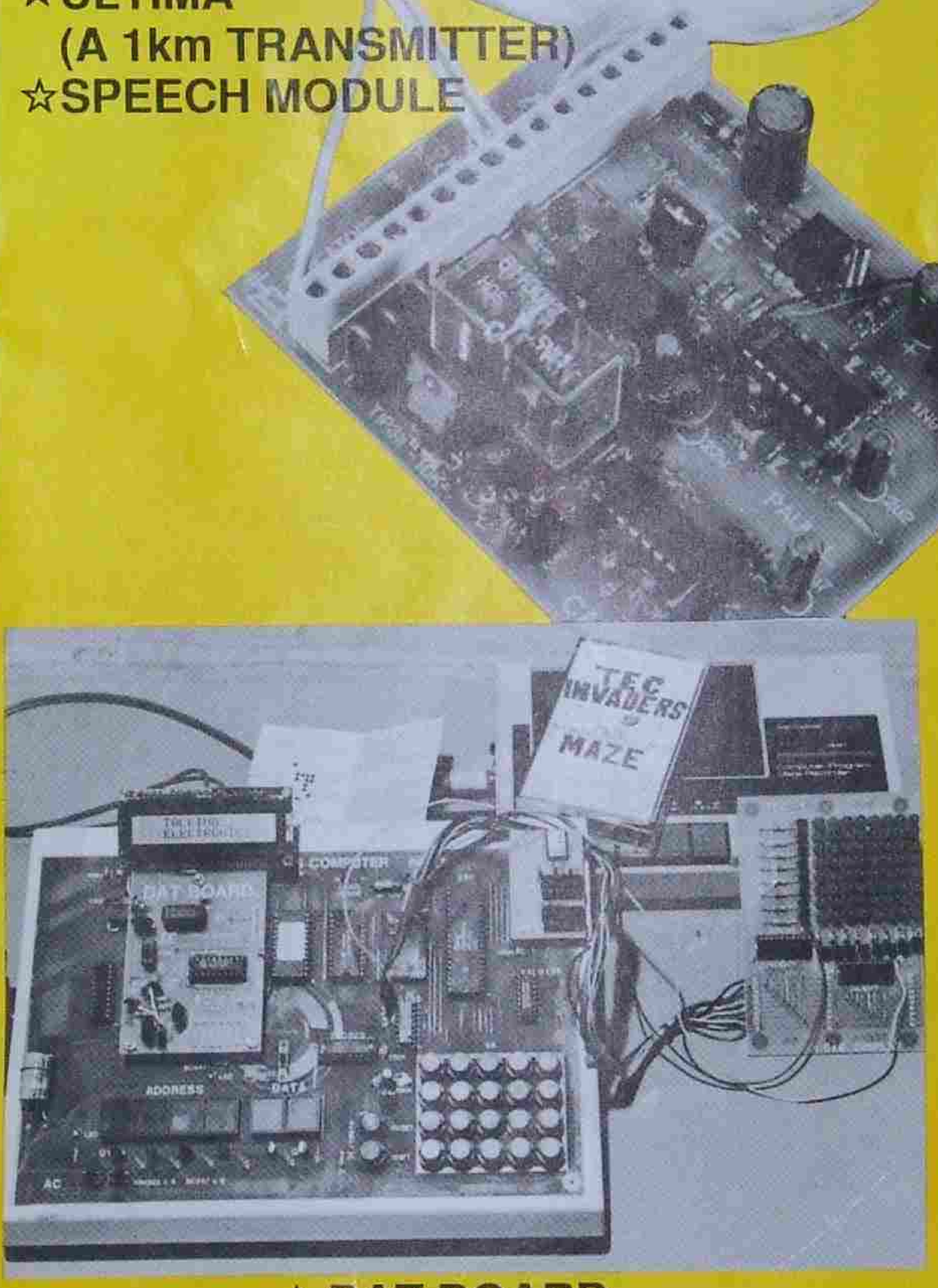
May '89
Reg by Aust. Post.
Publication No: VBP 4256

TALKING ELECTRONICS®

$3.00
$4.60NZ

Issue No 15

☆CAR ALARM
☆ULTIMA
(A 1km TRANSMITTER)
☆SPEECH MODULE



☆ DAT BOARD

☆ FM RADIO

☆ ORGAN

15058920k

TALKING ELECTRONICS

Vol.1 No: 15.

Editorial

It's great to get another issue of TE out. If you think the delay between issues is intentional you're wrong. It's much more complex than that.

The instant we get an issue out the orders start to flow and it takes about three months before they taper off enough for us to get back to the drawing board for the next issue.

Since the last issue we have doubled in size and now have 6 staff and 2 part-time helpers. Even so, we can't get the issues out any quicker as each project takes a long time to prepare, when you take as much care as we do.

Most of the delay is not our doing at all.

It comes from the run-arounds from suppliers and the like. Take the trouble we have had with designing a new microprocessor project. It is practically impossible to find a suitable CPU thats cheap and surpasses the Z-80. We have come to so many dead ends that we have given up on the idea for the moment.

Even with the range of common parts as used in TE kits, we experience shortages of components for nearly every kit and it's a constant hassle waiting for things to come in.

Never-the-less we have been hard at it, producing ideas for beginners, add-ons for the TEC and projects for the more advanced experimenter.

The DAT board is our latest add-on for the TEC and includes a software package that advances the TEC's capability quite considerably.

Also included is a speech project using the SP 0256 AL2 allophone chip. Although I find it difficult to understand some of the words it produces, this chip is the cheapest and best on the market at the moment. With this project we can finally say we have a talking project to substantiate our name "Talking Electronics."

Our cover project is a car alarm that offers all the features you have ever wanted, and at a price that beats anything else on the market.

For the beginner we have a couple of starter projects that will introduce the magic of electronics.

All in all we hope to cater for everyone and I hope you have noticed our new format and different type-style.

This is the first issue from our desk-top publishing set-up. After spending nearly $20,000 and experiencing 2 hard-disk crashes with the loss of weeks of work, we can say we are on the way to producing page-finished copy for the magazine. We will tell you more about our system next time as we have a whole story to relate! It's a bit like the photocopy saga all over again.

See the centre pages of this issue for a current list of kits and books we have released. The notebook series is especially important as notebook number 5 has a BEC (Basic Electronics Certificate) set of questions to show you the content of a course that gets you started in electronics via Australia-wide TAFE colleges. If you have ever wanted to know where to start, this is it. The BEC is the first step to take.

Our shop at Moorabbin has just opened to coincide with the release of this issue. If you are passing by that way, call in and see Ross, he will be only too glad to show you the enormous range of Public Domain software and all the other things he has on the shelves.

Don't forget our kits (we have over 100 different models) and I hope to see you sending for something in the near future.

For now,

Colin Mitchell.

PUBLISHERS NOTE:

Talking Electronics is designed by Colin Mitchell at 35 Rosewarne Avenue, Cheltenham, Victoria 3192, Australia. Articles suitable for publication should be sent to this address. Ring us first on (03) 584 2386.

*Maximum recommended retail price only

CONTENTS

5 CAR ALARM
-A 3 chip car alarm that is as effective as a commercial unit

13 ORGAN
-A 32 key stylophone organ with three modes of play

17 TEC TALK
-A forum on the TEC computer

22 THE TAPE SYSTEM
-A description on the uses of a tape system

26 SPEECH MODULE
-Gives the TEC computer speech capabilities

32 EPROM PROGRAMMER REVISITED
-Modifications to improve the operation of the eprom burner described in issue 13

37 Order Forms & Price List

47 THE DAT BOARD
-Adds LCD and tape save to the TEC computer

52 THE LIQUID CRYSTAL DISPLAY
-A description on the operation of the LCD

56 MAGIC SQUARE
-A program for the TEC using the 8 x 8 Display

59 A SNEAK PREVIEW
-A look at our next Bugging Book

61 THE ULTIMA
-Our 1km FM transmitter

67 Shop Talk

68 FM RADIO
-A simple two chip FM receiver using the TDA 7000

74 Drigs

ADVERTISERS INDEX

PC BLUE LISTING	4
EXPERIMENTER PARTS Co.	12
TALKING ELECTRONICS Pty. Ltd.,	37, 38, 39, 40
TALKING ELECT. Public Domain	35, 36, 41, 42
TALKING ELECTRONICS SHOP	33, 34, 43, 44
AUST. DIGITAL ELECT. SCHOOL	60

Technical: Jim Robertson
CAD Artwork: Paul Loiacono
Enquiries: 10 minute queries
584 2386 8am - 6pm.

Our Talking Electronics Shop, 2 Central Ave., Moorabbin, carries the full range of TE kits and parts and over 2000 Public Domain titles for IBM and compatible computers

TE's

CAR ALARM

PROTECT YOUR CAR FROM THEFT WITH THIS FULL-FEATURED ALARM

Police records show that, in 1988, 780 cars were stolen per 100,000 population. That's 117,000 cars per year for a population of 15 million. Or approximately 1 car every 4.5 minutes!

Most cars are usually stripped and sold as spare parts to wreckers or taken by joy riders, for an evening of what they call fun, or broken into just for the contents.

Insurance companies have increased their premiums accordingly and some offer a small discount if your car is fitted with a car alarm. That's what inspired us to design an alarm for the magazine.

A lot of thought has been put into the design and all the common features have been included, including a few of my own.

Here they are:

*Runs off a 12v car battery.

*Has battery backup. (Kept charged when the engine is running)

*Will flash a signal lamp as soon as the ignition is switched off, regardless of whether the alarm is swithed on or not.

*As soon as the ignition is switched off, a beep will be produced for approximately 5 seconds (to remind you to turn the alarm on).

*A 10 second exit delay is provided to allow you to leave the vehicle and lock all doors. (The delay can be removed, as explained in the article)

*The indicators will flash twice after the 10 second exit delay to indicate the alarm is activated and ready.

*Two delayed inputs on-board (active low, ie must be taken to ground or low) to trigger the alarm.

*The alarm is activated approximately 5 seconds after an input has been taken low to give plenty of time for it to be turned OFF by the operator.

*The siren will sound for approximately 2 minutes and shut down ready for another break-in attempt.

*Indicators flash in conjunction with the siren to give a visual indication that the vehicle has been broken into.

*An ignition killer cuts off the ignition as soon as the alarm is turned on.

HOW THE CIRCUIT WORKS

The best way to describe how the circuit works is from the power supply section and battery charger.

The battery charger and power supply is fairly straight foward. The car battery voltage is approximately 12v. This passes through a 1N5404 power diode to the input of a 5 volt regulator. The ground pin of the regulator has two resistors to increase the output voltage to about 9 volts and supplies the rest of the circuit. The key switch connects power to sections of the circuit to activate the alarm.

When the engine is started, the voltage across the car battery increses to about 14 volts. This is due to the battery charging. When the voltage rises to 14 volts, there will be sufficient voltage drop across the 1N5404 diode to keep the backup battery charged.

As soon as the ignition is switched off, the voltage on the 47k on the ignition line is removed, thus enabling pin 13 of IC1 and allowing the gate to operate as a low frequency oscillator with an even mark-space ratio. Output pin 12 drives the base of a BC338 transistor via a 1k resistor which in turn switches a dash lamp on and off. The lamp will flash at about one flash per second regardless of whether the alarm is switched on or not.

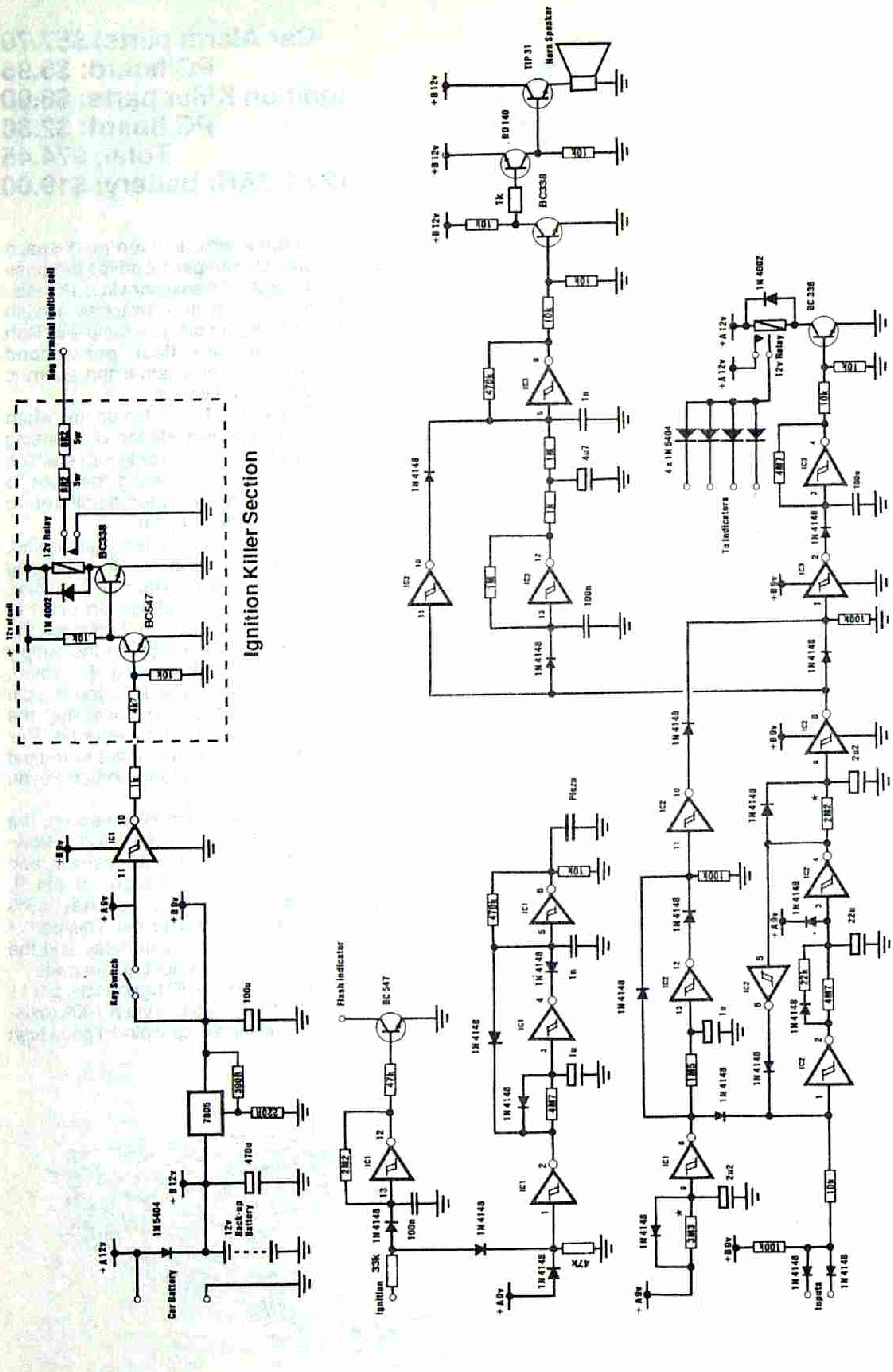
Pin 1 of IC1 will also go low when the ignition is switched off, causing pin 2 to go high. This in turn enables pin 5 of IC1, allowing the gate to operate as another oscillator to directly drive a piezo.

While the tone is being generated, a 1uF electro is slowly charging from pin 2 of IC1 via a 4M7 resistor, increasing the voltage on pin 3 of IC1. After about 5 to 8 seconds the voltage reaches 66% of the supply voltage (approximately 6 volts). Output pin 4 will go low, causing pin 5 of IC1 to go low, disabling the oscillator and the tone will stop. This part of the circuit is used to remind you to tum the alarm on before you leave the vehicle.

When the alarm is turned on, the 2.2uF electro on pin 9 of IC1 is slowly charged via a 3M3 resistor, and increases the voltage on pin 9. When the voltage reaches 66% supply, pin 8 will go low. This part of the circuit is the exit delay and the process takes about 20 seconds.

When pin 8 of IC1 goes low, pin 11 of IC2 is forced low via a 100k resistor. Thus the output pin 10 goes high

CAR ALARM CIRCUIT



enabling the indicator flash circuit, causing all four of the car's indicators to flash. The 1uF on pin 13 of IC2 is slowly discharging, thus decreasing the voltage on pin 13. When the voltage on pin 13 drops to 33%, the output pin 12 will go high, taking pin 11 high, causing the output pin 10 to go low. This disables the indicator flash circuit. This process will flash the indicators twice, indicating the alarm is activated and ready.

Pin 1 of IC2 is also enabled when pin 8 of IC1 goes low and is ready to detect when either or both alarm inputs are pulled low. When this happens, pin 1 of IC2 will go low and the output pin 2 will go high, quickly charging the 22uF electro on pin 3 of IC2. When the voltage on pin 3 rises to 66%, the output pin 4 will go low, taking pin 5 low with it. This causes pin 6 to go high and holds pin 1 high to prevent the circuit from being retriggered. Pin 2 will go low, allowing the 22uF electro to slowly discharge through pin 2 via the 4M7 resistor.

While pin 4 of IC2 is low, the 2.2uF on pin 9 of IC2 is slowly discharging through pin 4, via the 2M2 resistor and when the voltage drops to 33%, the output pin 8 will go high. This part of the circuit is the entry delay and the process takes about 10 seconds. When the 10 seconds are up, both the siren circuit and the indicator flash circuit are enabled.

When pins 8 or 10 of IC2 are high, pin 1 of IC3 will go high and the output pin 2 will go low, enabling pin 3 of IC3, and the gate will operate as a low frequency oscillator with even mark-space ratio. The output pin 4 drives the base of a BC338 transistor, via a 10k resistor, to switch a relay and turn the car indicators on and off (via four 1N5404 power diodes).

The siren circuit is also activated when pin 8 of IC2 goes high. Pin 11 of IC3 goes high causing the output pin 10 to go low. This enables pin 5 of IC3 and the gate works as an oscillator to generate a tone. Pin 13 is also enabled and the gate works as a low frequency oscillator of about 2Hz (2 pulses per second). The purpose of the 1k resistor, the 4.7uF electro and the 1M resistor, is to alter the output frequency of pin 6, as the output pin 12 rises and falls to give a WAH WAH tone rather than a pulsed-tone effect.

Pin 6 is then fed into the input of a simple three transistor amplifier via a 10k resistor. The collector of the first transistor, a BC547, is taken to the base of the second, a BD140. The collector of the BD140 drives the base of the third transistor, a TIP31. This is an emitter follower and the emitter drives an 8 ohm horn speaker to give a very loud output.

The ignition killer is a seperate board controlled by the main board. When the alarm is switched on, pin 11 of IC1 goes high, causing the output pin 10 to go low. This is then passed to the ignition killer board.

The ignition killer consists of three main parts; two transistors and a relay. The positive of the board goes to the positive terminal of the ignition coil. When pin 10 of IC1 is low, the first transistor in the ignition killer (a BC547) is turned off, allowing the current flowing through its 10k collector resistor to flow through the base of the second transistor (a BC338) switching the trasistor on. If the ignition is started, the 12v 10A relay will close, connecting two series 8R2 resistors from the negative terminal of the ignition coil to ground, making it impossible to start the car.

CONSTRUCTION

Construction is straight foward. The board may look difficult to build but is really quite simple.

The first thing to do is to inspect the P.C. board for any holes not drilled or shorts due to poor etching. Some holes may be covered by solder, but this can be removed by applying a hot soldering iron to the land to melt the solder.

PARTS LIST

CAR ALARM - MAIN BOARD

1 - 220R All resistors on the
1 - 390R main board are 1/4 Watt
4 - 1k
8 - 10k
1 - 22k
1 - 33k
1 - 47k
3 - 100k
2 - 470k
2 - 1M
1 - 1M5
2 - 2M2
1 - 3M3
3 - 4M7

2 - 1n Greencap
3 - 100n Greencap
2 - 1u 25v Electrolytic
2 - 2u2 25v Electrolytic
1 - 4u7 25v Electrolytic
1 - 22u 25v Electrolytic
1 - 100u 25v Electrolytic
1 - 470u 25v Electrolytic

21 - 1N4148 Signal Diodes
1 - 1N4002 1A Power Diodes
5 - 1N5404 3A Power Diodes
3 - BC338 Transistor
1 - BD140 Transistor
1 - TIP31 Transistor
1 - 7805 Regulator
3 - 74C14 or 40106 IC's

3 - 14 Pin IC sockets
1 - FBR611D012 12v 10A SPDT Relay
1 - 16 way screw terminal strip or 2 x 8 way
1 - Small Piezo
1 - Key Switch
1 - 12v Dash Lamp
1 - 8R 10 Watt Horn Speaker
1 - Nut and Bolt for regulator
8 - 2m heavy duty hook up wire (2 Red, 2 Black, 2 Brown and 2 Blue)
3 - 2m medium duty hook up wire (3 different colours)
4 - 1m medium duty hook up wire (different colours)

1 - 1m of light duty hook up wire (any colour)
20cm Tinned Copper Wire for links and tests

CAR ALARM P.C. BOARD

IGNITION KILLER SECTION

2 - 8R2 5 Watt
2 - 10k 1/4 Watt
1 - 47k 1/4 Watt

1 - 1N4002
1 - BC547
1 - BC338

1 - FBR611D012 12v 10A SPDT Relay

IGNITION KILLER P.C. BOARD

Lay out the components of the kit on a clean spot on the bench and compare each with the list above. In some cases we may have substituted a part due to the difficulty in obtaining it or as a modification to the circuit.

This is a digital circuit and a wide tolerance can be accepted for most of the parts. If a part is substituted, a note will be included in the kit.

The easiest way to construct the kit is to start with all the components that lie flat on the board. Starting with the 3 links, you will need three lengths of tinned copper wire approximately 2cm long. It will be easier to do these one at a time. Bend the wire like a staple and fit each end through a hole. Solder one end and with a pair of long nose pliers, gently grab and pull the unsoldered end until the link is lying flat on the board. Don't pull too hard or the wire will break. Once the link is flat, solder the other end. Repeat for the other two links.

Next those resistors that lay flat on the board. These can be soldered either way round, but make sure you put the correct resistor in each place. If you are unsure of the resistor colour code, flick through some of your notes or books for the information. There is a complete colour code on the back cover of issue No. 1 of our magazine.

Next those 1N4148 signal diodes that lay flat on the board. These look like little glass beads with a black or coloured band near one end. This is the cathode end and should be placed over the bar or line on the overlay. This is important or the circuit will not function properly.

The IC sockets are next. Looking from the top of an IC socket, you will notice an indent at one end. If you turn the socket so that the indent is to your left, pin one will be directly below. The three IC sockets should be placed on the board so that the indent is over the dot on the overlay.

Solder any two opposite pins of each IC socket. While holding the board in your left hand (if you are left handed, hold the board in your right hand) with the components facing your palm. Gently push the IC sockets while reheating the corner pins to ensure they are sitting flat on the board. You can now solder the rest of the pins.

Now the upright resistors and signal diodes. Take extra care not to put the diodes around the wrong way. Remember the cathode is represented by a band at one end.

Next the power diodes can be soldered in place. There are six of these. One 1N4002 and five 1N5404's. They should also be placed on the board the right way around, taking note of the cathode.

The capacitors and electrolytics go in next. The electros should be placed on the board so that the positive lead goes down the hole marked with a PLUS "+" sign. The greencaps can be placed on the board either way around.

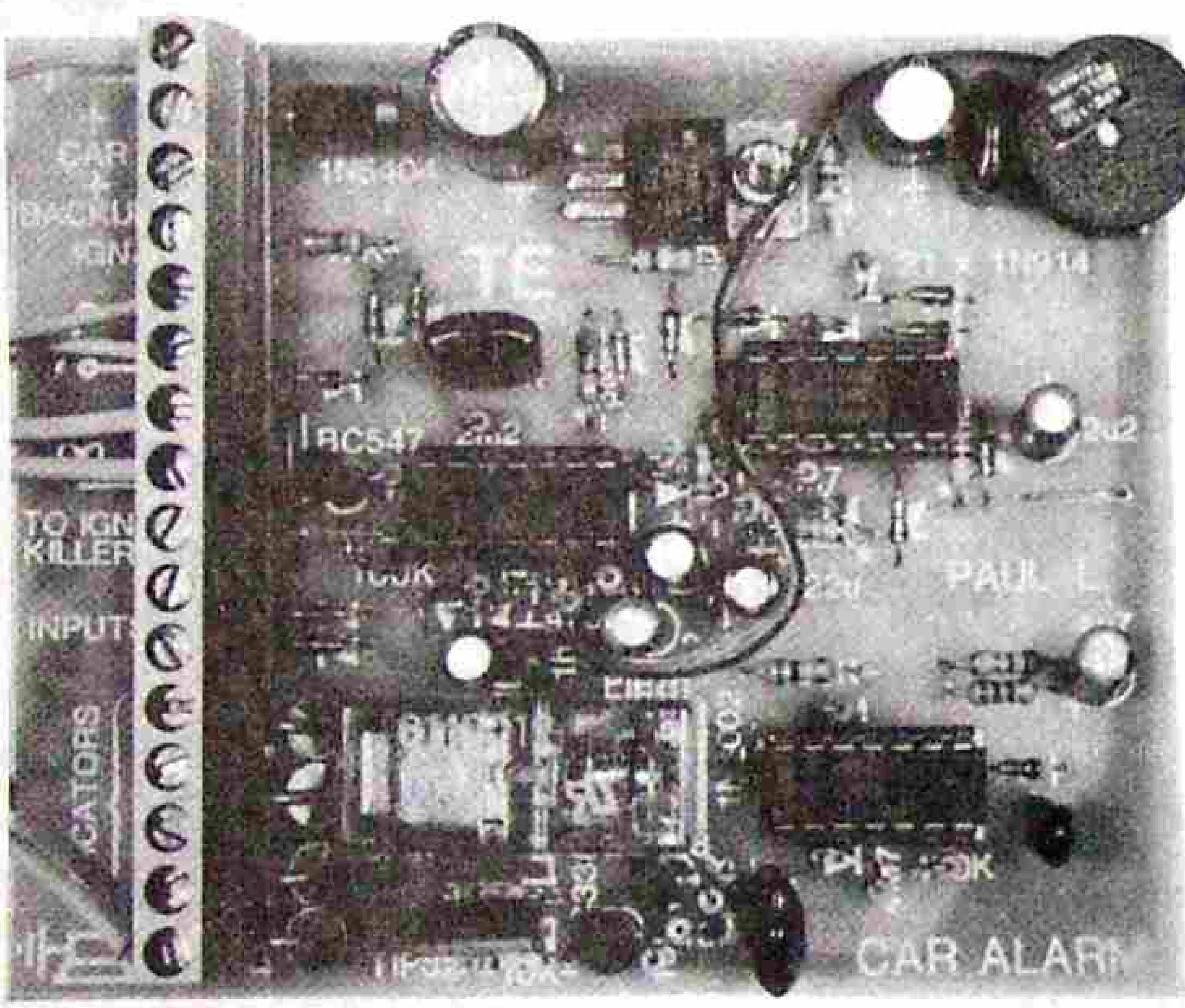
Next the transistors and voltage regulator. These are soldered in, taking note of the orientation. The small signal transistors are placed as per the overlay. The power transistors are placed so that the emitter of each goes down the hole labelled "e". The overlay for each power transistor shows one side thicker than the other. This represents the back (the metal part).

The 7805 voltage regulator is placed on the board face up and must be bolted down first then soldered. If it is soldered first, the pins may be pushed through the board when the bolt is being tightened. This may cause the lands to lift off and the whole thing may become messy.

The 16-way screw terminal, relay and piezo are the last items to be added. The terminal strip is soldered with the openings facing out. There is not much you can get wrong with the relay as it only fits one way on the board. The piezo can be soldered either way around and the case can be glued or "blu tacked" to the top right corner to keep it from moving around.

The three 74C14 ICs are inserted into their sockets with pin one towards the left of the board. Construction of the main board is now complete.

You should have no trouble in building the ignition killer board by yourself. The only odd parts are the two 8R2 5 watt resistors. These are placed either way around, with the identification facing up for easy recognition.



The overview of the CAR ALARM board clearly shows where the major components are placed. Note the orientation of the IC's, regulator and the power transistors.

This photo shows the ignition killer board, ready for installation.

TESTING THE ALARM

Before installing the alarm, we suggest you test it first to avoid the inconvenience of pulling everything out if it doesn't work.

Cut 16 lengths of tinned copper wire, each 2cm long and insert them into the screw terminals. Using jumper leads, connect "IGN." to "+ CAR".

If you have a 12v globe, connect it between one of the "indicator" outputs and "ground". A LED and 1k resistor could be used if a 12v globe is not available. Connect the horn speaker between the "horn speaker symbol" (terminal 16) and ground. Connect the dash lamp to the terminals with the "lamp symbol" (terminals 7 and 8) and the key switch to the terminals with the "switch" symbol (terminals 5 and 6). Turn the key anticlockwise to ensure the switch is in the OFF position.

Connect the ignition killer board input to the main alarm board; the positive to "+ CAR" and the negative to "ground".

Now connect a 12v battery, capable of delivering a constant current of 1 AMP comfortably, or a power supply, between "+ CAR" and ground, making sure the positive of the supply goes to "+ CAR" and negative to "ground".

While power is connected, remove the jumper lead between "IGN." and "+ CAR". A tone will be heard and the dash lamp will flash. The tone will be heard for about 5 seconds.

Turn the key clockwise to the "on" position. As soon as this is done the relay on the ignition killer board will close and after about 10 seconds, the indicators will flash twice to indicate the alarm is ready to detect a low signal on the inputs.

With a jumper lead, connect one end to ground and touch any of the inputs with the other for about 2 seconds. There is a delay to stop glitches and spikes in the power from triggering the alarm. This delay can be changed. 5 seconds after the alarm is triggered, the siren will sound and the indicators will flash. This will continue for about 2 minutes and then stop, ready for another trigger.

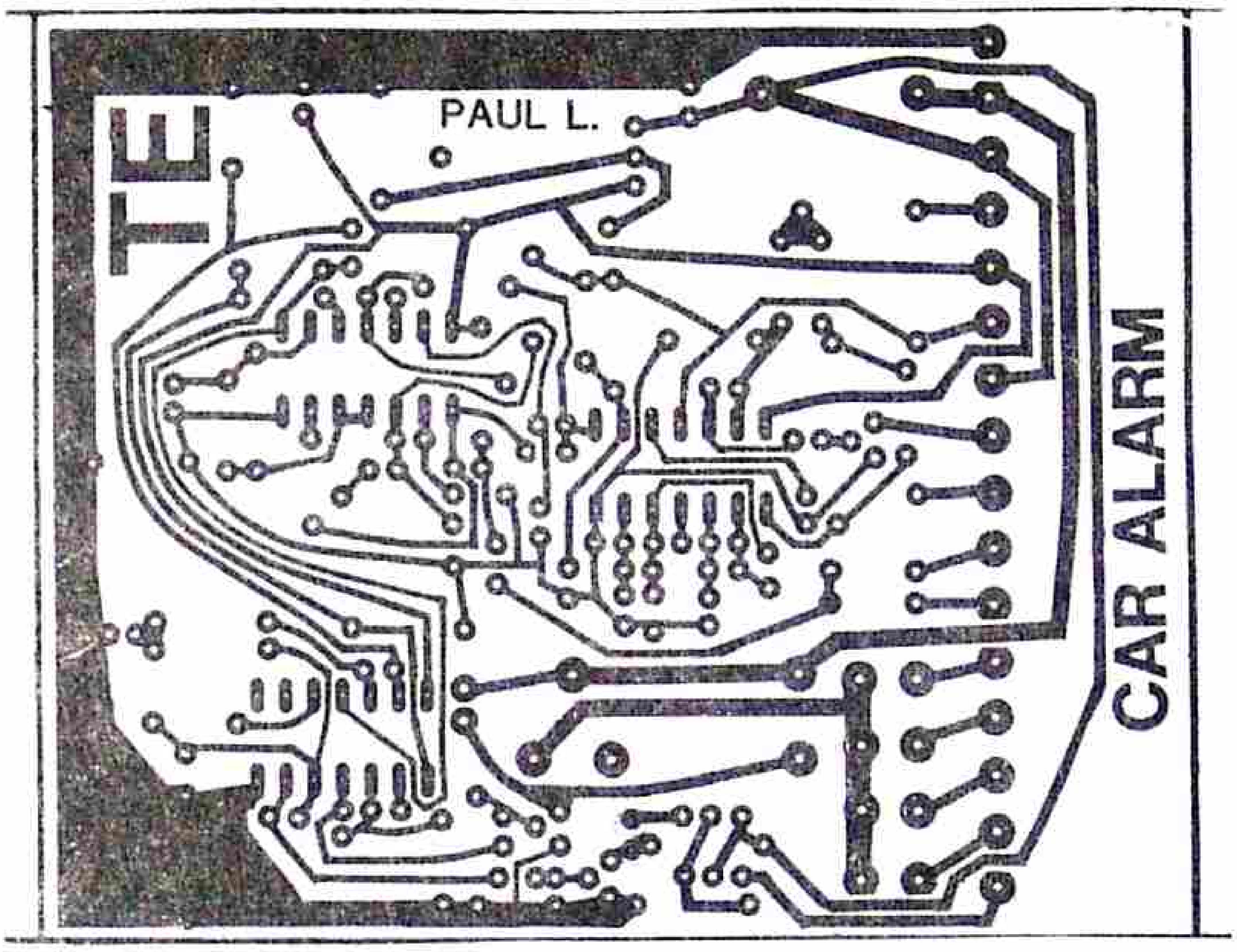
If you had no luck with this section, go to the IF IT DOESN'T WORK section.

If all went well you can change the delays to suit your needs. The 3M3 on pin 9 of IC1 is the exit delay and is increased to increase the delay or reduced to reduce the delay. The 22k on pin 3 of IC2 is the sensor delay and the 2M2 on pin 9 of IC2 is the entry delay. These two are also adjusted in the same way as for the exit delay to increase or reduce the delays. All these resistors are marked with an asterisk on the overlay.

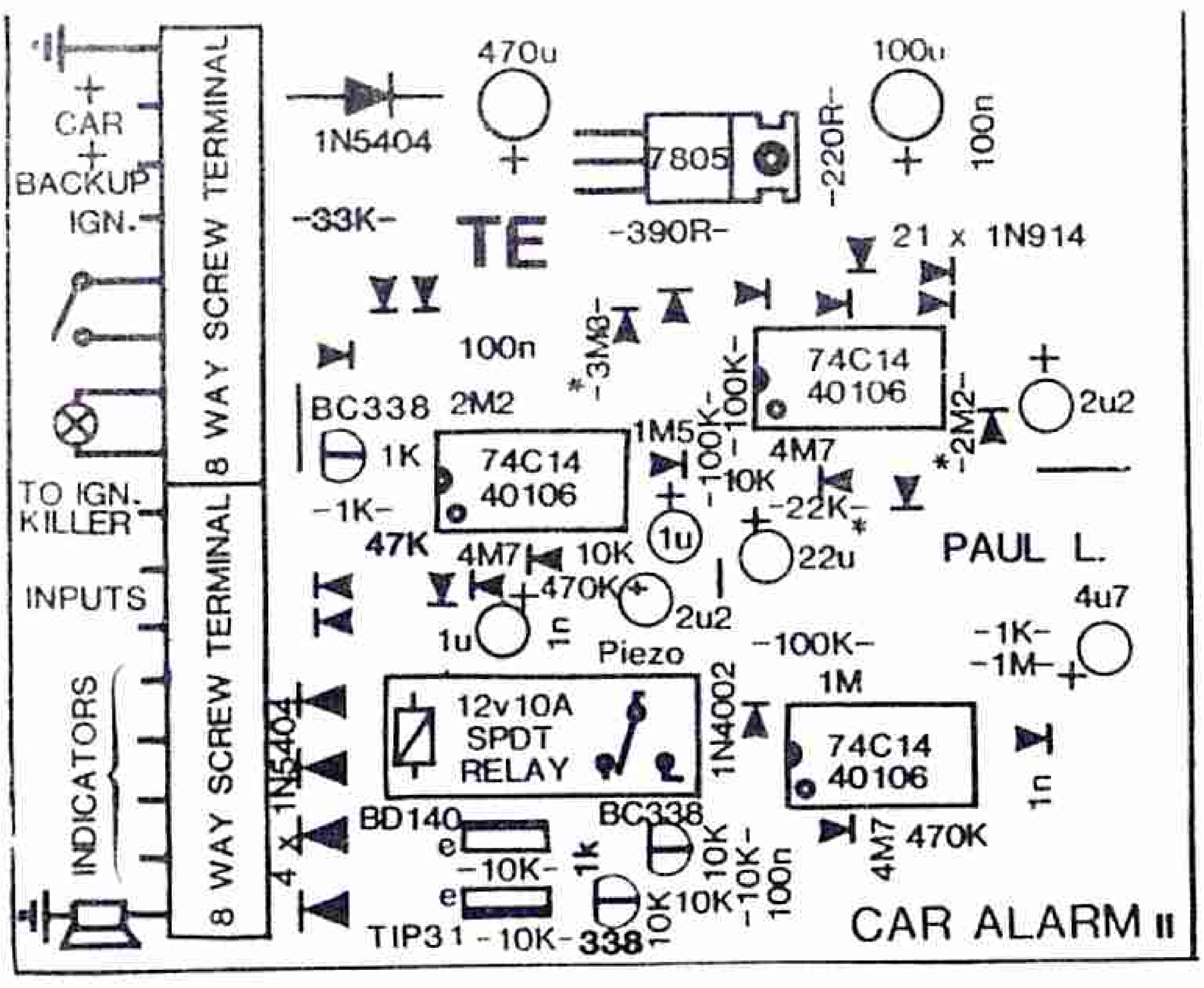
Now you can install the alarm in your vehicle. Due to the fact that all cars are different, we will leave the installation entirely to you. We will list where each terminal goes and how to install the ignition killer. A simple diagram below will show you how to wire the alarm.

Starting from the top:-

1 - To any ground connection on the vehicle.



CAR ALARM ARTWORK



2 - "+ CAR" - Directly to the positive of the vehicle (preferably to the fuse box).

3 - "+ BACKUP" - To the positive of the backup battery.

4 - "IGN." - To any point in the fuse box that goes high only when the ignition switch is on.

5 and 6 - To the alarm key switch.

7 and 8 - To the dash lamp

9 - To the ignition killer input.

10 and 11 - To the door switches or any other alarm sensor.

12, 13, 14 and 15 - to each of the four car indicators.

16 - To the horn speaker. The other end of the speaker goes to ground.

The ignition killer is mounted in a small jiffy box and is placed in the engine compartment close to the ignition coil. Connect the negative terminal to ground and the positive to the positive terminal of the ignition coil. The input of the unit goes to the main alarm board and the left-over wire, labelled "NEG. IGN. COIL", is screwed to the negative terminal of the ignition coil.

Now the alarm is ready to be tested, BY A REAL CAR THIEF.

IF IT DOESN'T WORK

First remove any power from the board and check thoroughly for any shorts, solder bridges, or cracks that may go through a track. Then check that all the parts are on the board, in their correct position and round the right way. If everything looks to be in order re-apply the power and check the voltages on the board. You should have 12v - 14v coming into the board and the output of the regulator should be about 9v.

If the dashlamp doesn't flash or does not illuminate at all when the ignition switch is off, short between collector and emitter of the BC338 transistor. The lamp should illuminate.

With a multimeter or logic probe measure pin 12 of IC1. Take pin 13 low. Pin 12 should go high, turning the transistor on and illuminating the lamp. If pin 12 goes high and the lamp still does not come on, replace the transistor. Take pin 13 high. Pin 12 should go low, turning the transistor off. If pin 12 does not go low, the gate is faulty.

If no beep is heard from the piezo, check to see that pin 2 of IC1 is high. If so, take pin 3 low and check if pin 6 is oscillating. If so, the piezo is faulty. If not, while pin 3 is still low, pin 4 should be high. If you take pin 5 high. Pin 6 should go low and vice versa.

If the beep does not go off, the 1uF on pin 3 is not charging due to a very leaky electro or the 4M7 is open. If the beep is only off when the alarm or ignition is on, the fault will be the diode between pins 2 and 5 is open.

If there is no exit delay, the diode on pin 9 of IC1 could be shorted or the 2.2uF electro dry or open and have very high resistance.

If the alarm does not activate, check pin 8 of IC1. A high on pin 9 will make pin 8 go low and vice versa. Check pin 6 of IC2. Take pin 3 of IC2 low, pin 4 should go high and pin 6 should go low. Take pin 3 high, pin 4 should go low and pin 6 should go high.

Next check pin 2 of IC2. Assuming there are no inputs connected and the alarm is switched on, short across the 22uF on pin 3 to discharge the electro. Pin 2 should be low. Take pin 1 low, pin 2 should go high and pin 4 should go low shortly after. Next check pin 8 of IC2. Taking pin 9 low will make pin 8 go high and vice versa. If there is no entry delay, the diode between pins 4 and 9 could be shorted or 2.2uF not connecting properly.

If the indicators do not flash twice, 10 seconds after the alarm has been switched on, take pin 13 of IC2 low. Pin 12 should go high and vice versa. Take pin 13 high and take pin 11 low, pin 10 should go high and taking pin 11 high will make pin 10 low. If the diode between pins 12 and 11 is open, the indicators will always flash when the alarm is switched on and if the diode between pin 8 of IC1 and pin 11 of IC2 is open the indicators will always flash when the alarm is switched off.

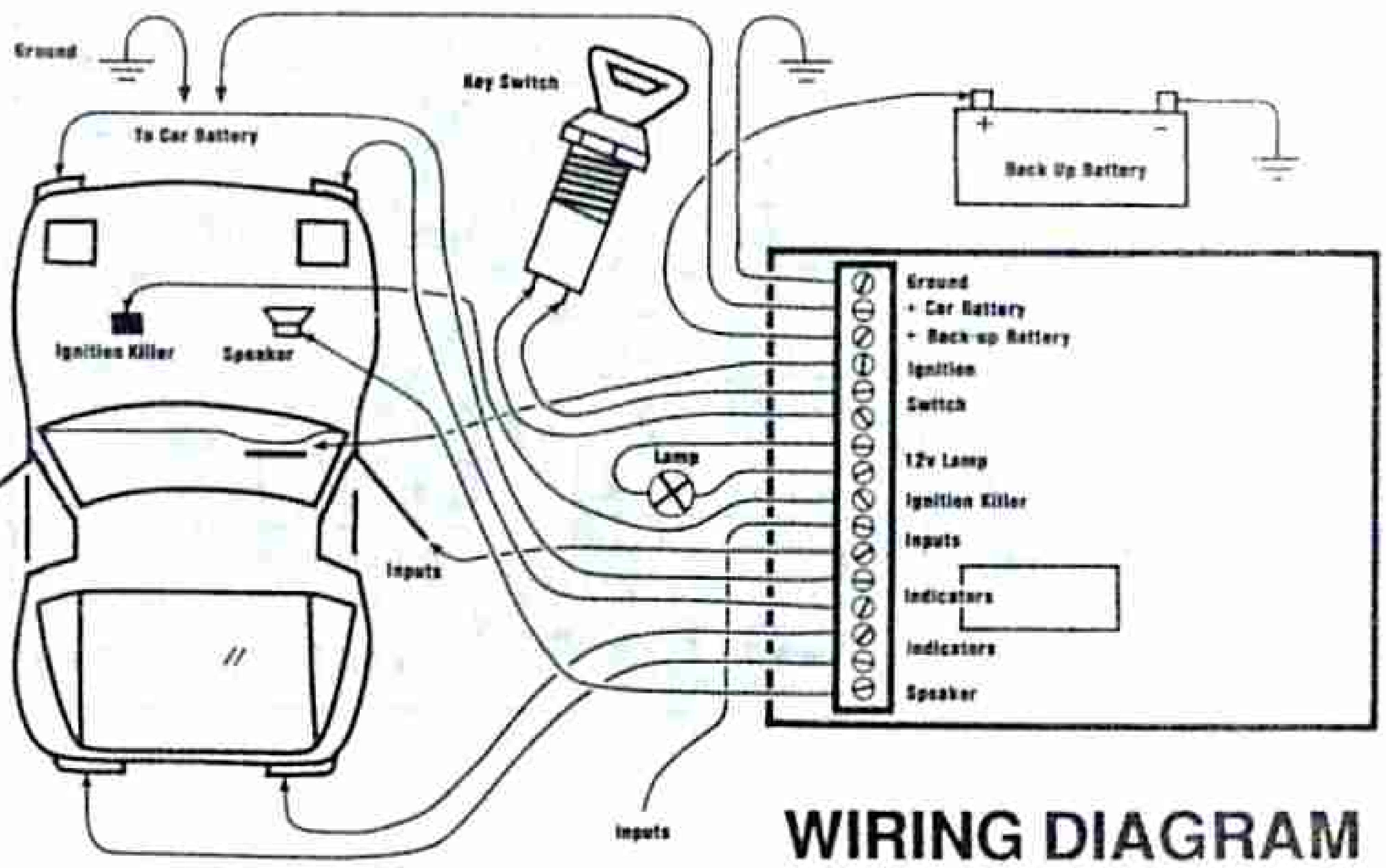
If the indicators do not flash at all, check that the diode between pin 10 of IC2 and pin 1 of IC3 is not open. Switch off the alarm and take pin 1 of IC3 low, pin 2 should go high and vice versa. While pin 1 is low pin 4 should be high, turning on the transistor which turns on the relay. If the relay is not activated and pin 4 is high, short between the collector and emitter of the transistor. If the relay still does not pull in, the transistor is faulty and should be replaced. If the diode between pins 2 and 3 is open, the indicators will never stop flashing.

If the siren doesn't work properly, switch the alarm off and take pin 9 of IC2 low. Pin 8 will go high enabling the first oscillator, at the same time taking pin 10 of IC3 low which should enable the final oscillator. If the diode between pins 10 and 5 of IC3 is open, a tone will be heard continuously from the horn speaker.

If there is never any sound, the fault will be in the amplifier section. Connect a multimeter to measure the voltage on the collector of the BC338 transistor. The meter should read about 12v. Connect a 10k resistor between +12v and the base of the BC338. The meter should now read approximately 0v.

When the reading on the collector of the BC338 is 12v, the collector of the BD140 should be approximately 0v and the emitter of the TIP31 should be 0v. When the reading on the collector of the BC338 is 0v, the collector of the BD140 should be 11.5v and the emitter of the TIP31 should be approximately 11v. If this is not the case, replace the suspect transistor.

If the ignition killer stage doesn't work, connect the meter to measure pin 10 of IC1. Switch the alarm on,



WIRING DIAGRAM

pin 10 should go low. Switch the alarm off and pin 10 should go high. If this is the case then the fault is in the ignition killer board.

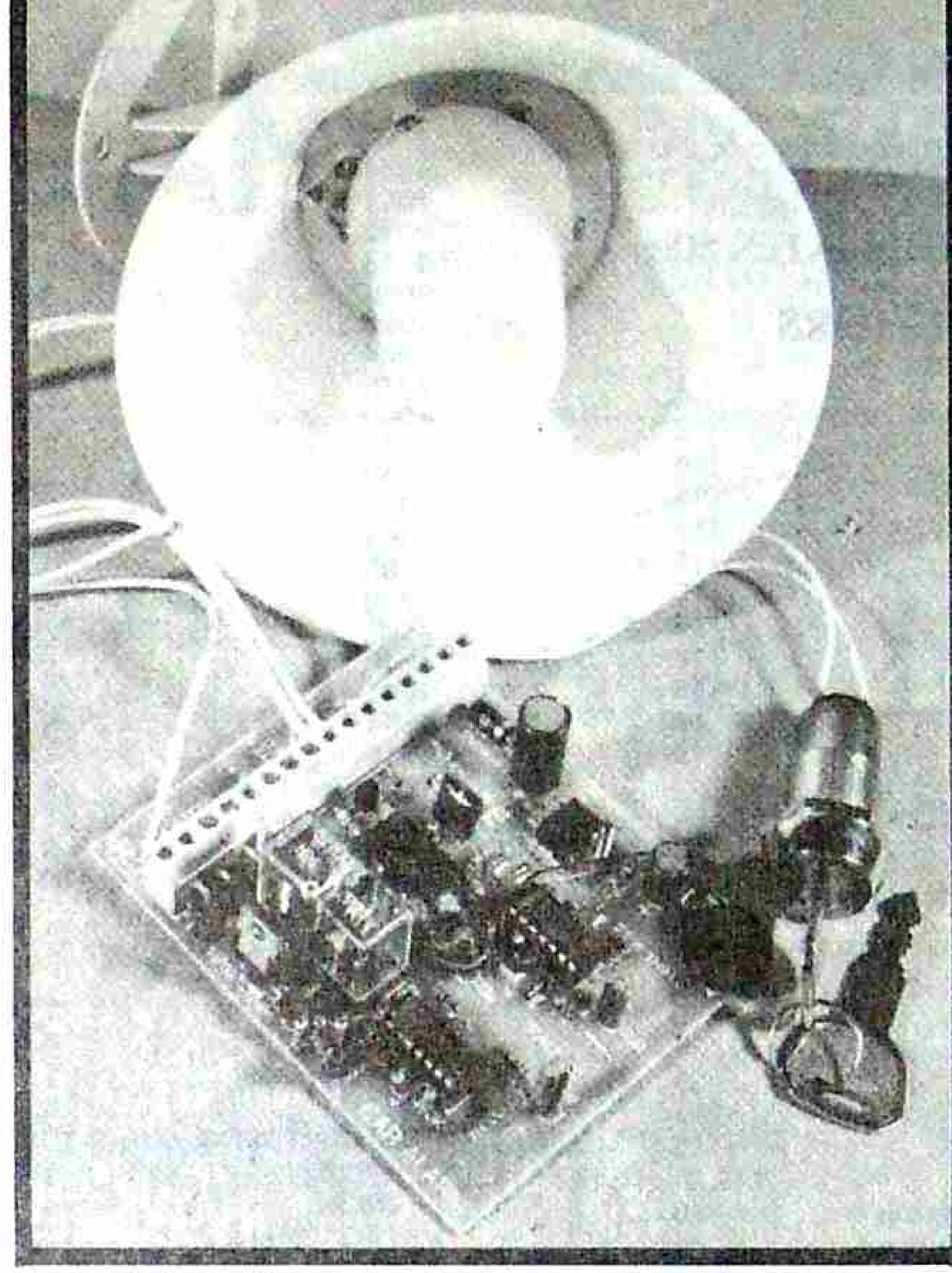
Switch the alarm on and short between the collector and emitter of the BC338 transistor. The relay should switch on. When the alarm is on, there should be 12v on the collector of the BC547 that drives the base of the BC338 to turn on the relay. When the alarm is off, there is 0v on the collector of the BC547 and nothing to drive the base of the BC338 and therefore the relay is de-activated. If this is not the case, replace the suspect transistor.

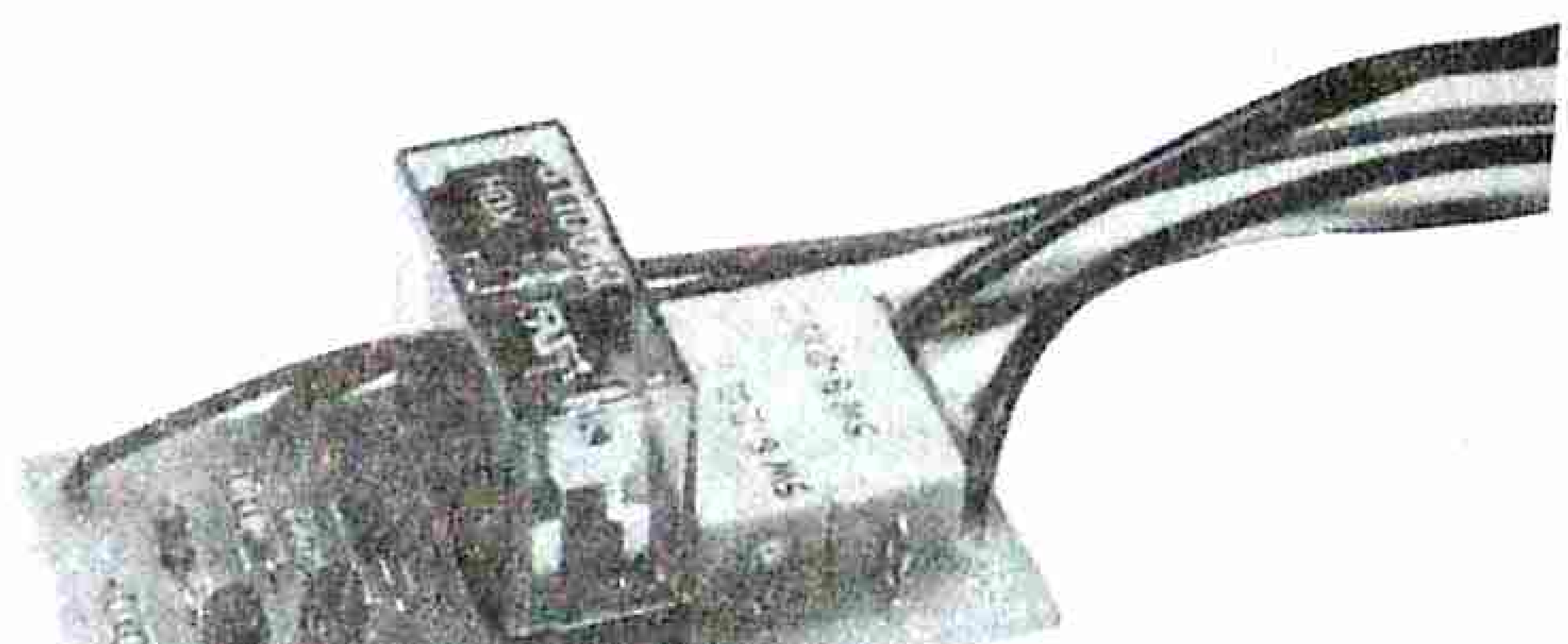
If you still have trouble repairing the alarm, we offer a backup service for repairing OUR KITS for a special price of $9.00 plus parts and postage.

If the parts are bought seperately through us and other retailers, the cost for repair by us will be $14.00 plus parts and postage.

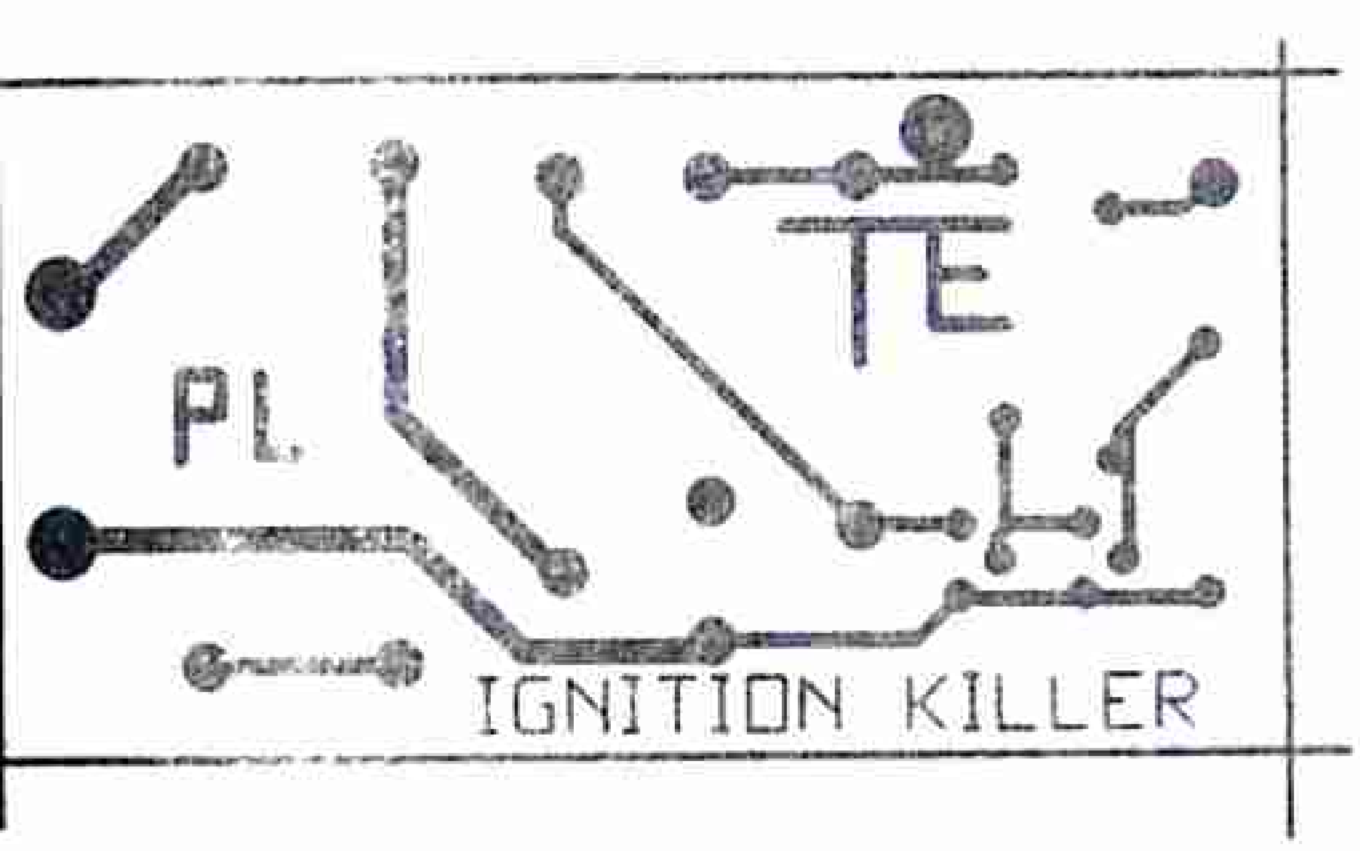
There will be add-on projects for the CAR ALARM in future issues. These will include:

* UHF REMOTE SWITCH
* ULTRASONIC MOVEMENT DETECTOR
* BATTERY BACKED UP SIREN

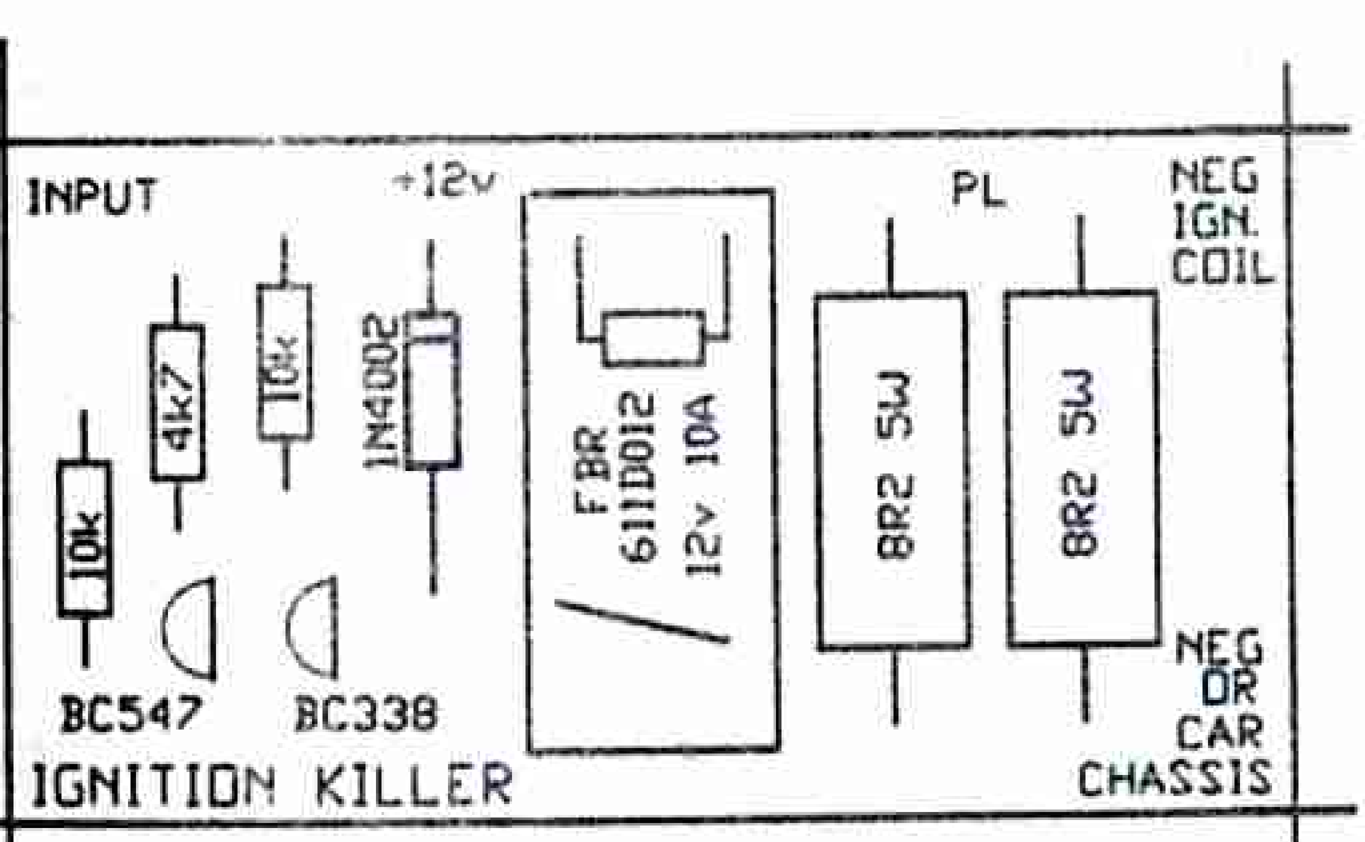


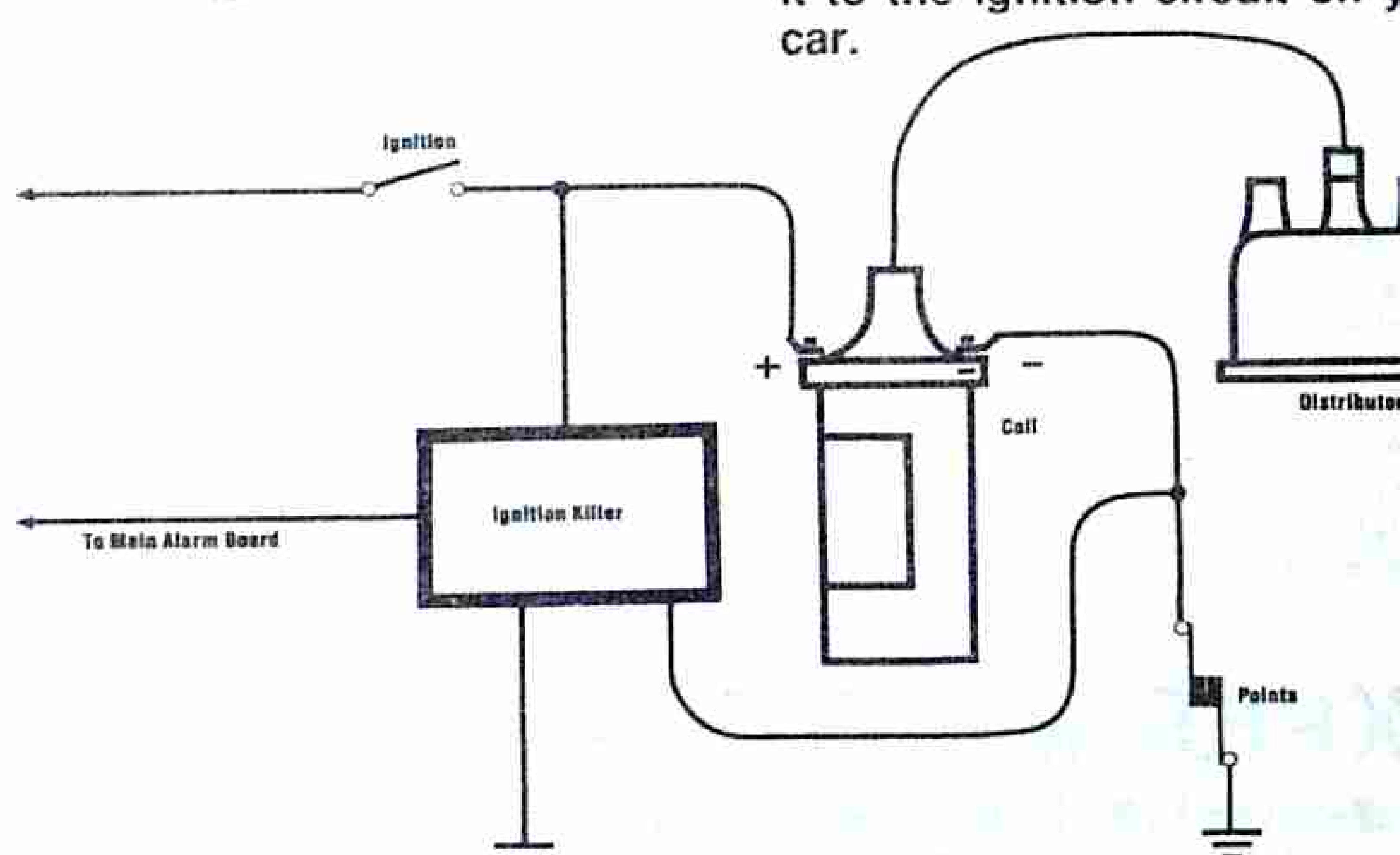


The ignition killer PC and wiring diagram to show how to connect it to the ignition circuit on your car.



IGNITION KILLER ARTWORK

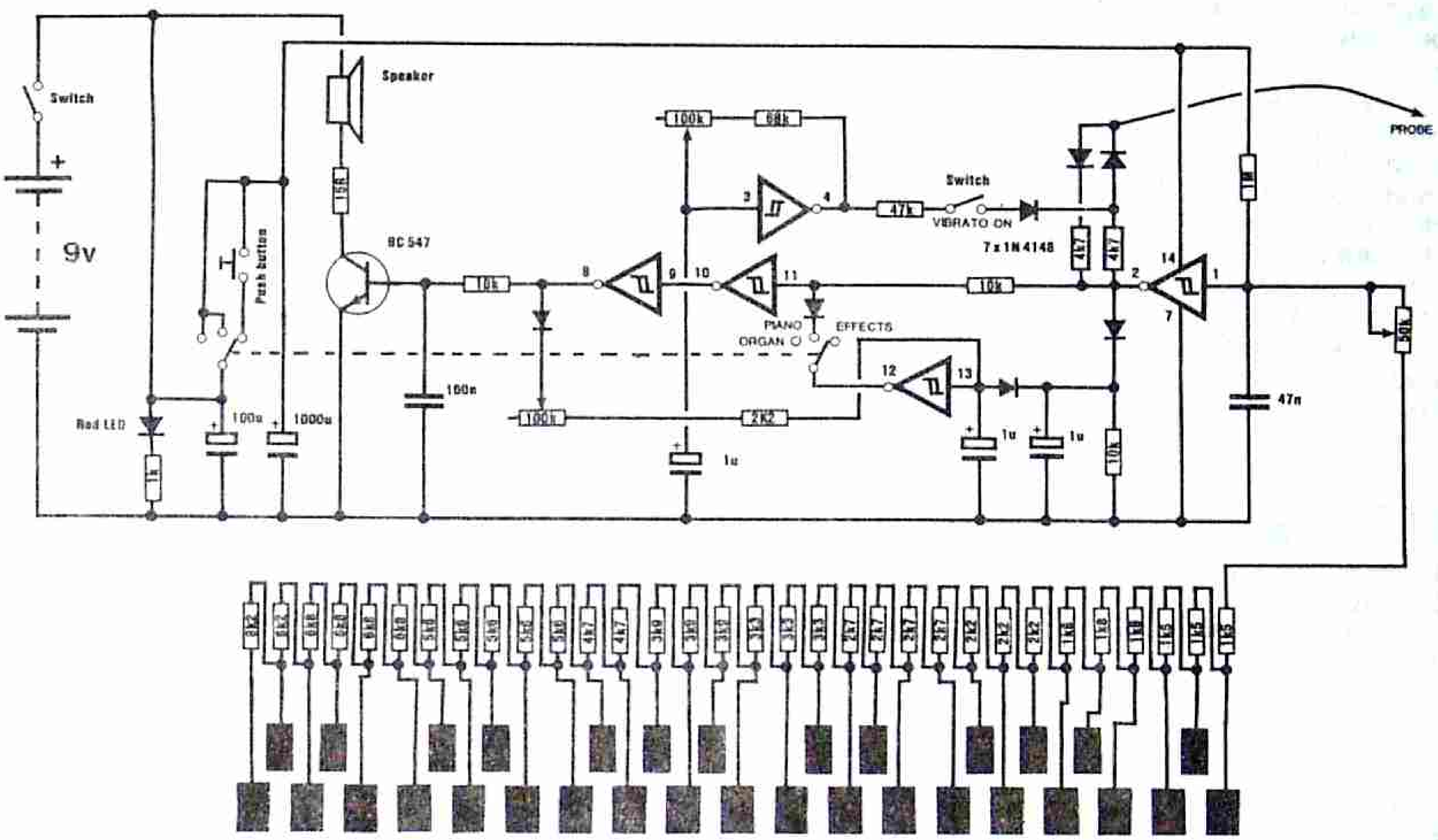




Build TE's:

ORGAN

A musical toy for young and old



ORGAN CIRCUIT

This is a great little toy for musical tinkering. It offers more than two octaves of notes and a variety of effects that will keep a player amused for hours.

You can put it together in a couple of hours and have lots of fun yourself.

My interest in a simple organ began some 20 years ago when Rolf Harris popularized a "Stylophone" organ. It used a stylus to play the notes and had a range of about two octaves and included vibrato.

I remember buying one of these toys and failed miserably to produce anything more than a few bars of the simplest tune.

However the concept never left me and over the years a number of similar stylus organs have been produced, both in magazines and on the market.

Many of the magazine circuits were grossly over-designed and too costly to build. None used a single chip like ours nor did they have the features we have built into our design.

Ours is a "skeleton" design and needs no case as all the components fit on the track-side of the board so that the keys can be accessed without having to turn the board over.

I am surprised no other designs have used this approach as it makes the best use of the board and allows it to sit on a surface without the need

PARTS LIST

1 - 15R
1 - 1k
3 - 1k5
3 - 1k8
4 - 2k2
4 - 2k7
3 - 3k3
3 - 3k9
4 - 4k7
5 - 5k6
4 - 6k8
2 - 8k2
3 - 10k
1 - 47k
1 - 68k
1 - 1M

1 - 50k mini trim pot
2 - 100k mini trim pots

1 - 47n
1 - 100n
3 - 1uF electrolytics PC mount
1 - 100uF 16v PC mount electro
1 - 1000uF 16v PC mount electro

7 - 1N 4148 signal diodes
1 - 5mm red LED
2 - 5mm red LEDs for trim pots
1 - BC 547 transistor
1 - 40106 Hex Schmitt trigger IC (effects section does not work as well with 74c14 IC)
2 - SPDT slide switches
1 - DP 3T slide switch
1 - PC mount push switch
1 - 9v battery
1 - 9v battery snap
1 - 14 pin IC socket
1 - 8R speaker
1 - 50cm hook-up flex
1 - paper clip for probe

1 - ORGAN PC BOARD

(pen barrel for probe not supplied)

for rubber feet etc.

With modern technology, an organ such as ours could be designed to fit into a birthday card, with touch sensitive keys and a single chip under a blob of epoxy. But unfortunately none of this technology is available to us in Australia and we have to be content with chips, resistors and a stylus.

As it stands, the Organ would make a great gift for a youngster and could even be adapted to go into a baby's play pen (without the stylus) to give various tones when large objects were touched.

We will leave this sort of adaption to you and show you how the basic model can be put together.

The use of a hex Schmitt trigger gives plenty of scope for effects and we have created 3 modes of play: Standard or Organ, Piano and Effects. The effects mode produces a gliding tone that will last quite a few notes and produce a lot of fun.

But the organ is not limited solely to playing around. By setting middle C accurately via a frequency counter or from someone with perfect pitch, you can play simple tunes that will sound quite authentic.

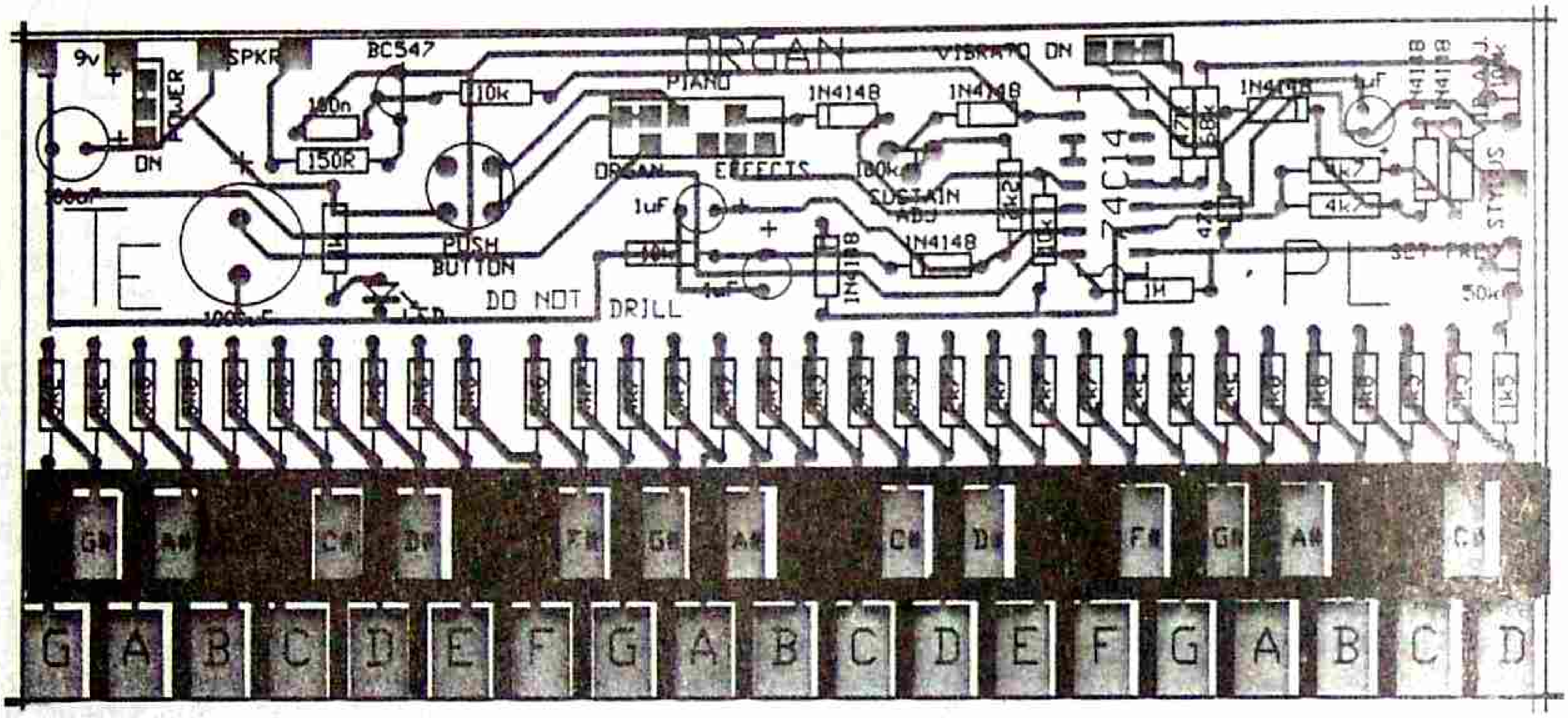
The set of resistors that have been chosen for the train gives an amazingly accurate scale and all the work of setting each of the frequencies has been done for you.

HOW THE CIRCUIT WORKS

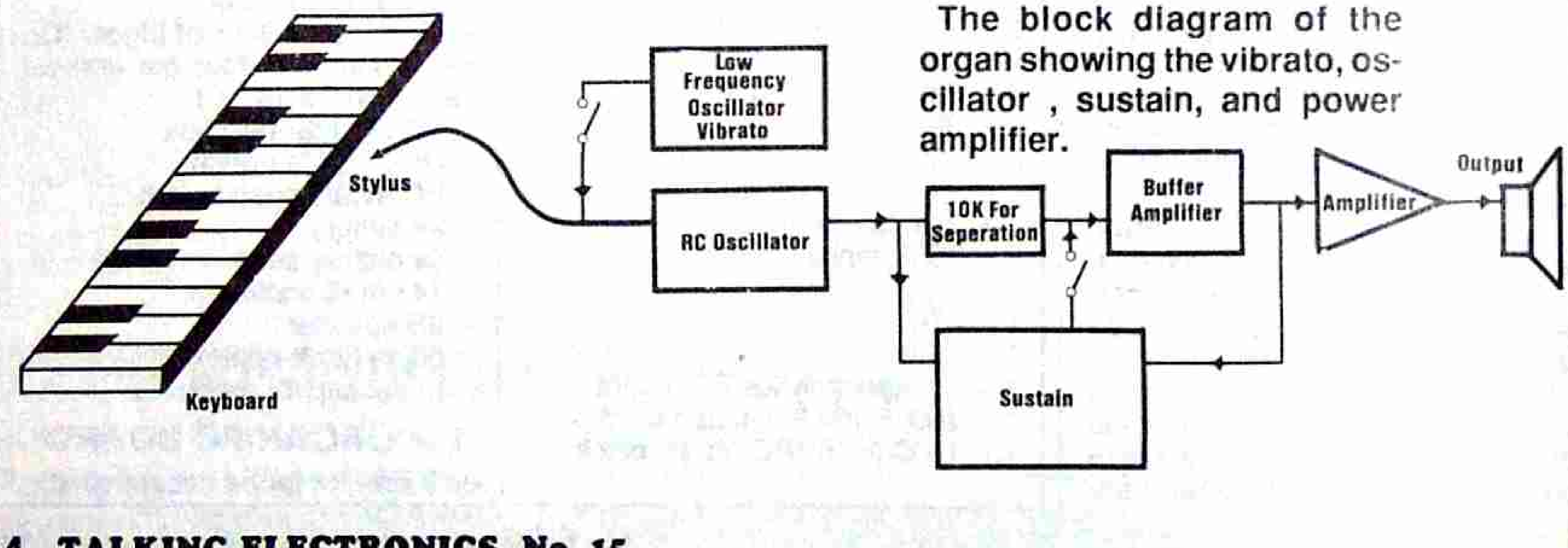
The organ has been designed around a voltage controlled oscillator made up of the Schmitt trigger between pins 1&2. The 47n, combined with the train of resistors connected to the keys, makes up the RC network and the stylus completes the path to the output of the chip.

The frequency of the oscillator depends on which key is touched by the stylus and these keys are laid out similar to that of a piano to help with understanding the keyboard.

Middle C is the fourth white note from the left and for those that are not familiar with a keyboard, middle C is the first note you look for when



The overlay acts as a resist to allow the keys to be accessed by the stylus. This makes a very effective keyboard.



The block diagram of the organ showing the vibrato, oscillator , sustain, and power amplifier.

presented with an unfamiliar keyboard.

Above the white notes are the black notes and you will notice we have labeled them with the sharp equivalent. This is due to the hatch symbol being available on the typewriter. As a point of interest, all the black notes have a flat equivalent such that A sharp is the same note as B flat. This will help you read music in B flat etc.

The output of the oscillator passes through two Schmitt buffer stages to a buffer transistor. The transistor drives an 8R speaker via a 15R current limiting resistor to give a pleasant sound for individual practicing.

When the vibrato section is fed into the oscillator via the vibrato switch, a 3Hz free-running oscillator is combined with the organ tone to produce a very pleasing effect.

The speed of the vibrato can be set via the 100k vibrato speed pot near the edge of the board.

When the selector switch is in the mid position, the "piano" effect is selected and this has the effect of chopping the note into a short burst, similar to a piano.

The length of the burst is controlled by the gate between pins 13 and 12. When it is switched into operation, the input line is initially low.

When a note is played, the input delay circuit, made up of the 10k resistor and 1uF electrolytic is charged via the diode on pin 2. This causes the cathode of the diode on pin 13 to go high and effectively puts its shorting effect, out of operation.

The 1uF on the input (pin 13) can now charge via the diode on pin 8 and the 100k "sustain adj" control.

When the 1uF charges to 2/3 supply voltage, the output of the gate goes low, decking pin 11, and inhibits the tone.

The third position of the selector switch is "effects".

In this mode the chip gets its supply from a 1000uF electrolytic and as the voltage falls an amazing thing happens....The frequency of the tone increases. You can take advantage of this by playing a few notes and see what happens. The push button will restore the charge in the electrolytic for further playing.

BEFORE YOU START

This project uses a special PC board in which the keys are laid out on the copper side of the board. So that the organ can be placed on a table, the components are also fitted to the copper side and this means no holes are needed.

This form of mounting creates a very effective project. You can see all the components as well as the track-work, at the same time.

It's great for a demonstration project, and when you are playing with the organ, you will be reminded of how it goes together.

Let us not kid you. Soldering the parts on the copper side of the board is more difficult than normal construction and you have to be very good at soldering to get a good finish.

But most important you will need a very fine tipped soldering iron to get under the IC socket and switches.

If you don't have a very fine tip, don't start, you will only mess up the board and damage the lands.

The temperature of the iron should not be too hot otherwise it will melt the glue that holds the copper tracks to the board. Yes, the copper tracks are stuck to the board with glue!

Since there are no holes in the board to provide added strength, it is important that none of the lands fall off.

The circuit may be simple and inviting but construction is at an intermediate level. If you are a beginner, you should ask for assistance - we can only tell you how to do it, we can't show you how.

CONSTRUCTION

With this project, the order in which you fit the parts is important.

Start with the 3 switches, the IC socket and the push button. Make sure the switches stand upright when soldering to the first land as they cannot be straightened by bending as this will rip the land off the board.

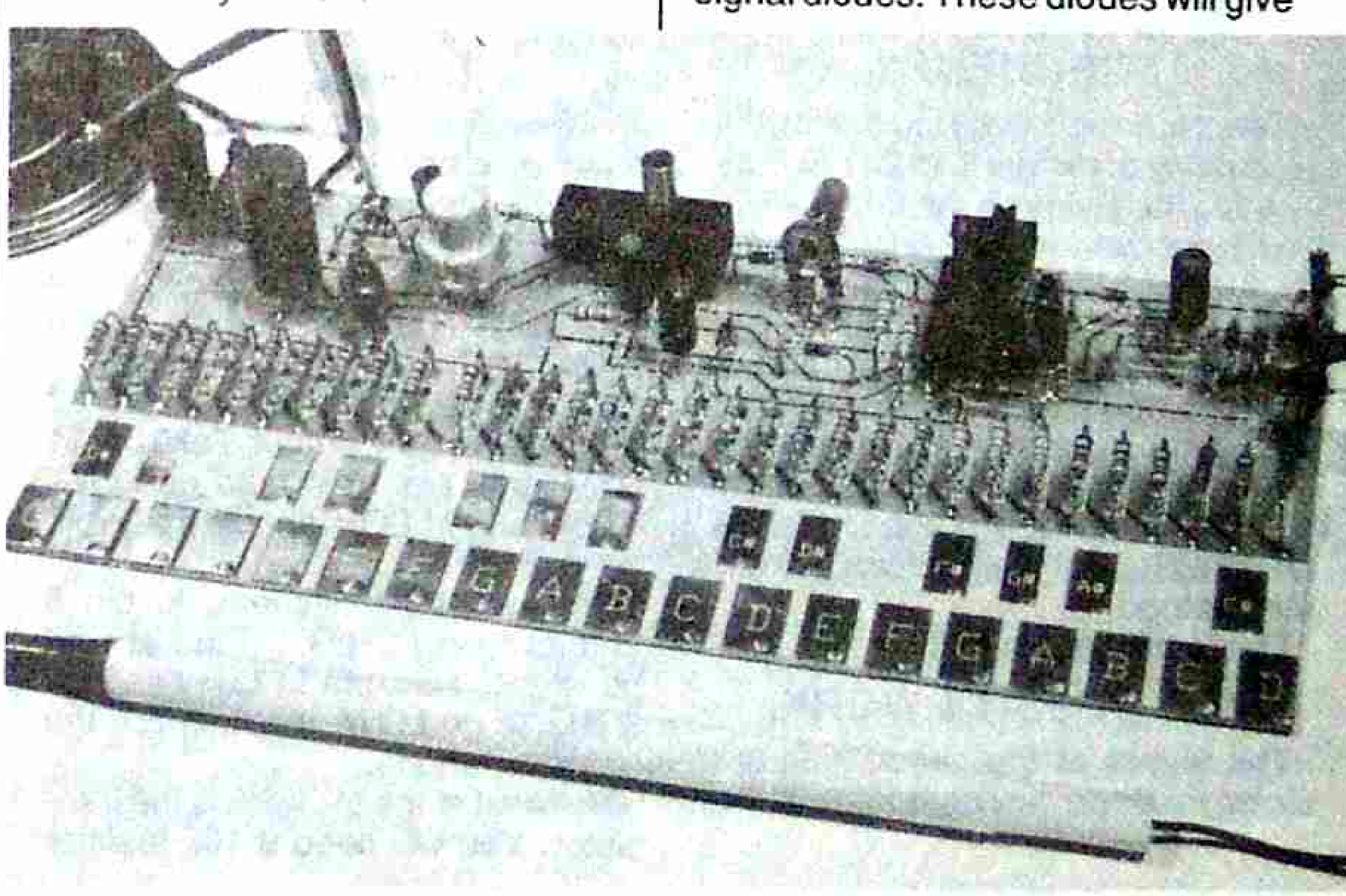
The notch (or cut out) for the IC is identified on the board and this indicates pin 1 is near the keys. Set the pins of the IC socket on the lands and "tack" two opposite corners. Make sure the socket is square with the side of the board and carefully solder each land so that the pins connect firmly.

The next components are the electrolytics and capacitors. Cut the leads so that they are 3mm long and tin them - this means adding a small dob of solder to each lead and land so that the actual soldering process requires little or no extra solder. You merely heat up the joint with the iron and the connection is made. This reduces soldering time and prevents the lands coming off.

Next fit the 3 mini trim pots. When these are in place, the sustain and vibrato pots have 5mm red LEDs soldered to the wipers so that the controls can be turned with the fingers.

When soldering the leads to the wiper, make sure solder does not run onto the fixed section otherwise the control will be frozen.

Next fit the "Power ON" LED and signal diodes. These diodes will give



you the first taste of the difficulty in soldering a component with short leads.

You may need a pair of long nose pliers to prevent burning your fingers and also to get components into awkward places.

The same care applies to the BC 547 transistor. Cut the leads slightly shorter and bend them to match the pads on the board. Tin the leads and the pads and use a pair of pliers to hold the transistor in place while the leads are being soldered.

You are now about half way, with only the resistors and a little wiring remaining.

Lay the resistors on the bench in the same positions as on the PC board so that you cannot make a mistake with the values.

Cut all the leads short and bend each to 90° so that they are all ready. Start at one end of the board and fit each as you come to it.

It does not matter which way around they are placed however if the tolerance band is kept to one end, it is easier to read the values when troubleshooting.

The stylus is made from an old pen barrel and the probe is a nail or paper clip fitted into the end. It connects to the PC board via a 40cm length of hook-up wire and to prevent the lead breaking off, it is tucked under the body of the nearest diode, to act as an anchor.

Connect the speaker via short lengths of hook-up flex and solder the battery snap in position.

The organ is now ready for setting the frequency.

SETTING THE FREQUENCY

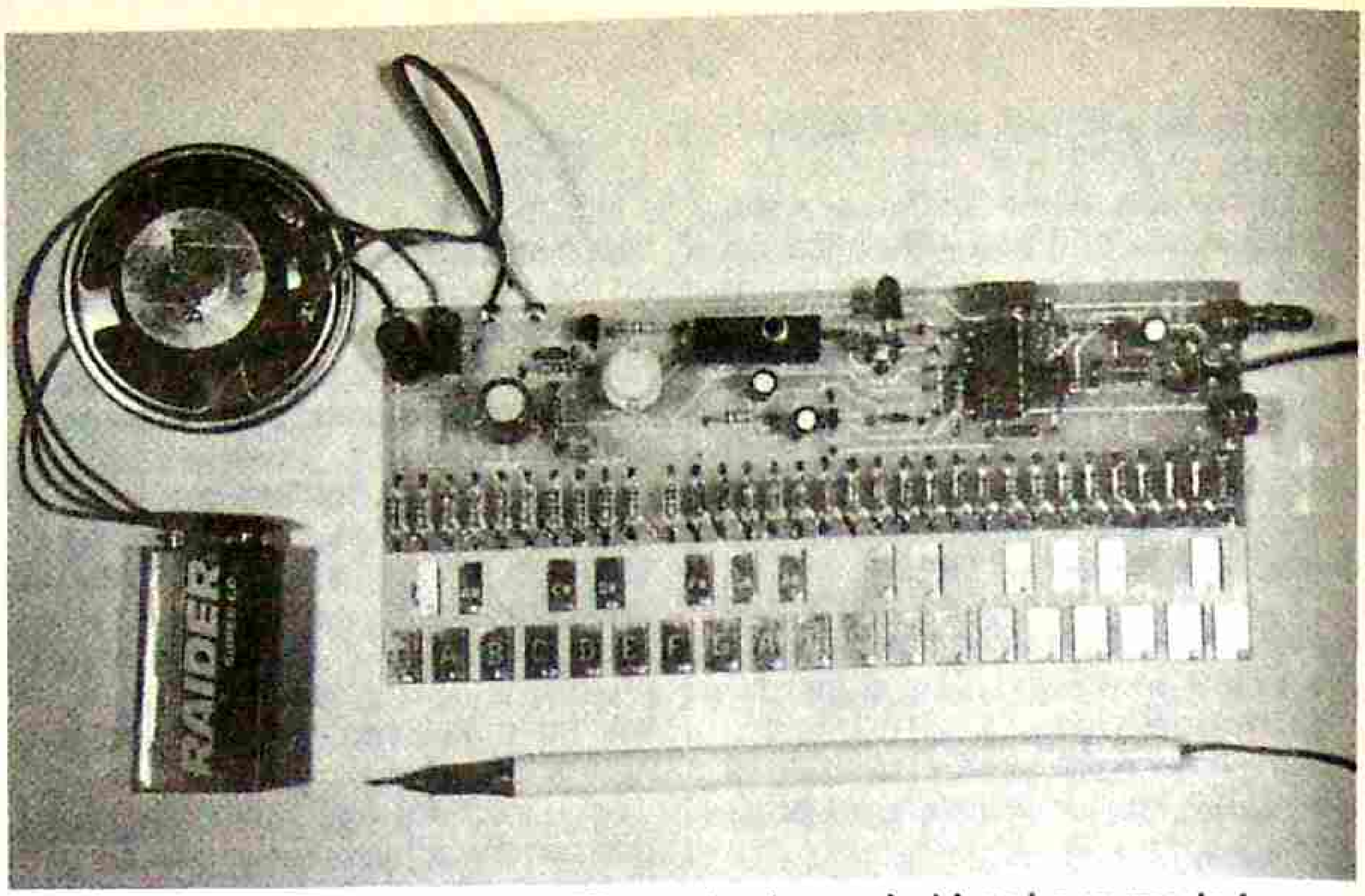
The organ is set by adjusting the frequency of middle C to 261.6Hz by holding the stylus on the fourth white key from the left and adjusting the frequency via the "set frequency" trim pot.

You can use a frequency counter or by comparing one of the notes of the organ with another musical instrument, you can listen for any "beating" between the notes. It is amazing how close you can get via this method.

That's all there is to it.

IF IT DOESN'T WORK

The layout of the circuit diagram closely follows the components on the board and this will assist fault finding.



Our organ ready for playing. The stylus (or probe) has been made from a ball-point pen barrel with a nail fitted into the nib end. The legend (or overlay) covers the tracks of the keyboard so that you can use the stylus to get a gliding effect across the keyboard.

If a tone is not produced from the speaker when the stylus touches the keys, the first thing to check is the position of the large slide switch.

It must be switched to organ or piano and NOT to the effects mode. Next check the "Power ON" LED, connections to the speaker, the battery voltage and the voltage on pin 14 of the IC.

If one of the diodes near the stylus comes adrift, the output will cease. You can check this by taking a jumper lead from either 4k7 resistor and any of the keys.

If there is still no output, short between pin 2 of the 74c14 and any of the keys.

For the next tests you will need a multimeter or logic probe. Connect the multimeter between pin 2 and negative and take pin 1 HIGH. Pin 2 should go LOW and vise versa.

If you do not get this result, there may be a short or bridge on the board, or the chip may be faulty.

If this gate works, move to pin 10 and measure the output. Take pin 11 HIGH and pin 10 should go LOW. Taking pin 11 LOW should make pin 10 go HIGH.

Connect the multimeter to pin 8 and take pin 11 HIGH. Pin 8 should go HIGH. Take pin 11 LOW and pin 8 should go LOW. If not, check the board for shorts.

The next stage to check is the transistor. You will need a 10k resistor for this test. Connect the multimeter (set to volts scale) so that the positive lead is taken to the +9v and the negative is on the collector of the BC 547. There should be no reading on the meter. Connect the 10k resistor between positive and base of the transistor. The needle should indicate a voltage. If not, check to see that no shorts exist and the transistor is soldered correctly.

Next the sustain stage. This only comes into operation when the 3-way switch is in the mid (piano) position. If this does not work properly, check pins 13 and 12 for inversion.

If the tone is only heard once when the stylus touches one of the keys, the fault is due to the 1uF capacitors not discharging fully and this could be due to either the 10k resistor being open or the signal diode being faulty.

If the tone does not turn off, the fault is the diode on pin 11 being open or the capacitors in the sustain section not charging. Test the diode on pin 8 for correct operation.

If the vibrato stage does not work, check the tracks on the board, the gate between pins 3 and 4 and the vibrato switch.

This is the first board off our CAD program and we are very pleased with its appearance.

If you are careful with assembly, you too will be pleased with the results and I am sure you will find someone who would love the Organ as a present.

TEC TALK

This page is for TEC owners. Through this, we can conduct a forum on the uses and future of the TEC. As we cannot reply to every letter sent in, we will attempt to answer letters of common interest through this page.

When writing in, put your letter on a seperate page if you are ordering kits etc. This helps us file things in some sort of reasonable order.

SENDING IN PROGRAMS VIA TAPE

We are looking forward to readers sending programs to us. There aren't any in this issue due mainly to the shortage of space. Hopefully, issue 16 will contain several pages or more.

A big factor in deciding weather or not we publish a "reader send-in" is the way the program is sent to us.

If you do have something to send, here is what we want you to do.

Provide us with a copy of the program Save it on tape with a crystal speed of half 3.58MHz. Put your name and address on the tape so we can send it back, if requested.

We also need documentation on the program. Write what it does and where it runs in memory and include any notes you may have generated. The first thing we will do is disassemble it and load it into our IBM clone. Here we can format it for publishing.

For the sake of our disassembler, please, if you can, put tables at the end of the program code and write down where the tables are located. This way we can use our HEX dump routine and tack the tables on at the end of the code.

JMON UP-GRADES

JMON has been designed to be upgraded without losing software compatibility.

Some likely changes are the removal of the low speed tape save (unless there is a storm of protest). This will decrease the software overhead in the tone routine and even-up the period measurement. The result will be an increase in the tolerance of different TEC frequencies and different tape speeds. This should make it possible to freely interchange half 3.58MHz and half 4.MHz tape software as well as allowing poorer quality tape players to be used.

The single stepper, which has no effect on the MONitor at all, may be shifted to a more specialized ROM to increase the stepper's abilities.

The keyboard and LCD RST's will not be changed, so any routine you write using these will run on future up-grades. The same cannot be said if you directly call into JMON. So don't do it!

ISSUE 15 CONTENT

Missing from the TEC section are two usual features. The reader send-ins and tutorial section.

The reason for this is mainly due to lack of room. Already the TEC section is the largest ever and the material left over will be a good start for issue 16.

A different direction is planned for issue 16. The basic lay-out will be two MAJOR add-ons and the rest of the article will be filled with programs (mine and yours, so send them in). There are a couple of reader send-ins that we intend to publish, so if you have sent something in already, we haven't just tossed it in the bin!

JIM's PACKAGE

This package is centered around JMON. The main feature is a complete line-by-line disassembly of the JMON ROM. I hope that, with careful study, you will be able to look at any instruction and understand its role.

My programming style is very optimized. Generally my programs are short and to the point. This does make them a little difficult to read but at the same time by studying and learning my ideas, your own programming abilities will be improved.

If the role of every instruction escapes you, you will still learn important concepts and a better way to do some things.

I wish I had a Jim's package when I was just starting out!

The package will be 20 pages long as this is the limit of our collating photocopier. If there is enough room, some other notes and programs will be included.

It is a pity that such a listing was not available for the earlier MONitors.

Because Jim's package contains every byte in JMON, you can actually burn your own JMON ROM.

Keep in mind that this means typing in 2k worth of program and one mistake will ruin the whole MONitor.

If you feel up to it then go to it. We don't mind you doing this ONCE for YOUR OWN USE.

This offer does not apply to schools or commercial buyers.

If you don't wish to try to type out JMON, I present you with this offer:

Purchase and pay for JMON and Jim's package together, and save $3.50.

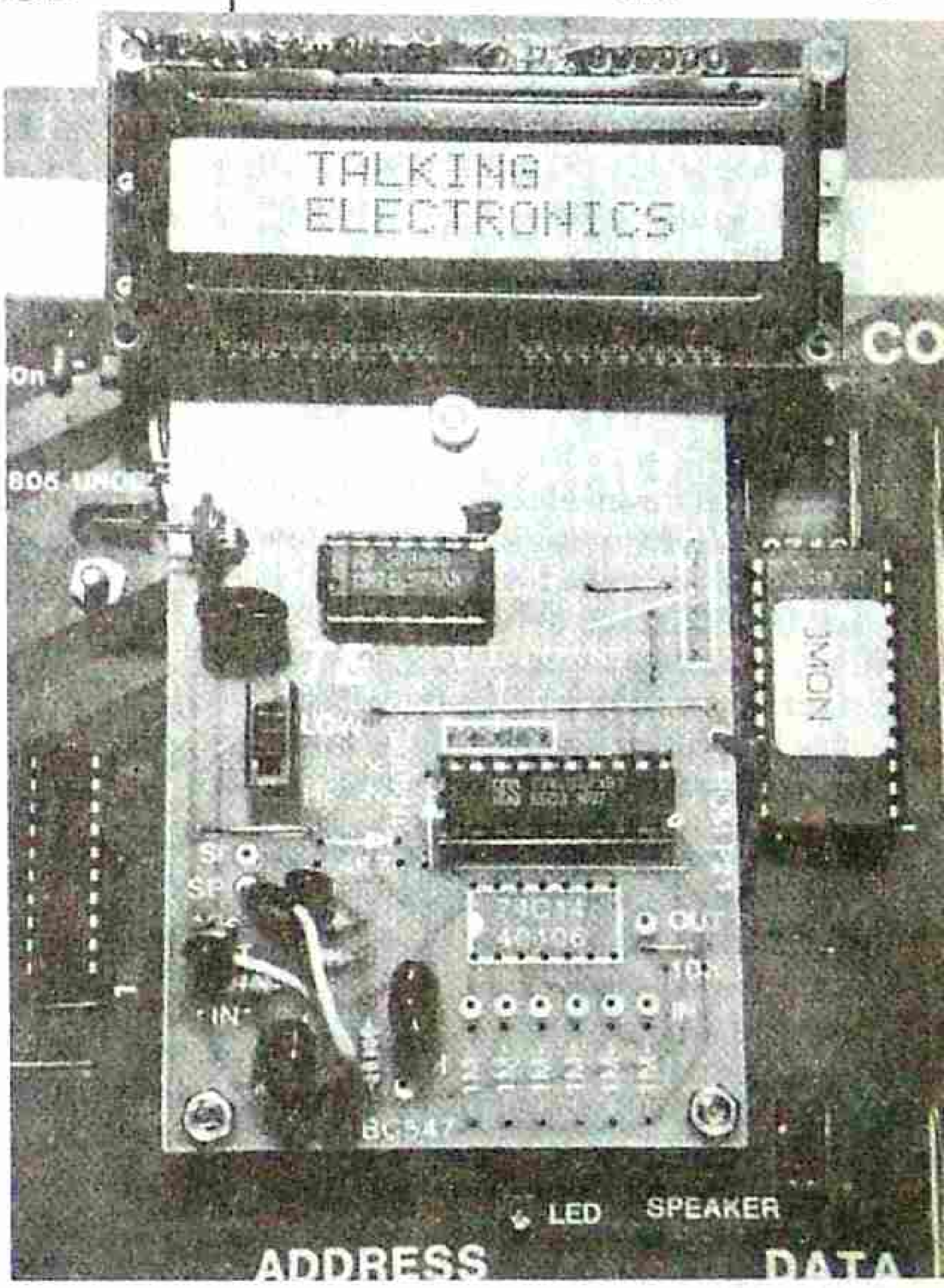
This means the total price for both is $27.50 instead of $31

ISSUE 16 and the TEC

Most of issue 16's TEC pages have been allocated and about half of these are already finished.

If you have some thing for us, don't waste time if you want the chance to see it in print.

- Jim



Completed DAT board displaying "TALKING ELECTRONICS." See the DAT board article starting on P. 47, and LCD article on P.52. Read the whole TEC article before starting anything!

INTRODUCTION

A discussion on Talking Electronics latest monitor for the TEC computer

Article and monitor by Jim Robertson

JMON is a big step ahead for the TEC.

Some of the contents of JMON are: a highly improved Monitor Program, a versatile Tape Storage Program, software for driving a liquid crystal display, a Menu Driver for utilities, a Perimeter Handler, User Reset Patch, Single Stepper and Break Pointer with register display software, and simplified access to utilities and user routines.

JMON also uses indirect look-up tables stored in RAM. This idea leaves the door open for many possibilities.

All the above and more is contained in 2k bytes.

The following is a description of the major blocks in the ROM.

THE MONITOR PROGRAM

To support new features added to the TEC, a new interactive monitor program has been written. The new monitor is, by itself, a considerable upgrade over previous monitors and when combined with other software in the monitor ROM, gives great features for the TEC user.

Major improvements have been made in the MONitor software, to allow quicker entry and editing of code. This has been achieved by adding such features as auto key repeat and auto increment. If you add the LCD, its larger display and cursor control software open up a second level of improvement.

THE TAPE SOFTWARE

The TAPE SAVE facility is versatile and reliable.

Some of the functions include: 300 and 600 baud tape SAVE, auto execution, LOAD selected file, LOAD next file, LOAD at optional address, TEST tape to memory block and TEST tape check sum. Both tests may be combined with other options.

The tape software uses the universal MENU driver and perimeter handler. These routines allow easy selection of cassette functions (e.g. Load, Save, etc.) and easy passing of variables to the tape software.

The tape software contains check-sum error detection that allows the user to know if the load has failed. A check-sum compare is performed after every page (256 bytes) and also after the leader is loaded. This means the user does not need to wait until the end of a load or test for error detection.

Each full page to be loaded, tested or saved, is displayed on the TEC LED display. Up to 16 pages are displayed.

Upon completion of a tape operation, the MENU is re-entered with an appropriate display showing:-END -S (END SAVE); PASS CS (CHECK SUM); PASS Tb (TEST BLOCK); PASS Ld (LOAD); FAIL CS (CHECK SUM); FAIL Tb (TEST BLOCK); FAIL Ld (LOAD).

The one exception is when an auto execute is performed after a successful load.

The tape software will display each file as it is found and also echo the tape signal.

LIQUID CRYSTAL SOFTWARE

This software is called from the monitor program. It is possible to de-select this software to allow the liquid crystal display to perform a user-defined purpose while the monitor is being used.

The Liquid Crystal Display is being accessed as a primary output device to the user during the execution of the monitor. Eight data bytes are displayed at a time and a space between each for the prompt (it appears as a "greater than" sign). Four digits in the top left hand corner show the address of the first byte.

In the bottom left hand corner is a current mode indicator and this lets you know which particular mode JMON is in. E.g. Data mode, Address mode etc.

The prompt points to the next location to have data entered, or if at the end of the 8 bytes being displayed, the prompt parks at the top left corner indicating a screen change will occur on the next data key press. This allows revision before proceeding.

It is possible to use the monitor with only the LCD unit, the only drawback being the actual current value of the address pointer is not displayed (the value shown in the address portion of the LED display). However this is only minor.

MENU DRIVER

This is a universal routine used to select various utilities routine from JMON. It is already used by the tape software and the utilities ROM. It may also be easily used by the TEC user.

The Menu Driver displays names of functions in the TEC LED display. The number of different names is variable and may be user defined. It is possible to step forward and backward through these names.

A 3-byte jump table with an entry for each name provides the address of the required routine. A jump is performed upon "GO."

To have a look, call up the cassette software by pressing SHIFT and ZERO together. If you have not fitted a shift key, the cassette software can be addressed by pressing the address key, then the plus key, then zero.

To move forward through the MENU, press "+". To move backward, press "-". Notice the automatic FIRST-TO-LAST, LAST-TO-FIRST wrap around.

Pressing "GO" will take you into the perimeter handler.

PERIMETER HANDLER

Like the Menu Driver, this is a universal program and may be easily used by the user.

This routine allows variables to be passed to routines in an easy manner. The variables are typically the start and end address of a block of memory that is to be operated on, such as a load, shift, copy, etc.

A 2-character name for each 2-byte variable is displayed in the data display while the actual variable is entered and displayed in the address display.

The number of variables may be from 1 to 255 and is user definable.

The data display is also user definable.

It is possible to step forwards and back through the perimeter handler in the same fashion as the MENU driver.

When a "GO" command is received, control is passed to the required routine via a 2-byte address stored at 0888 by the calling routine.

The SINGLE STEPPER and BREAK POINT handler.

A single stepper program can be important when de-bugging a program. It effectively "runs" the program one step at a time and lets you know the contents of various registers at any point in the program.

If you have ever produced a program that doesn't "run", you will appreciate the importance of a single stepper. Many times, the program doesn't run because of an incorrect jump value or an instruction not behaving as the programmer thinks.

The single stepper runs through the program one instruction at a time and you can halt it when ever you wish. By looking at the contents of the registers, you can work out exactly what is happening at each stage of the program.

The single stepper operates by accessing a flip flop connected to the Maskable Interrupt line of the Z-80. It can be operated in the manual mode, in which a single instruction is executed after each press of the "GO" key. In the auto mode, 2 instructions are executed per second.

BREAK POINTS

Break points work with groups of instructions. They allow register examination in the same way as a single stepper. The advantage of break points is that there is no time wasted stepping slowly through a program. This is particularly important as some programs contain delay loops and they may take weeks to execute at 2Hz!

Break points are one of the most effective ways to debug a program!

STARTING WITH JMON

JMON is straight forward to use. Some new habits must be learnt, however they are all quite easy.

JMON has 4 modes of operation. They are:

DATA MODE, ADDRESS MODE, SHIFT MODE and FUNCTION MODE.

The data address and shift modes are not new but have been, in part, changed in their operation. The function mode is new to the TEC and I am sure you will find it useful. Below is a description of each mode.

THE DATA MODE

The data mode is used to enter, examine and edit, hex code into RAM memory. It is identified by one or two dots in the data display and the word "DATA" in the bottom left hand corner of the LCD display. It is similar to the data mode on all previous MONitors.

The data mode has a sub-mode called AUTO INCREMENT. This is a default setting, meaning that it is set to auto increment on reset. The user may turn off the auto increment sub-mode if desired.

When in the auto increment mode, the current address pointer in the address display is automatically pre-incremented on each third data key press.

A SINGLE DOT in the RIGHT-MOST LED display indicates the current address will be incremented BEFORE the next nibble received from the keyboard is stored.

This allows the user to review the byte just entered. If an incorrect nibble is entered, the internal nibble counter MUST BE RESET by pressing the ADDRESS KEY TWICE. Then two nibbles may be entered at that location. This is a slight annoyance at first, but it is a small price to pay for such a powerful feature as auto increment!

After two nibbles have been entered, the prompt on the LCD is IMMEDIATELY updated and points to the next memory location, or in the case of the last byte on the LCD, the prompt PARKS AT THE TOP LEFT CORNER signifying an entire screen update UPON THE NEXT DATA KEY PRESS.

This allows the user to revise the entered code before continuing.

You must be in the data mode to perform a program execution with the "GO" key. (Actually, you can be in the SHIFT mode also.)

Because of the auto key repeat, and "auto increment", it is possible to fill memory locations with a value by holding down a data key. This may be useful to fill short spaces with FF's or zero's.

Because the LCD prompt is advanced immediately after the second nibble being entered while the LED display is advanced on the third nibble received, the "+" key will advance only the LED display while the "-" key will shift the LCD prompt back two spaces, if either are pressed immediately after the second nibble is entered. This may seem strange but is the result of a clever design which allows for revision of entered code on either display before proceeding.

ADDRESS MODE

This is identified by 4 dots appearing in the address display of the LED display and "ADDR" in the LCD bottom corner.

The address key is used to toggle in and out of this mode.

The address mode will be entered by an address key press from either the data or function mode. An address key pressed while in the address mode will result in a return to the data mode.

While in the address mode, data keys are used to enter an address while the control keys (+, -, GO) are used to enter the function mode. No auto zeroing has been included, therefore 4 keystrokes are required to enter any address.

SHIFT MODE

This mode allows easy manual use of the cursor. The shift works by holding down the shift key and at the same time, pressing a data key.

The monitor must be in the data mode and only data keys work with the shift.

Sixteen functions are available but only ten have been used in this monitor.

The shifts are:

Shift-zero: Cassette MENU is displayed.

Shift-one: Cursor back one byte.

Shift-two: Start single stepper at current address.

Shift-four: Cursor forward 4 bytes.

Shift-five: Break from shift lock (see function mode).

Shift-six: Cursor back 4 bytes.

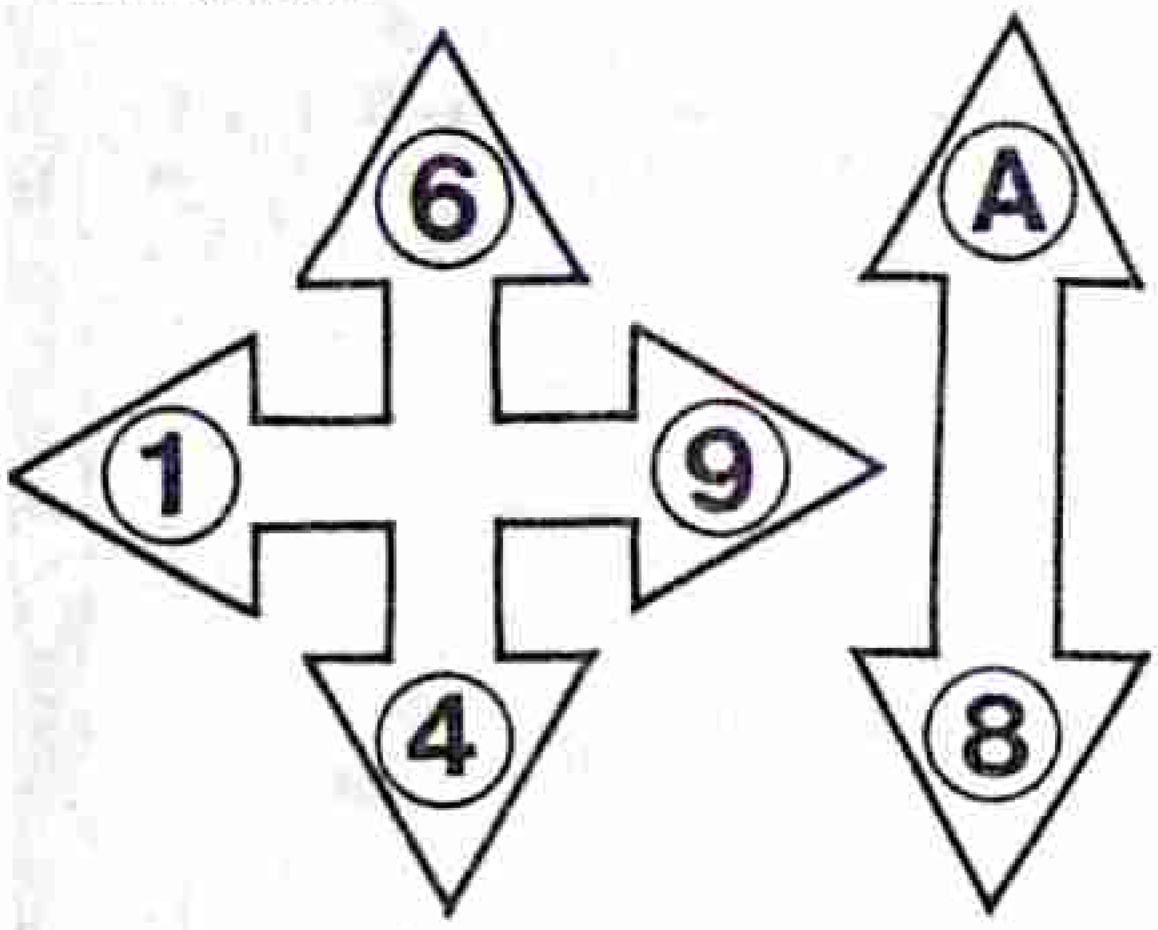
Shift-seven: Enter register examination routine.

Shift-eight: Cursor forward 8 bytes.

Shift-nine: Cursor forward 1 byte.

Shift-A: Cursor back 8 bytes.

Note that 1, 4, 6 and 9 form a cross and 8 and A form an arrow and each is positioned to correspond to their cursor movement.



Keys 1, 4, 6 and 9 move the cursor LEFT, RIGHT, UP AND down on the LCD.
Key "A" shifts the screen back to display the previous eight bytes.
Key "8" shifts the screen forward eight bytes.

When editing a program, the shift enables fast movement through the memory. Data entry is achieved by releasing the shift key.

The shift mode is not identified explicitly on either display.

THE FUNCTION MODE

This has been provided to enable a quick way to call commonly used routines. Only three keystrokes are required invoke up to 48 different routines.

The function mode is broken up into 3 sections.

They are: Function select-1, Function select-2 and Function select-3.

Each is identified by a single dot in the address display: right-most for function 1, second right for function 2 and third right for function 3. On the LCD display, the functions are identified by: Fs - 1, Fs - 2, or Fs - 3 in the bottom left corner.

Fs-1, Fs-2 and Fs-3 are entered FROM THE ADDRESS MODE by pressing the "+" key for Fs-1, the "-" key for Fs-2, the "GO" key for Fs-3.

It is possible to swap between sections without coming out of the current function mode by pressing the required function select key. After entering the required section, A DATA KEY IS THEN USED TO SELECT ONE OF SIXTEEN ROUTINES.

The address of these routines are stored in a look-up table.

SECTION-1 - the SHIFT-LOCK FEATURE.

Section-1 is selected FROM THE ADDRESS MODE by pushing the "+" key. The keys 0, 1, 2, 4, 5, 6, 7, 8, 9 and A then have the functions as listed in the shift mode. (Key 5 has the function of returning to the data mode.)

Cursor control routines return back to section-1 to enable continuous cursor movement (shift-lock).

The look-up table for the jump addresses for section-1 is at 07E0.

SECTION-2

Section-2 is selected from the address mode by pushing the "-" key. This is unused by any existing software and is available to the user.

HERE'S HOW TO USE IT:

Using the section-2 is very easy. All that is required is to enter the address(es) of the required routines in a table. The table begins at 08C0. The first two bytes at 08C0 correspond to the zero key in section 2. While the second two (08C2) correspond to key one etc.

Here is a short program as an example:

08C0: 00 09 04 09 08 09

(These are the addresses of the routines).

```
0900: 3E EB 18 06 3E 28 18 02
0908: 3E CD D3 02 3E 01 D3 01
0910: 76 C9
```

Now push ADdress, "-", "0" and the routine at 0900 will be CALLED from the MONitor. Reset the TEC and try ADdress, "-", "1" and ADdress, "-", 2.

Because these routines are CALLED from the MONitor, you may use a return (RET, C9 or RET NZ etc.) instruction to re-enter the MONitor in the same state as you left it. e.g. in the function select-2 mode.

SECTION-3

This has been reserved for the utilities ROM at 3800. The table for Section-3 is at 3820.

USING THE SINGLE STEPPER

Getting the single stepper to work is simple enough, however there is some skill required to understand its limitations and knowing how to avoid them.

To start with, you need a program that you require to be SINGLE STEPPED.

This program may be anywhere in memory except in the lowest 2k (the MON ROM).

This is because the MON select line is used as part of the timing. You may call into the MON ROM but only the first instruction will single step, and when returning out of the ROM, the next instruction will also not be stepped. (However they will be executed at normal speed.)

Programs that use the TEC's keyboard require careful attention as you cannot step them in the normal way. This is because there is no way to distinguishing between key-presses for the single stepper and those for the subject program.

This reduces the usefulness of the single stepper a little however thoughtful software design enables a fair degree of flexibility and this problem may be side-stepped.

The key use of the single stepper is as a de-bugging aid. When you are writing programs. effective use of the single stepper usually requires that while writing your programs, you allow for the use of the single stepper by leaving room to place one byte instructions that turn ON and OFF the single stepper.

Programs using the keyboard may be stepped by turning OFF the stepper. This allows areas requiring use of the keyboard to run in real time while other areas may be single stepped. This applies only to programs that use the keyboard routines provided inside JMON.

The only disadvantage here is that after completing your program you may have NOPs left.(from where you blanked over the single stepper control bytes).

The keyboard controls for the single stepper are as follows:

To start single stepping from the current address, this is what to do: From the data mode, press shift-2. This will start the single stepper. The first instruction will be performed and the address will be displayed as "PC" (Program Counter) on the single stepper. To examine the registers, press "+". The left two nibbles correspond to the high order byte and in the case of register pairs, the left-hand register. You may go backwards also by pressing "-". The registers displayed are : PC, AF, BC, DE, HL, IX, IY, AF', BC', DE', HL' and SP, in this order.

To step the next instruction, press GO. You can also step continuously at about 2Hz by pressing any data key.

When in the auto step mode, you can stop at any time and examine the registers by pressing "+" or "-", or bring it back to the manual mode by pressing GO.

The address key resets back to the MONitor unconditionally. The control bytes for the single stepper are as follows:

To stop single stepping in a program: F3 (disable interrupt).

To restart in a program: EF (restart 28). This causes a restart to 0028 where a routine passes the start address (which is actually the return address of the restart 28 instruction) to the single stepper. It also enables the interrupts and then returns to the next instruction which is then single stepped.

This SINGLE STEPPER is only a first model. Hopefully, when more room is available, some improvements can be added. One improvement on the "cards," is allowing it to be interfaced with a utilities ROM. This ROM will extend the display capabilities, allow editing while stepping and to disassemble on the LCD each instruction as it is stepped. If you have any ideas or requirements, write in and tell us.

BREAK POINTS

Break points are locations in a program where execution is stopped and the registers are examined in the same fashion as the single stepper. The advantages over single stepping include real time execution and less or no control bytes in a program. They also usually allow much quicker fault finding.

As a trade-off move, only a simple (but effective!) form of break-point is available with JMON. This allows for more MONitor functions and also eliminates the need for extra hardware.

More complex methods automatically remove the break-point control byte and re-insert the correct op-code and allows re-entry to the program.

USING JMON BREAK-POINTS

Break points are achieved by using a restart 38 instruction. The op-code for this is FF and all that is required is for it to be placed where ever you require your break point.

Before running your program, make sure the TEC is reset to 0900. This is necessary to clear the auto-repeat on the stepper/break-point register display. (This is explained in the LCD section).

Simply run your program as normal. When the break point is reached, the register display routine is entered. The value of the program counter display WILL NOT BE VALID on the first occurring break unless you provided the address of the break point at 0858. This minor flaw was unavoidable without considerable additional software which would have "eaten" memory like there's no tomorrow!

If you allowed for break commands in your program, you may then have multiple breaks and step to the next break with the GO key.

However if you placed a break command over an existing instruction then no further breaks will be valid and you should never try multiple breaks in this case AS YOU MAY CRASH THE MEMORY.

In the above case, make a note of the contents of the registers and return to the monitor via the address key and then examine memory locations, if required. (You may enter the register examination routine via shift-7). Further breaks should be done by removing the existing break and placing it where required and re-executing the program from the start.

Some other good ideas are to load the stack away from the MONitor's RAM area. (08F0 is good but make certain that 08FF does not contain AA - as this prevents the MONitor rebooting its variables on reset and your stack may have accidentally crashed these variables.) Also, if you are using the LED display scan routine in the MONitor ROM, shift your display buffer to 08F0 by putting this address into 082C/2D. Now you can examine your stack and display values after returning to the MONitor.

There is a conditional way to cause breaks. To do this requires a conditional jump relative with FF as the displacement. If the condition is met, the jump is made and jumps back onto the displacement which then becomes the next op-code! Remember this as it is a very useful idea. You cannot continue on with multiple breaks after a break caused by this method.

Break points are a quick way to debug a program. It is very important that you familiarize yourself with them. They have been the single most important programming aid used when writing most of JMON and the utilities ROM.

SUMMARY:

Clear the auto-repeat via the reset.

: Use FF to cause a break

: PC not valid on first break.

: For multiple breaks, provide spaces for the break control byte.

: Shift stack and display buffer (optional)

: Use FF as displacement for conditional breaks.

Finally, make sure you write down when, where and why, each time you insert a break-point.

ACKNOWLEDGMENT

Thanks to MR. C PISTRIN of Traralgon VIC. His SINGLE STEPPER program for the MON-1B inspired me to include one in JMON and provided me with a circuit for the hardware section.

See page 47 for the circuit

THE TAPE SYSTEM

This discussion covers all the areas needed to use the tape save and its various options.

TEC CONSIDERATIONS

The tape software works on any type of TEC, the only consideration is the various different clock speeds.

The following description generally applies to TEC's with a crystal oscillator that is fitted with a colour burst (3.58MHz) crystal and divide-by-two stage.

If you are still using a 4049 based oscillator, the tape system will work ok, but it will be very important to note the TEC clock speed when saving as the TEC must be set to the same speed when re-loading. Another problem can be the drift in frequency over a temperature range and the different oscillator frequencies between TEC's.

When saving a tape, the best idea is to wind the clock up to full speed, and then turn back the speed control pot one quarter of a turn. This will allow you compensate for speed drift if ever required.

The tape also works very reliably with a 4MHz crystal and divide by two stage, however a tape written using a 3.58MHz oscillator cannot be loaded by a TEC that uses a 4MHz oscillator, and vice versa.

If you are sending programs into TE on tape, they must be recorded with the 3.58MHz crystal. (divided by two).

The tape system has been extensively tested and found to be very reliable under a wide range of conditions. We don't expect you to have any trouble in getting it to work reliably for yourself.

LET'S BEGIN

To start with, you need a JMON monitor ROM as the tape software is inside this ROM.

Secondly, you will need a cassette recorder with both "mic" and "ear" sockets. Any audio cassette player of reasonable quality should be ok, provided it has the two sockets mentioned above.

We have tested more than six types, and found them to be quite suitable.

Thirdly, you will need to have constructed the cassette interface on the LCD interface board and have made up the two connecting cables, with 3.5mm plugs on each end. Finally you will need a new C60 or C90 cassette of the better quality types, such as TDK or Sony. We found the cheap tapes from the junk shops or supermarkets to be unreliable. (Some of them didn't work AT ALL, so don't take the chance).

Now connect the "mic" on the tape recorder to the "tape out" from the TEC and "ear" socket to the "tape-in" on the TEC. (It's a good idea to mark the cables between the recorder and the TEC to prevent incorrectly connecting the leads).

Insert a tape and we are ready to learn how to operate the system.

HOW TO OPERATE THE TAPE SYSTEM.

We will start by saving a few bytes at 0900. Enter at 0900 the following: 01 02 03 04 05 06 07 08 09 0A.

OK. Now connect up the tape recorder as described above and call up the tape software by pushing shift and zero at the same time or Address, "+","0" consecutively.

The TEC display will now show "SAVE-H" and this is the heading for SAVE at HIGH SPEED. Now select this by hitting "GO"

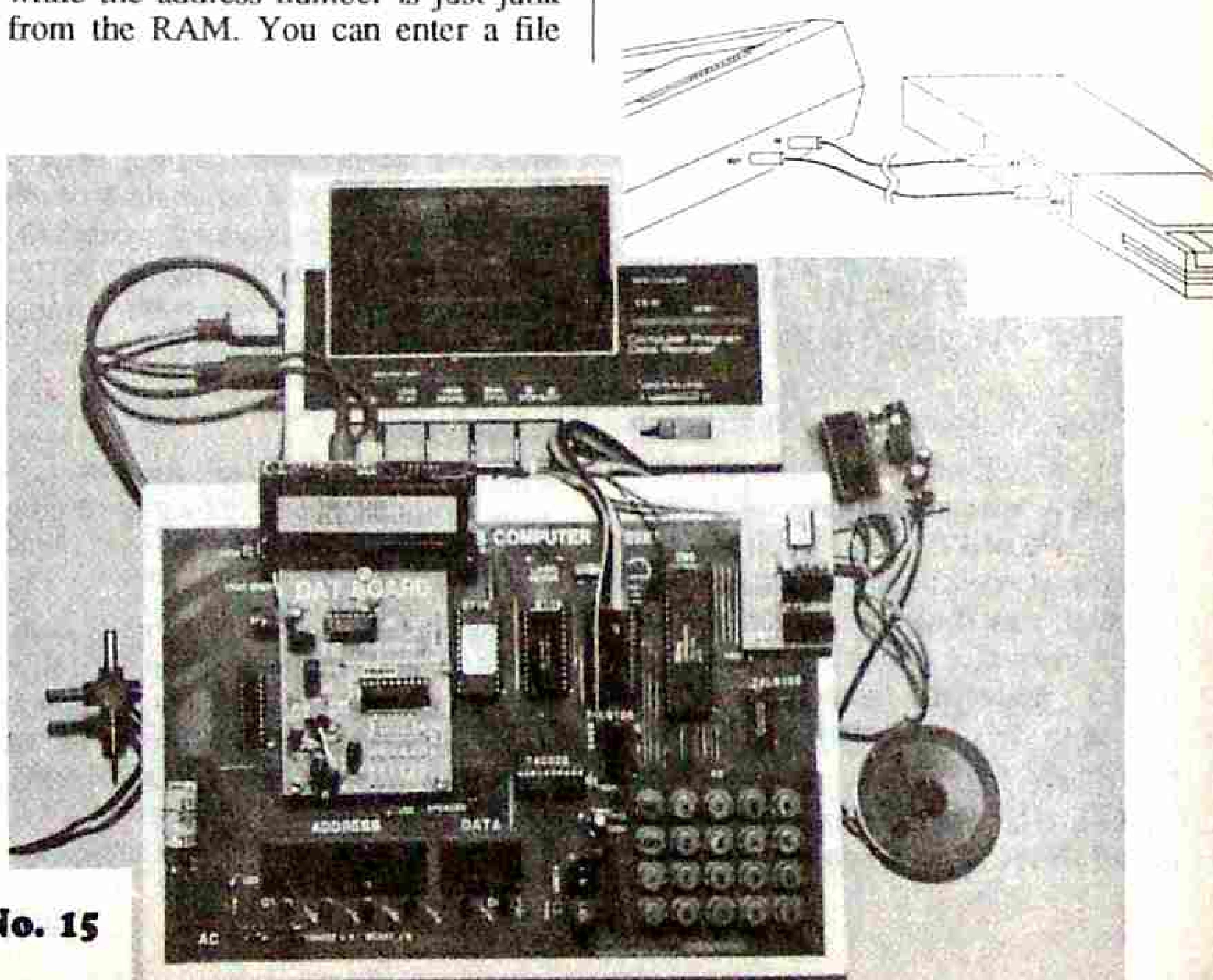
The display will now have a random two-byte value in the address display and "-F" in the data display. The "-F" in the data display is for the file number, while the address number is just junk from the RAM. You can enter a file number by pressing the data keys. Enter anything you want. The numbers you enter will shift across in the same fashion as when entering an address on the MONitor. Then when you have entered a file number, press the "+" key. The data display will now show "-S". This is where you enter the start of the block you wish to output. Enter 0900, and then press "+". The data displays now show "-E." This is where you enter the address of the last byte of the block to be saved.

Enter 090A and press "+". The next data display is "-G". This is the OPTIONAL AUTO-GO address, this is always set to FFFF by the software as this is its NON-ACTIVE state, i.e. NO AUTO-GO upon a re-load. ANY other value entered here will result in an automatic execution upon a SUCCESSFUL LOAD AT THE ADDRESS ENTERED HERE.

We don't require auto execution, so leave this at FFFF.

Now press play and record on the tape recorder and wait for the clear plastic leader to pass if at the start of a tape. Incidentally, it is not a good idea to remove this leader as has been advised in another magazine, as it protects the tape from stretching and possibly breaking, when rewound.

When the tape is right, press GO. The display will blank and a continuous tone will be heard from the speaker. After a few seconds the file information will be outputted and then a period of high frequency tone. This "middle sync" tone is to cover the time that the filename is displayed when re-loading.



After the high tone, the code will be outputted and also a digit will appear on the TEC LED display. This is the number of COMPLETE pages to be saved. In this case it will be zero.

A point to raise here is that if you ever accidentally enter a start address that is HIGHER than the end address, when GO is pressed, the software will detect this and display "Err-In". In this case, Push "+" or "-" to go back to the perimeter handler where you can correct the error.

When the code has been saved, a short end tone will be heard and then the menu will re-appear with "-END-S", meaning end of save.

Once the code has been saved, rewind the tape.

To re-load the tape press the "+" key and you will see "SAVE-L" on the display, then "TEST-BL", "TEST-CS", then you will come to "LOAD-T" (for load tape). Note that there is no "TEST-H" or "TEST-L" for low and high speeds as the test and load routine will load either speed automatically.

Press GO. The data display shows "-F" for file number. This will be as you left it when you saved. When loading or testing from tape, the file number here determines which file will be subject to the selected operation. If you enter FFFF here, the next file found will be used, regardless of its file number.

For now, we will leave it as it is.

Next push "+". The data display will show "-S", meaning Start address. This is always set to FFFF by the software. The start address allows you to optionally load a file or test a file at an address different to the one on the tape, (which is the address from which it was saved). To demonstrate its operation and to make it a more convincing trial, we will enter 0A00. The file will now be loaded at 0A00. If you press the "+" key again, you will be back at the file name. (This last point demonstrates the programmable number of "windows" feature of the perimeter handler. It was set up for 2 "windows" by a short routine entered from the Menu driver before passing control to the Perimeter handler, remember that there was 4 "windows" when you saved the file).

Now press GO.

The display will blank. Now start the tape playing. The sound from the tape will be echoed on the TEC speaker. Soon the leader will be heard and it should sound as crisp as when it was saved. If not, experiment with the volume. The interface allows for a wide variation of volumes but 3/4 volume is a good place to start.

After the leader has passed, the file name is loaded and should appear on the display. If it was not correctly loaded, "FAIL-Ld" will appear. In this case experiment with the volume and retry. After a few seconds the file name disappears and the number of complete pages to load are displayed on the middle digit. The code is now being loaded.

The code is loaded very quickly and hopefully a "PASS-Ld" will appear. If not, re-try with a different volume setting. After you have successfully loaded, hit reset and ADdress 0A00 and 01, 02 etc. will be found.

If you are unable to get a successful load after many attempts, then skip ahead to the trouble shooting section.

Now we have a successful load, we will experiment with the TEST BLOCK function.

Change a byte in the 0A00 block. Now call up the tape software (Shift-0, or ADdress, "+" ,0), select "TEST-BL", and LEAVE FFFF at the optional start. ("-S") Then rewind the tape and play it back like you did when loading.

At the end of the test, the display comes back with "PASS-TB". Now do this again, but this time enter 0A00 at the optional start and FFFF for the file. This will demonstrate the load/test next file feature.

Because 0A00 has been entered in the optional start "window", the test will be between tape and the code at 0A00.

Rewind the tape and press "GO" on the TEC, then play the tape.

Because a byte has been changed, the test this time will fail and the display will show "FAIL-TB."

Use the test-block feature whenever you wish to compare a tape file with a memory block or test that a save operation was successful.

If ever revising software on a tape of which you do not have a copy in memory, use the test checksum (TEST-CS) to ensure that the file is good. By use of the "LOAD NEXT FILE" feature (FFFF in the file number window) you can go through a tape completely, checking each file.

THE "AUTO-GO"

To use the Auto-GO feature, you must enter the required GO address WHEN YOU SAVE THE FILE. The go address is entered under the "-G" data display.

Experiment with the following:

0900: 21 10 09 11 00 08 01 06

0908: 00 ED B0 CD 36 08 18 FB

0910: 6F EA C6 EB E3 EB.

Save this as described above, but this time enter 0900 under the "-S" heading, 0921 under "-E", and 0900 under "-G".

Now re-load it and if the load is successful, the program will start automatically and an appropriate display message will appear.

USING THE TAPE SYSTEM.

The primary use for your tape save system is as a mass storage device for your files.

Files may be saved and loaded as described previously, the important addition here is good paper-work habits. It is very important to keep a log of your files or you will quickly forget what you have, where it is located, and you will end up writing over your files!

Your log system must include identifying each cassette and the side of the tape, the files on the cassette in the correct order, how many of each file, the date and any notes on the file. If your recorder has a tape counter facility, it makes good practice to record the readings from this, so that files may be quickly found anywhere on the tape.

Also a great aid is to log approximately the location of each file e.g. half-way, 30 seconds from rewind from end etc.

Apply the above idea to the start of vacant area on the tape also.

Another very good way to use the tape system is as a "RUNNING LOG", where a whole side of a cassette is used to save a developing program, stage-by-stage. If you crash your program, you can re-load it back from tape. A good idea here is to use the high byte of the file number as the program identification and the low byte as the progressive count or version number on the tape.

When you have a final version, then save that on a permanent cassette. The "RUNNING LOG" cassette can then be used over and over.

Once again, paper work is very important. Make sure you document any differences between successive files. This may help later in de-bugging. Also, always include the date and time as this will give a chronological order to your work.

If you are wondering how many times you should save a file, and at what speed, the answer really depends on the reliability of your system.

The major factors in reliability are your tape player and cassette quality and how well you constructed your interface. If any of these are borderline, the system may work but you may have a higher than normal failure rate. Our tests show reliability at better than 98% on saves of 2K blocks. Different cassettes and players were used over many months and rarely did a fault creep in.

You can test your system out by saving the monitor 10 times on each speed and then perform a BLOCK TEST. You should get at the very least, 17 out of 20 passes. If not, some trouble-shooting may be required. If you get 19 or 20 you could probably get away with high speed saves and not have to worry about checking them on your running log. For permanent storage, a good system is a high speed save, then two low speed saves and check each afterwards.

The low speed save should be more reliable than the high speed save as the low speed save will tolerate the occasional hiccup. However, this extra reliability does not cover all possible causes of failure, e.g. problems related to frequency or bandwidth restrictions of your tape player as the period is not changed only the ratio of pulses.

Finally, a file that is absolutely necessary to be retrieved from tape must be stored on two tapes. This provides a double back-up facility against accidental erasure or damage.

"OH NO!" IT DOESN'T WORK.

If your tape system fails to work correctly, then check the interface board or better still, have a friend check it.

Eliminate any problem and re-try.

If problems still exist, test the cassette player with a normal pre-taped audio tape. The music should sound normal and not flutter. If it flutters, the tape player is due for a service or replacement, or if battery operated, the batteries may be flat.

Various sections may be eliminated by listening to the tape signal. If the signal saved on the tape sounds ok when played back on the player, but is not heard on the TEC, check the input section of the interface board and also the "E" output of the player with a pair of Walkman-type headphones.

It is possible that the volume output is not high enough to be amplified on the interface board.

This is very unlikely though on ordinary tape players but we found this to be the case with our VZ-200 data cassette player.

If no signal is getting to the TEC and everything else seems to be ok, test the input buffer by setting the tape software to load and taking the input high and low with a jumper lead.

The LED on the speaker should echo the inverse of the input. If not, shift the jumper to the collector lead of the input transistor and repeat the process.

If the speaker LED now toggles, the input transistor is faulty.

If not, investigate the latch chip. Make sure all the pins are well soldered and the feed-throughs are connecting properly.

If the tape signal is heard, in the TEC speaker, but the file number is not recognised, loaded correctly, or the tape fails to load the data blocks consistently, try a better quality cassette tape. If problems persist, try a different player as the signal may be distorted or not have enough amplitude.

If you still can't get it to go, a repair service is available for $9.00 plus $2.50 postage.

RUNNING OLD PROGRAMS WITH JMON

Most old programs will run with JMON without to much alteration.

For most, they will only need to be relocated from 0800 to 0900 and that's all. Those that use old MON-1 routines such as the running letter program or the tune player can't, of course, run with JMON.

Of those which use the keyboard, most can be easily altered but some require a complete overhaul.

These ones listed below cannot run on JMON, or require more extensive changes that those presented here:

SPIROID ALIENS, HALILOVIC'S PIANO, BIG BEN CHIMES, WINNERS CALL, YOU'RE DEAD FUNERAL DIRGE, TOCCATA, THE STRIPPER, ADDING AUTO REPEAT, AUTO RETURN AND STOP, AUTO MOVEMENT AND HALT, THE ROMMED PRINTER SOFTWARE and SPACE INVADERS SHOOTING.

Those not mentioned should run ok if re-located to 0900 and the mods listed below are done, if required. These mods apply only to routines which use the keyboard.

TEC KEYBOARD THEORY

Basically the keyboard usage of earlier routines is broken up into two types: Those which halt and wait for a input via the interrupt, and those which intialize the input buffer (the interrupt vector register or I register) and read it "on the fly."

The first type may again be broken into two groups. Those which explicitly read the value from the input buffer, with a LD A,I instruction (ED 57), and those which assume the input to be inside the Accumulator after the interrupt has occurred. (Remember the earlier MON-1 series did not save the accumulator during the NMI routine but instead returned with the input key value inside the accumulator. A disastrous state of affairs)!!

JMON

The specially-provided routines in JMON will work for all the above types, the only difference being the way you alter the program in question. Here's how to alter the routines:

To up-date any type which uses a HALT instruction (76H) as part of the keyboard input section, change the HALT instruction (76H) to a RST 08 (CF).

The RST 08 routine SIMULATES the halt instruction by first looping until NO key is pressed then looping until a key IS pressed.

After a key press is received the input value is masked to remove unwanted bits and stored in the input buffer, (the interrupt vector register), in identical fashion to the old interrupt routine.

Now if the halt instruction is immediately followed by a LOAD A,I (ED 57), you may leave it as it is or remove it as it is not required any more as the input value is returned in A.

If a program doesn't have a HALT instruction but uses the keyboard, then look for the LOAD A,I instruction (ED 57). Change this to a RST 20 (E7) and place a NOP over the unused byte. Notice that this IS NOT the same RST instruction as above.

Be careful not to mistake the LOAD A,I (ED 57) with a LOAD I,A (ED 47) otherwise you program may get upset and go on strike.

Programs which have neither a HALT or LD A,I instruction cannot be altered by any of the above methods because they enter a continuous loop and require the interrupt to force an input value into the accumulator. A classic example of this is the "space invaders shooting" on page 14, issue 14. This above loop is located at 0821. (while you're looking at this, grab a pen and change the byte at 0812 from 02 to 01, at least it will run correctly with MON-1)! All the above types are among those listed as not being suitable for modification via these methods.

FINALLY

If you find a program which doesn't work (we haven't tried them all) or something else interesting, please write and let us know.

USING THE KEYBOARD IN YOUR PROGRAMS

The new keyboard set-up is no more difficult to use now than before. In actual fact it is easier and requires less bytes than before thanks to the use of the RST instructions.

Four RST's are provided to handle the keyboard in different ways. The first RST we shall look at is RST 08 (CFH). This RST is a "loop until a NEW key press is detected" routine. If you refer to the section on running old programs, you will see that this RST is used to simulate/replace the HALT instruction. (You know how to use it Already!)

An important feature of this RST is that it ignores any current key PRESSED, that is if a key is being pressed when this RST is performed, it will not be recognized. This mimics the NMI which only recognized a key press once. (This is why the auto-repeat feature could not be done with the keyboard hooked up to the NMI).

When this RST detects a valid key press, it inputs the value from the key encoder and masks the unwanted bits and stores the input in the interrupt vector register (as did the MON-1 series). The input value is also returned in the accumulator. The shift key can not be read from this (or any other MONitor keyboard routine) as the shift input bit (bit 5) is masked off.

Here is an example of its use:

```
0900  CF       RST 08
0901  FE 12    CP 12
0903  20 04    JR NZ,0909
0905  3E EA    LD A,EA
0907  18 06    JR 090F
0909  FE 01    CP 01
090B  20 F3    JR NZ,0900
090D  3E 28    LD A,28
090F  D3 02    OUT (02),A
0911  3E 01    LD A,01
0913  D3 01    OUT (01),A
0915  18 E9    JR 0900
```

The first thing you should notice when you enter and run the above, is that the "go" key is not detected when the routine is first started, even though it is being pressed. This is because the first part of the RST loops until the key being pressed is released. The RST then loops until a new key press is detected. When the RST returns, the input value is both in the interrupt vector register and the accumulator. The rest of the routine tests for either a 01 or "GO" key and outputs to the display.

Use this RST when ever you want the TEC to go "dead" and wait for a key press.

The second RST is RST 10 (D7). This is similar to the first RST but has one very important difference. The difference is that this RST DOES NOT wait for a key being pressed to be released before returning. While this is not as likely to be used as much as the first RST, it does have some good uses. Any program which requires some action to take place while there is a key pressed, but do nothing when there is not, may make good use of this RST. Some possible uses include random number generation on the time the key is held down; count while a key is pressed; turn on a relay while a key is pressed etc. As you can see, this RST simulates momentary action switches.

This RST exits with the input stored in the same fashion as the above RST.

The third RST is RST 18 (DF). This is a LED scan loop and keyboard reader. The scan routine will scan the 6 TEC LED displays once with the display codes addressed by the address at 082A. (0800 is stored here by JMON. You can leave it as it is, just store what ever you want at 0800 before using this RST). After the scanning routine is done, the keyboard routine is called. The keyboard routine is actually called from RST 20. What happens is this. After the scan has been called from the RST 18, the program continues on at 0020, which is the start address for RST 20. So the RST 18 is the same as RST 20 EXCEPT THAT RST 18 CALLS FASTSCAN. Therefor the description below applies to BOTH RST 18 AND RST 20.

This keyboard routine is very intelligent and is able to detect several different conditions.

One important feature is that it "remembers" if it has already detected the one key press and it ignores it if it has. This provides us with a "ONE AND ONLY ONE" key recognition for each key press. Each key press is "recognized" on the first detection.

The key is checked for a "FIRST KEY PRESS" by the use of a flag byte. When the routine is entered AND NO KEY IS PRESSED, this flag byte is CLEARED. When a key is detected, the flag byte is checked. If zero, the key is accepted as a "FIRST KEY PRESS." The flag byte is then set to stop further "validating" of the same key press. The input value is then masked and returned in the Accumulator (only).

If the flag byte IS NOT CLEAR, then the key is not recognized as "valid."

Careful consideration was giving to the interaction of the MONitor and user routines so that the "GO" command from the MONitor WILL NOT BE TREATED AS THE FIRST KEY PRESS of a user routine. (This was achieved by using the same flag byte for both JMON and any user routine).

HOW TO INTERPRET THIS RST

If a key is recognized as a "FIRST KEY PRESS" then the ZERO FLAG will be set to its active state (a logic 1) and the MASKED KEY INPUT will be returned in the accumulator.

If the key is NOT valid then the ZERO FLAG will be clear AND the accumulator WILL HAVE ALL ITS BITS SET (FF).

(FOR ADVANCED PROGRAMMERS)

In addition to the zero flag being conditionally set, the RST 20 (E7) also sets the carry conditionally, according to the following conditions:

If there is a key pressed then the carry will be SET REGARDLESS of whether it is a "first key pressed" or NOT. If NO key is pressed then the carry is cleared.

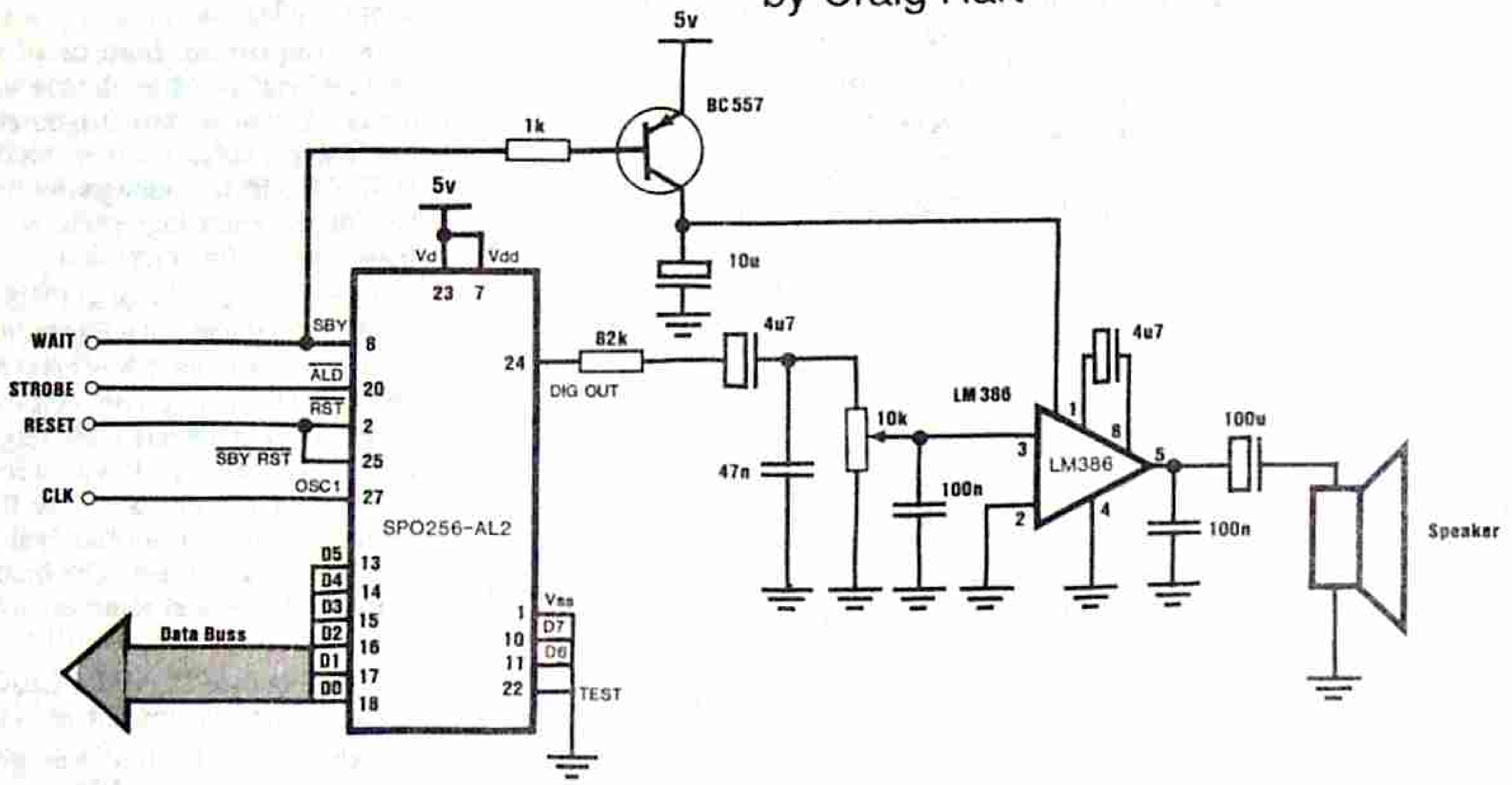
This allows you to interpret the keyboard the way you want, while still giving you the convenience of using the RST to do some of the work.

Jim's section cont P 47.

Speech Module

Add speech to your TEC!

by Craig Hart



SPEECH MODULE CIRCUIT

Since the dawn of time, Colin has been fascinated by electronic speech synthesis, so it was with immense joy that we discovered the SPO256A-AL2 speech chip. This chip is a universal speech unit that can be made to speak almost any English word. The price was cheap and the interface was minimal, it was just too good to pass up! So I took took up the project and this is the result.

The module is interfaced to the TEC, and the TEC controls what is said. The only requirement is that you have a crystal oscillator, as the module requires a 3.58MHz clock signal from the unit. Demonstration programs have been included for testing and simple word sequencing, and these programs will show how the unit is accessed.

This is the ideal companion project to go with the I/O board, and a robot created out of the two projects will cause a real stir if it speaks a comment in response to what it is sensing in its environment.

The module is connected via an 8 way ribbon cable and 4 flying leads. The ribbon cable picks up D0-D5, and the 5v supply. The other 4 leads connect to STROBE 05, WAIT, RESET, and CLK. Note that only the lower six bits of the data bus are used by the speech chip. The reasons for this will be explained later.

OPERATION

The operation of the unit is straight forward, but it is important to understand its operation so that you can use it once you have built it. The SPO256A-AL2 is made to speak by sending it a series of ALLOPHONES. An allophone is the smallest individual sound that the unit can speak. Words and sentences are formed by outputting a series of allophones, one after the other.

Each allophone is assigned a number and this number is loaded into the chip via the TEC data bus, then the ALD line is pulled low (by strobe line 05).

The SPO now commences to speak the allophone and indicates so by pulling the WAIT line low, halting the TEC until the module is ready for more data. The BC557 is turned on hard by this and the LM386 amplifier is switched on.

Sound is clocked out of the unit at a rate determined by the CLOCK line. For normal speech this is 3.58MHz. Sound is filtered by an R-C network, to make the sound more "human like" and amplified by the LM386.

PARTS LIST

All resistors 1/4W 5%
1 - 1k Brown Black Red
1 - 82k Grey Red Orange

1 - 10k trimpot.

1 - 47n greencap.
2 - 100n monoblock.
2 - 4u7 electrolytic.
1 - 10u electrolytic.
1 - 100u electrolytic.

1 - BC557 transistor.
1 - LM386 amplifier IC.
1 - SPO256A-AL2 Speech IC.
1 - 8 pin IC socket.
1 - 28 pin IC socket.

1 - 8 ohm speaker.
4 - PC pins.
4 - PC pin connectors.
1 - 20 cm length 14 way ribbon cable.
1 - 24 pin DIP header.
1 - 10 cm length 2mm heatshrink tubing.

1 - 'SPEECH MODULE' PC board.

When speech output ceases, the wait line goes HIGH, and the TEC is able to continue processing. In doing so, the BC557 is switched off and thus the LM386's power supply is switched off. The reason for doing this is due to the high input impedance of the chip; it is prone to picking up stray noise. The most common noise source is the scanning of the LED displays! This results in an uncomfortable buzz when the unit is not speaking and by switching the power to the amplifier this has been eliminated.

THE ALLOPHONE SET

The SPO has little intelligence about what you want it to speak. You cannot simply feed it a word, and have it say the correct pronunciation in every case. (Although other chips do have this capability) Instead you, the programmer, must translate each word into the appropriate allophone(s) for that word. There are 64 individual allophones, and each sounds different. In these 64 allophones, there are 5 pauses of various lengths, corresponding to word and sentence breaks.

By consulting the Allophone reference table you can look up what you think the right sequence is then play around with different pronunciations of the same basic letter, until you reach the best sounding word. It can be a tedious process, but many common words have been pre calculated and a list appears at the end of the article, along with the table of individual allophones.

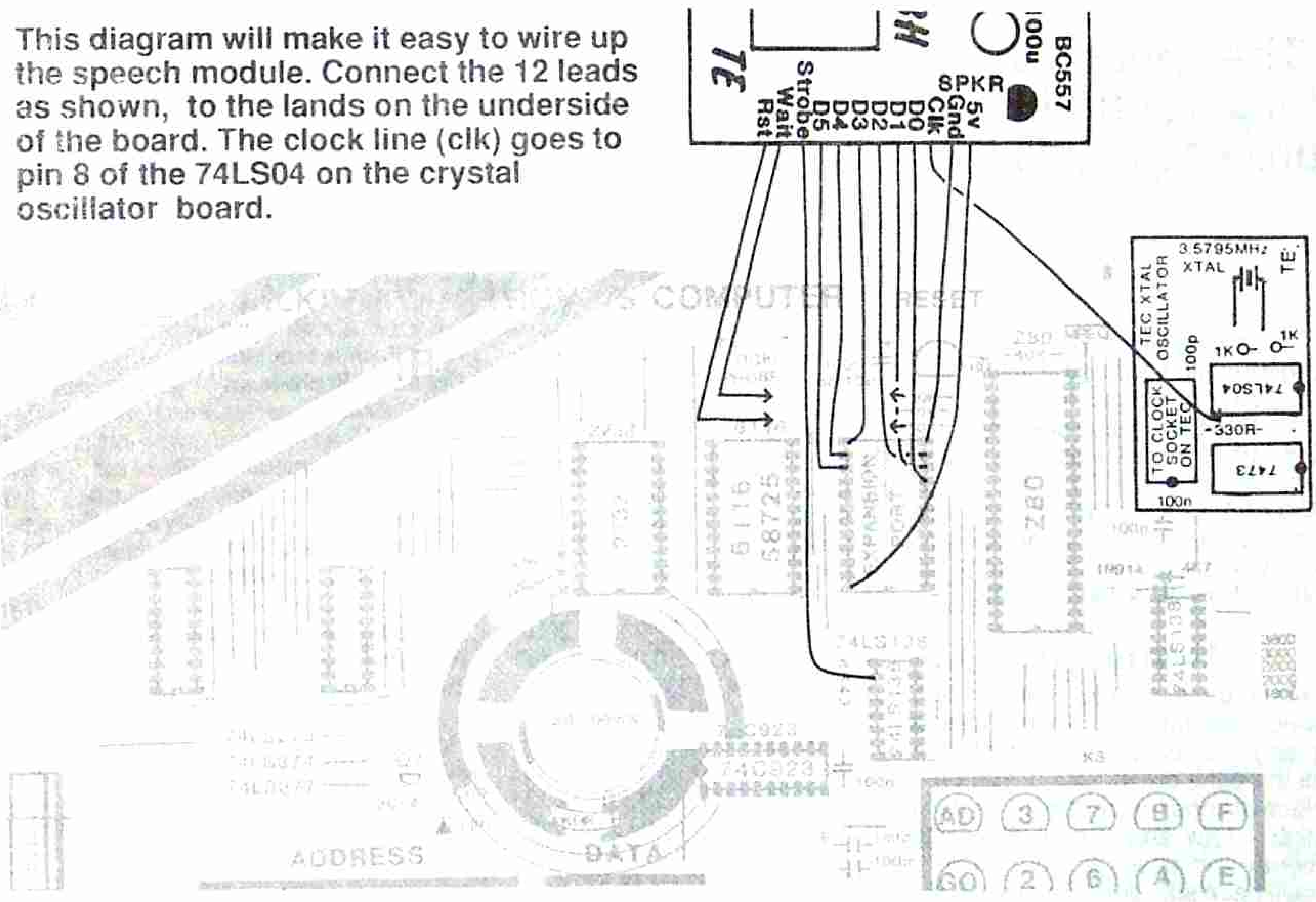
Take a sample word : ALARM. Sound out the word slowly, letter by letter. Now look for a matching sound in the list. Write down your guess and progress through the word. Where you have two or more choices, pick the allophone of the appropriate length. For alarm, I chose AA LL AR MM, or 18 2D 3B 10. Add a pause to the end and the terminating byte 04 FF. Plug the data into the test program at 0910 and run it.

It sounds a little cut-off in the first 'a', so try a longer 'A' i.e. AX (0F) and try again. Enter 0F at 0910 and run the program again. Sounds better now doesn't it!

By following this method, you should be able to come up with any word within a short space of time. Remember, the secret is to sound each letter and syllable out and then search for the best allophone of the group. The sample word provided gives you a context in which the allophone is used. This is useful when deciding between TT1 and TT2 etc.

We also discovered that it was much easier to produce an understandable word if you used the slang way of saying it. The speech module always produces the same type of sound for any given allophone, so if you stick to spelling only, then the words always come out very strange. If you use slang then you will find that the resulting word is much easier to understand.

A perfect example of this came up when we first started work on the project. We bought our first sample chip from Tandy. It came with a list of words and full specification data. When the project was working, we started trying some given examples, and although the examples were recognizable, they were not very clear. Then Ross said to try the slang pronunciation. Voila! perfect. The words which were before just average became clearer and much more recognizable.



This diagram will make it easy to wire up the speech module. Connect the 12 leads as shown, to the lands on the underside of the board. The clock line (clk) goes to pin 8 of the 74LS04 on the crystal oscillator board.

PAUSES AND REPEATING ALLOPHONES

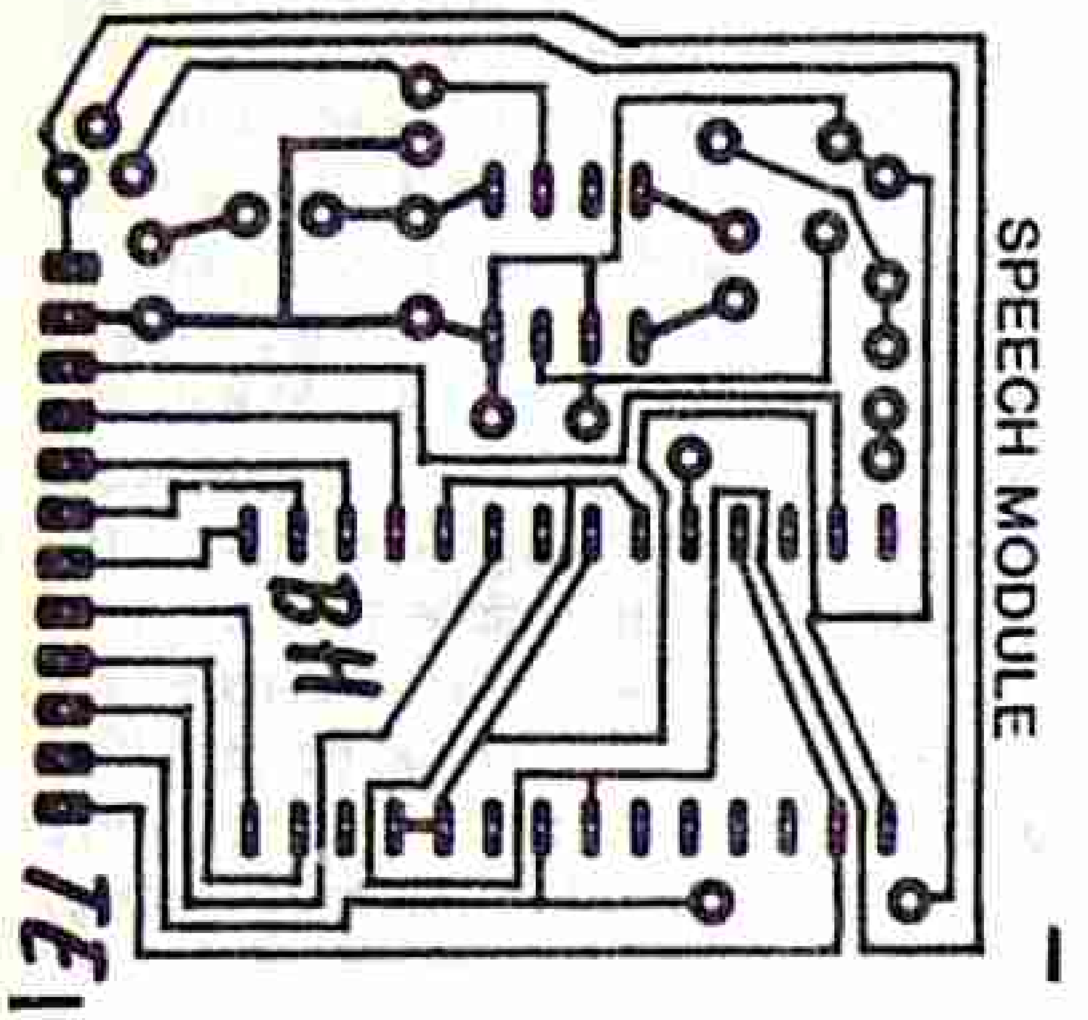
The five pauses are worthy of a separate mention. You must always pause after a word, to make the SPO stop talking. Use a PA1 or a PA2. Use PA3 or PA4 between sentances. Refer to the following table for when to use PA1, PA2, and PA3 DURING words.

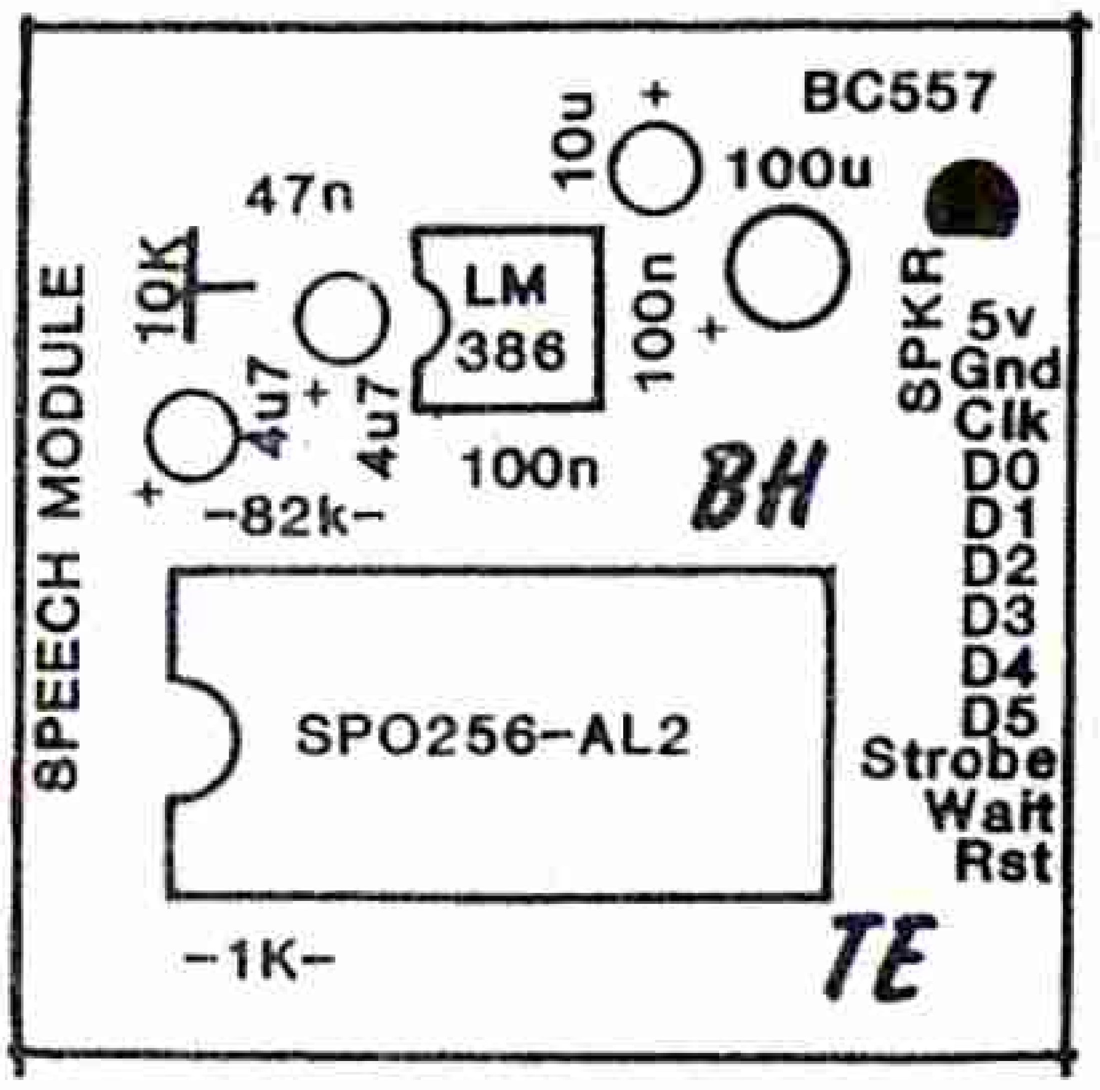
PA1 Before BB, DD, GG and JH.
PA2 Before some BB, DD, GG and JH.
PA3 before PP, TT, KK, and CH.

Begin by inserting the resistors. Solder them in and cut their leads short. Next insert the Capacitors, observing polarity with the Electrolytics.

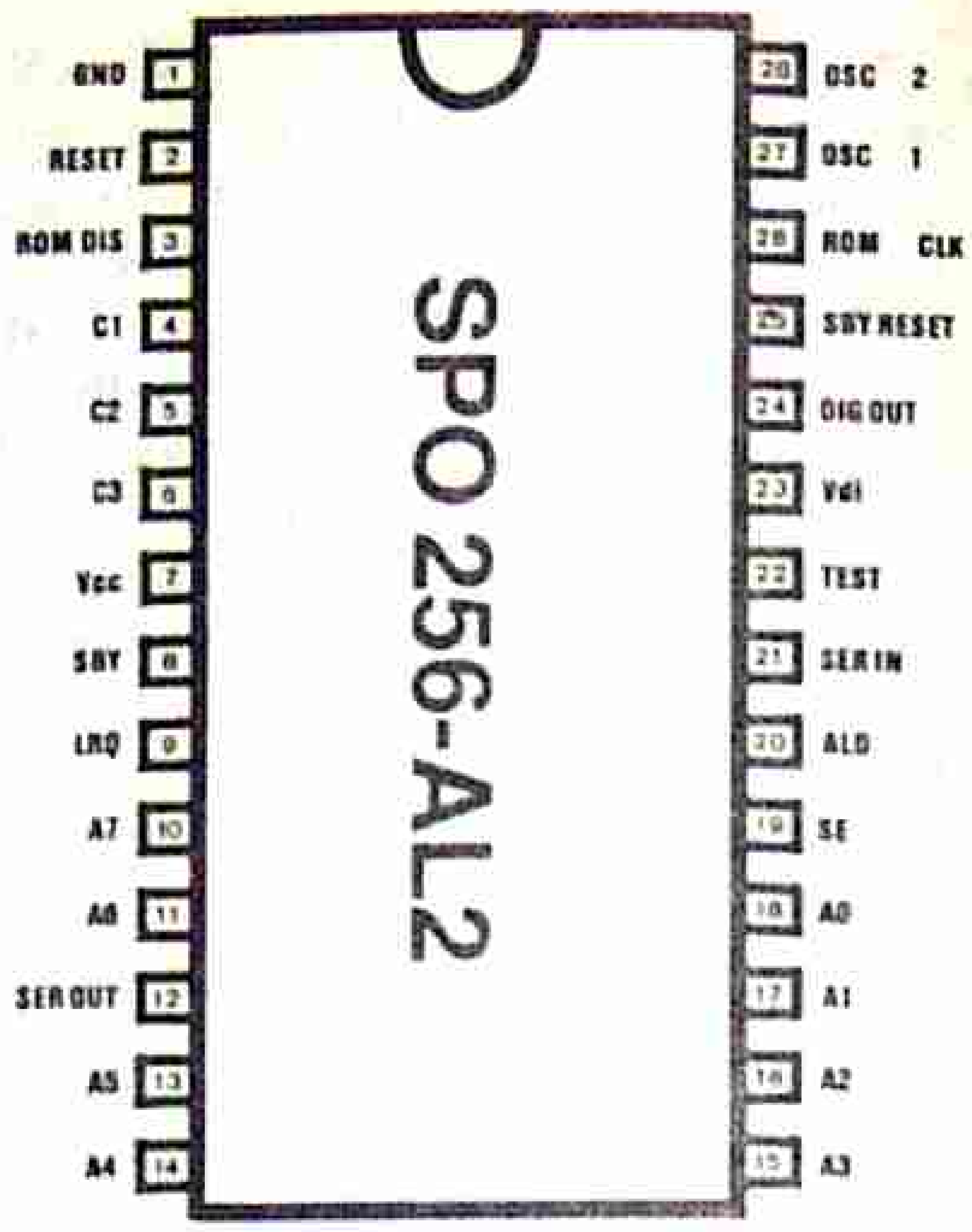
Insert and solder the trimpot, then finally the transistor. Turn the trimpot fully towards the SPO - this is full volume and should be set here until testing is complete.

Check to see that you have a BC557 and insert it according to the 'D' on the overlay. Lastly insert the two IC sockets and plug the chips in, being careful to orientate pin one with the mark on the PC and avoid touching the pins of the SPO256A.





The speech board is very simple. Don't forget you will need the crystal oscillator project to get the 3.58MHz clock line.



The pin out of the SPO-256-Al2 allophone chip.

Strip 6 wires from the ribbon cable, then connect the remaining 8 between the data lines and the DIP header. Connect power with the last two strands. Follow the diagram and you can't go wrong. Separate 4 of the remaining wires into individual lengths and solder into the 4 remaining holes on the module.

Attach a matrix pin connector to the other end of each wire for connection to the TEC. Heatshrink each connector with the tubing supplied. A note on heatshrinking: Don't skip this section because you think it's a waste of time or too hard to do. Heatshrinking the connectors strengthens them and the wire is

A repeating allophone is one which can be spoken twice and flow along. i.e. EY EY produces 'AY pause AY', while FF FF produces one long 'Ffff'. Only 10 of these 64 allophones are repeatable like this. They are: IH EH AE UH AO AX AA FF TH & SS. Use these allophones in preferance to long timed syllables like SH in SHirt, WE in tWEnty, or SH in leaSH.

CONSTRUCTION

Although a simple project, care should be taken to ensure that a good job is done, so do not rush. Lay all the parts out in front of you on a piece of paper or cardboard (Not the High - Low shagpile of the living room!) and check to see that you have been supplied with everything.

TEST PROGRAM

```
0900  21 10 09  LD HL,0910     HL = Points to start of table.
0903  7E        LD A,(HL)      Get next Allophone.
0904  FE FF     CP FF          End of table ?
0906  28 05     JR Z,090D      Yes, HALT.
0908  D3 05     OUT (05),A     Speak allophone.
090A  23        INC HL         Next allophone.
090B  18 F6     JR,0903        Say next ...
090D  76        HALT           EOT, stop until key pressed.
090E  18 F0     JR,0900        Key pressed, say again.

0910  0D 17 17 02 2A 0C 2C 04  Your allophones are entered
0918  04 2A 0F 10 00 31 16 0D  from 0910 onwards.
0920  33 04 04 FF              this says 'TALKING COMPUTER'

0910  1B 07 2D 35 00 36 07 2F  Here is another greeting
0918  04 06 00 1A 10 00 12 13  message.
0920  00 0D 13 03 13 03 37 13
0928  03 08 18 10 09 31 16 11  The TEC introduces itself
0930  33 04 04 04 38 20 00 30  here!
0938  0C 1D 37 09 13 32 04 FF
```

much less likely to break off. If you always melt the wire when shrinking over a candle, then try using the BARREL not the tip of your soldering iron. This gives you a better controlled heat source and a neat job can be done on those small connections.

The last two lengths of wire connect to the speaker. Wire these up and the board is complete. Now for connection to the TEC. You will need to have your crystal oscillator inserted. If you do not currently own a crystal oscillator, you must purchase one with a 3.58MHz crystal. If you have a different frequency crystal fitted, it must be around 3.2 - 4.0MHz otherwise the sound will be too high or low pitched. A 2MHz or 8MHz crystal will not suffice.

Insert a PC pin in port 5 pad, a second pin in the board for the WAIT line, and a third pin in the board for the RESET line. Most users will already have done so, but if not, see the wiring diagram for the three pin locations.

The other pin you will have to connect as best you can. To tap the 3.5MHz signal DO NOT connect to pin 6 of the Z80. This is because the crystal's frequency is divided by two before reaching the TEC board. Instead, solder a PC pin onto pin 8 of the 74LS04 on the crystal oscillator PC. This is the 3.5MHz clock output.

TESTING

Plug everything together and power up. If your TEC locks up or the unit makes strange sounds, remove power and go to the section on troubleshooting. Your TEC should start up as normal, with the unit deadly quiet. Enter the TEST PROGRAM and you should be greeted with a message. Listen carefully and let your hearing adjust to the metallic pitch. If all you can hear is junk, check your program, then if still no go, proceed to the troubleshooting section.

If the test program produces recognisable output, try the other examples and then try making up a few words of your own. You will soon find that you can say just about any word, once you get the right allophones.

There can be hours of fun even getting it to correctly pronounce your name. 'Paul' is easy enough, but what about 'Vouzopolous'?? or even common words like 'construction' and 'calculator'?? With such a versatile unit, the sky's the limit.

ALLOPHONE REFERENCE TABLE

NUMBER	ALLOPHONE	DURATION	SAMPLE
00	PA1	10 ms	PAUSE
01	PA2	30 ms	PAUSE
02	PA3	50 ms	PAUSE
03	PA4	100 ms	PAUSE
04	PA5	200 ms	PAUSE
05	OY	420 ms	Boy
06	AY	260 ms	Sky
07	EH*	70 ms	End
08	KK3	120 ms	Comb
09	PP	210 ms	Pow
0A	JH	140 ms	Dodge
0B	NN1	140 ms	Thin
0C	IH*	70 ms	Sit
0D	TT2	140 ms	To
0E	RR1	170 ms	Rural
0F	AX*	70 ms	Succeed
10	MM	180 ms	Milk
11	TT1	100 ms	Part
12	DH1	290 ms	They
13	IY	250 ms	See
14	EY	280 ms	Beige
15	DD1	70 ms	Could
16	UW1	100 ms	To
17	AO*	100 ms	Aught
18	AA*	100 ms	Hot
19	YY2	180 ms	Yes
1A	AE	120 ms	Hat
1B	HH1	130 ms	He
1C	BB1	80 ms	Business
1D	TH*	180 ms	Thin
1E	UH*	100 ms	Book
1F	UW2	260 ms	Food
20	AW	370 ms	Out
21	DD2	160 ms	Do
22	GG3	140 ms	Wig
23	VV	190 ms	Vest
24	GG1	80 ms	Got
25	SH	160 ms	Ship
26	ZH	190 ms	Azure
27	RR2	120 ms	Brain
28	FF*	150 ms	Food
29	KK2	190 ms	Sky
2A	KK1	160 ms	Can't
2B	ZZ	210 ms	Zoo
2C	NG	220 ms	Anchor
2D	LL	110 ms	Lake
2E	WW	180 ms	Wool
2F	XR	360 ms	Repair
30	WH	200 ms	Whig
31	YY1	130 ms	Yes
32	CH	190 ms	Church
33	ER1	160 ms	Fir
34	ER2	300 ms	Fir
35	OW	240 ms	Beau
36	DH2	240 ms	They
37	SS*	90 ms	Vest
38	NN2	190 ms	No
39	HH2	180 ms	Hoe
3A	OR	330 ms	Store
3B	AR	290 ms	Alarm
3C	YR	350 ms	Clear
3D	GG2	40 ms	Guest
3E	EL	190 ms	Saddle
3F	BB2	50 ms	Business

* = Repeating Allophone.

BASIC DICTIONARY

0	2B 3C 35
1	30 0F 0B
2	0D 1F
3	36 27 13
4	28 17 17 27
5	28 06 23
6	37 0C 29 37
7	37 37 07 07 23 0C 0B
8	14 11
9	38 06 0B
10	0D 07 07 0B
11	13 2D 07 23 34 0B
12	0D 2E 07 3E 01 23
13	1D 33 0D 13 0B
14	28 17 27 0D 13 0B
15	28 0C 28 0D 13 0B
16	37 0C 29 37 0D 13 0B
17	37 37 07 07 23 0C 0B 0D 13 0B
18	14 11 0D 13 0B
19	38 06 0B 0D 13 0B

A	14
Alarm	0F 2D 3B 10
Alex	1A 2D 07 29 37
Alexandra	1A 2D 07 29 37 1A 0B 15 27 0F
All	17 17 2D
Am	1A 10
Amateur	1A 10 1A 11 31 33
An	1A 0B
And	1A 0B 15
April	14 01 09 0E 0C 2D
Are	3B
At	1A 0D
August	17 1E 22 0F 37 11

B	3F 13
Baby	01 3F 14 01 3F 13
Bathe	3F 14 36
Bather	3F 14 36 33
Be	3F 13
Becky	3F 07 29 13
Bee	3F 13
Beer	3F 3C
Beth	01 3F 07 1D
Birthday	01 3F 33 1D 01 21 07 14
Bite	01 3F 06 03 11
Blank	01 3F 2D 1A 0B 02 29
Bob	01 3F 18 18 01 3F
Bread	1C 27 07 07 00 15

Brett	01 3F 27 07 03 11
Brother	01 3F 27 0F 1D 33
Buy	3F 18 06
By	3F 18 06
Byte	01 3F 06 03 11
Bytes	01 3F 06 03 11 2B
C	37 37 13
Calendar	2A 1A 1A 2D 07 0B 01 21 33
Calling	08 17 3E 2D 0C 2C
Cat	2A 1A 02 0D
Check	32 07 07 02 29
Checked	32 07 07 02 29 0D
Checker	32 07 07 02 2A 33
Checkers	32 07 07 02 2A 33 2B
Checking	32 07 07 02 2A 0C 2C
Checks	32 07 07 02 2A 37
Clock	2A 2D 18 18 02 29
Close	2A 2D 35 37 37
Clown	2A 2D 20 0B
Collide	08 0F 2D 06 36
Computer	2A 0F 10 09 31 16 11 33
Cookie	08 1E 2A 13
Correct	2A 34 07 07 01 29 01 11
Corrected	2A 34 07 07 01 29 01 0D 0C 01 15
Correcting	2A 34 07 07 01 29 01 0D 0C 2C
Correct	2A 34 07 07 01 29 01 11 37
Crane	08 27 14 0B
Crown	2A 27 20 0B
D	21 13
Data	21 18 18 01 11 33
Date	21 14 02 0D
Daughter	21 17 0D 33
Day	01 21 14
December	15 13 00 37 07 30 1C 33
Dennis	21 07 0B 0C 37
Disk	21 0C 37 37 29
Divided	21 0C 23 06 01 21 0C 01 15
Do	03 21 16 1F
Drive	21 27 06 36
Drives	21 27 06 36 2B
E	13
East	13 37 11
Eight	14 11
Eighteen	14 11 0D 13 0B
Eighty	14 0D 11 13
Eleven	13 2D 07 23 34 0B
Emergency	13 10 33 0A 07 0B 37 13
Engagement	07 07 00 0B 24 14 01 0A 10 07 07 0B 01 02 0D
Engages	07 07 00 0B 24 14 01 0A 0C 2B
Engaging	07 07 00 0B 24 14 01 0A 0C 2C
Enrage	07 0B 0E 14 01 0A
Enraged	07 0B 0E 14 01 0A 01 15
Enrages	07 0B 0E 14 01 0A 0C 2B
Enraging	07 0B 0E 14 01 0A 0C 2C
Error	07 07 27 00 33
Extent	07 2A 37 0D 07 07 0B 0D
Exterminate	07 29 37 0D 33 10 0C 00 14 0D
F	07 07 28 28
Father	28 3B 12 33
February	28 07 1C 00 19 1F 34 13
Fifteen	28 0C 28 0D 13 2B
Fifty	28 0C 28 0D 13
Fir	28 34
Five	28 06 23
Fool	28 1E 1E 2D
Force	28 3A 37 37
Four	28 17 17 27
Fourteen	28 17 27 0D 13 0B
Forty	28 17 27 0D 13
Freeze	28 28 0E 13 2B
Freezers	28 28 0E 13 2B 33 2B
Friday	28 27 06 01 21 14
From	28 27 18 10
Frozen	28 28 0E 35 2B 07 0B
G	0A 13
Glenn	01 22 2D 07 2C
H	14 01 02 32
Happy	39 1A 09 13
Has	1B 1B 1A 2B
Have	1B 1B 1A 23
Hello	1B 07 2D 35
Hertz	39 39 34 11 2B
How	39 20
Hundred	39 0F 0F 0B 01 21 27 0C 0C 00 15
I	06
Idiot	0C 01 21 0C 0C 0C 0F 11
In	0C 0B
Input	0C 0B 00 09 1E 11
Is	0C 2B
It	0C 03 11
J	0A 07 14
January	0A 1A 0B 1F 31 34 13
John	0A 18 0B
Julie	0A 31 3E 13
July	0A 1F 2D 06
June	2A 1F 0B
K	2A 07 14
Karen	2A 1A 27 00 07 0B
Kilo	2A 0C 2D 35
Know	38 35
Kristy	08 27 0C 37 11 13
L	07 07 3E
Live	2D 13 23
M	07 07 10
March	10 3B 32
Mark	10 3B 29
May	10 14
Memory	10 07 10 18 27 13
MHz	10 07 24 0F 39 39 34 11 2B
Minute	10 0C 0B 0C 02 0D
Minutes	10 0C 0B 0C 02 0D 2B
Modem	10 35 01 21 07 10
Monday	10 0F 0F 0B 01 21 14
Month	10 0F 0B 1D 1D
Mother	10 0F 36 33
My	10 06
N	07 07 0B
Name	38 14 10
Naughty	38 17 17 02 11 13
Nine	38 06 0B
Nineteen	38 06 0B 0D 13 0B
Ninety	38 06 0B 0D 13
No	38 35
November	38 35 00 23 07 10 1C 33
O	35
October	18 29 00 11 35 1C 33
Of	18 23
On	18 0B
One	30 0F 0B
Or	3A
Our	20 33
P	09 13
Past	09 3B 37 0D
Penelope	01 02 09 07 0B 07 2D 35 09 13
Penny	01 02 09 07 0B 13
Point	09 05 0B 11
Q	2A 31 1F
R	3B
RAM	27 01 1A 1A 10
Rebecca	0E 33 3F 07 02 08 3B
Ross	0E 18 37 37
S	07 07 37 37
Saturday	37 37 1A 02 0D 33 21 14
September	37 07 09 11 07 10 1C 33
Seven	37 37 07 07 23 0C 0B
Seventeen	37 37 07 07 23 0C 0B 0D 13 0B
Seventy	37 37 07 07 23 0C 0B

	0D 13
Sister	37 37 0C 37 0D 33
Six	37 0C 29 37
Sixteen	37 0C 29 37 0D 13 0B
Sixty	37 0C 29 37 0D 13
Son	37 0F 0B
Sound	37 20 0B 15
South	37 37 20 1D
Space	37 09 14 37
Speech	37 09 13 32
Statement	37 01 11 14 01 11 10 07 0B 11
Sunday	37 37 0F 0F 0B 02 21 14
T	0D 13
Talker	0D 17 17 01 29 33
Talking	0D 17 17 02 2A 0C 2C
Television	0D 07 2D 0C 23 0C 37 0C 18 0B
Ten	0D 07 07 0B
Test	0D 07 37 01 11
Testing	0D 07 37 01 11 0C 2C
The	12 13
There	36 07 2F
Thirteen	1D 33 0D 13 0B
Thirty	1D 33 0D 13
This	12 0C 37
Thousand	1D 20 2B 1A 0B 15
Three	36 27 13
Thursday	1D 34 2B 01 21 1A 14
Tim	0D 1C 10
Time	0D 06 10
To	0D 1F
Today	0D 1F 21 14
Tuesday	0D 31 2B 01 21 14
Twelve	0D 2E 07 3E 01 23
Twenty	0D 2E 07 0B 0D 13
Two	0D 1F
U	31 1F
V	23 13
Vision	23 0C 26 0C 0C 18 0B
W	21 0F 01 3F 3E 1F
Want	2E 18 0B 02 11
Wednesday	2E 07 07 0B 2B 01 21 14
What	30 18 02 11
Who	39 1E 1F
With	30 0C 1D
X	07 07 02 29 37 37
Y	2E 06
Year	19 3C
Yes	19 07 37 37
You	19 1F
Your	19 3A
Z	2B 07 02 15
Zero	26 13 27 35

IF IT DOESN'T WORK

If your speech unit does not work, DON'T PANIC. Firstly, check your wiring. Most errors are in wiring, causing the TEC to lock up. Look for obvious faults like shorts, dry joints, components of wrong value or orientation. Check that your chips are inserted correctly - pin one of each chip faces AWAY from the off-board wires.

If you bought your parts from all over the place, make sure you get a SPO256A-AL2 device. Other suffix numbers are not acceptable.

Check that the trimpot is turned all the way towards the SPO256A - full volume. You can temporarily short between the collector and the emitter of the BC557, to turn the amplifier on fully. This should produce a lot of hiss, and touching pin 3 of the LM386 should produce a buzzing sound.

Check that you have +5v on each chip, and that the SPO's reset pin (pins 2 and 25) are normally HIGH, and that they follow the reset pin of the Z80 (pin 26).

If all you get is garbage then you probably have the data lines wired around the wrong way. Check against the wiring diagram, and have a friend check it as well. Look for pins bent up under the SPO and not connecting with the IC socket. Check the program through and make sure that you are sending it the correct data.

If you are totally lost, give us a call. Sometimes we can solve a problem straight away, and most times within a few minutes. If all else fails, we offer a repair service. Costs are:

Basic repair $ 7.00
SPO256A replacement $15.00
Postage $ 3.00

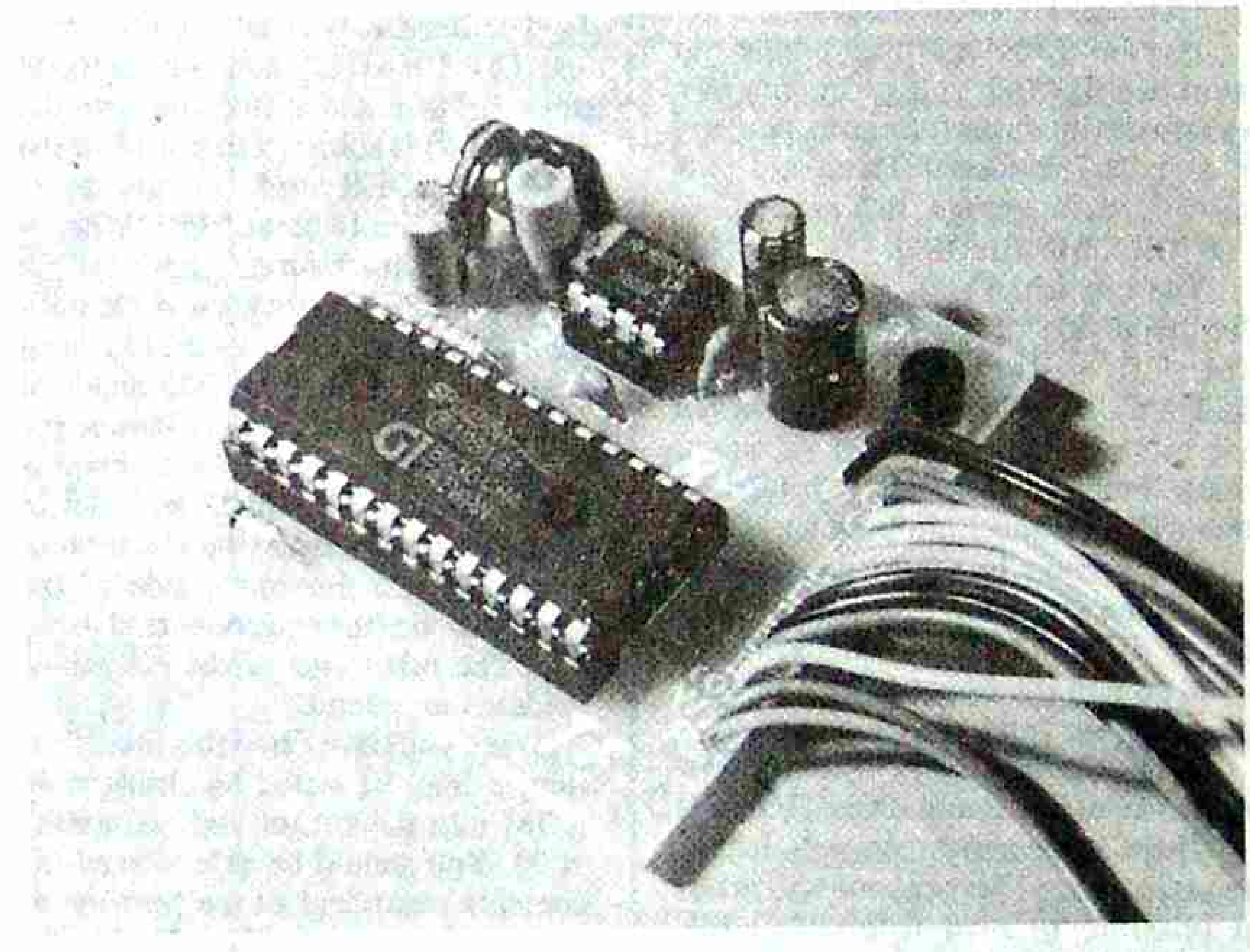
If your SPO256A-AL2 is damaged, you will be charged extra due to its high replacement cost.

MODIFICATIONS

If you don't intend to fit a crystal oscillator to your TEC, you can put a crystal on the speech board. Simply fit the crystal across pins 27 and 28 of the SPO256A. Then fit a 27p between pin 27 and ground, and a 27p between pin 28 and ground. This enables the SPO's internal oscillator. We did not include this on the basic board because we wanted to keep the price as low as possible, in order to counter balance the cost of the SPO256A. We reasoned that most people will change over to JMON, therefore purchasing a crystal oscillator anyway.

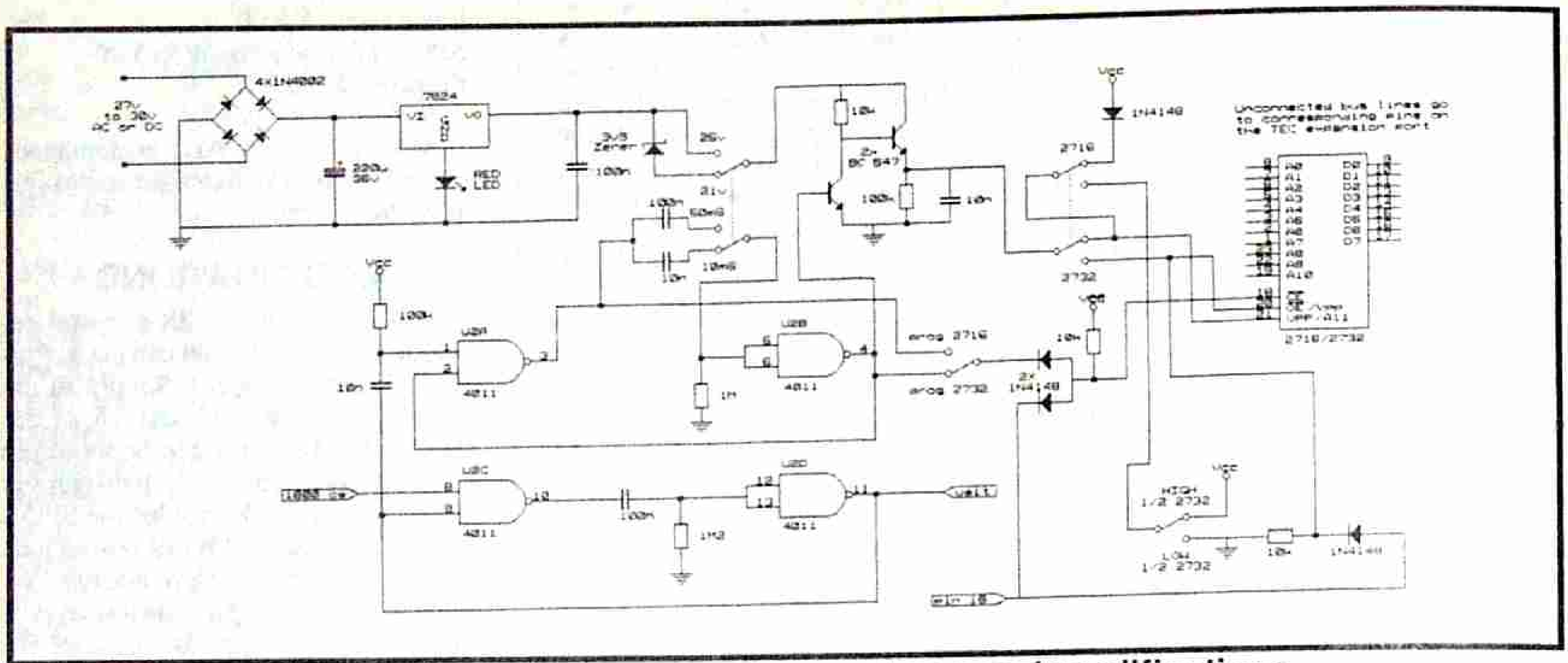
If you find that you are using long silent periods between words, you may find that you can hear an annoying click from the speaker as the LM386 gets switched. This is because the 10u capacitor is too low in value. Increase this capacitor to 22u or 47u and the problem should go away.

If you need to make the output louder, change the 4u7 between pins 1 and 8 of the LM386 to 10u. This increases the gain of the LM386 to 200.



EPROM PROGRAMMER REVISITED

Parts $2.30



Circuit diagram showing all corrections and modifications

CIRCUIT DIAGRAM CORRECTION

A mistake has been made with the circuit diagram on page 20 in issue 13.

The 100k resistor between pins 8 and 1 of the 4011 does not exist on the board and pin 8 is actually directly connected to the ROM SELECT LINE. It is not coupled through the 10n capacitor (via the 100k mentioned above) as shown.

CIRCUIT UP-GRADES

If your EPROM programmer is working ok and you're completely satisfied with its performance, perhaps it is best left alone. There are two modifications though, that are HIGHLY RECOMMENDED:

The first is the 100k resistor on the left-hand side of the EPROM socket (next to a diode) SHOULD BE REDUCED to 10k. This will allow for far more reliable readings (if yours doesn't read at all or very poorly, then this will almost certainly fix it).

The second is a 10n greencap is connected across the 100k resistor next to the EPROM socket on the right-hand side of the board (when looking at it from the top).

This 10n greencap is to prevent spikes from damaging the EPROM.

There are some other very handy mods to make. This next one will make it possible to read from 2732 (4k EPROMs) without having to slide the switch across. The BIG advantage of this is that it is possible for the software to read from the 2732 just after you have programmed each location. The software can then diagnose a failure and re-try or abort quickly. The software routine is provided below which will do this for either a 2716 or 2732.

Three additional parts are required for this mod. They are two IN 4148 diodes and a 10k resistor.

The first diode is soldered between the DIP-HEADER and the EPROM socket. The cathode (the end with the band on it) is soldered to pin 18 of the DIP-HEADER and the anode is soldered to pin 18 of the EPROM socket. Next, the track running between pin 18 of the EPROM socket and the middle of (program 2716 read 2732)/program 2732 switch is cut. The anode of the second diode is soldered to the pin 18 side of the cut and the cathode is soldered onto the middle terminal of the switch. One end of the 10k resistor is soldered to the anode side of the second diode (the end connected to pin 18). The other end of the resistor is soldered to ground.

Once you have fitted this modification, it may be tested by fitting a 4k ROM into the socket and addressing 1000. You should be able to read the contents regardless of the position of the read/program 2732 switch. The high/low switch is still used to select which half of the EPROM you wish to read and the read/program switch is used to select the type of EPROM you wish to program.

The next mod is a little more involved but is an important one if you wish to re-program some of the EPROMs supplied by TE.

The programming requirements of some types of more modern (but now obsolete) EPROMs are not compatible with the current set-up of the EPROM programmer. This mod allows the EPROM programmer to be used with a wider variety of EPROMs. The mod does this by switching the programming voltage from 25v to 21v and reduces the programming pulse from 50mS to 10mS.

The parts required for this mod are: one DPDT switch, one 10n greencap, one 100n greencap, a 3v9 zener diode

PARTS LIST
(For all mods)

2 - 10k

1 - 10n greencap
1 - 100n greencap

2 - 1N4148 diodes
1 - 3v9 Zener diode

1 - DPDT switch
1 - 10cm tinned copper wire
1 - 10cm hook-up wire

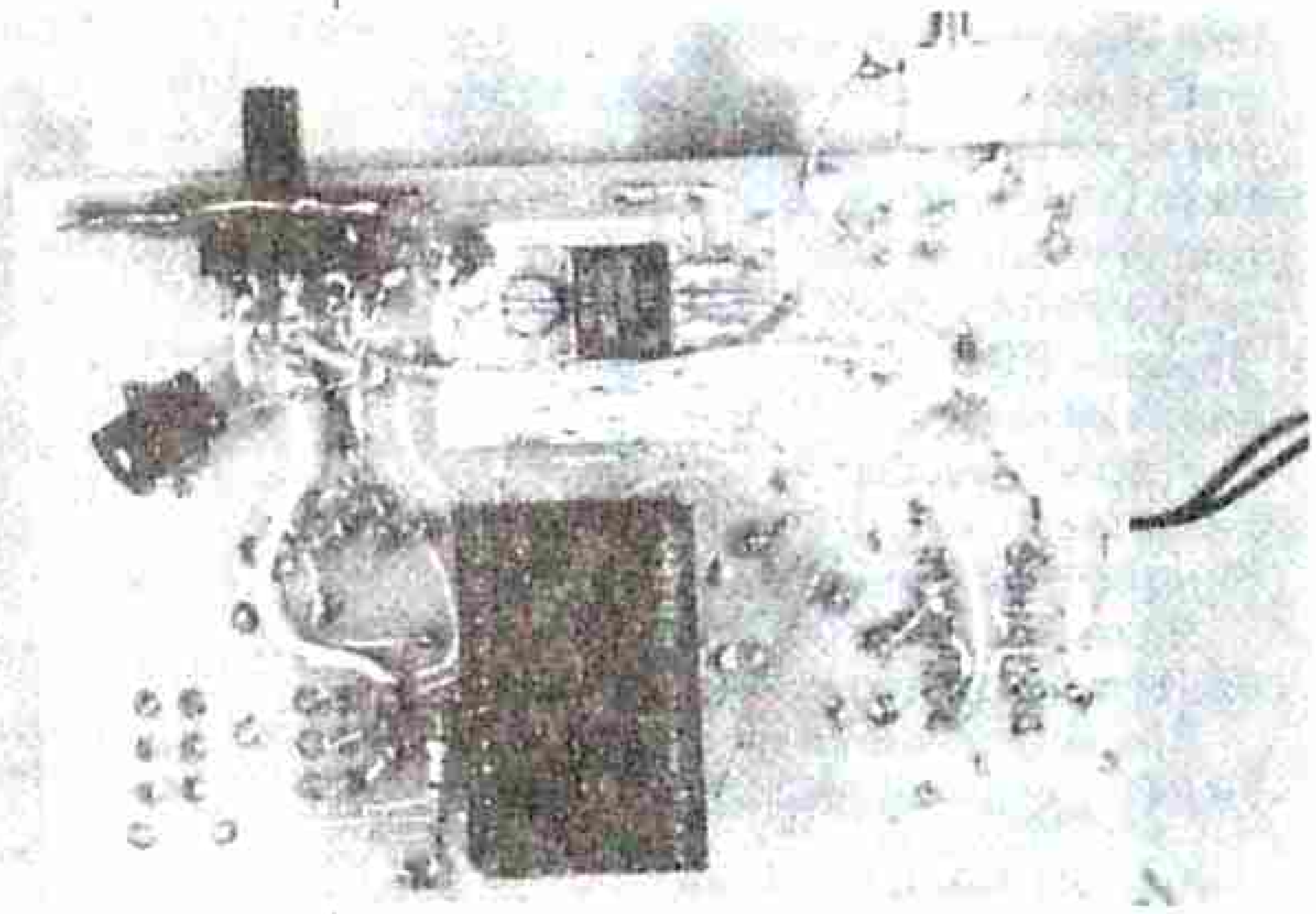
cont. P 45 . . . .

. . . . .from P. 32.

and some hook-up wire. To start, mount the switch on the bottom of the PCB by drilling two holes and wrapping tinned copper wire around the switch (see photo). Next cut the track between the output of the 24v regulator and the transistor switching block. The bottom middle terminal of the switch is connected to the transistor side of the cut. Connect the bottom right-hand side terminal to the regulator output and also solder the cathode end of the zener to this junction. The anode end of the zener is soldered to the bottom left-hand side of the switch. The zener, which is connected between the regulator and the high voltage switching section, drops the programming voltage by about 4v.

This completes the voltage switching section. Below is the programming pulse length mod.

The photograph on the right shows how the parts on our prototype are mounted. The description of the parts placement in the text, corresponds to this photo.



Remove the 100n greencap on the extreme left-hand side of the board (top view). Solder one end of the new 100n to the top right-hand side of the switch. Take the 10n cap and solder one side of this to the top left-hand side of the switch. The other ends of the caps are soldered together and a jumper is also soldered onto this junction. The jumper is then soldered to pin 3 of the 4011. Another jumper is soldered between pin 6 of the 4011 and the top middle terminal of the switch.

When the switch is in the right-hand position (top view), the EPROM programmer is set up for the modern 21v/10mS EPROMs.

One of these types of EPROM is being supplied by TE. It can be identified by the following markings:

TMS
2732A-25JL
LHE XXXX (DATE CODE)

To increase the reliability of the programmer, another mod is suggested. Follow the track from the ROM select line to where it joins the 10n cap. Cut both the tracks that join to the cap at this junction. Then run a link from the ROM select input pad to pin 8 of the 4011. Now run a jumper from the wait pin to the now isolated end of the greencap.

This mod slightly delays the programming pulse to the EPROM by triggering it from the wait line, not the input ROM select line.

The software for burning EPROMs provided in issue 13 is only very basic. There is one VERY IMPORTANT ADDITION to make to the issue 13 software. After you have loaded BC, DE and HL, as described in issue 13, add the following:

```
XOR A (AF)
LD I,A (ED 47)
JUMP 0700 (C3 00 07)
```

These instructions stop the noise on the expansion port which is a result of several TEC design oversights.

The following software is designed to be burnt into either a MON-1 or MON-2 EPROM at 0700.

The software is JUMP TO with the "from address" in HL, the "to address" in DE and the number of bytes in BC.

Before it attempts to burn into the EPROM, it checks that the area to be programmed contains only FF's. If not, the routine displays an "F", for FULL in the data display and halts. You may continue on and burn the EPROM by hitting "GO". Each location is checked after it is burnt and if not correct, it is reprogrammed several more times before being aborted. The routine then displays "E" for ERROR.

You must do the "read 2732" mod to program 2732 EPROMS.

An added feature to this software is that it flashes the address being programmed on the TEC display.

EPROM BURNING SOFTWARE

```
0700  AF          XOR A
0701  ED 47       LD I,A
0703  CD 90 07    CALL 0790
0706  7E          LD A,(HL)
0707  12          LD (DE),A
0708  D5          PUSH DE
0709  D9          EXX
070A  D1          POP DE
070B  CB 9A       RES 3,D
070D  D5          PUSH DE
070E  01 F0 0F    LD BC,0FF0
0711  C5          PUSH BC
0712  CD 5A 07    CALL 075A
0715  7B          LD A,E
0716  CD 5A 07    CALL 075A
0719  7A          LD A,D
071A  CD 5A 07    CALL 075A
071D  C1          POP BC
071E  01 10 00    LD BC,0010
0721  C5          PUSH BC
0722  CD 6E 07    CALL 076E
0725  C1          POP BC
0726  0B          DEC BC
0727  78          LD A,B
0728  B1          OR C
0729  20 F6       JR NZ,0721
072B  D1          POP DE
072C  1A          LD A,(DE)
072D  D9          EXX
072E  BE          CP (HL)
072F  20 08       JR NZ,0739
0731  23          INC HL
0732  13          INC DE
0733  0B          DEC BC
0734  78          LD A,B
0735  B1          OR C
0736  20 CE       JR NZ,0706
0738  C7          RST 00
0739  C5          PUSH BC
073A  01 05 00    LD BC,0005
073D  CB DA       SET 3,D
073F  7E          LD A,(HL)
0740  12          LD (DE),A
0741  10 FE       DJNZ,0741
0743  CB 9A       RES 3,D
0745  1A          LD A,(DE)
0746  BE          CP (HL)
0707  20 03       JR NZ,074C
0749  C1          POP BC
074A  18 E5       JR 0731
074C  0D          DEC C
074D  20 EE       JR NZ,073D
074F  C1          POP BC
0750  3E C7       LD A,C7
0752  D3 02       OUT (02),A
0754  3E 01       LD A,01
0756  D3 01       OUT (01),A
0758  76          HALT
0759  C7          RST 00
075A  F5          PUSH AF
075B  CD 63 07    CALL 0763
075E  F1          POP AF
075F  0F          RRCA
```

```
0760  0F        RRCA
0761  0F        RRCA
0762  0F        RRCA
0763  E6 0F     AND 0F
0765  21 B0 07  LD HL,07B0
0768  85        ADD A,L
0769  6F        LD L,A
076A  7E        LD A,(HL)
076B  02        LD (BC),A
076C  03        INC BC
076D  C9        RET
076E  21 F0 0F  LD HL,0FF0
0771  06 06     LD B,06
0773  0E 01     LD C,01
0775  7E        LD A,(HL)
0776  D3 02     OUT (02),A
0778  79        LD A,C
0779  D3 01     OUT (01),A
077B  0E 40     LD C,40
077D  0D        DEC C
077E  20 FD     JR NZ,077D
0780  07        RLCA
0781  4F        LD C,A
0782  AF        XOR A
0783  D3 01     OUT (01),A
0785  23        INC HL
0786  10 ED     DJNZ,0775
0788  C9        RET
0789  FF        RST 38
078A  FF        RST 38
078B  FF        RST 38
078C  FF        RST 38
078D  FF        RST 38
078E  FF        RST 38
078F  FF        RST 38
0790  D5        PUSH DE
0791  C5        PUSH BC
0792  CB 9A     RES 3,D
0794  1A        LD A,(DE)
0795  FE FF     CP FF
0797  20 09     JR NZ,07A2
0799  13        INC DE
079A  0B        DEC BC
079B  78        LD A,B
079C  B1        OR C
079D  20 F5     JR NZ,0794
079F  C1        POP BC
07A0  D1        POP DE
07A1  C9        RET
07A2  3E 47     LD A,47
07A4  D3 02     OUT (02),A
07A6  3E 01     LD A,01
07A8  D3 01     OUT (01),A
07AA  76        HALT
07AB  18 F2     JR 079F

07B0 EB 28 CD AD 2E A7 E7 29
     EF 2F 6F E6 C3 EC C7 47
```

PRINT-2 AND PRINT-3 SOFTWARE

With the changes to the keyboard handler routines in both MON-2 and JMON, an up-dated printer ROM has been produced.

The new software is burnt into the same ROM at higher locations. When MON-2 was released, an up-dated ROM called print-2 was included in the printer interface kits. This gave you the same routines with an altered keyboard section. It was also a little more fancy as it showed the start address on the LED display as you typed it in. Unfortunately, Print-2 did not include a "dump string at 0900" routine to replace the dump from 0800 which is now unusable as MON-2 uses 0800 for its variable storage.

With the advent of JMON, the same arrangement has been used. The JMON printer routines are located higher again, so in the one ROM you have the printer software for all three MONitors. The list routine for JMON is an improvement on both earlier software packages, as JMON's routine uses the perimeter handler to allow you to enter both a START and END address. Print-3 includes a "dump from 0900" routine which can be used with MON-2.

The ROM with the JMON routines in it is called PRINT-3 and is supplied with the printer interface as standard.

JMON's hex dump routine is at 1A20, the typing routine at 1AA0 and the "dump string at 0900" routine is at 1AC0.

Below is a dump of PRINT-3. Burn the additional section(s) in PRINT-1/2 ROM.

The graphic demonstration routines in PRINT-1 will work with all MONitors.

```
1800 3E 0D D3 06 3E 0A D3 06
     76 ED 57 17 17 17 17 57
1810 CD 5D 18 76 ED 57 82 57
     CD 61 18 76 ED 57 17 17
1820 17 17 5F CD 5D 18 76 ED
     57 83 5F CD 61 18 C3 49
1830 18 3E 0D D3 06 3E 0A D3
     06 7A CD 5D 18 7A CD 61
1840 18 7B CD 5D 18 7B CD 61
     18 06 08 3E 20 D3 06 1A
1850 CD 5D 18 1A CD 61 18 13
     10 F1 C3 31 18 1F 1F 1F
1860 1F 21 6C 18 E6 0F 85 6F
     7E D3 06 C9 30 31 32 33
1870 34 35 36 37 38 39 41 42
     43 44 45 46 FF FF FF FF
1880 21 00 08 7E FE FF 20 05
     3E 11 D3 06 C7 D3 06 23
1890 18 F1 FF FF FF FF FF FF
     FF FF FF FF FF FF FF FF
18A0 21 C3 18 7E FE FF 28 05
     D3 06 23 18 F6 06 0A 21
18B0 CF 18 7E FE FF 28 05 D3
     06 23 18 F6 10 F1 3E 11
18C0 D3 06 C7 0D 0A 0A 0A 0A
     0A 0A 12 43 30 0D FF 49
18D0 2C 44 33 32 30 2C 30 0D
     4D 31 32 30 2C 30 0D 44
18E0 38 30 2C 2D 31 36 30 0D
     4D 32 32 30 2C 2D 38 30
18F0 0D 44 31 36 30 2C 2D 38
     30 2C 31 34 30 2C 2D 31
1900 36 30 2C 32 30 30 2C 2D
     31 36 30 0D 4D 31 35 30
1910 2C 2D 31 32 30 0D 44 32
     30 30 2C 2D 31 32 30 0D
1920 4D 33 32 30 2C 2D 38 30
     0D 44 32 36 30 2C 2D 38
1930 30 2C 32 34 30 2C 2D 31
     36 30 2C 33 30 30 2C 2D
1940 31 36 30 0D 4D 33 36 30
     2C 2D 31 32 30 0D 44 34
1950 30 30 2C 2D 31 32 30 0D
     4D 34 36 30 2C 2D 38 30
1960 0D 44 34 34 30 2C 2D 31
     36 30 0D 4D 32 2C 2D 32
1970 0D 43 33 0D FF FF FF FF
     FF FF FF FF FF FF FF FF
1980 76 ED 57 E6 0F 17 17 17
     17 57 76 ED 57 E6 0F 82
1990 D3 06 18 EC FF FF FF FF
     FF FF FF FF FF FF FF FF
```

The next block is the PRINT-2 additions:

```
19A0 76 3A E0 08 E6 0F 17 17
     17 17 57 76 3A E0 08 E6
19B0 0F 82 D3 06 18 EA FF FF
     FF FF FF FF FF FF FF FF
19C0 3E 0D D3 06 3E 0A D3 06
     3E 29 21 D8 08 06 06 77
19D0 23 10 FC CD 00 1A 32 D8
     08 CD 00 1A 32 D9 08 CD
19E0 00 1A 32 DA 08 CD 00 1A
     32 DB 08 CD D8 01 CD 89
19F0 02 50 59 C3 31 18 FF FF
     FF FF FF FF FF FF FF FF
1A00 3E FF 32 E0 08 CD A0 02
     3A E0 08 FE FF 28 F6 E6
1A10 0F C6 FF CD 70 01 D6 01
     C9 FF FF FF FF FF FF FF
```

Below is PRINT-3 additions:

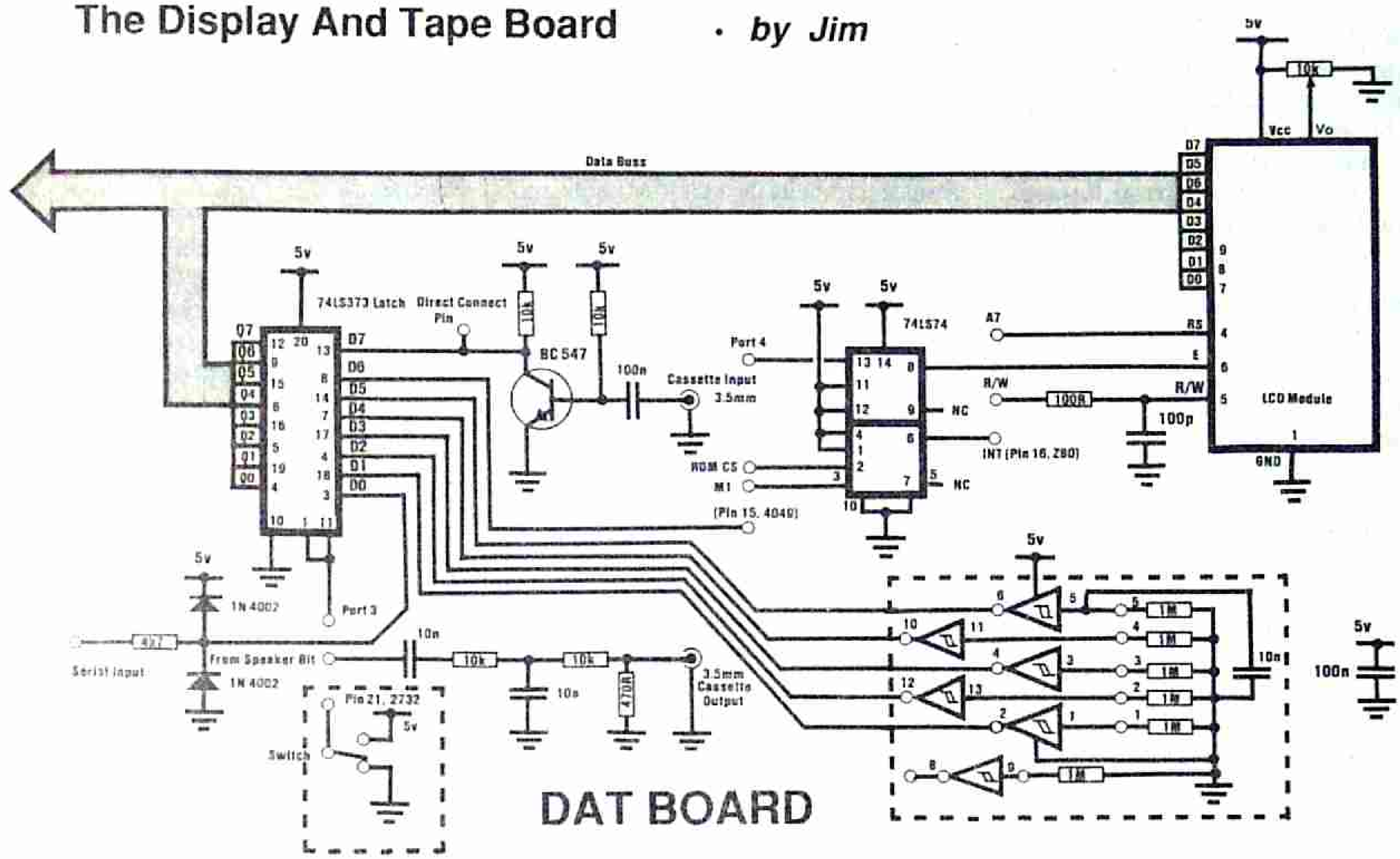
```
1A20 21 34 1A 11 80 09 01 0A
     00 ED B0 21 00 00 22 9C
1A30 08 C3 44 00 00 00 3E 1A
     99 08 00 01 50 1A 04 A7
1A40 04 C7 04 EB FF FF FF FF
     FF FF FF FF FF FF FF FF
1A50 3E 0D D3 06 2A 98 08 7C
     CD 82 1A 7D CD 82 1A 06
1A60 08 C5 3E 20 D3 06 7E CD
     82 1A 23 C1 10 F3 3E 0D
1A70 D3 06 3E 0A D3 06 ED 5B
     9A 08 E5 B7 ED 52 E1 38
1A80 D6 C9 F5 0F FF 0F 0F CD
     8B 1A F1 E6 0F C6 90 27
1A90 CE 40 27 D3 06 C9 FF FF
     FF FF FF FF FF FF FF FF
1AA0 CF E6 0F 07 07 07 07 57
     CF E6 0F 82 D3 06 18 F0
1AB0 D3 06 18 EC FF FF FF FF
     FF FF FF FF FF FF FF FF
1AC0 21 00 09 7E FE FF 20 05
     3E 11 D3 06 C7 D3 06 23
1AD0 18 F1 FF FF FF FF FF FF
     FF FF FF FF FF FF FF FF
```

TE

THE DAT BOARD

The Display And Tape Board

· *by Jim*



DAT BOARD

This board will change the way you program for ever. The DAT BOARD is perhaps the most vital addition to the TEC ever. Not just a part time "add on," but rather a permanent addition to your TEC.

Once you start using it, we think you'll agree.

The name "DAT" is an acronym for Display And Tape. While others brawl over "their" DAT, (have you seen one?), we have quietly slipped in the back door with our version.

The DAT BOARD provides these functions:

* 16x2 LCD display.
* Cassette tape I/O interface.
* Single stepper module.
* 5 Buffered and latched input bits.
* 1 Inverter for general use.
* Diode clipped input line. (For RS232 input)
* MON select switch.

PORT 3

Port 3 addresses an input latch. Below is a break-down of the bits on port 3.

BIT#
0 - Serial in
1 - input 1
2 - input 4
3 - input 2
4 - input 5
5 - input 3

The above are the inputs from the 74C14.

6 - key pressed signal.
7 - Tape input.

CONNECTION

Up until now, TEC add-on's have been connected via the expansion port. We wished to avoid this as there are too many devices cluttering up this area already. The search was on for a better place to put our new board. We decided upon the blank area left of the eprom, because it is common to all TEC's and has up until now not been used by anything else.

But there's nothing to connect to there! I hear you say. Well not quite, Simply solder a cut-up I.C. socket onto the links and you have an (almost) instant data buss socket. The DAT BOARD has a set of feed downs that push into the sockets and serve the dual purposes of connection and fixation.

PARTS LIST

1 - 100R
1 - 470R
4 - 10k
1 - 10k mini trimpot

1 - 100p ceramic
2 - 10n greencaps
3 - 100n greencaps

1 - BC 547
1 - 74LS373
1 - 74LS74

1 - 5mm LED (for trimpot handle)
2 - 3.5mm sockets
1 - 20 pin IC socket
2 - 14 pin sockets (one to cut-up)
1 - 20cm 12 way ribbon cable
1 - 50cm figure-8 shielded cable
1 - 1.2 metres hook-up wire
4 - 3.5mm mono plugs
1 - 100cm tinned copper wire
1 - Female matrix connector
3 - 32mm x 2.5mm bolts
9 - 2.5mm nuts
1 - 16 character x 2 line LCD*
1 - DAT PC Board

*Don't Forget: The LCD display can be bought separately.

The feed downs are simply lengths of stiff wire soldered to the underside of the P.C. that extend about 1 to 1.5 cm down to push into the I.C. sockets.

The fixing of the DAT BOARD is also aided by three "stand offs," in the form of three bolts with nuts to tighten against the board. These may extend through the TEC board if you want as there is no track work underneath.

CONSTRUCTING THE DAT BOARD

Originally, the kit of parts for the DAT BOARD was going to be supplied in two sections. We have changed our minds since, but have decided to present these construction notes unchanged. The first thing to do, is to fit ALL the links, regardless of what section you are constructing.

If you have already built the TAPE and keyboard section and/or are now constructing the LCD/SINGLE STEPPER interfaces then skip ahead to the respective notes. Once you have built the LCD section skip back to the notes on inserting the feed downs, stand-offs and control buss leads.

THE TAPE AND LATCH SECTION

Most the components for the TAPE SECTION are fitted on the bottom left corner of the board. The exceptions being a 100n greencap, that goes on the middle left of the board, the latch chip and its socket. Fit these in the order you prefer and then solder a short piece of tinned copper wire in the hole marked "SP."

This is where the female matrix connector will slide on. If you are wondering why we recommend a piece of tinned wire instead of a male matrix pin, the reason is that the force needed to push a female over a male matrix pin is far to great to be healthy for the TEC or DAT PCBs. (The keyboard is destructive enough). The tinned wire can be tinned again to give just the right fitting diameter, if required.

After fitting all the components, cut the length of hook-up wire into 4 equal sections. Strip and tin each end of all the lengths. Solder two pieces to the ground strip next to the tape in and tape out pads on the DAT BOARD. The other ends of these wires solder to the top tags of the 3.5mm sockets.

Solder the two remaining wires to the tape in and tape out pads. The other ends are soldered to the DIAGONALLY OPPOSITE tags on the 3.5mm sockets. Keep track of which socket the wires are joined to, and mark them accordingly.

Drill two holes large enough for the 3.5mm sockets in the back or side of the RETEX case and fit the sockets in place. Strip the ends of the shielded cable and twist the shield into one strand. Remove the covers of the 3.5mm plugs and slide them onto the figure 8 cables, so they are back to back. Solder the shields to the larger tags on the plugs. The middle conductor is soldered to the smaller tags. Do this for each of the four ends.

Solder a 5cm piece of hook-up wire on the 1K resistor which connects the output latch to the speaker transistor. The wire is soldered on the LATCH SIDE of the resistor. The other end of the wire is soldered to the female matrix connector. This matrix connector slides over the pin marked SP on the DAT BOARD.

Now you are ready to insert the feed downs.

INSERTING THE FEED DOWNS

The feed down are made of stiff tinned wire of about 2cm length. The easiest way to solder these is to solder a continuous length in each hole, and then trim it down afterwards. Do this for all the feed downs and try to get them straight as possible.

The feed downs plug in to a cut-up IC socket soldered across the links near the EPROM. The socket is soldered where the links form a straight line as they disappear into the TEC PCB. (See diagram). If you want, you may make the feed downs longer, remove the links, and permanently solder the DAT BOARD in place. Of course, you will need to put jumpers beneath the board to replace the missing links. This arrangement will provide a far more reliable circuit connection. Make sure you have finished the board COMPLETELY before you do this, as you will not be able to solder underneath the board afterwards.

CONNECTION OF THE CONTROL LINES

There are 10 control lines that are soldered to the bottom of the TEC board. A 20 cm 12 way ribbon cable is used to make all the connections. The ribbon cable is soldered to a row of pads on the DAT BOARD about 2.5cm below the top edge. The ribbon cable is soldered to the BOTTOM SIDE of the DAT BOARD and then drops down between the TEC board and the RETEX case (if you have one).

All the connections to the DAT BOARD are printed on the solder side of the board while the connections to the TEC are made as per the wiring diagram.

The two 3.5 mm sockets for the tape in/out are mounted in either the back or side of the RETEX CASE. If you do not have a case, then the sockets can be connected with short pieces of wire and left "floating." We do not recommend that you drill holes in either the TEC or DAT boards for the sockets. This is to save the expensive TEC board from the excessive force involved in plugging and unplugging the leads. The best idea is to hold the sockets when inserting the leads.

THE STAND-OFFs

In addition to the feed downs, three bolts act as stand-offs. The head of these bolts sits on the TEC board or, if you wish, you may drill into the board and feed the bolts up through the board. If you have the original TEC-1 board with the 8212 latch chips, the top bolt will not be able to be feed through the board as there is track work associated with the (now aborted) on-board tape interface and battery backed RAM.

If you have drilled the holes, then feed the bolts up from the bottom of the TEC and lock each in place with a nut. A second nut is screwed down to about 1cm off the TEC board on each bolt. This sets the height of the DAT BOARD. The DAT BOARD is then placed over the two bolts and a third nut is tightened onto the DAT BOARD.

If you to not wish to drill into your TEC, which is quite understandable, then place a nut on each bolt and wind it down to about 1cm from the head. Poke the bolts through the the hole in the DAT BOARD and tighten down the second nut.

Next, insert the board and note how high it is off the TEC. Ideally it should be 1.5 to 2cm off the board. Trim the feed downs until you are happy with the height. Adjust the stand-offs until they all sit neatly on the TEC board. Finally, a blob of blu-tack can be used to secure the top stand-off on to the board. This will help keep the DAT BOARD square on the TEC.

TESTING THE LATCH/TAPE INTERFACE

The latch is easily tested by running up JMON. If the keyboard works then the latch is obviously working. You can test each bit of the latch by taking the remaining inputs to ground. These pins are connected to pins 2,4,6,8 and 12 on the 74C14 socket and also pin 3 of the latch chip itself. Make sure that you don't have the 74C14 fitted as this may damage the chip.

The following program will echo the latch on the LED display:

```
0900 3E 3F D3 02 DB 03 E6 3F
0908 D3 01 C3 00 09
```

To test the tape, refer to the pages on using the tape system that show how to use and trouble shoot the tape interface.

THE SINGLE STEPPER/LCD INTERFACES

If you are constructing this section before the tape/latch section, you will need to make a modification to the TEC. The mod is to add a 4k7 resistor between pin 15 of the 4049 and pin 10 of the Z80. The purpose of this mod is to route the DATA AVAILABLE SIGNAL to the DATA BUSS. Without this, JMON is unable to read the keyboard. (This mod is described numerous times throughout this issue). The LCD interface consists of just four components. They are a D flip flop, a 100p cap, a 100R resistor and a 10k trimpot. The D flip flop, (that was spare) is configured to act as an INVERTER!! This design saved us from having to use another chip.

The single stepper interface simply uses one half of a dual D flip flop!

CONSTRUCTION NOTES

These 2 interfaces are simple to construct. Just take care with the orientation of the 74LS74 chip. If you have a spare LED on hand then you can solder it onto the trimpot to use as a knob (one is provided in the kit).

FITTING THE LCD

Place the LCD FACE DOWN on the work bench and feed a 5cm length of tinned copper wire into each hole on the LCD. Solder the wires in place and then, starting at one end, trim the wires to form a ramp. This helps you to insert the 14 wires one-at-a-time into the DAT BOARD. The DAT BOARD edge connector is placed at the top of the DAT BOARD and the LCD overhangs the board like a verandah.

Insert the LCD into the DAT BOARD as best you can. A second person with a pair of tweezers could help tremendously in getting each wire down its hole.

After you have fitted the wires into their holes, position the LCD to the height you want. This should be about 1cm to 1.5cm, and carefully solder it in place.

TESTING THE LCD

After you have finished construction and wired the DAT BOARD to the TEC as shown in the wiring diagram, you're ready to go. Fit the board in place and turn the 10k trimpot clockwise when looking at it from the left. Turn it as far as it goes, then turn it back just slightly. This sets the contrast level and if it is not approximately at the position described above, nothing will appear on the LCD.

If you have JMON then fit it into the EPROM socket and power up the TEC. All things being equal, the display will show the following:

```
0900>xx xx xx xx
Data  xx xx xx xx
```

If not, the most likely cause is that one of the data lines is not getting to the display. The easiest way to check this is to type in the following:

```
0900 3E 55 D3 04 C7
```

AFTER you have entered this, connect a jumper between port 4 and the wait line of the Z80. When you have done this, hit go.

The TEC should go "dead." Now, with a logic probe, test the edge connector of the LCD. Starting from the right, the logic levels should be: H, L, H, L, H, L, H AND L.

If not, then check all the connections and retry until right. If the connections are right, but there is nothing on the display, check the voltage on pin three of the LCD. This voltage should be in the range of 0.5v to 1v. Adjust the trimpot until you measure this voltage.

Still no luck? Turn off the TEC, hold reset down and turn the TEC back on while still holding down the reset. The top row of the LCD should be dark and the bottom line should be light. If not then there maybe no power getting to the LCD, the contrast voltage may be incorrect (but you have already checked this), or the display has been damaged, they are all tested before they leave TE).

If the top line is dark when power is applied but the display does not respond when reset is released, then put your logic probe on pin 6 of the LCD. Hold down the "+" key and watch the logic probe. Pin three should pulse HIGH each time the TEC beeps. If not then check that you have the wire going to port 4 in the correct place. Check the track work around the 74LS74 chip and the chip itself.

If pin 6 seems ok, then check that the 100p cap is fitted as this is VERY IMPORTANT. Pin 5, the r/w line, should always be pulsing. Check this with the logic probe.

The only other line left to test is the register select (RS). This line is address 7, and the easiest way to check this is with a continuity tester. If the LCD clears when power is applied, but nothing appears on the LCD, then it is odds-on that the cause is address 7 not being wired correctly.

TE REPAIR SERVICE

Still can't get it going? Check it all through again, keep in mind that the most likely cause is a mistake in your wiring. As a VERY last resort (after ringing us) send it in and we'll see what we can do.

Our repair fee is $9.00, plus $2.50 for post and handling. This includes replacement of all parts except the LCD (that was tested before leaving us). Before you send it in, remove the control buss wires (the ribbon cable) from the DAT BOARD. Pack it up securely and send it down. If you want the tape section tested leave the 3.5mm sockets connected.

TESTING THE SINGLE STEPPER

This is easy. With JMON fitted, enter this at 0900:

```
0900: 00 00 00 00 00 C3 00 09
```

Now, press shift 2. The single stepper will show 0900 PC. Press any data key and the single stepper will cycle automatically. The occasional clicking you (may) hear is a result of the interaction of the interrupt response cycle and the decoding of the 74LS138 decoder chip.

If the single stepper doesn't work, then check your wiring as it is doubtful that the 74LS74 chip is faulty.

WHAT THE LCD INTERFACE DOES

The LCD is designed to directly interface to microprocessors. Unfortunately there are two main types of microprocessor buss timing and the LCD is designed for the wrong type (as far as we are concerned). In order to get the LCD to interface to the Z80, a little bit of juggling with the timing is needed.

The first problem is the the LCD requires an active HIGH Enable signal. This has been achieved by inverting the PORT 4 I/O select line. This inverting is done by the spare D flip flop on the DAT BOARD. By looking at the TRUTH TABLE for the 74LS74, I found that it was possible to configure it as an inverter if I used the CLR pin as the DATA input!

To cut a long story short, the idea worked. Eureka!

The next problem is the LCD requires R/W to be stable on the falling edge of the E signal. If you look at the Z80 timing, you will see that the R/W line and the IORQ change state simultaneously. By the time that IORQ has gated port 4 and the port 4 signal has been inverted, the R/W line will actually change (slightly) before the E line on the LCD!

To overcome this problem, a simple RC network has been placed on the R/W line. This RC delay holds the R/W line stable while the E line goes low. The time we are talking about here is just a fraction of a microsecond, but that is all it takes for the chips in the LCD to accept or reject the in-coming signals.

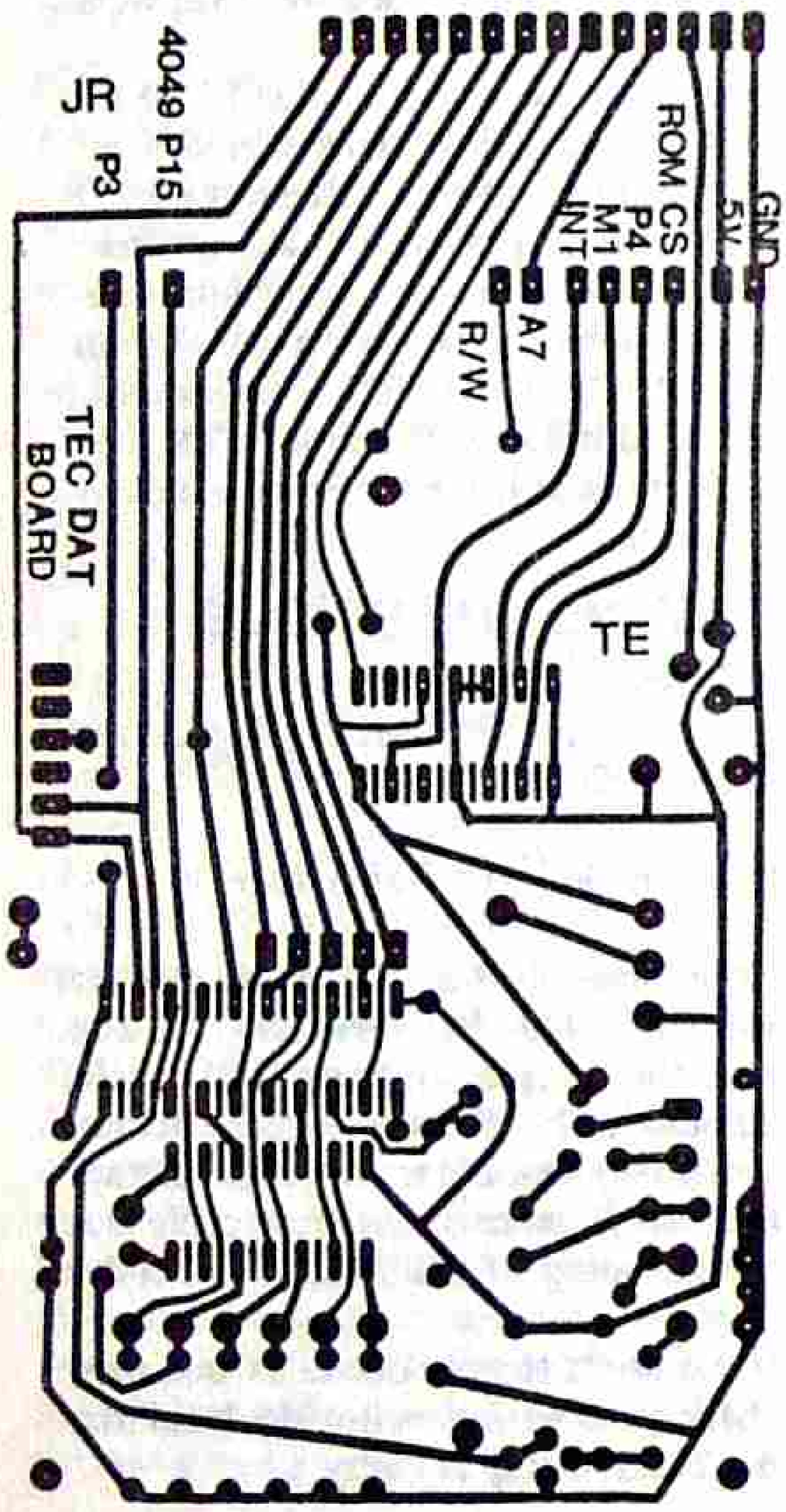
Another problem is that the LCD requires 2 ports to communicate with the Z80. It also wants to decode the second port itself. This is a common requirement of many peripheral devices, and the solution provided here is also useful for all these.

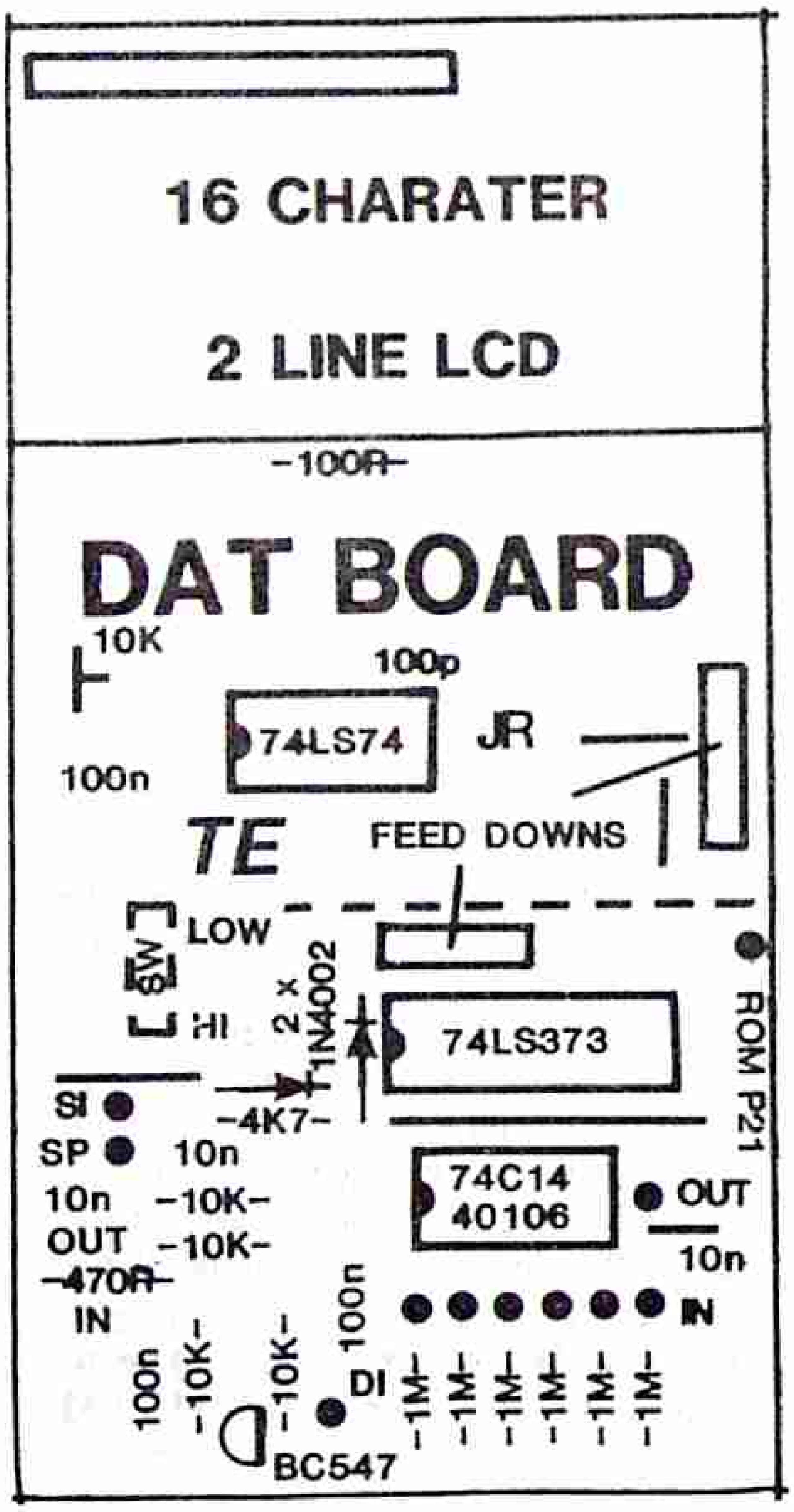
To give the LCD its second port, and let it decode it for itself, address line 7 has been presented to the LCD. This means that the second port is decoded (by the LCD) on port 84.

DISPLAY CONTRAST

The LCD requires an external voltage to set the contrast level.

The contrast of LCDs varies with temperature and viewing angle. To allow for this, the LCD has an external contrast control. The contrast is controlled by adjusting the voltage on this pin.





This is the function of the 10k trimpot, that is wired as a voltage divider.

OPTIONS

Several optional extras can be added to the DAT BOARD. Below is a description of each:

MON SELECT SWITCH

When you add the DAT BOARD, there may not be enough room between the board and the EPROM to fit your MON select switch. If this is the case, provision has been made to fit the switch to the DAT BOARD. Simply install the dotted link and move your switch to the dotted switch position on the DAT BOARD. Run a wire between the pin marked 'ROM P21' and pin 21 of the EPROM.

SERIAL INPUT

The SERIAL INPUT (SI)

This input is for a serial signal, or a RS232 level signal from a printer or RS232 device. This input clips the signal, which can be +/-15V to +/-25V, to safe logic levels.

This signal winds up as D0 on the 74LS373.

THE 74C14

This has been added to increase the versatility of the DAT BOARD. Some possibilities for it include a touch sensitive qwerty key pad, an external time reference, a thermistor controlled oscillator for temperature measurement or just buffered inputs. Nothing permanent has been planned for it, it is mainly for experimentation. We are open to your ideas!

THE DIRECT CONNECT PIN

This is located between the transistor and the 6 x 1M resistors. The purpose of this pin is to allow direct connection between two TECs. One TEC can down load to another through the tape software or a serial communication program. (I have a 9600 Baud routine that also talks to IBM's and compatibles).

THE UNUSED INVERTER

The input for the unused inverter is the right most matrix pin on the bottom right-hand side of the DAT BOARD. The output is the matrix pin directly above it.

HOW THE TAPE CIRCUIT WORKS

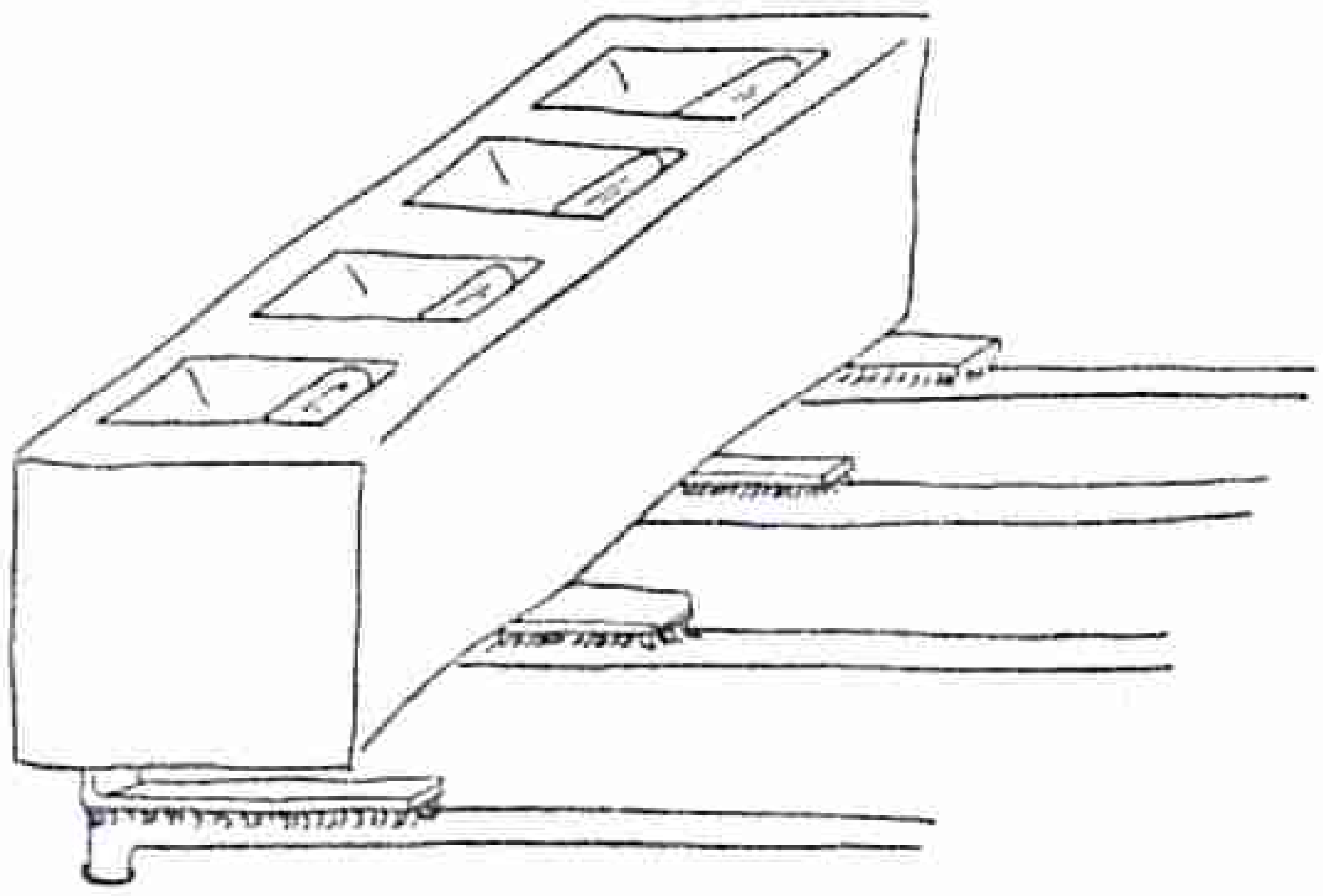
There's not much to describe about the tape circuit as all the hard work is done by software. The output section consists basically of an AC coupled LOW PASS filter with some attenuation on the end to prevent the digital level voltage from over driving the cassette players input.

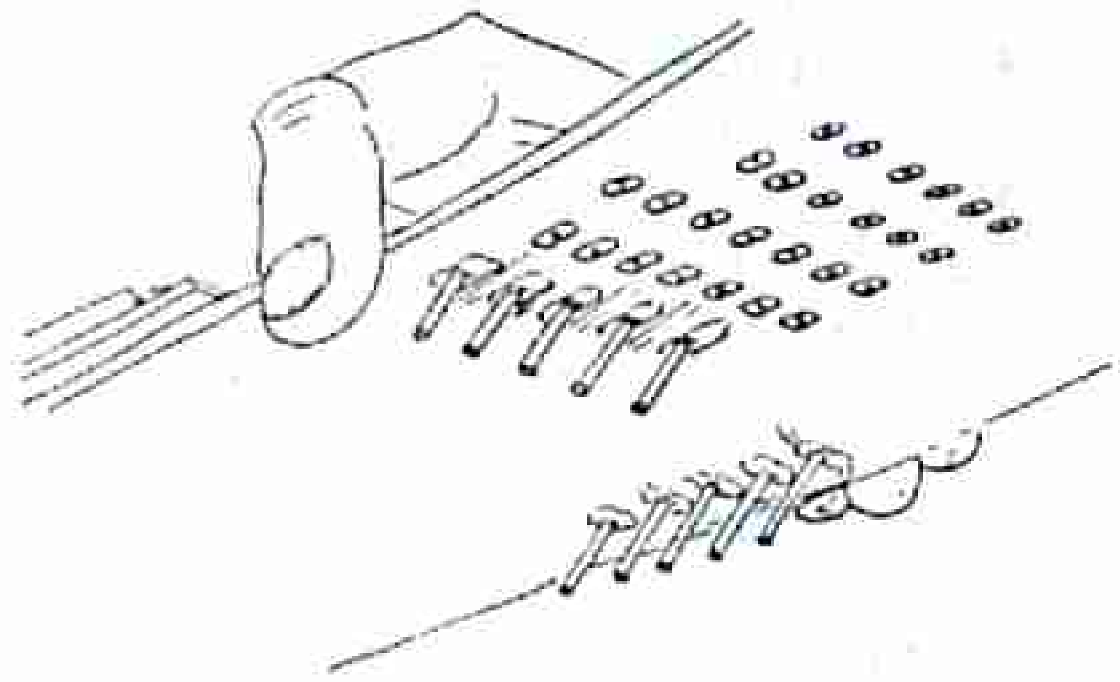
The input section is just a simple AC coupled common emitter transistor amplifier with the base heavily biased on. The bias on the transistor is important as this ensures that the software is able to read a steady logic 0 when no (AC) input is present.

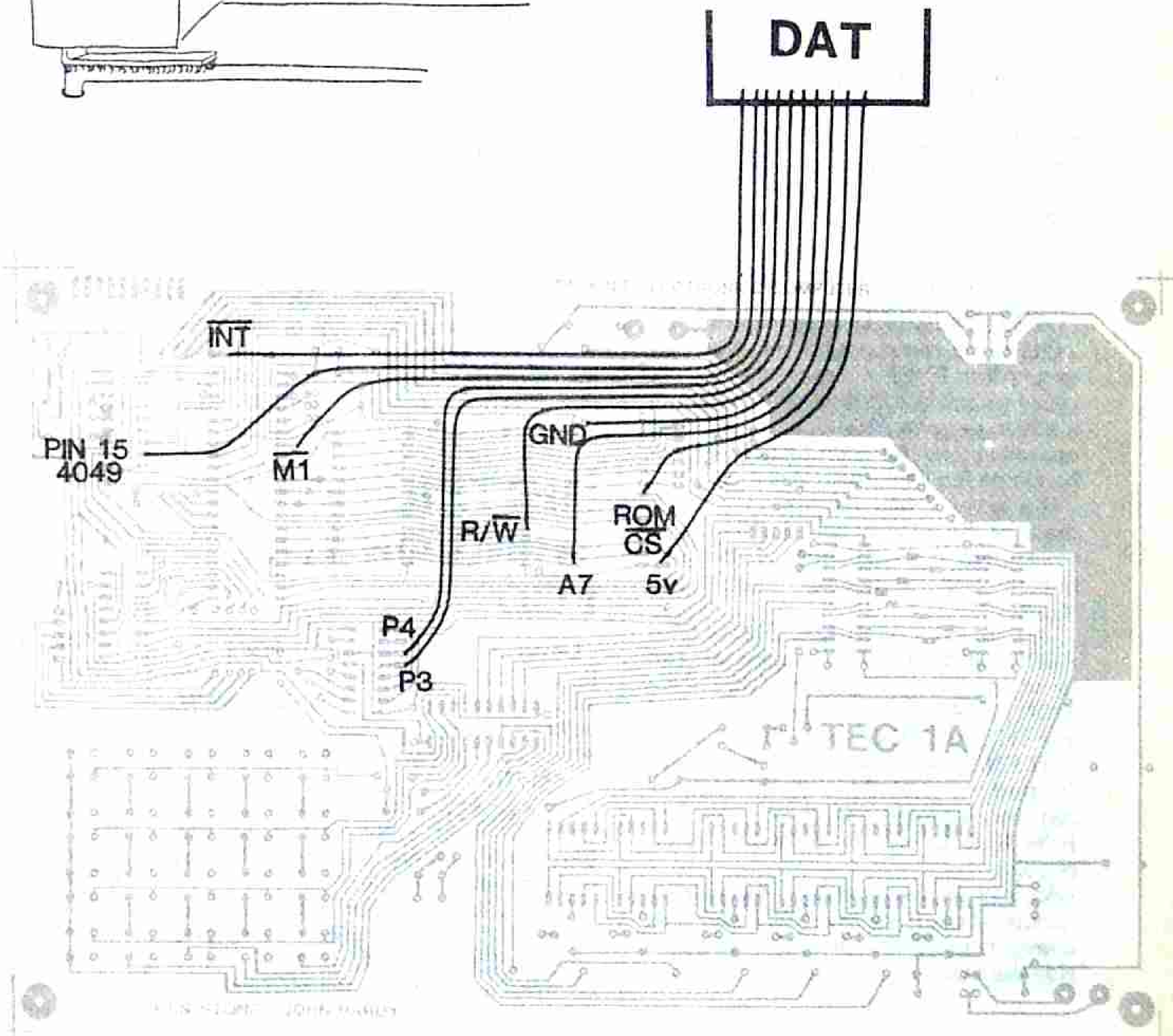
HOW THE SINGLE STEPPER INTERFACE WORKS

The single stepper INTERFACE works by interrupting the Z80 after each instruction. The interrupts are generated from a D flip flop on the DAT BOARD. Each time the Z80 fetches the first byte of an instruction a special signal called M1 is generated. This M1 is used to clock the ROM CS line into the D flip flop. The Q-bar output of the flip flop is connected to the INTerrupt pin. This means that an interrupt will be requested on every instruction fetch unless the instruction was fetched from the MONitor ROM.

It is important to prevent interrupts while executing in the MONitor ROM. If we don't, then an interrupt will occur just after it is re-enabled, at the end of the stepper routine. Immediately following the EI (enable interrupt), is a RETurn. If an interrupt occurs on this RETurn, then the stepper routine is re-invoked and each time this RETurn is reached, the program loops back to the stepper routine forever!! (If it wasn't for this problem we would not require any external hardware at all).







TOP RIGHT
Bottom side of the DAT board with the feed-downs fitted.

TOP LEFT
Diagram showing how the cut-up IC socket is mounted on the links.

LEFT
Wiring diagram showing where the "flying leads" from the underside of the DAT board are connected to the TEC.
Note that the diagram DOES NOT show the wires leaving the DAT board in the correct order, only the correct places on the TEC board. Use the labels on the underside of the DAT board for the correct DAT wiring positions.

THE LIQUID CRYSTAL DISPLAY

by Jim.

INSIDE THE DISPLAY

The display has three internal registers though which all communication is done. These are the registers:

THE DATA REGISTER

The data register is a read or write register to which all DISPLAY DATA (in ASCII format) and BIT MAPPED PROGRAMMABLE CHARACTERS are sent. This DATA register acts as a TEMPORARY BUFFER between the internal DISPLAY RAM or CHARACTER GENERATOR RAM (both described below) and the host computer (our TEC).

Characters may also be read from this register.

Internal operations transfer data between this register and the internal RAM (or between RAM and this register). This register is located on port 84H.

THE INSTRUCTION (or CONTROL) REGISTER

The instruction register receives all instruction bytes. ALL bytes sent to this register will be interpreted as CONTROL by the LCD. This is a WRITE ONLY register and is decoded on port 04.

THE ADDRESS COUNTER/BUSY FLAG

Bit 7 of this register is used as the busy flag. After EVERY operation it goes HIGH to indicate that the display is not ready to perform ANY type of additional operation yet. As soon as the display becomes "on line" again, it will go LOW.

The lower 7 bits are the current address of the internal cursor. All read or write operations occur between the data register and the address held in this register.

This register is READ ONLY. (The ADDRESS COUNTER is set or altered by instructions sent to the INSTRUCTION REGISTER and then transferred into THE ADDRESS REGISTER by an INTERNAL operation). This register is located on port 04 with the control register. Internal decoding gates the R/W line to select between each register.

As well as the registers, the display contains both RAM and ROM. Below is a description of the internal memory inside the LCD.

THE DISPLAY RAM

All the display information sent to the DATA REGISTER is transferred into the DISPLAY RAM by an internal operation. This RAM can hold 80 bytes of display information. While the LCD may only display 32 characters at a time, the extra bytes allow for the display to be shifted or can serve as general purpose storage RAM. An unusual feature of the display RAM is that the address from the last location on the top line (27H) to the address on the bottom line (40H) IS NOT CONSECUTIVE.

THE CHARACTER GENERATOR ROM

This ROM contains 192 different 5x7 dot matrix characters. These include full upper and lower case Alphabet characters, numbers, maths symbols, Greek and Japanese characters.

All of the most used characters are here. Any type of character we need that is not there, can be made up on the CHARACTER GENERATOR RAM.

THE CHARACTER GENERATOR RAM

The CHARACTER GENERATOR RAM allows us to define up to 8 different characters of our choice. The format of each is a 5x8 dot matrix with the cursor making up the 8th row. Any or all can be displayed together on different parts of the LCD and also may appear in several places at once. We can use this to make games characters.

GETTING SOMETHING ON THE LCD

Using the LCD is easy because it contains its own "intelligent" chips which do all the hard work for us. From JMON, putting anything on the LCD is VERY easy because the LCD has been set-up by JMON.

JMON sets the LCD to shift the cursor right after each entry. You cannot see the cursor as it has been turned off by the software in JMON.

To aid with the experiments below, put FF at 0821 (the LCD will stop changing after the first F) and AA at 08FF. These disable the LCD from the MONitor (the FF at 0821) and stop the MONitor rebooting its variables on a reset (the AA at 08FF). The MONitor will reset to 0A00 to remind you that the variables have not be re- booted on reset. (Unless a key was held down while reset was pushed, in which case you must again put FF at 0821 and AA at 08FF).

Ok, lets start by putting the letter L on the screen.

Firstly we must clear the screen and send the cursor home. This may be done by one instruction - 01. We output this to the control register on port 04. Before we can output to the LCD we must wait until it is ready. Because this is required to be done frequently, the RST 30 instruction has been used to do this for us. The RST 30 reads the LCD busy flag and loops until it goes LOW.

Ok lets type this in:

```
0A00  F7      RST 30
0A01  3E 01   LD A,01
0A03  D3 04   OUT (04),A
0A05  76      HALT
```

Reset, Go

The display will go blank and the (invisible) cursor will return to home (top left-hand corner). The 01 instruction sets all the display RAM locations to 20H (space). The 01 instruction doesn't affect any previous mode setting or display options (discussed below).

Now enter this with the RST over the HALT at 0A05:

```
0A05  F7      RST 30
0A06  3E 4C   LD A,4C "(L)"
0A08  D3 84   OUT (84),A
0A0A  76      HALT
```

Reset, Go

The letter L appears in the top left corner.

Ok, now as before, put this in with the RST over the HALT:

```
0A0A  F7      RST 30
0A0B  3E 43   LD A,43 "(C)"
0A0D  D3 84   OUT (84),A
0A0F  F7      RST 30
0A10  3E 44   LD A,44 "(D)"
0A12  D3 84   OUT (84),A
0A14  76      HALT
```

Reset, Go

The above section outputs two more bytes to the DATA REGISTER.

Until now we have just been using a simple method to output data. This has shown us the basic way to talk to the LCD. Now that we have come this far and learned the basics, we'll advance to something more useful.

The code below will output a word onto the bottom line of the LCD. The display DATA will be held in a table at 0B00.

```
0A14  F7        RST 30
0A15  3E C0     LD A,C0
0A17  D3 04     OUT (04),A
0A19  01 84 06  LD BC,0684
0A1C  21 00 0B  LD HL,0B00
0A1F  F7        RST 30
0A20  ED A3     OUTI
0A22  20 FB     JRNZ 0A1F
0A24  76        HALT

0B00 4D 41 53 54 45 52
```

To set the cursor to the bottom line we output 80 to the instruction register (bit 7 sets the cursor address entry) + 40 (which is the actual address of bottom left display) = C0.

The OUTI instruction is new to our repertoire. It's operation is to output the byte addressed by HL to the port addressed by C. HL is then incremented and B is decremented. If B becomes ZERO the ZERO FLAG is set and the operation is complete. This instruction can output up to 256 bytes at a time.

Because we need to check the busy flag we loop back to the RST 30 until all the bytes have been done. If we didn't need to check the busy flag we could have used the OTIR instruction which automatically repeats itself until B=0.

All the above is done with the cursor switched off. For the next section we want to have the cursor on. To switch on the cursor output 0E to the instruction register on port 04.

```
0A00  F7        RST 30
0A01  3E 0E     LD A,0E
0A03  D3 04     OUT (04),A
0A05  76        HALT
0A06  C7        RST 00
```

Now let's see what does what on the display.

Using the above routine, output the bytes below one at a time, to port 04 and HALT between each. (leave what's on the display there).

Check the function of each on the table of controls.

18 1C 1C 1C 02 14
14 10 0C 0F 08 0C

Good luck!!

SETTING THE ENTRY MODE

The display may be configured to perform several different functions UPON EACH DATA BYTE ENTRY. They are:

1 INCREMENT CURSOR ADDRESS after storing inputted data byte (06H). This is our normal mode.

2 DECREMENT CURSOR ADDRESS after storing input (04).

3 SHIFT THE DISPLAY RIGHT after entry (05).

4 SHIFT THE DISPLAY LEFT after entry (07).

Each mode is selected by outputting the byte shown to port 04.

Once the entry mode is set it IS ONLY CHANGED BY ANOTHER ENTRY MODE SET COMMAND. None of the other control bytes will alter the entry mode.

The shift on entry feature (05,07) has been found to be difficult to use and even appears to contain design bugs.

You may experiment with it but we won't be using it in these notes.

The CURSOR DECREMENT may come in handy sometimes but it's more likely to be useful to processors which move blocks of data around in a more limited way to the Z80.

RUNNING WORDS ON THE LCD

Running words along the LCD is also simple because the LCD'S intelligent chips do most the work for us again. Our job is to enter the words we want to scroll (up to 16 characters per line for this routine) and send shift commands each time we want a shift.

The routine below is entered in 3 sections. Each section is a logical progression and increases the programs abilities. You can look at the instructions in each section and compare it to what the section does. This way you can learn how to put blocks together to use the display any way you want. Before entering the code below put FF at 0821 and AA at 08FF as described before.

Enter this and INCLUDE the NOPS and the table at 0B00 then run it:

```
0A00  3E 01     LD A,01
0A02  D3 04     OUT (04),A
0A04  F7        RST 30
0A05  3E 06     LD A,06
0A07  D3 04     OUT (04),A
0A09  F7        RST 30
0A0A  3E 0C     LD A,0C
0A0C  D3 04     OUT (04),A
0A0E  F7        RST 30
0A0F  00        NOP
0A10  00        NOP
0A11  00        NOP
0A12  00        NOP
0A13  00        NOP
0A14  01 84 10  LD BC,1084
0A17  21 00 0B  LD HL,0B00
0A1A  F7        RST 30
0A1C  ED A3     OUTI
0A1D  20 FB     JRNZ 0A1A
0A1F  F7        RST 30
0A20  3E C0     LD A,C0
0A22  D3 04     OUT (04),A
0A24  F7        RST 30
0A25  21 30 0B  LD HL,0B30
0A28  06 10     LD B,10
0A2A  F7        RST 30
0A2B  ED A3     OUTI
0A2D  20 FB     JRNZ 0A2A
0A2F  76        HALT

0B00: 54 41 4C 4B 49 4E 47 20
0B08: 20 20 20 20 20 20 20 20
      (TALKING)
0B30: 45 4C 45 43 54 52 4F 4E
0B38: 49 43 53 20 20 20 20 20
      (ELECTRONICS)
```

This will put "TALKING" on the top line and "ELECTRONICS" on the bottom line of the LCD and stop. Study the above section and see if you can work out the role of each instruction.

Now we'll add the shift section. Enter this with the first "NOP" over the last "HALT" and run it:

```
0A2F  00        NOP
0A30  00        NOP
0A31  3E 18     LD A,18
0A33  D3 04     OUT (04),A
0A35  01 00 60  LD BC,6000
0A38  0B        DEC BC
0A39  78        LD A,B
0A3A  B1        OR C
0A3B  20 FB     JRNZ 0A38
0A3D  18 F2     JR 0A31
```

The above code loads the shift instruction (18H) into the accumulator and outputs it to the control register on port 04.

As you can see it shifts the display, but this method is not very good if we want to shift only a few characters as we must wait for them to be shifted through the entire display RAM before they re-appear. To overcome this we can count the number of shifts and reset the display with a 02 command, as soon as all the letters have been shifted outside the display. The 02 instruction resets the display from shift WITHOUT CHANGING the contents of the DISPLAY RAM, CHARACTER GENERATOR RAM, or the CONTROL MODE. Because we would like the words to shift across the entire display and re-appear as soon as they have all gone, we must load the words just outside the screen to the right. The following additions make the words start shifting into the display from right-to-left.

Ok, Now enter the following, AT THE ADDRESSES SHOWN:

```
0A0F  3E 90     LD A,90
0A11  D3 04     OUT (04),A
0A13  F7        RST 30
--------------------
0A22  3E D0     LD A,D0
```

The above instructions set the DISPLAY RAM ADDRESSES to the RAM locations just right of the screen. The address of the top line is 90 and the address of the bottom line is D0. (Actually these are the addresses + 80H, the SET ADDRESS instruction).

```
0A2F  16 1B        LD D, 1B
```

(The D register is our shift counter).

```
0A3D  00      NOP
0A3E  00      NOP
0A3F  15      DEC D
0A40  20 EF   JRNZ 0A31
0A42  3E 02   LD A,02
0A44  D3 04   OUT (04),A
0A46  F7      RST 30
0A47  18 E6   JR 0A2F
```

The last group makes up the shift counter and resets the display when the counter reaches Zero. When the 02 command is received by the LCD the display is returned to its NORMAL position. This means that the inputted data is returned to WHERE IT WAS ENTERED (just right of the screen). Now, when the next shift command is received, the letters start to shift left back on to the screen.

QUESTION:
Why don't we need to wait for the BUSY flag to go low after the shift instruction?

If you wish to change the number of characters to be shifted, you may do so by putting your new characters at 0B00 for the top line +and at 0B30 for the bottom line. Unused locations should have 20 (space) inserted until 16 locations are filled. (From 0B00 to 0B10 and from 0B30 to 0B40). The value of the loop counter loaded into D at 0A2F should also be changed. The value of the loop counter is best set to 10H + the number of letters occurring in the longest line.
e.g. For the the example above:
ELECTRONICS = 11 (0BH) Letters.

So add 0BH + 10H = 1BH.
So 1BH is loaded into D at 0A2F.
To understand the above formula better, try 1C and 1A and see the result.

FINAL NOTES
The slow response of the LCD detracts from the effectiveness of the shifting a little but by experimenting with the delay at 0A35 you should be able to get a good compromise between speed and display clarity.

The above shifting method is just one of dozens of ways we could have used. A more complex program could shift information across and out one end and load new information in the other to create a running information display.

Use the blocks in this program and the others to make up your own display routines. If you come up with something interesting, write in. We would love to see what you've come up with.

DESIGNING YOUR OWN CHARACTERS

You can have up to eight different characters stored in a character-generator RAM. Each character is displayed on the screen when it is addressed in the display RAM. The addresses are between 0-7. The user-defined characters are made up of an 8x8 matrix (only 5 columns are displayed, the cursor makes up the 8th row.)

To set up a character, 8 bytes are outputted to the character- generator RAM. The first byte makes up the top row (only the 5 lower bits are displayed). The second byte makes up the second row etc.

Before sending the 8x8 character (actually a 5x8 character), the entry mode must be set (if not already) to address-increment with no display shift (06) and a set character RAM address operation must be done.

The control byte for this is 40 + the address of the first byte of each character-matrix. E.g: 40, 48, 50 for characters 1, 2, 3 etc.

Once a character is set up, it is displayed by placing its address in the DISPLAY DATA RAM. Before doing this the DISPLAY RAM must be selected via 80 + address.

OK, let's put our own character on the LCD.

```
0A00  F7      RST 30
0A01  3E 01   LD A,01
0A03  D3 04   OUT (04),A
0A05  F7      RST 30
0A06  3E 40   LD A,40
0A08  D3 04   OUT (04),A
0A0A  01 84 08  LD BC,0884
0A0D  21 00 0B  LD HL,0B00
0A10  F7      RST 30
0A11  ED A3   OUTI
0A13  20 FB   JRNZ 0A10
0A15  F7      RST 30
0A16  3E 80   LD A,80
0A18  D3 04   OUT (04),A
0A1A  F7      RST 30
0A1B  3E 00   LD A,00
0A1D  D3 84   OUT (84),A
0A1F  76      HALT
```

0B00:

11, 0A, 04, 11, 0A, 04, 11, 0A

Experiment with the values in the table and see how it all goes together. By increasing the value loaded into B, to 10(hex) (at 0A0C) a second character may be programmed at the same time. The table for the second character will start at 0B08. This will be displayed when a 01 is written into the DATA DISPLAY REGISTER. Experiment and see if you can get 8 characters appearing in several places at once on the display.

MYSTERY EFFECT

The routine below produces a very interesting effect. It uses the PROGRAMMABLE CHARACTER GENERATOR to produce 8 different characters some of which are displayed several times. We won't tell you the effect, we'll let you type it in and see for yourself. You won't be disappointed!

The program consolidates much of what we have learned about "driving" the LCD. If you experiment further and add a shift to it then it will be a complete revision of what we have covered in these pages.

Now that you know how to use the LCD, start writing some programs that use it. If you come up with something interesting don't hesitate to send it in to TE. We would be very interested in some simple animation or an adventure game or anything that others would be interested in seeing. Go to it!

```
0A00  F7        RST 30
0A01  3E 01     LD A,01
0A03  D3 04     OUT (04),A
0A05  F7        RST 30
0A07  3E 06     LD A,06
0A08  D3 04     OUT 04
0A0A  21 00 0B  LD HL,0B00
0A0D  01 84 10  LD BC,1084
0A10  F7        RST 30
0A11  ED A3     OUTI
0A13  20 FB     JRNZ 0A10
0A15  F7        RST 30
0A16  3E 40     LD A,40
0A18  D3 04     OUT (04),A
0A1A  21 20 0B  LD HL,0B20
0A1D  06 40     LD B,40
0A1F  F7        RST 30
0A20  ED A3     OUTI
0A22  20 FB     JRNZ 0A1F
0A24  F7        RST 30
0A25  3E C0     LD A,C0
0A27  D3 04     OUT (04),A
0A29  21 10 0B  LD HL,0B10
0A2C  06 10     LD B,10
0A3E  F7        RST 30
0A3F  ED A3     OUTI
0A31  20 FB     JRNZ 0A3E
0A33  76        HALT
```

```
0B00: 20 4D 49 52 52 4F 52 20
0B08: 49 4D 41 47 45 21 20 20
0B10: 20 00 01 02 02 03 02 20
0B18: 01 00 04 05 06 07 20 20
0B20: 00 11 11 11 15 15 1B 11
0B28: 00 0E 04 04 04 04 04 0E
0B30: 00 11 12 14 1E 11 11 1E
0B38: 00 0E 11 11 11 11 11 0E
0B40: 00 11 11 1F 11 11 11 0E
0B48: 00 0F 11 11 17 10 11 0E
0B50: 00 1F 10 10 1E 10 10 1F
0B58: 00 04 00 00 04 04 04 04
```

CONCLUSION

This concludes this issues instalment on the LCD.

Study the previous notes carefully and get to know the LCD fully. There is enough information here for you to write routines using the LCD and we would like to see some ideas sent to us for issue 16.

The LCD will be supported further in issue 16 and if all goes well, we will have a cheap, full alpha-numeric key-board with supporting software. I am working towards the stage were you can anotate your routines and send the text and the routine in on tape. We can then load them into our desk top publisher.

Don't forget if you have any good ideas or questions about the TEC, send them in to "TEC TALK."

Below is the table of LCD control bytes. Use these in conjunction with the previous notes

Instruction	Code										Function	Execution time
	RS	$R/\overline{W}$	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0		
(1) Display clear	0	0	0	0	0	0	0	0	0	1	Clears all display and returns cursor to home position (address 0)	1.64 ms
(2) Cursor Home	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position. Shifted display returns to home position and DD RAM contents do not change.	1.64 ms
(3) Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets direction of cursor movement and whether display will be shifted when data is written or read	40 µs
(4) Display ON / OFF control	0	0	0	0	0	0	1	D	C	B	Turns ON/OFF total display (D) and cursor (C), and makes cursor position column start blinking (B)	40 µs
(5) Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves cursor and shifts display without changing DD RAM contents	40 µs
(6) Function Set	0	0	0	0	1	DL	1	*	*	*	Sets interface data length (DL)	40 µs
(7) CG RAM Address Set	0	0	0	1	A_{CG}						Sets CG RAM address to start transmitting or receiving CG RAM data	40 µs
(8) DD RAM Address Set	0	0	1	A_{DD}							Sets DD RAM address to start transmitting or receiving DD RAM data	40 µs
(9) BF/Address Read	0	1	BF	AC							Reads BF indicating module in internal operation and AC contents (used for both CG RAM and DD RAM)	0 µs
(10) Data Write to CG RAM or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM	40 µs
(11) Data Read from CG RAM or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM	40 µs

* : Invalid bit
A_{CG} : CG RAM address
A_{DD} : DD RAM address

I/D = 1 : Increment
I/D = 0 : Decrement

S = 1 : Display shift
S = 0 : No display shift

D = 1 : Display ON
D = 0 : Display OFF

C = 1 : Cursor ON
C = 0 : Cursor OFF

B = 1 : Blink ON
B = 0 : Blink OFF

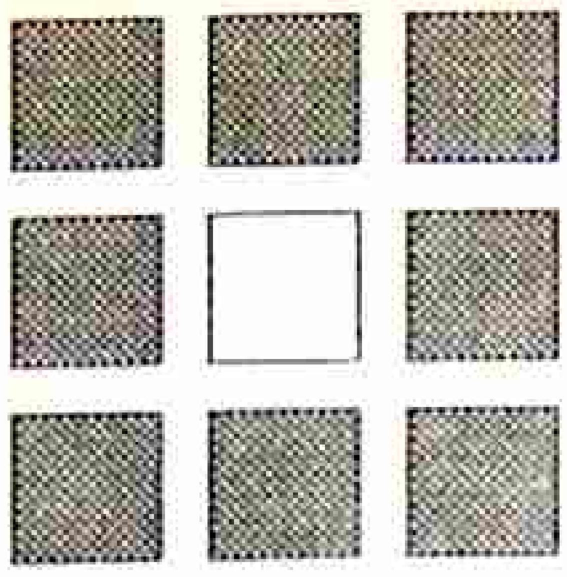
S/C = 1 : Display shift
S/C = 0 : Cursor movement

R/L = 1 : Right shift
R/L = 0 : Left shift

DL = 1 : 8 bits
DL = 0 : 4 bits

BF = 1 : Internal operation in progress
BF = 0 : Instruction can be accepted

MAGIC SQUARE



by Jim Robertson

This is a fun game for the 8x8 that will have you amused and frustrated for hours.
The object is to light up the outside square of the 8x8. The game is made up of three 2x2 boxes of LEDs with a space between each. This makes full use of the 8x8 to display a playing field that is actually 3x3.
Nine keys are used to play the game and each key corresponds to a group of LEDs on the display.

TO SET UP

This game, like JMON, requires EITHER a 4k7 resistor between the NM1 (pin 17 of the Z-80) and D6 (Pin 10 of the Z-80) OR the LCD expansion board with the input chip fitted on port 3.
The 8x8 is fitted to ports 5 and 6 with the port select strobe of the left-hand latch going to port 6.
This is very important! (once you master the game, try swapping them over, this will invert the playing field and gives you a mirror image to work with).
The 8x8 is placed with the LEDs above the latch chips.
It is important to fit the 8x8 before typing in the code or at least hold down the reset if you have already entered the code, by using your third hand.
MAGIC SQUARE has been written to run with the TEC crystal oscillator however it will work with the 4049 oscillator but the tones will be lower pitched.

TO PLAY

Type in the code and save it if you have a tape system. Now address 0C00 and press GO. The code is placed at 0C00 to allow Simon and Magic Square to be saved, loaded and played together (however they do not require each other). (Unfortunately Simon has been held over to issue 16 because of the shortage of space in this issue).
After starting the game, a random pattern appears. By pressing the game keys, the playing field will change. Each key has a particular effect that remains constant throughout the game. The effects of each key is for you to work out!
The keys used for the game are: 4, 5, 6, 8, 9, A, C, D and E.
As you can see, these make up a 3x3 box pattern on the keyboard.
Go to it! The object of the game is to light up the outside border with the centre OFF.
A fair point to add is that it is always possible to do this regardless of the starting pattern - believe it or not!
When (if!) you finally succeed, your effort will be greeted enthusiastically on the 8x8. The game may be re-started by hitting the GO key.

HOW THE SOFTWARE WORKS

Three random numbers are generated from the time it takes to release the GO key and also from the refresh register.
The three lowest bits of these three bytes are used to form a 3x3 matrix. The top 5 bits are ignored.
All processing, pattern changing and testing is done on this 3x3 matrix. After processing, this matrix is converted to its equivalent 8x8 display and then scanned. A loop is used to scan the 8x8 and read the keyboard until a key is detected.
When any key is detected for the first time, a flag byte remembers this and the program will ignore any subsequent pushes.
This allows each key to be processed just once. When no key is pressed, the flag is cleared to allow the next key to be processed.
When a key is pressed and allowed as a "FIRST KEY" press, it is checked for a corresponding table entry. If no corresponding value is found, the key is ignored. This is how the unwanted keys are masked.
After a key has been validated a table entry 9 bytes higher is accessed. This entry is a byte that will be exclusive-ORed with the first byte of the 3x3 matrix.
A second byte 9 bytes higher again contains the low order byte of the address of the 3x3 matrix entry. The first byte is now EX-ORed with the matrix byte and the result stored as the new updated matrix byte. This is how the patterns are changed.
The above process is repeated for the second and third matrix bytes. The exact same process described above is used. The entry for the second byte is 9 bytes higher than the first and the address 9 higher again.
The same convention is used for the third entry. This convention allows a loop to be used for all three matrix bytes. This loop is located at 0C49.
After the above process, the 3x3 is checked for the required box pattern. If correct, the pattern is converted to its 8x8 format and flashed with accompanying tones.
If the pattern is not complete, the program loops back to the main playing loop.
A routine at 0CAB converts the 3x3 to 8x8 display format. This routine is called after all the required processing has been performed on the 3x3 matrix. This routine is a loop that gets each 3x3 matrix byte, calls another routine to convert each matrix bit to two 8x8 bits and spacing, then stores the result twice and adds a blank line.
The last blank line is ignored by the scan routine and the result is an 8x8 format.
At 0CC4 a loop converts one bit to two and adds spacing. This is done by shifting the matrix bit into the carry and if the carry is clear, the two 8x8 bits are left clear and shifted twice for the 2x2 box bits and once for the space between.
If the carry is set, the 2x2 box bits are set by rotating the SET CARRY into the 8x8 byte and also setting bit 7 before rotating. This will then set the carry after the first rotation, ready for the second rotation. The third rotation clears the space bit. After this is done three times, the 8x8 byte is rotated back to remove the last unwanted space before returning.

THE TONE ROUTINE

The tone routine is located at 0CD8. The duration of the tone period is in D while the cycle count is in E. The "KEY PRESS" beep uses this value loaded into DE while other tones such as the restart tone load DE before calling the tone routine.

SCAN ROUTINE

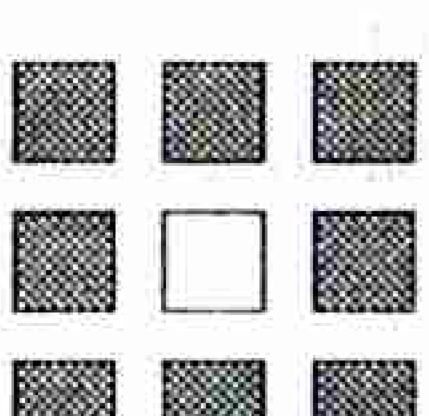
The scan 0CE7 is a straight-forward multiplex routine except that it scans backwards. This allows the 8x8 to be right-way-around while keeping the rest of the program straight forward (otherwise the 8x8 buffer would need to be loaded backwards).

"Magic Square" contains a number of very valuable "building blocks" that can be used in your own programs. It can stand studying for many hours to see how the various operations have been achieved. The fully documented program is presented on the next two pages and you should add your own notes alongside Jim's to help you understand what is happening at each step.

Colin Mitchell.

MAGIC SQUARE PROGRAM

	0C00	11 00 00	LD DE,0000	Random number generated
	0C03	13	INC DE	by the duration it takes the player to release the key at the start of the
	0C04	DB 03	IN A,(03)	program.
	0C06	CB 77	BIT 6,A	
	0C08	28 F9	JR Z,0C03	
	0C0A	ED 5F	LD A,R	The value of the refresh register is loaded into the accumulator.
	0C0C	82	ADD A,D	D register is added to the accumulator and stored as the first value.
	0C0D	32 40 0D	LD (0D40),A	E register is added (with carry) and stored as the second value.
	0C10	8B	ADC A,E	Registers are added to the accumulator and shifted to produce the
	0C11	32 41 0D	LD (0D41),A	third random number. This is also stored.
	0C14	82	ADD A,D	
	0C15	83	ADD A,E	
	0C16	07	RLCA	
	0C17	32 42 0D	LD (0D42),A	
MAIN	0C1A	CD AB 0C	CALL 0CAB	Call 3x3 to 8x8 conversion routine.
PLAYING	0C1D	CD E7 0C	CALL 0CE7	Call scan.
LOOP	0C20	DB 03	IN A,(03)	Test for key press.
	0C22	CB 77	BIT 6,A	If bit 6 on port 7 HIGH then no key is pressed.
KEY	0C24	28 06	JR Z,0C2C	Jump if key pressed otherwise clear "key pressed" flag and loop until
PRESSED	0C26	AF	XOR A	key pressed. Otherwise clear.
	0C27	32 43 0D	LD (0D43),A	"key pressed" flag.
	0C2A	18 F1	JR 0C1D	Loop until key pressed.
	0C2C	3A 43 0D	LD A,(0D43)	Test "first key press" flag.
	0C2F	B7	OR A	
	0C30	20 EB	JR NZ,0C1D	Jump if key already pressed, otherwise set key pressed flag
	0C32	3E FF	LD A,FF	
	0C34	32 43 0D	LD (0D43),A	
	0C37	21 00 0D	LD HL,0D00	HL = base of valid key table.
	0C3A	01 09 00	LD BC,0009	BC = number of valid key entries
	0C3D	DB 00	IN A,(00)	Get input value from encoder chip
	0C3F	E6 1F	AND 1F	mask unwanted bits
	0C41	ED B1	CPIR	block compare with increment.
	0C43	20 D8	JR NZ,0C1D	NZ means no right entry. After all values tested, ignore key.
KEY	0C45	CD D8 0C	CALL 0CD8	Key valid. Call key pressed beep.
VALID	0C48	2B	DEC HL	Decrement HL as CPIR increments it before testing the zero flag.
	0C49	11 09 00	LD DE,0009	DE = table index.
	0C4C	06 03	LD B,(03)	Set B for 3 loops. One for each matrix byte.
	0C4E	19	ADD HL,DE	Get value to EX-OR with matrix.
	0C4F	7E	LD A,(HL)	Save in A.
	0C50	19	ADD HL,DE	Calculate address of low byte of matrix byte and put in HL.
	0C51	E5	PUSH HL	Save for later.
	0C52	6E	LD L,(HL)	Set HL to matrix byte address.
	0C53	AE	XOR (HL)	Toggle bits and store
	0C54	77	LD (HL),A	as updated matrix byte
	0C55	E1	POP HL	Recover HL
	0C56	10 F6	DJNZ,0C4E	Loop for 3 bytes.
	0C58	21 40 0D	LD HL,0D40	Check for box pattern. (HL) = first matrix byte.
	0C5B	7E	LD A,(HL)	
	0C5C	E6 07	AND 07	Remove unwanted bits
	0C5E	FE 07	CP 07	and test for 7 (111)
	0C60	20 B8	JR NZ,0C1A	Jump to main playing loop if not 7, otherwise
	0C62	23	INC HL	Test second matrix byte.
	0C63	7E	LD A,(HL)	
	0C64	E6 07	AND 07	
	0C66	FE 05	CP 05	Test for 5, (101)
	0C68	20 B0	JR NZ,0C1A	Jump if not, otherwise
	0C6A	23	INC HL	do third matrix
	0C6B	7E	LD A,(HL)	byte which should
	0C6C	E6 07	AND 07	be equal
	0C6E	FE 07	CP 07	to 7 (111)
	0C70	20 A8	JR NZ,0C1A	Jump if not box pattern.
PATTERN	0C72	CD AB 0C	CALL 0CAB	Pattern right so call 3x3 to
DONE!	0C75	11 30 00	LD DE,0030	8x8. Load DE with win tone
	0C78	CD DB 0C	CALL 0CDB	and call tone routine.
	0C7B	06 03	LD B,03	Set B for 3 flashes.
	0C7D	C5	PUSH BC	and save count
	0C7E	16 10	LD D,10	D = scan counter
	0C80	CD E7 0C	CALL 0CE7	Call scan.
	0C83	15	DEC D	Loop until D = 0
	0C84	20 FA	JR NZ,0C80	
	0C86	AF	XOR A	Clear display.
	0C87	D3 06	OUT (06),A	
	0C89	CD D8 0C	CALL 0CD8	Call beep.
	0C8C	01 00 15	LD BC,1500	Load BC with off time
	0C8F	0B	DEC BC	and delay.



	0C90	78	LD A,B	
	0C91	B1	OR C	
	0C92	20 FB	JR NZ,0C8F	
	0C94	C1	POP BC	Recover flash loop counter
	0C95	10 E6	DJNZ 0C7D	and loop for 3 flashes.
	0C97	CD E7 0C	CALL 0CE7	Call scan.
	0C9A	DB 00	IN A,(00)	and loop continuously
	0C9C	E6 1F	AND 1F	looking for the GO key
	0C9E	FE 12	CP 12	to be pressed.
	0CA0	20 F5	JR NZ,0C97	Jump if GO not pushed.
	0CA2	11 80 00	LD DE,0080	Load DE with restart tone
	0CA5	CD DB 0C	CALL 0CD8	Call tone.
	0CA8	C3 00 0C	JP 0C00	Restart game.
3x3	0CAB	06 03	LD B,03	B = loop counter set for 3 conversions.
to	0CAD	21 40 0D	LD HL,0D40	HL = address of 3x3 matrix.
8x8	0CB0	11 50 0D	LD DE,0D50	DE = 8x8 buffer.
MATRIX	0CB3	C5	PUSH BC	Save loop counter.
TO	0CB4	7E	LD A,(HL)	Get matrix byte.
DISPLAY	0CB5	CD C4 0C	CALL 0CC4	Call 1 to 3 bit conversion.
FORMAT	0CB8	12	LD (DE),A	Save first display
	0CB9	13	INC DE	
	0CBA	12	LD (DE),A	byte twice
	0CBB	13	INC DE	and then
	0CBC	AF	XOR A	add
	0CBD	12	LD (DE),A	a blank line
	0CBE	13	INC DE	increment to next display buffer.
	0CBF	23	INC HL	Increment HL to next matrix byte.
	0CC0	C1	POP BC	Recover loop counter.
	0CC1	10 F0	DJNZ 0CB3	Repeat for 3 bytes.
	0CC3	C9	RET	done.
1 TO 3 BIT	0CC4	01 00 03	LD BC,0300	B = 3 loops. C is cleared ready to receiver display byte.
CONVER-	0CC7	0F	RRCA	Rotate matrix byte to set or clear carry.
SION	0CC8	30 02	JR NC,0CCC	Jump NC to shift C 3 places
	0CCA	CB F9	SET 7,C	else set bits 1 and 2 of C with SET CARRY and
	0CCC	CB 11	RL C	bit 7
	0CCE	CB 11	RL C	rotate C left
	0CD0	CB 11	RL C	Last rotation inserts space
	0CD2	10 F3	DJNZ 0CC7	do for 3 loops
	0CD4	CB 19	RR C	remove last space
	0CD6	79	LD A,C	place result in A.
	0CD7	C9	RET	done.
BEEP	0CD8	11 50 50	LD DE,5050	D= period E = loop counter
	0CDB	AF	XOR A	Clear A.
TONE	0CDC	D3 01	OUT (01),A	Sound out to speaker.
	0CDE	42	LD B,D	Delay for tone
	0CDF	10 FE	DJNZ 0CDF	period.
	0CE1	EE 80	XOR 80	Toggle bit 7,A (speaker bit)
	0CE3	1D	DEC E	Decrement loop counter.
	0CE4	20 F6	JR NZ,0CDC	Loop until zero.
	0CE6	C9	RET	Done.
SCAN	0CE7	21 57 0D	LD HL,0D57	HL = end of 8x8 buffer.
	0CEA	06 80	LD B,80	B = scan bit output byte.
	0CEC	7E	LD A,(HL)	Output first display
	0CED	D3 05	OUT (05),A	byte to port 5
	0CEF	78	LD A,B	then output scan bit
	0CF0	D3 06	OUT (06),A	to port 6.
	0CF2	06 40	LD B,40	short multiplex
	0CF4	10 FE	DJNZ 0CF4	display delay
	0CF6	2B	DEC HL	Decrement HL to next display byte
	0CF7	47	LD B,A	replace scan bit in B.
	0CF8	AF	XOR A	clear accumulator and
	0CF9	D3 06	OUT (06),A	output to port 6.
	0CFB	CB 08	RRC B	Shift scan bit loop until scan bit
	0CFD	30 ED	JR NC,0CEC	falls into carry
	0CFF	C9	RET	then return.

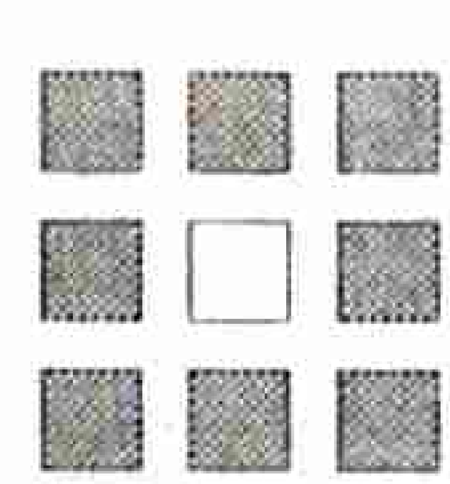
TABLES:

0D00: 04 05 06 08 09 0A 0C 0D 0E 06 04 00 07 02 00 03

0D10: 01 00 40 40 40 40 40 40 40 40 40 40 06 04 02 07

0D20: 02 01 03 01 41 41 41 41 41 41 41 41 41 00 04 06

0D30: 00 02 07 00 01 03 42 42 42 42 42 42 42 42 42

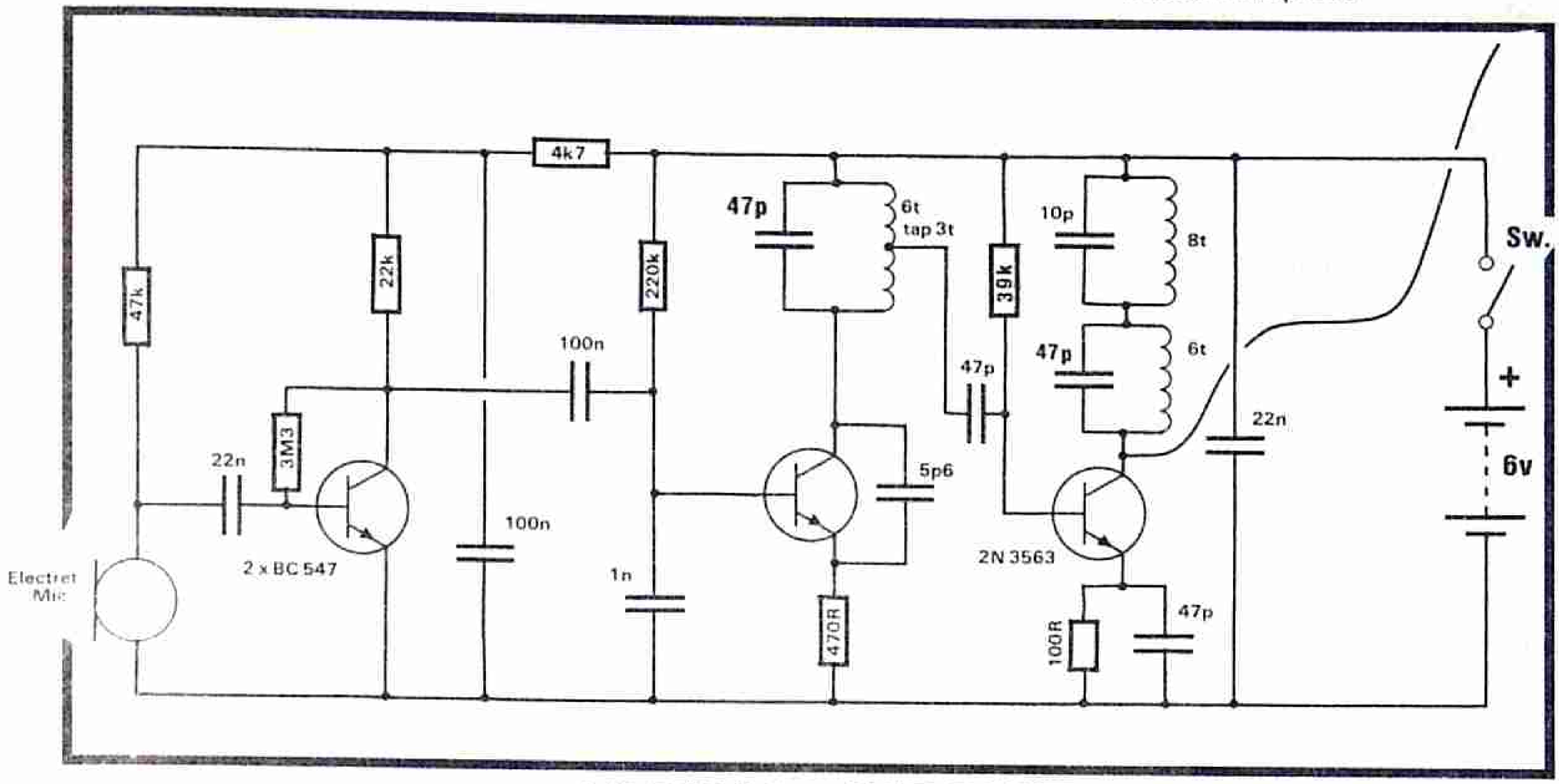


Build TE's

ULTIMA

Our 1km FM transmitter

Parts $10.00
PC board $2.50
Total $12.50
Don't forget the LED Power Meter kit: $1.10



ULTIMA CIRCUIT

Encouraged by the popularity of all our previous bugs, we have designed another FM transmitter, this time with a range of 1km.

The efficiency of this design is slightly higher than the Amoeba as the power consumption is about double and yet the Effective Radiated Power (ERP) is about 4 times.

This is borne out by the fact that the range is about twice as far - it requires 4 times the power to double the range.

In this project we will cover some of the design features that make the circuit so efficient. Basically it is the higher voltage. The way it works is this:

This design has a minimum threshold of about 1v so that a circuit powered by a 1.5v battery has very little potential for producing any output. At 3v it can reach about 400 metres and at 6v it can reach about 1km. You will recall the increased range of the Voyager was purely due to the 9v supply and this time the 6v supply, as well as the tuned output, will assist in producing the output power.

With any transmitting device, the primary requirement is range. To achieve this, an effective antenna system must be used and although a dipole or loaded antenna provide a slight antenna gain, the usual purpose for which this project is intended, makes a special antenna impractical.

The main use for the the Ultima is as an emergency beacon or signaling device and in such a situation, the user may only have the branch of a tree to throw the antenna over.

For this reason, and to keep the transmitter compact, a length of hook-up wire about 1.6metre long is provided.

You could say it's another version of our Hiker's alarm, this time without the beep, and is small enough to be carried by anyone on a hiking expedition or skiing holiday.

If a number of adventurers got together and produced a few units, they could install them on and around the area they frequent and provide a network of communication links in case of an emergency.

PARTS LIST

1 - 100R (all resistors 1/4 watt)
1 - 470R
1 - 4k7
1 - 22k
1 - 39k
1 - 47k
1 - 220k
1 - 3M3

1 - 5p6
1 - 10p
4 - 47p
1 - 1n
2 - 22n
2 - 100n monoblocks

2 - BC 547 transistors
1 - 2N 3563 RF transistor
1 - 6 turn tinned copper wire coil
1 - 6 turn enameled wire coil
1 - 8 turn enameled wire coil
1 - electret microphone insert
4 - AAA or AA cells
1 - mini slide switch SPDT
1 - length hook-up wire for antenna (1.6m long)

1 - "ULTIMA" PC BOARD

(You can specify AA cells or AAA cells)

It would only take one or two instances where a transmitter of this type proves to be a life saver and its worth will be incalculable.

The advantage of transmitting on the FM band should be obvious as most cars are now fitted with an FM receiver and they can be used to pick up the transmission without the need for any special receiving equipment.

If a frequency were set aside at the lower end of the FM band for such a purpose, a safety channel could be set up at all outdoor activity centres. The transmissions would not interfere with any commercial operations as few if any FM stations operate in remote areas.

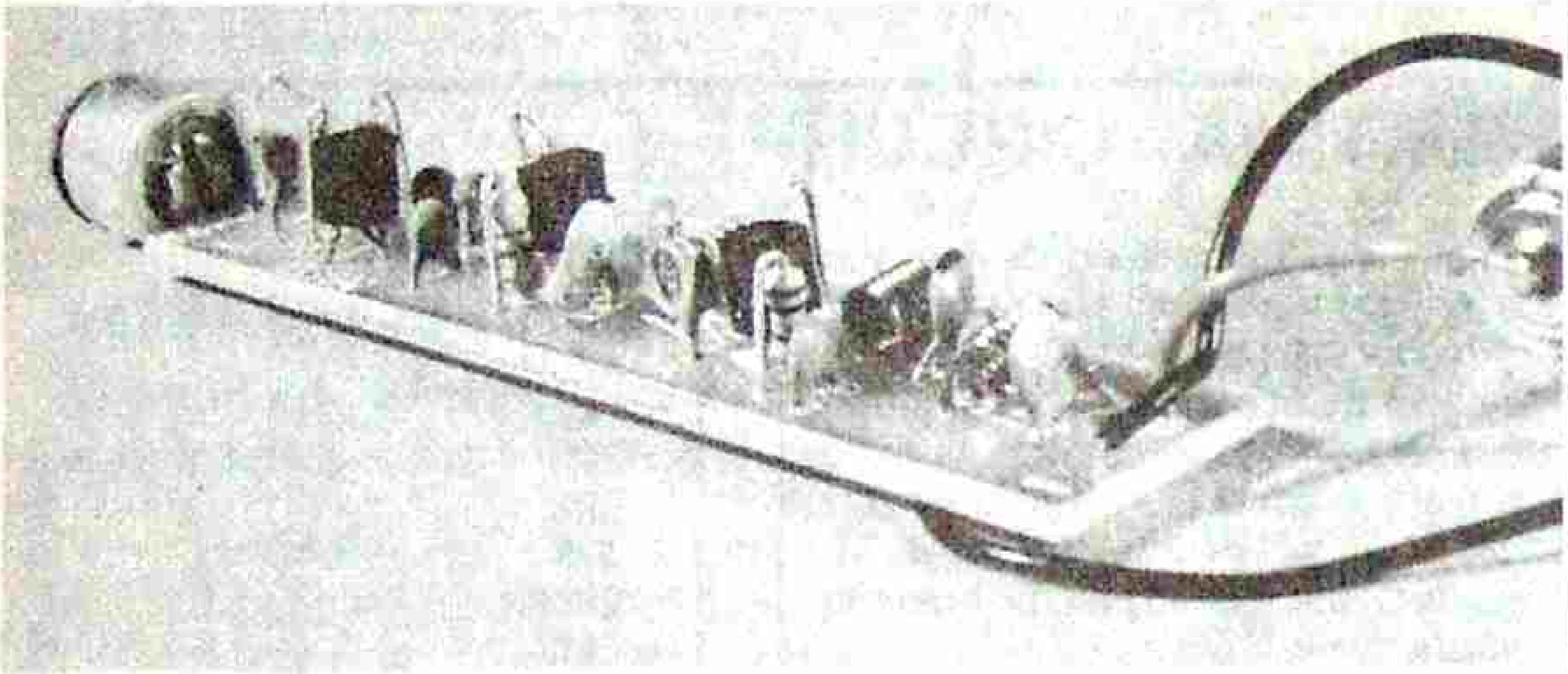
Maybe something like this will be introduced in the near future. It's a "catch 22" situation where you would have to start by cutting through a lot of red tape to get something like this accepted nationally.

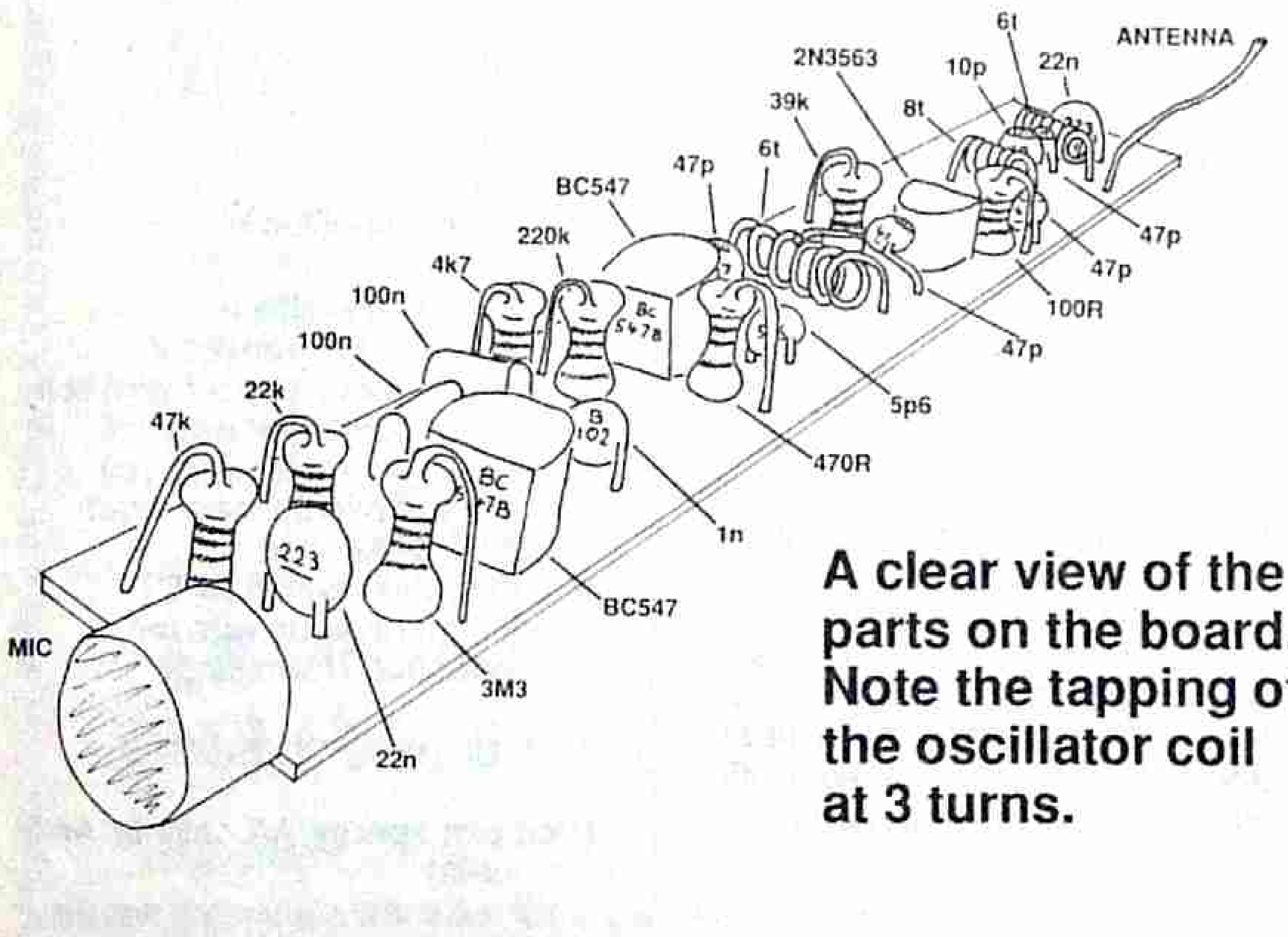
The Ultima project is built on a PC board 1.5cm x 7cm and can be fitted into a variety of cases. The most appropriate will depend on the intended use and if you want the project to be as small as possible, it can be heatshrunk and made virtually watertight.

FM transmission is highly efficient and a few milliwatts can get you a long way. The clarity of FM, its lack of background noise and immunity to electrical interference makes it ideal for faithfully monitoring all types of sounds.

The impressive range of the Ultima makes it ideal as a surveillance device, for monitoring remote sheds and buildings, where the slightest disturbance will be picked up and relayed to the monitoring point.

The Ultima is not intended as a hand-held microphone as firstly the microphone sensitivity is too high and the circuit is not "tight" enough to prevent the effect of stray capacitance of your body causing the frequency to drift. See our hand-held transmitter, the ANT, for this.





A clear view of the parts on the board. Note the tapping of the oscillator coil at 3 turns.

HOW THE CIRCUIT WORKS

The main purpose of a transmitter is to get the greatest distance with the least current consumption.

To this end, the Ultima is the ultimate in design. It contains 3 novel features that have possibly never before been incorporated in the one design.

The front end is a simple common emitter stage with a gain of about 70-100, and decoupled from the battery via a 4k7 and 100n capacitor. These two components prevent motor-boating (instability) at low frequency.

There's another little known fact to be aware of, when designing the front end of a high gain amplifier such as this.

Since the electret microphone is an active device (it contains a FET), the gain of the FET must not be allowed to be too high, otherwise the front end will break into oscillation (sometimes called front-end squeal).

The gain is kept low by making the load resistor high and that's why we have chosen 47k.

Coupling the audio to the RF oscillator is a 100n capacitor and this value is needed to pass the low audio frequencies.

The first of our "unusual" features is the 220k turn-on resistor on the base of the oscillator transistor.

After a great deal of experimenting we found a very high value base resistor was sufficient to turn the transistor ON and produce a high amplitude waveform.

The oscillator circuit has been designed to oscillate at about 88MHz.

The tuning capacitor in the oscillator is 47p and when the frequency is set, it remains set. The Ultima is not a tunable bug and you should select a clear spot on the dial when setting the bug up, so that you don't interfere with any radio stations.

Apart from being illegal to transmit over a commercial station, you would have almost no chance of achieving any range if you were to do so.

The output of the oscillator is taken from the third turn of the oscillator coil and this is common practice in transmitter circuits - it has been done with the Amoeba and has proven to be very successful.

Quite often the tapping of a coil is very critical and half a turn either side of the optimum value will reduce the performance considerably.

The idea of tapping a coil is to produce an auto transformer. This has nothing to do with cars (auto's) but means a two-winding transformer is produced by using only a single winding.

We all know what a transformer does and one of its features is to change a high voltage, low current packet of energy into a low voltage, high current package.

In a limited way, that's what the tapping on the oscillator coil does. It produces a very efficient matching between the oscillator section and the output stage.

The aim of the tapping is to pick off as much signal as possible without overloading the oscillator.

In technical terms we need a considerable amount of current to drive the output stage as its gain is quite low (possibly about 5-20), depending on the quality of the transistor and the frequency of operation.

At 100MHz, the reactance of the 47pF in the base of the output transistor will be only about 30 ohms and that's why a small value such as this can be used.

The output stage has a number of features worth mentioning.

Firstly the emitter capacitor and resistor network may appear to be unnecessary as the stage is not a full bridge design. These components reduce the peak current and increase the input impedance of the output stage as seen by the oscillator stage. This improves and guarantees "start-up" of the circuit.

To get the greatest range out of the Ultima, the output stage must be peaked and this is an essential part of the tuning and aligning. The tuned circuit in the output stage consists of a 6 turn coil and 47pF capacitor. The 8 turn coil is effectively an RFC (Radio Frequency Choke) to improve the matching of the output circuit to the antenna. The 10p capacitor across the RFC is designed to reduce harmonics and prevent interference on TV sets in the vicinity.

As output power is increased (such as from the Gnat design, to this), the effect of harmonics becomes a real problem.

Shielding the project will have little effect as they are already appearing on the antenna and it's too late to suppress them once they get this far.

As we have said, one of the important stages in the construction is to peak the output. For this, a simple POWER METER is required and its construction is also covered. The circuit attaches to a standard multimeter and uses the meter to give an indication of the signal strength. It also has a Light Emitting Diode that illuminates to give an indication of the relative power being emitted.

The illumination of the LED will give you a valuable indication when peaking the circuit and will show how critical the values are at high frequency.

An on/off switch allows the project to be turned off when not required and the current consumption of our prototype was measured at 8mA, allowing a set of AA cells to last for about 150 hours.

The 2N 3563 transistor in the output has been chosen as it gives the best performance at the least cost.

Surprisingly, a BC 547 will operate quite satisfactorily as an oscillator at 100MHz but provides very little gain at this frequency.

It only requires a gain greater than unity to function as an oscillator but in the output, a 2N 3563 provides a gain of between 5-20 and delivers the output we need.

No matter what other tricks you add to the output stage, you will not achieve better performance - for the same current.

The suggested maximum current for a 2N 3563 is 10-12-15mA (depending on the specification sheet you use), and only by decreasing the value of the bias resistor, does the output rise by a noticeable amount.

As you experiment with the circuit and peak it, you will see the changes in output power on the meter and/or the LED. But if the range is increased at the expense of more current, you haven't improved the efficiency.

Our prototype produced 2v across the LED/resistor combination and the chip inside the LED could be seen to glow quite noticeably.

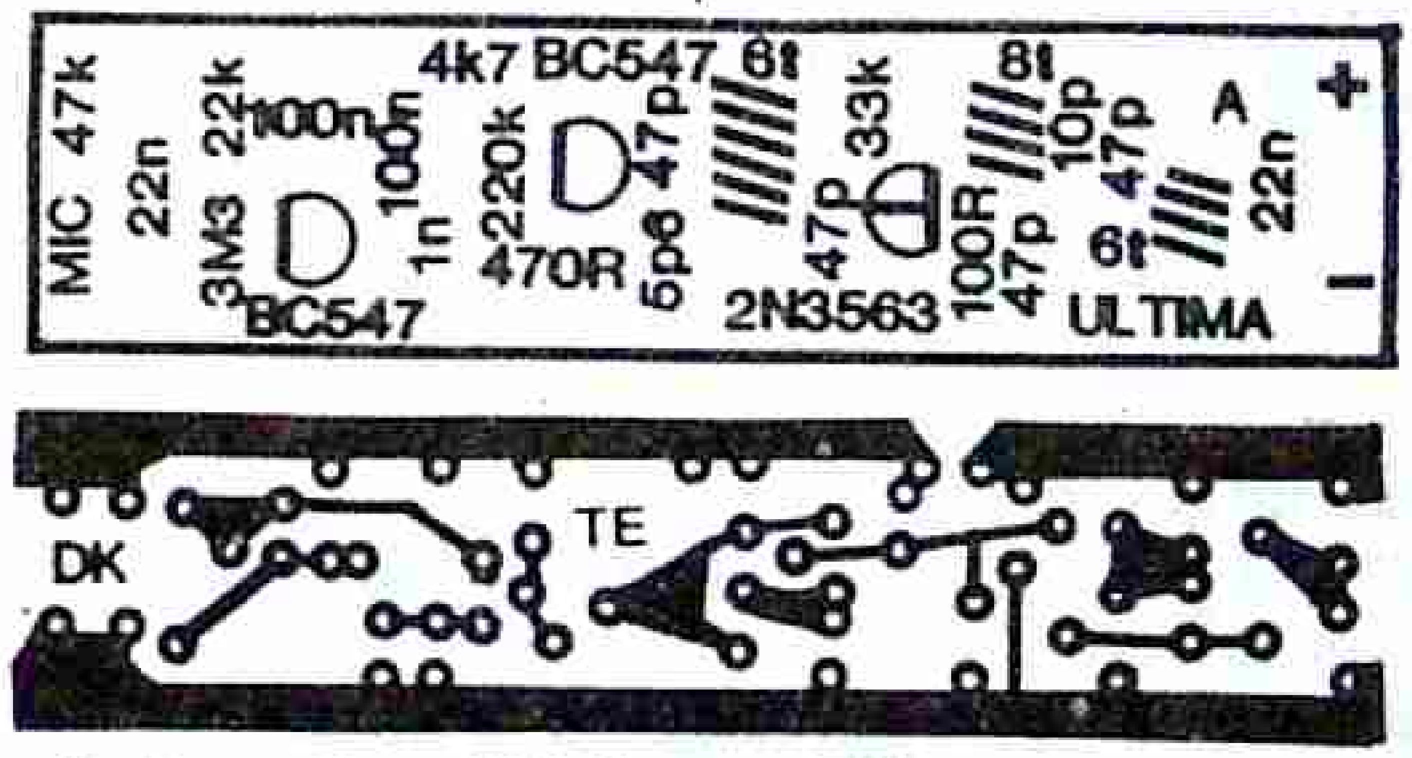
Not a great deal of weight can be put on this as a lot of RF energy is floating around the meter and will get into the movement to create a false reading.

But if you can see the LED glowing you know that energy MUST be present and that's why we have called it a LED POWER METER.

ASSEMBLY

A complete kit of components, including pre-wound coils and roll-tinned PC board is available from us and we hope you will buy more than one kit! Many readers buy two or more kits for their friends and some keep coming back for more! We have sold over 10,000 ANT bug kits already! The artwork for the PC board is also given so that you can make your own, if you are set up to do so.

The economics of making your own board is dubious.



PC artwork for the Ultima. The switch is soldered directly to the AAA cell. The positive and negative leads connect to the board via short lengths of hook-up wire.

By the time you buy the blank board, cover it with ultra-violet sensitive material and spend an hour or so etching and drilling, you could have bought a professionally made board from us for less.

Before arriving at the final design, we built at least 12 experimental versions, all with parts hanging off the board, in a birds-nest array.

Without exaggeration, it has taken hundreds of hours of experimenting and at least 100 trips "around the block" to see how effective each modification has been.

Sometimes an improvement via the power meter did not co-relate to an improvement in range. You must always make a "field test" - it's the only proof.

Although the circuit is not critical as such, a lot of work has gone into the selection of each component and the layout of the board.

If you want your model to perform as good as ours, it is important to use the exact same components and the same layout. That's why I suggest you buy a kit.

If you are a seasoned hobbyist, and know what you are doing, you will know the type of components to use. But if you are unsure or missing one or two of the critical parts, don't take the chance, buy a kit.

Kits have proven the most popular way of making a project.

The ratio of "kits" to junk-box assembly is greater than 50:1, in our experience of over 100,000 kits. Kits are generally no more expensive than buying the parts individually and quite often some of the special components are not readily available, like the wire for the coils, the RF transistor and high quality electret microphones.

I am not going to go through the finer points of placing the parts on the board as we expect you will have made some of our other designs already.

The coil winding details are below for those who wish to wind their own. You can use slightly different gauge wire but the coil diameter and spacing is critical.

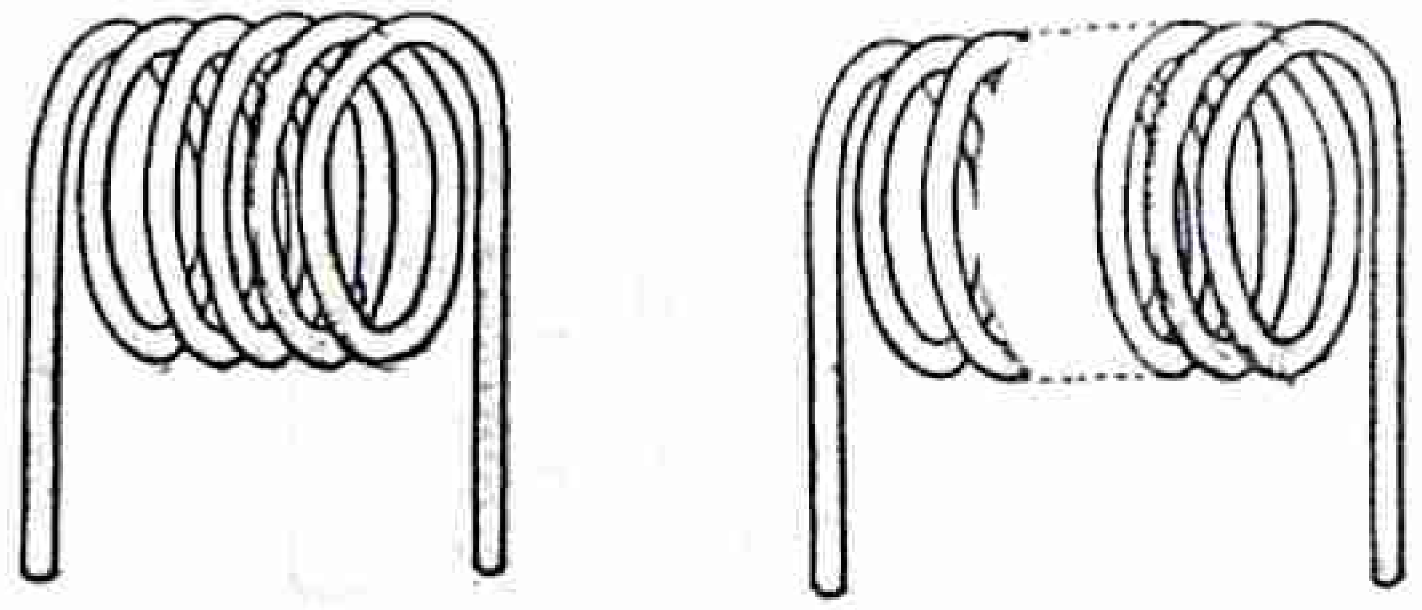
All the parts for the Ultima fit onto the board almost exactly as they appear in the circuit diagram. This makes the circuit very easy to follow and is one of the factors to good layout.

Two of the coils are enamel coated and the enamel must be scraped off or burnt off before fitting them to the board. It is not satisfactory to expect the enameling to burn off at the time when the coil is being soldered in place.

All the components should be pressed firmly up to the board before soldering. This also applies to the transistors. They should be pushed down to the same height as the resistors as can be seen in the photograph.

There are three points that may be new to many constructors. The first is the soldering of the 47pF to the top of the 3rd turn of the oscillator coil (the tinned copper wire coil).

The turns are counted from the "earthy" end and earthy end means the end that is not alive or oscillating.



COIL WINDING DETAILS:

1 - 6 turn tinned copper coil (.6mm wire) wound on a 3mm diam screwdriver.

1 - 6 turn enameled wire coil (.5mm wire) wound on a 3mm diam screwdriver.

1 - 8 turn enameled wire coil (.5mm wire) wound on a 3mm diam screwdriver.

The turns are counted at the top of each loop. Six loops at the top indicates a six turn coil etc.

The end connected to the positive rail is not "alive" and although it not earth as far as DC is concerned, the rail is "dead" as far as the signal is concerned - the active end is the collector end.

When making the connection, make sure the solder does not bridge across two turns as this will create a shorted turn and the transmitter will either not work at all or operate on a much higher frequency - we have had both occur, although I would prefer to state that the shorted turn will prevent the circuit from operating at all.

You should also take care that bending the leads of the 47pF does not damage the capacitor or create an open circuit. They are fairly delicate.

Finally, removing the enamel of the coils, must be mentioned.

The acid fumes that are produced when burning the enamel are so strong that they will attack the plating on the tip of your soldering iron.

I have already gone through a number of tips and put it down to this procedure as the pitting has developed after removing the enamel from a number of coils. Normally a tip will last for months but after a few "removals," the tip succumbs to a "crater." Soon after, the tip falls off.

You should remove the enamel with a blade, sandpaper or file and the ends must be cleaned and tinned before fitting the coil.

All the rest of the construction is straight-forward. The PC board contains a legend and the accompanying photos will show where everything goes.

The next stage is tuning and peaking but before this can be carried out, you will need to make up the LED POWER METER.

LED POWER METER

To get the best performance out of this transmitter, an RF power meter is used to maximize the output.

Since the output is very low, a conventional RF power meter cannot be used. We need an RF milliwatt meter.

The circuit described in this section is very simple and uses a standard multimeter to show the reading.

Across the input is a LED and resistor and the degree of brightness of the LED, together with the voltage reading on the meter, gives an indication of the energy level.

A digital multimeter may be used but the presence of RF may produce a false reading.

Likewise, the radiated energy may upset some analogue meters and you may get full scale deflection on the 15v range as well as the 250v range! But the LED won't lie. It will accurately indicate the RF and you can see the change in brightness as you adjust the coils in the output stage.

LED POWER METER

PARTS LIST:

1 - 470R
1 - 100p ceramic
1 - 100n greencap

2 - 1N 4148 signal diodes
1 - 5mm red LED
1 - 5cm hook-up wire
2 - paper clips

NO PC board required

Build the circuit for the LED Power Meter exactly as shown in the photo and make sure the input lead is exactly 5cm long. If you keep to the same layout as shown, your readings will closely coincide with ours.

When dealing with RF, lead length is very important and if the input lead is longer, the meter will produce a lower reading.

The type of multimeter will also affect the reading and this is why we cannot give a quantitative value for the output.

We don't have any means of providing a "standard" as we don't have any bench-mark or reference point.

As soon as you build the power meter you will be able to test the transmitter.

Connect the input lead to the antenna point (don't fit the antenna lead yet) and switch on. Switch the multimeter to the 2v (or 2.5v or 5v) range.

Keep the transmitter away from the meter to prevent the movement being influenced by the RF and you should get a reading of about 2v.

The LED will glow quite noticeably and you can see the output on the LED before the circuit is peaked.

Next you have to tune the transmitter to a clear spot on the dial. For this part of the operation you will need a transistor radio, or more preferably, a radio with a tuning indicator.

Keep the radio at least 2-3 metres away and tune to a clear spot.

The lead of the power meter is used to radiate the signal and you will get a range of 10 metres or so.

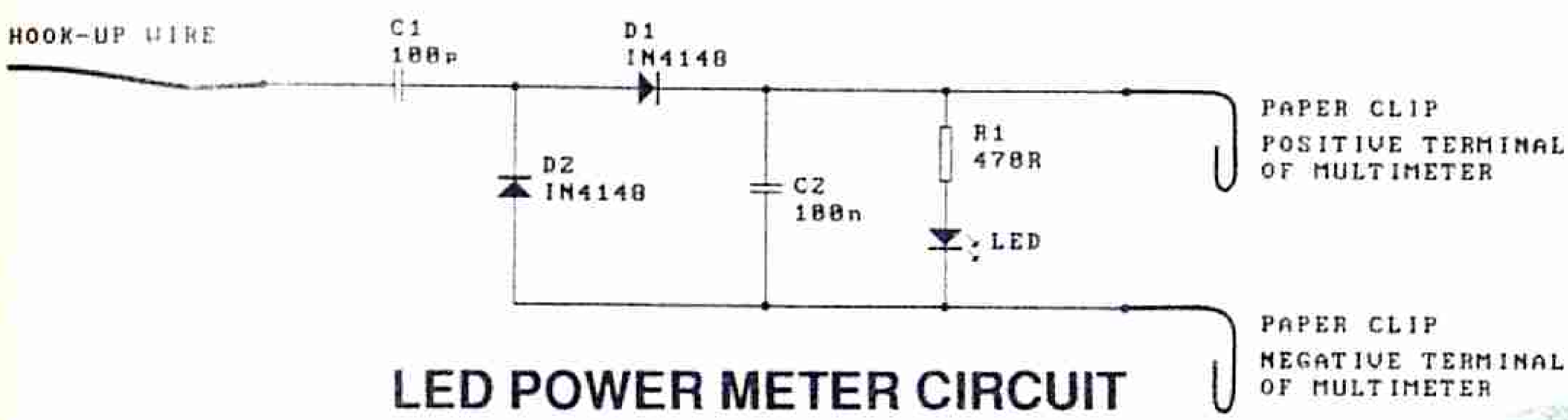
Turn the transmitter ON and adjust the turns of the tinned copper wire coil to get a feedback whistle from the radio.

There may be a number of spots where a signal is detected and it is important to pick the fundamental.

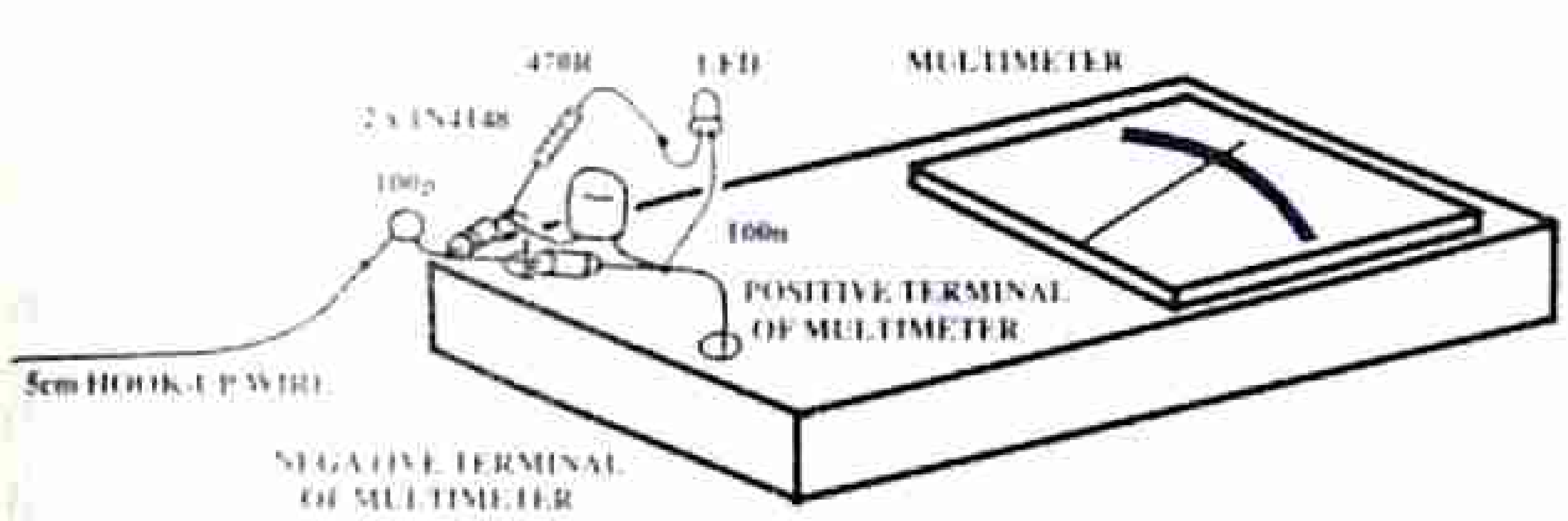
The weaker side-tones have no range and this is where a tuning indicator comes in. It will quickly indicate the fundamental frequency.

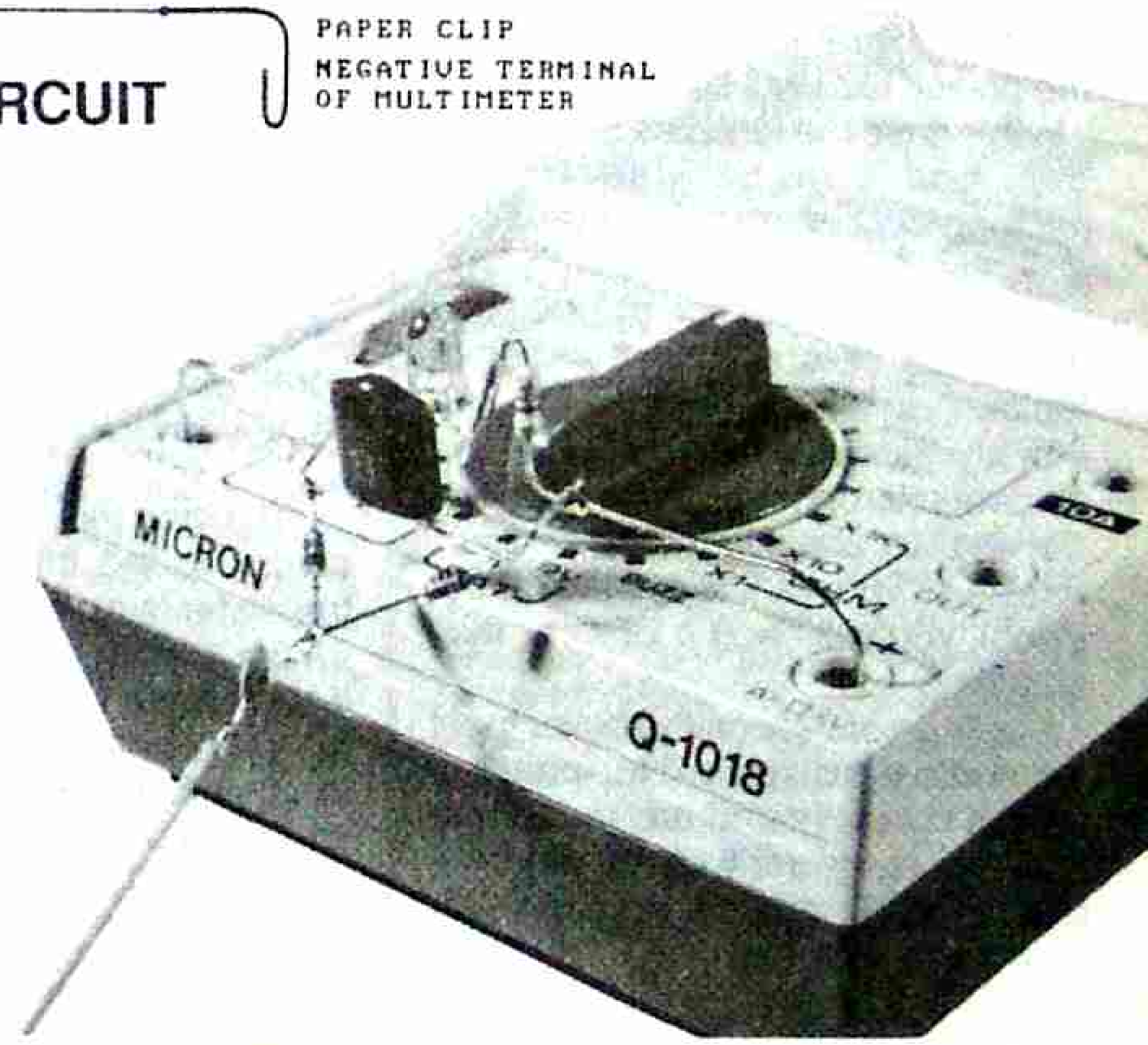
Move the radio away to confirm that the fundamental is being detected.

Once the frequency of transmission has been set, the peaking of the tuned circuit (in the output stage) must be undertaken. This is done by squashing or stretching the turns of



LED POWER METER CIRCUIT





LED POWER METER

These diagrams give a clear view of the LED POWER METER connected to an ordinary multimeter. The parts are soldered to two paper clips and connected to the input terminals of the meter. Set the meter to low DC volts scale. Make sure the input lead is 5cm long to get the same readings as discussed in the article.

the 6 turn coil and as you do this, you will be able to see the effect on the LED POWER METER.

In this way the output stage can be optimized and you can be sure the maximum output has been gained.

MODS

If you want to detune the transmitter to below 88MHz or run it at the top end of the band, you will have to change the oscillator section as well as the output section as they both determine the frequency.

To detune the transmitter to say 85-87MHz, you will need to add one turn to both the oscillator coil and the tank coil, making them 7 turns.

To operate at 108MHz, you will need to change both the 47p's in the oscillator stage and tank circuit to 39p and follow the peaking procedure as described above.

IF IT DOESN'T WORK

Before you get involved in any technical problems, have someone check the construction for dry joints, parts placement, shorts, and the like. Nine times out of ten it's something simple.

Next check the current. It should be 8-10mA and any value outside this range will indicate a fault is present. Also check the voltage across the power rails and the voltage across the front end (about 5v).

Next you must determine if the fault is in the audio section or the RF section.

If the carrier is being produced you will get a blank spot on the dial and the power LED will light up.

In this case the fault lies in the front end and can be due to the microphone being around the wrong way, the audio transistor being faulty, the 100n capacitor not passing the signal or one of a number of faults.

Firstly check the voltage across the electret microphone. If it is between .7v and 2v, the mic is drawing current and if you want to see the output signal, you will need a CRO or mini amplifier.

A voltage of 2-3v on the collector of the audio transistor will indicate it is biased correctly and any value outside this range may mean the transistor is faulty or has a gain above or below that expected.

In a self-biasing stage such as this, the gain of the transistor sets the collector voltage and maximum amplification is achieved when the collector is sitting at mid rail.

Two other components affecting the audio are the 22n and 100n capacitors.

They may be damaged due to soldering and the only way to check them is by bridging another capacitor across each or using a mini amplifier to detect the audio.

If you have access to a CRO, you can observe the signal at each point in the circuit and almost no fault will escape you.

Replacing all the components in the front end is a last resort but may be necessary if simple tests do not reveal the fault.

On the other hand, if no carrier is produced, the oscillator or output stages will be at fault.

Firstly check the oscillator stage by removing the 47p capacitor from the oscillator coil and connecting the input lead of the LED POWER METER to the collector of the BC 547.

The meter should deflect slightly but the LED will not come on as it requires at least 1.7v to be present before the LED will start to illuminate.

The LED and resistor do not have to be removed as they do not impose any load on the circuit until the voltage rises to above 1.7v.

If the meter does not deflect, the fault will lie in the oscillator stage. The first component to change is the transistor, then the 5p6 feedback capacitor.

Make sure the 220k turn-on resistor is providing a voltage to the base of the oscillator transistor. It will be difficult to measure this voltage accurately as the stage is producing RF and will upset the meter reading.

Make sure the 470R has a voltage across it and this will indicate the transistor is turned-on.

If the 47p capacitor has shorted, or if two turns of the coil are touching, the circuit will not oscillate. If these tests fail to locate the fault, you should assemble another kit and come back to this one later.

NO deflection of the LED POWER METER can only mean one thing. The output stage is not operating.

Firstly test the output transistor and make sure the voltages are correct. Don't forget the 47p that taps onto the 3rd turn of the oscillator coil. Replace it if you think it has been damaged during soldering.

The enamel on the coils must be removed before soldering as it does not conduct electricity at all.

Make sure the 39k provides a turn-on voltage and you can try shorting across the 100R if you think this resistor is open.

A low RF output can be due to an open 22n capacitor across the battery, the coils in the collector being spread too far apart or a faulty RF transistor.

The only other possible explanation is damage to some of the components during soldering.

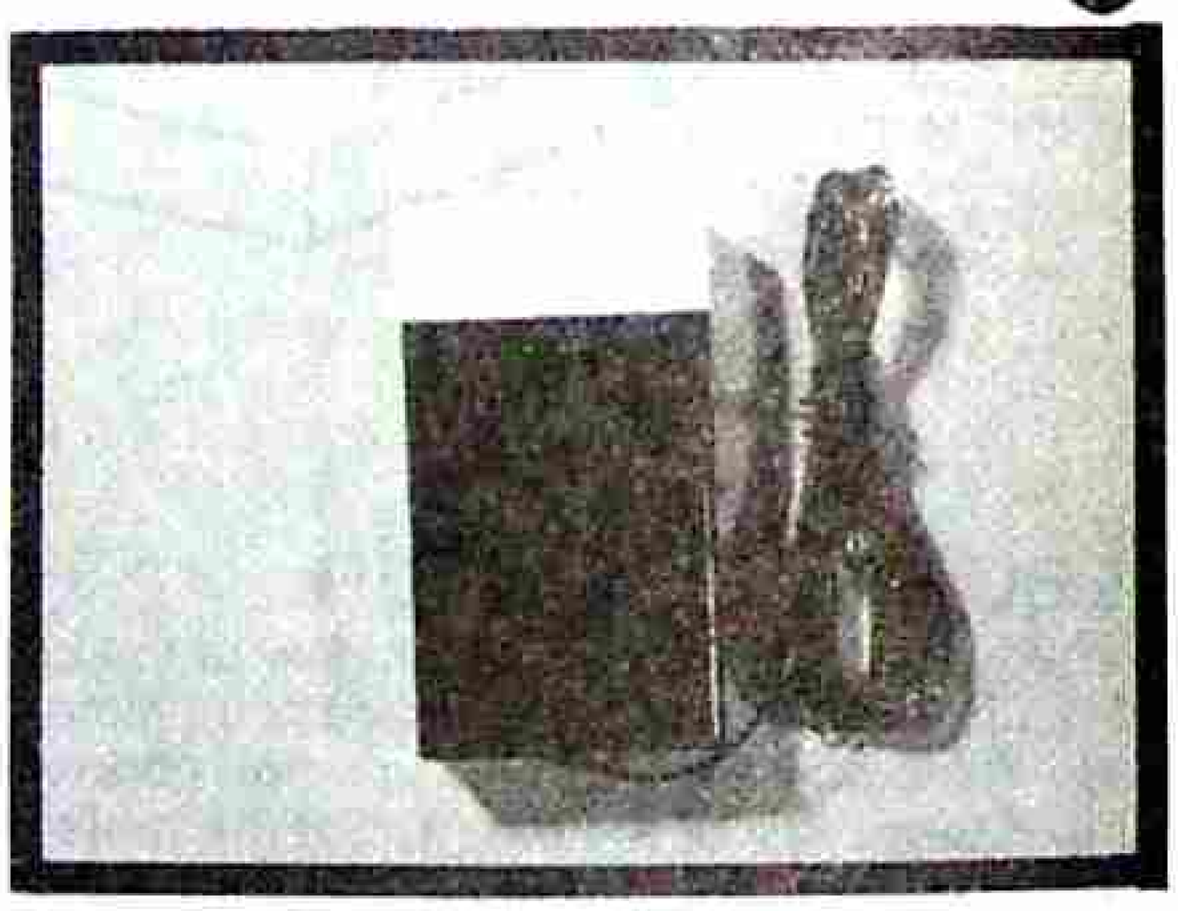
For the cost of another kit, the quickest and easiest way is to start again. You will take more care with the next construction and hopefully things will work out successfully.

MOUNTING IN A BOX

The project can be mounted in a jiffy box that is large enough to take the PC board and a battery holder. This will give it protection from the weather etc if used in an outdoor situation.

If you want the project to be as small as possible, the PC board can be heatshrunk but you must be careful that the heatshrinking does not move the coils and change their frequency. Do not use a voltage higher than 6v as the circuit is not designed for it and harmonics will be produced that will interfere with TV reception.

I hope you find this project as interesting as we did. The range is most impressive and it is incredible that such a low output will reach 1km.



SHOP TALK

I hope you have noticed the typesetting in this issue. It's been done on a laser printer.

For those not up with this modern marvel, a laser printer is simply a photocopier with a point source of light (the laser) that scans across a light-sensitive drum to set up the charges. After that, the printer works in the normal way to a photocopier and out comes the page of printing.

In our case the printer is a 300 dot model and produces a quality that is suitable for directly pasting up for the magazine. The advantage of a laser printer is flexibility.

Staff members can prepare articles at home, on their own IBM compatible computers and bring them to work for page making on our hard-disk system, using Ventura Publisher.

Although all publishing packages have a number of limitations and take a long time to learn to drive, we have found Ventura to be the most suitable for our requirements.

Computer preparation has certainly been a quantum leap for us. It was at first difficult for us to see the benefits of outlaying a lot of funds for a magazine that comes out so infrequently. But with the promotion and dedication of Jim, Paul and Craig, the benefits were soon realised.

The second advantage of computer setting is the interchange of articles. We can write an article for our magazine and send it to another publication where they will set in their format without any additional work.

This also introduces the possibility of readers sending in articles for the magazine. All you need is an IBM or IBM compatible and a good word processing package like wordstar.

Articles need to be produced in unformatted form, and we will use Ventura to transform it into a finished format and output it to the laser printer.

Our PC boards are also being prepared on a computer using a Protel package and we have successfully produced the Organ board and Ignition Killer PC.

The completed files are sent to our PC board manufacturer on floppy disk where it is automatically plotted onto film.

The file is also converted into a drilling guide for a high-speed NC drill and within half and hour a drilled board is ready.

The photo plotter produces the track-work negative and component print and using these a single prototype can be produced via a photographic process.

The board is pre-coated with a laminate and when exposed to light through the negative mentioned above, the unexposed laminate can be washed off. From there it goes through the normal etching and tinning process and finally the component print is screen printed onto the top side.

Prototyping time has been reduced to 4 hours for a single sided board and 8 hours for a double-sided plated-through hole board.

We have liaisons with several PC board suppliers and if you want to produce a board for the magazine we can assist you with artwork design packages.

The only problem we are having is with drawing circuit diagrams. Paul is having trouble creating his own library using ORCAD and if anyone can help, please let us know.

As you may already be aware, we have opened up a Talking Electronics shop at 2 Central Ave, Moorabbin, to display all our products and provide a customer service that could not be provided in the cramped quarters at Rosewarne Ave.

Already, after only 2 weeks of opening and without any advertising, we are getting a number of customers who say they have heard about the shop. The word is spreading quickly and after this magazine hits the newsstands, things will really start to move.

Without blowing our trumpet, everyone knows we provide a service and range of kits that cannot be beaten.

After all, we have survived for 5 years, and increased sales each year, without ever spending a cent on advertising.

With the added space of the shop we will be able to increase our range of goods and include CB's, Radar Detectors, 240v inverters and antennas etc.

We also intend to directly import a number of lines and offer them at amazingly low prices.

Some devices are marked up 100% by the importers before we get them and its impossible to sell them at a resonable price.

The SAB 0600 is a typical example. It's a 3-note doorbell chip. It costs over $6 each when bought by the hundred and the equivalent from Hong Kong is 50c!

The same applies to panel meters, switches and so many hardware lines.

We have a member of staff going to HK in a month to start the importing business. I hope it works out.

The secret of our survival has always been the wide variety of lines. No one particular line has kept us solvent but a little dribble from 100's of different kits has kept us bouyant. We're making headway but only slowly.

With this, you are possibly surprised to see issue 15. Without constant pressure from Jim, the articles for this issue would still be sitting in the filing cabinet.

The conversion to computer type-setting was both very expensive and much more involved than we envisaged.

But with the setting-up done, we will be able to bring out the issues with increased regularity.

In this issue, Jim has shown the TEC is far from a limited design and via his 30 page article is showing how the TEC can be expanded.

If you have not yet bought the TEC computer, now is the time to buy a kit. Start with the notes in issue 10 and work through the experiments to issue 14. Then you can add the DAT board and Speech Board and see how versatile the computer is.

Many new products such as "beer ticket" vending machines, speech in lifts, speech for the disabled and alarm systems, are still using Z-80 processors. If you want to get into product design, the TEC is one of the best ways to do it. Try it, you won't be disappointed.

Jim is waiting to see the response to the TEC article. As long as readers are showing interest in the TEC, Jim is happy to stick around and share his knowledge with us.

Some of the planned articles for this issue have had to be held over due to the lack of space. While we would have liked to bring them to you, it is just not economical for us to add the extra pages. To add the extra pages, we would have to include advertising to cover the additional cost. This would mean we have less pages for our articles. A no-win situation!

Looking on the bright side, we have already several completed articles ready for issue 16.

NZ Readers

We have a very reliable representative in New Zealand. Write to: Trevor Cooper, 33 York St., Timaru, New Zealand. Phone: 83787. He stocks all our kits, books and fully constructed security devices.

We also have a number of shops in Aust. that buy large quantities of kits. See if your local electronics store is one of them.

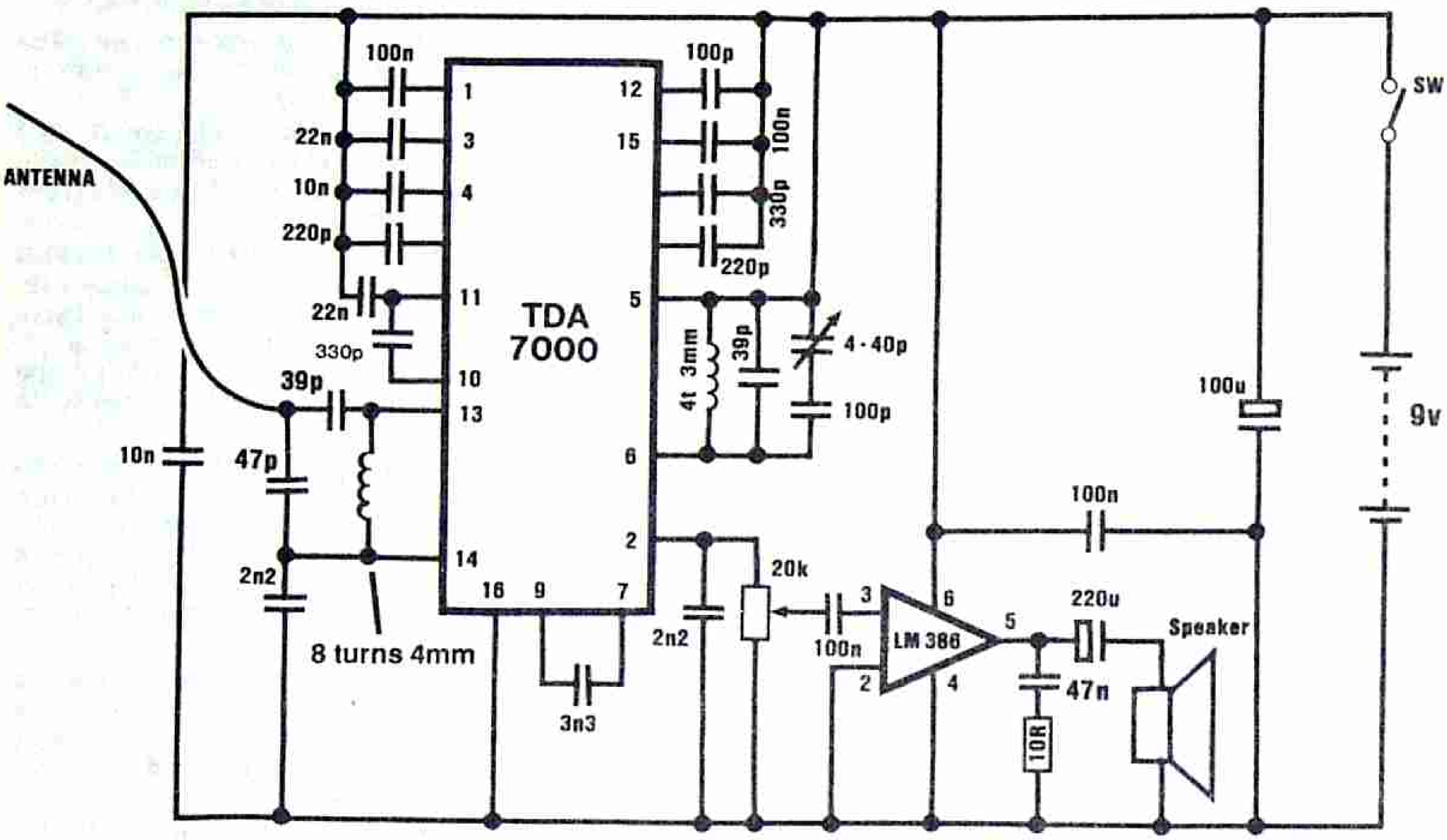
Don't forget, our very survival relies on the support we get from readers with the purchase of kits etc. See our re-designed order form in the centre pages for the kits in this issue. Pull-out the lift-out section for the "Shop" and you will find the complete litst of kits, including the "Earwig" (an Electronics Australia article) and Jim's package.

That's it for now, keep up the reading and building!

- Colin.

FM RADIO

A radio you can build into a cigarette pack



FM RADIO CIRCUIT

This FM Radio has been designed to complement our range of FM transmitters. With an FM bug and an FM radio, you can produce an FM link.

The receiver described in this article uses two chips and is small enough to fit into a case about the size of a pack of cigarettes.

Use can use either an earphone or small speaker as the output device and create a one-way link for a baby monitor etc. This will allow you to be in constant contact with a baby or sick child.

There are many uses for an FM link and I am sure this will satisfy your needs.

We have had many requests for such a link, from those who have built up one of our bugs and here's our fully-tested design.

We made a number of simpler FM radios but they did not work satisfactorily. We started with a synchrodyne design but found it too difficult to tune in the stations. Then we tried another design similar to reactance in AM reception but found the whole arrangement too fiddly and susceptible to drift when being handled.

When a dedicated chip such as the TDA 7000 came on the market, we had to try it.

The results were most impressive. Using only a short antenna we could pick up all the local stations.

As with all radio designs, the most difficult component to obtain is the tuning gang. We would have liked the proper thing but since no one manufactures FM radios in Australia, we have had to settle for second best.

It's an air trimmer with an added knob in the form of a 5mm LED to tune the range 88 - 108MHz.

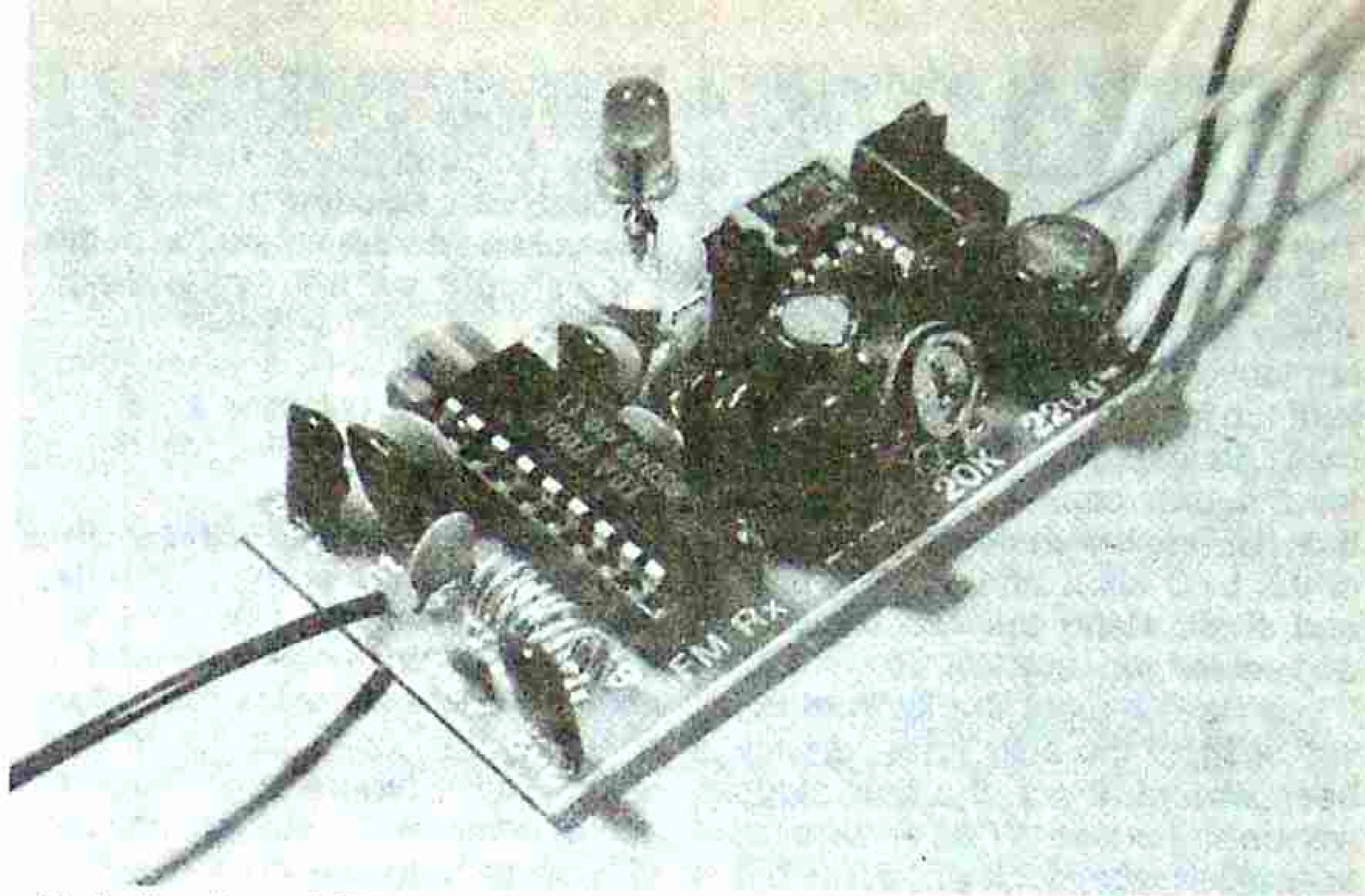
The advantage of making your own radio is the band can be broadened to 75 - 115MHz so that a private band can be created on either side of the commercial range. This is the area I suggest you choose as any transmission must not infringe on the commercial band.

PARTS LIST

1 - 10R 1/4 watt
1 - 20k trimpot
2 - 39p ceramic
1 - 47p ceramic
2 - 100p ceramic
2 - 220p ceramic
2 - 330p ceramic
2 - 2n2 greencap
2 - 3n3 greencap
2 - 10n greencap
1 - 22n greencap
1 - 47n greencap
1 - 100n greencap
3 - 100n monoblock
1 - 100u 16v electrolytic
1 - 220u 16v electrolytic
1 - 5 - 40p Philips trimcap
1 - 5mm LED for trimcap knob
1 - LM 386 Low Power Amp
1 - TDA 7000 FM Radio IC
1 - 8 pin IC socket
1 - 18 pin IC socket
1 - 4 turn 3mm coil
1 - 8 turn 4mm coil
1 - SPDT slide switch
1 - 8R speaker
1 - 9v battery snap
1.5m hookup wire for antenna

FM RECEIVER P.C. BOARD

At the low end of the band there are courier companies and at the high end, airport traffic. No matter where you settle, you must not sit on top or near any other operator as this will not only limit your range but your transmission will be picked up by others. After all, who wants to hear a baby crying over the radio!
This circuit offers an introduction into FM reception and you will be able to determine the critical nature of many of the components.
This project should not be made on your own layout as the critical nature of the tuned circuits will prevent the radio from operating. At least that's what we found.
Essentially the tuning capacitor and 4 turn coil must be kept close to each other and the chip for the tuned circuit to operate and you are advised to use our layout to be sure the circuit will work first go.
Once you have it working successfully, the oscillator coil can be adjusted to shift the band up or down. More about this later.

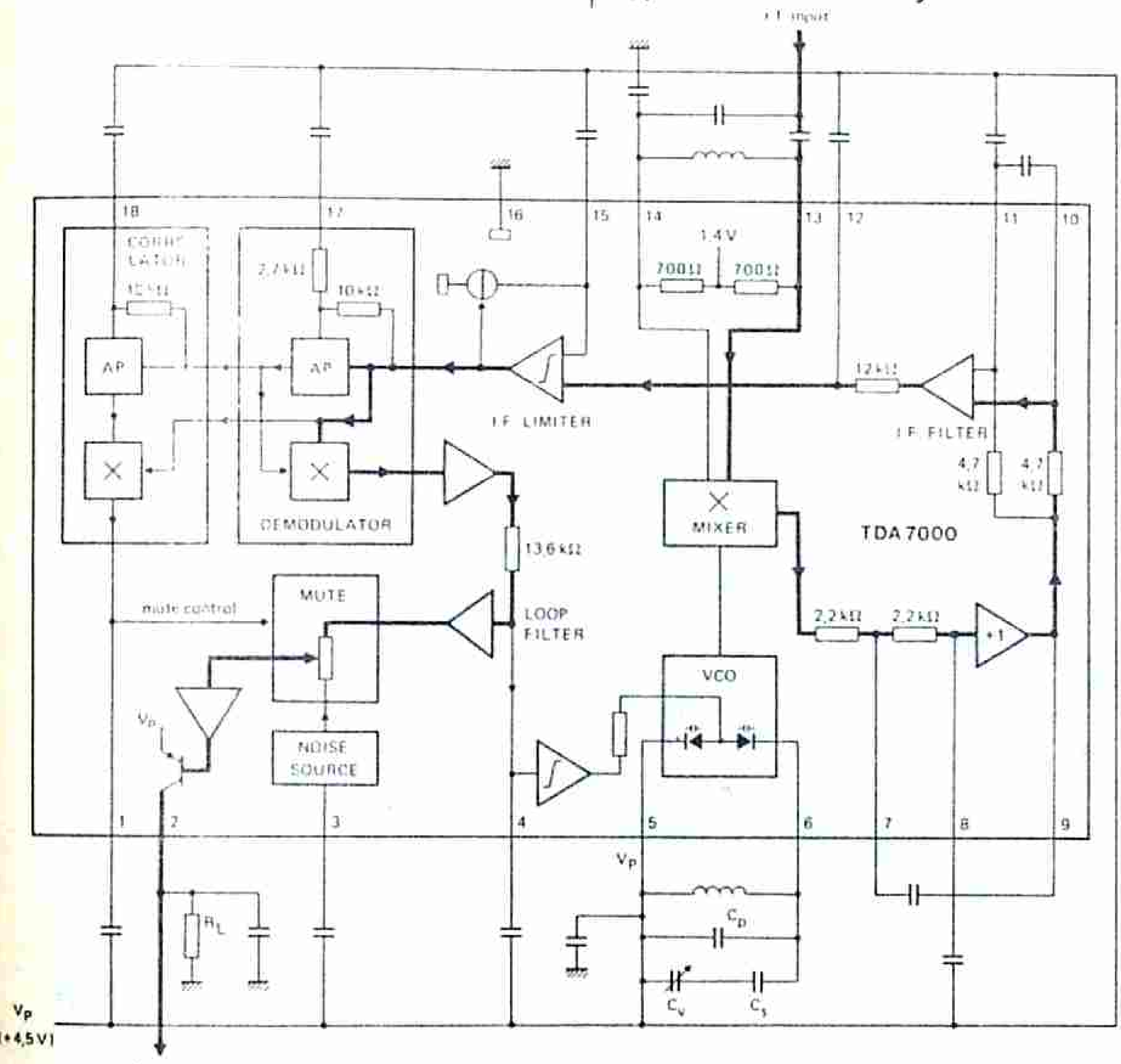


Note the 5mm LED soldered to the shaft of the air trimmer to act as a knob.

With this project, as with many of our designs, we have not suggested a case to house it in. Although there are many plastic boxes around, we have not settled on one particular type as the availability of chocolate boxes etc will vary from state to state.
We are always on the lookout for suitable boxes and sometimes the perfect answer pops up by chance. Both the Tic Tac box for the Amoeba and Livil box for the Ultima came as a passing suggestion from a customer. Maybe there's something along the same lines for the FM radio?

CONSTRUCTION

Place the contents of the kit on a clean part of the bench and check that everything has been supplied. If there are any notes in the kit they will explain any changes that have been made since the release of the magazine.
Inspect the board for any shorts or cracks (we have changed PC board manufacturers and our latest batch of boards are absolutely perfect. Not a single hole has been forgotten and no holes are filled with solder.)
The first component to be fitted is the 10R resistor. It is the only resistor in the kit and goes near the on-off switch.
The 8 pin and 18 pin IC sockets are fitted to the board so that the notch covers the half circuit on the overlay. This is important so the the chips can be fitted correctly at the end of assembly.
The capacitor and 2 electrolytics are next. If you have trouble working out their value, refer to some helpful hints on the back cover of the first issue of Talking Electronics.

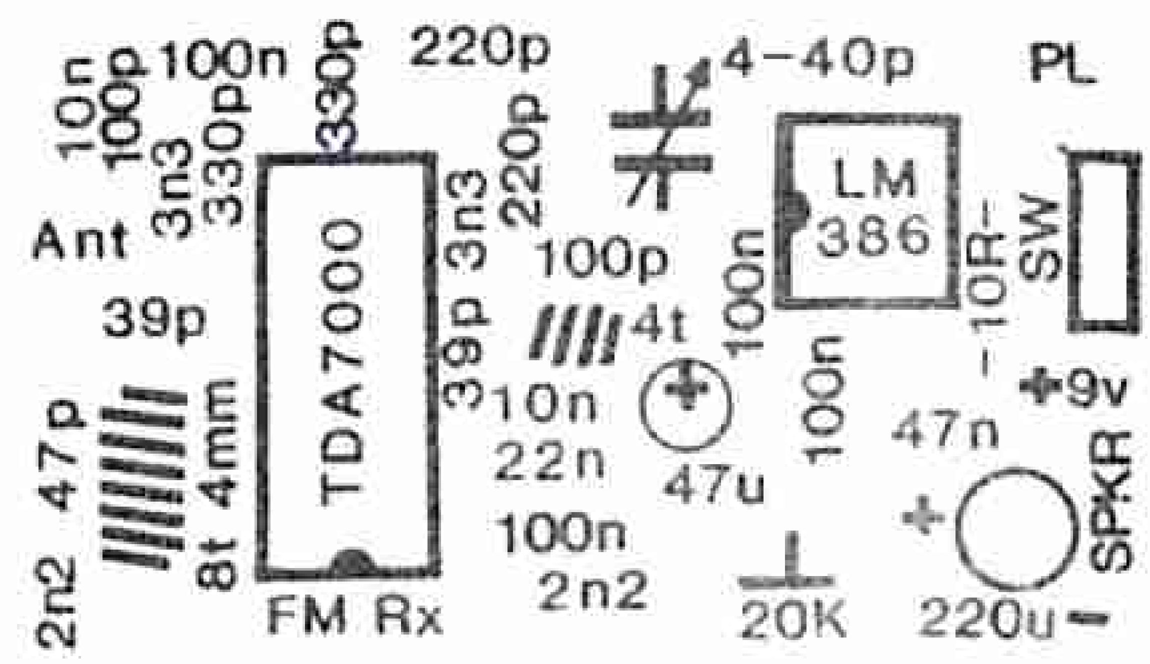


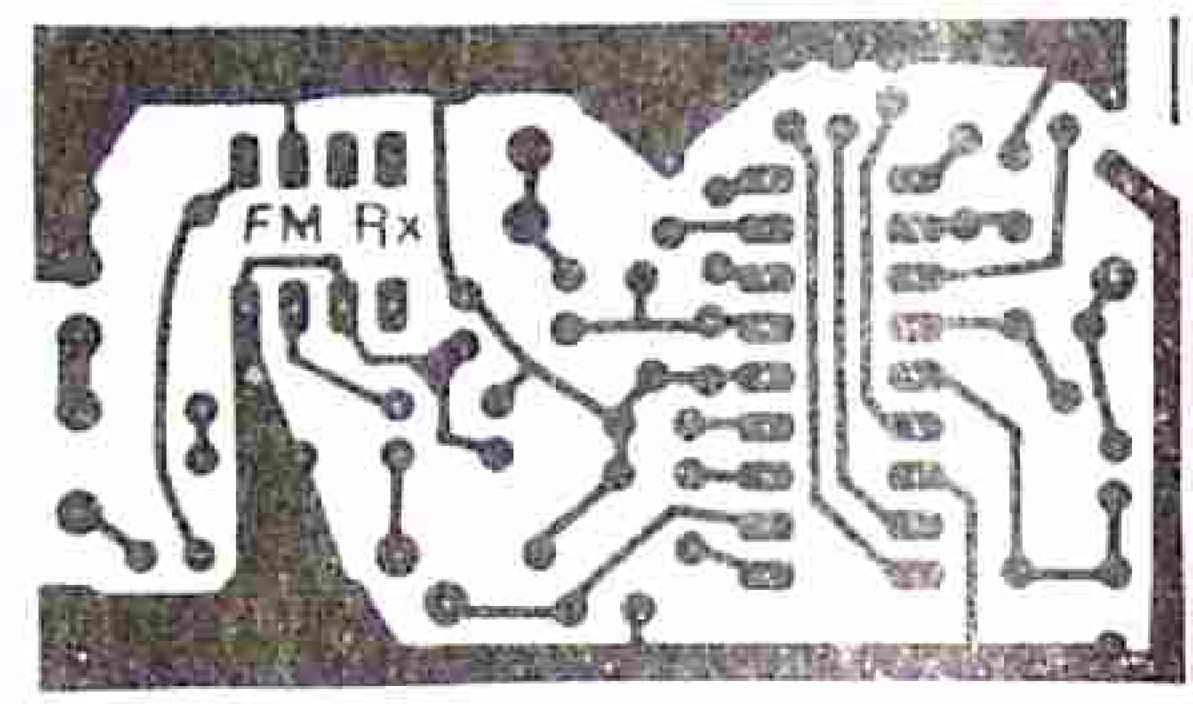
Block diagram of the TDA 7000 chip. The signal path through the chip is shown in heavy lines. The capacitors around the chip correspond to those in our circuit.

The electrolytics are polarised and should be placed on the board so that the positive lead goes down the hole marked with a "+" symbol.
The 20k mini trim pot and the trim cap fit on the board one way only. When soldering these in place, do not leave the soldering iron on the joint too long or the plastic parts will melt. An insulating handle is made for the trim cap by soldering the 5mm LED to the shaft. Cut the leads of the LED short and tin both LED and shaft. Bring the two together and solder very quickly.
If you have bought the kit from us, you will find the coils have already been wound. If not, the next steps will be important. Wind 4 turns of enamelled wire clockwise around a 3mm Philips screwdriver. Heat the ends of the coil with the soldering iron to remove the enamel and tin the ends. This will make it easier to solder it to the board.
Wind 8 turns of tinned copper wire clockwise around a 4mm Philips-head screwdriver and solder both coils to the board in the positions identified on the layout.
The antenna, switch, battery snap and speaker and the next items to be added. The speaker is fitted via two 10cm lengths of hook-up flex and the antenna is 100cm of hook-up flex. Paint one end of the slide switch with red nail polish to identify the "on" position.
Fit the two chips into their sockets and give the board one final look-over. If all is in order, connect a 9v battery and turn the trim cap to pick up a station.
To move the whole band higher, squeeze the turns of the 4 turn coil together. To move the band down, separate the turns slightly.

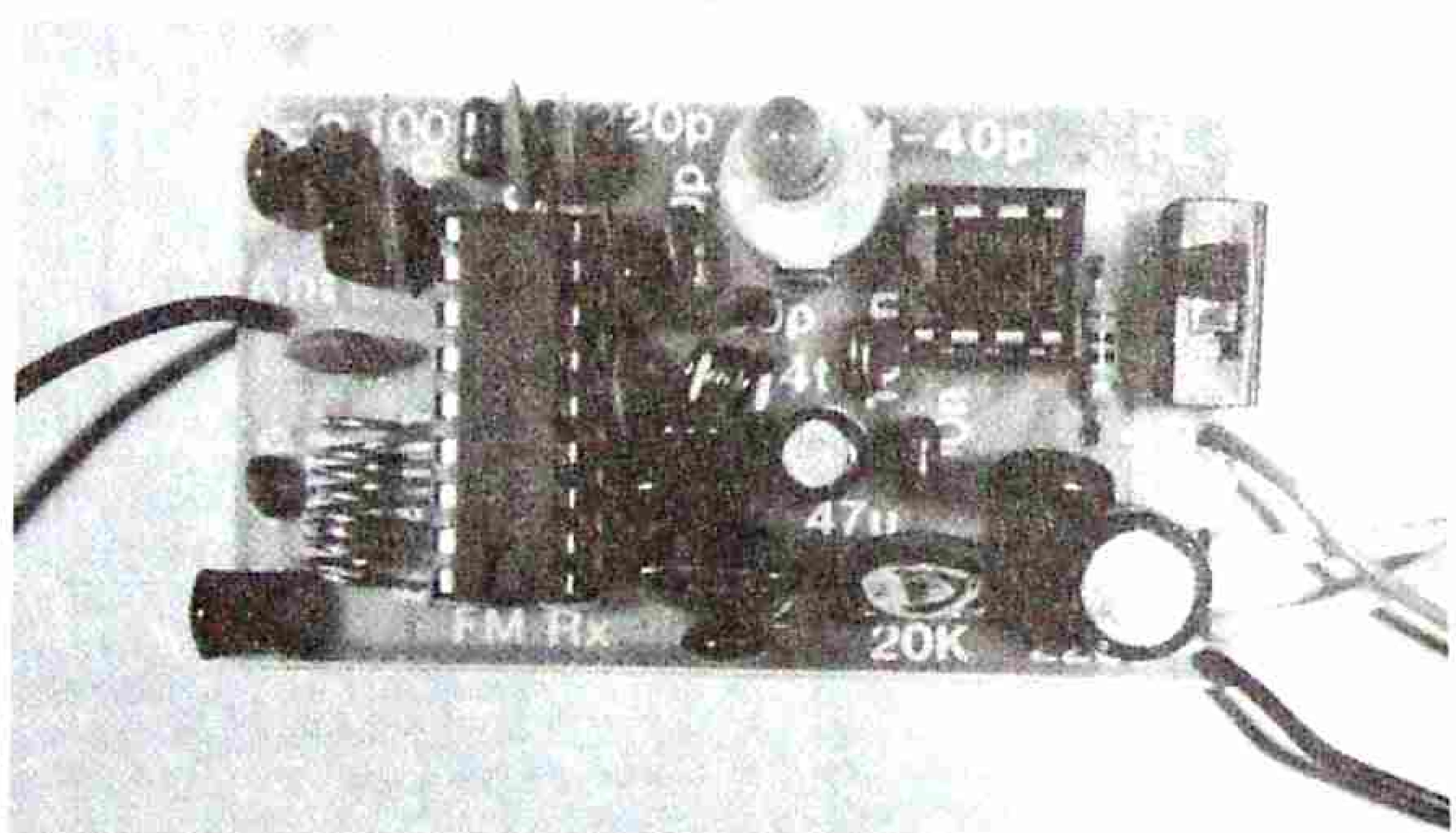
IF IT DOESN'T WORK

If the radio doesn't work, you will have to decide which section is at fault. There are basically two sections. The FM stage and the audio stage.
Turn the volume control to mid position and touch the centre leg with a screwdriver. You should hear a click or hum in the speaker. If nothing is heard, the audio section is at fault.
Measure the voltage across the chip (pins 6 & 4) and the current taken by the circuit (approx 50mA).
If these readings are correct, you should make sure the speaker is wired correctly and the output electrolytic is soldered in place.
The voltage on the output pin of the chip should be half rail voltage (4.5v).
If this does not solve the problem, the only other possibilities are the IC socket being faulty or the chip being damaged.
Make sure you do not connect the battery around the wrong way as the chips will be damaged instantly.
If a click is heard in the speaker, but no audio comes from the radio chip, check all the capacitors for correct value. These are the main external components for the chip as they are difficult to fabricate on the silicon substrate. If you pull the leads on a ceramic, or take too long when soldering, the connections inside the capacitor can come adrift. The only solution is to replace the capacitor.
Make sure the turns of the 8 turn coil are not touching each other. Measure the voltage across the chip (9v) and see that the chip has been inserted into the socket with all pins making contact.
There is really little else you can do without test equipment. The only trouble we had with our first prototype was the tuned circuit containing the 4 turn coil. The capacitor must be close to the coil and chip for it to work and it is important to use the coil as supplied in the kit to be sure the circuit works.
We will leave housing the radio to you. The antenna can be cut shorter if the radio stations are strong.
Try the radio with the Ultima and see if you get the same range as a commercial receiver. Let us know how you go.





Full-size artwork for those who produce their own boards. Remember, the layout is critical. Especially the tank circuit made up of the 4 turn coil and capacitor.



Plan view of the completed FM receiver showing the placement of all the parts on the board. Make sure the chips are inserted correctly as shown in the photo, with pin 1 of each chip covering the dot on the overlay

DRIGS

How many times have you flipped through a magazine in the vain hope of finding an interesting article? As you thumb through the pages, the hopes are dashed and finally you come to the last page.

There you find an article of untold wit and interest to make the purchase of the magazine worthwhile.

I hope this isn't the present case but now that you have reached the end I want to summarise all that's happened in the TE world and bring together the projects in this issue.

Firstly we have the CAR ALARM. It has a range of features to rival anything on the market and can be built for under $100. If you have a car worth saving, this alarm will keep your mind at rest.

The TEC article is quite large in this issue as a result of Jim's tireless effort. Some readers, not interested in the TEC, will think it goes on for too long. But one of the biggest criticisms of worthwhile projects in other magazines is the lack of back-up and support.

Generally the project extends over an issue or so and is never heard of again. All those who build it up are left high and dry with little understanding of its full potential.

Not so with the TEC. We have gone a full circle looking for other microprocessors, to rival the Z-80. But after spending thousands of dollars and hundreds of hours we have come to the conclusion that the Z-80 is the best (overall) and cheapest on the market.

With this we have no hesitation in bringing you pages of assistance in designing and developing programs in Z-80 code and it is our firm belief the the Z-80 will be around for years to come.

The two new projects for the TEC are the DAT board and Speech board.

The DAT board is a boon for programmers as it allows programs to be written and analysed one step at a time via a single stepper program.

The DAT board interfaces the TEC to a tape recorder to allow the storage of programs in a very convenient form.

The software to drive the DAT board and the tape interface is contained in Jim's new MONitor ROM called JMON.

JMON is the result of 9 months continuous refinement and hundreds of different versions have been created over that time. The end result is certainly a very good MONitor package.

Because JMON is a considerable advancement over MON 2, if you are building the computer from scratch you should start with MON 1B/2 and go through the experiments contained in issues 10, 11, 12, 13, and 14.

One essential add-on for the TEC is the NON VOLATILE RAM (issue 13) and if you would like to create a matrix of 64 pixels, the 8x8 is a must.

After these you can build the DAT BOARD and appreciate its wide range of capabilities.

In the TEC article we have supplied a game program called MAGIC SQUARE.

It can be typed in and played on the 8x8 display. MAGIC SQUARE is fully documented and it is hoped that you will appreciate how the routine works as much as you enjoy the game.

MAGIC SQUARE will have you baffled for hours. Once you work out how to get the square out, see it you can work out how the program works!

Once you have typed it in, it can then be saved on tape and recalled later.

If the TEC computer has grabbed you, a documentary package is available from Jim for $15.00 plus $2.50 postage. In this you get a full line-by-line explanation of how all the JMON routines work. As Jim put it, he hopes that you can understand the purpose of every instruction. Also if you purchase this package, you help Jim offset his costs on developing JMON and the DAT BOARD etc.

Drigs is the dregs!

There are also some other notes on programming in the package and will prove to be more beneficial than buying a $25 book on the Z-80 by a "bandwagon" publishing Co.

If you have really been bitten by the bug, you can buy a program tape with two TEC games, written by Cameron Sheppard. The first program is called MAZE and is played on the 8x8. It consists of a 27x30 playing field and the aim is to get out. This will keep you occupied for weeks!

The other game is "TEC invaders" and it's a bit like Space Invaders on a smaller scale.

These programs came as a result of co-operation between Jim and Cameron. It all started when Cameron came into TE some years ago with his TEC INVADERS. At the time the program was far to long to be published and the best efforts of Cameron and myself were unable to shorten it. One day Jim found the program and thought it would be a good program to put onto tape. Jim rang Cameron and they formed an agreement together.

Jim provided Cameron with a tape system and Cameron did some work required to add the finishing touch.

The rest is history!

To go over the TEC projects once again: DAT BOARD and PCB is $55.35 or $16.35 if you want just the tape and single stepper facilities without the LCD. Speech is $27.25, JMON is $16.00. Jim's package is $15.00, Cameron's tape is $6.50, Jim's EPROM programmer up-date is $2.30. These are all essential if you want to get into programming.

Next we have a beginners project (although the soldering requires a fair degree of skill). It's an Organ along the lines of a stylus organ and is great fun to play. It looks most impressive when built up and is ideal as a gift for the budding Beethoven.

A miniature FM radio has been a constant request from readers who have constructed one of our FM bugs. It's small enough to be hidden and allows you to create your own FM link.

And finally we have the FM bug that everyone's been waiting for. Our 1km bug, the ULTIMA. It's a sneak preview of out next bugging book "Security Devices." Once the word got out that it had been developed, we started selling kits! Now it's available for everyone and provided you are careful, you can experiment and achieve ranges of 1km and more.

The articles for Security Devices are nearly ready for final page-making and they should be going to the printer very soon.

Apart from the 1km bug and FM radio, we have included 6 other security-related projects to add to our range.

Many of these are not available on the market while others cost hundreds, if not thousands of dollars. (Take for instance the Pen bug. It sells for over $3,000 on the commercial market!)

You can save a fortune by building things yourself and at the same time, learn how its all done.

Look out for this book, as well as future issues of the magazine at your local newsagents or send for a subscription and be assured you don't miss a copy.

ISSUE 16

We hope to see both issue 16 and 17 out in '89. Issue 16 should not be too far off as we have have numerous articles left over from this one.

Jim is designing an expansion board for the TEC. The board was to be presented in this issue but it became clear that there just wasn't the room. Jim's board increases the memory by 20k. 8k of this is a battery backed RAM. There is an on-board "intelligent" EPROM programmer that when not being used to read and program EPROMS, can be used as 20 general input/output lines.

So we come to the end of another packed issue. So full that we didn't have any room for the adverts. Ah! Such is life. A magazine without advertising.

ASCII DISPLAY TABLE FOR LCD

Lower bit 4 bit \ Upper bit 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
××××0000	CG RAM (1)		0	@	P	`	p		ー	タ	ミ	α	p
××××0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
××××0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
××××0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
××××0100	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
××××0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
××××0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
××××0111	(8)	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
××××1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
××××1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
××××1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
××××1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	×	万
××××1100	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
××××1101	(6)	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
××××1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
××××1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█