ETI
ELECTRONICS TODAY INTERNATIONAL
THE ELECTRONICS, SCIENCE & TECHNOLOGY MONTHLY
February 1989 £1.50

NICAM Explained

Intelligent Plotter

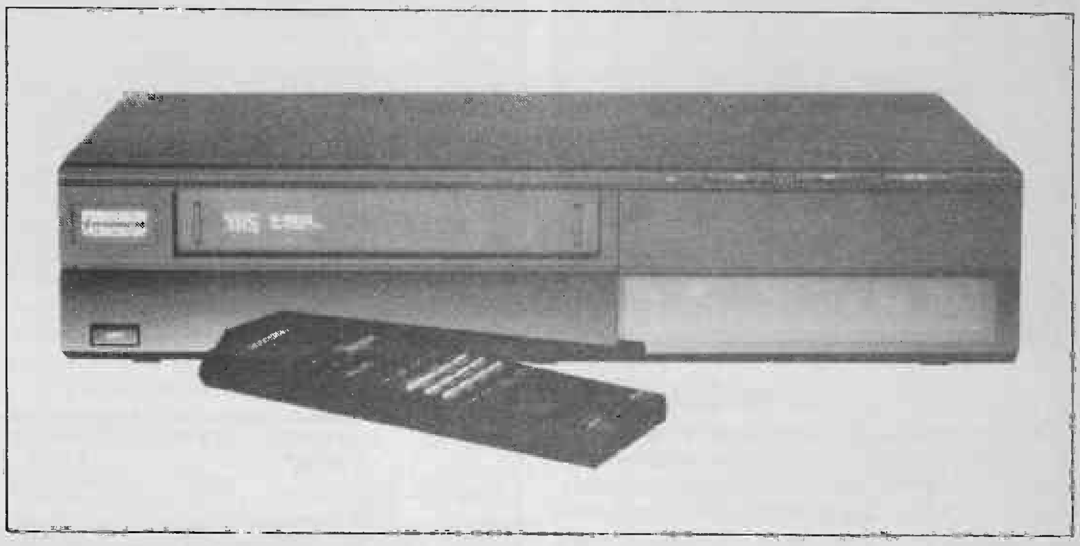
Ion Counter

Ultrasonic Pet Scarer

COMPRESSOR/LIMITER/GATE
Keeping quiet about music

AUDIO•COMPUTING•MUSIC•DOMESTIC

VOICES FROM THE OTHER SIDE



Ferguson's FV14T NICAM compatible VCR

John Linsley Hood examines the evolution and workings of the Nicam 728 stereo TV sound system

NICAM

Ever since the introduction of high quality 'stereo' VHF FM transmissions, there has been a wish on the part of the engineers working in the TV transmission field to be able to offer a comparably high quality sound system to accompany the 625-line colour TV picture.

The problems here are threefold. Any system has to be fully compatible with existing TV transmissions and TV receivers. Also the system has to slot into the existing 8MHz channel separation allocations, without degradation of the existing pictures — or sound — quality, and without being a nuisance to adjacent channels. Finally there is a reluctance of the TV manufacturers to make, and a similar unwillingness of Joe Public to buy, TV sets which even begin to do justice in terms of speaker size or amplifier performance to the quality of the existing mono TV sound signal, which is fully equal to that of the mono VHF FM transmissions.

Initial thinking on the part of the broadcasting engineers was to expand the existing TV FM sound signal to a stereo version using the well known 'Zenith-GE pilot tone' system used for VHF FM sound broadcasts, and I believe that experimental broadcasts took place in West Germany using such a system in the late 1970s. However, the advent of CD and digitally encoded sound signals encouraged the exploration of this type of signal as a possible option for providing an additional pair of TV sound channels which need not even be in the same language — a useful feature in bi-lingual communities such as Hong Kong.

The problems with digitally encoded signals is that they are very extravagant with broadcast bandwidth, and there is very little additional space to spare within the transmitted TV spectrum (shown in Fig. 1).

Bandwidth increases dramatically with each extra bit of resolution. It is generally acknowledged that 14-bit resolution is about the threshold needed for truly 'Hi-Fi' sound quality, though the existing BBC stereo sound system is only '13-bit' encoded for distribution from studio to transmitter. These broadcasts, nevertheless, seem to have quite a good reputation for audio quality among the 'Hi-Fi' fraternity.

The NICAM System

The solution proposed by the BBC was to use their NICAM 3 system for the compression of a 14-bit encoded sound signal from the studio into a 10-bit signal for transmission — already used by the BBC for their inter-transmitter signal distribution network — which would reduce the required broadcast bandwidth to a level which could be accommodated (just) within the channel separation allocations.

The NICAM system (Near Instantaneous Companding Audio Multiplex) takes the 14-bit input digital signal (2^{14} or 16384 step resolution) sampled at 32 kHz (which gives a maximum practicable HF bandwidth of about 14.5 kHz), and breaking it down into 1ms blocks, each containing 32 samples. Each block is then re-coded to 10-bit resolution (1024 steps), using a scale factor determined by the largest signal within that block.

The compression factor used, within one of five ranges, is then signalled by an added 3-bit group so that the correct amplitude range can be restored on decoding the digital signal. A parity bit is then added to each 10-bit sample to check the 6 most significant bits and allow the rejection of any group that has been corrupted by noise or other interference.

The TV Sound Channel

The existing TV sound system is shown in the block diagram of Fig. 2. In this, the signal from the tuner head is amplified by a common IF stage (usually with an IF passband of 33-41 MHz) and fed to a standard AM demodulator to extract the 'negative modulation' picture and 'sync' signals. The sound signal, which is frequency modulated, then appears as an HF sideband centred on 6MHz, and is amplified by a separate 6MHz IF stage and FM demodulator of conventional type.

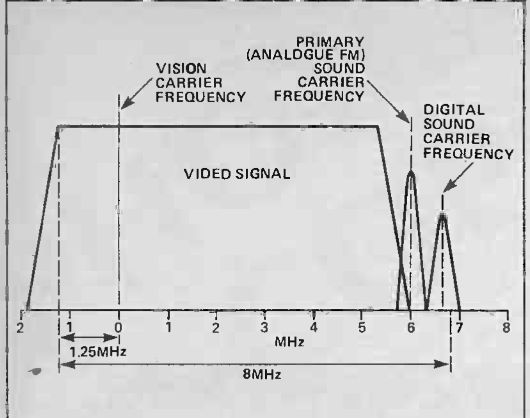


Fig. 1 Spectrum of the TV broadcast signal

This use of the difference frequency between the TV picture carrier and the sound modulation to generate the 6MHz 'inter-carrier' sound channel has the very great advantage that the sound IF frequency is generated by the transmission and not by the actual tuning of the TV set, which might otherwise be very critical and need a very accurate AFC to hold the set on tune.

With the NICAM system (called NICAM 728 because the digital data stream contains 728 kbits/sec) a separate, also frequency modulated — or more strictly phase modulated — signal is added at 6.552MHz above the video carrier, which has a bandwidth of about 720kHz (Fig. 1). To try to economise on transmitted bandwidth for this added digital signal, quadrature phase shift keying (QPSK) is used to modulate the signal, with each 2-bit data block represented by a shift in the phase of the signal as shown in Fig. 3.

Since this introduces some interference with the TV picture for its own and adjacent signals, the amplitude of this signal is set at −20dB (ref. the TV carrier) and the normal TV sound carrier is also reduced by 3dB to −10dB (ref. the vision carrier) to minimise the patterning produced by the interference between the sound signal and the 4.43MHz 'chrominance' carrier.

Advantages of NICAM 728

Although this is a very complex system and impracticable to implement without some custom designed ICs, the advantages of the added digitally encoded channel are sufficiently persuasive that, following approval by the DTI in September 1986, it has also been adopted by the European Broadcasting Union as the new TV stereo sound standard, as well as being considered or adopted by Hong Kong, New Zealand, Australia, and Scandinavia.

The advantages are that the bit stream can be divided into two quite separate channels so that in addition to providing a left and right stereo sound channel of precisely equal amplitude and very high stereo separation (dependant more on the studio or the subsequent receiver electronics than any shortcomings in the system), it can also provide two unrelated sound channels — even perhaps in different languages — or a stream of data of 704kbits/s for teletext or computer data, or a mono channel plus 325 kbits/s of data.

Additionally the error correction facility offered by the added parity bit allows a good quality, undistorted sound signal to be recovered, with low background noise even when the signal is so weak, or multi-path distortion is so bad, that the actual picture is barely readable.

There are of course some problems, quite apart from the circuit complexity. These stem partly from the digitally encoded signal, and partly from the nature of TV camera techniques.

The 'Hi-Fi' buffs who have listened to the stereo test signals broadcast from Crystal Palace and Wenvoe on NICAM 728 equipped stereo sound video recorders which have been creeping into the shops over the past six months or so have noted that the quantisation noise inherent in the digital signal can cause some degradation in sound quality, due to noise modulation or 'granularity', on the reproduction of the voice or single instruments (as compared with the existing FM TV sound signal heard through comparable audio equipment). With 'pop' music or other sound signals which generally involve high modulation levels, this minor defect is said to be undetectable.

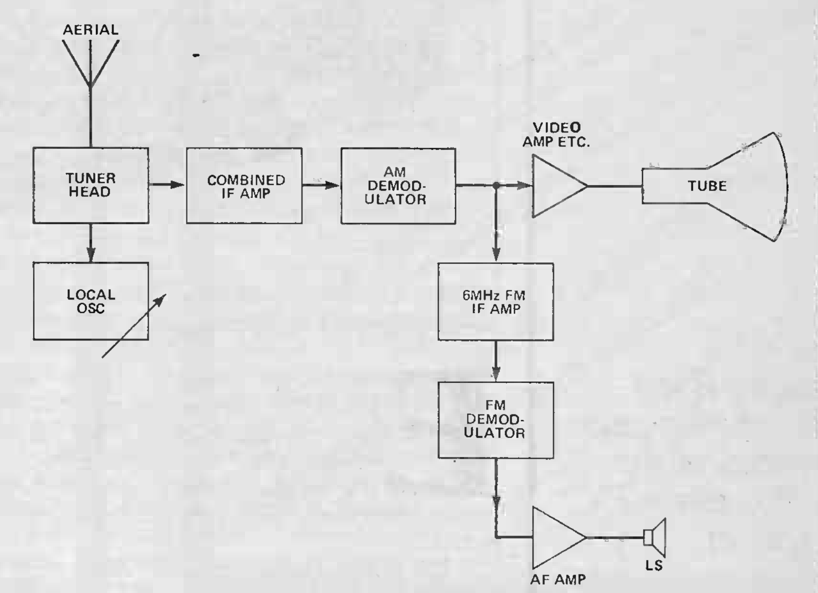


Fig. 2 Conventional TV sound system

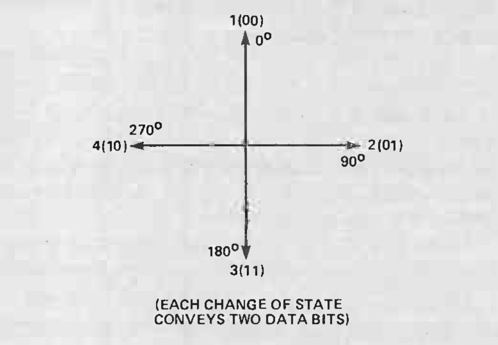


Fig. 3 Convention for quadrature phase shift keying (QPSK)

The second problem, which might be ironed out when the studio staffs become more familiar with the system, relates to the current practice of swapping camera positions around during the broadcast. With a mono sound signal no problem arises, but in stereo the sound image may frequently fail to correlate with the picture as seen.

As I mentioned above the system is complex, and as a 'linear' engineer I suspect that part of the complexity derives from the propensity of my 'digital' colleagues to mess things about just for the fun of it! The basic specification is contained in a draft standard jointly issued by the BBC and the IBA in September 1986, and this deals principally with the organisation of the recurrent 728-bit frame which is transmitted

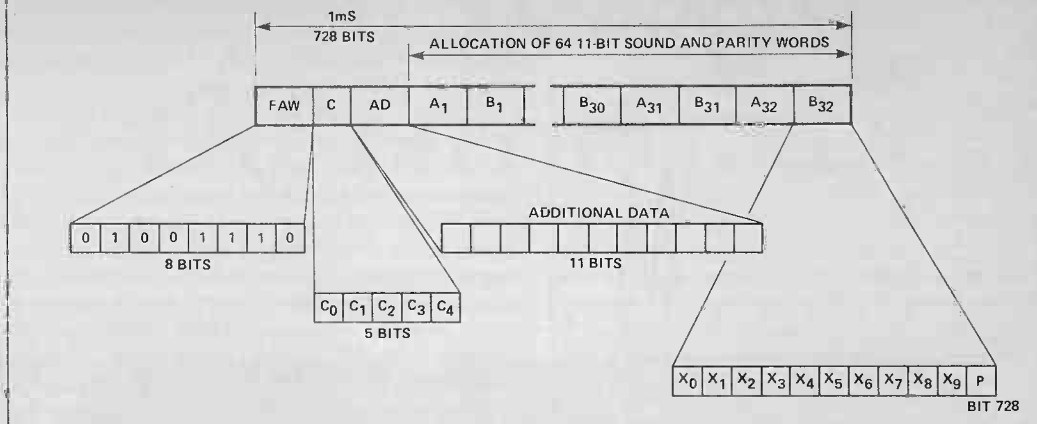


Fig. 4 Structure of 728-bit stereo sound frame

every millisecond.

Each of these frames has the format shown in Fig. 4 and is built up from an initial 8-bit frame alignment word (FAW), a 5-bit control information word, 11 additional data bits (for a purpose to be invented later), and a final group of 704 bits carrying the sound and parity bits, arranged as 64 × 11 bits, in which the L and R channels are sent alternately.

To prevent the possibility of a transmission error being carried over into successive data blocks (due perhaps to a protracted noise pulse) each 64-bit segment is interleaved with its neighbours to separate adjacent segments by 16 frames. Then to disperse the transmitted energy more uniformly across the additional sound carrier, the whole 704-bit word is scrambled using a pseudo-random number generator. This will also have the effect of lessening any patterning due to sound/vision interference.

So on reception, the frame alignment and control information words must first be interrogated to identify and isolate the frame, to de-scramble and then de-interlace the received 704-bit signal, to check whether it is a sound signal or data that is being received, and whether the digital signal is identical to the normal TV sound (which will allow a reversion to the FM sound channel in the event of the absence of a digital sound signal).

Finally when the data group is identified as a sound signal, the 3-bit scale factor command is used to restore the received 10-bit amplitude range to the original 14-bit resolution. The time delay involved in the temporary storage of received signal during this interrogation process is about 13ms and the degradation of the S/N ratio is about 3dB. This is unimportant for any single encode-decode process but can become noticeable if these steps are cascaded in the distribution chain feeding the transmitter.

The digital output from the NICAM demultiplexer stage — comprising the 704-bit sandwich of L and R channels — is then fed to a conventional DAC and audio amplifier. A block diagram of the layout of a typical stereo sound equipped TV is shown in Fig. 5.

From the description given above, it will be clear that the construction of a suitable NICAM 728 demultiplexer would be quite complicated if it were to be built using standard microprocessor and memory blocks. Since for commercial validity such things as TV sets must be inexpensive to construct, dedicated ICs will inevitably arise to fill this need, of which one of the first in the UK is the NICAM 728 demultiplexer IC produced as a joint exercise between Ferguson (Thorn-EMI) and Texas Instruments.

At the moment, details of this are available only on a 'non-disclosure' basis, which isn't a lot of use for technical writers. However other ICs from other sources will soon follow, and competitive pressures will lower the walls of secrecy.

When Will Stereo Sound TV Be Available?

Suitable receivers are already appearing and DIY ICs, one hopes, will follow. However, the BBC officially pleads that poverty prevents them from modifying their transmitters, or even using those they have modified, until 1991. The IBA are unlikely to wait for them and expect to commence a stereo service in London and Yorkshire this year. For the meantime, anyone interested in improving the sound quality of an existing TV could do a lot worse than make a receiver to capture the inter-carrier 6MHz FM signal and to feed that to a decent demodulator, AF amplifier and LS unit.

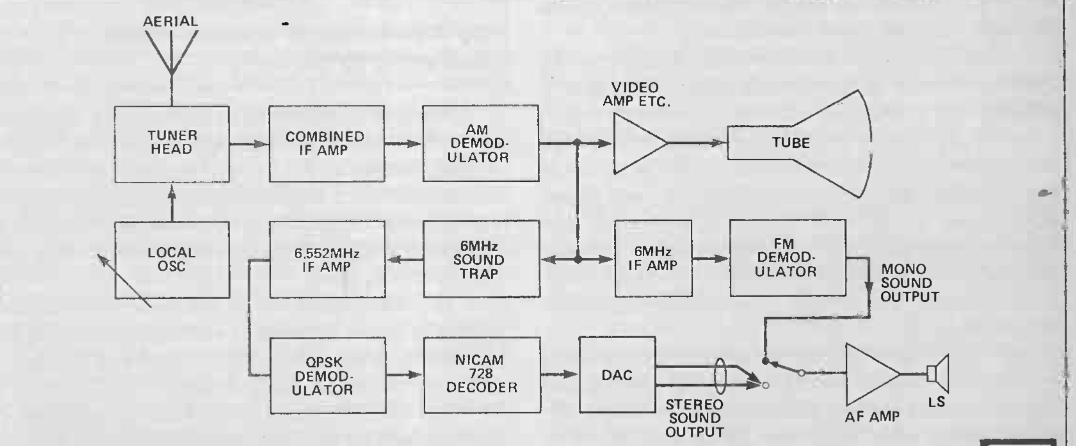


Fig. 5 Block diagram of Nicam 728 TV receiver

NICAM

LANGUAGE LESSON

Mike Barwise continues to survey the field of computer languages and this month looks at assemblers and compilers from the designer's viewpoint

Last month I discussed interpreters and now I want to go on to compilers. If you remember, an interpreter takes your human readable (machine meaningless) source or input code and at run time resolves each command into a call to a resident machine code routine.

Compilers do this at *compile time* and create a memory or disk file which consists of pure machine code. This is later called for execution. So at run time, a machine code program is executed, which eliminates the 'what do I do next?' questions of the interpreted languages. However, the real job of compilation involves quite a bit more sophistication than this description suggests. There are strategies used by different compilers (or by the same compiler for different jobs) which yield greater or less efficiency in the executable code.

The lowest level compiler strategy is that used by assemblers. At this level, each machine operation must be understood and utilised by the programmer to generate his routines. This is of course the position of maximal control if the programmer is good enough but the effort/performance ratio is very high.

Assemblers

Various advances in assembler design have led to new features, some of which are good and some not so good. Let's look at a few.

The first to appear was the process of 'linking'. The idea is that equipment-oriented and general purpose routines such as hardware drivers, stack handlers, interrupt control and so on would not be written from scratch by the programmer but would be available as pre-compiled machine code. This code would be written by the assembler creators so that it was relocatable (placeable anywhere in memory in relation to the user code) and would be included in the user program after this was written and debugged by a process of post-linking.

This means calling a special application program which adds the linked code and then scans it, amending any pointers or calls to the linked modules that it finds in the user code. The process was devised not so much as a user convenience but as a memory saver in the old days (20 years ago) when core memory was horrendously expensive. It allowed less memory to be used to compile larger programs.

In spite of the availability of masses of cheap memory these days, linking still persists as a reflex mechanism, although it is really redundant. Its only apparent advantage is that it reduces *compile* time but it is generally forgotten that the saving is lost as the total job consists of *compile plus link* time.

The next step forward was the 'macro'. At its simplest, a macro is just an ASCII string which is inserted in the source code when called by a short special instruction. At this level, it simply saves typing. As an example (in trusty 6502) a macro could be:

ROL; ROL; ROL; ROL;

which will place a set of four left rotates in the code whenever it is called. However, at this level, macros are not unduly useful. They do save typing but great care has to be taken to see that they begin and end at the right points not to confuse the system. They are really limited to the very simple blocks such as that in my example.

A major advance in macros was the development of the 'object oriented module.' This is effectively a black box (a device with a known function but concealed mechanism) with a specific set of input and output parameters. The object oriented module was not specifically designed as a type of assembler macro but it constituted the major breakthrough nevertheless. Now, to call a macro of this type, you would simply instruct:

FILL, start, bytes, value

where FILL is the name of the macro and start, bytes, and value are three parameters (the start address, extent and content of the memory area you wish to fill). You would not care what the actual code is or whether it was in-line code or a subroutine. You could, however, create routines yourself and use them as macros.

The ultimate assembler of this type would consist of a massive library of routines in source code, requiring the programmer to do little except generate a list of macros in due order. Such an assembler has, as far as I know, not been implemented. All assembler writers have baulked at the massive task of library generation.

The remaining major bugbear of the assemblers is the equation of variables and constants. At some stage in the creation of source code, every human readable (ASCII) reference to a memory address or memory content must be explicitly referenced to a binary (or hex) value.

All assemblers to date require the programmer to perform this task, normally in a ruddy great list at the start of the source code. This time consuming and error prone task is one of the main reasons for assemblers falling into disfavour as *high level compilers* have developed.

The final nail in the coffin of the assemblers from the point of view of the general programmer is the lack of a debug environment. There are systems (machine code monitors) which allow you to mark chosen points in your code ('breakpoints') and cause the code to pause and display status information but you still have to understand the detail of what is going on to make use of the displayed information.

High Level Compilers

High level compilers score heavily on these points. The programmer does not (apparently) need to understand the details of execution of the routines called. Variable and constant assignments are automatic and a high level test environment is provided.

CHIP IN

Let us look at these 'advantages' in more detail.

The strategies employed by high level compilers can be quite sophisticated: in C for example, executable code is installed either in-line or as a subroutine depending on the syntactic context. A PRINT instruction could be compiled as a call to a subroutine accompanied by parameters (what to print, how to print it), the routine being installed by the first call and then referred to by subsequent calls. However, a conditional operation such as IF . . . THEN would normally be compiled as in-line code each time it is encountered, due to the simplicity of the structure (a subroutine call would take as long as the routine itself to execute, doubling overall execution time).

Variable and constant assignments become 'declarations' in high level compilers. All labels must be listed together with the amount of memory allocated to them (byte, word and long word variables: 1, 2 and 4 bytes respectively) but the actual memory addresses where they are stored are assigned by the compiler and the programmer is not told about it.

This has its pros and cons. It makes programming more error free to someone unfamiliar with processor architecture but it makes the resultant code devilishly difficult to examine at machine level.

The final powerful feature of the high level compilers is the dedicated test environment. The compiler checks for mis-typed instructions, bad syntax, unassigned labels, duplicate labels and so on as it proceeds (which is also quite common among assemblers of better quality) but after compilation also provides the facility to single-step and trace execution in terms of the *input* code, displaying routine and user labels. This is a quick process compared with assembler tracing, as the integrity of each called function can be assumed, so it is unnecessary to visually examine their internal execution.

Conclusions

It would probably seem from the foregoing overview that high level compilation is the best choice for all programming. Wrong! It is the most cost-effective way to write software and so has become the order of the day for commercial purposes. However, the ascendancy of the high level compiler has itself curtailed the development of assemblers.

There is no reason why the advantages currently evident in high level compilers could not be incorporated into an advanced assembler — auto variable assignment, a very large library, an adequate test environment. The additional benefit would be the control available to an experienced programmer, allowing maximisation of software performance. The snag is that nobody has bothered.

The unfortunate result is progressively more and more cumbersome and inefficient software which demands increasingly powerful and memory intensive hardware just to perform adequately. Many high level compiler-generated PC graphics packages now demand 80286/80386 processors to run at an acceptable speed, whereas a clever assembler level programmer could probably produce similar results on the older 8086 generation.

The real problem is that commercial software writing is dedicated to the seller rather than the user of the resultant package — quick production in a high competition marketplace targeted to a non-critical buying public.

The time is long overdue for users to start griping and moaning. They *are* starting but there is a long way to go. The plain fact is that modern micro hardware could perform a whole lot better if the software writers used its power to the full, rather than relying on that power to compensate for their poor product.

ETI

OP-AMPS

Paul Chappell shows how you can learn all about the dark side of feedback without ever mentioning an op-amp

So far in the investigation of negative feedback we've seen only the good effects. Now comes the time to remove our rose coloured lenses and see what will really happen to our oven controller.

One implicit assumption I've been making all along is this: the heat output of the element depends only on the current flowing through it at any instant in time. Sounds reasonable if you say it quickly. So let's test it. What I want you to do is to turn on a heating ring on your electric cooker until it glows red. Then turn it off. There's no current flowing in the ring, so it's not giving out any heat, right? So you'll be happy to touch it? I think not!

When you turn on the ring at maximum current, it doesn't immediately start giving out full heat — you can probably rest your hand on it for a few seconds without coming to any harm and it may take a minute or more before it glows red. When you turn it off, it continues to give out heat for several minutes after the current has been removed. The reason is, of course, that it has to heat itself up before it can transfer heat to anything else and this gives it a built-in delay or time constant.

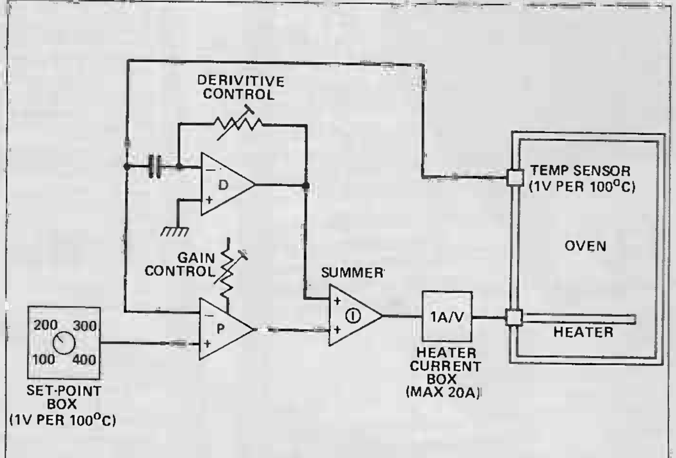


Fig. 1 The revised oven controller with varying proportional band

If everything in electronics or physics was instantaneous, we'd already be at the end of the story — you'd know how to make a control system or feedback loop as good as you could ever want it and we could pass on to something else. It's the delays and time constants that mess everything up and lead to the constant refining of control theory.

The practical consequences in our particular control system — the oven — you can imagine for yourself. With a narrow proportional band, the current is reduced a degree or so below the set-point but the heating element still contains enough heat to push the temperature up by perhaps 50°C. The temperature overshoots the mark and continues rising until the heater and oven are at the same temperature.

What happens next depends on how quickly the oven is losing heat to the air outside. If you're lucky the temperature might fall back within the proportional band gently enough for the controller to catch and stabilise it . If not, it may undershoot, then overshoot again, and so on.

One way to overcome the problem would be to use a very wide proportional band. As the current decreases, the rate of temperature rise will slow down well below the set point and if slowed down enough to take care of all the delays in the system, the controller will work again.

However (if you remember this far back) the idea of the control system was to get the temperature to set-point quickly, by applying maximum power for as long as possible and to stabilise it against disturbing influences, which worked best when the proportional band was narrow. By widening the prop-band we've almost eliminated these advantages, so we're back to square one.

If you think about it, what we really want is something to leave maximum power for as long as possible and to put the brakes on just before set point is reached. Given the limitations of the heater and oven, this would essentially mean taking off the power about 50°C below the set point so that the overshoot would become a rise towards the set temperature.

How to do it? It's no good putting the proportional band 50°C below set-point because the temperature would just fall back again. What we need is something to shift the proportional band downwards while the temperature is rising, then let it rise as the temperature change rate slows down. Sounds complicated but it's not so very hard in practice.

Figure 1 shows the revised circuit. The gain control on amplifier P (for 'proportional') varies the width of the proportional band, as before. Added to its output is a time derivitive signal from amplifier D. This signal becomes increasingly negative the faster the temperature is rising but will be zero if the temperature has stabilised.

Let's suppose we turn the set-point knob to 300°C and start up the oven. We'll say, for the sake of putting a few figures in, that the gain control on amplifier P is set to 100, giving a proportional band of 20°C. To avoid the overshoot from stored heat in the element, we want to start cutting off the current at, say, 250°C. The output from amplifier P at this point is 50V, so if we adjust the derivitive control to give −30V at the output of amplifier D at this temperature, this should do the trick.

If the output of amplifier D remained at −30V, the effect would simply be to push the proportional band downwards by 30°C. By 270° the heater current would be zero, the temperature would continue rising to 300°C or so and then would fall back to around 260°C. But this doesn't happen because as the temperature rise slows down, so the output of amplifier D rises until at the peak of overshoot, when the temperature is not changing at all, its output will be zero. The trick is to adjust the P and D controls so that the peak of what would have been an overshoot actually coincides with the set-point.

In many control situations, a few rough calculations are made to choose fairly sensible initial settings for the controls, then the final adjustments are made by trial and error. You can get anything from an over-damped response to oscillation by fiddling with the pots!

Just to complete the picture, there's one more major fault with the control system. The temperature always stabilises below the set-point. By narrowing the proportional band we can make it end up as close as we like to the set point but narrowing the proportional band too much makes the system prone to oscillation. The usual solution to this little niggle is shown in Fig. 2.

Here we have yet another dollop of voltage being

CIRCUIT THEORY

fed into the summer, this time from an amplifier which integrates the difference between the set point and the actual temperature. Suppose that the set point is 300°C and that the temperature, by the action of the P and D sections of the circuit, has settled at 290°C. The integrator will have 10mV across its input so the output will slowly rise as the capacitor charges.

The rise in voltage will cause a slight increase in the heater current, causing a gradual increase in the oven temperature. This reduces the current from the proportional amplifier and also causes the integrators output to rise more slowly, since the voltage between its terminals will have decreased.

If you think about it, you'll see that the circuit will end up exactly at the set-point. The output of amplifier I will now be steady because its inputs are both at the same voltage, the outputs of amps P and D will both be zero — the first because the set point and temperature voltages are equal and the second because the temperature is not changing — so the entire voltage to sustain the fixed temperature is supplied by amp I.

My description of the operation is obviously over-simplified. The integrator capacitor will be charging up the whole time the temperature is below set-point and will have an unsuitable voltage on it initially. The derivitive amplifier will also want to have its say as the temperature is varied by the integral amp. The purpose of each part, though, is as I've described it and with careful setting up of the controls it can do just what its supposed to do.

If you were to build a practical version of the control part of this circuit, and put it in a nice box with digital displays, you'd have made something instantly recognisable in industry as a PID, or three-term, controller. A number of manufacturers sell them as general-purpose control systems — anything from a few hundred pounds upwards, if you want one. They can cope with 95% of the control requirements in industry, laboratories or wherever the need arises. The circuits are, I should say, a lot more refined than Fig. 2 but my diagram has all the essential ingredients.

Back To Op-amps

We've strayed rather a long way from the supposed topic of these articles — op-amps. My aim was to show you that negative feedback is not something to do with a pair of resistors around an IC — it makes itself felt all over the place.

If you've got an intuitive notion of what its all about and can see that its not necessarily a good thing in all circumstances, then its all been worthwhile! I originally intended to give you a Cooks tour of positional control (servos and the like) too, but we've spent enough time on digressions. In another article, perhaps.

I usually end by telling you where I intend to lead you the following month — and always regret having done it when I come to write the article and find my enthusiasms are elsewhere. This time I'll avoid all that. Next month's article will be something about op-amps, probably, and to do with electronics almost certainly. More than that I cannot say. See you then!

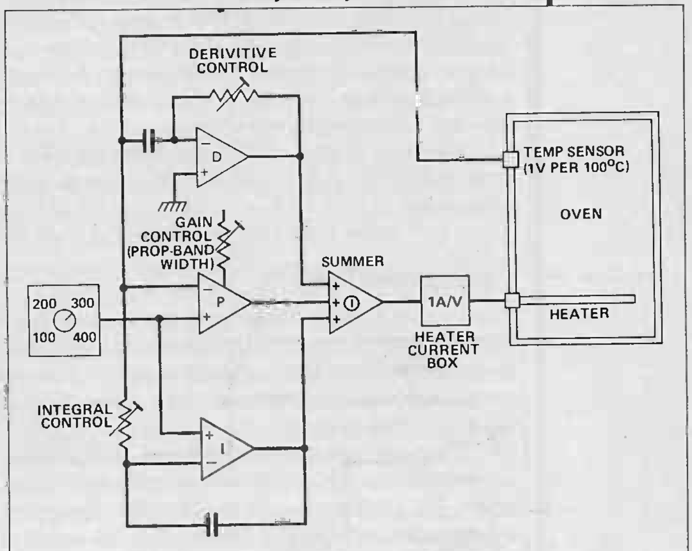


Fig. 2 Simplified diagram of a PID controller

A range of commercial PID controllers from Control and Readout

COMPRESSOR LIMITER GATE

Ian Harvey squeezes a high quality compressor/limiter and a noise gate into the same box all ready for the stage or studio

PROJECT

Go into any recording studio and alongside the racks of flash MIDI samplers, digital reverbs and PCM recorders you will find the humble end of effects units — compressors and gates.

The effectiveness of these units in producing a polished sound is often overlooked and many people opt to spend their money on trendier hardware — hardly surprising as a decent compressor will set you back £200 or more. So here is the DIY answer — a flexible compressor/limiter good enough for semi-professional use, which includes a completely separate noise gate for good measure.

Total cost, including a decent 19in rack case, is about £60 — roughly the price of one of those nasty footpedals . . .

Compression

Essentially a compressor reduces the dynamics of a signal by turning down the loud bits and turning up the quiet bits (Fig.1) Extreme compression, where the output level is actually constant for a range of input levels, is called limiting.

A popular and fairly 'musical' way of varying the degree of compression is to have a 'soft knee' characteristic (Fig.2) where the compression becomes more severe as the input level is turned up. This design is a soft-knee one but has a separate *Compression* control to allow flexibility with input levels.

An immediate problem is that if a loud signal (like music) is attenuated and a quiet signal (like hiss) is amplified, the signal-to-noise ratio will inevitably be made worse. Guitar-type footpedals usually have a problem with noise anyway, and this factor often makes them unusable for serious recording work.

Care has been taken to keep the noise generated by this design to an absolute minimum but as most input signal sources will not be perfect anyway a basic noise-gate is included to shut off the output when the signal falls below a given level. The *Threshold* and *Gate Time* controls set the triggering level and opening time of this gate.

Finally the compressor has *Attack* and *Decay* controls which determine how quickly it responds to the input level rising and falling. Generally these will be set to reflect the attack and decay of the signal. A 'fast' signal like percussion needs to be accurately and quickly followed, whereas a 'slow' signal like vocals should receive smooth compression to avoid 'pumping' or 'breathing' effects.

Gating

Fortunately, a gate is a lot simpler. Basically it is a switch which only allows an audio signal through

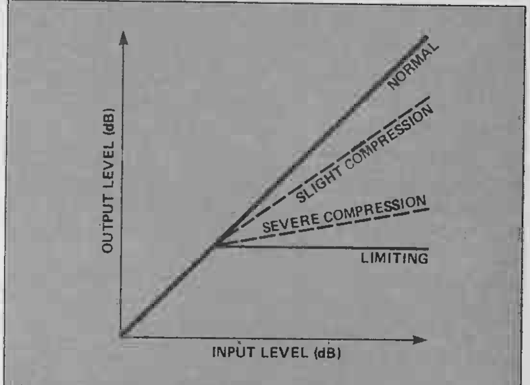
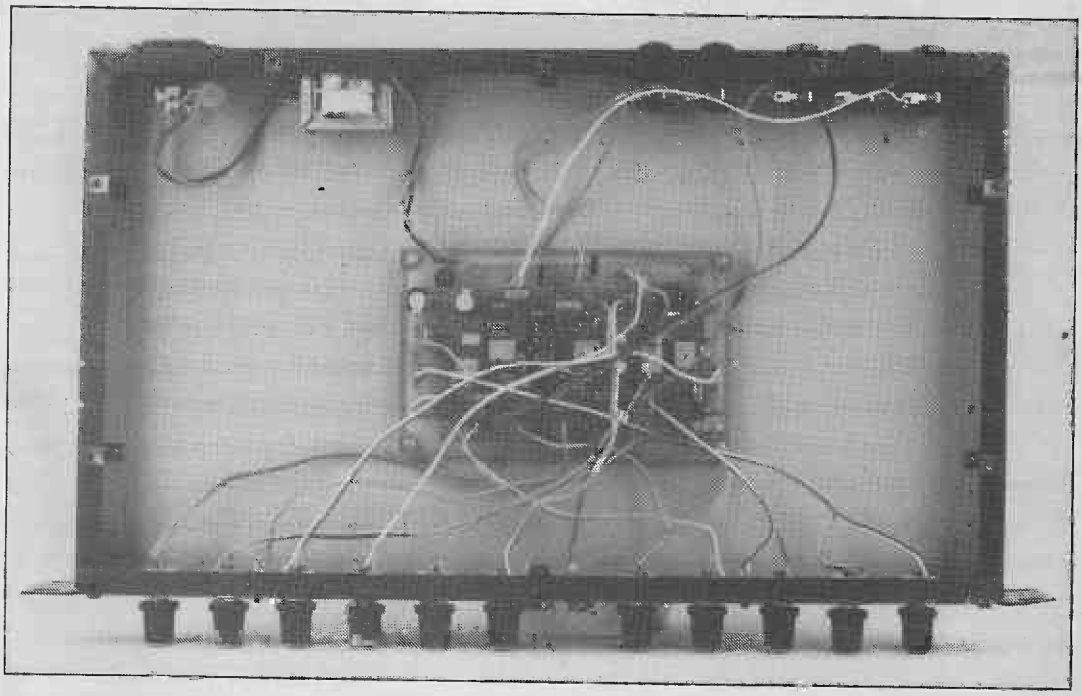


Fig.1 Degrees of compression



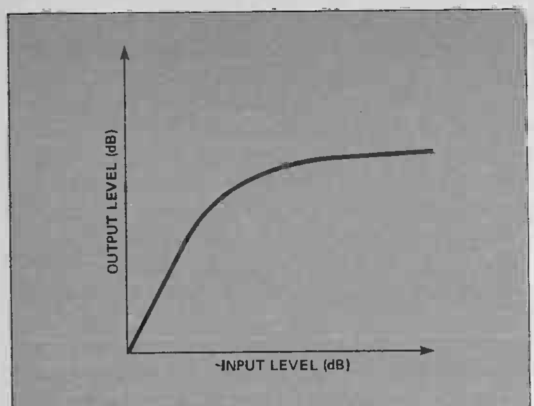


Fig.2 Soft-knee compression

when 'triggered'. Triggering occurs when the signal goes above a level set by the *Threshold* control. The speed with which the gate opens and closes is set by *Attack* and *Decay* controls.

With some 'ragged' input signals the gate can be triggered several times in quick succession on the start of a note. This usually sounds naff so to avoid it a *Hold Time* control is provided which holds the gate open for a certain minimum time once triggered.

Finally an external input socket is provided so the gate can be triggered with a different signal to that being gated.

Construction

Building the unit is relatively straightforward with no non-standard (or at least non-Maplin) parts. Setting-up requires a voltmeter and some form of signal source and listening equipment. All components except the mains transformer and front panel controls. (Fig.6) mount on one PCB (Fig.5).

Start assembly of the board with the links, resistors and diodes, then the IC sockets, PCB pins for the off-board connections, capacitors and finally IC8 and IC9 (these do not require heatsinking). The PCB, T1 and front panel controls can then be mounted in the chosen case, and lastly everything is wired together.

CW, CCW and W on the overlay diagram refer to the clockwise, counterclockwise and wiper connections to the pots. The input jacks SK1 and SK3 are wired directly to the RV1 and RV7 respectively, signal to the CW end and ground to the CCW end. The CCW and W terminals are then wired to the PCB pins. Do not put any ICs in their sockets at this stage.

A few points: Using an LM833 (or NE5532) for IC1 will give a noticeably better noise performance than the usual TL072 types — definitely worth the extra 50p or so. Don't be tempted to replace C8, C9, C15 or C16 with electrolytic types.

On the prototype, the PCB was mounted on metal spacers so as to connect signal ground (the track around the fixing holes) to the case as in Fig.7a. Mains earth was left unconnected (it should never be directly connected to signal ground), so for safety reasons the mains transformer had to be well isolated from the case. If for any reason isolation is not possible, connect the case to mains earth and mount the PCB on insulating spacers. With either system the signal ground can then be linked to mains earth via a 100R resistor (Fig.7b).

Setting Up

Before inserting any ICs, switch on and check the power LED lights and there are the correct supply voltages present at the IC sockets. These should be +12V to IC1 pin 8, IC2 pin 11, IC3,4 pin 7, IC5 pin 8, IC6,7 pin 7 and −12V to IC1 pin 4, IC2 pin 6, IC3,5,6, pin 4.

If all is well, turn off and insert the ICs. (Note IC3 is oriented opposite to all the others). Set all the front panel controls to minimum (anticlockwise) and internally set PR2 and PR3 fully clockwise and PR1 and PR4 midway. Connect a voltmeter across R21, switch on and adjust PR2 until the meter reads approximately 1.6V.

Now connect a signal source to the compressor input and an amplifier to the output. Turn up the compressor input level. The signal should be audible at the output and LED1 should light. Turn the *Threshold* control on the compressor right up until LED1 goes out (reduce the input level if necessary). Adjust PR3 until the output signal is just inaudible.

If the *Threshold* control is now turned down LED1 will relight and the output will reappear, probably with a noticeable click. PR1 is to be adjusted until this click disappears. To do this it is best to use music with a regular beat as the input and adjust the *Threshold* until LED1 only triggers on the beat. Keep the input level low so as to hear the click better.

This last step needs to be repeated for the Gate channel. Connect the music to the Ext Trigger input, leave the main input unconnected and monitor the output. Adjust the *Gate Threshold* until LED2 triggers on the beat and tweak PR4 until the corresponding clicks (this time without the music) at the output disappear. Put the music through the main input (just to check it's working OK) and then you're ready for action!

Applications

The unit is designed mainly for line-level signals, although high-output guitars can be used with success. After plumbing it in, the input level controls should be set so that clipping is just avoided. Then the desired degree of compression can be set with the *Compression* control.

The compressor can be used on almost any signal whilst recording. By keeping the signal level within a restricted range, it will give a better signal-to-noise ratio on tape and when mixing down, it will help preserve the balance — no more accidentally louder-than-average guitar notes sticking out like a sore thumb.

Compression may be used to add sustain to guitar and bass — usually a fast attack and medium fast decay are a good starting point. The short attack time will reduce the transients at the starts of notes, which is useful on bass but is often not desirable on guitar, especially acoustic. Increasing the attack time will allow these transients through without being

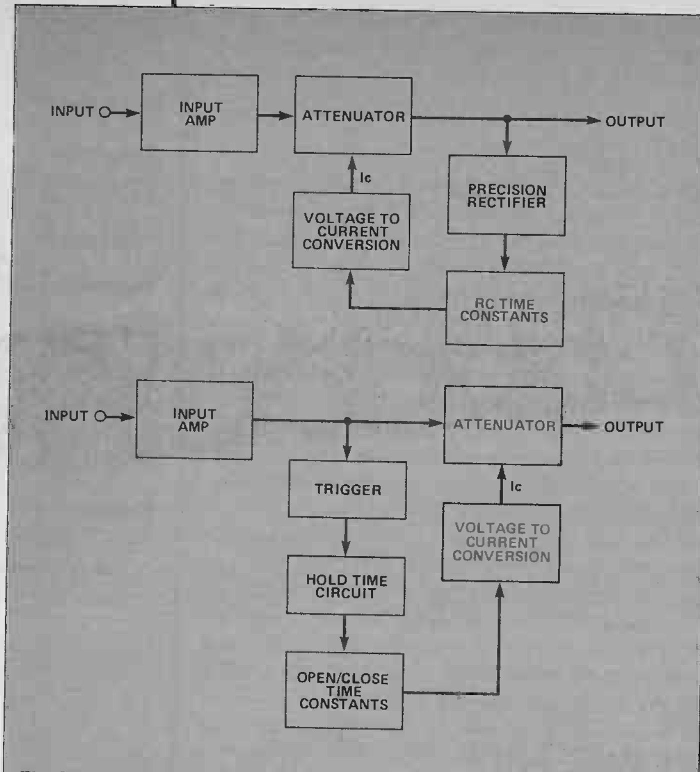


Fig.3 Block diagram of the compressor/limiter/gate

compressed.

For use on percussion, fast attack and decay times are needed. Kick drums and snares will have sustain and 'body' added. The *Gate Time* can be increased to give a soft start to the beat.

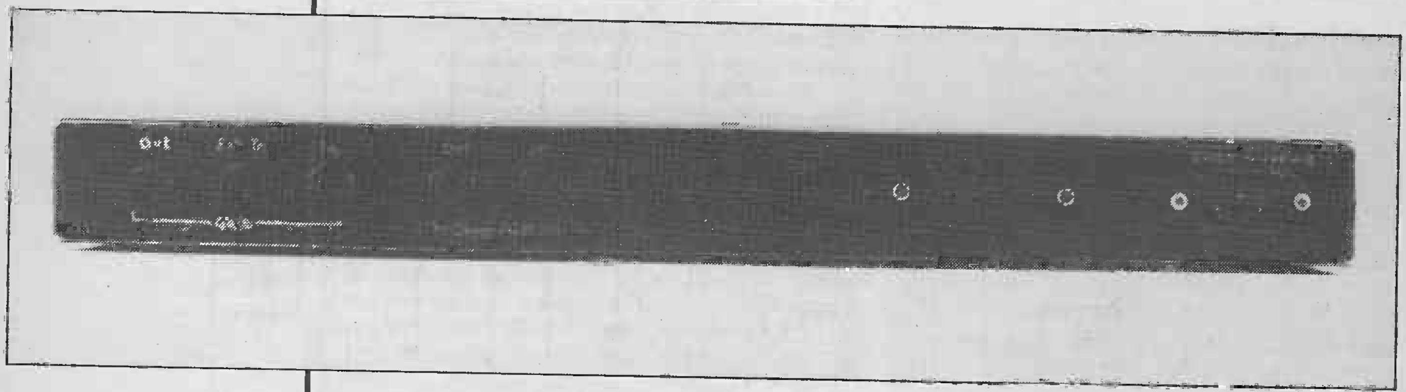
Vocalists with a tendency for wild crescendos (most of them) benefit from gentle compression to keep an even level on tape. Slowish *Attack* and *Decay* times are best and the degree of *Compression* should not be too severe.

The major application for gating is as a simple noise gate. The gate is set to trigger when any signal appears and just shuts off in the gaps, reducing the apparent noise. Don't expect a noise-gate to solve all your hiss problems though. You will still have the hiss when the signal is present and if the signal has no high frequency components it will still be audible.

It is often more distracting having hiss appearing and disappearing all the time than a constant background level. When miking a drum kit, gating can be used to stop spillage of the hi-hats onto snare or toms mikes.

A gate can also be used to reshape notes. Using a slow *Attack* time can make a guitar sound like a violin or drums sound backwards and is equally good on fretless bass. Altenatively using a fast *Attack* and *Decay* time and setting the triggering *Threshold* higher than normal will remove fret buzzes, string squeaks and so on.

Lastly there are many creative possibilities



HOW IT WORKS

The unit is based around an LM13700 VCA chip with two separate control blocks – one for the noise-gate function and the other for the actual compression. See Fig.3 for a simplified diagram. The input amplifier is provided to match the unit to the wide range of levels typical in home recording set-ups, and to give a low-impedance drive to the VCA.

The circuit diagram is shown in Fig.4.

The VCA itself is a fairly conventional LM13700 circuit but with additional treble pre-emphasis and de-emphasis capacitors C2 and C3. These bring the noise level (especially at subjectively more annoying high frequencies) to an acceptable level. Note that the inputs and outputs to the 13700 are actually current sources and sinks, so the combination C2/R4 and C3/R8 is all that is needed to give a theoretically flat overall frequency response.

The noise-gate control block based around IC3 and IC4 is indentical in operation to that in the Gate channel, with the exception that certain parameters are not variable. Hold Time is fixed at 100ms by C7/R16 and the Attack and Decay controls are replaced by RV3 and D3 which fixes the decay time at 470ms and the attack time is adjustable 0-470ms.

The compression section uses a precision rectifier and smoothing to convert the output signal to a DC voltage and a block to reduce the gain of the amplifier as this voltage increases. The precision rectifier is based around IC5 and produces a negative output via diodes D4 and D5 to charge smoothing capacitor C9. The diodes ensure that C9 charges via RV4 and discharges via RV5.

The control voltage is applied to the gate of FET Q2, whose source terminal is held at a constant voltage by Q1. As the output level increases this voltage becomes more negative and cuts off Q2's drain current. The non-linear transfer function of Q2 give the compressor its 'soft-knee' characteristics. Limiting will occur as the output signal (and the control voltage) nears the cut-off voltage of the FET.

As there is quite a spread of parameters between different 2N3819's, the source voltage of Q2 must be adjustable to give a known drain current under no signal conditions – when its gate is at 0V. PR2 is use to set this to about 0.75mA. R21 is used simply to allow this current to be measured without disturbing the circuit.

The output of the noise-gate block on C8 is buffered by Q3 and fed to the base of Q1 via PR3 and R20. When the gate is untriggered C8 is charged to +12V and Q1's base voltage rises. This increases Q2's source voltage and cuts off the drain current. PR3 controls the amount of cut-off and is normally set to give complete attenuation.

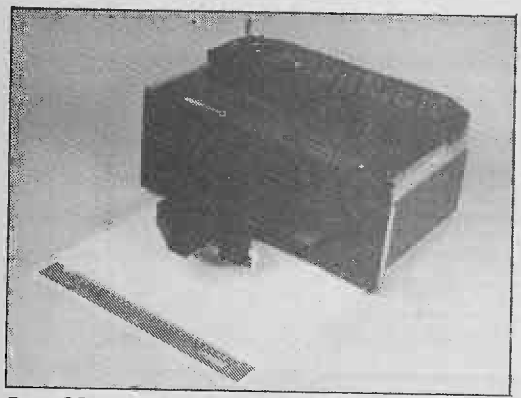
Fig 3 shows the block diagram of the gate. The input amp and VCA are similar to the compressor. IC6 is used as a comparator and when the input signal goes above the trigger level set by RV8, the output swings to +12V and charges C15. Together with RV9 and Schmitt trigger IC7, this forms a monostable used to set the gate hold time.

IC7's output swings between +12V (idle) and 0V (triggered), charging and discharging C16 via the attack and decay controls RV10 and RV11. This sets the control current to the LM13700 via Q4 and R44 – the maximum current being set by R44 as 0.75mA approximately.

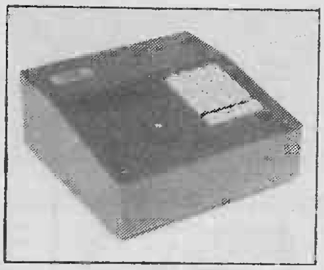
February 1989

VOLUME 18 No 2

Geoff Bains: Editor
Jez Ford: Assistant Editor
Paul Chappell: Projects Editor
Jerry Fowler: Technical Illustrator
Heather Hopkinson: Design
Mark Warford: Photography
Selina Gayle: Ad Manager
Heather Wust: Classified Sales
Andrew Selwood: Copy Control
Mark Webb: Group Editor



Page 32



Page 46

Page 38

ISSN 0142-7229

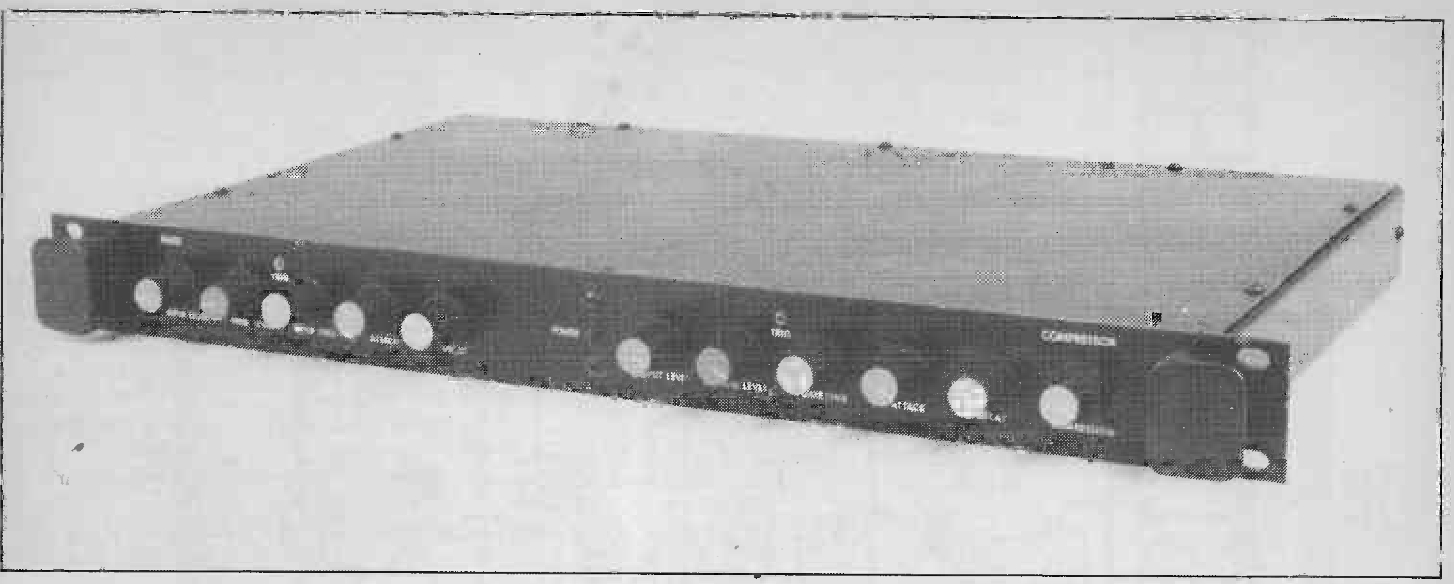
ABC Member of the Audit Bureau of Circulation

● Subscription rates — UK:£18.00. Europe:£22.20. Middle East:£22.40. Far East:£24.00. Rest:£22.70 or US$39.00.

ARGUS PRESS GROUP

1 Golden Square
London W1R 3AB
Tel: 01-437 0626
Telex: 8811896

REGULARS



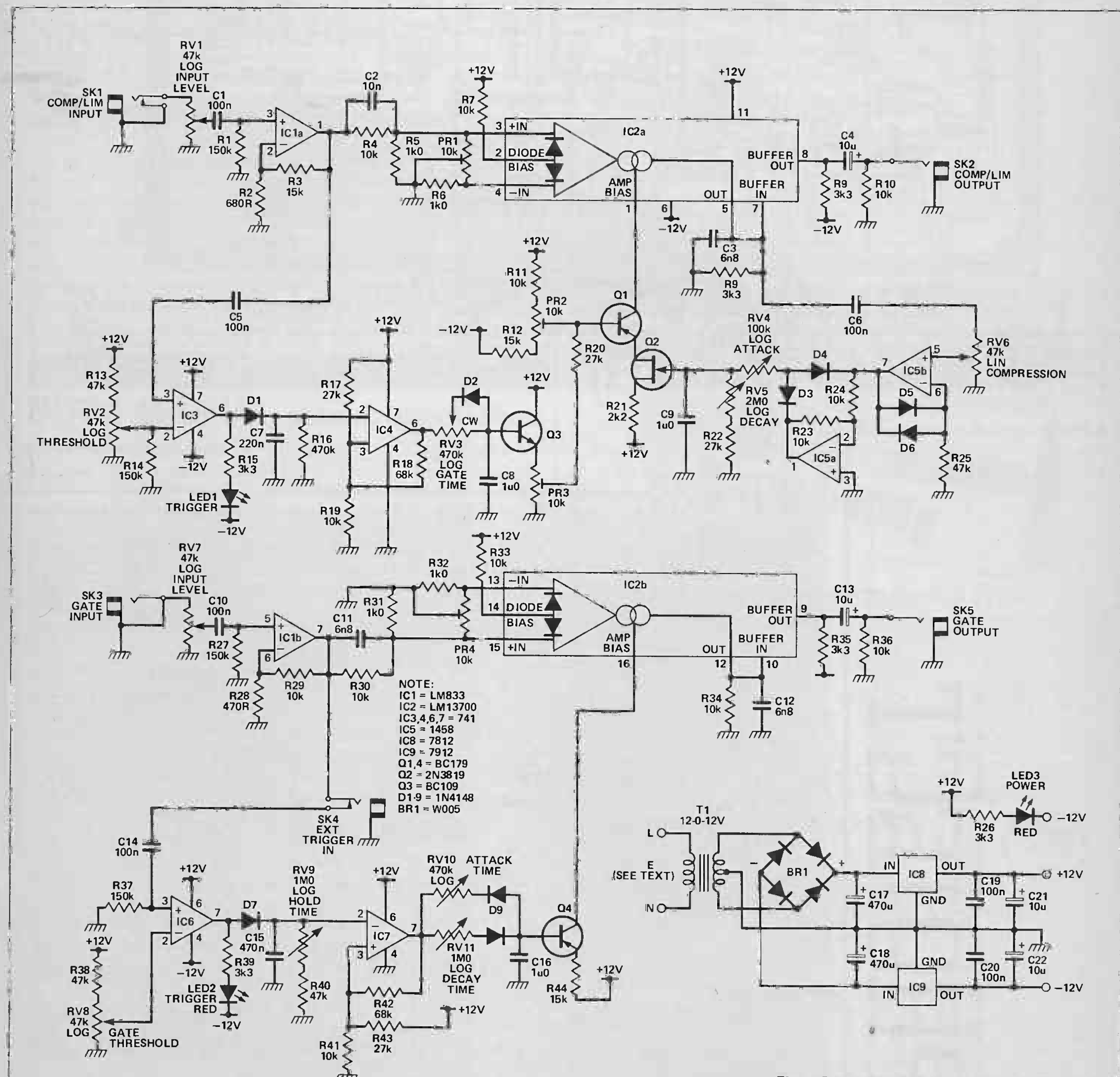


Fig.4 Circuit diagram of the complete unit

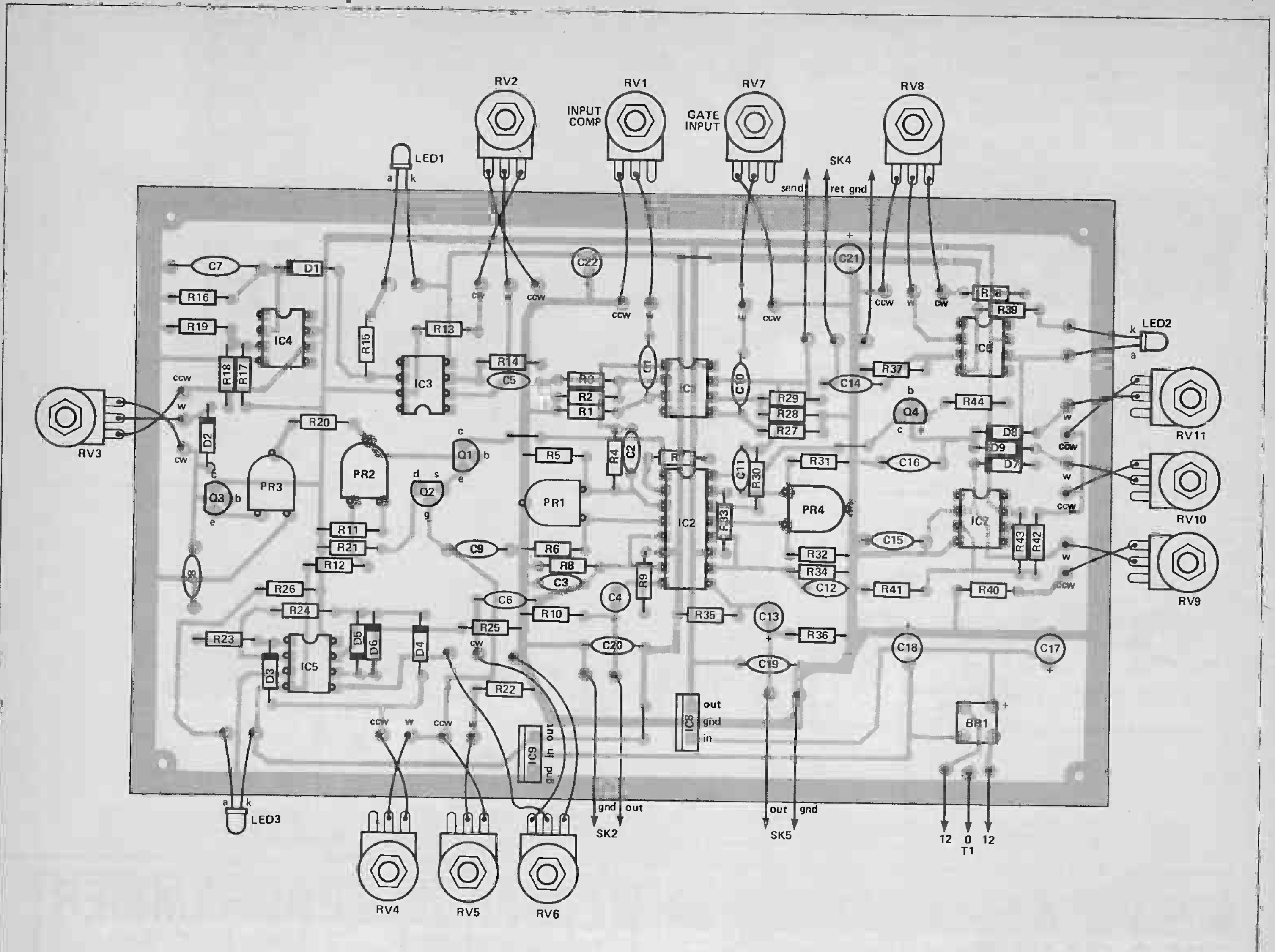


Fig.5 The component overlay for the compressor/limiter/gate

PROJECT

PARTS LIST

RESISTORS (all ¼W 5%)	
R1,14,27,37	150k
R2	680R
R3,8,12,44	15k
R4,7,10,11,19,23,24,29,30,33,34,36,41	10k
R5,6,31,32	1k
R9,15,26,35,39	3k3
R13,25,38,40	47k
R16	470k
R17,20,22,43	27k
R18,42	68k
R21	2k2
R28	470R
RV1,2,7,8	47k log
RV3,10	470k log
RV4	100k log
RV5	2M0 log
RV6	47k lin
RV9,11	1M0 log
PR1-PR4	10k horiz preset

CAPACITORS	
C1,5,6,10,14,19,20	100n polyester
C2	10n polyester layer
C3,11,12	6n8 polyester layer
C4,13,21,22	10µ 16V radial electrolytic
C7	220n polyester
C8,9,16	10µpolyester layer
C15	470n polyester layer
C17,18	470µ 25V radial electrolytic

SEMICONDUCTORS	
IC1	LM833
IC2	LM13700
IC3,4,6,7	741
IC5	1458
IC8	7812
IC9	7912
Q1,4	BC179
Q2	2N3819
Q3	BC109
D1-9	1N4148
D1-3	Red LED
D13	W005 bridge

MISCELLANEOUS	
SK1-5	¼in jack socket with break contacts
T1	12-0-12V 250mA miniature transformer

PCB. Case. Knobs to taste. Optional 100R earthing resistor (see text). Nuts and bolts

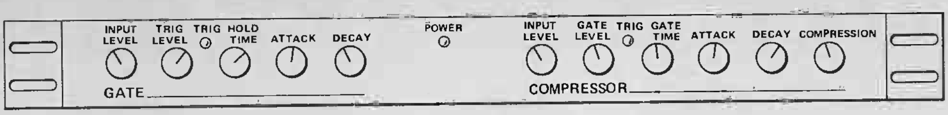


Fig.6 Suggested front panel layout

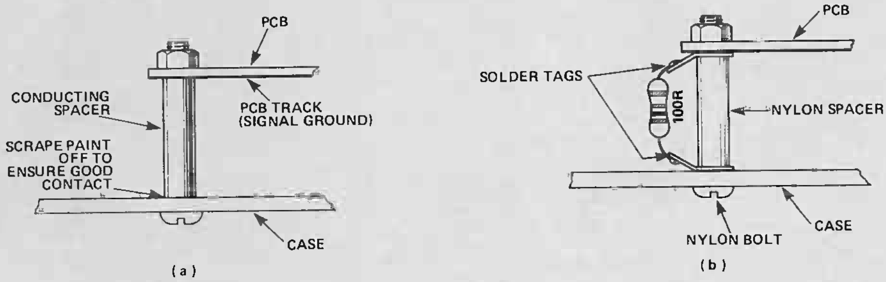


Fig.7 Earthing the board (a) With metal spacers (b) With insulating spacers

available with the Ext Trig input such as producing an electronic drum sound by feeding a note from a synth into the main input and using the bass or snare signal to trigger it.

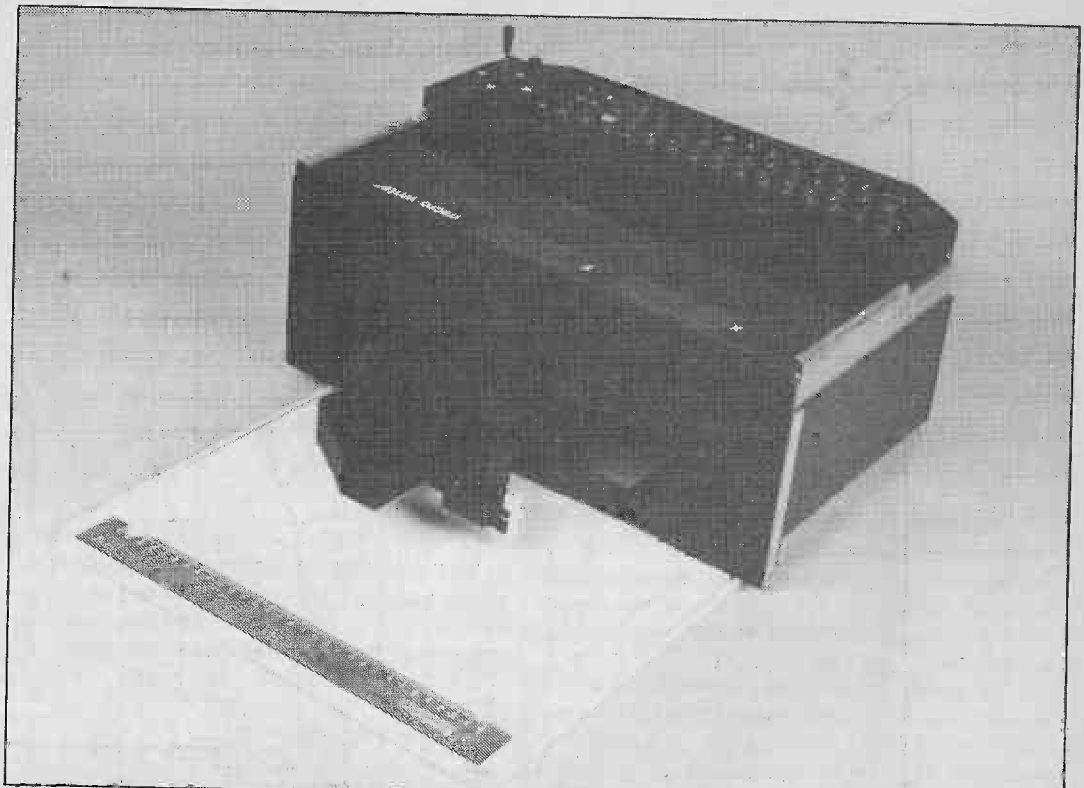
Frequency selective gating can be done by splitting a signal in two, routing one half direct to the input and the other to the Ext Trig input via a graphic or parametric EQ. This can then be used to separate a bass drum from a percussion track by boosting the bass on the EQ. Boosting the midrange will usually get the snare. These can then be fed through further effects and mixed back in with the percussion track to give a beefier bass/snare or whatever. Lastly, remember there are never any rules about using compression or gating, and the more weird and interesting noises you end up with the better.

Happy twiddling.

BUYLINES

All the components for this project are easily available from normal sources. A suitable case (1U size) is available from TJA developements, 19 Welbeck Road, Harrow HA2 0RN.

INTELLIGENT PLOTTER



Bob Joyce presents the first part of his step-by-step guide to an intelligent text and graphics plotter

UPDATE

The Intelligent Plotter is a 2-axis plotter based on the 6502 microprocessor. It uses standard plotter pens and has a precise resolution of 0.1mm to produce engineering artwork on foil, paper or other materials.

The design has two separate drives, each with a 1.8° stepper motor driven in half-step mode to provide 400 steps per revolution of the motor. A QWERTY keyboard has been included to allow for text entry and there are two 4-digit displays to show program data, speed parameters and current pen position.

Initial start up accuracy is maintained by single

HOW IT WORKS

The Intelligent Plotter uses a unipolar drive board having a combination of chopper control and RL control. The circuit is shown in Fig. 1. The current through the coils is detected by the sense inputs of IC2 and switches off the corresponding output transistors when the preselected current (set by RV1) is reached.

Two 50W ceramic resistors are included in series with the motor coils to reduce the heat dissipation of the motor and output stages. A higher resistance value will however cause a reduction in the maximum stepping frequency possible, so a compromise value must be made for each stepper motor. The values given in *Parts List* are suitable for a 32V supply driving a 5V stepper motor (See *Buylines*).

Logic signals of 0-5V are used to control movement, direction and type of step.

The L297 can be used in full step mode with pin 19 at 0V or in half step mode with pin 19 at 5V. To rotate the motor a clock signal is applied to a voltage divider (a 4V7 zener diode clips the input signal if excessive). IC1 inverts the input signal once or twice depending on the position of the DIL switch SW1 thus providing the necessary clock signal to pin 18 of IC2. After 0V volts has been applied this clock signal will cause the motor to move on the rising edge of the next clock pulse.

The four motor drive signals from IC2 are logically ANDed with the relevant inhibit signal from IC2 (to provide output current limiting) and through the power stages of IC5 and IC6 to the bases of the output darlington transistors Q1-4 to drive the stepper motor coils. 50W resistors (R26, 27) are used to reduce the heat dissipation of the stepper motors and drivers. A higher value will reduce the possible stepping frequency so a compromise value must be found for a particular stepper. The values given here are suitable for a 32V supply driving a 5V stepper motor (see *Buylines*).

To provide rotation the coils are energised in a set sequence as follows:

```
FULL STEP
coil A    0 1 1 0 0 1 1 0 0 1 1 0
coil B    1 0 0 1 1 0 0 1 1 0 0 1
coil C    0 0 1 1 0 0 1 1 0 0 1 1
coil D    1 1 0 0 1 1 0 0 1 1 0 0
. . . . . . . .: . . . . : . . . . . . . .
(code repeated every 4 steps)

HALF STEP
coil A    0 0 1 1 1 0 0 0 0 0 1 1
coil B    1 0 0 0 0 0 1 1 1 0 0 0
coil C    0 0 0 0 1 1 1 0 0 0 0 0
coil D    1 1 1 0 0 0 0 0 1 1 1 0
. . . . . . . .: . . . . . . . . . . . .
(code repeated every 8 steps)
```

The drive board supports both full and half step modes, selected by SW1c. For normal operation half step operation is used. The resolution is 0.1mm with a 200 step (1.8°) stepper motor. Thus when it is operating in the half step mode one revolution of the stepper motor produces a linear distance of $200 \times 2 \times 0.1mm = 40.0mm$.

The 240V AC mains supply is transformed down to 24V and rectified by BR1. Capacitor C14 provides smoothing to the unregulated DC signal which is passed through R7 and D1 into the 723 voltage regulator (IC3) where the voltage is regulated to about 12.5V.

Q5 is used to provide increased current capability from the 723 and feeds the auxiliary output. Resistor R21 provides current limitation. The 723 regulated output is applied to IC4 which regulates the 5V supply.

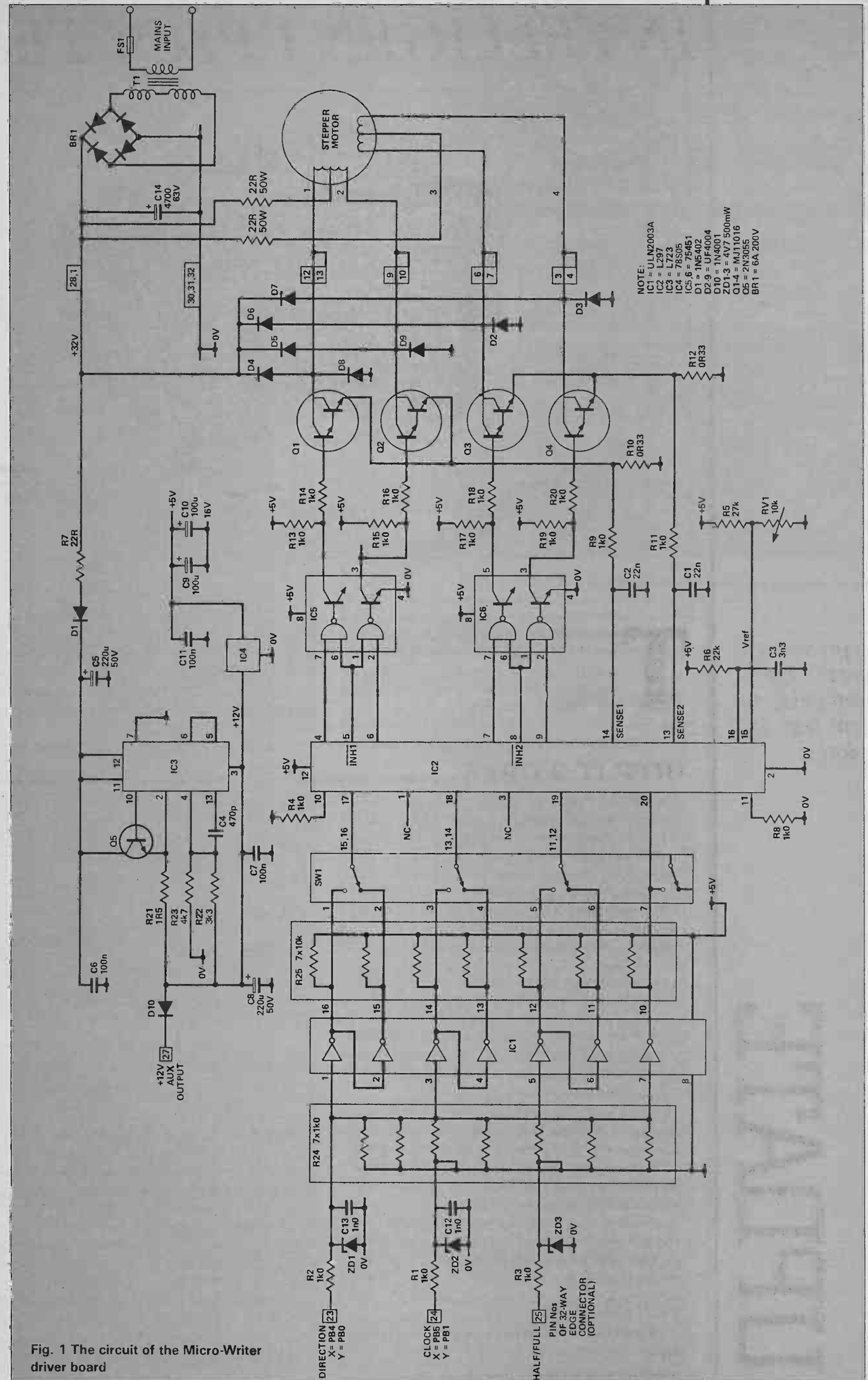


Fig. 1 The circuit of the Micro-Writer driver board

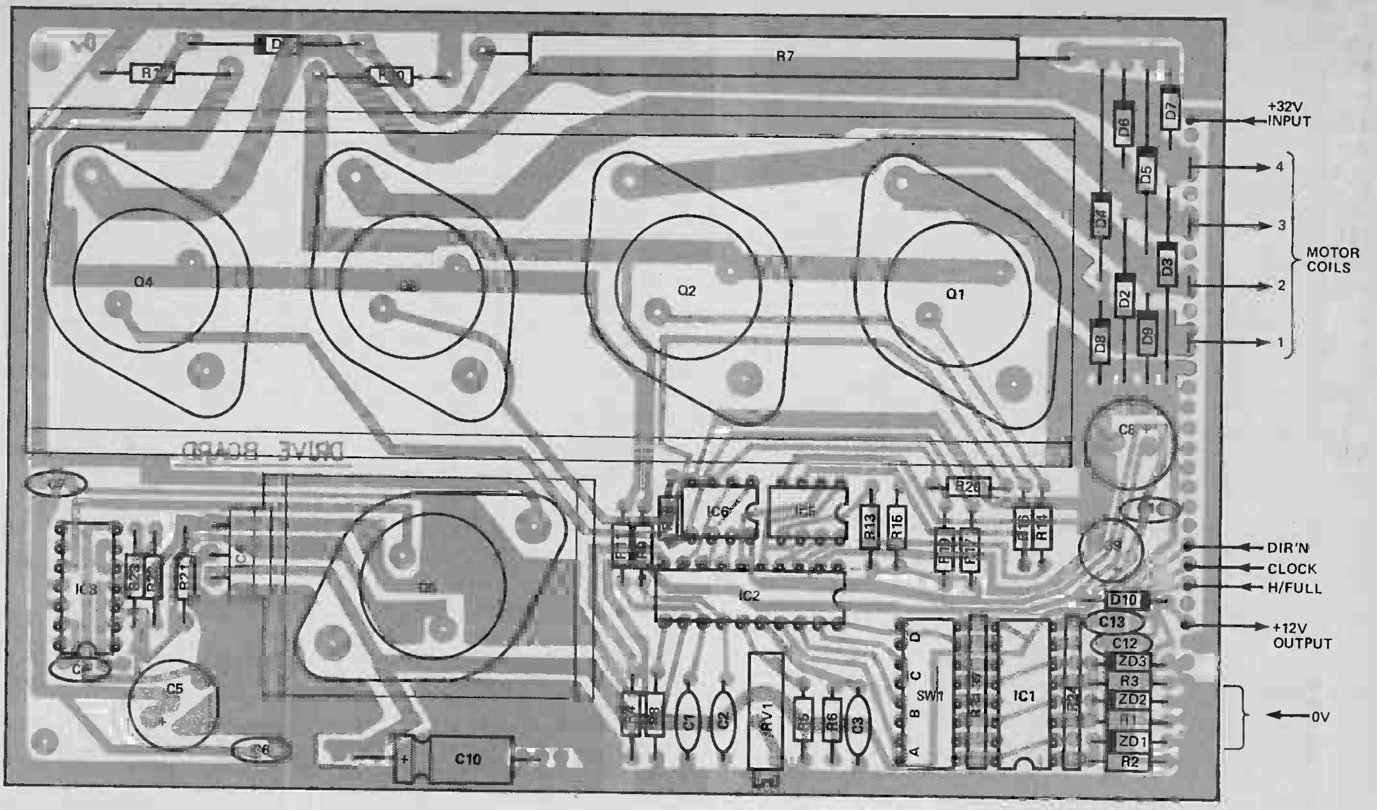


Fig. 2 The component overlay for the driver board

UPDATE

key automatic datumming. An inbuilt joystick allows for operation of the Plotter for freehand drawing.

Taking Steps

This first article deals with the stepper drive board. Each direction of movement requires a separate stepper motor so we need two drive boards.

The boards utilise a stepper motor driver IC from SGS, the L297, to perform the necessary conversion from direction, clock and step type (full step or half step) to four coil outputs. These outputs are not able to drive the stepper motors directly — this is done by four darlington power transistors via a couple of logic driver ICs. The 5V input control signals to the drive board can be taken from the Intelligent Plotter microprocessing board, from a computer (using say a BBC user port), or from a square wave oscillator for manual operation.

Stepper motors convert digital signals into rotary motion. The stepper motor rotates a single step with each pulse, thus providing precise positioning without the need for feedback encoders. When stepped at high rates the rotation appears continuous. Reverse stepping is simply a matter of reversing the coil energising sequence.

To obtain maximum speeds the stepper motor must be accelerated (referred to as ramping) into its slow range and decelerated into its base speed to stop or change direction. If the motor is reversed without deceleration, loss of step can occur with a corresponding loss of accruacy.

On the Micro-Writer the base speed, acceleration rate and maximum speed can all be programmed from the keyboard at any time between plotting.

Construction

The component overlay for the drive board is shown in Fig. 2. Solder all components except the power transistors and the voltage regulator (Q1-5 and IC4)

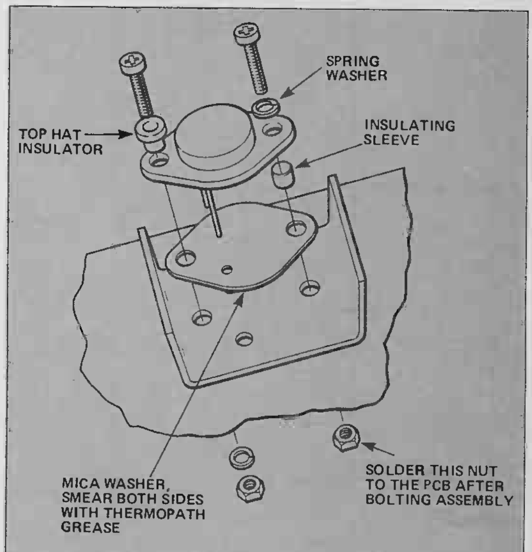


Fig. 3 Mounting the power transistors

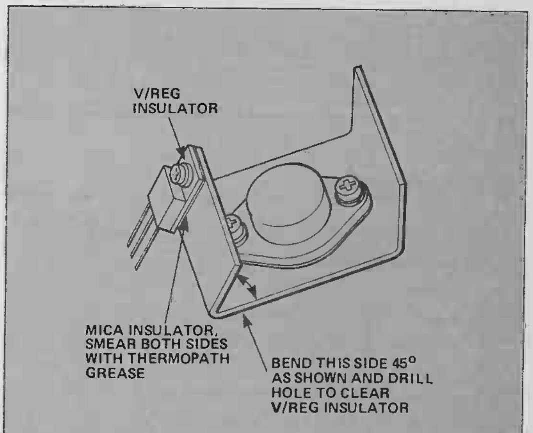


Fig. 4 Mounting the voltage regulator IC4

PARTS LIST

RESISTORS (all ¼W 5%)

R1-4, 8, 9, 11, 13-20	1k0
R5	27k
R6	22k
R7	22R 17W 5% ceramic
R10,12	0R33 2W5 5%
R21	1R5 2W5 5%
R22	3k3
R23	4k7
R24	10k 7-commoned resistor network
R25	1k0 7-commoned resistor network
R26,27	22R 50W

CAPACITORS

C1,2	22n polycarbonate
C3	3n3 polystyrene
C4	470p mica
C5,8	200μ 50V radial elec
C6,7,11	100n sub min polyester
C9	100μ 16V radial
C10	100μ 16V axial
C12,13	1n0 sub min polyester
C14	4700μ 63V elec

SEMICONDUCTORS

IC1	ULN2003A
IC2	L297
IC3	L723
IC4	78S05
IC5,6	75451
Q1-4	MJ11016
Q5	2N3055
D1	1N5402
D2-9	UF4004
D2-9	UF4004
D10	1N4001
ZD1-3	4V7 500mW zener

MISCELLANEOUS

BR1	6A 200V bridge rectifier
FS1	250mA antisurge
SW1	4-pole changeover
T1	50VA 0-12 0-12 transformer

MANUAL BOARD

R28, 30, 32	1k0
R29	220R
R31	820R
RV1	1M0 dual gang linear pot
C15	0μ100n
C16	0μ470n polystyrene
C17	47μ 25V tantalum
C18	1n0
C19	10n
IC7	NE555
Q6	BC109C

PCB. Optional 32-way connector and 90° socket. Heatsinks (see *Buylines*). Nuts and bolts.

BUYLINES

Most components are available from Electromail (Tel:(0536) 204555). The stepper motor is part 332-082, SW1 part 337-554 and the heatsinks part 401-778 (or make your own from aluminium).

A 40-track disc is available for the BBC with the BASIC program and an interrupt driven machine code program to test the operation of up to two drive boards with independent speed and direction control. This is available for £10 from the author at 104 Craythorne Avenue, Handsworth Wood, Birmingham B20 1LN.

The PCB is available from the ETI PCB Service on (0442) 41221.

noting polarity of diodes, capacitors and IC holders. The resistor networks (R24 & R25) consist of 7 commoned resistors, the commoned end is marked with a dot. The higher wattage resistors (R7, 10, 12 & 21) should be mounted 1-2mm off the PCB.

The power transistors Q1-5 should be loaded next as shown, ensuring that no shorting occurs.

Next mount the voltage regulator IC4 and secure to heatsink (Fig. 4) prior to soldering it in. Again ensure that no shorting occurs.

Testing

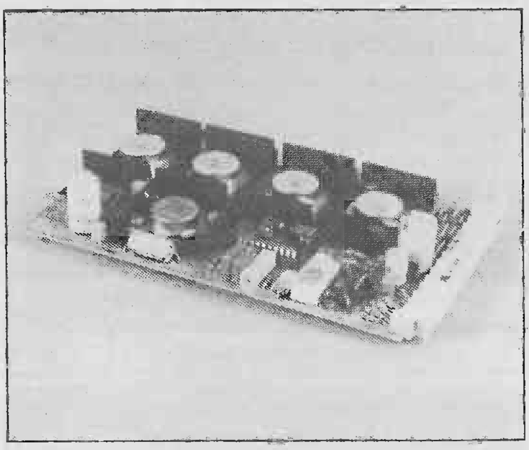
Before inserting any ICs into their sockets, closely inspect the PCB tracks for shorts or bad joins. Time spent here is well worth the effort compared with tracing the fault later!

Check each power transistor resistance from base to emitter and collector, any low reading here suggests a fault. Finally check that all polarized components are correctly orientated.

To set up the drive board, firstly check the 32V supply before connecting it to the drive board. It should be around 30-33V.

If this is okay discharge the smoothing capacitor C14 with a 100R resistor (not a screwdriver!). Insert IC3 into its socket, connect the power supply to the drive board (either using the optional 32-way connector or direct soldering), then switch on and check the auxiliary 12V output at D10 and the 5V on the +end of C10.

Then insert the rest of the ICs. Don't proceed if the 5V supply is more than 0.25V out. In case of errors check the voltages around IC3 and Q5 for shorts or incorrect value resistors.



Calibration

Rotate RV1 to the anti-clockwise end and connect the motor as shown in Fig. 5. Connect the power supply but not the computer or manual oscillator yet.

Switch on power and try rotating the motor spindle by hand — there should be very little torque. Now increase the current to the motor windings by rotating RV1 clockwise until the motor current to the coils is equal to the manufacturer's data for the motor in use (1A on the specified motor).

Alternatively, rotate RV1 until the motor torque is sufficiently high to be very difficult to rotate, then rotate RV1 one further revolution clockwise.

Switch off power and connect the computer user port to the motor board as shown below in Fig. 6.

Software

To enable the stepper motor drive boards to be checked, I have included a simple BBC computer

```
10 REM SINGLE AXIS
20 REM BASIC STEPPER MOTOR DRIVER
30 BOB JOYCE
40 REM 1988
50
60 MODE7
70 ON ERROR RUN
80
90 REPEAT
100   PROCinit
110   PROCinput
120   PROCstepper
130   UNTIL FALSE
140 END
150
160 DEF PROCinit
170 ddrb=&FE62
180 irb=&FE60
190 orb=&FE60
200 ?ddrb=&FF
210 ENDPROC
220
230 DEF PROCstepper
240 IF dir=1 THEN ?orb = ?orb AND &EE
250 IF dir=2 THEN ?orb = ?orb OR &11
260 FOR noofsteps = 0 TO steps
270   PRINTTAB(30,4);steps-noofsteps;SPC(10)
280   ?orb = ?orb OR 02
290   ?orb = ?orb OR &20
300   FOR delay=0 TO actualspeed : NEXT
310   ?orb = ?orb AND &DD
320   NEXT
330 ENDPROC
340
350 DEFPROCinput
360 CLS
370 PRINTTAB(1,2)CHR$(134);CHR$(141);"Enter the number of steps"
380 PRINTTAB(1)CHR$(134);CHR$(141);"Enter the number of steps"
390 INPUTTAB(30,4)step$
400 steps=VAL step$
410 IF steps <=0  PROCerror(step$):GOTO370
420 IF steps >999999999 PROCerror(step$):GOTO370
430 PRINTTAB(1,8)CHR$(134);CHR$(141);"Enter dir  1(cw) 2(acw)"
440 PRINTTAB(1)CHR$(134);CHR$(141);"Enter dir  1(cw) 2(acw)"
450 INPUTTAB(30,10)dir$
460 dir=VAL dir$
470 IF dir >2 PROCerror(dir$):GOTO430
480 IF dir <1 PROCerror(dir$):GOTO430
490 PRINTTAB(1,14)CHR$(134);CHR$(141);"Enter speed in revs/min.(1 to 48)"
500 PRINTTAB(1)CHR$(134);CHR$(141);"Enter speed in revs/min.(1 to 48)"
510 INPUTTAB(30,16)speed$
520 speed= VAL speed$
530 IF speed >48PROCerror(speed$):GOTO490
540 actualspeed = 48/speed
550 ENDPROC
560
570 DEF PROCerror(X$)
580 CLS : VDU7
590 PRINTTAB(0,6) "Sorry!";CHR$(136);CHR$(129);X$
600 PRINTTAB(0,18) ;CHR$(137);CHR$(135);"Is not acceptable try again"
610 FOR delay=1 TO 5000
620   NEXT
630 CLS
640 ENDPROC
```

Listing 1 Basic Checking Program

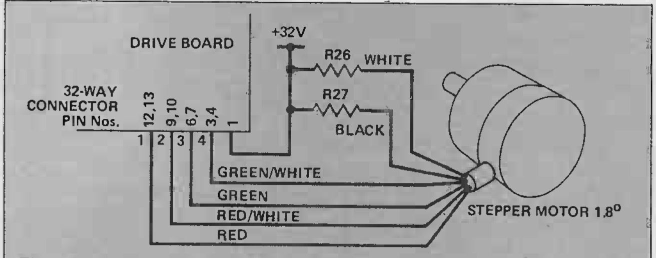
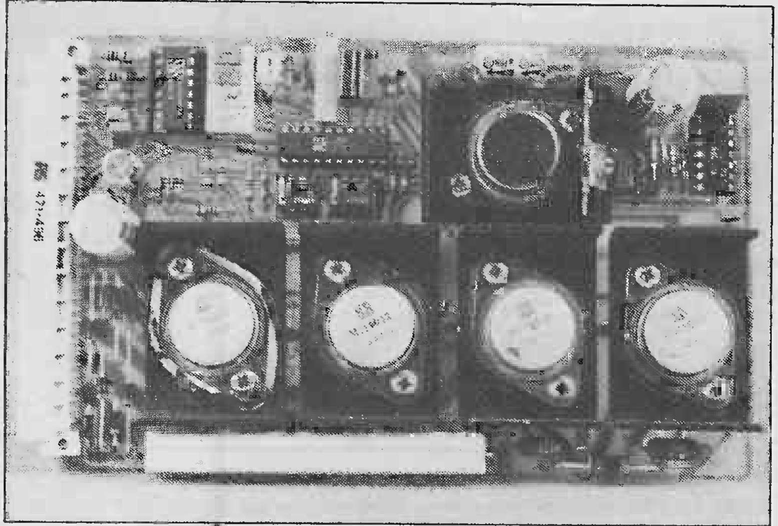


Fig. 5 The connections to the stepper motor



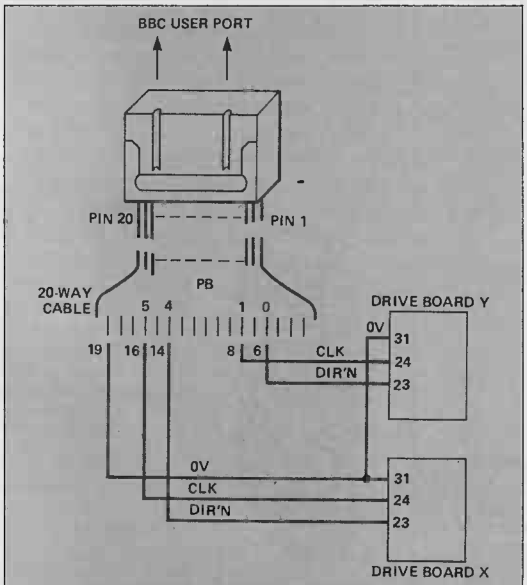


Fig. 6 Off-board connections for use with a BBC micro

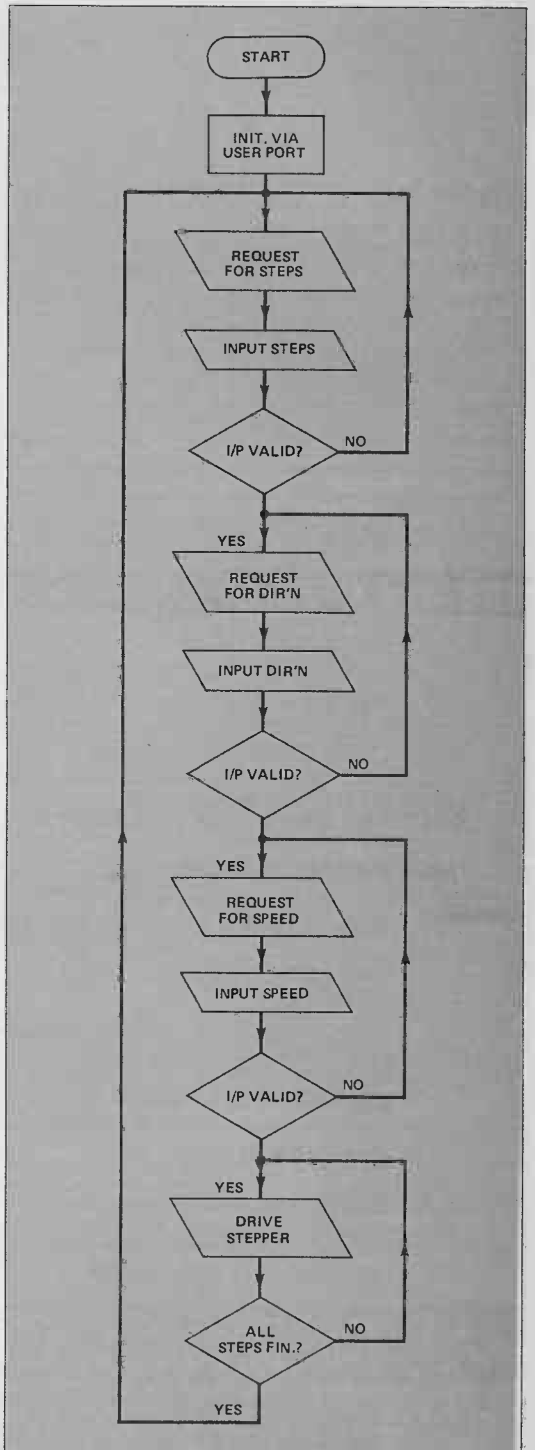


Fig. 7 The flow chart for the BBC program

Basic program listing (Listing 1).

The flow chart in Fig. 7 shows the sequence of operation of the program. Type in the program carefully (make sure you distinguish between O and zero 0). Save it before running to prevent corruption (if you are unsure if the program is running correctly then temporarily remove line 70 (ON ERROR RUN) — most faults should be captured by the operating system and displayed as an error).

To enable full flexibility of speed and operation on a BBC micro, 6502 machine code must be used. The author has written an interrupt-driven machine code program to drive one or both drive boards and change speed and direction independently. This machine code program together with the basic program and a menu control are available on a 40-track disc or cassette — see *Buylines* for details.

Next month we will move on to the mechanics of the Intelligent Plotter.

ETI

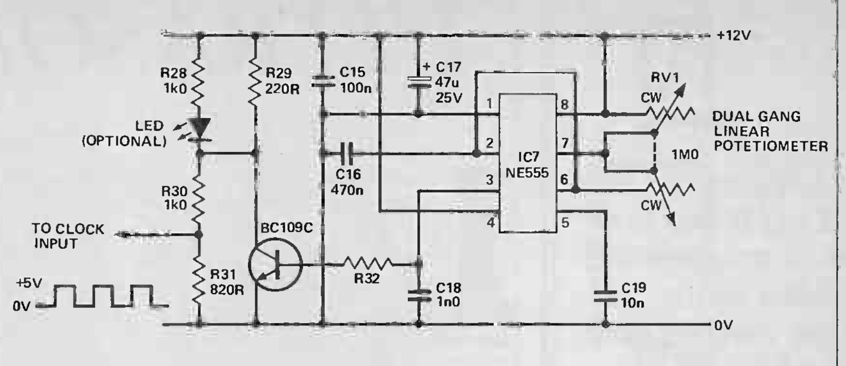


Fig. 8 A suggested circuit for manual control

MANUAL CONTROL

Figure 8 shows a suggested circuit for stand-alone operation, using the ubiquitous 555 timer IC. The 555 operates in astable multivibrator mode to produce the necessary square wave oscillator. Note the adjustment is via a dual gang 1M0 linear pot.

This circuit can be used instead of the microprocessor board or computer if you require simple manual operation only. The connections for such a system are shown in Fig. 9.

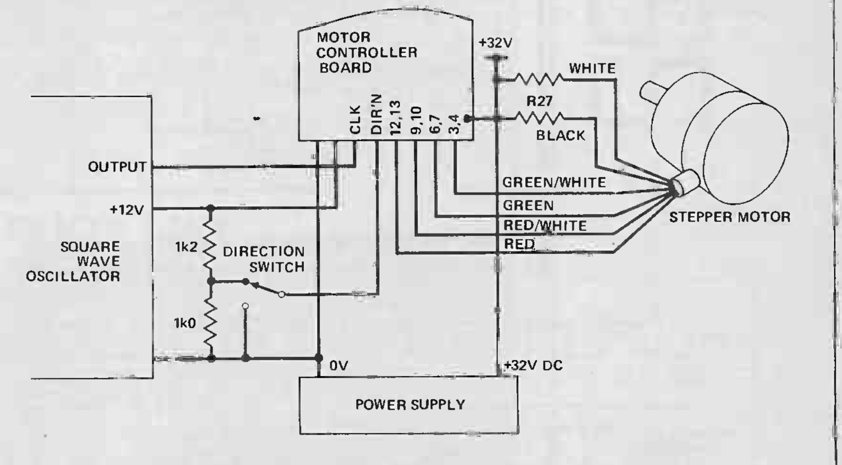


Fig. 9 Connections for manual control

ULTRASONIC HORN

Gary Calland has run out of old boots to throw at moggies but now has a more humane method of ridding himself of pests

Is your garden the local haunt for the neighbourhood dogs and cats? Do they find your flowerbeds the ideal play area and your dustbin just the place for a small snack? If so, then the Ultrasonic Horn is for you!

Ultrasonics is the region of sound waves above about 20kHz. This high frequency cannot be heard by humans but it can be heard by our animal friends. The ability of animals to hear ultrasonics was first used in the form of dog whistles. Here, the ultrasonic signal is relatively quiet and pleasant to the dog. However, the signal from the early ultrasonic television remote control units were quite different and had strange effects on family pets. If loud enough, the ultrasonic signal is piercing and irritating (but harmless) to animals and they tend to get as far away from it as they possibly can.

The Ultrasonic Horn described here is small but produces an extremely loud 23kHz ultrasonic signal and should be enough to scare away even the most determined animal from the property.

Ever had a dog or cat run out in front of your car causing you to dangerously brake or swerve to avoid it? Such incidents could be a thing of the past if the Ultrasonic Horn was fitted to your car. A quick burst of noise will ensure your presence is felt, dissuading the animal to shoot out in front of your car.

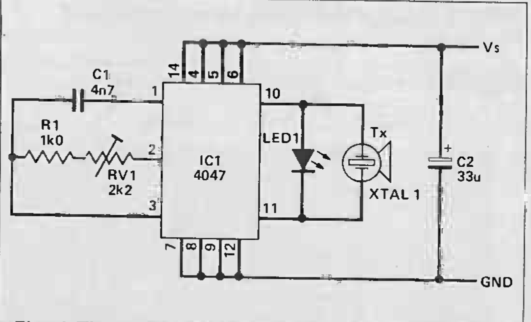


Fig. 1 The circuit diagram of the Ultrasonic horn

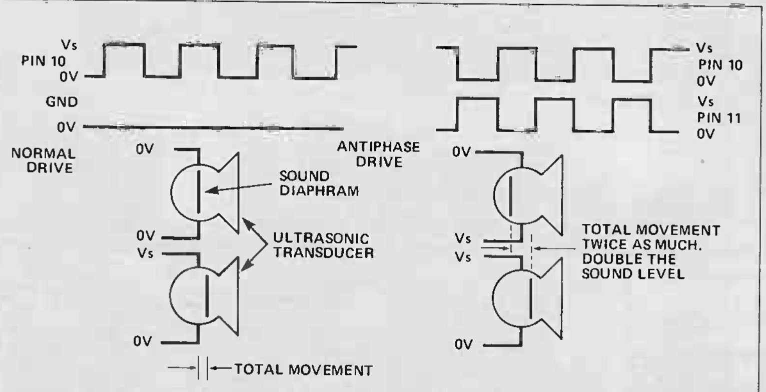


Fig. 2 The increased volume obtained from the ultrasonic transducer with an anti-phase drive signal

Construction

Construction is extremely simple if either the PCB version (Fig. 4) or the stripboard version (Fig. 5) is built. In either case the IC should be mounted in an IC socket to reduce the risk of damage by heat or static electricity. Remember that touching any pins of the CMOS IC may render it useless due to the static charge that exists on a person — this should be avoided!

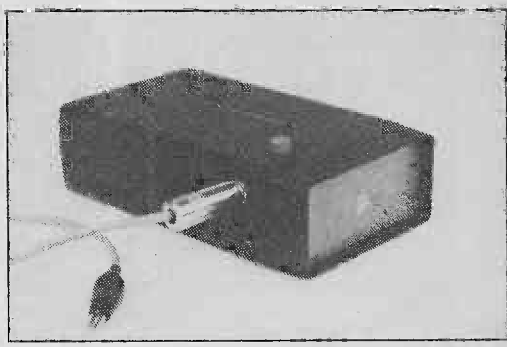
Note the polarity of the capacitor C2 but the LED may be soldered either way round. If the stripboard version is to be built make sure all the eight breaks are made in their correct positions with a twist drill bit. Both versions are small enough to be mounted in virtually any case.

1st CLASS

Published by Argus Specialist Publications Ltd., 1 Golden Square, London W1R 3AB. Tel: 01-437 0626. UK newstrade distribution by SM Distribution Ltd., 6 Leigham Court Road, London SW16 2PG. Tel: 01-677 8111. Overseas and non-newstrade sales by Magazine Sales Department, 1 Golden Square, London W1R 3AB. Tel: 01-437 0626. Subscriptions by Infonet Ltd., 5 River Park Estate, Berkhamsted HP4 1HL. Tel: (0442) 876661. US subscriptions by Wise Owl Publications, 4314 West 238th Street, Torrance, CA90505 USA. Typesetting and origination by Project 3 Filmsetters, Whitstable. Printed and bound by The Chesham Press, Chesham. Covers printed by Loxley Brothers Ltd., Sheffield.

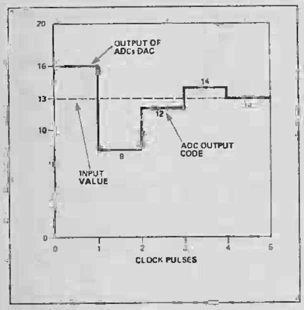
ETI
ELECTRONICS TODAY INTERNATIONAL

FEATURES/PROJECTS

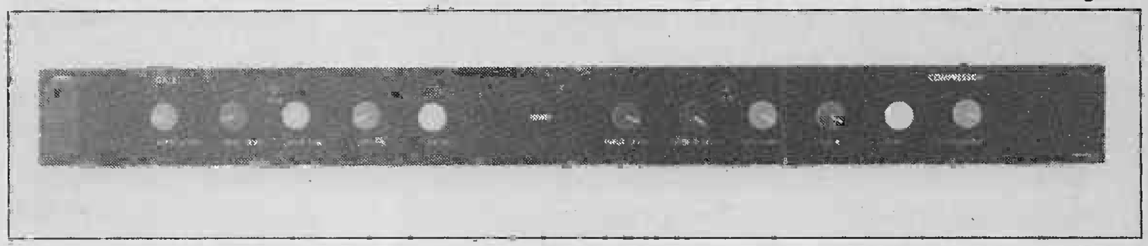


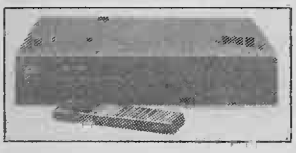
Page 40

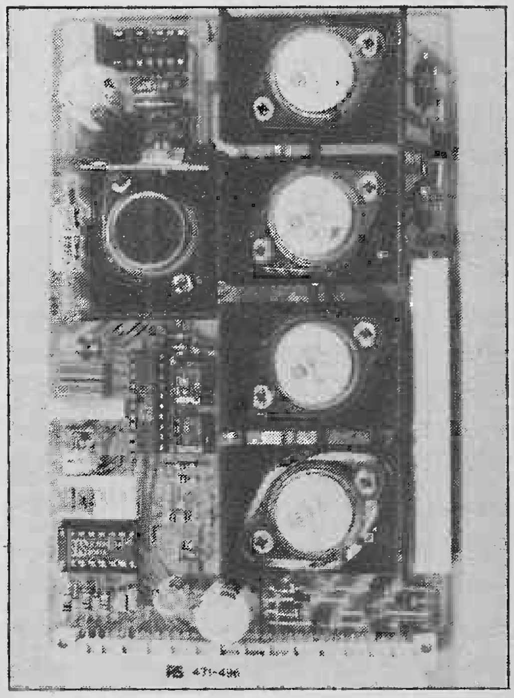
Page 46



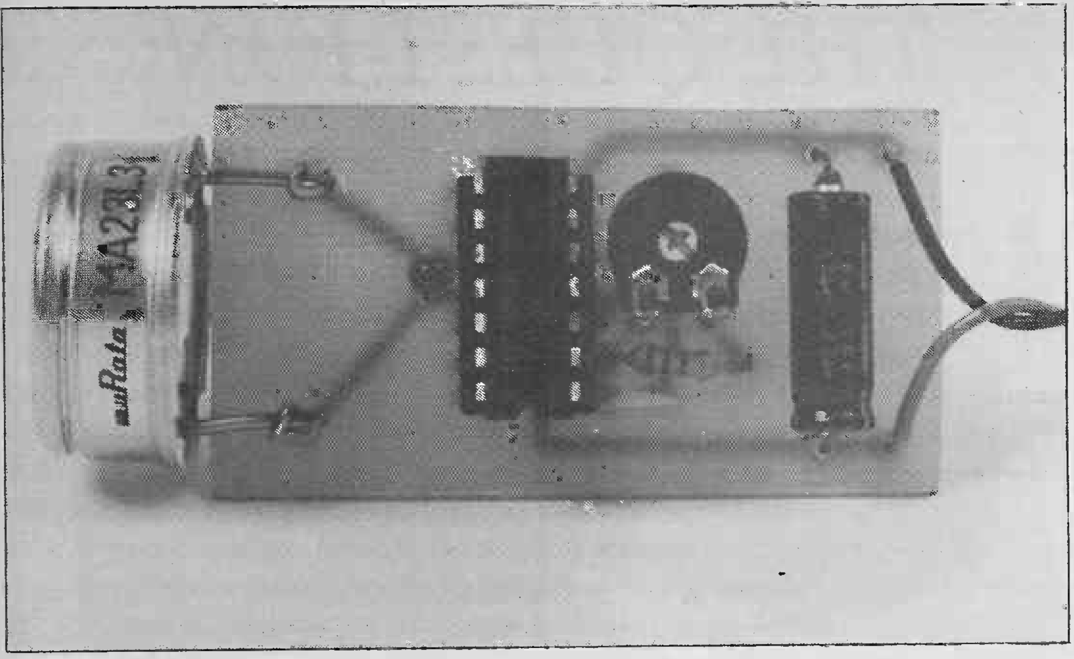
Page 26







Page 32



In Use

Any supply voltage in the range 3-20V can be used. The higher the voltage, the louder the ultrasonic signal. When power is applied, the LED will glow brightly. RV1 should be adjusted until the LED dims slightly. At this point, RV1 is set correctly and the ultrasonic transducer will be fed its required 23kHz signal. LED1 can now be carefully removed as it is no longer needed and only draws more current than needed from the supply. Fit a switch if required into the supply line. The Ultrasonic Horn is now ready for use.

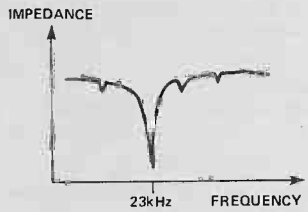


Fig. 3 The characteristic of the ultrasonic transducer

HOW IT WORKS

The Ultrasonic Horn circuit is based around the CMOS 4047 astable/monostable IC (see Fig. 1). Here it is wired in the astable mode and produces square waves at a frequency set by C1, R1 and RV1. RV1 allows the frequency of oscillations to be varied from between about 17kHz to about 25kHz so the output frequency can be tuned to the frequency best suited to the ultrasonic transducer.

This is driven in antiphase by the complementary outputs, pin 10 and 11. When pin 10 is high, pin 11 in low and when pin 10 is low, pin 11 is high so by connecting the transducer across these rather than to ground, the voltage fed to the transducer is virtually doubled to twice the supply voltage so producing a louder sound volume. See Fig. 2.

The LED is temporary and is used to tune the 4047 oscillator. The ultrasonic transducer is build to oscillate at one specific resonant frequency, in this case 23kHz. At this frequency, the impedence of the transducer drops sharply as in Fig. 3. As a result, the current through it increases and the current through LED1, connected in parallel, decreases. So the LED dims when the ultrasonic transducer is tuned correctly.

PARTS LIST

RESISTORS (all ¼W 5%)	
R1	1k0
RV1	2k2 horizontal preset
CAPACITORS	
C1	4n7 ceramic
C2	33µ axial electrolytic
SEMICONDUCTORS	
IC1	4047
LED1	Any LED
MISCELLANEOUS	
XTAL1	23kHz piezo transducer
PCB or Stripboard. Case.	

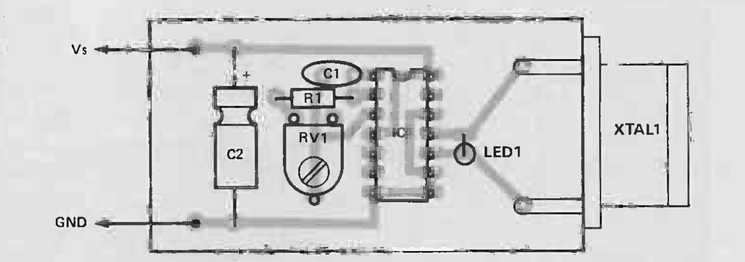


Fig. 4 The component overlay for the PCB version of the Ultrasonic horn

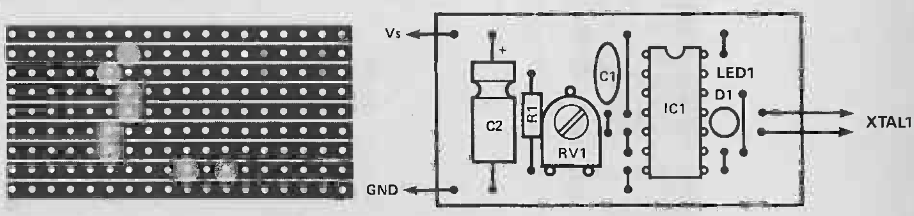


Fig. 5 The track cuts and component overlay for the stripboard version of the Ultrasonic horn

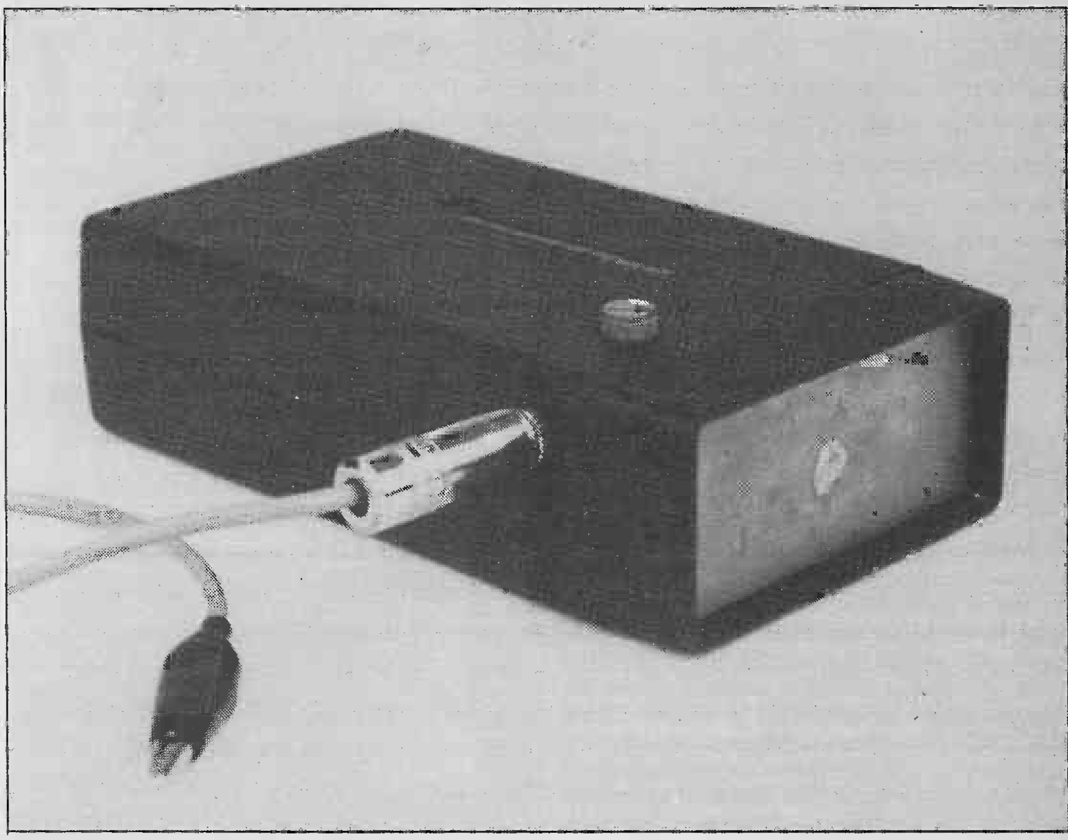
BUYLINES

The 23kHz piexo transducer is available from Cricklewood Electronics, 40 Cricklewood Broadway, London NW2 3ET. The PCB is available from the ETI PCB Service.

THE QUEST-ION

Paul Chappell searches high and low for ions in the air with this nifty charge checker

PROJECT



The Quest-Ion is a hand-held meter which sniffs out ions in the air. It can sort out the good ones from the duds if you're thinking of buying a commercial air ioniser, check the efficiency and output level of one you've made yourself, tell you whether the ions are all collecting around the emitter or diffusing throughout the room, do an ion-survey of your house or office — in short it will tell you anything you want to know about ions in the air.

The whys and wherefores of air ions were explained in detail in the Variat-Ion air ioniser project in October 1988. Briefly, the story goes like this. Through various natural processes, air molecules can pick up an electrical charge. If they gain a negative charge they're called negative ions, or neg-ions. With a positive charge they become pos-ions.

Pos-ions are blamed for all kinds of unpleasant things — feelings of depression, headaches, sickness, and so on. Neg-ions, on the other hand, are credited with all kinds of good effects — feeling alert, healthy and ready for anything. So you can see it might be useful to know which variety you're living in.

With so many sources of pos-ions around, many people don't take any chances. They make or buy air ionisers to cancel out the positive charge and fill their houses with an excess of neg-ions. Whether these ionisers work as well as they might depends on how effectively the ions are distributed around the room once they leave the emitter. The only way to be sure is to take reading with an ion meter, which is what this project is all about.

The Circuit

The circuit of the Quest-Ion is shown in Fig. 1. The basic principle of its operation is quite straightforward. The ions are collected by the electrode and cause a current to flow in the resistor chain R2-R8. The voltage developed across these resistors is amplified by IC1 and applied to a string of comparators IC2a-IC6a. The comparators are set to trigger in turn as the voltage at IC1 output rises or falls. Each triggering lights one of the LEDs. The arrangement acts as a dot-mode bar graph display of the ion level.

If you do a few sums you'll find it's far from being either log or linear — the only two varieties available in IC form — hence the multi-IC circuit.

With SW1 pressed to disconnect the collection electrode, RV2 is adjusted so that the amber LED (LED2) is lit. This shows that no ions (or less than 50 million a second — hardly worth bothering about!) are being picked up. When SW1 is released the electrode is connected to the circuit and the LED dot moves to register any ions in the air.

The layout of the display is shown in Fig. 2 — if the dot moves up to the red LED it shows that pos-ions are present, if it moves downwards to one of the green LEDs it will show how many neg-ions are being picked up.

With RV1 in the 'calibrated' position (minimum resistance) and assuming that each air ion is associated with one electronic charge, it takes 50 million ions per second to shift the display from amber to red (for pos-ions) or to the first green LED (for neg-ions). Anything below this won't register. It would be a fine thing to have a meter that was more sensitive still (you can, in fact, increase the sensitivity by adjusting RV1) but before getting carried away let's look at the difficulties we've already made for ourselves.

First of all, our 50 million ions per second only produce a current of 8pA in the input circuit. Not a lot, really. It is, in fact, pushing ordinary op-amps and PCBs very close to their limits. The problem is that any disturbing factor which can produce a similar current will also register on the display. The list is as long as you care to make it: RF and mains radiations, changes in op-amps bias current, thermal effects, capacitance effects, and so on.

Something you might not have thought to add to the list is conduction through the PCB itself, or across its surface. In last year's electric fence project I mentioned that materials we normally think of as insulators often turn out not to be when very high voltages are present. The same thing applies when you're dealing with very tiny currents and one of the things you can no longer think of as being a good insulator is the fibreglass that separates tracks on the PCB!

To put some figures to it, a pair of adjacent tracks on a clean, dry PCB may have a resistance of the order of 10^{12} Ohms between them. This means that for every 1V difference in their voltages, a current of 1pA will flow between them. Not very important when your circuit has milliamps flowing around it but when you're dealing with signals at the pA level it makes a huge impact! If the PCB is not clean — if it still has traces of etchant on it, or flux residues from soldering, general dust and dirt, or has absorbed moisture from the air, the situation can be very much worse.

The techniques for dealing with this problem are well known to any instrumentation designer but since ETI has not published a circuit of this nature within living memory, you might like to know a little about the basic principles. In particular, about the guard ring used in the Quest-Ion.

Guard Duty

There are a number of techniques for reducing leakage currents — most of them fairly obvious. If a glass-epoxy PCB has too low a resistance, use a material with a higher one! Polycarbonate and PTFE are better but expensive. Standing the sensitive parts of the circuit on PTFE insulated pins is another possibility. The device used in the Quest-Ion, a guard ring, works in a different way. You won't find it in the parts list — it's etched onto the PCB.

Figure 3a shows the situation we have to deal with. The diagram shows the underside of the PCB around IC1. The + input is right beside the −V supply pin. Assuming that the + input is at about 5V, as it will be in the Quest-Ion, and that the tracks don't run close together for more than about ¼in, the current flowing between them might be about 5pA. If the sensitive input track runs for any appreciable distance, as it must in this circuit since it connects to several other components, there will be other tracks in the vicinity which will add or take more current — and who can guess what the overall result might be?

A first line of defence would be to keep the sensitive track as far as possible from any others or even to run it as a PTFE insulated wire, but a better solution is shown in Fig. 3b. What you do is to surround the sensitive input, and anything connected to it, with a complete ring. Then you find a point in the circuit which is at almost the same voltage as the + input, but at a low impedance. Connect this point to the ring. If you do that, virtually no current will flow between the ring and the input because they're near as makes no odds at the same voltage. Any currents flowing from other tracks on the PCB will be absorbed by the guard and won't go where they can do any harm.

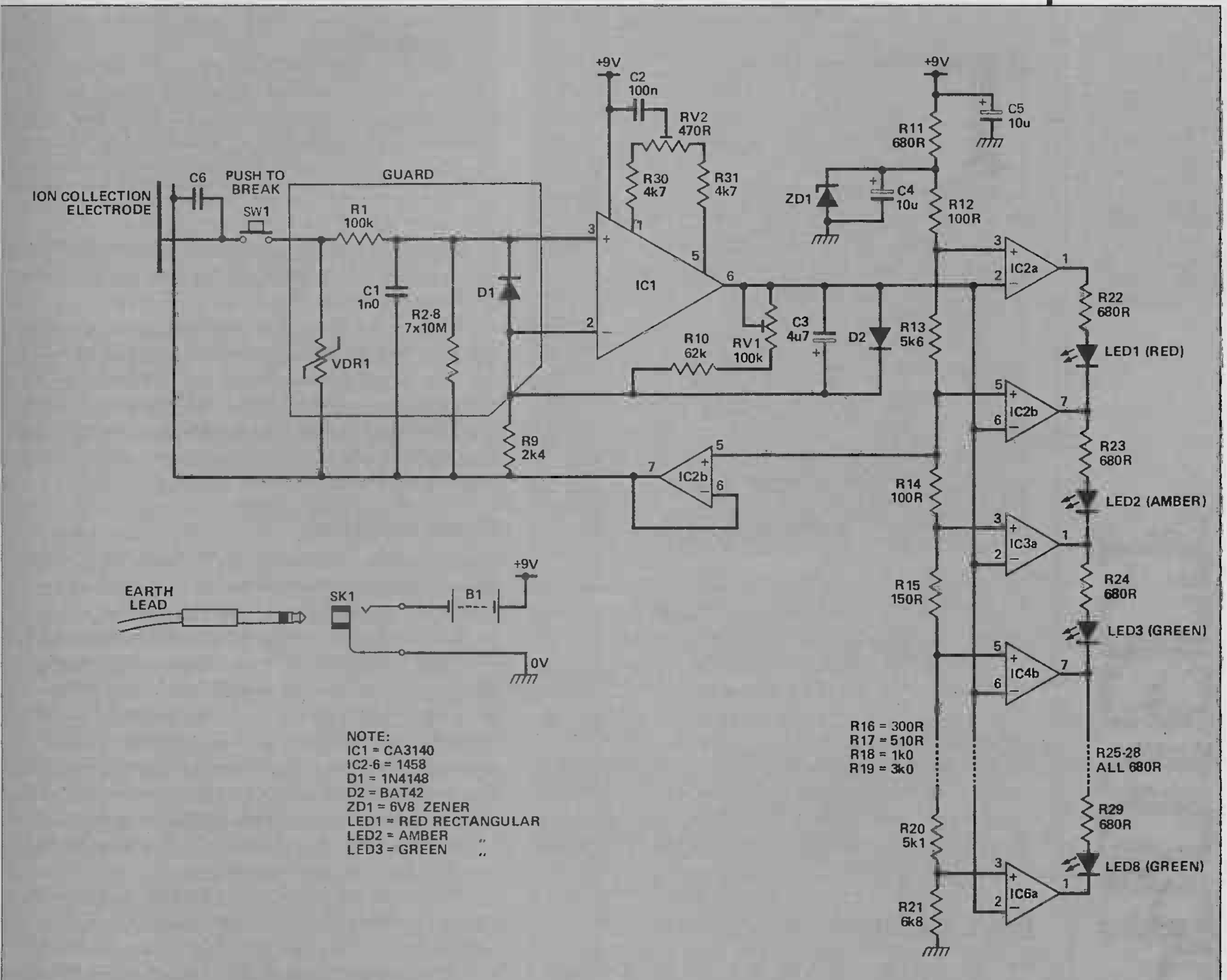


Fig. 1 Circuit of the Quest-ion

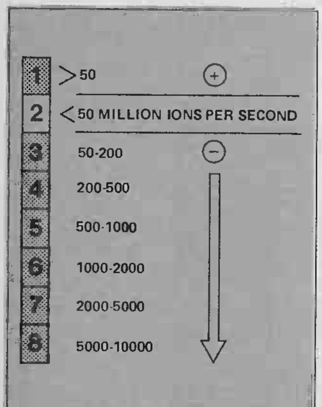


Fig. 2 The meter scale

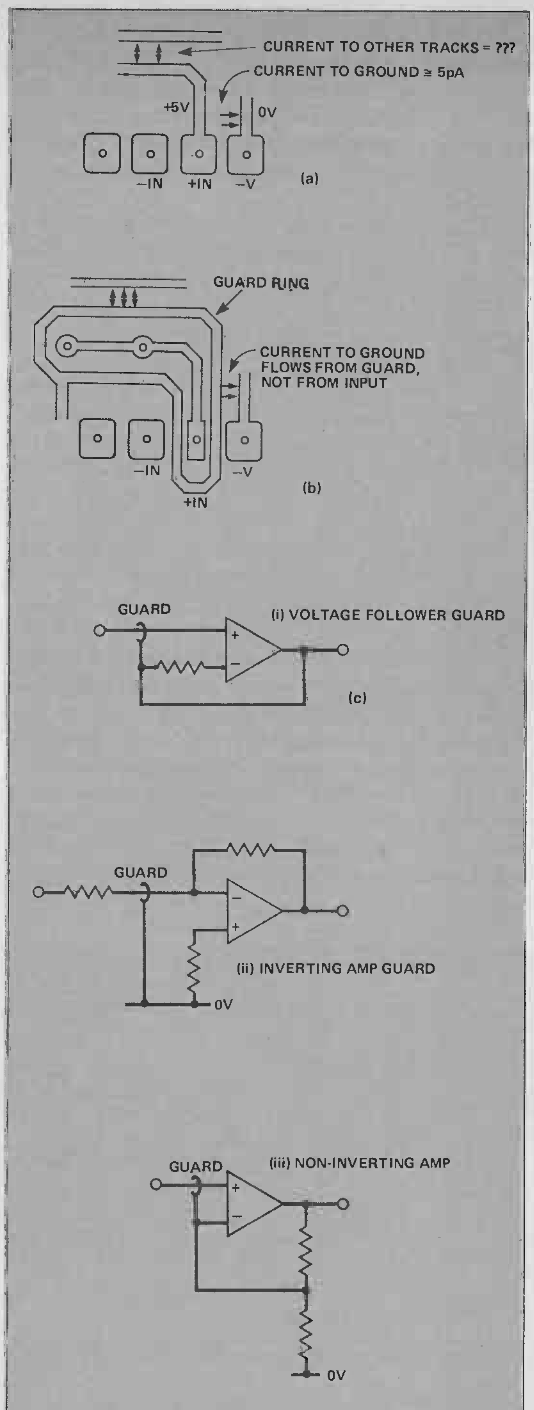


Fig. 3(a) The leakage problem
(b) The solution
(c) Driving the guards

It may seem like a long shot that there will actually be another point in the circuit that's at the same voltage as the input but because of an op-amp's action of maintaining its two inputs at the same voltage there almost always is a suitable point. Fig.3c shows where to connect a guard for various common op-amp configuration.

In the follower and inverting configurations it is usually advisable to guard both inputs — the series resistors are likely to be large making the other input vulnerable to leakage too. The non-inverting configuration is not strictly Kosher — it should have another resistor in series with the, input — but that's the configuration I've used in the Quest-Ion (for very practical reasons) so that's the way it's gonna stay.

On the PCB you'll easily find the guard ring — just start at pin 3 of IC1 and follow it around. You' ll also see that the track from the output of IC2 (pin 7)

PARTS LIST

RESISTORS (all ¼W 5% unless specified)	
R1	100k
R2-8	10M
R9	2k4 1%
R10	62k 1%
R11,22-29	680
R12,14	100R 1%
R13	5k6 1%
R15	150R 1%
R16	300R 1%
R17	510R 1%
R18	1k0R 1%
R19	3k0 1%
R20	3k1 1%
R21	6k8 1%
R30,31	4k7
CAPACITORS	
C1	1n0 ceramic
C2	100n ceramic
C3	4µ7 tantalum
C4,5	10µ miniature electrolytic or tantalum
C6	1n0 ceramic
SEMICONDUCTORS	
IC1	CA3140
IC2-6	1458
D1	1N4148
D2	BAT42
ZD1	6V8 zener
LED1	Red rectangular LED, 0.2in long
LED2	Amber rectangular LED, 0.2in long
LED3-8	Green rectangular LED, 0.2in long
MISCELLANEOUS	
B1	PP3 battery and clip
SK1	3.5mm jack socket
SW1	Miniature push-to-break switch

PCBs. 3.5mm jack plug. 32/0.2 wire and croc clip. Case.

interposes itself between the sensitive parts of the circuit and the rest, also with the intention of cutting down leakage currents.

The guard on its own is not panacea for all PCB ills. If the board is not clean (and I mean *really* clean, not just clean looking) and not insulated against moisture penetration, a lot of the good work will be undone. Don't cut corners on the board preparation when you come to build the Quest-Ion!

Construction

The component overlay for the PCB is shown in Fig.4. Before beginning, wipe the board over with a little iso-propanol, particularly the top side which you won't be able to reach once the components are in place.

Leave the LEDs till last — they are a bit tricky to put in because the lead length has to be just right to bring them flush with the top surface of the case. There should be no problem with any of the other components as long as you remember to put in the links that run under IC1 and IC2 before you solder in the ICs themselves. You'll see from the photograph that I used sockets in the prototype — this is not recommended for the final version (I removed them from mine after the photograph was taken!) because they give yet another apportunity for leakage currents to make a nuisance of themselves.

With the PCB assembled (all but the LEDs) you can turn your attention to the case top for the bar of LEDs. The position and dimensions of the slot are

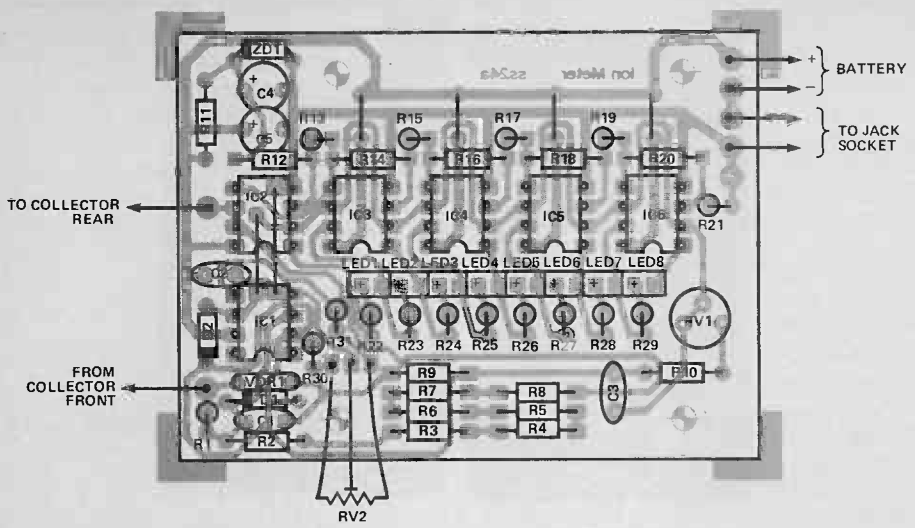


Fig. 4 Component overlay for the main PCB

shown in Fig.5.

The way I made the hole on the prototype was to use a miniature routing bit in a PCB drill. This is not ideal, since the high speed drill tended to melt rather than cut its way through the plastic, so the job had to be tidied up with a sharp modelling knife. It looks OK as long as you don't inspect it too closely!

An alternative is to use the old trick of drilling a line of holes along the slot, then cutting between them with a sharp knife. With patience you'll end up with a respectable looking hole. For the impatient there are two possible cop-outs. Either use 3mm round LEDs and drill holes 0.2in apart for them, or bodge a slot of roughly the right dimensions and make a paper or plastic label to go over the top. (The latter is a useful standby anyway in case you make a pig's breakfast of the cutting!)

The other holes are much easier: you need one for the jack socket on one side of the case lid, one for the push-switch on the other and if you follow my layout, one for RV2 on the top. My reason for putting the switch and pot in the positions I did was to make the zeroing as easy as possible — with the meter held in the palm of the left hand the switch falls comfortably under the thumb and the pot can be adjusted with the right hand without obscuring the LEDs.

If you choose different positions, watch out that the switch body doesn't foul the line of LEDs (which it will if you put it too far back) and that RV2 doesn't try to occupy the same part of the case as the jack socket. It's a basic law of physics that two objects can't be in the same place at the same time but an easy one to overlook!

The other hole in the top of the case is optional. I drilled it so that RV1 could be altered from outside the case while I was experimenting with the circuit — you might prefer to set it up and leave it alone.

My method of making round holes, by the way, is to use the standard ETI drill bit, which has been in the drill so long it's bonded in place. Then I widen them out to suit the switch or socket or whatever with a reamer. I've sung the praises of reamers before — if you haven't already got one, make this the day that you treat yourself. You'll bless the day for ever!

Back to the circuit board, your main task is now to put in the LEDs. The way I'd suggest doing it is like this. Take one of the green LEDs and holding the top and bottom sections of the case together (without the front section on) measure a length of lead so that with the top of the LED flush with the very top of the case, the leads will reach the floor of the bottom section. Cut the leads to just a teensy bit shorter than this length. Take the red LED and cut its leads to the same length. Push the red LED into the LED1 position and the green LED into the LED8 position so that their leads are only just through the PCB. Rest the PCB on

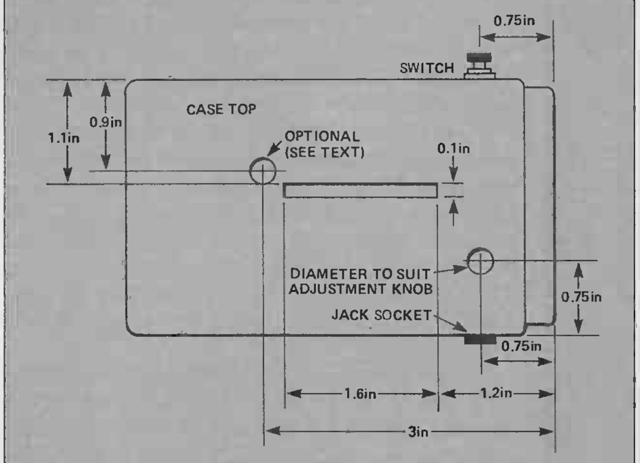


Fig. 5 Drilling the case

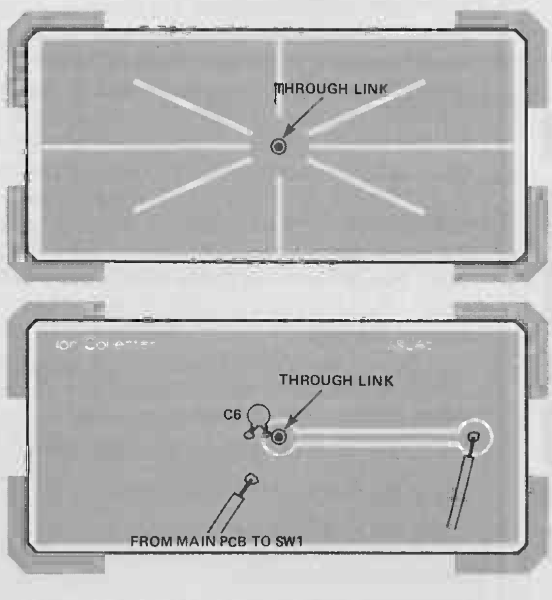
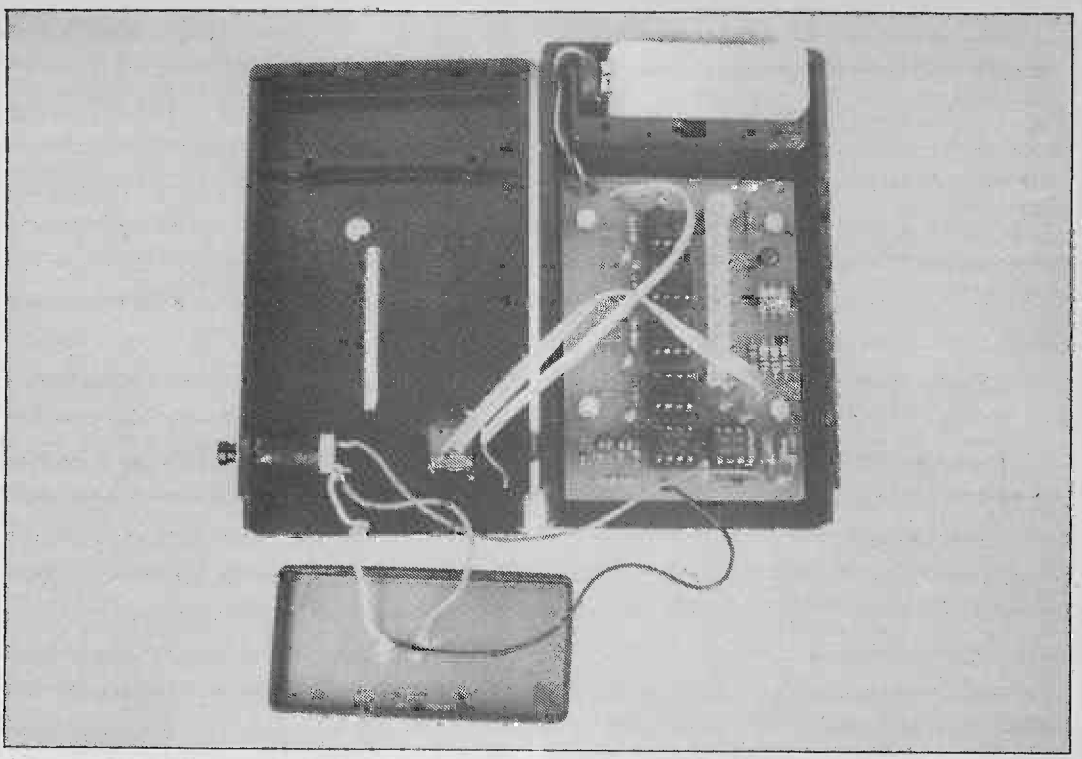


Fig. 6 The collector PCB and its connections



the supports in the bottom of the case and gently bring the lid down so that the LEDs go through the slot and, with the two case sections together, stand a little proud of the top surface. Push them down so they're flush with the surface, remove the lid and without disturbing the position of the LEDs turn the board on its side and solder them.

Put them back in the case to make absolutely sure they're in the right position (melt the solder again and adjust them a little if necessary). Push all the other LED leads through their respective PCB holes in positions LED2 to LED7, turn the board upside down on a flat surface (of course it won't balance — prop it up!) and adjust all the LEDs so that they are level. Then solder them all in place, trim the leads, and you're through.

Now for the wires: 4in of 3-way ribbon cable for RV2, 4in of 2-way ribbon for the socket, 4in of stranded wire to link switch to PCB, 2½in of stranded wire to run from PCB to rear of collection electrode, and one PP3 battery snap with its leads trimmed to about 2½ to 3 inches. Once all these are in place, it's time to pay attention to finishing the PCB.

As I pointed out earlier, the performance of the meter will depend very much on how carefully you finish off the PCB. The first thing you need to do is to clean away every trace of soldering flux, grease and dirt from the surface of the PCB. Once again, it's a job for IPA, or iso-propanol. This is a good solvent to use because it removes everything you want to remove, and at the same time is nice stuff to handle — no deadly fumes or need to wear gloves as there is with the more aggressive solvents. Don't make the mistake I did, though, and buy cloths impregnated with the stuff — they snag on the solder joints and leave hairs all over the board.

A good stiff brush is the best way to apply it. Look closely at the board and make sure you get off all traces of flux as you go. Rinse the board either with more IPA or with de-ionised water (sold in car accessory shops for topping up batteries). Dab the board dry, then leave it in a warm place for a few hours to dry out thoroughly.

For the best possible performance, the final thing is to cover the board with resin to prevent any moisture absorption. Any PCB lacquer will do, although if you've got a healthy bank balance and want to go for the best, a modified silicone conformal coating is hard to beat. Follow the instructions for whatever brand you use, spray both sides of the board, and allow enough time for the resin to dry before doing anything else to it.

While you've waiting for the main PCB to dry there are a few odd jobs you can be getting on with. Fixing the switch and socket to the case is just a matter of tightening a couple of nuts. The zeroing pot, RV1, will have to be glued. The trick here is to make sure you don't glue the knob to the body of the pot!

Don't try putting any glue on the top surface — it will just squeeze into places you'd rather it didn't go. Hold the pot against the surface of the case, then trace some glue around the outside of the pot body with a matchstick. If you lift the pot just a little away from the surface of the case, then press it back, enough glue will get underneath to form a firm bond. Quick setting Aradite gives a firm bond — you'll have to keep the pot pressed in place for a minute or two until the resin starts to go firm. After that you can leave it to cure on its own.

Making the earth lead is the next job. Use about four feet (more if you like) of 32/0.2 stranded wire. Solder one end to a crocodile clip. Remove the cover from the jack plug, thread the cover onto the wire (easy to forget!) then solder the earth wire to *both* terminals of the plug. Screw the cover back on and you've got your earth lead. It doubles as an on-off switch — plugging it in turns the meter on by connecting the negative terminal of the battery to the rest of the circuit, which is the reason for shorting the two terminals of the plug together.

The collection electrode is shown in Fig. 6. The part with the starbust pattern is the collecting side, which points outside the case. The plain side with the village pond pattern is the inside. There's nothing very complicated to do here — you need to solder a through-link to join front and rear of the PCB — this goes through the central hole and is soldered on both sides. You need a wire to transfer the current to the circuit via SW1 — about 2½in of stranded wire soldered to the surface of the other copper cicle.

There's C6 to be soldered in the position shown and finally the rear connection wire from the main PCB to be soldered anywhere onto the large copper area. Both sides can now be cleaned with IPA, and the rear (but *not* the front) of the collector can be sprayed with lacquer.

Once both the PCB and the collector are dry you can put everything into the case. Screw in the main PCB and slot the collector into the front section of the case. Connect up the wires as shown in Fig.7. Bring the top half of the case down over the bottom half, tucking the wires in as you go and guiding the LEDs through their slot. Push on the front of the case. Screw the top and bottom of the case together (inside the battery compartment), fit a PP3 battery and you've ready to go!

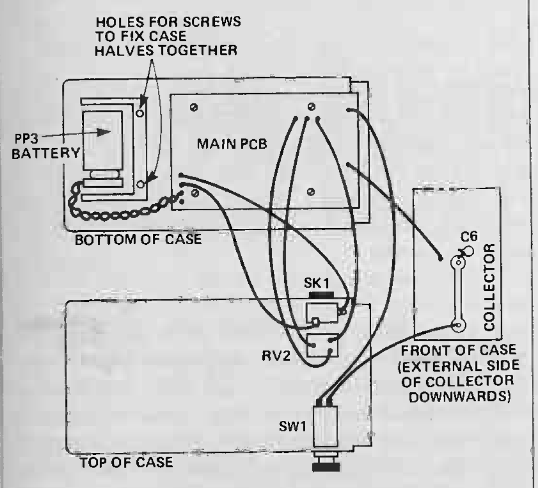


Fig. 7 Wiring it all together

Searching For Ions

The basic procedure for using the meter is this. Plug in the earth lead to turn it on and connect the croc-clip to any earthed object such as the metalwork of electrical equipment, a central heating radiator or any other grounded metal. Now press the collector disconnection switch and adjust the zeroing pot until the amber LED lights. This is best done away from any strong ion sources, or you'll discover the limits of the switch's insulation.

Since there is a slight lag in the Quest-Ion's response, the way expert meter handlers do the zeroing is to turn the pot until the LED just lights, then back it off a fraction to prevent the display drifting onto the next LED in line. When the amber LED is firmly on, release the switch and begin your measurements.

While you're getting used to the meter there are a few things you'll notice. One is a slight kick in the display, which quickly settles back, whenever you release the disconnection button. This is caused by charge building up on C6 and is a useful feature of the meter, as I'll explain in a moment. You'll also notice that the display moves if you move the meter about — with a breeze blowing on the collector it can pick up more ions but comparative measurements are best made in still air unless you want to know the effects of, say, using a fan.

Ioniser Efficiency

One way you might want to use the meter is to compare different ionisers. Close to the emitters there will be enough ions to send the reading way off scale but more important is how far the ions are travelling. If they remain in a cloud around the emitter, that's no good at all. If you take all readings at a distance of, say, three feet, you'll be able to make comparisons of the useful ion output of different brands.

If you've built your own ioniser you might still get a buzz out of comparing it with others, particularly with commercial models but you'll probably be more interested to know if there's any way to improve it. After the initial electrostatic repulsion from the emitter the ions rely on air convection to carry them around, so I'm not giving anything away by pointing out that the largest single improvement you can make is to use a fan to blow the ions where you want them. Blow over the top of the ioniser towards the Quest-Ion and up goes the reading by several points — positive proof if any were needed.

The best type of fan to use is the piezo-electric variety — they are almost silent in operation and produce a gentle breeze rather than a gale-force wind, which is exactly what you want. The Quest-Ion now comes into its own in telling you where to place the fan for maximum effect. If you have the ioniser by your bedside, for instance, you'll want the ions around your face as you sleep. Put the Quest-Ion on your pillow and adjust the position and direction of the fan until the reading is as high as you can make it.

With the internal emitter version of the Variat-Ion, the output depends very strongly on the position of the needle points relative to the holes in the rear panel. You can optimise the output by placing the Quest-Ion about three feet away from the ioniser and adjusting the board position for a maximum reading. So as not to risk shocks or attract sparks, move the board with a plastic knitting needle and keep you hands well away from all of the circuitry. If you use glue that remains liquid for a while, you can do the adjusting and gluing all in one operation.

For experiments with different emitters, go about it just the way you would compare different ionisers. Make all readings at a distance of three feet from the ioniser and see which emitter gives the highest reading.

Ion Surveys

Because of the lower limits of the Quest-Ion's sensitivity (50 million ions per second) it is a little more difficult to make readings of very low ion levels. By turning RV1 fully anti-clockwise you can increase the sensitivity by about 2½ times but that's the limit for direct reading. For lower ion levels the Quest-Ion uses C6 as an integrator: it collects ions over a period of time then displays the total.

The way you go about it is this. Push the collector disconnect button for a set period of time. One minute, say. During this period C6 will charge up as ions reach the collector — the voltage it finally reaches will depend on the number of ions and the length of time the switch is pressed. Not being connected to the PCB, C6 won't discharge through R2-8 during the integration time. Once your chosen time period has elapsed, release the switch and note the highest level the display reaches. This indicates the ion level.

Using this technique you can track down any sources of positive ions in your home or at work and see just what's in the air you're breathing. If it's not practical to get rid of the offending object (you're not going to chuck out your TV, now are you?) it will at least tell you the best site for an ioniser.

BUYLINES

A complete parts set for the Quest-Ion is available for £19.78, inclusive of postage and VAT, from Specialist Semiconductors, Founders House, Redbrook, Monmouth, Gwent NP5 5LU.

If you prefer to search out components for yourself, the PCBs are available from our PCB service, most components from your usual suppliers and any that you have difficulty with from Specialist Semiconductors.

PROJECT

ETI

ETI STAND-ALONE EPROM PROGRAMMER

Gordon Bennett racks his memory to present the complete code and software for last month's EPROM Programmer hardware

Well there you are then, the hardware all finished, the smoke all cleared, the charred bits replaced and the fire extinguisher recharged, so what happens when you plug in the system ROM and with trembling fingers switch on (you mean your fingers aren't trembling after all that? Ah! you're a stronger man than I, Gunga Din!).

The first thing should be the prompt 'rEAdy' appears on the display. It if doesn't then don't despair immediately, switch off and give it a check over — particularly the links associated with the mapping of the system VIA into the I/O page.

If it's all there try switching on again. If there's still no prompt but the display shows a lit digit (normally the one on the right) then one possibility which could give this sympton is a fault with the reset circuit on the single board controller. During software development some trouble was experienced with this not appearing to operate correctly with the above result, due to the processor not entering the ROM at the reset vector. Mike Bedford, the hardware designer, thinks its something to do with my fingers and as the fault disappeared at a later stage it probably was.

If there is still no prompt, I think it's now a good time to despair!

Looking on the bright side and assuming the prompt is there, try pressing the *Fill* button, the message 'Addr' should appear, press *Enter* and the message 'Len' should replace it, *Enter* again and 'data' should now be in the display. Another *Enter* should immediately give the message 'done' and finally a last press of *Enter* returns to the prompt 'ready'.

At this point you should be reasonably confident that the system ROM is installed and working.

Onwards by pressing the following keys and seeing what you get — *Baud, RS232, Algm, Format,* and *Device* — using the *Enter* key in between each of these to return to the 'ready' prompt.

Unless you are unbelievably lucky these, with the exception of *Algm*, will display what can best be described as rubbish. This is because the battery backed RAM will be in a random state and previously stored data (if there was any) such as set up parameters will have been lost. This will show up in the menus displaying rubbish data as the pointers (normally stored in this RAM) to their start addresses are now wrong.

One way around this would be to use the *Edit* function to set the correct values into the RAM, but this would be a long winded affair requiring the addresses and the data that goes in them to be known.

Don't worry though, there is an initialisation routine tucked away at the end of the *Algm* menu which does not lose its pointer because it is always set to *Fast* on power up. The initialisation routine will set all the other pointers to the first item in their respective menus.

The routine is entered by pressing the *Algm* button and then pressing the *Down* key, when the message INIT should be displayed, pressing *Enter* at this point will initialise all the pointers and reset the *Algm* to *Fast*. This procedure should not then be necessary again unless the RAM should lose its power for any reason.

Due to the battery backing of the RAM the user can store new set-up values into the programmer, so if development work is being done requiring a certain address range to be programmed into a set of EPROMs, this can be set up once and retained

Function	Display Messages		
Prog	Src (M)	Len	Dest (E)
Read	Src (E)	Len	Dest (M)
Verify	Src (M)	Len	Dest (E)
Copy memory	Src (M)	Len	Dest (M)
Split program for 16 bit	Src (M)	Len	Dest (E)
Input RS232	Dest (M)	Len	
Output RS232	Src (M)	Len	
Fill memory	Addr (M)	Len	Data
Search mem.	Addr (M)	Len	Data
Edit memory	Addr (M)		
Test	O/Ps checksum or ERASED message		
Device type	13 off (see table 2)		(cyclic menu)
Data format for RS232	Intel, S-rec, Binary		(cyclic menu)
RS232 comms params	No. Data and Stop bits Parity: Odd, Even, None		(cyclic menu)
Baud rate	50-19200		(cyclic menu)
Algorithm type	SLOW :PULSE :FAST		(cyclic menu)

M = source or destination address in programmer memory
E = source of destination address in EPROM

Table 1 Programmer operations and display messages

PROJECT

through power down so that it is instantly available at switch on for the next session.

Similarly all baud rates and formats are retained, the only fixed option being the *Algm* which is, as mentioned above, always set to *Fast* on power-up.

In Command

The range of commands available is shown in Table 1, together with the questions that will be asked by each routine, such as 'Src' for the source address. The letter following the source or destination in the table refers to where the address applies, if an 'M' then it refers to memory in the programmer and if an 'E', then the address is in the EPROM.

Where the message 'cyclic menu' is shown this refers to the fact that a menu is displayed by that function. A menu is stepped through by using the *up* and *down* keys and the final value left in the display when the *enter* key is pressed is the one selected.

What I propose to do is to give an explanation of what each button does when pressed and this will I hope form the necessary guide to using the programmer.

Button By Button

● *Device* — This option selects the type of EPROM to be operated upon by the other options on the top row of keys namely *Program, Read, Verify* and *Test*. The range available is shown in Table 2. The device is selected from a menu accessed by the *up, down* and *Enter* keys. After initialisation the device type is set to 2758.

● *Prog* — This asks for the source address in memory, the length of the area to be programmed and the destination address in the EPROM to which this area is to go.

From this it is obvious that the same data can be programmed to many different addresses in the EPROM and that the 'zero' position in the memory addressing range has no fixed relationship to the zero address in the EPROM.

This lack of fixed relationships leads to a vast increase in the flexibility of the machine and holds true for the *Read* and *Verify* operations as well as *Prog*. This is illustrated by Figure 1.

The MkII programmer software required data in memory to be aligned with EPROM addresses before programming, easy to implement and error check. The increased flexibility of the new arrangement leads to difficulties in error checking and it has in most cases been removed. So take care.

● *Read* — This asks for the source address in the EPROM, the length and the destination in memory to which the data will be read. It is obviously possible to read the same information into memory at different locations merely by specifying different destination addresses.

● *Verify* — This option asks for the source address in memory, the length of the data to be verified and the destination address in the EPROM. There is no checking for invalid addresses and if EPROM programming has occurred in lots of different operations, the contents of programmer memory may not reflect the state of the EPROM data at the final point.

The verification need only be used to check the success of the *Read* operation as the programming algorithms do an automatic verify of the addresses accessed in the EPROM. If you wish to run a separate verify operation on the blown EPROM you should ensure the verify is only over those addresses just programmed.

As with the READ and PROGRAM, the message 'runnIng' is displayed during operation. The message PASS appears if the data verifies and the message FAIL if it doesn't.

The FAIL message only lasts for a couple of seconds and gives way to a display of the form 'AAAAmmee' showing the address at which the failure occurred, the data read from the programmer memory and the data from the EPROM, in that order, from left to right on the display.

EPROM Type	Software Supported	Algorithm FAST/SLOW	Whether Programmed
2758	YES	S	NO
2716	YES	S	YES
2516	YES	S	YES
2732	YES	S	YES
2732A	YES	S	YES
2532	YES	S	YES
68732	NO	–	–
2764	YES	F/S	YES
2764A	YES	F/S	NO
2564	YES	F/S	NO
68764	NO	–	–
27128	YES	F/S	YES
27128A	YES	F/S	NO
27256	YES	F/S	YES
27512	YES	F/S	NO
27513	NO	–	–
2816	NO	–	–
2864	NO	–	–

Table 2 The EPROMs supported by the programmer

● *Test* — This option is slightly different in that it asks no questions but directly enters the routine and displays the message 'testing' until it has finished. It tests if the EPROM is completely blank (all the bits are set to '1').

If the test is successful then the message 'ErASEd' is displayed. If the EPROM fails the test because it holds data at one or more locations then the checksum for the whole EPROM is displayed in the form of 'Csssss'.

The 6-digit hexadecimal value is obtained by adding all the bytes in the EPROM, handy for comparing EPROMs to see if they contain any differences. Although it is possible to obtain the same checksums with data from differently coded EPROMS, if they are supposed to be the same it is definitive.

● *Split* — This toggles between 8-bit and 16-bit mode, setting a flag and displaying either the message '8-bit' or '16-bit'.

Split operation is a special case for the programmer as its use is mainly for producing EPROMs for 16-bit machines. It obviously needs two 8-bit EPROMs to cover the whole width of a 16-bit bus. This means that the date will be split up into a high byte and a low byte, all the high bytes going into one EPROM and all the low bytes into another. The data from a 16-bit machine will appear as one block of contiguous 16-bit words and this is how it should be loaded into the programmer.

When *Split* is selected the programmer will then select all the low bytes of data and flow them into contiguous locations in the first EPROM of the type

```
0000 4C 8E E4 4C BB F8 4C E1 F8 45 50 52 4F 4D 20 55
0010 74 69 6C 69 74 69 65 73 20 56 35 2E 30 28 43 29
0020 20 47 2E 4A 2E 42 45 4E 4E 45 54 54 20 31 39 38
0030 37 00 31 32 33 34 35 36 37 38 39 30 31 32 33 34
0040 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30
0050 31 32 33 34 35 36 37 38 39 30 3C 3C 3C 3C 3C 3C
0060 34 34 34 34 34 34 34 34 98 98 98 90 90 98 92 92
0070 90 92 92 90 90 90 18 18 18 08 08 18 10 10 10 10
0080 10 10 00 10 88 88 88 85 A5 88 22 C2 00 22 C2 C1
0090 C5 C7 01 01 01 01 01 10 02 02 10 02 02 01 01 01
00A0 06 06 06 06 06 0A 06 06 0A 06 06 06 06 01 01 01
00B0 01 01 01 00 01 01 00 01 01 01 01 01 03 07 07 0F
00C0 0F 0F 1F 1F 1F 3F 3F 7F FF 3F 00 00 00 00 00 00
00D0 0F 19 0F 1F 0F 19 19 19 FF FF FF FF FF FF FF FF
00E0 FF FF FF FF FF FF 02 07 05 08 FF FF FF FF 02 07
00F0 01 06 FF FF FF FF 02 05 01 06 FF FF FF FF 02 07
0100 03 02 FF FF FF FF 02 07 03 02 0A FF FF FF 02 05
0110 03 02 FF FF FF FF 02 07 06 04 FF FF FF FF 02 07
0120 06 04 0A FF FF FF 02 05 06 04 FF FF FF FF 02 07
0130 01 02 08 FF FF FF 02 07 01 02 08 0A FF FF 02 07
0140 02 05 06 FF FF FF 02 07 05 01 02 FF FF FF 02 07
0150 05 01 03 FF FF FF 31 32 33 34 35 36 37 38 39 30
0160 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36
0170 37 38 39 30 31 32 33 34 35 36 37 38 39 30 00 00
0180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0190 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
01A0 FF 03 FF 07 FF 07 FF 0F FF 0F FF 0F FF 1F FF 1F
01B0 FF 1F FF 3F FF 3F FF 7F FF FF FF FF FF FF FF FF
01C0 FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01D0 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF
01E0 FF FF FF FF 05 00 FF FF FF FF FF FF 07 05 FF FF
01F0 FF FF FF FF 01 01 00 FF FF FF FF FF 01 03 05 FF
0200 FF FF FF FF 01 05 00 FF FF FF FF FF 03 00 00 FF
0210 FF FF FF FF 06 00 00 FF FF FF FF FF 01 02 00 00
0220 FF FF FF FF 01 08 00 00 FF FF FF FF 02 04 00 00
0230 FF FF FF FF 03 06 00 00 FF FF FF FF 04 08 00 00
0240 FF FF FF FF 07 02 00 00 FF FF FF FF 09 06 00 00
0250 FF FF FF FF 01 09 02 00 00 FF FF FF FF FF FF FF
0260 FF FF FF FF 07 10 00 0D 0D 10 10 01 07 10 00 0D
0270 0D 10 10 02 07 10 0E 1C 0E 16 10 01 07 10 0E 1C
0280 0E 16 10 02 08 10 00 0D 0D 10 10 01 08 10 0E 1C
0290 0E 16 10 01 08 10 16 1B 16 0E 10 01 08 10 16 1B
02A0 16 0E 10 02 31 32 33 34 35 36 37 38 39 30 31 32
02B0 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38
02C0 39 30 31 32 33 34 35 36 37 38 39 30 26 26 66 66
02D0 26 66 06 06 FF FF FF FF FF FF FF FF 30 B0 30 B0
02E0 10 10 10 90 FF FF FF FF FF FF FF FF 0F 0A 05 17
02F0 FF FF FF FF 05 15 1B FF FF FF FF FF 1D 1C 15 05
0300 0E FF FF FF 18 16 13 17 FF FF FF FF FF FF FF FF
0310 FF FF FF FF 0B 13 16 0A 11 12 FF FF 18 16 17 0E
0320 15 FF FF FF 05 19 11 0E 0C 05 FF FF FF FF FF FF
0330 FF FF FF FF 33 30 20 10 31 21 11 32 22 12 00 01
0340 02 03 13 23 00 00 00 00 0F 00 10 20 30 0B F0 97
0350 F2 CC F1 10 F1 15 E8 8C E8 3F EC 1C EA B1 EC 4F
0360 F5 EF EA 37 ED 84 F0 80 E9 66 ED B3 EB 0F C0 00
0370 C0 80 00 80 C0 00 00 40 C0 C0 C0 C0 40 00 00 40
0380 80 40 C0 C0 40 C0 40 00 80 C0 C0 40 80 40 FF FF
0390 FF FF FF FF FF FF F0 A0 98 B8 E8 78 78 B0 F8 F8
03A0 F8 68 50 A8 58 58 00 08 E8 00 70 40 28 48 40 08
03B0 00 28 20 D8 F8 E8 FF FF FF FF FF FF FF FF 1E 11
03C0 0E 0A 0D 12 FF 05 11 0C FF 0A 17 FF 15 0E 16 FF
03D0 0D 0E 05 17 FF 0C FF 0C 1F 0A 16 1E 0E FF 17 0E
03E0 05 17 13 16 1E FF 16 1B FF 0A 0D 0D 11 FF 0D 0A
03F0 17 0A FF 1D 0A 05 05 FF 0F 0A 18 15 FF 0E 11 11
0400 1B 11 FF 0E 11 0A 05 0E 0D FF 0B 0A 1C 0D FF 05
0410 17 1B 1D FF 1D 0A 11 13 17 12 FF 0E 1C 0E 16 FF
0420 00 0D 0D FF 16 00 16 0E FF 01 06 1A 0B 13 17 FF
0430 10 08 1A 0B 13 17 FF 0D 00 16 0E FF 00 1C 17 1D
0440 1C 17 FF 18 16 1D 1C 17 FF 11 1C 16 16 13 16 14
0450 FF 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35
0460 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31
0470 32 33 34 35 36 37 38 39 30 0B 0A 16 14 FF FF FF
0480 FF FF FF FF FF FF FF FF FF FF FF FF FF FF A2 FF
0490 9A D8 58 20 D4 E4 20 A4 E5 D8 20 F1 E5 20 A4 E5
04A0 20 BA E6 A5 6F C9 01 90 F7 A5 61 0A AA BD 4D E3
04B0 85 87 E8 BD 4D E3 85 88 A9 E4 48 A9 99 48 20 31
04C0 E7 6C 87 00 FF FF FF FF FF FF FF FF FF FF FF FF
04D0 FF FF FF FF A9 30 8D 21 BC 8D 23 BC 8D 27 BC 8D
04E0 2B BC A9 FF 8D 20 BC 8D 22 BC 8D 26 BC 8D 2A BC
04F0 A9 00 85 75 85 6B 85 6C 85 6D 85 6E 85 46 85 47
0500 85 6F 85 60 85 45 A9 07 85 33 A9 20 85 76 A2 4E
0510 BD 7E E1 9D 00 04 CA 10 F7 A9 F0 8D C3 BF A9 FF
0520 8D C2 BF A9 CC 8D C1 BF A9 C2 8D CE BF A9 41 8D
0530 CB BF A9 4F 8D C6 BF A9 12 8D C7 BF 8D C5 BF 60
0540 A2 00 BD 09 E0 F0 07 9D 00 C0 E8 4C 42 E5 A2 00
0550 BD 09 E0 F0 09 DD 00 C0 D0 08 E8 4C 50 E5 A9 FF
0560 85 4D 60 FF FF FF FF FF FF FF FF FF FF FF FF FF
0570 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0580 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0590 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
05A0 FF FF FF 60 A9 04 8D 27 BC 8D 2B BC 8D 2A BC A9
05B0 90 8D 26 BC A9 34 8D 21 BC 8D 23 BC A9 00 8D 25
05C0 BC 8D 20 BC 8D 22 BC A9 FF 8D 24 BC A9 04 8D 25
05D0 BC A9 00 8D 24 BC 85 73 A6 48 BD CC E2 85 4A 29
05E0 FE 85 49 BD DC E2 05 43 85 4B E6 4B 85 4C E6 4C
05F0 60 A9 BF 85 50 A9 E3 85 51 20 1B E6 60 A0 07 A9
0600 10 99 F0 00 88 10 F8 C8 84 60 60 FF FF FF FF FF
0610 FF FF FF FF FF FF FF FF FF FF FF 20 FD E5 A0 00
0620 84 60 B1 50 C9 FF F0 0B 99 F0 00 C8 C0 08 F0 03
0630 4C 20 E6 60 20 AE F8 A9 C2 8D CE BF A5 6B C9 00
0640 D0 20 A5 6C C9 00 F0 03 20 80 E6 20 BA E6 A5 6F
0650 C9 01 D0 E3 20 6B E6 A2 04 20 31 E7 CA D0 FA 4C
0660 37 E6 A9 02 8D CE BF 20 AE F8 60 A2 07 B5 F0 48
0670 A5 61 95 F0 68 85 61 CA E4 64 F0 03 4C 6D E6 60
0680 A6 64 E8 86 37 B5 F0 48 E8 E0 07 D0 F8 68 95 F0
0690 CA E4 37 D0 F8 A9 10 95 F0 A2 04 20 31 E7 CA D0
06A0 FA 20 AE F8 A9 C2 8D CE BF 60 FF FF FF FF FF FF
06B0 FF FF FF FF FF FF FF FF FF FF A9 02 8D CE BF A9
06C0 00 85 6F A2 03 AD CF BF 29 C0 1D 49 E3 8D CF BF
06D0 A9 C2 8D CE BF 20 31 E7 AD CF BF 29 0F C9 0F F0
06E0 18 20 31 E7 20 31 E7 AD CF BF 29 0F C9 0F F0 09
06F0 A0 FF C8 4A B0 FC 4C 0A E7 A9 02 8D CE BF CA 30
0700 03 4C C5 E6 A9 C2 8D CE BF 60 86 65 06 65 06 65
0710 06 65 06 65 98 18 65 65 85 65 AC 48 E3 B9 34 E3
0720 C5 65 F0 06 88 10 F6 84 61 60 84 61 A9 01 85 6F
0730 60 48 98 48 8A 48 A2 28 A0 00 88 D0 FD CA D0 FA
0740 68 AA 68 A8 68 60 48 29 F0 4A 4A 4A 4A 85 61 20
0750 6B E6 68 29 0F 85 61 20 6B E6 60 FF FF FF FF FF
0760 FF FF FF FF FF FF FF FF FF FF FF A9 00 85 68 85
0770 69 A4 64 C8 C0 08 F0 21 B9 F0 00 C9 10 90 05 A9
0780 00 99 F0 00 A2 04 06 68 26 69 CA D0 F9 18 65 68
0790 85 68 90 02 E6 69 4C 73 E7 60 FF FF FF FF FF FF
07A0 FF FF FF FF FF FF FF FF FF FF 38 A5 71 E5 76 20
07B0 46 E7 A5 70 20 46 E7 60 A0 00 B1 70 20 46 E7 60
07C0 A0 00 A5 7B 20 46 E7 60 A9 10 85 61 20 6B E6 60
07D0 68 68 68 68 60 FF FF FF FF FF FF FF FF FF FF FF
07E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
07F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0800 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0810 FF FF FF FF FF A5 4A 8D D2 BF A5 4C 8D D3 BF 20
0820 AE EE 20 8A EE A5 3E 85 3A 18 A5 3F 65 76 85 3B
0830 E6 45 20 51 EF A9 43 85 50 A9 E4 85 51 20 1B E6
0840 A5 38 C9 01 F0 03 4C 32 E9 C9 02 F0 03 4C 6F E9
0850 20 50 F3 F0 0A 20 71 E8 A2 00 81 70 4C 50 E8 A9
0860 37 85 50 A9 E4 85 51 20 1B E6 20 8A F8 20 AE F8
0870 60 AD D1 BF 29 08 F0 F9 AD D0 BF 60 FF FF FF FF
0880 FF FF FF FF FF FF FF FF FF FF FF FF A5 49 8D D2
0890 BF A5 4B 8D D3 BF 20 67 EE A5 3A 85 3E 18 A5 3B
08A0 65 76 85 3F 20 51 EF A9 3C 85 50 A9 E4 85 51 20
08B0 1B E6 A5 38 C9 01 F0 03 4C 10 E9 C9 02 F0 23 4C
08C0 6E E9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
08D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
08E0 FF FF 20 50 F3 F0 17 A2 00 A1 70 A8 20 F2 E8 4C
08F0 E2 E8 AD D1 BF 29 10 F0 F9 98 8D D0 BF A9 37
0900 85 50 A9 E4 85 51 20 1B E6 20 8A F8 20 AE F8 60
0910 A9 3A FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0920 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0930 FF FF 20 71 E8 C9 3A D0 F9 20 71 E8 85 40 20 71
0940 E8 85 31 20 71 E8 85 30 20 71 E8 F0 01 60 A2 00
0950 20 71 E8 81 30 E6 30 D0 02 E6 31 A5 31 C5 83 D0
0960 06 A5 30 C5 82 F0 E6 C6 40 D0 E5 4C 32 E8 60 60
0970 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0980 20 FD E5 20 D9 E9 20 8A F8 A5 6D D0 0C A5 6E D0
0990 28 A5 6B F0 F1 20 AE F8 60 20 AE F8 A5 48 C9 07
09A0 F0 0D E6 48 18 A5 5E 69 08 20 DE E9 4C 86 E9 A9
09B0 00 85 48 20 DE E9 4C 86 E9 20 AE F8 A5 48 F0 0D
09C0 C6 48 38 A5 5E E9 08 20 DE E9 4C 86 E9 A9 07 85
09D0 48 A9 38 20 DE E9 4C 86 E9 A5 48 0A 0A 0A 85 5E
09E0 A9 64 85 50 A9 E2 85 51 18 A5 50 65 5E 85 50 A5
09F0 51 69 00 85 51 20 1B E6 20 48 EF 60 FF FF FF FF
0A00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0A10 FF FF FF FF FF FF FF FF FF FF FF FF FF FF E2 20
0A20 51 EF A9 00 85 4F 20 50 F3 20 D1 EA 20 C8 E7 20
0A30 C8 E7 20 48 EF A9 05 85 64 20 AE F8 A9 C2 8D CE
0A40 BF 20 BA E6 A5 6B C9 00 D0 53 A5 6C C9 00 D0 35
0A50 A5 6D F0 0A A9 00 85 4F 20 94 EA 4C 3C EA A5 6E
0A60 F0 08 85 4F 20 94 EA 4C 3C EA A5 6F C9 01 D0 CC
0A70 A9 02 8D CE BF 20 AE F8 20 6B E6 20 48 EF A9 01
0A80 85 39 4C 3C EA A5 39 C9 01 D0 09 20 80 E6 20 48
0A90 EF 4C 3C EA A9 02 8D CE BF 20 AE F8 60 A5 39 C9
0AA0 01 D0 1A 20 6B E7 20 94 EA A2 00 A5 68 81 70 20
0AB0 D1 EA A5 68 20 46 E7 20 48 EF 20 48 EF A5 4F D0
0AC0 03 4C 26 EA C6 70 A5 70 C9 FF D0 02 C6 71 4C 29
0AD0 EA 20 FD E5 A9 FF 85 64 20 AA E7 20 B8 E7 60 FF
0AE0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF A9
0AF0 DE 85 50 A9 E3 85 51 20 1B E6 A9 00 85 41 85 8A
0B00 85 89 85 88 85 3A 85 3B 85 3E 85 3F A9 FF 85 3C
0B10 A6 7D BD BC E0 E0 0C D0 0B A9 7F 20 24 EB A9 80
0B20 85 3D A9 FF 85 3D 20 51 EF A9 01 85 7C 20 E8 F1
0B30 A5 41 D0 15 A9 03 85 50 A9 E4 85 51 20 1B E6 20
0B40 48 EF 20 8A F8 20 AE F8 60 A9 D5 85 50 A9 E3 85
0B50 51 20 1B E6 A9 01 85 64 A5 8A 20 46 E7 A5 89 20
0B60 46 E7 A5 88 20 46 E7 20 48 EF 20 8A F8 20 AE F8
0B70 60 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0B80 FF A5 3E 85 58 A5 3F 85 59 A5 3C 85 56 A5 3D 85
0B90 57 A5 3A 85 54 A5 3B 85 55 60 A5 58 85 3E A5 59
0BA0 85 3F A5 56 85 3C A5 57 85 3D A5 54 85 3A A5 55
0BB0 85 3B 60 20 FD E5 20 0C EC 20 8A F8 A5 6D D0 0C
0BC0 A5 6E D0 28 A5 6B F0 F1 20 AE F8 60 20 AE F8 A5
0BD0 38 C9 02 F0 0D E6 38 18 A5 5E 69 08 20 11 EC 4C
0BE0 B9 EB A9 00 85 38 20 11 EC 4C B9 EB 20 AE F8 A5
0BF0 38 F0 0D C6 38 38 A5 5E E9 08 20 11 EC 4C B9 EB
0C00 A9 02 85 38 A9 10 20 11 EC 4C B9 EB A5 38 0A 0A
0C10 0A 85 5E A9 14 85 50 A9 E3 85 51 18 A5 50 65 5E
0C20 85 50 A5 51 69 00 85 51 20 1B E6 20 48 EF 60 FF
0C30 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 20
0C40 67 EE 20 AE EE 18 A5 3F 65 76 85 3F 20 51 EF A2
0C50 00 20 50 F3 F0 07 A1 70 81 80 4C 51 EC A9 37 85
0C60 50 A9 E4 85 51 20 1B E6 20 8A F8 20 AE F8 60 18
0C70 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0C80 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0C90 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0CA0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0CB0 FF 20 E2 EE 20 8A EE 20 06 EF 20 51 EF 20 1A EE
0CC0 A2 00 20 50 F3 F0 1B A1 70 C5 5C D0 0D C9 85
0CD0 50 A9 E3 85 51 20 1B E6 A9 03 85 64 20 AA E7 4C
0CE0 F0 EC A9 F8 85 50 A9 E3 85 51 20 1B E6 20 48 EF
0CF0 20 8A F8 20 AE F8 60 FF FF FF FF FF FF FF FF FF
0D00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0D10 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0D20 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0D30 FF FF FF FF FF FF FF FF FF FF 20 E2 EE 20 8A EE
0D40 20 06 EF 20 1A EE A2 00 20 50 F3 F0 07 A5 5D 81
0D50 70 4C 48 ED A9 37 85 50 A9 E4 85 51 20 1B E6 20
0D60 8A F8 20 AE F8 60 20 FD E5 20 E0 ED 20 8A F8 A5
0D70 6D D0 2D A5 6E D0 49 A5 6B F0 F1 20 AE F8 A9 00
0D80 85 73 A5 46 C9 01 D0 05 85 73 4C 9F ED C9 03 D0
0D90 0E A9 00 85 7D 85 48 85 38 85 46 A9 07 85 43 60
0DA0 20 AE F8 A5 46 C9 03 F0 D0 E6 46 18 A5 5E 69 08
0DB0 20 E5 ED 4C 6C ED A9 00 85 46 20 E5 ED 4C 6C ED
0DC0 20 AE F8 A5 46 F0 0D C6 46 38 A5 5E E9 08 20 E5
0DD0 ED 4C 6C ED A9 03 85 46 A9 18 20 E5 ED 4C 6C ED
0DE0 A5 46 0A 0A 0A 85 5E A9 EC 85 50 A9 E2 85 51 18
0DF0 A5 50 65 5E 85 50 A5 51 69 00 85 51 20 1B E6 20
0E00 48 EF 60 E6 7E A5 7E C9 10 60 FF FF FF FF FF FF
0E10 FF FF FF FF FF FF FF FF FF FF A9 49 85 50 A9 E4
0E20 85 51 4C 1B E6 A5 7D 0A AA A5 6F F0 0E A5 68 9D
0E30 00 04 E8 A5 69 9D 00 04 4C 46 EE BD 00 04 85 68
0E40 E8 BD 00 04 85 69 60 FF FF FF FF FF FF FF FF FF
0E50 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0E60 FF FF FF FF FF FF FF A9 C5 85 50 A9 E3 85 51 20
0E70 1B E6 20 48 EF A9 03 85 64 20 34 E6 20 6B E7 A5
0E80 68 85 3A A5 69 85 3B 20 48 EF A9 CC 85 50 A9 E3
0E90 85 51 20 1B E6 20 48 EF A9 03 85 64 20 34 E6 20
0EA0 6B E7 A5 68 85 3C A5 69 85 3D 20 48 EF 60 A9 D0
0EB0 85 50 A9 E3 85 51 20 1B E6 20 48 EF A9 03 85 64
0EC0 20 34 E6 20 6B E7 A5 68 85 3E A5 69 85 3F 20 48
0ED0 EF 60 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0EE0 FF FF A9 E9 85 50 A9 E3 85 51 20 1B E6 20 48 EF
0EF0 A9 03 85 64 20 34 E6 20 6B E7 A5 68 85 3A A5 69
0F00 85 3B 20 48 EF 60 A9 EE 85 50 A9 E3 85 51 20 1B
0F10 E6 20 48 EF A9 05 85 64 20 34 E6 20 6B E7 A5 68
0F20 85 5D 85 5C 20 48 EF 60 FF FF FF FF FF FF FF FF
0F30 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0F40 FF FF FF FF FF FF FF A2 08 20 31 E7 CA D0 FA
0F50 60 A5 45 D0 25 A5 3E 85 77 A5 3F 85 78 A5 3C 18
0F60 65 3E 85 79 A5 3D 65 3F 85 7A 18 A5 75 65 3A 85
0F70 70 A5 76 65 3B 85 71 4C 9C EF A5 3A 85 77 A5 3B
0F80 85 78 A5 3C 18 65 3A 85 79 A5 3D 65 3B 85 7A 18
0F90 A5 75 65 3E 85 70 A5 76 65 3F 85 71 38 A5 77 E9
0FA0 01 85 80 A5 78 E9 00 85 81 18 A5 79 69 01 85 82
0FB0 A5 7A 69 00 85 83 38 A5 70 E9 01 85 70 A5 71 E9
0FC0 00 85 71 A9 00 85 45 60 FF FF FF FF FF FF FF FF
0FD0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0FE0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0FF0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1000 FF FF FF FF FF FF FF FF 20 FD E5 20 61 F0 20 8A
1010 F8 A5 6D D0 0C A5 6E D0 28 A5 6B F0 F1 20 AE F8
1020 60 20 AE F8 A5 7D C9 0C F0 0D E6 7D 18 A5 5E 69
1030 08 20 66 F0 4C 0E F0 A9 00 85 7D 20 66 F0 4C 0E
1040 F0 20 AE F8 A5 7D F0 0D C6 7D 38 A5 5E E9 08 20
1050 66 F0 4C 0E F0 A9 0C 85 7D A9 60 20 66 F0 4C 0E
1060 F0 A5 7D 0A 0A 0A 85 5E A9 E6 85 50 A9 E0 85 51
1070 18 A5 50 65 5E 85 50 A5 51 69 00 85 51 20 1B E6
1080 20 48 EF 60 20 FD E5 20 DD F0 20 8A F8 A5 6D D0
1090 0C A5 6E D0 28 A5 6B F0 F1 20 AE F8 60 20 AE F8
10A0 A5 43 C9 0E F0 0D E6 43 18 A5 5E 69 08 20 E2 F0
10B0 4C 8A F0 A9 00 85 43 20 E2 F0 4C 8A F0 20 AE F8
10C0 A5 43 F0 0D C6 43 38 A5 5E E9 08 20 E2 F0 4C 8A
10D0 F0 A9 0E 85 43 A9 70 20 E2 F0 4C 8A F0 A5 43 0A
10E0 0A 0A 85 5E A9 E4 85 50 A9 E1 85 51 18 A5 50 65
10F0 5E 85 50 A5 51 69 00 85 51 20 1B E6 20 48 EF 60
1100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1110 A5 47 F0 03 4C 7B F5 20 67 EE 20 AE EE 20 51 EF
1120 20 1A EE 20 11 F5 A9 80 85 7C 20 E8 F1 20 62 F3
1130 D0 15 20 A4 E5 A9 F3 85 50 A9 E3 85 51 20 1B E6
1140 20 8A F8 20 AE F8 60 A9 F8 85 50 A9 E3 85 51 20
1150 1B E6 20 48 EF A9 FF 85 64 20 AA E7 20 B8 E7 20
1160 C0 E7 20 8A F8 20 AE F8 60 FF FF FF FF FF FF FF
1170 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1180 FF FF FF FF FF FF FF FF FF A6 7D BD 5A E0 8D 23
1190 BC A9 00 8D 20 BC A9 04 8D 25 BC BD 68 E0 8D 26
11A0 BC A9 12 8D 2A BC 60 FF FF FF FF FF FF FF FF FF
11B0 FF FF FF FF FF FF FF 78 AD 2A BC 49 10 8D 2A BC
11C0 20 7D F3 AD 24 BC 85 7B A2 00 58 60 A5 47 F0 03
11D0 4C F0 F5 20 67 EE 20 AE EE E6 45 20 51 EF 20 1A
11E0 EE 20 11 F5 A9 00 85 7C A2 00 20 99 F3 20 89 F1
11F0 20 83 F3 20 50 F3 F0 29 20 EF F2 20 83 F3 20 B7
1200 F1 A4 7C F0 17 30 0E 20 32 F2 C9 FF F0 DF A9 01
1210 85 41 4C ED F1 C1 70 F0 D4 4C 21 F2 81 70 4C ED
1220 F1 60 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1230 FF FF 18 65 88 85 88 A5 89 69 00 85 89 A5 8A 69
1240 00 85 8A A5 7B 60 FF FF FF FF FF FF FF FF FF FF
1250 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1260 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1270 FF FF FF FF FF FF 78 A6 7D BD 5A E0 8D 23 BC A9
1280 00 8D 20 BC A9 04 8D 25 BC BD 76 E0 8D 2A BC BD
1290 84 E0 8D 26 BC 58 60 A5 47 F0 03 4C 8F F6 20 67
12A0 EE 20 AE EE 20 51 EF 20 1A EE 20 11 F5 A5 7D C9
12B0 06 90 03 4C C5 F3 A9 01 85 73 A2 FF 20 99 F3 20
12C0 76 F2 20 D9 F4 20 7D F3 20 50 F3 F0 1F 20 EF F2
12D0 A2 00 A1 70 8D 24 BC 78 20 A7 F3 A0 21 A2 FF CA
12E0 D0 FD 88 D0 FA 20 A7 F3 58 4C BF F2 4C 5C F4 A5
12F0 80 8D 22 BC BD AE E0 F0 08 A5 81 8D 20 BC 4C 25
1300 F3 AD 20 BC 05 84 8D 20 BC A5 81 29 10 F0 08 AD
1310 20 BC 09 08 8D 20 BC A5 81 29 08 F0 08 AD 26 BC
1320 09 01 8D 26 BC A5 81 29 20 F0 08 AD 23 BC 09 08
1330 8D 23 BC A5 81 29 40 F0 08 AD 26 BC 09 02 8D 26
1340 BC A5 81 29 80 F0 08 AD 20 BC 09 80 8D 20 BC 60
1350 E6 70 D0 02 E6 71 E6 80 D0 02 E6 81 A5 81 29 E7
1360 85 84 A5 80 C5 82 D0 04 A5 81 C5 83 60 FF FF FF
1370 FF FF FF FF FF FF FF FF FF FF FF FF FF A0 40 88
1380 D0 FD 60 A0 20 88 D0 FD 60 FF FF FF FF FF FF FF
1390 FF FF FF FF FF FF FF FF FF A9 00 8D 25 BC 8E 24
13A0 BC A9 04 8D 25 BC 60 A6 7D BC A0 E0 B9 20 BC 5D
13B0 92 E0 99 20 BC 60 A2 FF 20 99 F3 20 76 F2 20 D1
13C0 F4 60 4C 5C F4 A2 00 86 85 86 73 E8 86 86 20 50
13D0 F3 20 62 F3 F0 EC 20 B6 F3 20 7D F3 20 EF F2 A2
13E0 00 A1 70 8D 24 BC 20 BE F4 A6 85 E8 86 85 A6 7D
13F0 BD CA E0 C5 85 F0 05 20 FA F4 D0 DA 20 B6 F3 20
1400 7D F3 20 EF F2 A5 85 0A 0A A8 A6 7D E0 0C D0 05
1410 A0 02 4C 21 F4 BD CA E0 C9 0F F0 05 98 38 E5 85
1420 A8 84 86 20 BE F4 A6 7D BD CA E0 C5 85 D0 96 20
1430 FA F4 F0 91 20 F1 F4 20 A4 E5 A9 79 85 50 A9 E4
1440 85 51 20 1B E6 20 48 EF 20 99 F4 60 FF FF FF FF
1450 FF FF FF FF FF FF FF FF FF FF FF FF 20 F1 F4 AD
1460 2A BC 29 12 8D 2A BC A9 37 85 50 A9 E4 85 51 20
1470 1B E6 20 51 EF 20 11 F5 A9 80 85 7C 20 E8 F1 20
1480 62 F3 D0 15 20 A4 E5 A9 F3 85 50 A9 E3 85 51 20
1490 1B E6 20 8A F8 20 AE F8 60 20 A4 E5 A9 F8 85 50
14A0 A9 E3 85 51 20 1B E6 20 48 EF A9 FF 85 64 20 AA
14B0 E7 20 B8 E7 20 C0 E7 20 8A F8 20 AE F8 60 78 20
14C0 A7 F3 A4 86 A2 B0 CA D0 FD 88 D0 F8 20 A7 F3 58
14D0 60 AD 26 BC 29 7F 8D 26 BC A5 7D C9 0C F0 09 AD
14E0 2A BC 09 1B 8D 2A BC 60 AD 2A BC 09 1A 8D 2A BC
14F0 60 AD 26 BC 09 80 8D 26 BC 60 A2 00 20 99 F3 20
1500 89 F1 20 F1 F4 20 83 F3 20 EF F2 20 B7 F1 C1 70
1510 60 A6 7D BD BC E0 C5 7A B0 14 A9 FD 85 50 A9 E3
1520 85 51 20 1B E6 20 8A F8 20 A5 F8 4C D2 E7 60 FF
1530 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1540 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF A5
1550 47 D0 12 A9 01 85 47 A9 E4 85 51 A9 29 85 50 20
1560 1B E6 4C 74 F5 A9 00 85 47 A9 E4 85 51 A9 30 85
1570 50 20 1B E6 20 8A F8 20 AE F8 60 20 67 EE 20 AE
1580 EE 20 51 EF 20 A9 EF 20 1A EE 20 11 F5 A9 80 85
1590 7C 20 08 F6 20 62 F3 D0 15 20 A4 E5 A9 F3 85 50
15A0 A9 E3 85 51 20 1B E6 20 8A F8 20 AE F8 60 A9 F8
15B0 85 50 A9 E3 85 51 20 1B E6 20 48 EF A9 FF 85 64
15C0 20 AA E7 20 B8 E7 20 C0 E7 20 8A F8 20 AE F8 60
15D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
15E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
15F0 20 67 EE 20 AE EE E6 45 20 51 EF 20 A9 EF 20 1A
1600 EE 20 11 F5 A9 00 85 7C A2 00 20 99 F3 20 89 F1
1610 20 83 F3 20 50 F3 F0 2F E6 70 D0 02 E6 71 20 EF
1620 F2 20 83 F3 20 B7 F1 A4 7C F0 17 30 0E 20 32 F2
1630 C9 FF F0 D9 A9 01 85 41 4C 0D F6 C1 70 F0 CE 4C
1640 6A F6 81 70 4C 0D F6 A5 42 D0 1F A9 D7 85 50 A9
1650 E3 85 51 20 1B E6 20 8A F8 20 AE F8 E6 42 20 51
1660 EF 20 A9 EF 20 50 F3 4C 08 F6 A9 00 85 42 60 FF
1670 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1680 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 20
1690 67 EE 20 AE EE 20 51 EF 20 A9 EF 20 1A EE 20 11
16A0 F5 A5 7D C9 06 90 03 4C 49 F7 A9 01 85 73 A2 FF
16B0 20 99 F3 20 76 F2 20 D9 F4 20 7D F3 20 50 F3 F0
16C0 25 E6 70 D0 02 E6 71 20 EF F2 A2 00 A1 70 8D 24
16D0 BC 78 20 A7 F3 A0 21 A2 FF CA D0 FD 88 D0 FA 20
16E0 A7 F3 58 4C B3 F6 20 E3 F7 A5 42 D0 1F A9 D7 85
16F0 50 A9 E3 85 51 20 1B E6 20 8A F8 20 AE F8 E6 42
1700 20 51 EF 20 A9 EF 20 50 F3 4C AA F6 A9 00 85 42
1710 60 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1720 FF A5 42 D0 1F A9 D7 85 50 A9 E3 85 51 20 1B E6
1730 20 8A F8 20 AE F8 E6 42 20 51 EF 20 A9 EF 20 50
1740 F3 4C 49 F7 A9 00 85 42 60 A2 00 86 85 86 73 E8
1750 86 86 20 50 F3 F0 CA E6 70 D0 02 E6 71 20 B6 F3
1760 20 7D F3 20 EF F2 A2 00 A1 70 8D 24 BC 20 BE F4
1770 A6 85 E8 86 85 A6 7D BD CA E0 C5 85 F0 05 20 FA
1780 F4 D0 DA 20 B6 F3 20 7D F3 20 EF F2 A5 85 0A 0A
1790 A8 A6 7D E0 0C D0 05 A0 02 4C 21 F4 BD CA E0 C9
17A0 0F F0 05 98 38 E5 85 A8 84 86 20 BE F4 A6 7D BD
17B0 CA E0 C5 85 D0 93 20 FA F4 F0 8E 20 F1 F4 20 A4
17C0 E5 A9 79 85 50 A9 E4 85 51 20 1B E6 20 48 EF 20
17D0 25 F8 60 FF FF FF FF FF FF FF FF FF FF FF FF FF
17E0 FF FF FF 20 F1 F4 AD 2A BC 29 12 8D 2A BC A9 37
17F0 85 50 A9 E4 85 51 20 1B E6 20 51 EF 20 A9 EF 20
1800 11 F5 A9 80 85 7C 20 08 F6 20 62 F3 D0 17 20 A4
1810 E5 A9 F3 85 50 A9 E3 85 51 20 1B E6 20 8A F8 20
1820 AE F8 4C 21 F7 20 A4 E5 A9 F8 85 50 A9 E3 85 51
1830 20 1B E6 20 48 EF A9 FF 85 64 20 AA E7 20 B8 E7
1840 20 C0 E7 20 8A F8 20 AE F8 60 FF FF FF FF FF FF
1850 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1860 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1870 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1880 FF FF FF FF FF FF FF FF FF FF 20 AE F8 A9 C2 8D
1890 CE BF 20 BA E6 A5 6D D0 0F A5 6E D0 0B A5 6C D0
18A0 07 A5 6B D0 03 4C 8A F8 A9 02 8D CE BF 60 A9 00
18B0 85 6D 85 6E 85 6C 85 6B 85 39 60 85 34 D8 A5 34
18C0 40 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
18D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
18E0 FF 48 98 48 8A 48 D8 AD CD BF 30 03 4C 78 F9 29
18F0 02 D0 13 4C 49 F9 FF FF FF FF FF FF FF FF FF FF
1900 FF FF FF FF FF FF 2D CE BF F0 3E 68 48 AA A9 FF
1910 E0 00 D0 05 85 6B 4C 31 F9 E0 01 D0 05 85 6C 4C
1920 31 F9 E0 02 D0 05 85 6E 4C 31 F9 E0 03 D0 02 85
1930 6D A9 02 8D CE BF 4C 78 F9 FF FF FF FF FF FF FF
1940 FF FF FF FF FF FF FF FF FF A4 33 B9 F0 00 AA 98
1950 8D C0 BF AD CF BF 29 30 8D CF BF BD 96 E3 85 35
1960 AD C0 BF 05 35 8D C0 BF BD 6E E3 0D CF BF 8D CF
1970 BF 88 10 02 A0 07 84 33 AD CD BF 8D CD BF 68 AA
1980 68 A8 68 40 FF FF FF FF FF FF FF FF FF FF FF FF
1FF0 FF FF FF FF FF FF FF FF FF FF 03 E0 00 E0 06 E0
```

Listing 1 The used portion of the 2764 control EPROM

NEWS

LOW RISC

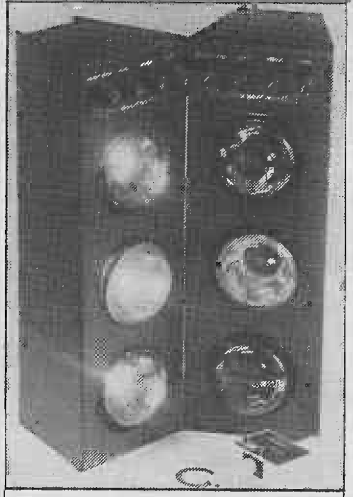
Riscomp of Buckinghamshire is busting burglars with the best of them with its new CPU9000 PIR security system.

The unit is designed to require no installation procedures, so that it may simply be placed in a convenient position. As long as the unit has an adequate view of the area to be protected, all the operator needs to do is switch on and get out.

The passive infra-red detector can spot a moving hot thing at 30 feet and triggers a 103dB siren. It can be powered from the mains or from 12V DC battery.

The CPU 9000 costs £67.72 + VAT. Contact Riscomp, 51 Poppy Road, Princes Risborough, Bucks HP17 9DB. Tel: (084 44) 6326.

DESKTOP DISCO



Voltmace (better known for an excellent range of BBC micro joysticks) has taken the old favourite sound-to-light home disco light units to new extremes with the Discoflex.

Retailing at £29.95 the Discoflex looks for all the world like a standard three channel sound-to-light system with a built-in microphone and three spotlamp sockets. But for another £19.95 you can add an Isodata box to the Discoflex to interface the unit having any micro with an 8-bit user port. The Isodata unit provides opto-isolation of the computer from the mains and enables you to flash or sequence the lights to your (or your computer's) heart's content.

Further details from Voltmace, Unit 9 Bondor Business Centre, London Road, Baldock SG7 6NG. Tel: (0426) 894410.

VIDEO BY PHONE

Panasonic is soon to market an impressive variation on the idea of facsimile — pictures sent over the phone. Rather than use only flat drawings and printed paper, the WG-R2 digitises an image with a video camera. A still video image can then be sent down an ordinary phone line to another machine at the other end in about ten seconds.

The WG-R2 is on sale in Japan and should be available in this country some time later this year. The unit (without a video camera) is expected to retail for about £200.

As well as sending facsimiles, the system is expected to find uses for verifying the identity of a caller for security sensitive calls and even for forming the basis of a security alarm system.

Panasonic is on (0753) 73181.

BONEX BONANZA

A brand new catalogue of components and kits from Bonex is now available to trickle the fancy of many an electronics designer and enthusiast.

The Bonex catalogue features 140 pages of passive components, hardware and semiconductors along with a wide range of kits — mainly in the amateur radio field.

The 1989 catalogue costs £1.50 from Bonex, 12 Elder Way, Langley Business Park, Slough SL3 6EP. Tel: (0753) 49502.

MORE RGB PCBs

Labcenter Electronics has upgraded the remarkably cheap PC-B package for the Amstrad PC1512 and other IBM PC compatible micros. The new PC-B Pro is mouse based with a large and user extendable package library, configurable object sizes, back netlist generation and many more features.

PC-B Pro requires 512K RAM and supports the EGA and VGA graphics adapters and Amstrad's own 16-colour graphics. The package costs £230.

In case that isn't enough for your creative urges, Labcenter has also produced an autorouting package for PC-B Pro. PC-B AR can operate with empty or partially routed single or double sided boards with a rats nest or netlist as input. The user can intervene at any time to stamp his mark on the process.

PC-B AR requires a PC with 640K RAM and also supports EGA, VGA and Amstrad 16-colour displays.

PC-B AR costs £170 if bought along with PC-B Pro or £180 if bought as a later upgrade.

Full details from Labcenter Electronics, 14 Marriner's Drive, Bradford BD9 4JT. Tel: (0274) 542868.

For owners of the humbler Amstrad PCW range of micros, Harlech Electronics has recently launched a drafting program capable of PCB track layout or producing schematic drawings. The program costs £29.95.

For Spectrum owners, Harlech has an Electronics Calculator package capable of taking the grind out of a myriad electronic mathematical chores. The program simulates inductors, capacitors, resistors in series and parallel, transformers and tuned circuits and handles frequency to wavelength conversion and coil turns inductance, all for just £6.50.

Harlech Electronics is at Noddfa, Lower Road, Harlech, Gwynedd LL46 2UB.

CHEAP CHANNELS COMING

Many major companies have entered the budget end of the satellite TV receiver market to compete with Amstrad's much vaunted £199 receiver for Murdoch's Astra-based channels.

Sir Clive Sinclair's Cambridge Computer is to produce a system using a 60cm flat square 'dish' similar (though using different principles) to the BSB system for just £149.95.

Cambridge Computer claims this will pick up Astra signals anywhere in the UK. A £179.95 version will include remote control and a £299.95 version stereo sound and a graphic equaliser.

It is likely the receivers and antennas will be made by SCI in Scotland — the current manufacturers of Cambridge Computer's Z88 and will be available 'early in 1989' (but haven't we heard this before somewhere . . .).

Ferguson is promising equipment for both Astra and the BSB satellite but as yet is saying little except that a receiver for both satellites, with built-in D-MAC decoder will be available. Micro-X has narrowly undercut Amstrad with a basic system for £190 and a tracking system for reception from both satellites for £280 available soon. The Micro-X system is based on existing electronics used in current up-market systems.

Grundig is another manufacturer not new to STV. The STR20 package will cost £349 and although fixed, can be upgraded with a polarator for £30 and various decoders/descramblers in the future.

There are also systems promised from Solara (for £300, aimed at DBS but without a D-MAC decoder at present), from Alba (£200-300 aimed at Astra viewers) and from NEC. NEC's system is based on exciting NEC equipment and will cost £699. A cut-down version for £399 is also promised.

Meanwhile, things are far from static on the programme providers front. Rupert Murdoch is now planning to scramble at least one of his four Astra chanels (the Sky Film channel, and possibility Eurosport the sports channel as well).

A monthly fee of £12 is envisaged for the film channel but that will also include the subscription to the Disney channel, the cartoon and general entertainment channel with four million subscribers in the US.

The Disney channel gives Murdoch's five channel Astra package access to an impressive collection of films (the real audience attraction) from Century Fox (a Murdoch owned company) and from Disney and Murdoch has also added yet another (clear) channel of 'classic' films to his Astra empire.

Disney hopes its own channel will help to promote the soon-to-be completed European Disneyland in Paris.

Meanwhile BSB has clinched an 85 million pound deal with Columbia Pictures to use all its films on the direct broadcast satellite subscription film channel. Along with deals already struck with other studios such as Cannon, MGM, Universal and Warner, this move gives BSB a reputed 850 films to choose from.

To add to the satellite war, Robert Maxwell has, at the time of writing, all but signed his MTV music channel and a new European news channel to Astra, making nine English speaking channels on that satellite.

SWIFT SATELLITE SETUP

Budding DIY Satellite TV installers can find a comprehensive but easy-to-read guide to installation of satellite dish antenna in a publication from Swift Satellite TV Services.

The 50-page booklet covers a wide range of installation procedures and is aimed specifically at the UK with special emphasis on Astra and DBS reception. The book has been adopted by the Confederation of Aerial Industries.

Called 'Satellite Television Installation Guide', the book is available for £8.95 plus 95p postage from Swift Television Publications, 17 Pittsfield, Cricklade, Swindon SN6 6AN.

HOW IT WORKS

For those people who feel that the only real way to build a project is from the bottom up and thus decide to take on the task of building their own control ROM for the programmer, here is a brief description of the main keyboard and display routines to indicate how I did it. This is followed by the address of a good psychiatrist, whose services you will find essential before the end.

The main keypad of the programmer operates in a polling mode and is only read when a character is required by the function in operation at that time. Mostly this is when data is to be input and the program is waiting for data keys to be pressed.

The other time is when the rEAdy prompt is on display and a command key is required to initiate an operation.

The keypad returns codes to the program that are produced by sending a mask byte to one of the ports causing the 74LS139 decoder to play a low on one of its output lines. The value of this mask is stored in the X register and corresponds to a vertical column on the keypad. Any key pressed will put one of the horizontal rows low and these are connected to another port to produce a byte value as a unique code for any key.

The layout of the keypad and the method of forming the codes is shown in Fig. 1.

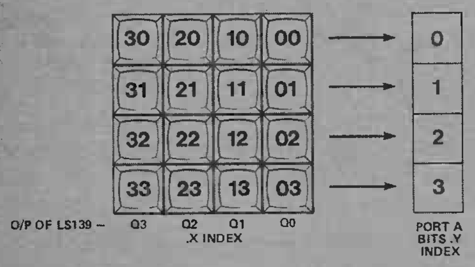


Fig. 1 Code forming from the keypad

From this is can be seen that the codes are formed from a four bit mask value in X used for the hi nibble and another four bit value read into Y from port A and used as the lo nibble to form a byte code which will be one of those shown here.

CODE	33	30	20	10	31	21	11	32	22	12	00	01	02	03	13	23
INDEX/VALUE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

The interpretation of these codes is different, depending on whether the programmer is waiting for a command of alphanumeric input. In the case of the command the index if used to access a table of command sub-routine addresses, otherwise it is used directly.

The four keys at the botton of the keypad (*Up, Down, Abort* and *Enter*) are used in a slightly different way by the program in that they generate interrupts. When one of these keys is pressed, an interrupt is generated, the processor stops executing the program at the end of the current instruction and goes to the Interrupt Service Routine (ISR). Once there a check is made to see if this is a keyboard interrupt, if so the keyboard portion of the routine is entered. There the key pressed is identified and a flag set to indicate which it was. The ISR is then left and control returned to the routine that was interrupted. This routine must check for the flags corresponding to the keys in which it is interested and take the appropriate action.

The display is also driven from the ISR. On initial power up of the programmer the control program executes an initialisation subroutine that sets up the VIA timer on the main processor board to interrupt at regular 5ms intervals. When these occur the ISR code checks that this is not a keyboard interrupt and then enters the display section.

The display section of the ISR obtains a index value from the display buffer used to the tables shown below, containing the bytes required to be sent to the system VIA ports to light the segments of the currently selected digit on the display. The display count is then decremented or reset to seven (if at zero) ready for the next display interrupt.

The table used for the display O/P are as follows (there are two as the display segments are split between two ports and a quick count will show that there are more than just the alphanumerics available).

TABLE A

C0	00	C0	80	00	80	C0	00	C0	00	40	C0	C0	C0	C0	40
00	40	80	40	C0	C0	40	C0	40	00	80	C0	C0	40	80	40

TABLE B

F0	A0	98	B8	E8	78	78	B0	F8	F8	F8	68	50	A8	58	58
00	08	E8	00	70	40	28	48	40	08	00	28	20	D8	F8	E8

Only one display is serviced on each call to the ISR, this takes approx 100ms ensuring that the minimum time is spent with the processor locked up in the interrupt service routine.

The main channel of communication in and out of the programmer occurs through the display buffer, an area of eight bytes in memory that contain anything to appear on the LEDs.

Command sub-routines are responsible for obtaining values from (and outputting their status messages and results to) the display buffer, and in general if an input is expected by the programmer, whatever is on the display when the ENTER key is pressed will be used.

---

selected, and then prompt for a change of EPROM with the message 'ChAngE' and blow all the high bytes into this.

It is important to understand that no matter if the block of data is loaded into the programmer on an even numbered address or an odd numbered address, the first byte is treated as the low byte and the second byte is the high byte and so on up through the memory. Because of this it is probably wise to always load blocks of data for use with split on even word boundaries.

The *Read* and *Verify* options also operate when in split mode, in a similar way as that described for programming.

Talking To The Host

The main method of getting data in and out of the programmer is along the RS232C lines to a host computer, using the *In* and *Out* functions.

Routine *In* will prompt for the destination address in memory where the code is to be loaded and the length of the file to be input (if the length is not known then you may have a problem with hex files).

The *Out* routine will send the file in the format selected to the host computer and requires only the source address of the data in memory and the length of the data to be sent.

Before using the *In* and *Out* options the *RS232* and *Baud* options should be selected to set up the serial command and control words for the 6551, and the *Format* for send or receive.

- *RS232* — This option allows the setting of the number of bits in the word (currently 7 or 8), the type of parity (odd, even or none) and the number of stop bits (1 or 2). These options are listed in Table 3 showing the form they take on the display.

There are only eight options implemented at present, these being a subset of those available using the 6551 Asynchronous Communications Interface Adapter (ACIA) and corresponding to those obtainable from the 6850 ACIA — these should satisfy most situations.

Directly after initialisation the RS232 parameters are set to 7 bits, odd parity and 1 stop bit.

- *Baud* — Baud rates from 50 to 19200 are listed in Table 4. The default rate is 1200 baud. There may be problems with the higher rates as there is no Xon/Xoff protocol implemented — the only control line on the RS232 port is the DSR control into the programmer.

Word Length	Parity	Stop Bits	Message Displayed
7-bit word	Odd parity	1 stop bit	7 Odd 1 (default)
7-bit word	Odd parity	2 stop bits	7 Odd 2
7-bit word	Even parity	1 stop bit	7 EuEn 1
7-bit word	Even parity	2 stop bits	7 EuEn 2
8-bit word	Odd parity	1 stop bit	8 Odd 1
8-bit word	Even parity	1 stop bit	8 EuEn 1
8-bit word	Odd parity	1 stop bit	8 nonE 1
8-bit word	Odd parity	2 stop bits	8 nonE 2

Table 3 The RS232 communications parameters available

50	baud
75	baud
110	baud
135	baud
150	baud
300	baud
600	baud
1200	baud (default)
1800	baud
2400	baud
3600	baud
4800	baud
7200	baud
9600	baud
19200	baud

Table 4 Available baud rates

I/O I/O, It's Off To Work We Go

The programmer is capable of receiving and sending data to and from a host computer in three different formats — plain hexadecimal with no data checking, Intel encoded data with checksum and Motorola S record data with checksum.

These are selected by the *format* key in conjunction with the *up, down* and *enter* keys, as shown in Table 6. On initialisation the format is set to 'binary' as the default.

The binary data consists of a stream of hexadecimal bytes placed directly into memory until the specified length of data has been received. Sending binary is similar in that a block of memory is sent as normal bytes until the entered length has been transmitted.

No checks are made to see that the data is valid either in reception or transmission, other than those carried out by the UART as part of its internal operation.

The INTEL data format is:

: nn aaaa tt dd . . . . dd cc <CR>

where:

- : — delimiter character that starts each record
- nn — count of data bytes in the following field
- aaaa — load address of the first data byte
- tt — type byte (0 = data record, 1 = termination)
- dd . dd — data bytes equal in number to nn
- cc — two's complement checksum byte (ex. :)
- <CR> — carriage return character

If the checksum byte at the end of the data does not match that derived from the data an error message is displayed. If the data is longer than the length input to the routine the data is stopped at that point in memory where the length is exceeded.

The S Record format is slightly more complicated, there being a number of different types of record to indicate the header, data and end of data points.

A 'S0' record is the header. The data following this has 'S1' records for 8-bit data, withe the terminator being a 'S9' record. Other record types are in use with larger address fields and some special uses, but the programmer only recognises S0, S1 and S9 types, the others are accepted without error but are ignored.

The S RECORD data format is:

Sx nn aaaa dd . . . . dd cc <CR>

where:

- S — delimiter character that starts each record
- x — record type (0 = start, 1 = data, 9 = end)
- nn — count bytes to follow (aaaa + dd . dd + cc)
- aaaa — load address of the first data byte
- dd . dd — data bytes
- cc — one's complement checksum byte (ex. Sx)
- <CR> — carriage return character

Data for decoding and encoding is normally placed in a buffer area from which it can be transferred into memory. The checksum and address can be calculated before transmission. The standard number of data bytes sent by the programmer in each record is fixed at 32, wrapped up with address, byte count and checksum before transmission.

The possible length of record is dependent on the length of buffer reserved. This is currently one page of 256 bytes, so providing the record (including all delineator and checksum bytes) is equal to or less than this it should be received correctly.

Changing The Code

The *EDIT, FILL* and *COPY* options together with *SEARCH* form the main set of commands for the programmer to modify copy and delete sections of code in memory.

The EDIT function allows bytes in the memory to be changed and is highly time effective if only a few changes are required to the code. However it does not allow insertion of new bytes or removal of an old one, merely the changing of what is present to begin with.

When first entered, *EDIT* asks for the address at which you wish to start editing. This is displayed in the four digits on the left of the display and the next two digits display the data found in memory at this address. The last two are blank, waiting for input from the user. Pressing the *Abort* key at this point will cause an exit from the editor and return you to the rEADy

Message	Display	Comment
msg1:	rEAdy	The main prompt
msg2:	SrC	Request for a source address
msg3:	At	Byte found 'At'
msg4:	LEn	Request for code length
msg5:	dESt	Request for a destination address
msg6:	C ssssss	Checksum display
msg7:	CHAnge	Request to change EPROM (split)
msg8:	tESting	Test function display
msg9:	no	You're doing something naughty
msgA:	Addr	Request for an Addr word
msgB:	dAtA	Request for data byte
msgC:	PASS	Pass verify
msgD:	FAIL	Fail verify
msgE:	Error-	You can't do what you just did
msgF:	ErASEd	EPROM erased correctly
msg10:	bAud	Baud rate
msg11:	StoP	You Aborted something
msg12:	PArity	Parity setting
msg13:	EuEn	Even parity
msg14:	Odd	Odd parity
msg15:	nonE	No parity
msg16:	16-blt	16 bit split mode set
msg17:	8-blt	8 bit normal mode set
msg18:	donE	Operation is finished
msg19:	OutPut	I/O link is outputting data
msg1A:	InPut	I/O link is receiving data
msg1B:	runnIng	Function is currently operating
msgFF:	bAng	Fast programming algorithm outran its counter, EPROM faulty

Table 5 Message displays and explanation

PROJECT

prompt with no change made to the data in memory. Pressing *Down* will set the direction that the editor will travel through memory, so that it moves to lower addresses on successive bytes. The default is *Up* to higher addresses.

Pressing *Enter* will display the next byte in memory either above or below the currently displayed one, depending on whether the *Down* key has been pressed or not.

Once the data has been entered the *Abort* key will act as a *Delete* key to remove erroneous characters. Pressing *Enter* now puts the currently displayed data into memory then updates the display from memory to show that it changed correctly and after a short pause (about a second) moves on to the next location.

Fill asks for the address in memory from which the fill operation is going to start and the length of the block to be filled. Then the data byte to be used for filling is requested — any byte value that is required from &00 to &FF. In most cases it will be &FF to make sure that nothing will be blown into the EPROM where no code is required.

'Running' is displayed and 'done' on completion — press *Enter* to return to main prompt.

Copy prompts for the source address in memory from which you want to copy, the length of the block of data and the destination address to which the block is to be copied. It should be noted that it operates from the bottom up (from low addresses up memory) and thus it is not able to copy a block that overlaps at the top end of the original block as the data will effectively overwrite itself. It can, however, quite safely move the data down in memory even if overlapping. The method for copying up memory must be to do two copy operations, the first to an address clear of the original block and the second into the required position.

Search asks for the start address that the search is to run from, the length of memory to be searched and the hexadecimal byte that is to be looked for. If the byte is not found in the selected range the programmer displays the message 'FAIL' to show that it could not find it.

However, if the byte is found the message 'At xxxx' is displayed where 'xxxx' is the address in memory at which the byte was found. To search further just repeat the function.

Algm — This allows the setting of the programming algorithm for the programmer with three options displayed in the menu — FAST, SLO and PULSE as shown in Table 6.

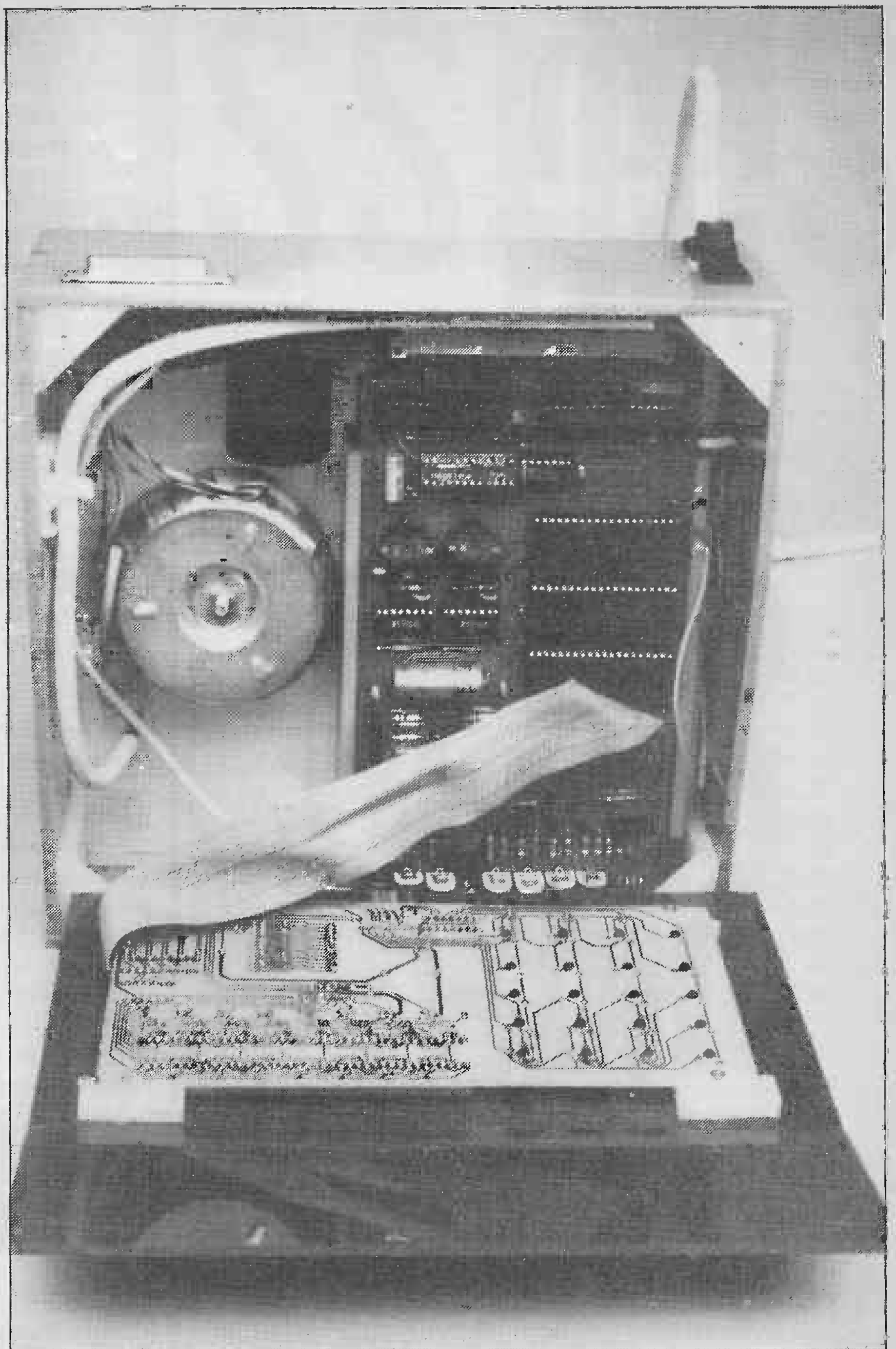
The FAST algorithm is always selected by the programmer to start up and operates for all EPROMs larger than a 2732 — automatically defaulting to SLO for 2732 and smaller EPROMs that do not support the FAST programming method.

The SLO algorithm programs by using 50ms pulses and typically would take two minutes to program a 2716.

The PULSE option is available for later expansion to micropulse techniques but is not implemented in this design. Selecting it will default to FAST.

The one other function on the *Algm* button is the 'INIT' state allowing the programmer to be reset to a known state if for any reason the RAM should become corrupted as discussed earlier.

In use there are more than 30 different messages that might be displayed either for prompts, results of commands or because something went wrong. A list of these is given in Table 5 along with a short comment to explain the prompt or give some idea of what may have been the problem.



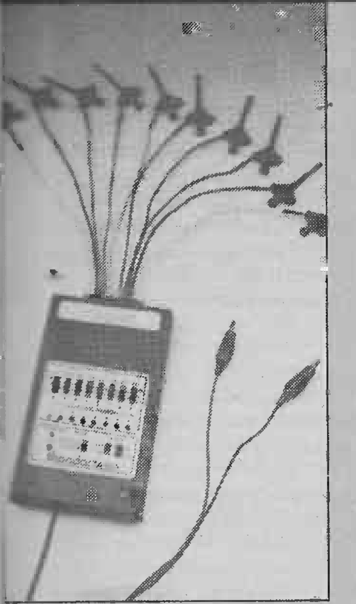
BUYLINES

The system ROM for this project is available from G. J. Bennett, 8 Hollymoor Gardens, Beaminster, Dorset, DT8 3NH and costs £24.95. Whilst this may seem a large amount for a hobby project it remains a small part of the overall cost in hardware and represents a considerable amount of thought and time in software development stage. The intention is to support it for some time to come.

Whilst the hex code as published here is the latest available the author retains the right to improve and upgrade the software at any time. To this end any bugs or errors discovered by users should be notified as soon as possible with detailed instructions for reproducing them. If any major changes, patches or bug fixes become necessary as a result, an updated ROM will be supplied, or for those people wishing to carry out their own fixes the details can be supplied on a BBC DFS or ADFS format 5.25in disc (40 or 80 track).

For those with a masochistic bent of mind, the source code for the system ROM is available on a BBC format disc (types as above) for the sum of £17.50. This totals some 100K of source code in nine files, and though written for the Watford Electronics 'Romas' Assembler should with minor mods be useable on 'ADE' or similar assemblers.

ANALYSIS IN MINIATURE



An eight-channel logic analyser in a hand-held box for less than £100 may seem too much to ask but Thandar has produced just that. The TA100 is really a replacement for multiple logic probes but it can be used instead of more complex and expensive fully fledged logic analysers in many situations.

Each channel of the trigger word can be set to 0, 1 or don't care. The logic status of each channel is continuously shown on the LED display with pulse-stretching to enable short pulses and high frequencies to be clearly displayed. In free run mode signal activity is continuously displayed and the trigger output generated each time the trigger is recognised can be used to trigger an oscilloscope to get a real-time view of the date activity at the trigger point.

In triggered mode the display again continuously tracks the signal activity until the trigger is recognised which halts the analyser and holds the display at the trigger word. The trigger output can be used to start a recording device such as a digital storage oscilloscope or logic analyser.

A typical application would be checking that correct data is being written to a port in a microprocessor system. With the analyser set to be clocked by the port's latch signal and the trigger word set to don't care, the first word written to the port would be captured on the eight data lines each time that analyser is run. Alternatively, a specific trigger word can be set-up to check the required data is being correctly written to the port and the trigger output can be used to trigger an oscilloscope so that actual signal timings can be observed.

The TA100 costs £89 + VAT from Thandar Electronics, 2 Glebe Road, Huntingdon PE18 7DX. Tel: (0480) 412451.

INTERACTIVE GET-TOGETHER

Philips and Sony have finally announced the official specification for the compact disc interactive system (CD-I). The 'green book' containing the pronouncements of the two companies is now available to licensees worldwide.

The CD-I system is a joint development by Philips and Sony, much like CD video and CD audio itself. The system can use all kinds of information such as music, speech, still images, video, animation, computer graphics and text. It is aimed at applications as diverse as training systems, retail catalogue and ordering systems and sophisticated computer games.

The idea of CD-I goes back almost as far as CD itself. The provisional specification was issued in 1986 and many companies are currently looking at commercial applications. Prototype players and software authoring tools are already in the field so commercial products should be in the shops in the not too distant future.

Further details are available from Philips Electronic and Associated Industries, Philips House, 188 Tottenham Court Road, London W1P 9LE. Tel: 01-436 4044.

MOISTURE MEASURING METER

Electronic Temperature Instruments has temporarily abandoned measuring temperature with the release of a moisture meter for seeing how soggy your walls are (or anything else made of cement, wood, paper and foodstuffs, tobacco and who knows what else).

The M49 Moisture Meter has no probes but searches out water with the help of a magnetic field, which can happily penetrate most materials which are likely to absorb water. Moisture levels up to 20% by weight (which is pretty wet) can be measured and displayed on a 3-digit LCD.

The M49 costs £170 from ETI Ltd, PO Box 81, Worthing BN13 3PW. Tel: (0903) 202151.

BIG SALE — EVERYTHING MUST GO

The Queen's speech to parliament in November confirmed the Government's intentions for the privatisation of the electricity generating and supply industry.

The Central Electricity Generating Board will be split into two. The largest, to be known as National Power, will own 70% of the power stations including all the nuclear ones. The smaller company, PowerGen, will own the rest. The National Grid will be nominally owned by the 12 privatised area electricity boards although it will be effectively independent.

The area boards will be responsible for ensuring continuous supply but will be allowed to build a (small) number of generating stations.

The privatisation of the area boards is expected to start in spring of next year with the generating board following suit in 1991 or 1992. The whole sale is expected to raise nearly 30 billion pounds, dwarfing all previous privatisation issues.

However, there is speculation that such a large issue cannot be satisfactorily carried out in this time.

In particular there is still much to be worked out for the contracts between the area boards (the retailers) and the generating companies as to who will bear the risks for capital costs and who will pay for cleaning up the industry in line with Mrs Thatcher's much vaunted 'green' policy.

The retailers will have little in the way of capital but will be required to maintain a supply whereas the generating companies will have a large capital base but be under no obligation to build new power stations.

Until such points are settled neither the City nor the public are going to be too keen to invest in the new companies.

COMPANY NEWS

Eighteen months after GEC's bid to take over Plessey was turned down by the Monopolies Commission and about a year after GEC and Plessey merged their telecommunications business to form GPT, GEC is making another bid for the whole of Plessey.

GEC has formed a consortium with German electronics giant Siemens to place the 1.7 billion pound bid which Plessey is keen to fight off. As yet Plessey is deviously lobbying Whitehall to block the bid again but the general feeling is that the DTI and MD may not oppose a takeover because of the lack of other suitable suppliers to meet defence needs.

The GEC-Seimens bid has helped Plessey's shares to rise sharply over eight points while the six monthly results to September showed a 10% increase in profits, in no small part due to the success of the GPT venture and to the Ferranti semiconductor division acquired by Plessey a year ago.

Meanwhile the trouble-torn Super Channel satellite TV programme provider has been placed under a court appointed administration at the request of the new majority shareholders Betatelevision of Italy.

Betatelevision bought a 55% stake in the company from the consortium of 14 ITV companies which founded the channel with Virgin nearly two years ago. It seems the Italians found the company in worse trouble than anticipated, with losses running at around ¾ million pounds each month, and called in a court ruling to protect Super Channel from its rapidly growing group of creditors and to keep it going while their two million pound rescue package takes effect.

One of the major creditors is Virgin (by way of its stake in Music Box) which now finds itself in a paradoxical position of being both a major shareholder and creditor. Intense negotiations between Virgin and Betatelevision are underway as how to best save the company which is rumoured to have accumulated losses of around 60 million pounds.

In a bid to stave off shortage of RAM chips, Amstrad paid 75 million dollars in October for a near 10% stake in US chip manufacturer Micron in return for the right to buy 9% of Micron's RAM output. However Texas Instruments, the only other major US RAM manufacturer, has now sought a court injunction to stop Micron making RAM chips on the grounds that they infringe TI patents.

Nevertheless, it is thought to be unlikely that an injunction will be granted until the technically tortuous merits of TI's case are explored, by which time prices are likely to have fallen and the next generation of Japanese high capacity RAM chips will have appeared.

Representatives of Japanese electronics industry have visited Washington to try to persuade the US government to lift the sanctions on importing Japanese electronics goods and materials. The 100% tariffs on selected products were imposed in May 1987 to persuade Japan to open its gates to foreign (ie US) chips.

The Japanese claim they have met the US demands by increasing the import of foreign chips. The foreign share of the Japanese semiconductor market has risen by about 2% since imposition of the tariffs. Although the US semiconductor industry is naturally still opposed to lifting the sanctions, it is felt the US government may now take a more friendly position as many of the pro-sanction lobby have left with the Reagan administration.

READING BY NUMBERS

Martin Pill looks at the different types of ADC available and advises on their choice

Analogue to digital converters (ADCs) are devices which, as their name suggests, receive as input an analogue signal and produce a corresponding digital output. Their main use is to allow digital electronics and computers to make use of data obtained from the real world (which is by and large analogue). There are several basic techniques of analogue to digital conversion, each with its peculiar advantages and limitations. The one selected for a particular task will be determined by factors such as speed of conversion required, accuracy and cost.

The first thing to remember with ADCs is that the output will only rarely be exactly equivalent to the input. The reason for this is that the input voltage being analogue can take an infinite range of values, while the output is restricted to moving in discrete steps each equal to one LSB. For example an 8-bit ADC has 28 or 256 possible output codes. If one is used in a voltmeter to measure the 240V mains supply with a full scale of 256V then the resolution of the system would be 1V. If a variable transformer was used to reduce the voltage down to zero it would appear to go down in 1V steps.

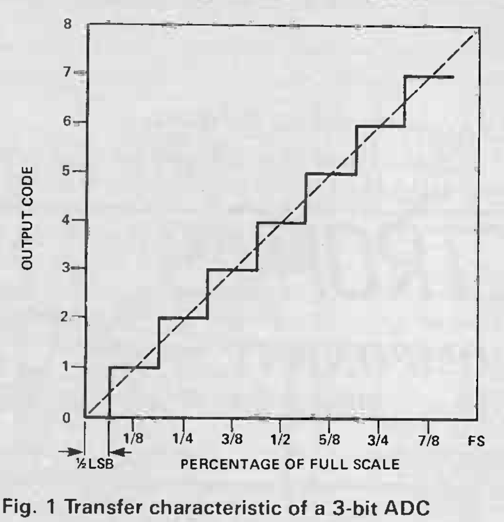


Fig. 1 Transfer characteristic of a 3-bit ADC

Ideal Lines

Figure 1 shows the transfer characteristic of a 3-bit ADC. The 'ideal' straight line is shown superimposed on the actual. There are eight possible output codes and each one with the exception of zero and seven exists over an analogue increment of one LSB. The ADC is usually adjusted so the the transitions between the codes occur at $+\frac{1}{2}$ LSB on either side of the nominal analogue voltage and therefore the output will always be within $+\frac{1}{2}$ LSB of the input. If this were not done the output could at times be almost 1 LSB away from the voltage being measured.

From one point of view, it could be said that all ADCs operate in the same way. Rather than measuring the input voltage in some empirical fashion and then converting it to a digital value they instead compare the unknown input voltage with known internally generated voltages until a match is found. This is rather like weighing vegetables on old fashioned scales which use individual weights and determine the numbers of weights needed to make them balance. The mass of each weight is known and so the weight of the vegetable is known also.

The *parallel* or *flash* converter is the simplest type of ADC but it is also the fastest, some examples being able to perform over ten million conversions per second. In an n-bit parallel converter (Fig. 2) the input voltage is fed to one input of each of n voltage comparators, the other inputs of which are connected to an equally spaced reference voltages provided by a resistor ladder. The output of the comparators whose

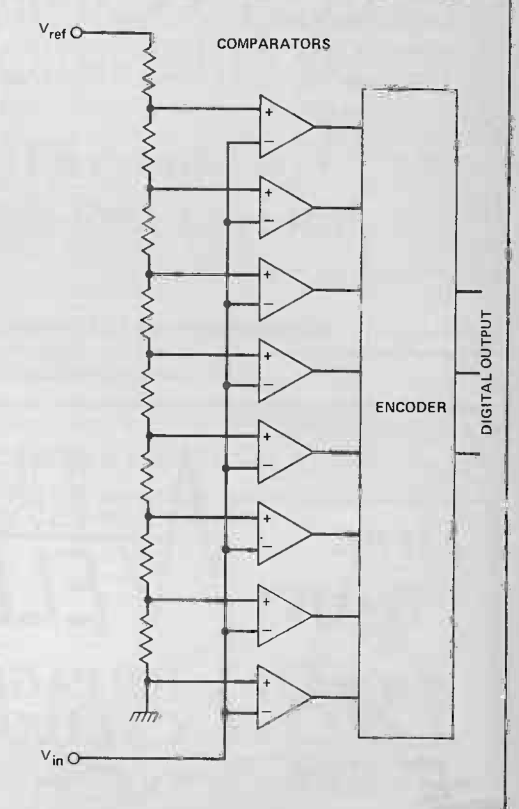


Fig. 2 Parallel encoded ADC

reference voltage is less than the signal voltage will be low, whilst the output of those whose reference voltage exceeds the input voltage will be high. This allows an encoder connected to the comparators to produce a coded digital output which corresponds to the signal input voltage.

The reason for the converter's speed is that the delay time between the input and output is only the sum of comparator plus encoder delays. However, the large number of comparators required (16 for a 4-bit converter, 256 for an 8-bit) makes them very expensive and so they are only found in applications which require their exceptional speed. These include radar data processing and digital video systems.

By contrast, *single slope integration* is slow and not particularly accurate. The signal voltage is applied to one input of a comparator (Fig. 3) whilst the other input is fed from a ramp generator which produces a smoothly increasing voltage rising from 0V to the maximum input voltage. This is easily achieved by putting a current source in series with a capacitor — as the capacitor charges up, the voltage across it increases linearly.

A binary counter is started simultaneously to count pulses from a stable clock and when the ramp voltage equals that of the signal voltage, the comparator output will change from low to high to stop the counter. The time for which the counter has been running is determined by how long it took for the ramp generator voltage to become equal to the signal input voltage and so the count is the digital output. At the end of conversion the capacitor is discharged and the counter reset, ready for another cycle.

Sloping Off

Single slope integration is relatively simple but cannot be used where high accuracy is required as it puts too severe a requirement on the stability and accuracy of the capacitor and comparator. However, it is widely used where good resolution and the uniform spacing of adjacent levels are more important than absolute accuracy, such as in pulse height analysis where the amplitude of a pulse is held by a peak detector and then converted into a digital value.

Dual slope integration avoids most of the capacitor and integrator problems inherent in single slope integration. The principle is shown in Fig. 4. The input voltage is integrated by charging a capacitor for a fixed time interval with a current proportional to the input voltage. This means the final voltage across the capacitor (and hence the charge stored by it) is also proportional to the input voltage.

The capacitor is then discharged by a constant current until the voltage across it reaches zero. The time taken to discharge the capacitor is recorded by a counter driven from a stable fixed frequency clock. The discharge time is proportional to the input voltage and so the value of the counter can be used as the digital output.

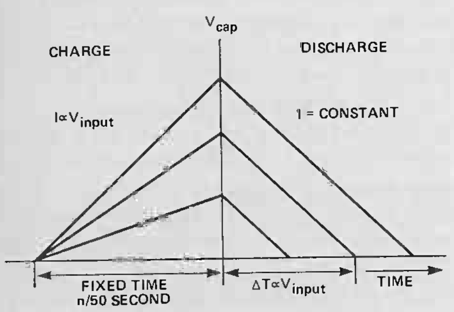


Fig. 3 Single slope ADC

At first sight this method might seem to offer little advantage over single slope integration. In single slope integration you are measuring the time taken to charge a capacitor up towards a fixed maximum value, whereas a dual slope converter seems to do the opposite and measures the time taken to discharge back to zero. However this start and return to zero is what makes the difference. The rate of charge and discharge of the capacitor is inversely proportional to its capacitance. So, if over a period of time the value of the capacitor in the single slope converter changes this will affect the resultant output. However with a dual slope converter the change of capacitance will affect the charge and discharge cycles equally and cancel out.

High accuracy converters precede each conversion cycle with an auto zero cycle in which the input is held at zero and the resulting output of the converter is subtracted from the subsequent measurement to cancel out any errors produced by measurements near zero.

Another advantage of dual slope integration is that the clock does not need to have high long term stability as any change in the clock speed will affect charge and discharge cycles equally. In fact, the only critical factor in a dual slope converter with auto zeroing is the discharge current and this is not a problem as precision voltage and current sources are easy to produce.

As always with these things there has to be a snag. With the dual slope converter it is speed. All these charge and discharge cycles take an inordinate amount of time and this means such converters tend to be used in applications where high resolution and low cost are more important than speed. For a fixed cost you will get the greatest precision from a converter that uses this technique.

Another advantage is the excellent rejection of powerline interference which can be obtained by relating the initial conversion time to the powerline frequency of 50Hz. Dual slope converters find their widest application in precision digital multimeters.

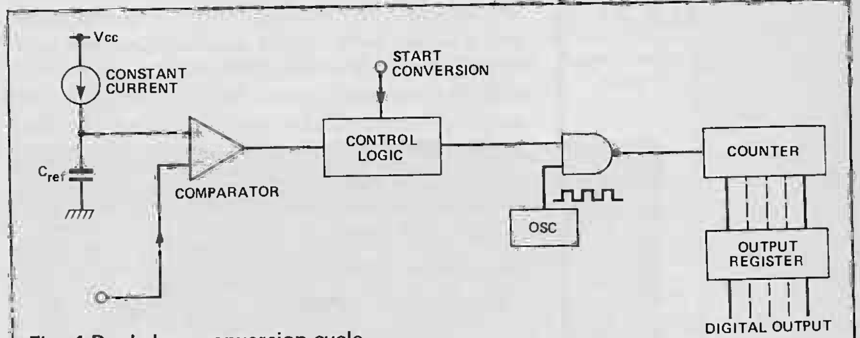


Fig. 4 Dual slope conversion cycle

A To D And Back Again

A number of analogue to digital converters also contain a digital to analogue converter (DAC) DACs have an opposite function to the ADC as they are fed with a digital code as their input and produce a corresponding analogue voltage at the output. Such ADCs work by adjusting the inputs to their DAC until its analogue output equals the input signal voltage. At this point the DAC input code corresponds to the correct value for the ADC output. It's a bit like tuning a musical instrument by comparing it with another which is in tune.

The speed of this type of device is clearly determined by the time taken to find the correct DAC input code and a number of different methods are in use.

The staircase and comparator uses one of the simplest and most straightforward methods, as can be seen in Fig. 5. When the start conversion signal is received, the counter starts counting up from zero, feeding its count into the input of the DAC. The DAC output and the input voltage are fed to two inputs of a comparator. The comparator output will remain low whilst the output from the DAC is below the input voltage but when they become equal, the comparator output will flick high and stop the counter. Thus its count will become the digital output code.

This method is very simple and cheap but it is also slow. The time taken for one conversion increases in proportion to the input voltage so that for a full scale conversion it requires $2n-1$ clock pulses where n is the number of bits.

All the converters described above assume that nothing is known about the input signal before conversion and that it may be anywhere within the range of the converter. This is often not the case and there are many instances in instrumentation for example

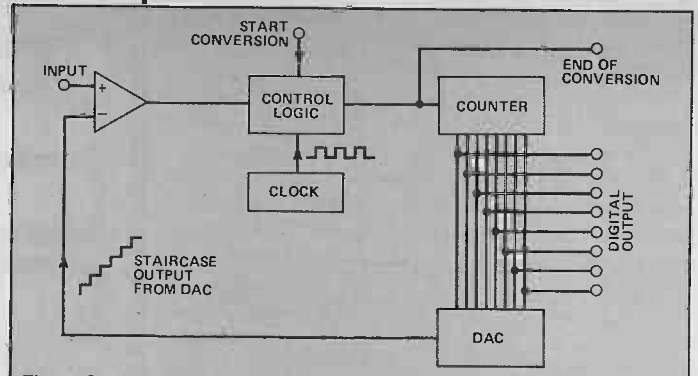


Fig. 5 Staircase and comparator ADC

where there is likely to be little difference between successive readings. The tracking converter is a version of the staircase and comparator which is especially suited to these situations because as its name implies it is designed to follow changing input voltages (Fig. 6).

Like the staircase and comparator this also uses a DAC driving a comparator but there the similarity ends. The DAC inputs are driven by an up-down counter which for the first conversion counts up from zero until the DAC output equals the signal input. However, instead of the counter being reset for the next cycle it merely counts up or down under the control of the comparator until the DAC output is once

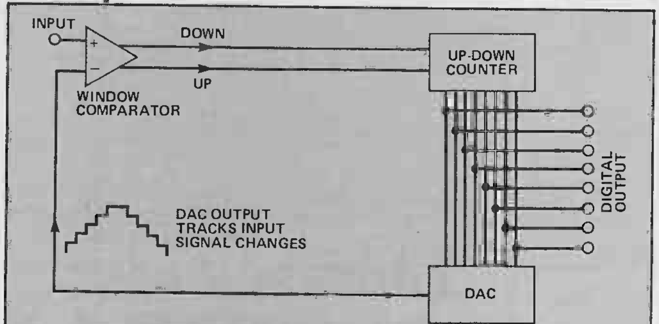


Fig. 6 A tracking ADC

again equal to the signal input voltage. So the time taken for each conversion is determined by the degree of change of the signal. It is slow in responding to jumps in the input signal but will easily follow a smoothly changing waveform.

Successive approximation is the most popular and widely used ADC technique, as it is relatively accurate, quite fast and cheap. In effect it is a sophisticated version of the staircase and comparator, using a binary search to find the output which most closely approximates to the input. As before, the input signal is fed to one input of a comparator whose other input goes to the output of a DAC. However, rather than just incrementing the DAC in a serial fashion the much faster binary search-type method is used.

After setting all the bits to zero, each bit in turn is set to one, starting with the MSB. After each bit is set a comparison is done and if the output of the DAC exceeds the signal input then the bit is reset otherwise it is left at one. This continues until the output and input signals are equal.

Another way of looking at the process is to consider weighing a bag of sweets weighing 13g on some old fashioned scales with two pans and a supply of weights. The slowest way of doing it would be to add weights a gramme at a time until the pans balanced. It would be quicker to use successive approximation with weights of 16, 8, 4, 2 and 1 gramme. First try the 16g weight and discard it because it's too heavy. The 8g weight is tried next and left on the pan as is the 4 g. The 2g weight is discarded and he 1g balances the pans.

As can be seen from Fig. 7. this is a much faster way of arriving at the correct value and requires only n clock pulses for n-bit precision.

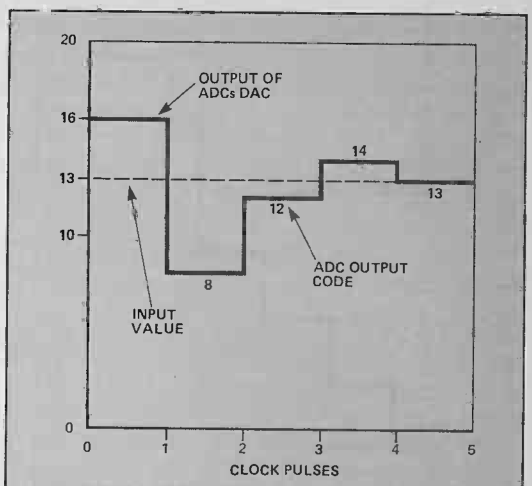


Fig. 7 The conversion cycle of a successive approximation ADC

In The Balance

The A to D conversion technique known as charge balancing uses a capacitor to keep track of the ratio of the input signal to a reference voltage. This method averages the input by integrating it over a fixed time interval for a single measurement. This method has two main advantages. The capacitor is used for both the signal and reference, and so reduces the importance of capacitor stability and accuracy. The same advantage is also conferred on the comparator. The result is that better accuracy can be obtained from a particular quality of component or equivalent accuracy can be obtained at a lower cost. The second advantage is that by choosing the conversion time interval to be a multiple of the power supply time period, the converter becomes insensitive to 50Hz 'hum' and its harmonics which may be superimposed on the input signal. Unfortunately, charge balancing converters tend to be slower than those using successive approximation.

Practical ADCs are subject to a number of sources of error. Some of these have already been mentioned. such as the requirement for single slope ADCs to have a very accurate and stable capacitor.

Quantising error is the name given to the feature described earlier — that the output and input can differ by up to $+\frac{1}{2}$ LSB. as within these limits you cannot be certain of the exact value of the input. As this is true of all ADCs it is usually taken for granted and not quoted on the data sheet.

The resolution of an ADC is simply the number of output bits the converter has. However, it must be remembered that resolution is not the same as accuracy. The useful resolution is the number of bits at which an ADC has no missing codes. A missing code is an output value (or code) that will never appear no matter what the input. So if the input voltage is increasing smoothly the output will appear to jump at a particular point as the ADC skips over the missing code. The transfer function for a 3-bit ADC with a missing code might look like Fig. 8 rather than Fig. 1 which showed an ideal ADC.

Missing codes can arise through the converter having a non-monotonic internal DAC. Figure 9 illustrates this and shows the transfer function of the reference DAC which is non-monotonic at input code six so that step six of the staircase decreases. If this was part of, say, a staircase and comparator ADC it would be impossible for the counter to be stopped at this code. If the input signal is less than the DAC output

ADCs

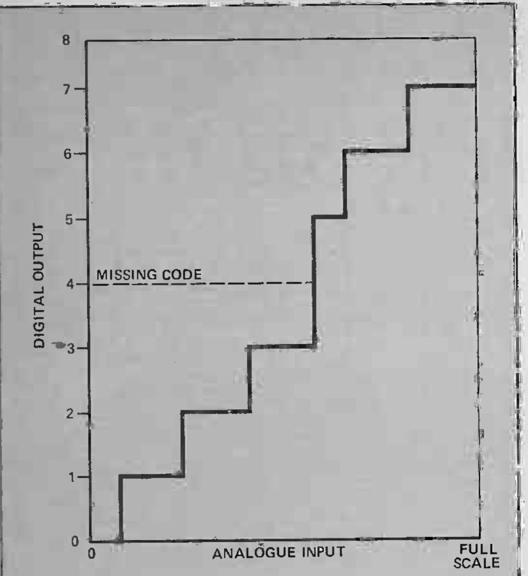


Fig. 8 Illustration of an ADC with a missing code

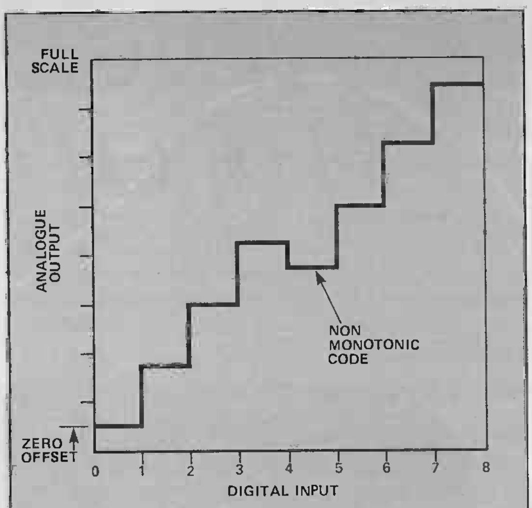


Fig. 9 Example of a non-monotonic DAC used in an ADC

for code five then the counter will be stopped before code six is reached. Alternatively, if the signal input is greater than output five it must also be greater than output six so the comparator will not change state at output six. Output code six will therefore never appear and will be a missing code. In practice many manufacturers guarantee their ADCs to have no missing codes over their operating temperature range.

Table 1 shows some of the parameters of a selection of ADCs.

There is, of course, no single answer to the choice of an ADC. Selection will depend very much on the exact use to which the device is to be put. However, with the information presented here you should be able to choose the right chip to suit your needs and your pocket.

Part No	Conversion Method	Resolution	Conversion Speed	Approximate Price
RS3300	Parallel (Flash)	6-bit	15MHz	£25
RS507C	Single slope	7-bit	1ms	£1.50
ZN439E	Successive approximation	8-bit	5μs	£5
8703	Charge balanced	8-bit	1.25ms	£12

Table 1. A selection of different types of ADC