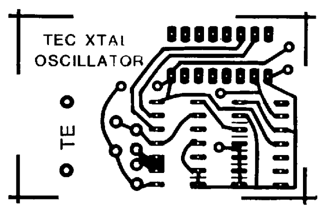
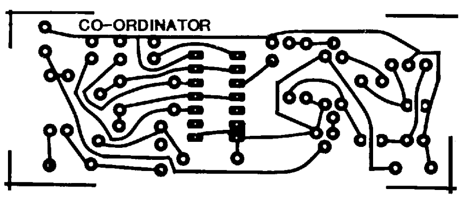
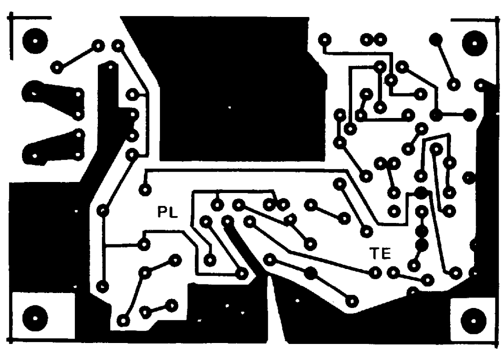
PC ARTWORK:


TEC
INPUT/OUTPUT
MODULE


CONTINUITY
TESTER
14
TE


TEC XTAL
OSCILLATOR
TE


CO-ORDINATOR


PL
TE

This will allow you to select your own supply voltage, with the necessary current capability.

When you are driving a motor, there will be three functions (or commands) needed. These are: ON/OFF (one command) FORWARD and REVERSE.

To achieve this, a number of lines (bits) will be required from the output port. Depending on the circuit used to drive the motor, either 2 or 3 bits will be required.

If you require the motor to operate in the forward direction as well as reverse, it will be necessary to use a relay. For a simple ON/OFF and FORWARD direction, a transistor can be used and only one bit (1 line) will be required. You can also get speed control from this line by including it in the program.

Basically speed control consists of outputting a high for a short duration and a low for a long duration and repeating the sequence about 100 times per second. To increase the speed, the duration of the high is increased and the low decreased. The only feature that remains constant is the repetition rate. It is essential to keep the pulses above 100Hz so that the motor rotates smoothly.

ASSEMBLY

By now you will be familiar with our assembly technique. Neatness is the overall aim. No matter how you build, the final result must be as neat as possible. This means the jumper links must be straight and sitting firm against the board, the LEDs must be close to the board and likewise the transitors, resistors and diodes. I thought it would be unnecessary to mention these points but we are still getting projects for repair in which the parts are mounted high above the board, the jumper links are twisted and kinked and the soldering is rough.

On the topic of soldering, It is important to use enough solder to cover the land and the hole. Again, we are seeing the smallest amount of solder on some joints, just enough to tack the lead to the land!

This is a very dangerous situation as you can create a problem that will be very difficult to locate. Sometimes the holes in the PC board cut through the track and the circuit relies on the solder to bridge the gap.

If you don't solder all around the lead, the copper track may contain a gap and obviously the project will fail to operate. Inspect the board before starting and check your workmanship after construction and you should have no problems in this area.

Begin assembly with the jumpers. Make sure they are straight and touching the board.

Next fit the resistors, followed by the LEDs transistors and two spike-suppressing diodes. The overlay shows how these components are placed.

The 5 spike-suppressing capacitors are next and must be fitted close to the board. The IC's are mounted in sockets and the dot on the overlay indicates pin 1. You will find one end of the IC socket has a 'cut-away' portion to match with pin 1.

Fit the relays, mini speaker and switches. Then inspect the board to make sure all leads have been soldered properly.

After adding all the parts to the board, the 5 jumper lines are added and a female matrix connector soldered to each lead. These are covered with heatshrink to prevent shorting between leads when connecting to the TEC board.

MATRIX PINS

You will notice the module in the photographs has a set of matrix pins on the output ports and also the relays. These pins are not included in the kit however you can buy some and fit them as shown in the photo if you wish.

The 5 pins included in the kit are for adding to the TEC PC board to take the 5 flying leads from the input/output board.

Paul has included a 9 pin input plug and a 10 pin plug for connecting to the TEC. These are not included in the kit but can be easily made from 18 pin and 20 pin IC sockets. They are small and delicate but will last a number of insertions and removals.

TESTING

The first program in the list is the test program. It has a short routine to flash the output LEDs so that every second LED is lit and then the others are flashed. The program repeats this a number of times then changes to detect an input from the input port. The result is indicated on the corresponding output LED.

If this sequence is not observed, the program should be double-checked. Make sure it contains the correct commands. Then check the flying leads. They must be connected to the correct outputs on the decoder chip. Refer to the line diagram for the position of each lead.

TEST PROGRAM

```
LD B,10          0900    06 10
LD A,AA          0902    3E AA
OUT (04),A       0904    D3 04
OUT (05),A       0906    D3 05
CALL DELAY       0908    CD 50 09
LD A,55          090B    3E 55
OUT (04),A       090D    D3 04
OUT (05),A       090F    D3 05
CALL DELAY       0911    CD 50 09
DJNZ             0914    10 EC
LD A,00          0916    3E 00
OUT (04),A       0918    D3 04
OUT (05),A       091A    D3 05
IN A,(03)        091C    DB 03
CPL              091E    2F
OUT (04),A       091F    D3 04
JR               0921    18 F9
LD DE,0000       0950    11 00 00
DEC DE           0953    1B
LD A,D           0954    7A
OR E             0955    B3
JRNZ             0956    20 FB
RET              0958    C9
```

The second program is a 12-note organ using a solf-touch key pad for the input and the mini speaker on the IN/OUT module as the output.

The ide of an organ may have limited possibilities in itself, but the knowledge of how to produce a tone will be very beneficial.

In robotics, for instance, a mouse can be equipped with a speaker to produce a tone when it touches an obstacle etc. The note sounds for as long as the robot touches the object.

The importance of the program is to show how a tone is produced and how the pitch can be altered by adjusting the delay value.

Follow through the program and see how this is done:

ORGAN PROGRAM

```
XOR A            0900    AF
OUT (01),A       0901    D3 01
OUT (02),A       0903    D3 02
OUT (04),A       0905    D3 04
OUT (05),A       0907    D3 05
LD HL,09FF       0909    21 FF 09
IN A,(03)        090C    DB 03
CP FF            090E    FE FF
JR Z,090C        0910    28 FA
LD BC,03FF       0912    01 FF 03
DEC BC           0915    0B
LD A,B           0916    78
OR C             0917    B1
JR NZ,0915       0918    20 FB
IN A,(03)        091A    DB 03
INC HL           091C    23
INC HL           091D    23
CP (HL)          091E    BE
JR NZ,091C       091F    20 FB
INC HL           0921    23
LD B,(HL)        0922    46
DJNZ 0923        0923    10 FE
LD A,04          0925    3E 04
OUT (05),A       0927    D3 05
LD B,(HL)        0929    46
DJNZ 092A        092A    10 FE
XOR A            092C    AF
OUT (05),A       092D    D3 05
IN A,(03)        092F    DB 03
CP FF            0931    FE FF
JR NZ,0922       0933    20 ED
JR 0909          0935    18 D2

at 0A00:
00    64    3C
FA    BD    CF
84    5C    34
DE    F3    AF
7C    54    2C
BE    D7    FF
74    4C
F9    B7
6C    44
DD    EB
```

The third program controls the 16 output lines via a 12-key phone pad.

To turn on one of the left-hand outputs (port 05), press the asterisk key then a number button from 1-8. The right-hand port (port 04), is accessed by pressing the 'hatch' key then a number from 1-8.

When a second number key is pressed, the corresponding output-line changes state. Thus a high output will go low and vice versa. To access the other latch, one of the control keys (asterisk or hatch) must be pressed.

The program is fully described beside each instruction and this will assist you to design your own programs.

An important point to remember is DEBOUNCE. The soft-touch keys require a time to settle down before a value can be read. This means a short delay must be included in the program (see address **0913** and **0914**).

The reason is the contacts in the pad are made from a carbon compound and they create a considerable amount of bounce when a key is pressed.

Since the computer is a high-speed piece of equipment, it will pick up an incorrect value if the three contacts in the switch are not closed when it is being read.

To overcome this a short delay is introduced between the time when a key is pressed and when it is read.

The program can be modified to suit your own requirements. For example: a randem output can be turned ON, or more than one output can be turned ON at the same time. A delay could be introduced to turn OFF and output after a set period of time or you could create a visual effect on a set of LEDs.

It's up to you. Study the program and try making some modifications.

For a very simple test program, try this:

```
3E FF
D3 04
C7
```

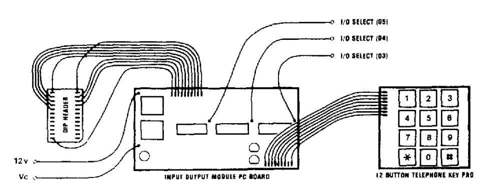
Eight LEDs will illuminate to show the program and board is working.

KEY PAD CONTROLS OUTPUT LINES

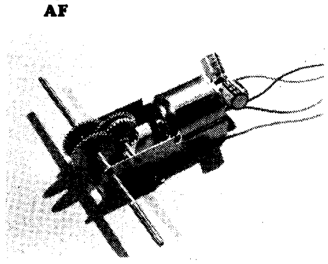
XOR A	**0900**	**AF**	Data for port 05 is stored at 0A05 and 0A04 for port 04. These two locations are initially cleared in the first 6 lines of the program. Later, you will see why we have chosen registers B and C for this operation.
LD B,0B	**0901**	**06 0B**	
LD C,04	**0903**	**0E 04**	
LD (BC),A	**0905**	**02**	
INC C	**0906**	**0C**	
LD (BC),A	**0907**	**02**	
LD HL,09FF	**0908**	**21 00 0A**	HL is the pointer for the byte table.
LD D,00	**090B**	**16 00**	D is the count register for the key.
IN A,(03)	**090D**	**DB 03**	The program inputs via port 03, looking for a key press. Any value other than FF will exit from the loop.
CP FF	**090F**	**FE FF**	
JR Z,090D	**0911**	**28 FA**	A short delay is created via the D register to give the pressure sensitive keypad switches a short period of time to settle to a value that can be read correctly.
DEC D	**0913**	**15**	
JR NZ,0913	**0914**	**20 FD**	
IN A,(03)	**0916**	**DB 03**	Input this key value via port 03 to the accumulator.
INC D	**0918**	**14**	The next 4 lines generate a value for D that will be the same as the key. This is done via a loop and incrementing D until the key value compares with the byte in the table will make D equal the key value.
INC HL	**0919**	**23**	
CP (HL)	**091A**	**BE**	
JR NZ,0918	**091B**	**20 FB**	
CP EB	**091D**	**FE EB**	The next 8 lines look for the STAR key or HATCH key and if either is pressed, C is loaded with either 05 or 04. This will allow the program to output to the correct port via the instruction **OUT (C),A** Also locations 0A05 and 0A04 use the C register for storage. In this way the C register serves a dual role and some of the powerful instructions such as OUT (C),A can be employed.
JR NZ,0925	**091F**	**20 04**	
LD C,05	**0921**	**0E 05**	
JR 0945	**0923**	**18 20**	
CP AF	**0925**	**FE AF**	
JR NZ,092D	**0927**	**20 04**	
LD C,04	**0929**	**0E 04**	
JR 0945	**092B**	**18 18**	
LD A,(BC)	**092D**	**0A**	Load A with the byte at location 0A05 or 0A04.
LD E,D	**092E**	**5A**	Store the key value for later use.
RRCA	**092F**	**0F**	The next 3 lines rotate the accumulator so that the wanted bit is rotated to the end of the register and thus only one TEST will be required.
DEC D	**0930**	**15**	
JR NZ,092F	**0931**	**20 FC**	
BIT 7,A	**0933**	**CB 7F**	Look at the highest bit and jump if it is zero. Otherwise execute next instruction.
JR Z,093B	**0935**	**28 04**	
RES 7,A	**0937**	**CB BF**	At this line the bit will be '1' and thus the program resets it to '0' and a jump is performed.
JR 093D	**0939**	**18 02**	
SET 7,A	**093B**	**CB FF**	The highest bit is SET via this instruction.
LD D,E	**093D**	**53**	Load D with the key value in readyness for rotating the accumulator back to it previous position. RLCA is a single byte instruction that rotates the accum and sets the carry flag. The bits don't enter the CARRY.
RLCA	**093E**	**07**	
DEC D	**093F**	**15**	
JR NZ,093E	**0940**	**20 FC**	
LD (BC),A	**0942**	**02**	Store the resulting byte in memory.
OUT (C),A	**0943**	**ED 79**	Output the byte to either port 05 or 04.
IN A,(03)	**0945**	**DB 03**	Look at the input port and loop the next 3 instructions until the key has been released. This is a debounce routine, essential to produce a clean key action.
CP FF	**0947**	**FE FF**	
JR NZ,0945	**0949**	**20 FA**	
JP 0908	**094B**	**C3 08 09**	Jump to the start of the main part of the program.

At 0A00:

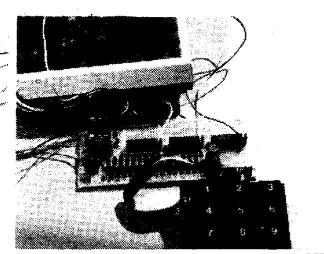
FA
DE
BE
F9
DD
BD
F3
D7
B7
CF
EB
AF



Wiring diagram showing the connection of the phone pad to the input/output module, and the module to the DIP header plug. Note: line '80' is not used when connecting the phone pad.



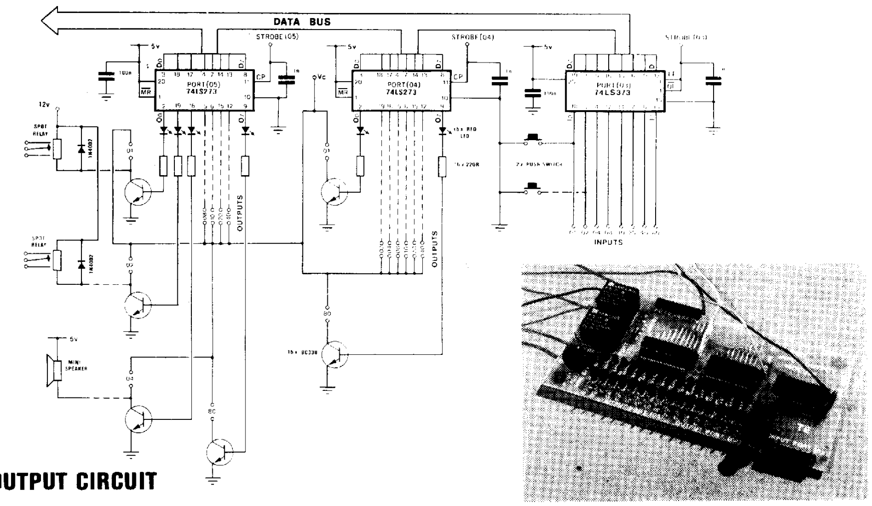
Photo, left: Motor and gearbox with two 100/16v electrolytics placed back-to-back to create a non-polar capacitor to reduce spikes from the motor. (i.e: the positive lead of each electro connects to a motor lead and the join of the negative leads is left 'floating').



Photo, right: The key pad connected to the input/output module via ribbon cable and to the TEC via hook-up flex.

INPUT/OUTPUT MODULE

Kit of parts: $33.80
PC Board: $5.00
Complete: $38.80



INPUT/OUTPUT CIRCUIT

This project allows the TEC to talk to the outside world and also accept information from the outside. It is the first interface we have described that brings the possibility of robotics to the TEC.

The INPUT/OUTPUT MODULE has one input port and two output ports. This means it will input 8 bits (8 lines) and output 16 bits (16 lines).

To allow the module to be functional as soon as it is constructed we have included two input switches and three output devices so that a simple program can be written and seen in operation. The output devices are two relays and a mini speaker. These will allow you to test the board and see how it operates, before adding any other devices.

We have included some test programs in the article and they will show the indicator LEDs in operation.

These LEDs indicate when a particular output is high and will be invaluable when trouble-shooting a fault in either a program or in hardware.

The 5 flying leads on the module are clearly marked and you will see the input port is controlled via strobe line 03 and output ports via strobe lines 04 and 05.

Each of the 8 input and 16 output lines is further identified by a hex value on the PC overlay and this will assist you when writing a program.

The most interesting use for the board will undoubtedly be for robotics and when designing in this field, a whole new world of mechanical and electromechanical terms will be encountered.

Before embarking on a design, it is important to have some idea of what you are going to create. It may be an arm, a wheeled vehicle or a mechanical controller such as a door opener, a lift, crane or remote controlled boat or plane.

No matter what the project, begin by collecting articles and notes describing similar or related devices and study how other designers have puts things together. Combine the features you like and make sketches and diagrams of how you indend yours to look.

The most important point is not to be too ambitious on your first attempt. Aim for a simple design, using maybe a single motor and gearbox with say one or two flashing lights and a speaker.

You will have sufficient interfacing problems with these to keep your inventive skills at work for a while.

The other point to remember is to select materials that you can readily obtain and don't choose thick material as this will be very difficult to work with.

PARTS

16 - 220R ¼watt

3 - 1n greencap
2 - 100n

2 - 1N 4002 diodes
16 - 3mm red LEDs
16 - BC 338 transistors

2 - 74LS273 IC
1 - 74LS373 IC

3 - 20 pin IC sockets
2 - PC mount push buttons
1 - Mini Speaker 80R
2 - SPDT relays

50cm tinned copper wire
5 - PC matrix pins
5 - Matrix connectors
10cm - Heatshrink tubing
15 - 20cm lengths of hook-up flex

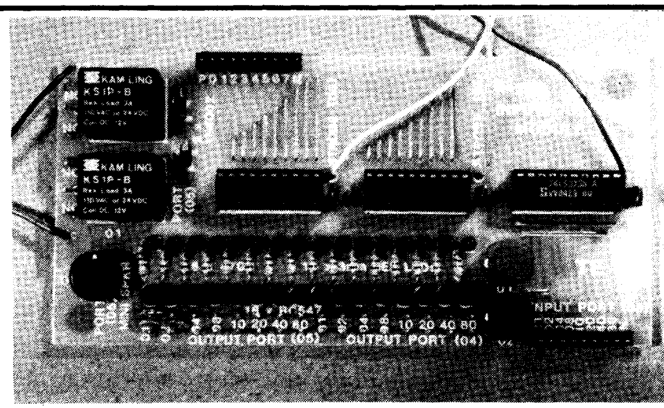
20cm - 10 core ribbon cable
1 - 12 key telephone pad

1 - INPUT/OUTPUT MODULE PC

3mm clear plastic sheet is the best choice as it can be cut, bent, folded and even heated into shape. It also looks appealing and being clear, you can see through it and this makes the project look more complex!

Equally suitable is PC board as it has a copper surface that can be soldered to and thus small brackets can be added for shafts etc.

The only material I would avoid is sheet metal. Even though it has good strength, the same can be provided via plastic with the use of a few strengthening pieces, without the difficulty of cutting folding and drilling. For tinplate to have any strength it must be reasonably thick and you will require heavy duty tools etc to shape it.



Another handy medium is wood, however this should be restricted to base panels and platforms, where a number of items need to be screwed into position. You should only use soft wood, as it will be lighter and easier to drill and screw into. Don't use nails for fixing or joining as they tend to work lose.

Lastly, don't be frightened to use parts you already have on hand, especially from the kitchen and laundry where you will find plastic bottles, lids and boxes ideally suited for turning into pulleys and wheels. Use all your imagination and initiative - you will need it as you are basically breaking new ground!

In robotics, lots of new terms need to be understood to make the project function properly. But the best way is the hard way. By trial and error. Terms like gear ratios, torque, drive speeds, strength of beams, can involve an enormous amount of mathematics. That's why it's best to look through articles and see how it has been done by others.

At the time of writing, only a very limited range of motors and gearboxes are available at the low end of the market and the best of these we found at Dick Smith Electronics.

The gearboxes are in kit form and require a small amount of assembly to fit the gears onto the shafts to produce a gearbox known as a compound gearbox.

A gearbox reduces the rotational speed of a motor and at the same time increases the torque.

Torque is the twisting or turning force of a shaft and after 3 or 4 gear reductions, a shaft will have a considerable turning force.

This will be sufficient to turn wheels or move a robot arm or lift a weight. Sometimes it is necessary to convert rotation into straight-line motion and this can be done with a rack and pinion, winch and string, crank and arm or wheel and track.

Apart from the problems you will encounter adapting the mechanics into the available space, there will be problems interfacing the motor to the electronics.

One of the major problems will be noise. Motors are inherently high noise producers and they must be kept far away from the electronics, both physically and electrically.

This may require a separate power supply so that noise and glitches from the motor do not get into the computer bus lines.

It will also be necessary to have high current available for the motor(s) as they draw a high current under load and if they stall, you must have sufficient current available to allow them to restart as soon as the load reduces.

A stalled motor can create a virtual short circuit and if connected to the computer 5v supply, the computer may drop out.

This has been avoided on the INPUT/OUTPUT MODULE by providing a separate supply line for the collectors of the output transistors and also the relays.