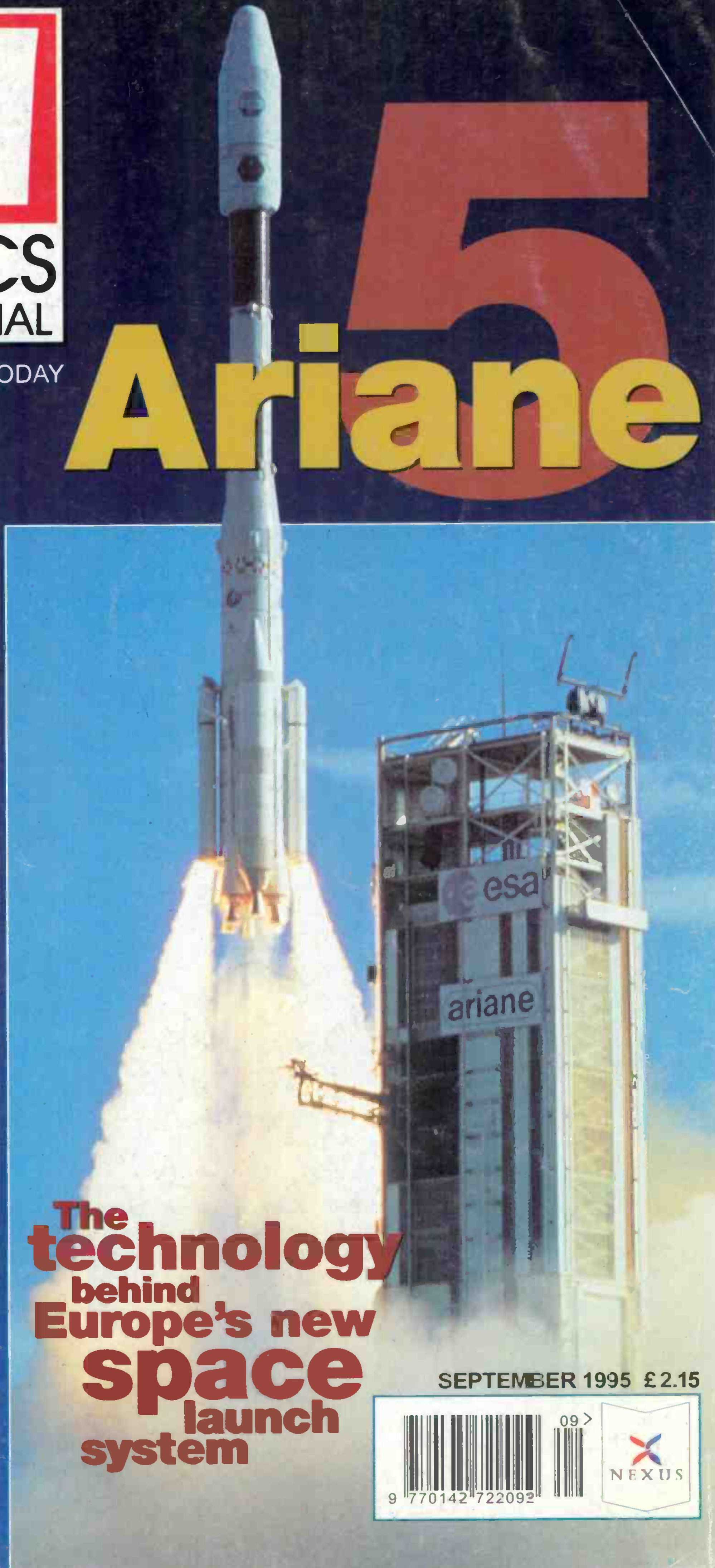
ETI

ELECTRONICS TODAY INTERNATIONAL

TOMORROW'S TECHNOLOGY TODAY

Ariane 5



Build a versatile digital message display

Fibre optic MIDI link for music without interference

Introducing the ETI ultra low cost BASIC programmable micro computer

Plus

- 8088 interrupt based control computer
- Dry battery tester
- Using the Stamp microcomputer
- Practical approach to the PIC

The technology behind Europe's new space launch system

SEPTEMBER 1995 £2.15

09

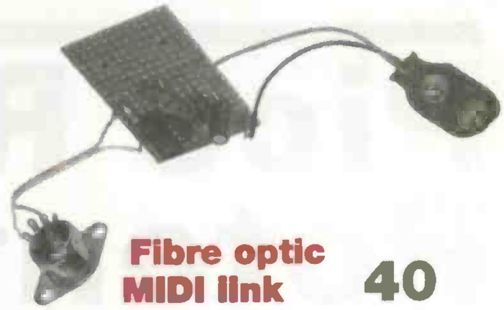
9 770142 722092

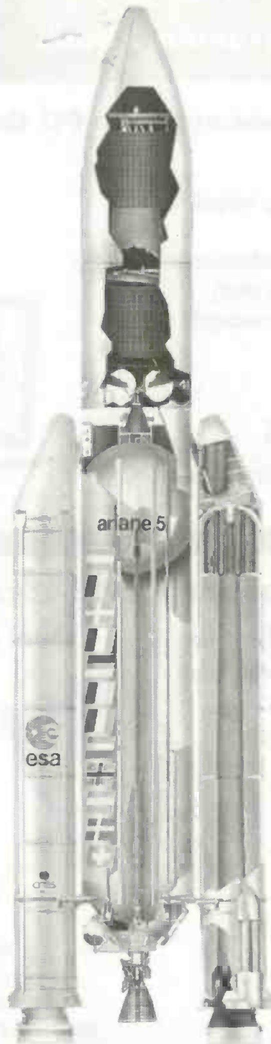
NEXUS

Contents

Volume 24 No.9

& Features Projects

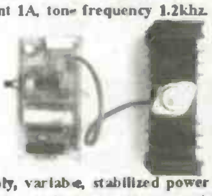




Regulars

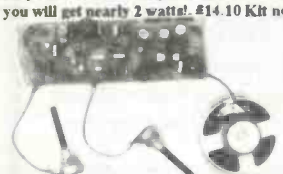
A mini FM transmitter, very high gain preamp, supplied complete with FET electret microphone. Designed to cover 88-108 Mhz but it is easy to change it to cover 63-130Mhz. Works with a common 9v(PP3) battery. 0.2W RF. £8.22 Kit no 1001.

Electronic siren kit with an impressive 5 watt power output. Ideal for car/bike alarms etc. 6-12vdc, max current 1A, tone frequency 1.2khz. £7.05 Kit no 1003.

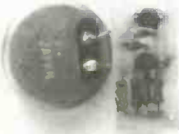


3-30v Power supply, variable, stabilized power supply for laboratory use. Shortcircuit protected, suitable for professional or amateur applications. 24v 3A transformer is also needed to complete the kit. £16.45 Kit no 1007.

Powerful 1 watt FM transmitter supplied complete with piezoelectric microphone. 8-30vdc. At 25-30v you will get nearly 2 watts!. £14.10 Kit no 1009.

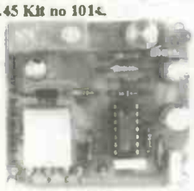


FM/AM Scanner, well not quite you have to turn the knob yourself but you will hear things on this radio (even TV) that you would not hear on an ordinary radio!. Receiver covers 50-160Mhz both AM and FM. Built in 5 watt amplifier. £17.62 Kit no 1013.



Mosquito repeller, modern way to keep midges at bay! Runs for about a month on a 1.5v battery. £8.22 Kit no 1015.

3 channel wireless sound to light system, mains operated, separate sensitivity adjustment for each channel, 1,200 watt power handling. Microphone included. £16.45 Kit no 1014.

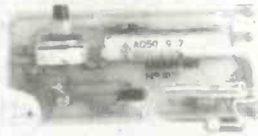


Motorbike/cycle trembler alarm, adjustable sensitivity, preset alarm time, auto reset. Could be connected to bikes horn etc. £14.10 Kit no 1011

0-5 minute timer, adjustable from 0 to 5 mins, will switch up to 2A mains. Perfect for alarms, photographic laboratories etc. 12vdc. £8.22 Kit no 1020.

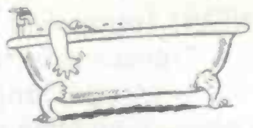
4 watt FM transmitter, small but powerful transmitter, 3 RF stages, microphone and an audio preamp include in kit. £23.50 Kit no 1028.

25 watt FM transmitter 4 RF stages, preamp required (our kit 1068 is suitable). Due to the complexity of the transmitter it is supplied in built up form only. £92.82 Kit no 1031.

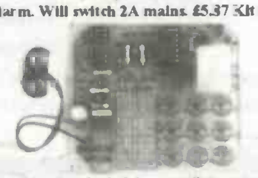


Strobe light, adjustable frequency from 1- to 60Hz (a lot faster than conventional strobes) mains operated. £18.80 Kit no 1037.

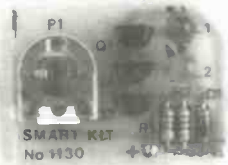
Ultrasonic radar ideal as a movement detector with a range of about 10 metres, automate your cat flap! 12v operation so ideal for cars, caravans etc. £16.45 Kit 1049.



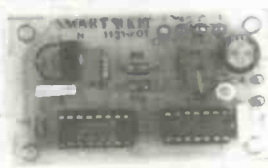
Liquid level detector useful for detecting fluid levels in tanks, fishponds, baths or as a rain or leak alarm. Will switch 2A mains. £5.87 Kit no 1081.



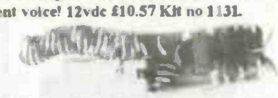
Combination lock 9 key, easily programmable, will switch 2A mains. Complete with keypad. 9v operation. £11.75 Kit 1114.



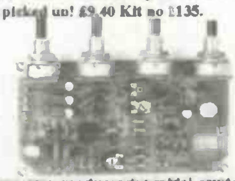
Phone bug detector, this device will warn you if somebody is eavesdropping on your phone line. £7.05 Kit no 1130.



Robot voice, interesting circuit that distorts your voice! adjustable, answer the phone with a different voice! 12vdc £10.57 Kit no 1131.



Telephone bug, small bug powered by the telephone line, starts transmitting as soon as the handset is picked up! £9.40 Kit no 1135.



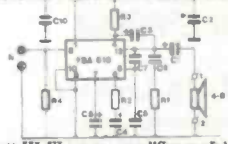
function generator, produces sinusoidal, saw tooth and square waves adjustable from 20hz-20khz, separate level controls for each shape. Will produce all 3 together. 24vac £17.62 Kit no 1008.

3 Channel light chaser, 800 watts per channel, speed and direction controls supplied with 12 leds and mains triacs, so you can use mains light bulbs if you want. 9-15vdc £19.97 Kit no 1026.

12v flourescent. A useful kit that will enable you to light 4' flourescent tubes from your car battery! (you will also need a 9v 2A transformer, not supplied) £9.40 Kit no 1069.

VOX switch, sound activated switch ideal for turning tape recorders on and off when sounds are heard. Makes the tape last a lot longer! adjustable sensitivity, built in delay. £9.40 Kit 1073.

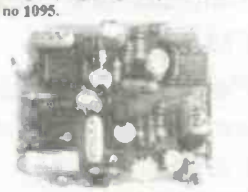
Incar sound to light, Put some atmosphere in your car with this mini 3 channel sound to light. Each channel has 6 led's. £11.75 Kit no 1086.



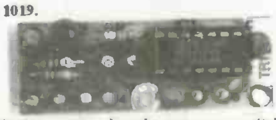
7 watt HI FI power amplifier useful, powerful, ideal for intercomms, audio systems, car use etc. 12-18vdc 500mA. £8.22 Kit No 1025.

Phone call relay, useful device that operates a relay when ever the 'phone rings, could be used to operate more bells or signalling lights etc. Will switch mains at 2A. £11.75 Kit no 1122.

Lead acid charger, two automatic charging rates, visual indication of battery state, ideal for alarm systems, emergency lighting etc. 100mA 12vdc. £14.10 Kit no 1095.



Car alarm system, works on voltage drop and/or vibration, entry and exit delays plus adjustable alarm duration. Good for cars, caravans etc. £14.10 Kit no 1019.

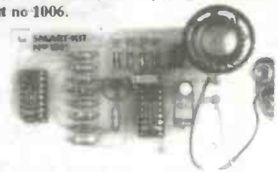


Portable alarm system, based on a mercury switch. The alarm continues to sound until the unit is disabled by the owner. Buzzer included. £12.92 Kit no 1150.

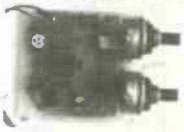
Preamp mixer, 3 input mono mixer, separate bass and treble controls plus independent level controls. 18vdc, input sens 150mv. 100mA. £17.62 Kit no 1052.

Mini metal detector, suitable for locating pipes in walls etc, range 15-20 cm, complete with case. 9v operation. £9.40 Kit no 1022.

800 watt single channel sound to light kit, mains operated, add rhythm to your party for only £9.40 Kit no 1006.



Sound effects generator, produce sounds ranging from bird chips to sirens, complete with speaker, add sound effects to your projects for just £10.57 Kit no 1045.



Guitar preamp with tone controls, small enough to fit inside any guitar, based on TL082 IC, 9-12vdc, 50mA. £9.40 Kit no 1091.

15 watt FM transmitter, 4 stage high power, preamp required. 12-18vdc. Can use either ground plane, open dipole, or Yagi. Supplied in built form only at £81.07 Kit 1021.

Telephone amplifier, Very sensitive amplifier which uses a 'phone pickup coil (supplied) will let you follow a conversation without holding the phone. £12.92 Kit no 1059.

TOP 10 BEST SELLING KITS CORNER

1. Variable speed control kit for 12v DC motors up to 30A! (you may need a heatsink for 30A) £19.97.
2. Composite video kit, converts composite signals into separate H sync, V sync and video £9.40.
3. Geiger counter kit, contains everything you need to build a working counter £22.32.
4. Solar energy kit, contains a solar panel, motor, buzzer and cable for experiments £5.87.
5. Electronic accupuncture kit, may help with migrane, poor circulation, backache etc. £8.22.
6. Electrifying apparatus kit, produces a weak adjustable high tension of 80-300v from a 9v battery, ideal for catching worms etc! £9.40.
7. Adapter bug kit, contains everything you need to build a professional bug built inside a standard 13A mains adapter! the bug is mains powered so it operates all the time the adapter is plugged in. Price is £16.45 for the complete kit including adapter. Hand tools and glue required.
8. Nicad charger kit automatic charger for cells from 1.2v to 15v, 7 settings 50600mA, transformer required 18-20v 600mA. £9.40.

9&10. Inverter kits, produce 240vac from a 12vdc supply, two versions are available a 15w one at £14.10, and an 80w version at £23.50.

EXPRESS COMPONENTS

how to order......

Remember to add £1.50 p&p.
By phone with a credit card.
By post with either a cheque, postal order or credit card details.
By fax with credit card details.

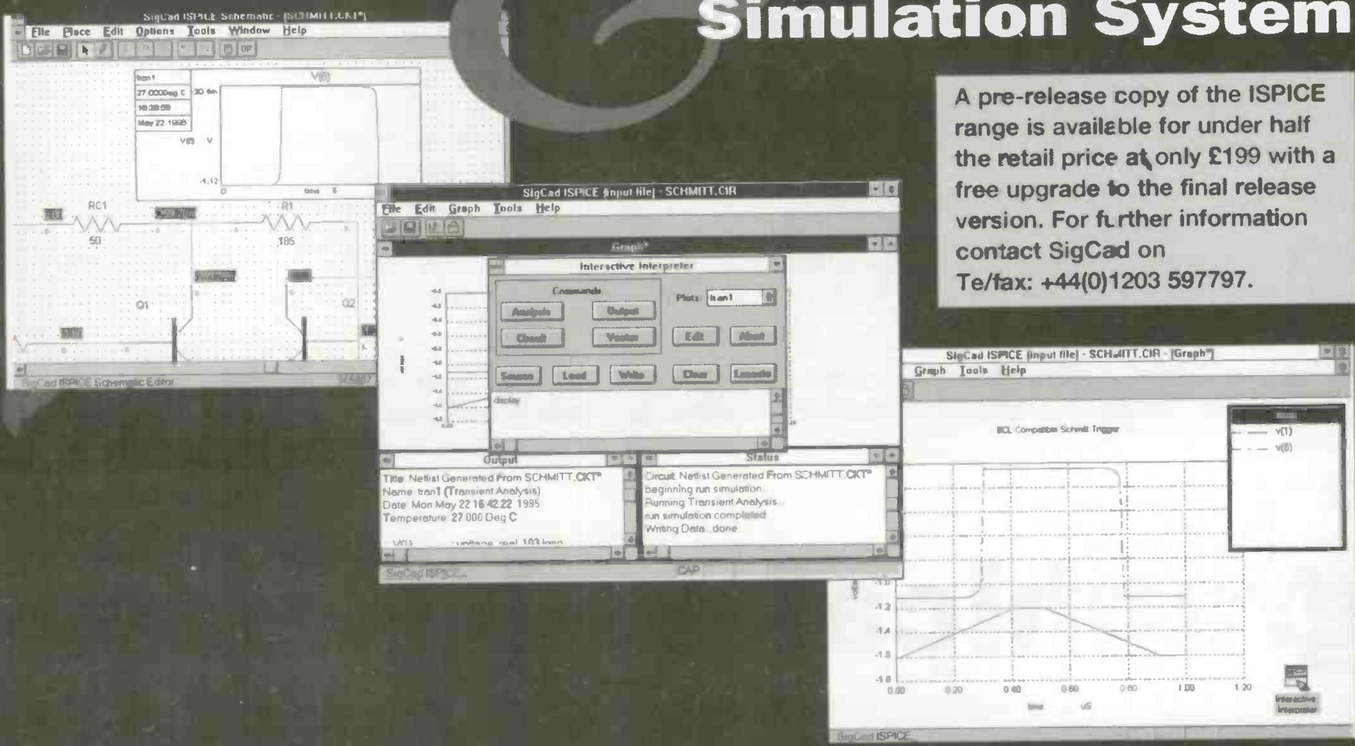
EXPRESS COMPONENTS, PO BOX 517 HOVE SUSSEX BN3 5QZ. DEPT ETI
TEL 01273 771156
FAX 01273 206875

Overseas orders please add £3.50 post and packing.

ETI News...

ELECTRONICS TODAY INTERNATIONAL

New Windows Based Integrated
Electronic Circuit Design & Simulation System



A pre-release copy of the ISPICE range is available for under half the retail price at only £199 with a free upgrade to the final release version. For further information contact SigCad on Te/fax: +44(0)1203 597797.

The New SigCad ISPICE range of products provides unlimited analogue simulation capabilities in Windows for all levels of users. The logical design steps have been broken down into multiple modules allowing users to bypass certain steps easily. This allows seasoned professionals and novices to interface to ISPICE at different levels.

SPICE is an integrated system composed of four modules, each providing different functions to ease difficulties encountered during circuit design and simulation. These are: ISPICE Schematic Editor, ISPICE Component Editor, ISPICE Netlist Editor and ISPICE (simulator and postprocessor)

SigCad ISPICE is based on Berkeley SPICE 3F5 with a number of extensions and improvements. Unlike other SPICE3 simulators, ISPICE has been interfaced with an advanced postprocessor so simulations and data analysis can be carried out in the same environment. Interactive communication with both the simulator and postprocessor is achieved via the Interactive Interpreter using its own command language.

SigCad has not limited the simulation capabilities of ISPICE, which has 10 types of analyses, including AC, DC Transfer Curve, Transient, Noise, Distortion, Temperature, Sensitivity (AC and DC) etc. The built in models include 6 types of MOSFETs, 3 types of transmission lines (lossless/lossy), SPICE2 nonlinear dependent sources including the new arbitrary source (B element), MESFET model and all other standard SPICE primitives.

The ISPICE Schematic Editor is designed to be an interactive front-end to ISPICE that greatly eases the burden of creating a complete SPICE netlist directly from the schematic. Most components can be placed on the schematic with a single keystroke and automatic node numbering is done in real time. Simulations can be run from the Schematic Editor environment allowing waveforms to be interactively cross-probed on the schematic and circuit operating point voltages can be updated as component values are changed. This provides a powerful debugging tool for designers as they are constructing their circuit!

New symbols for user defined models can be created and loaded in by using the ISPICE Component Editor. The ISPICE Netlist Editor allows designers to either create netlists or edit those generated by the Schematic Editor. SPICE2-style output and simulator generated errors and warnings can also be viewed, together with the netlist, in the Netlist Editor's environment.

ISPICE requires Windows 3.1 or higher on PC compatibles with 4MB of RAM. ISPICE does not require a co-processor and circuit size is unlimited!

New low voltage 4 bit microcontroller

NEC's new KPD753017 4-bit microcontroller has one of the most powerful LCD drive capabilities available on a product of this type. The device, which is part of the company's enhanced performance, low voltage 75XL range, has an integral 32 x 4 segment LCD driver enabling it to drive complex information displays.

75XL devices are enhanced performance versions of NEC's 75X range, currently the world's best-selling 4-bit microcontroller. They are particularly aimed at low cost applications which traditionally use more expensive 8-bit products, and the uPD753017 is particularly suitable for automotive electronics and industrial control. The devices also feature a powerful instruction set and a wide variety of on-chip peripherals.

The ultra-low power uPD753017 has guaranteed operation from 1.8V to 6.0V, the lower voltage enabling it to be driven from the 'dead' voltage of two lithium cells. This allows the construction of systems that are lighter and more compact than ever before.

For further information contact NEC Electronics (UK) Ltd, 01908 691133

New graphical multimeter

The new 860 series of Graphical Multimeters provide the user with the means of viewing waveforms, graphs, in-situ component signatures, trend plotting and logic activity. They are also quality digital multimeters with all the functions you would expect from the world's leading source of dmms.

B:Waveform Display mode provides a clear picture of noise, distortion, intermittent failures and glitches with either a fully automatic or manual set-up.

Trendgraph mode plots high resolution meter readings for up to 30 hours, in intervals from 1 second to 15 minutes, to detect such anomalies as power sags/surges and droops.

In-Circuit Component Test mode can view component signatures in-circuit. This allows for comparison of good/bad circuits and components.

Logic Activity mode offers a simple and effective method of isolating digital failures.

Meter mode is more traditional, offering both digital and analogue information of all parameters via a 32,000 count (4.5 digit) display and an analogue NeedleGraph.

The 860 Series of handheld graphical multimeters is readily available from all members of the Professional Instrument Distributors Association throughout the UK together with a full and competent back-up service.

For further information contact Quiswood Ltd on 01756 799737.

SocketModems Evaluation Kit

For further details, contact Telecom Design Communications Limited on 01256 332800.

Basingstoke based Rockwell distributor, Telecom Design Communications Limited, have anncunced a range of evaluation and development boards for Rockwell's 'Plug. and.Play' SocketModems.

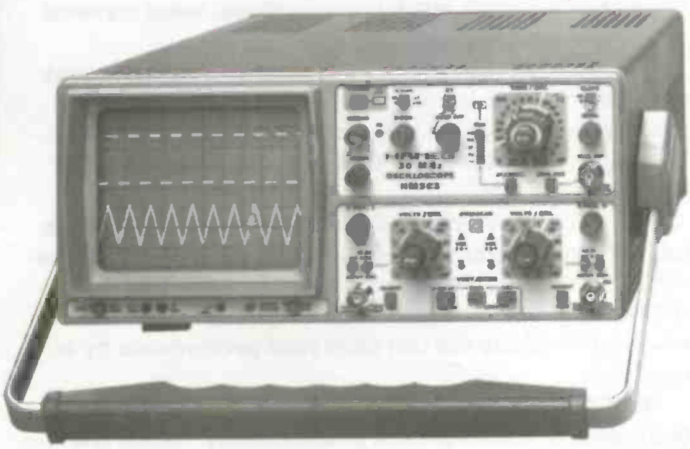
The new boards are available with or without DAA's (Data Access Arrangment or Line Interface), either as PC-plug in cards, or as external serial types. The serial versions can be connected either directly as TTL level serial, or with full RS-232 drivers.

Rockwell SocketModems are available for their entire range of modem capabilities, from simple 2400 bps data.only devices, to high speed 14400 bps data/fax modems with voice facility, error correction, data compression and cellular compatibility. V.Fast class versions will be announced shortly. All SocketModems may be used with the evaluation boards, enabling designers to quickly configure a range of products for desktop or portable use.

Additionally, TDC provide a range of BABT recognised line interfaces suitable for use with Rockwell's SocketModems. Available as DIP and low-profile SIP versions, their certificate of recognition will aid the gaining of approvals for connection to the PSTN, greatly enhancing time to market. TDC also provide demonstration code for use with the boards further speeding up the design and modem integration process.

Best Selling Oscilloscope

The new Hameg HM 303 Oscilloscope is available from BK Electronics, priced at £392.00 plus VAT. which includes 2 switched probes and a manual. Carriage is free of charge.
Tel: 01702-527572 Fax:- 01702 420243.



The new Hameg HM303 standard Oscilloscope succeeds the HM203, which sold over 170,000 worldwide. The bandwidth has been extended to 3OMhz. A lighter weight of only 5.6Kgs and a lower panel height reduced by 20mm has been achieved by incorporating a switch mode power supply. Another advantage is that the new oscilloscope will work from a mains supply of between 90 and 240V AC, without adjustment.

Brief specification:

Dual Channel, DC to 30Mhz 2mV/Div.; Overscan Indicator Time Base : l0nS to 0.5s/Div.: Variable holdoff: X10 Magnification Triggering : DC-l000Mhz Actlve TV-Sync-Seperator : LED Trigger Indicction Additional Features: Component Tester, 1Khv/1Mhz Calibrator

External printer port to RS232 card

PC adapter card manufacturers Brain Boxes have now added a parallel printer portto their own extremely flexible dual Serial Port card.

The LPT parallel printer port is accessible from the 25 pin female connector. It may be set as LPTI to LPT3 with Interrupts 5 or 7. Thus allowing users of 0S/2, 0S/2 Warp V3, Novell,and Unix full interrupt driven printer output. Full bi-directional capability is obtainable for scanners, file transfer etc.

Two Serial ports are included on the card, the first accessible externally using the standard 9 pin male D-connector, and the second using a serial cable to on-board headers. Each independent Serial Port may be jumper configured as COM 1 - C0M8 with interrupt line jumpers set to IRQ 2-7,10-12,14-15.

A fully detailed installation guide showing jumper settings for DOS, Windows and QS/2 is included in the 41 page manual. Also included is a utility disk with sample programs, Test and Terminal solfware with source code. Maxmum performance for Windows users can be obtained with the optional 16550 FIF0 chip version. This provides a 16 bit input and 16 bit output hardware buffer for each serial port allowing high data rates without data loss or overrun errors.

For further information contact Brain Boxes on 0151 220 2500.

Fast JEDEC0 compliant 2Mbit VRAM

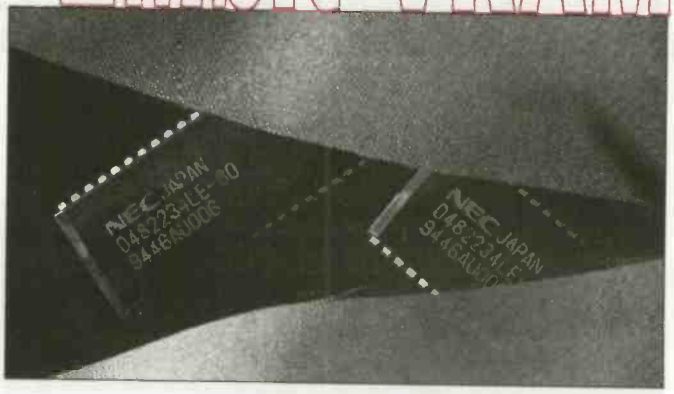
NEC Electronics is sampling fully JEDEC-compliant 2Mbit video RAMs (VRAMs) with random access times of 6Ons. JEDEC is a US-based standards body that defines the functionality and pin-outs for many semiconductor devices. Compliancy means that users of the uPD482234 and uPD482235 products, such as PC manufacturers, will not be tied to a particular manufacturer and will have flexibility of supply.

A VRAM is essentially a DRAM with an extra I/0 port, enabling a faster data transfer rate. They are used in desktop and portable PCs (particularly those running Windows-type applications), workstations, video cameras and video recorders.

The 2Mbit devices are organised as 256K x 8 bits, and are ideal for use in graphics applications. They are functionally compatible with previously released 2Mbit VRAMS but offer the advantage of faster serial access times down to l8ns.

The 2Mbit VRAM is available in two operational modes, both with random access performance down to 6Ons. The HPD482234 is a fast page mode device, while the HPD482235 is an extended output fast page mode device which can improve the fast page read performance by at least 20%.

Both versions feature a full-width split serial buffer, the two halves of which function independently. Hence one half can be outputting data while the other half is being loaded with the next data to be output. This removes the need to pause between reading rows (that is, it eliminates "flyback" time) and permits data to be output

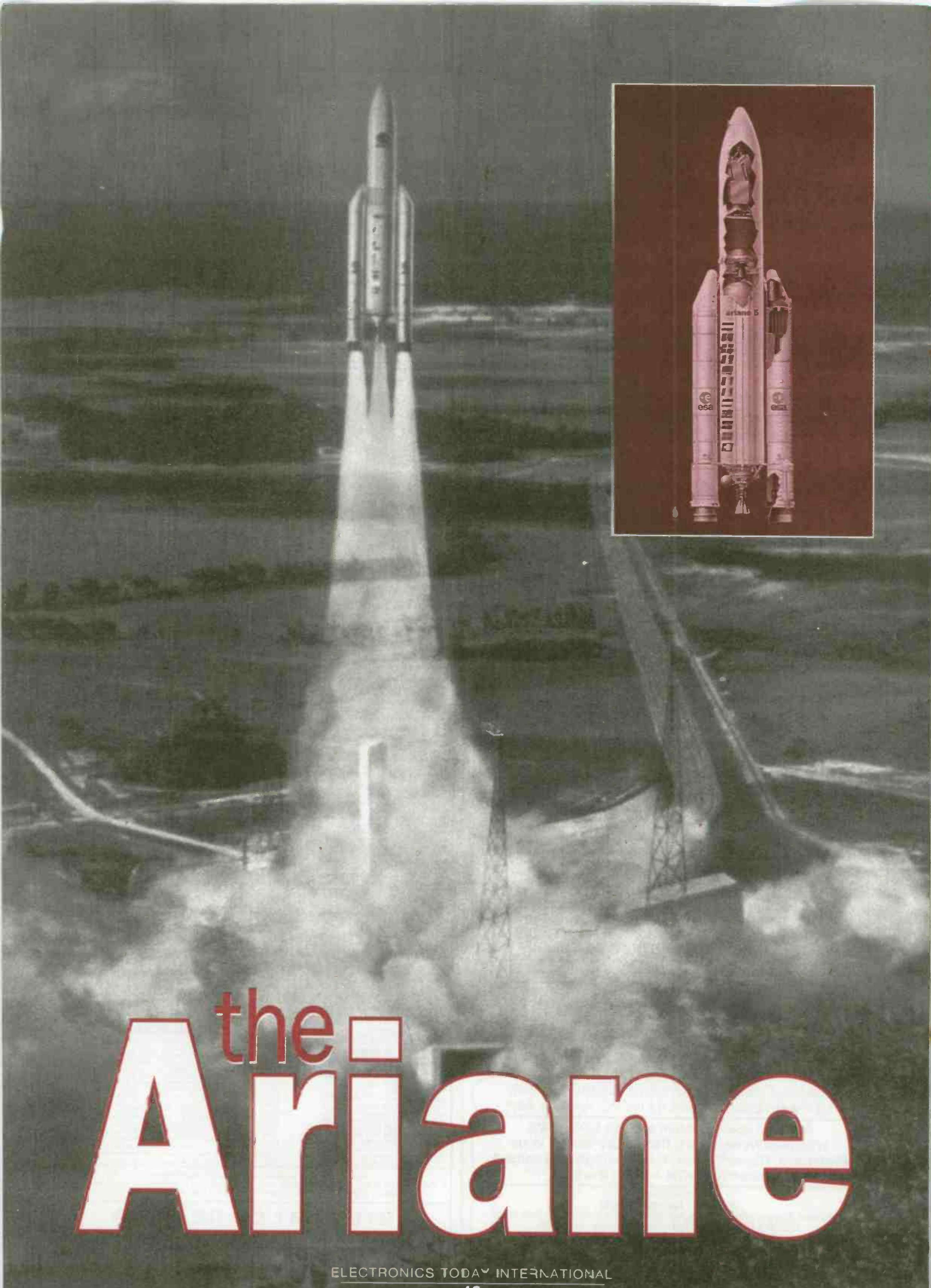


continuously at full speed.

The serial buffer is bi-directional, so the VRAMs are suitable for use in serial input applications such as video cameras or video recorders.

Other capabilities of the devices include a flash write mode, a block write mode, and the stopping column mode, which accelerates tiling operations. These features greatly increase the performance of the system when running Windows-type operations.

For more informAtion contact NEC Elecxtronics (UK) Ltd on 01908 691133.



the Ariane

The European Space Agency is about to launch a new ultra-reliable, high technology, rocket which is powerful enough for manned space launches. Douglas Clarkson takes a look at the system and the technology

The development of a new space launch system is one of the most expensive areas of modern technology. Such projects require specialist scientific and technical skills of a very diverse and precise nature. Also, the emphasis on ultra-high reliability places special emphasis on rigorous testing and proving of all components - both in the unassembled form and also in assembled modules such as rocket motors. This forces medium sized countries such as exist within Europe to form a collaborative agency such as the European Space Agency (ESA). The current membership and locations of major sites and offices of the European Space Agency (for which Ariane 5 Project is a major development item) is shown in figure 1.

The Ariane 5 programme was commissioned in November 1987. Few could have anticipated the major political changes that have subsequently taken place. The Soviet Union is no more, Germany is again a unified country and a severe recession has been experienced by Western countries. The key theme of the Ariane 5 rocket was to give Europe an independent launch capability for both routine launch satellites and also servicing the planned Space Station.

In its prime, initial role, however, the new Ariane 5 is an unmanned commercial launch vehicle with only minimal re-usable elements - the twin solid fuel rockets of the system. Such solid booster rockets are also used on NASA's Space Shuttle. The aspect of catering for a manned crew and the necessary life support infrastructure increased significantly the size and weight of NASA's 're-usable' Shuttle compared with that of an unmanned vehicle. A very significant element of NASA's budget is spent on the Shuttle missions. Each Shuttle mission costs in excess of $150 million. NASA is well aware of the desirability of reducing the cost of the next generation of launch vehicles.

While other designs such as HOTOL and its variants are still on the drawing board for the next generation of ESA launch vehicles, Ariane 5 will provide a continuing independent capability for Europe well into the next century. Within the European Space Agency, the FESTIP programme is seeking to identify the optimum post Ariane 5 launch vehicle.

There seems, however, a clear window of opportunity to develop a significantly cheaper launch vehicle. There is every reason to believe that the solution could be a HOTOL variant - a single stage to orbit vehicle with extra light superstructure and with an air breathing engine for the ascent in the atmosphere to 28 km. Such developments, however, are very closely tied to materials science. There is a requirement to produce materials with lightweight, increased strength and for specific zones of the craft high temperature tolerance e.g. on heated re-entry surfaces. While such development would have a direct relevance to space flight, such developments in materials science would also find useful application throughout the aerospace and other high technology industries.

The Ariane Family

The first test flight of Ariane 1 took place on 24th December 1979. The first operational flight of an Ariane rocket, however, took place on 4th of August 1984 as an Ariane 2/3 system. The first flight of the Ariane 4 system took place on 15th June 1988 and a typical Ariane 4 launch is shown in Figure 2. The central rocket motor is flanked by four peripheral rocket motors units. The Ariane 4 can deliver a single payload of around 10 tonnes into low earth orbit.

5

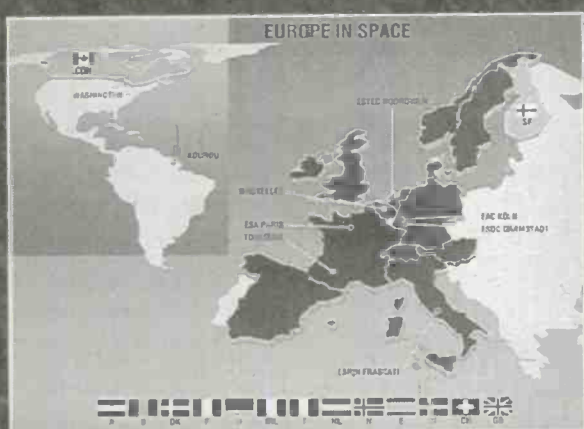


Figure 1: Participating states and locations of major sites and offices of the European Space Agency.

Ariane 5: A Summary

The basic design of the new Ariane 5 can be separated into the lower standard propulsion section and the upper payload section as indicated in figure 3. There is some degree of flexibility in the upper payload section which can be a single payload launch (unmanned), a dual payload launch (unmanned) or a manned flight stage. The prime interest now is to develop the vehicle system as a commercial launch craft for unmanned satellites. The final design of the Ariane 5 was approved during 1988 after the first successful launch of the Ariane 4 system. Figure 4 shows a cut away representation of the Ariane 5 craft.

The main difference between Ariane 4 and Ariane 5 is that the payload capacity has been almost doubled to 20.5 tonnes for low earth orbit or 6.8 tonnes to a high geostationary orbit such as is required for satellite broadcast and telecommunications. Also, part of the assembly - the new solid fuel booster stages are recoverable items. After being jettisoned from the cryogenic stage at a height of around 60 km, parachutes in the nose section of each spent booster rocket will be deployed in the atmosphere and the sections recovered by surface ships for subsequent inspection and possible re-use. While Ariane 5 is designed to be cheaper per launch than the Ariane 4 by around 10%, future generations of launch vehicles, utilising single stage to orbit technology will hopefully be significantly cheaper.

Ariane 5: The Fingers on the button

Table 1 shows the relative contribution from member states of the European Union and other associated countries to the Ariane 5 project.

Table 1
Contributions from participating countries

Country	Percentage Contribution
France	46.2
Germany	22
Italy	15
Belgium	6
Spain	3
Holland	2.1
Sweden	2
Switzerland	2
Norway	0.6
Austria	0.4
Denmark	0.4
Ireland	0.2
Not Covered	0.1

There is one major member of the European Union missing from this list - the UK, whose government seems to think that the development of space technology is a waste of money, and is therefore denying this country's aerospace industry any significant role in the future of a very rapidly expanding high technology area.

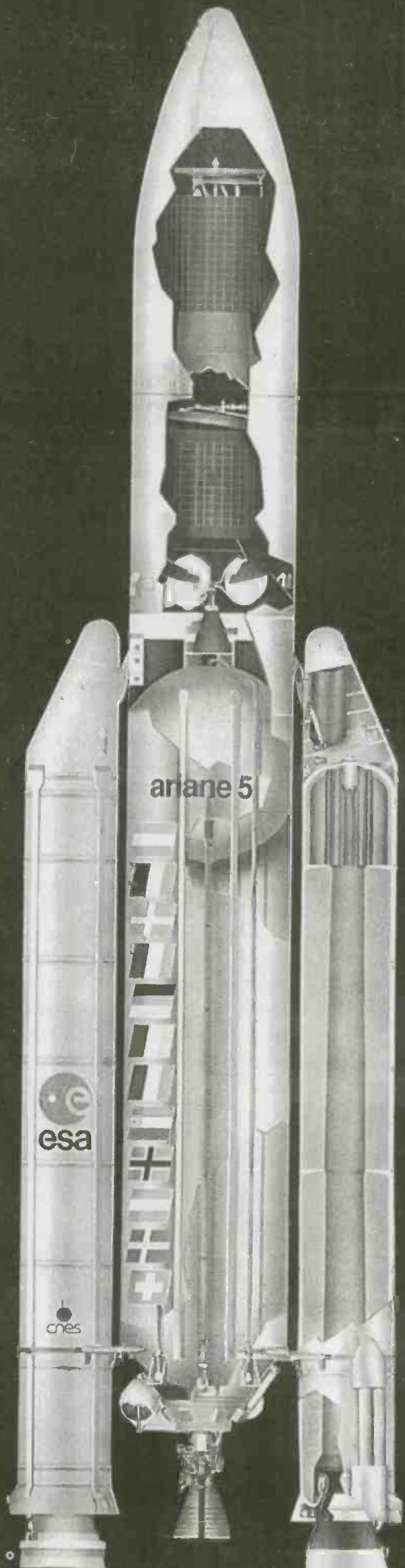


Figure 4: Cutaway representation of the Ariane 5 craft.

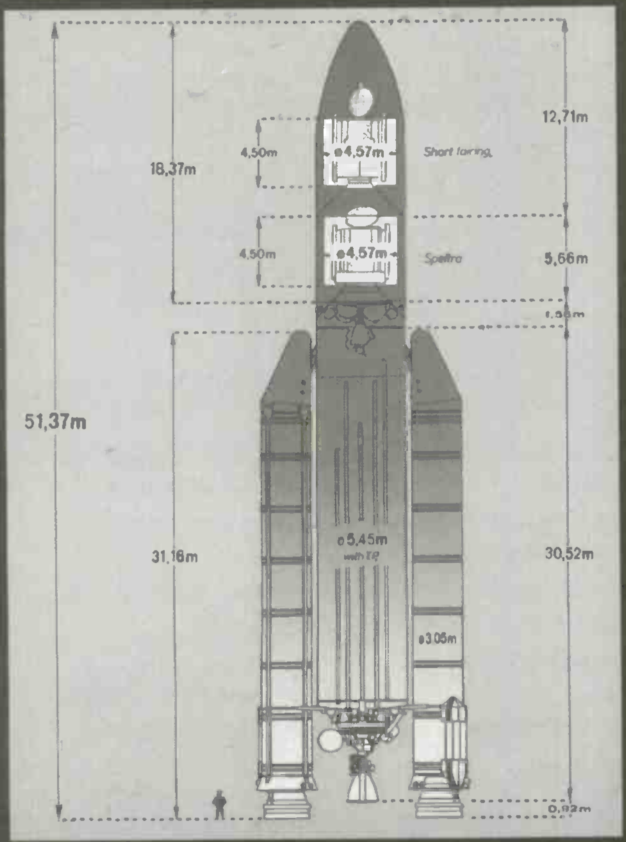


Figure 3: Structural design of Ariane 5 dual launch facility. The propulsion system consists of a central cryogenic stage and twin solid fuel boosters.

The Cryogenic Main Stage

Liquid hydrogen and liquid oxygen are used as fuel for the main cryogenic stage. This combination generates an excellent exhaust velocity of combustion products. Only liquid hydrogen and liquid fluorine react to produce a slightly higher exhaust velocity. The unit is some 5.4 m in diameter and 30.5 metres high and powered by a single Vulcain engine which is shown in figure 5. Between 114 and 120 tonnes of thrust are developed in a vacuum. The fuel contents of 25 tonnes of liquid hydrogen and 130 tonnes of liquid oxygen are burnt in just 570 seconds. The cryogenic stage with all its superb engineering, however, is expendable. When the cryogenic stage is exhausted it will fall away from the flight path and be lost in the Atlantic Ocean.

Testing of the cryogenic main stage is a key point of the Ariane project. The first test firing of a completed engine took place at Vernon (France) in July 1990. A second engine test facility was later commissioned later that year in Lampoldshausen in October 1990. By April 1994, around 180 tests had been completed on a total of 13 engines. The test programme should enable 350 tests to be completed on 16 engines and accumulate a total burn time of 90,000 seconds. This is equivalent to the burn time of some 160 missions. The test programme, however, is designed to optimise the manufacture and quality control of entire cryogenic systems.

Key components of the rocket motor such as the oxygen turbo pump are manufactured under clean room conditions. Part of the extensive series of tests undergone by the main cryogenic motor include vibration tests on a specifically designed shaker system. Such tests ensure that the structures remain secure during the initial ascent mode.

Solid Booster Stage

In common with the Space Shuttle, Ariane 5 makes use of twin solid stage booster rockets. Each booster is 3.03m in diameter, 30 metres high and has a lift-off mass of 270 tonnes. At take-off, each rocket delivers some 540 tonnes of thrust, though this is reduced after take-off to reduce the aerodynamic load on the launcher in the lower atmosphere. The solid stage boosters burn for a total of 130 seconds and contain 237 tonnes of solid propellant. The design and construction of the Ariane 5 solid rocket stages is somewhat reminiscent of an exotic 5th of November rocket. All of the components, the containing case, the propellant mixture and nozzle etc are designed to very high levels of precision and on a much larger scale.

As was discovered in the Challenger accident with NASA's Space Shuttle, a key feature is the robustness of the seals between the segments of the containment structure. The segments of the steel booster case are extensively checked by ultrasound to detect defects and cracks in metal structures. Each nozzle is fitted with a flexible bearing which can angle the booster thrust direction by up to 6 degrees on two axes.

During project development, elements of the booster stage are manufactured and tested at various sites. Booster cases are produced at Augsburg; thermal protection of casing and filling of forward booster section with propellant is undertaken at Colleferro (near Rome); nozzle manufacture and assembly is undertaken at Bordeaux while filling of the central and rear booster propellant, booster stage assembly and function tests have been undertaken at Kourou (French Guyana) since early 1993. Figure 6 shows an assembled booster stage Performance can be checked on a gantry mounted over a solid granite trench some 60m deep and 200m long as indicated in figure 7.

Upper Composite:VEH, SPELTRA and Fairings

The 'Upper composite' sits on the top of the main cryogenic stage and represents essentially the facilities for delivering the system payload into orbit. This stage has also a propulsion system but with much smaller fuel load than the main cryogenic stage. A specific engine - the Aestus - has been developed to deliver a maximum thrust of 27.5 kN. Girdled round the lower propulsion stage in the upper composite is the Vehicle Equipment Bay. This is where the complex control equipment such as internal navigation, on board computer and switching unit are housed.

The SPELTRA satellite carrier structure is designed to securely contain multiple payloads. The maximum outer diameter of this structure is 5.4 metres and is 20 metres high. For this configuration, a short fairing unit (twin shells at nose of rocket) is used. For single stage payloads, either a long or a short fairing unit (split shells) is required. The fairings are jetissoned at an altitude of around 110 km.

It is interesting to note a subtle difference between launches such as the Shuttle and rockets such as Ariane 5. In terms of adding to the clutter of rubbish in space, the Ariane system, much like traditional rocket systems, will add in orbit round the earth the Vehicle Equipment Housing for single stages and in addition the SPELTRA

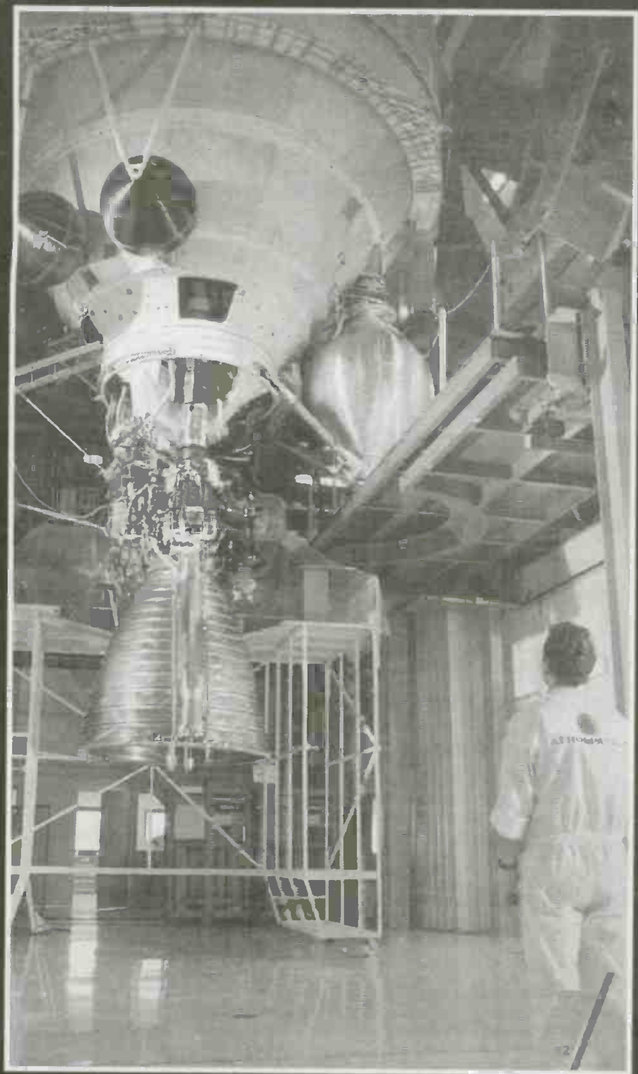


Figure 5: View of main Vulcain cryogenic engine

unit for multiple payloads. The Shuttle, however, should add very little by way of space junk into space. The large 'drop tank' of cryogenic fuel jettisoned by the Shuttle falls into the sea and the solid booster rockets are recovered.

Ground Facilities in French Guyana

It is planned to undertake up to 8 launches per month at the new launch pad facilities - ELA 3. This complex is based on the ELA 2 site used currently for the Ariane 4. Figure 8 shows an artist's impression of an Ariane 5 launch.

While the forward segment of the propellant stage is packed in Italy, the centre and rear segments are packed in a specially constructed Solid Propellant Production Unit which has been in service since 1992. For safety reasons these facilities are remote from other installations. As part of the process of final filling of propellant in French Guyana, the booster case is lined with a highly efficient insulating liner.

A cryogenic production unit for supply of liquid hydrogen and oxygen has been established close to the new Ariane 5 launch pad.

General Industrial Framework

Table 2 summarises the key industrial partners involved in the Ariane 5 Project.
Often, the main contractor will sub-contract work to a further range of sub-contractors. Thus, over 30 key sub-contractors are involved in the manufacture of the main cryogenic stage. The cost of the Ariane programme is set at 6130 MAU (Million Accounting units) at 1993 economic activity levels. The distribution of work among contractors reflects also the budget contributions of countries associated with the Ariane 5 project.

It is clear that the aerospace companies of major members of the European Union are not benefiting from the Ariane 5 project.

Table 2
Key industrial partners involved

Feature	Main Contractor
Solid Booster Housing	Aerospatiale
Booster Propellant /thermal system	Europropulsion
Main Cryogenic Stage	Aerospatiale
Cryogenic Motor	SEP
EPS	Deutch Aerospace/ERNO
Vehicle Equipment bay	Matra Marconi Space
Speltra	Dornier/Deutche Aerospace
Fairings	Oerlikon Contraves

While the ESA retains overall management of the Ariane 5 programme, CNES, the French Space Agency, is responsible for project management. Aerospatiale, in turn, is contracted to be the key industrial architect for system design and general software development. Aerospatiale therefore has the primary role of overall design and specifications of the major system components and modules.

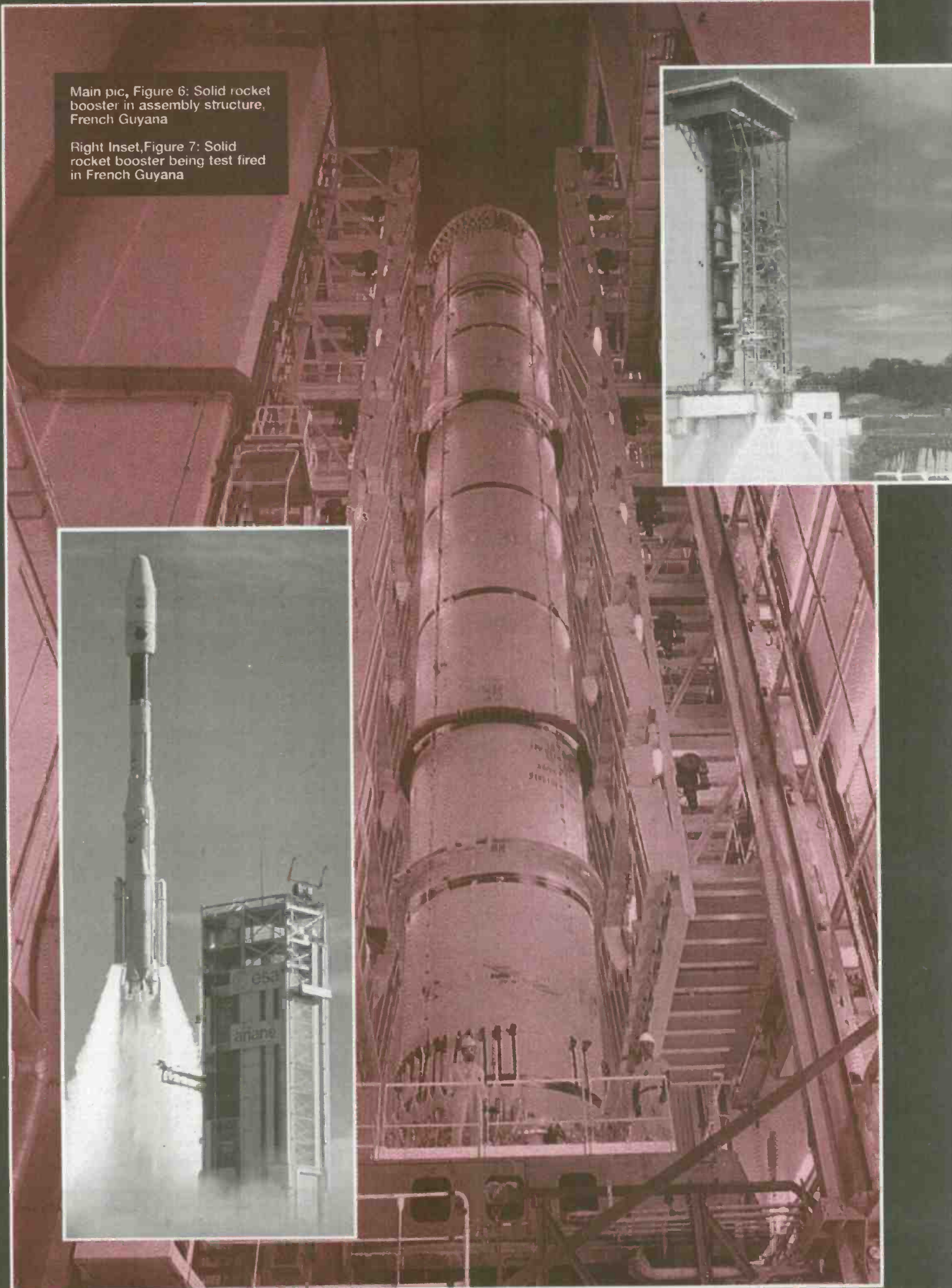
The Ariane 5 project is not only demanding in relation to the development of appropriate technology, it is also demanding in relation to project management. Even the best of technology can succumb to poor organisation and communications. The programme of development, however, since the beginning of the Ariane project has hopefully allowed such skills to develop. Also, the increasing ease of interchange of data across computer networks has allowed remote sites to exchange data in increasingly effective ways so that physical distance is not now an obstacle to collaborative development.

Summary

Whatever the pros and cons of development by Europe of the Ariane 5 at a political level, the one positive point is that by participating in such a project, Europe can be seen to be developing and retaining skills which will maintain a high level of competence in space vehicle launch technology. The investment in specific areas of technology will also allow the various companies involved to establish a foothold in such space technology niches on a global scale - to the benefit of the European Union as a whole.

Point of Contact:

European Space Agency,
8-10, rue Mario-Nikis,
75738 PARIS Cedex 15,
France.



Main pic, Figure 6: Solid rocket booster in assembly structure, French Guyana

Right Inset,Figure 7: Solid rocket booster being test fired in French Guyana

Dry Battery tester

Most ETI readers could probably turn up, without much effort, several sizes of dry cell in unknown states of charge, and the proud owner of a multimeter will be familiar with requests to "just check these out". A meter can tell you when a battery is beyond hope, of course, but to determine whether the remaining charge is sufficient for a particular application requires something a little more. Equally, cells known to balk at the 500mA drain of a torch might give hours more service in a portable radio and, with the tester presented here, it only takes a moment to check.

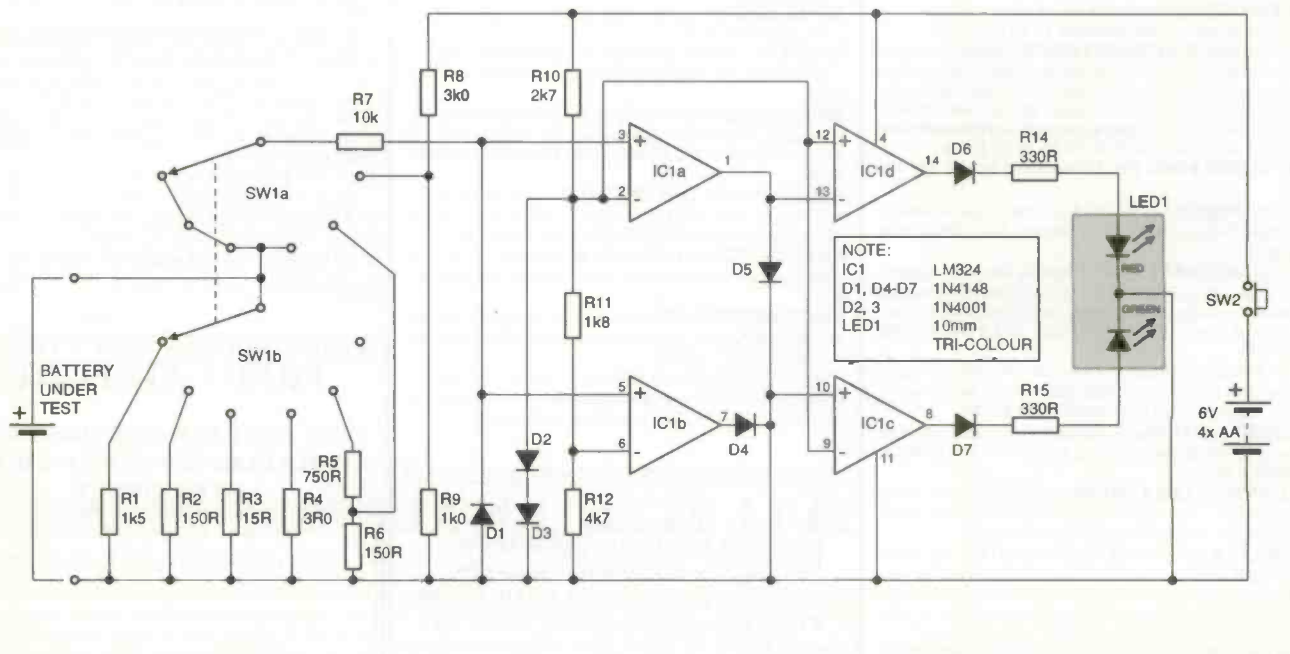


Fig. 1. Dry battery tester circuit

The common types of non-rechargeable batteries - zinc/carbon, zinc/chloride, and alkaline/manganese - give a nominal terminal voltage of 1.5V per cell when fresh. As a cell is discharged by its load, its internal impedance rises and this impedance forms a potential divider with the load, resulting in a reduced terminal voltage. Transplanting a partially discharged cell from a high to a low-drain application can therefore temporarily "restore" some extra terminal voltage, and allow some more life to be squeezed from what is nowadays a fairly expensive commodity. Of course, the environmental advantages of this approach are well known.

This design offers the following combinations of battery type and loading (suggestions are made of applications with similar loading):

Table 1
Battery types

Battery type	Voltage	Current loading	Application
1)	1.5V	1mA	alarm clocks
2)	1. 5V	lOmA	portable radio, remote control
3)	1.5V	lOOmA	personal stereo
4)	1.5V	5OOmA	torch, doorbell
5)	9V	1OmA	multimeter, smoke alarm, etc.
6)			Internal 6V battery check

I have found that an alarm clock running off one 1.5V AA cell will work for in excess of a year on a battery that is no good for anything else.

Construction

Figure 2 shows the component overlay if the PCB is used, but experienced constructors should have no trouble laying it out on stripboard. As usual, take care with the polarities of the diodes and IC. The rotary switch is likely to be supplied in solder tag form, so it will be necessary to snip off the "eyes" leaving pins to fit through the board. The off-board wiring is also shown in Figure 2, and the whole circuit is a comfortable fit into an ABS box, type MB3, as long as the tags on SW2 are bent perpendicular to its body to avoid fouling the circuit board. Figure 3 shows the box drilling details used on the prototype, although there's plenty of scope for personal taste. Connections to the battery under test were made with extra-flexible wire terminated in standard test prods.

In use

Operation of the unit is straightforward. To see if a battery is sufficiently charged for a given application, use the rotary switch to select a similar loading to that which the battery will see when in use (refer to the above table for examples). Then, with the test prods held to the corresponding battery terminals, depress the push-button. A green light indicates that the battery can maintain its voltage while operating into the selected loading, and hence that it is suitable for the chosen application. A yellow light implies a borderline situation, usually meaning that the battery will operate into the selected loading for only a short period of time before becoming too deeply discharged to be suitable - this advance warning can hopefully be used to obtain fresh cells. Finally, red indicates that the battery under test is incapable of sustaining the selected loading, and is suitable only for lower drain applications or the dustbin.

The self-test position should be used occasionally, though with sporadic use the life of the internal batteries should be long. Suitable replacements can be auditioned using the lOmA test Position!

How It Works

The reference voltages to which that of the cell under test is compared are derived from a pair of forward biased power rectifiers, D2 and D3. At the low current fed to them by RIO, the forward voltage of an 1N4001 is approximately O.6V (a signal diode such as a IN4148 is somewhat higher), giving an upper reference of 1.2V which is fed to comparator ICla. RII and R12 divide this down to a lower reference of O.9V for the second comparator IClb.

SWI carries out the combined task of selecting both the

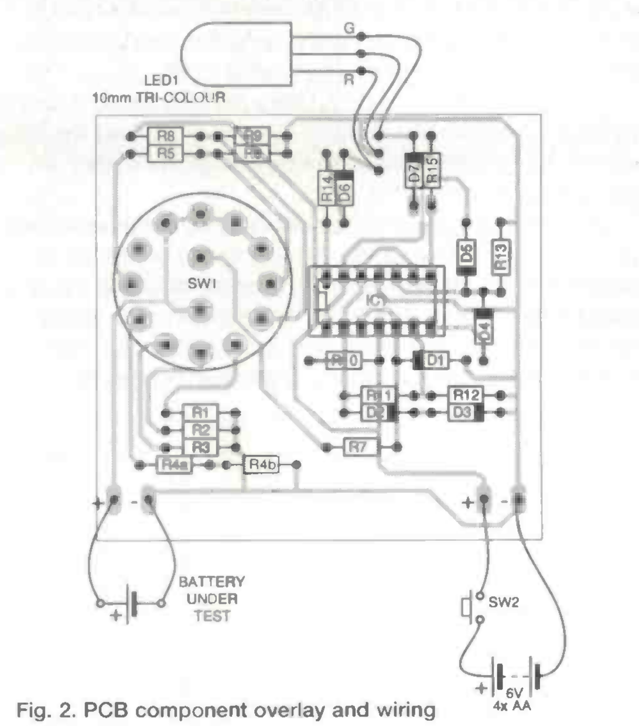


Fig. 2. PCB component overlay and wiring

type of cell to be tested and the loading presented to it. The first four positions are intended for l.5V cells, and the loading ranges from lmA (from a lK5 load resistor, Rl) to 5OOmA (with 3R0 of resistance from R4). In these positions, the cell voltage is fed directly - except for R7 and Dl, which protect the tester against reverse cell connection - to the non inverting inputs of the two comparators. The fifth position is for 9V PP3 batteries, R5 and R6 providing lOmA of loading whilst also dividing a full scale input of 9V down to l.5V for the comparators. With the switch in the final position, the tester gives an indication of the state of its own batteries, with R8 and R9 dividing in a similar way.

To translate the comparator outputs - IClb output goes high with a cell voltage exceeding O.9V, ICla when it exceeds 1.2V - to ones suitable for driving a tri-colour LED requires a bit of simple logic. Rather than use another IC, it seemed better to exploit the fact that a quad op-amp is often as cheap as a dual, and configure IClc and ICld as crude logic gates. The 1.2V reference is used to set their "logic threshold", and IC1d acts as an inverter while IClc with D4, D5, and R13 is an OR gate. To get the LED to illuminate red for cell voltages less than 0.9V, amber for between 0.9V and I.2V, and green for 1.2V or greater, the following truth table is needed:

Table 2
Truth table for different devices

IC1a O/P	IC1b O/P	red half LED	green half LED
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	1

Keen-eyed logicians will notice that the third row of the truth table is actually irrelevant, since a cell voltage cannot simultaneously register as less than 0.9V AND greater than 1.2V, and that this makes D5 similarly redundant. I have to own up to this oversight, and it is up to the constructor whether to save three pence by leaving it out!

The outputs of IClc and ICld drive the LED sections directly through current limiting resistors. D6 and D7 protect the LEDs while in the off-state from a reverse voltage of 6V (they are specified at 5V maximum).

SW2 provides power to the circuitry only while held down, preventing accidental battery drain while out of use. By its nature, the unit will only be used "every so often" so this is a useful measure. Note, however, that the cell under test is loaded even when the tester itself is off, so it is important not to leave one connected for any longer than necessary.

PARTS LIST

Resistors (0.25W, 5% unless otherwise stated)

RI	IK5	R8	3K0
R2,6	150R	R9	IK0
R3	15R	RIO	2K7
R4	2 off IR5 0.6W	RII	IK8
R5	750R	RI2	4K7
R7,I3	I0K	R14,I5	330R

Semiconductors

- ICI LM324
- DI,4,5,6,7 IN4I48
- D2,3 IN400I
- LEDI I0mm Tricolour with clip

Switches

- SwI 2-pole, 6-way rotary
- SW2 Miniature push-to-make (momentary action)

Miscellaneous

- Plastic box type MB3
- Red and black test prods
- Red and black extra-flexible wire
- 14 way IC Socket (optional)
- Knob for Switch (to personal taste)
- 4 off AA batteries plus plastic holders
- PCB or Stripboard welcome datacomp

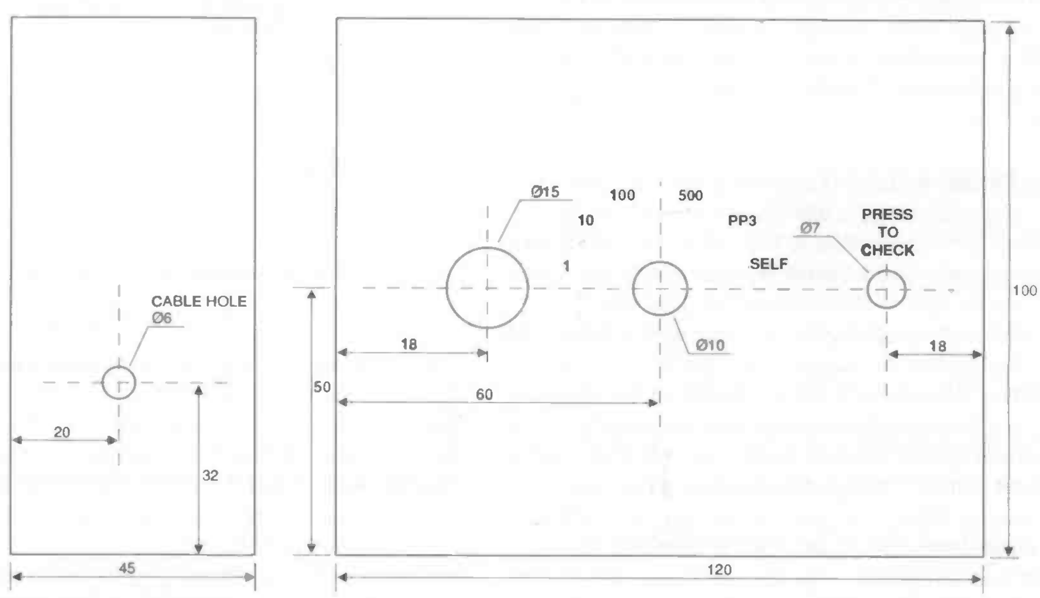


Fig. 3. Case constructon and ledgend

The ETI Basic microcontroller

In this introductory look at ETI's new autumn premiere project, Robin Abbott looks at the overall design philosophy

Next month ETI will be sold with a free PCB which is specially designed for the ETI Basic microcontroller. This is a major new project starting next month concerning a BASIC language for use with PICs. In this brief article we will take a look at the circuit of the PCB and how the peripherals may be programmed. The PCB will also be useful to anyone designing with 18 pin PIC devices since it can act as a PIC prototyping board. It will hold any 18pin PIC with crystal oscillator, a 5V power regulator, a low cost RS232 serial interface and an I2C interface for 8 pin devices. I/O access is provided through a 16 pin IC socket.

Figure 1 shows the circuit diagram of the PCB. IC3 provides the power supply, the 78L05 may be used for lower consumption projects and the 7805 can be used for projects of up to 1A consumption.

IC2 is the I2C device. The pin out shown is for EEPROM devices. However, most I2C devices have the same pinout. Operation of I2C devices is described in detail below. R1 and R2 pull up the data and clock lines used on the I2C interface. Link 1 is provided for use with the BASIC project. For normal operation, the link is left unconnected.

IC1 is the PIC. Any 18 pin device may be used - 16C54, 56, 58, 71, 84 etc. The lower 4 bits of port A are used for the I2C interface, and for the external serial interface. These bits are not available for use directly by the application- however, the RTCC input (which forms the RA4 input of the PIC16C71 and PIC16C84) can be freely used. TR1 and its associated circuitry form a power down reset circuit which clamps the clear input of the PIC to 0V when power is removed and drops below 4V. Without this, the PIC may write invalid values to the serial port or I2C device when power is removed.

The oscillator circuitry around pins 15 and 16 of the PIC is based on crystal or ceramic resonator devices. The frequency of the oscillator may be selected between 4MHz and 20MHz to allow the timing of the EEPROM and serial interfaces to operate successfully; the actual frequency is selected at compile time. Rs is shorted out by a copper track on the PCB, which should be cut if Rs is to be used with certain high frequency crystals.

The serial interface is provided as standard. It may be used with any serial interface which provides standard RS232 levels such as a PC; the output levels below 0v are taken from the input. The interface at the PIC is inactive high - i.e. when the interface is idle the input and output of the PIC is held at +5V. The start bit is signalled by a low going pulse followed by the 8 data bits.

The serial interface is a simple (and cheap) circuit which works well in practice with half duplex or full duplex links. D201 ensures that C201 can store the negative voltage which is normally present on the inactive input line, filters it, and uses it as a negative supply for the signal driven from the module. In testing, the average voltage on pin 2 of PL101 stayed below -3v which is sufficient to maintain RS232 levels although this will depend on the input impedance of the interface being driven, This circuit worked well with all PCs with which it was tested.

The I/O support from the module is provided through PL1. This is a 16 pin IC socket which should be connected to the application circuit using 16-way IDC cable with a 16 pin DIL IDC header. The pin out of the socket is shown in figure 2.

Using the serial port

I have found that a serial port is very useful when developing PIC applications, mainly as a debugging tool, even if the final project is to be a stand-alone system. It is often convenient when some trigger event occurs to print a value to the serial port which can then be read using a terminal emulator program on the host PC. Serial port driving routines take very little code on a PIC. Figure 3 shows a simple 9600bps serial driver for use with 16C5x or 16C71/84 devices and the general purpose PCB with a 3.58MHz crystal,. This example was taken from the PIC Programmer project recently featured in ETI. It occupies only 41 words of program space and needs only three bytes of memory, two of which are temporary variables which can be used elsewhere in the program.

To use the receiver, simply call RXCHAR which will wait for the next incoming byte and stores it in RXSER; on the 71/84 devices, the byte will also be returned in W. To use the transmitter save the byte to be transmitted in RXSER and call TXCHAR. As an example the program below is a complete 16c84 program which uses the code in Fig 3 to read every byte received on the serial port and send it back twice.

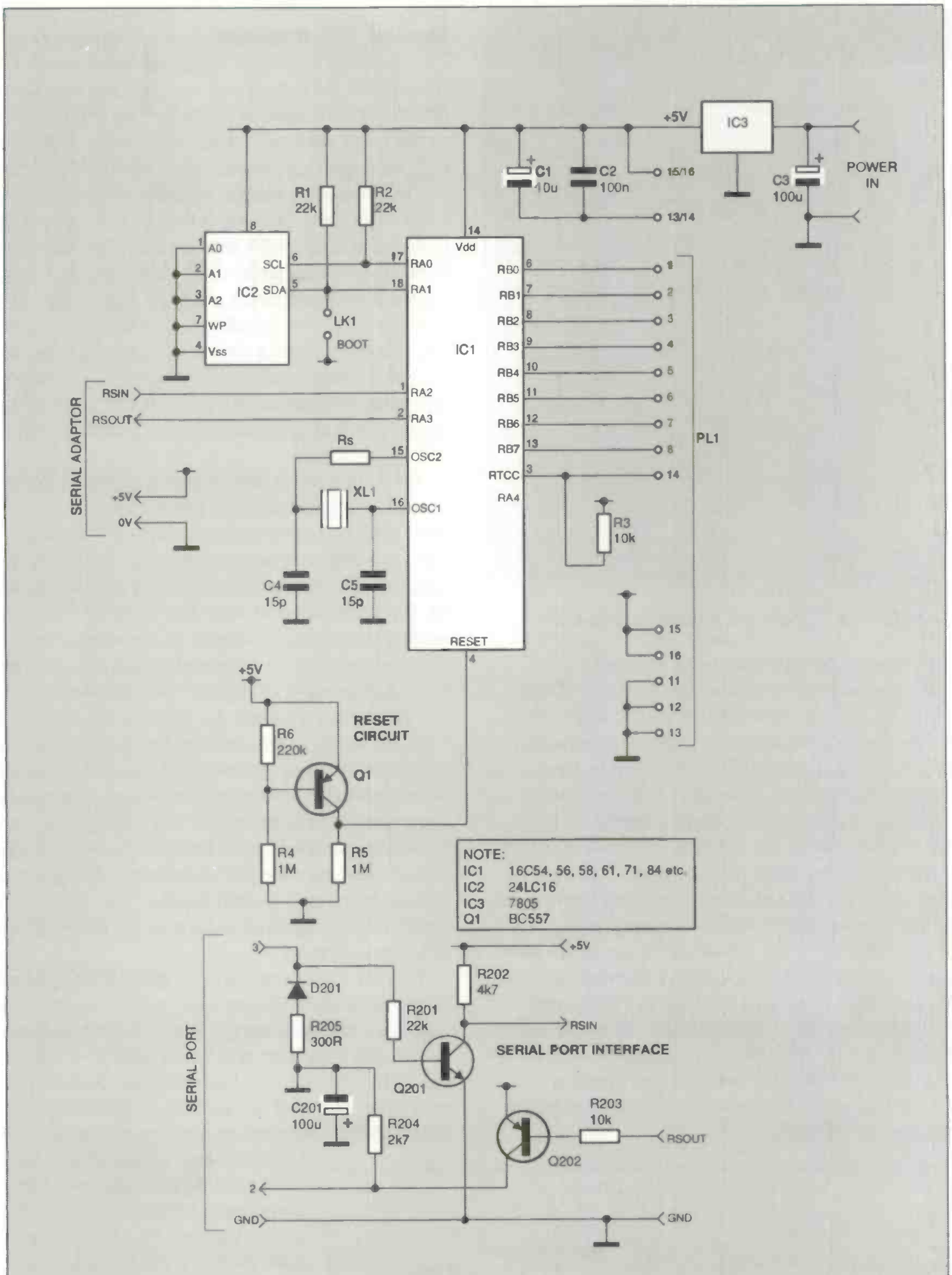


Fig.1. General purpose PIC PCB circuit

```
 SAVEBYTE EQU .15
 ORG 0

 MOVLW 0F7H                 ; SET SERIAL OUTPUT TO
DRIVE
 TRISA
 CLRW
 OPTION
LOOP  CALL RXCHAR           ; RECEIVE CHAR
 MOVWF SAVEBYTE             ; SAVE IT
 CALL TXCHAR                ; TRANSMIT
 MOVFW SAVEBYTE             ; STORE IT
 MOVWF RXSER                ; BACK IN RXSER
 CALL TXCHAR                ; TRANSMIT AGAIN
 GOTO LOOP                  ; AND GO AGAIN
```

Interrupt driven serial port

The code shown above is extremely simple, and will not transmit and receive simultaneously. Any incoming bytes during times when the processor is busy will be missed or misread. To correct this, Figure 8 shows the code for an advanced interrupt driven serial port. The application shown receives information on the serial port and sends it straight back at a full rate of 300bps with no flow control while the processor is free to perform other tasks. This is only available

Fig 2	
Socket Pin	
1	Port B bit 0
2	Port B bit 1
3	Port B bit 2
4	Port B bit 3
5	Port B bit 4
6	Port B bit 5
7	Port B bit 6
8	Port B bit 7
9	Not Used
10	Not Used
11	Ground
12	Ground
13	Ground
14	RTCC/RA4, pulled up with 10K resistor
15	+5V
16	+5V

Fig.2. Pin out of General Purpose PIC PCB control socket

for the 16C71/84 devices as it uses timer interrupts.

The serial port operates at 300bps with a crystal of 4MHz, and would operate at 1200 BPS at 20MHz. The serial port operates in full duplex mode and can receive and transmit information simultaneously and at the same time as the application program is running, using about 20% of the processor time with bytes being sent and received continuously, a lot less when the port is idle. It uses 63 words of program memory in the interrupt routine, and 11 bytes of RAM (although this could be cut down by making memory use more efficient at the cost of interrupt processing time).

Examine the example for the initialisation code which sets RTCC to operate at the 1MHz clock and to interrupt on overflow. To transmit a byte, write the byte to TXVAL, and execute the instruction BCF FLAGS,TXFREE. When the byte has been transmitted the TXFREE bit of register FLAGS will be set back to 1. When a byte has been received then the RECEIVED bit of register FLAGS will be set and the received byte will be found in RXBYTE.

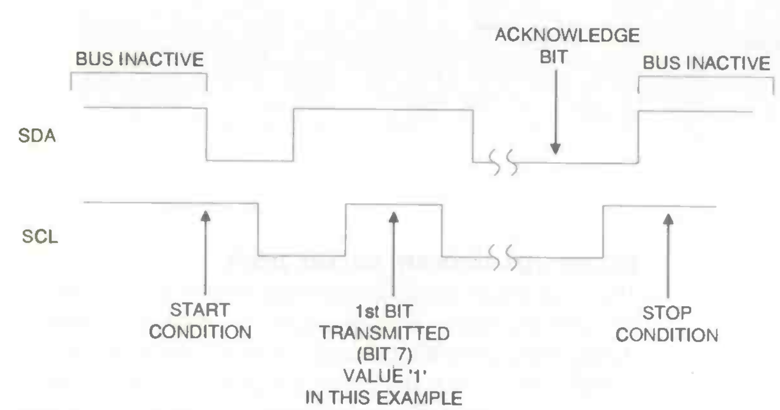


Fig.4. Start and stop conditions on I2C bus

Use of I^2C devices

The operation of the I^2C bus is explained here - it is used for a serial EEPROM in the BASIC project and may also be used for other peripheral devices which can be controlled from the module. We will look in detail at driving a 24LC16 serial EEPROM which provides a 2Kx8 EEPROM function.

The protocol is serial in nature, the SDA line is the data interface and the SCL line is the clock. In normal operation, both the SCL and SDA lines are high. The SDA line is only allowed to change when the SCL line is low. However, the exception to this is when a start or stop condition is signalled. A start condition is signalled by the SDA line dropping whilst the SCL line is high, a stop condition is signalled by the SDA line rising whilst the SCL line is high. All commands and data operations are preceded by a start condition, and transfers are ended with a stop condition. Fig 4 shows the timing for a start and stop condition.

An I^2C bus has a master device (in this case, the PIC) which controls all transfers on the bus by using the SCL and SDA lines. Data is transferred on the SDA line when the SCL line is high, and may change when the SCL line is low. Data is transferred 8 bits at a time, followed by a single acknowledge bit. Data is transferred most significant bit first. The acknowledge bit is sent either by the master or the slave device depending on the transfer, and is low for an acknowledgement, or high for no acknowledge.

Fig 5 shows a transfer for reading the current address of a 24LC16 device. In this cast the first byte (A1 hex) is the command for reading the byte at the current address. There is no acknowledgement of the transfer and the master generates a stop condition at the end of the transfer. If the master device acknowledges the transfer then the EERPROM will continue to output bytes from the EEPROM incrementing the address counter at the end of each transfer. This is the mode used by Control BASIC for rapid reading of the EEPROM when a program is running.

Figure 7 shows the full code for interfacing to a 24LC16 (or similar serial EEPROM devices). It uses four temporary registers - temp4 to temp7. Port A must be set on initialisation to drive the serial port and the I^2C SCL line. Call the routine EEREADRAND with a 2 byte address to which FSR points to read the contents of any address in EEPROM. Calling EEREADNEXT will read consecutive addresses in the EEPROM, and is much faster than calls to EEREADRAND. In both cases the read value is written to temp7, and to the W register. To write a byte put the value in the TOS register, point FSR to a 2 byte address and call EEWRITE, which takes a minimum of 10mS. Consult individual device data sheets for the control bytes and modes for I^2C devices. In a later article, a control program for interfacing to I^2C devices will be shown.

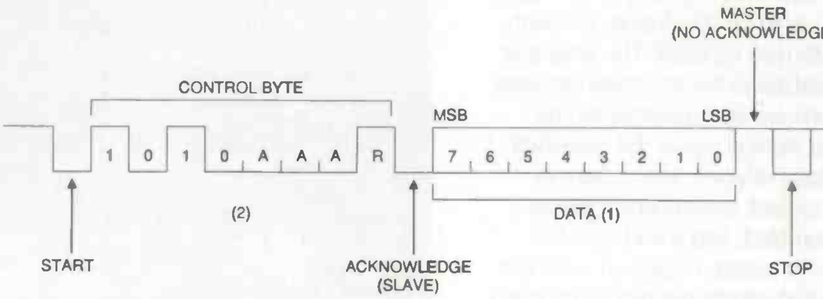
PIC BASIC

Next month, ETI will be starting a new series which allows developers to program PIC devices in BASIC. It is based on the general purpose PCB and uses 16C56, 16C71, or 16C84 devices, together with an EEPROM to hold the program. The BASIC language allows programs to be developed and tested much more rapidly than programs using PIC assembler. The EEPROM may be up to 8K in size, allowing programs of up to about 1,000 lines of BASIC code to be developed, although smaller and cheaper EEPROMS may be used. The language saves memory by allowing variables to be anywhere between 1 and 16 bits in length. Versions are also available for the larger PIC devices in the series making use of the extended memory and peripherals on these devices. The system is based on the Windows development system which allows multiple files to be edited, assembled, and linked, and for programs to be downloaded and tested. Finally, an autoboot facility allows the module to automatically run programs when it is powered up allowing stand-alone systems to be developed.

```
temp     EQU      tosh+1              ; Two
temporary files
temp2    porta             EQU 5               ;
Address of port A
rx  EQU 2                  ; Bit on port A used for
                            receiving
tx                         EQU 3               ; Bit
                            on port A used for
                           transmitting
rxser                      EQU .12      ; Holds byte
                           received or transmitted
intindex                   EQU .13      ; Temporary
                                variable
intindex1                  EQU .14      ; Temporary
                                variable

;
; This is the receive handler - used for
; reception of data on the serial port
at9600bps
; at 9600bps and clock=3.58MHz then 1 bit=93
; instructions, 1/2bit=47 instructions
;
; Return character in rxser and W
;
rxchar            btfsc porta,rx      ; 1/2
average time out of here = 2T
    goto rxchar          ; 2     Wait for start bit

    movlw 8              ; 1      pick up 8 bits
    movwf intindex       ; 1
    call delay46         ; 48     delay to just
after middle of start bit
                                  ;        so that
first sample is 1.5 bits after start

rxloop                  call delay83; 85
    nop                 ; 1
    bcf status,c        ; 1      sample incoming
bit
    btfsc porta,rx      ; 1/2
    bsf status,c        ; 1
    rrf rxser,f         ; 1      rotate in data,
LSB arrives first
    decfsz intindex     ; 1
    goto rxloop         ; 2
    movfw rxser
waitend           btfss porta,rx      ; now wait
for end of last bit if it was 0.
    goto waitend
    return


; Transmit single character in W
;
txchar                  movlw 9       ; 9 bits in
total send stop afterwards
    movwf intindex
    bcf status,c        ; first bit is start bit

txloop                  btfss status,c       ; 1/2
Set output bit
    bcf porta,tx        ; 1
    btfsc status,c      ; 1/2
    bsf porta,tx        ; 1
    call delay83        ; 85
    rrf txwork,f        ; 1
    decfsz intindex     ; 1
    goto txloop         ; 2
    bsf porta,tx        ; 1
    call delay83        ; 85     ; should be long
                                 enough for stop bit
    return              ; 2
; delay routines for serial port stuff

delay46                 movlw .14    ; 1      loop
time=3.(n-1)+7
delex                   movwf intindex1     ; 1
d34loop                 decfsz intindex1    ; 1/2
    goto d34loop        ; 2
    nop                 ; 1
    return              ; 2

delay83                 movlw .25    ; 1      loop
time=3.(n-1)+11
    nop                 ; 1
    nop                 ; 1
    goto delex          ; 2
```

Fig.3. A simple 9600bps serial driver for use with the 16C5x or 16C71/84 devices and the general purpose PCB with a 4MHz crystal.

```
flags EQU     FIRST    ; Flags byte (see below)
cpc EQU                flags+1      ; Program
counter lower
cpchEQU                cpc+1        ; PC upper
sp  EQU                cpch+1       ; stack
pointer
tos EQU                sp+1 ; Useful storage for
Top of Stack
toshEQU                tos+1
tempEQU                tosh+1       ; Two
temporary files
temp2                  EQU  temp+1
temp3                  EQU  temp2+1
temp4                  EQU  temp3+1
temp5                  EQU  temp4+1
temp6                  EQU  temp5+1
temp7                  EQU  temp6+1
temp8                  EQU  temp7+1 ; Available
for use during program-only used by PC control
temp9                  EQU  temp8+1 ; Available
for use during program-only used by PC control
temp10                 EQU  temp9+1

lastEQU                temp7+1


; Bits in flags byte

rfshadd                EQU  0       ; Set to
                           indicate EEPROM
                       address need refreshing
int EQU                1    ; An interrupt
occurred

rfshaddv               EQU  1
intv                   EQU  2
;
; Communications ports
;

picio                  EQU  5

scl EQU                0    ; Serial clock, port
a, bit
sda EQU                1    ; serial data
rsin                   EQU  2
rsout                  EQU  3

sclvEQU                1    ; values of bits
sdavEQU                2
rsinv                  EQU  4
```



NOTES: 1 DATA IS SENT FROM SLAVE, MASTER MUST TRI-STATE SDA DURING THIS PERIOD

2 AAA IS THE ADDRESS OF THE I²C DEVICE, HOWEVER FOR SOME DEVICES (EG. 24LC16) IT IS THE UPPER ADDRESS BITS OF THE MEMORY BEING ADDRESSED

Fig.5. Example I²C bus read

```
rsoutv                  EQU  8
;
; 12 bit addressing model
;
; Read a random byte from EEPROM - address in
FSR return in temp7 & w

eereadrand  call eewrtadd   ; set write address
    bsf picio,sda
    bsf picio,scl          ; drive clock high
    movlw 5
    movwf temp6            ; ack & start bit
    incf fsr               ; address upper
    clrc                   ; move address left into
                             control
    rlf 0,w                ; address upper
    iorlw 0A1h             ; control byte
    call weeprom0w

; Read next byte from EEPROM return in temp7 &
w

eereadnext     movlw 0ch
eerdgo  movwf temp6        ; ack, no
                             start or stop
    movlw ~(sclv|rsoutv)     ; Now read from data
                               line
    tris picio
    call weeprom0w         ; read byte, generate
                             STOP
retee                  movfw temp7
    return

; Write a byte to EEPROM, address in FSR, byte
                            in TOS
eewrite   call eewrtadd
    movlw 6
    movwf temp6            ; ack & stop bit
    movfw tos
    goto weeprom0w

; Write address part of read or write algorithm
eewrtadd   movlw 0ah
    call eerdgo            ; Read a byte with stop
bit set to end seq read
    movlw 5
    movwf temp6            ; ack & start bit
    incf fsr               ; address upper
    clrc                   ; move address left into
                             control
    rlf 0,w                ; address upper
    iorlw 0A0              ; control byte
    call weeprom0w         ; write control byte
    decf temp6             ; Set temp6 to 4 - ack
                           low from eeprom
    decf fsr               ; MUST LEAVE FSR
```

```
POINTING AT LOWER ADDRESS!
    movfw 0                ; address lower

; Write/read 8 bits to EEPROM, check ack
; Call with scl driving
; sda is set to read for a read, and drive for
; write
; temp4 is index
; temp5 is the data to write
; temp6 bit 0 indicates to generate start
; temp6 bit 1 indicates to generate stop
; temp6 bit 2 indicates to generate ack if set
; (nack if reset)
; temp6 bit 3 indicates to read
; temp7 reads input byte
; Ends driving sda & scl

weeprom0w  movwf temp5
weeprom9   movlw 8
    movwf temp4
    btfss temp6,0          ; check startbit
    goto nostart
    bcf picio,sda          ; start bit drops data
nostart    bcf picio,scl       ; drop clock
    btfsc temp6,3          ; leap forward if read
    goto rdeelop
;
wrteelop   bcf picio,sda    ;
    rlf temp5              ; now get correct port
                             bit
    skpnc                  ;
    bsf picio,sda          ;
    bsf picio,scl          ; clock now high
    bcf picio,scl          ; clock low again
    decfsz temp4           ; and loop
    goto wrteelop          ;
    goto endthis
;
rdeelop    bsf picio,scl; clock now high
    clrc                   ;
    btfsc picio,sda        ; now read input
    setc                   ;
    rlf temp7              ; Read in bit
    bcf picio,scl          ; clock low again
    decfsz temp4           ; and loop
    goto rdeelop           ;
;
endthis    bcf picio,sda; 1 Ack low or high ?
(low if eeprom is acking)
    btfss temp6,2          ;
    bsf picio,sda
    movlw ~(sclv|rsoutv|sdav)       ; Drive scl
& sda
    tris picio
    bsf picio,scl          ; 1 Send a clock
    bcf picio,scl
```

```
    btfss temp6,1              ; 1/2  stop bit ?
    return                     ; 1    If no stop
                           bit return with scl low
    bcf picio,sda
    bsf picio,scl              ; otherwise clock
                               high
    bsf picio,sda              ; If stop bit return
                        driving high on scl/sda
    return
```

Fig.7. Code for interfacing to a 24LC16

```
include "16c84.equ"
rx   equ 2                  ; Bit for receiving
tx   equ 3                  ; Bit for transmitting
seropt                      equ 8      ;options reg
saveintst                   equ .12    ; save
                                  status in intrt
saveintw                    equ .13    ; saves w in
                                  interrupt
txintcnt                    equ .14    ; Interrupt
                                  counter
txcnt                       equ .15    ; Transmit
                                  bit counter
txval                       equ .16    ;transmitted
                                  byte
rxval                       equ .17    ; Receiving
                                  work space
rxintcnt                    equ .18    ; Interrupt
                                  counter
rxcnt                       equ .19    ; receiving
                                  bit count
rxbyte                      equ .20    ; received
                                  byte
flags                       equ .21
txfree                      equ 0      ; 1 if tx
                                  can start
txgoequ 1                   ; 1 when transmitting
rxgoequ 2                   ; 1 when receiving
received                    equ 3      ; 1 when
                                 byte received

    ORG 0
    goto start
;
; Interrupt handler
;
    org 4
    movwf saveintw          ; save w and status
    swapf saveintw,f
    swapf status,w
    movwf saveintst

    decfsz txintcnt
    goto chkrx
    movlw .13
    movwf txintcnt          ; 13 interrupts/bit
    btfss flags,txgo
    goto chktxst
    tstf txcnt                  ; stop bit over
    skpz                        ; so clear flag
    goto chktx
    bcf flags,txgo

chktxst
    btfsc flags,txfree  ; tx request ?
    goto chkrx
    bsf flags,txgo      ; We are sending
    movlw 9             ; 9 bits in total
    movwf txcnt
    bcf porta,tx        ; start bit
    goto chkrx

chktx                   decfsz txcnt; flag
transmitting
    goto txlop
    bsf flags,txfree    ; Free for tx       goto
setop                                ; stop bit high
txlop    rrf txval                   ;
    btfss status,c                   ; Set output bit
    bcf porta,tx                     ;
    btfsc status,c                   ;
setop                                bsf porta,tx;
;
; Receiving interrupt check
;
chkrx  btfsc flags,rxgo; receiving ?
    goto rxing
    btfsc porta,rx          ; start bit ?
    goto endint
    movlw 8                 ; 8 bits
    movwf rxcnt
    movlw .19               ; delay to 1/2 way
    movwf rxintcnt          ; thru 1st bit
    bsf flags,rxgo          ; show receiving
    goto endint
rxing                       decfsz rxintcnt      ;
check timer count
    goto endint
    tstf rxcnt                  ; end of byte ?
    skpz
    goto nxtbit
    bcf flags,rxgo          ; Stop bit
    goto endint
nxtbit movlw .13
    movwf rxintcnt
    bcf status,c            ; sample input bit
    btfsc porta,rx          ;
    bsf status,c            ;
    rrf rxval               ; move to rxval
    decfsz rxcnt            ;
    goto endint
    movfw rxval             ; save rxid byte.
    movwf rxbyte
    bsf flags,received ; Show received
                               byte
;
; Interrupt over - restore flags & w
;

endint                      movlw 0f8h  ; clear
flags
                            andwf intcon,f
                            swapf saveintst,w
                            movwf status
                            swapf saveintw,w     ;
restore w
                            retfie
; Start up code and test routine
start                       movlw 0ffh           ;
clear port a
    movwf porta
    movlw 0f7h                  ; set drive
    tris porta
    movlw seropt            ; option reg.
    option
    bsf status,rp0
    movlw gievlrtiev        ; enable interrupts
    movwf intcon
    bcf status,rp0

    movlw .13                   ; Initial transmit
    movwf txintcnt          ; states
    clrf txcnt
    clrf flags
    bsf flags,txfree        ; free to transmit
;
; Test - look for bytes and send back
;
loopbtfss flags,received      ; Byte rxid ?
    goto loop
    bcf flags,received ; clear flag
    movfw rxbyte                ; send it back
    movwf txval
    bcf flags,txfree            ; request tx
    goto loop

end
```

Fig.8. Code for an advanced interrupt driven serial port

POWER AMPLIFIER MODULES-TURNTABLES-DIMMERS-LOUDSPEAKERS-19 INCH STEREO RACK AMPLIFIERS

★ PRICES INCLUDE V.A.T. ★ PROMPT DELIVERIES ★ FRIENDLY SERVICE ★ LARGE (A4) S.A.E. 60p STAMPED FOR CATALOGUE ★

OMP MOS-FET POWER AMPLIFIERS
HIGH POWER, TWO CHANNEL 19 INCH RACK

THOUSANDS PURCHASED BY PROFESSIONAL USERS

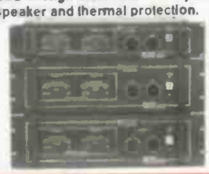
THE RENOWNED MXF SERIES OF POWER AMPLIFIERS

FOUR MODELS:- MXF200 (100W + 100W) MXF400 (200W + 200W) MXF600 (300W + 300W) MXF900 (450W + 450W)

ALL POWER RATINGS R.M.S. INTO 4 OHMS, BOTH CHANNELS DRIVEN

FEATURES: ★Independent power supplies with two toroidal transformers ★ Twin L.E.D. Vu meters ★ Level controls ★ Illuminated on/off switch ★ XLR connectors ★ Standard 775mV inputs ★ Open and short circuit proof ★ Latest Mos-Fets for stress free power delivery into virtually any load ★ High slew rate ★ Very low distortion ★ Aluminium cases ★ MXF600 & MXF900 fan cooled with D.C. loudspeaker and thermal protection.

USED THE WORLD OVER IN CLUBS, PUBS, CINEMAS, DISCOS ETC.



SIZES:- MXF200 W19"xH3½" (2U)xD11"
MXF400 W19"xH5¼" (3U)xD12"
MXF600 W19"xH5¼" (3U)xD13"
MXF900 W19"xH5¼" (3U)xD14½"

PRICES:- MXF200 £175.00 MXF400 £233.85 MXF600 £329.00 MXF900 £449.15

SPECIALIST CARRIER DEL. £12.50 EACH

OMP XO3 STEREO 3-WAY ACTIVE CROSS-OVER

Advanced 3-Way Stereo Active Cross-Over, housed in a 19" x 1U case. Each channel has three level controls: bass, mid & top. The removable front fascia allows access to the programmable DIL switches to adjust the cross-over frequency: Bass-Mid 250/500/800Hz, Mid-Top 1.8/3/5KHz, all at 24dB per octave. Bass invert switches on each bass channel. Nominal 775mV input/output. Fully compatible with OMP rack amplifier and modules.

Price £117.44 + £5.00 P&P

STEREO DISCO MIXER SDJ3400SE

★ ECHO & SOUND EFFECTS★

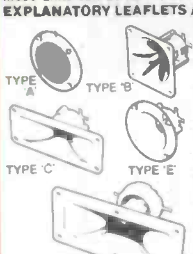
STEREO DISCO MIXER with 2 x 7 band L & R graphic equalisers with bar graph LED Vu meters. **MANY OUTSTANDING FEATURES:- including Echo with repeat & speed control, DJ Mic with talk-over switch, 6 Channels with individual faders plus cross fade, Cue Headphone Monitor. 8 Sound Effects.** Useful combination of the following inputs:- 3 turntables (mag), 3 mics, 5 Line for CD, Tape, Video etc.

Price £144.99 + £5.00 P&P

SIZE: 482 x 240 x 120mm

PIEZO ELECTRIC TWEETERS - MOTOROLA

Join the Piezo revolution! The low dynamic mass (no voice coil) of a Piezo tweeter produces an improved transient response with a lower distortion level than ordinary dynamic tweeters. As a crossover is not required these units can be added to existing speaker systems of up to 100 watts (more if two are put in series. **FREE EXPLANATORY LEAFLETS ARE SUPPLIED WITH EACH TWEETER.**



TYPE 'A' (KSN1036A) 3" round with protective wire mesh. Ideal for bookshelf and medium sized Hi-Fi speakers. Price £4.90 + 50p P&P.

TYPE 'B' (KSN1005A) 3½" super horn for general purpose speakers, disco and P.A. systems etc. Price £5.99 + 50p P&P.

TYPE 'C' (KSN1016A) 2"x5" wide dispersion horn for quality Hi-Fi systems and quality discos etc. Price £6.99 + 50p P&P.

TYPE 'D' (KSN1025A) 2"x6" wide dispersion horn. Upper frequency response retained extending down to mid-range (2KHz). Suitable for high quality Hi-Fi systems and quality discos. Price £9.99 + 50p P&P.

TYPE 'E' (KSN1038A) 3¾" horn tweeter with attractive silver finish trim. Suitable for Hi-Fi monitor systems etc. Price £5.99 + 50p P&P.

LEVEL CONTROL Combines, on a recessed mounting plate, level control and cabinet input jack socket. 85x85mm. Price £4.10 + 50p P&P.

ibl FLIGHT CASED LOUDSPEAKERS

A new range of quality loudspeakers, designed to take advantage of the latest speaker technology and enclosure designs. Both models utilize studio quality 12" cast aluminium loudspeakers with factory fitted grilles, wide dispersion constant directivity horns, extruded aluminium corner protection and steel ball corners, complimented with heavy duty black covering. The enclosures are fitted as standard with top hats for optional loudspeaker stands.

POWER RATINGS QUOTED IN WATTS RMS FOR EACH CABINET FREQUENCY RESPONSE FULL RANGE 45Hz - 20KHz

ibl FC 12-100WATTS (100dB) PRICE £159.00 PER PAIR
ibl FC 12-200WATTS (100dB) PRICE £175.00 PER PAIR

SPECIALIST CARRIER DEL. £12.50 PER PAIR

OPTIONAL STANDS PRICE PER PAIR £49.00
Delivery £6.00 per pair

IN-CAR STEREO BOOSTER AMPS

THREE SUPERB HIGH POWER CAR STEREO BOOSTER AMPLIFIERS
150 WATTS (75 + 75) Stereo, 150W Bridged Mono
250 WATTS (125 + 125) Stereo, 250W Bridged Mono
400 WATTS (200 + 200) Stereo, 400W Bridged Mono
ALL POWERS INTO 4 OHMS

Features:
★ Stereo, bridgable mono ★ Choice of high & low level inputs ★ L & R level controls ★ Remote on-off ★ Speaker & thermal protection.

PRICES: 150W £49.99 250W £99.99 400W £109.95 P&P £2.00 EACH

OMP MOS-FET POWER AMPLIFIER MODULES

SUPPLIED READY BUILT AND TESTED.

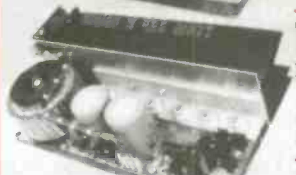
These modules now enjoy a world-wide reputation for quality, reliability and performance at a realistic price. Four models are available to suit the needs of the professional and hobby market i.e. Industry, Leisure, Instrumental and Hi-Fi etc. When comparing prices, NOTE that all models include toroidal power supply, integral heat sink, glass fibre P.C.B. and drive circuits to power a compatible Vu meter. All models are open and short circuit proof.

THOUSANDS OF MODULES PURCHASED BY PROFESSIONAL USERS

OMP/MF 100 Mos-Fet Output power 110 watts R.M.S. into 4 ohms, frequency response 1Hz - 100KHz -3dB, Damping Factor >300, Slew Rate 45V/uS, T.H.D. typical 0.002%, Input Sensitivity 500mV, S.N.R. -110 dB. Size 300 x 123 x 60mm.
PRICE £40.85 + £3.50 P&P

OMP/MF 200 Mos-Fet Output power 200 watts R.M.S. into 4 ohms, frequency response 1Hz - 100KHz -3dB, Damping Factor >300, Slew Rate 50V/uS, T.H.D. typical 0.001%, Input Sensitivity 500mV, S.N.R. -110 dB. Size 300 x 155 x 100mm.
PRICE £64.35 + £4.00 P&P

OMP/MF 300 Mos-Fet Output power 300 watts R.M.S. into 4 ohms, frequency response 1Hz - 100KHz -3dB, Damping Factor >300, Slew Rate 60V/uS, T.H.D. typical 0.001%, Input Sensitivity 500mV, S.N.R. -110 dB. Size 330 x 175 x 100mm.
PRICE £81.75 + £5.00 P&P



OMP/MF 450 Mos-Fet Output power 450 watts R.M.S. into 4 ohms, frequency response 1Hz - 100KHz -3dB, Damping Factor >300, Slew Rate 75V/uS, T.H.D. typical 0.001%, Input Sensitivity 500mV, S.N.R. -110 dB, Fan Cooled, D.C. Loudspeaker Protection, 2 Second Anti-Thump Delay. Size 385 x 210 x 105mm.
PRICE £132.85 + £5.00 P&P

OMP/MF 1000 Mos-Fet Output power 1000 watts R.M.S. into 2 ohms, 725 watts R.M.S. into 4 ohms, frequency response 1Hz - 100KHz -3dB, Damping Factor >300, Slew Rate 75V/uS, T.H.D. typical 0.002%, Input Sensitivity 500mV, S.N.R. -110 dB, Fan Cooled, D.C. Loudspeaker Protection, 2 Second Anti-Thump Delay. Size 422 x 300 x 125mm.
PRICE £259.00 + £12.00 P&P

NOTE: MOS-FET MODULES ARE AVAILABLE IN TWO VERSIONS: STANDARD - INPUT SENS 500mV, BAND WIDTH 100KHz. PEC (PROFESSIONAL EQUIPMENT COMPATIBLE) - INPUT SENS 775mV, BAND WIDTH 50KHz. ORDER STANDARD OR PEC.

LOUDSPEAKERS

LARGE SELECTION OF SPECIALIST LOUDSPEAKERS AVAILABLE, INCLUDING CABINET FITTINGS, SPEAKER GRILLES, CROSS-OVERS AND HIGH POWER, HIGH FREQUENCY BULLETS AND HORNS, LARGE (A4) S.A.E. (60p STAMPED) FOR COMPLETE LIST.

McKenzie and Fane Loudspeakers are also available.

EMINENCE:- INSTRUMENTS, P.A., DISCO, ETC

ALL EMINENCE UNITS 8 OHMS IMPEDANCE

8" 100 WATT R.M.S. ME8-100 GEN. PURPOSE, LEAD GUITAR, EXCELLENT MID, DISCO.
RES. FREQ. 72Hz, FREQ. RESP. TO 4KHz, SENS 97dB. PRICE £32.71 + £2.00 P&P

10" 100 WATT R.M.S. ME10-100 GUITAR, VOCAL, KEYBOARD, DISCO, EXCELLENT MID.
RES. FREQ. 71Hz, FREQ. RESP. TO 7KHz, SENS97dB. PRICE £33.74 + £2.50 P&P

10" 200 WATT R.M.S. ME10-200 GUITAR, KEYB'D, DISCO, VOCAL, EXCELLENT HIGH POWER MID.
RES. FREQ. 65Hz, FREQ. RESP. TO 3.5KHz, SENS 99dB. PRICE £43.47 + £2.50 P&P

12" 100 WATT R.M.S. ME12-100LE GEN. PURPOSE, LEAD GUITAR, DISCO, STAGE MONITOR.
RES.FREQ. 49Hz, FREQ. RESP. TO 6KHz, SENS 100dB. PRICE £35.64 + £3.50 P&P

12" 100 WATT R.M.S. ME12-100LT (TWIN CONE) WIDE RESPONSE, P.A., VOCAL, STAGE MONITOR. RES. FREQ 42Hz, FREQ. RESP. TO 10KHz, SENS 98dB. PRICE £36.67 + £3.50 P&P

12" 200 WATT R.M.S. ME12-200 GEN. PURPOSE, GUITAR, DISCO, VOCAL, EXCELLENT MID.
RES. FREQ. 58Hz, FREQ. RESP. TO 6KHz, SENS 98dB. PRICE £46.71 + £3.50 P&P

12" 300 WATT R.M.S ME12-300GP HIGH POWER BASS, LEAD GUITAR, KEYBOARD, DISCO ETC.
RES. FREQ. 47Hz, FREQ. RESP. TO 5KHz, SENS 103dB. PRICE £70.19 + £3.50 P&P

15" 200 WATT R.M.S. ME15-200 GEN. PURPOSE BASS, INCLUDING BASS GUITAR.
RES. FREQ. 46Hz, FREQ. RESP. TO 5KHz, SENS 99dB. PRICE £50.72 + £4.00 P&P

15" 300 WATT R.M.S. ME15-300 HIGH POWER BASS, INCLUDING BASS GUITAR.
RES. FREQ. 39Hz, FREQ. RESP. TO 3KHz, SENS 103dB. PRICE £73.34 + £4.00 P&P

EARBENDERS:- HI-FI, STUDIO, IN-CAR, ETC

ALL EARBENDER UNITS 8 OHMS (Except EB8-50 & EB10-50 which are dual impedance tapped @ 4 & 8 ohm)

BASS, SINGLE CONE, HIGH COMPLIANCE, ROLLED SURROUND

8" 50watt EB8-50 DUAL IMPEDENCE, TAPPED 4/8 OHM BASS, HI-FI, IN-CAR.
RES. FREQ. 40Hz, FREQ. RESP. TO 7KHz SENS 97dB. PRICE £8.90 + £2.00 P&P

10" 50WATT EB10-50 DUAL IMPEDENCE, TAPPED 4/8 OHM BASS, HI-FI, IN-CAR.
RES. FREQ. 40Hz, FREQ. RESP. TO 5KHz, SENS. 99dB. PRICE £13.65 + £2.50 P&P

10" 100WATT EB10-100 BASS, HI-FI, STUDIO.
RES. FREQ. 35Hz, FREQ. RESP. TO 3KHz, SENS 96dB. PRICE £30.39 + £3.50 P&P

12" 100WATT EB12-100 BASS, STUDIO, HI-FI, EXCELLENT DISCO.
RES. FREQ. 26Hz, FREQ. RESP. TO 3 KHz, SENS 93dB. PRICE £42.12 + £3.50 P&P

FULL RANGE TWIN CONE, HIGH COMPLIANCE, ROLLED SURROUND

5¼" 60WATT EB5-60TC (TWIN CONE) HI-FI, MULTI-ARRAY DISCO ETC.
RES. FREQ. 63Hz, FREQ. RESP. TO 20KHz, SENS 92dB. PRICE £9.99 + £1.50 P&P

6½" 60WATT EB6-60TC (TWIN CONE) HI-FI, MULTI-ARRAY DISCO ETC.
RES. FREQ. 38Hz, FREQ. RESP. TO 20KHz, SENS 94dB. PRICE £10.99 + 1.50 P&P

8" 60WATT EB8-60TC (TWIN CONE) HI-FI, MILTI-ARRAY DISCO ETC.
RES. FREQ. 40Hz, FREQ. RESP. TO 18KHz, SENS 89dB. PRICE £12.99 + £1.50 P&P

10" 60WATT EB10-60TC (TWIN CONE) HI-FI, MULTI ARRAY DISCO ETC.
RES. FREQ. 35Hz, FREQ. RESP. TO 12KHz, SENS 98dB. PRICE £16.49 + £2.00 P&P

TRANSMITTER HOBBY KITS

PROVEN TRANSMITTER DESIGNS INCLUDING GLASS FIBRE PRINTED CIRCUIT BOARD AND HIGH QUALITY COMPONENTS COMPLETE WITH CIRCUIT AND INSTRUCTIONS

3W TRANSMITTER 80-108MHz, VARICAP CONTROLLED PROFESSIONAL PERFORMANCE, RANGE UP TO 3 MILES, SIZE 38 x 123mm, SUPPLY 12V @ 0.5AMP.
PRICE £14.85 + £1.00 P&P

FM MICRO TRANSMITTER 100-108MHz, VARICAP TUNED, COMPLETE WITH VERY SENS FET MIC, RANGE 100-300m, SIZE 56 x 46mm, SUPPLY 9V BATTERY.
PRICE £8.80 + £1.00 P&P

PHOTO: 3W FM TRANSMITTER

POSTAL CHARGES PER ORDER £1.00 MINIMUM. OFFICIAL ORDERS FROM SCHOOLS, COLLEGES, GOVT. BODIES, PLCs ETC. PRICES INCLUSIVE OF V.A.T. SALES COUNTER. VISA AND ACCESS ACCEPTED BY POST, PHONE OR FAX.

BARCLAYCARD VISA

B.K. ELECTRONICS

UNITS 1 & 5 COMET WAY, SOUTHEND-ON-SEA, ESSEX. SS2 6TR
Tel. 01702 - 527572 Fax: 01702-420243

"HELLO

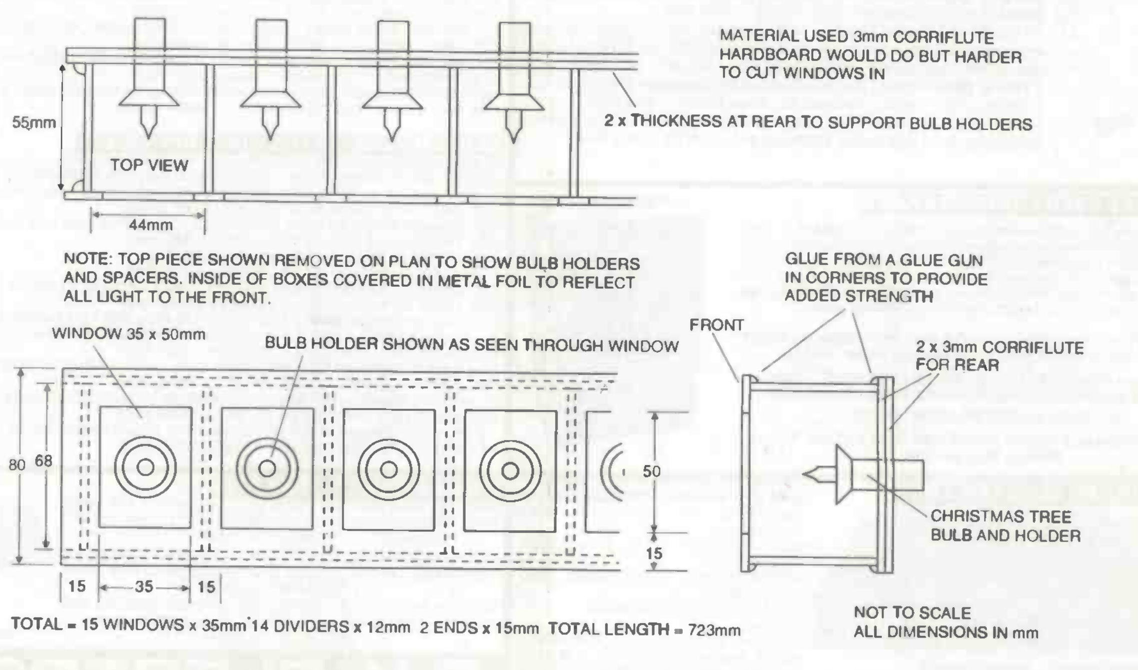
"Hello Mum" - this is just one of the printable messages that have been illuminated in the following project

A couple of years ago I was asked to design, build and supply a sign for a local primary school which would flash "Merry Christmas" in a way that would attract attention, just in case any of the parents had not been aware of the impending season of "I want". It was felt that flashing lights would add a Christmassy touch.

Recently, this project has been re-vamped to allow a class of fifth form secondary students to participate in a class project to build a message board display that would allow a message to be changed easily and that would bring into play several of the skills learnt in CDT lessons; these were to include electronics, woodwork, metalwork, corriflute construction graphics, planning delegation etc. The spec stated that the message board had to run independently of any external computers etc but could have a mains supply - only after the expense of rechargeable batteries had ruled their use out.

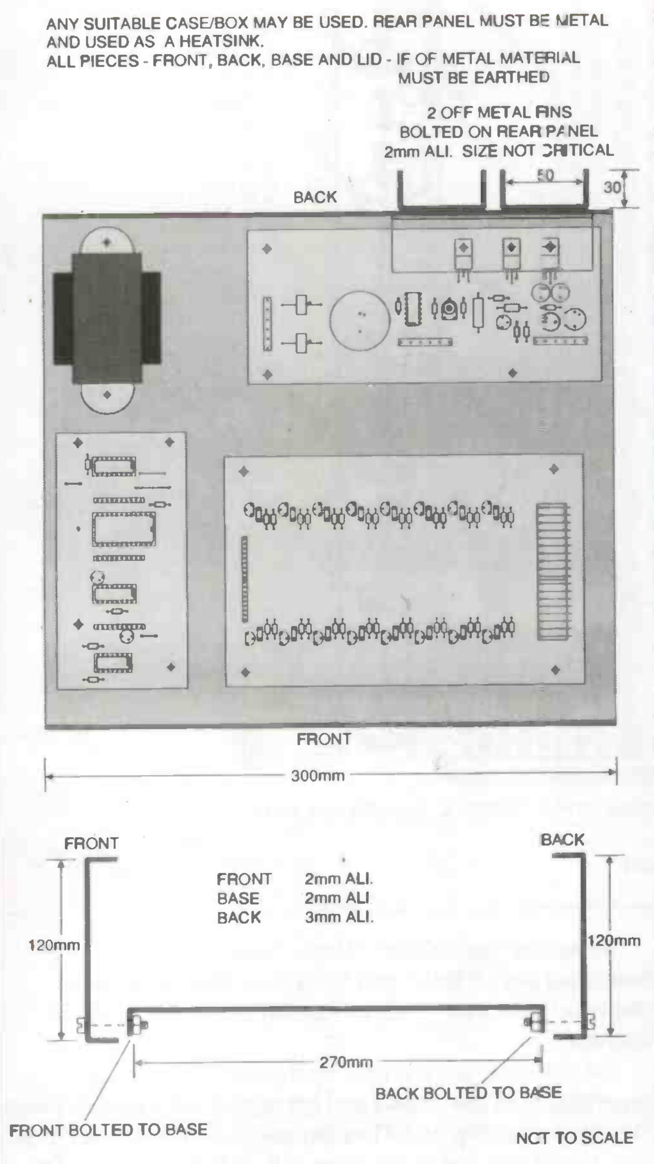
Three separate new PCBs were laid, the original having been built on vero board or strip board. The principle is that the display consists of 15 little boxes side by side in one larger box; each of the little boxes has an oblong cut out in the front and contains a Christmas tree lightbulb in its holder. The message is made up on transparent film and placed in front of these boxes and behind a diffusion screen (or nobbly bit of plastic to you and me). This has the effect of hiding the message until the light bulbs illuminate it from the rear. One of two methods can be used here - either to use a black character on a clear background or a clear letter on a blackened background.

The display scans from left to right building up the message one letter at a time. When a letter illuminates it stays on until the sequence has finished i.e. all 15 lights are on. After a delay it then blanks out and the message now builds up from the other end, one letter at a time, until all the lights are on, then after another delay the lights blank. This sequence repeats itself continuously - the speed is externally controllable. A switch option allows the direction to be frozen i.e. only build up

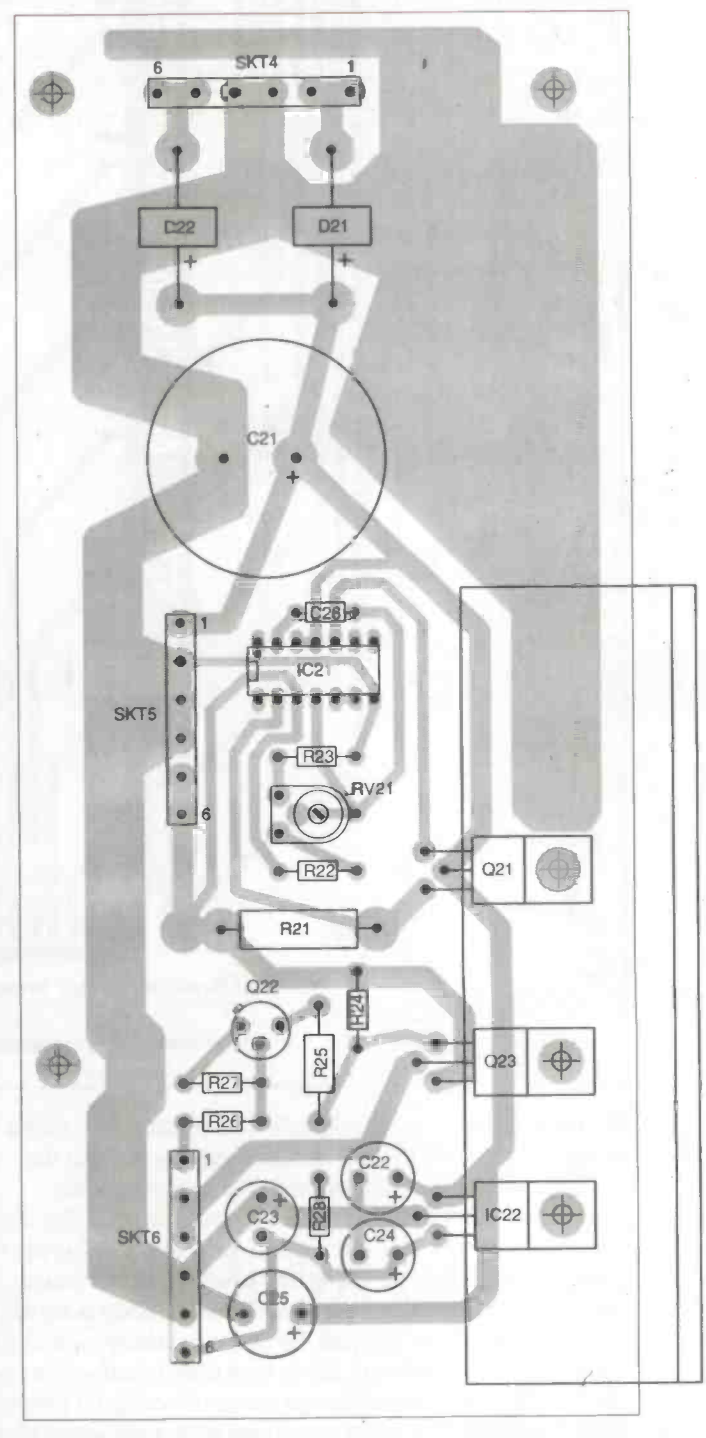


15 lamp boxes, one box divided into 15

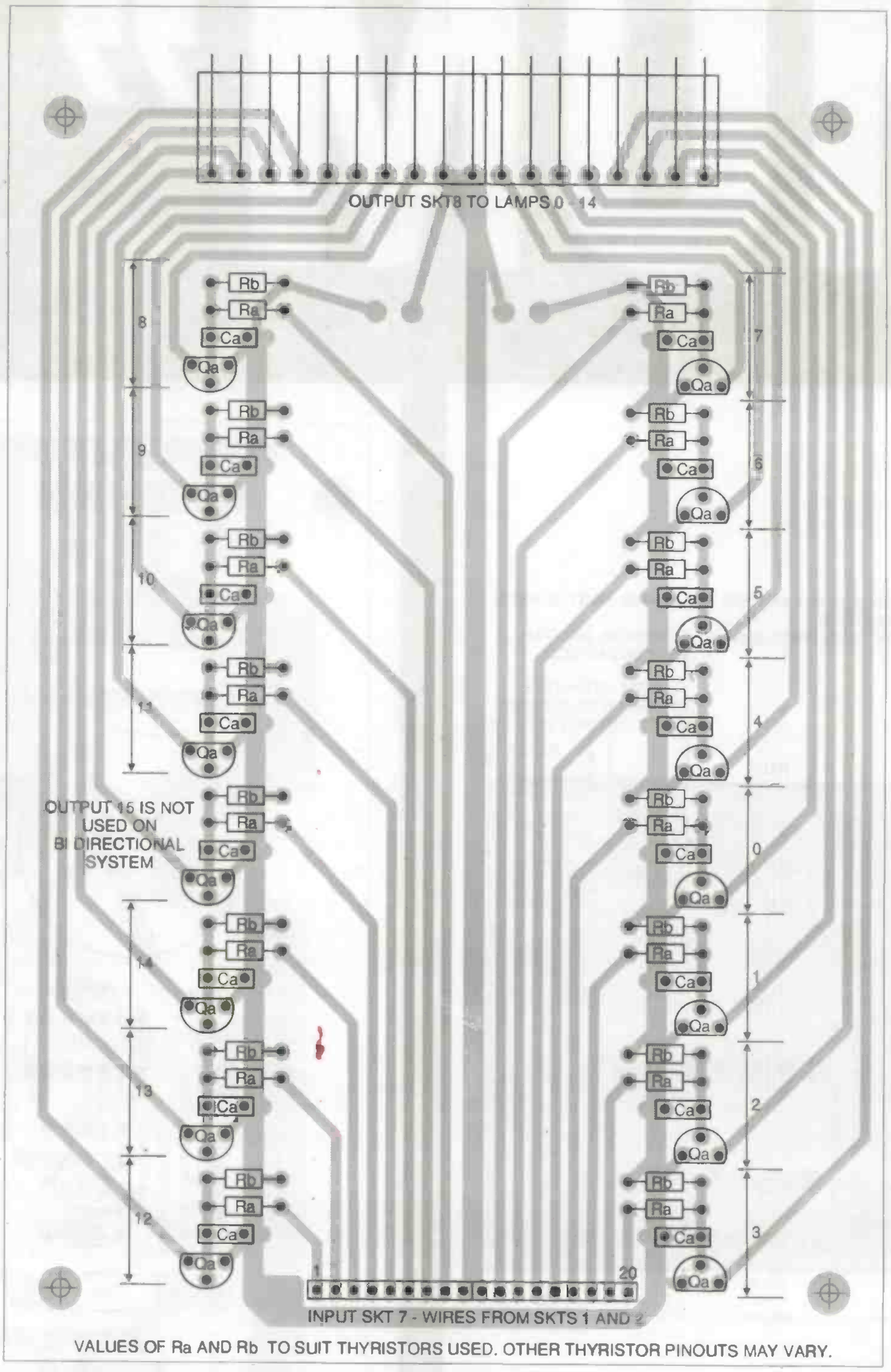
MUM"



Suggested layout and construction of control box



PSU PCB component layout



Driver board PCB component layout

from one end and, to allow instant 're-setability' after all the lamps are on rather than allowing a short time to read the message: this facility is provided with the option switch.

By using different coloured bulbs from a string of Christmas tree lights and metal foil reflectors in the little boxes behind the bulbs, an extremely interesting and effective multi-coloured display is produced very economically. The display is limited to 15 letters because of the way the direction change works; if the direction is limited from left to right then 16 letters can be used i.e. after a direction change, lamp 16 comes on before lamp 1; although this could have been sorted out electronically it would have complicated an otherwise simple, easy to follow circuit.

As well as "Hello Mum", "Happy New Year" other messages like "Toilets " and "Sale This Way" have been displayed with great effect as flashing lights always attract attention.

For domestic use it is best for the control box to be separated from the display and connected via a multiway lead. This keeps the size of the display box to a minimum and has only 12 volts on any of the wires to it so it is completely safe when little Jimmy tries to modify it with his new Duplo screwdriver set. The mains and controls are kept well clear in a metal case out of harm's way.

For work or school use the control box can be built on to the display box, care being taken not to expose any mains

SKT1	FUNCTION	TO
1	NOT USED	
2	'7' OUT	7/13
3	'6' OUT	7/14
4	'5' OUT	7/15
5	'4' OUT	7/16
6	'3' OUT	7/20
7	'1' OUT	7/18
8	'2' OUT	7/19
9	'0' OUT	7/17
10	0V	5/3

SKT2	FUNCTION	TO
1	+12 VOLTS IN	6/6
2	NOT USED	
3	'10' OUT	7/8
4	'11' OUT	7/7
5	'8' OUT	7/10
6	'9' OUT	7/9
7	'14' OUT	7/5
8	'15' OUT	7/6*
9	'12' OUT	7/3
10	'13' OUT	7/4

SKT3	FUNCTION	TO
1	+12 VOLTS OUT	3/2
2	PSU SWITCH OUT	6/1
3	SPEED POT	RV1
4	SPEED POT	RV1
5	———	
6	———	
7	0V	NOT USED
8	HOLD SET	SW1
9	0V	NOT USED
10	DIRECTION SET	SW1

SKT4	FUNCTION	TO
1	15VAC IN ⌉	15-0-15 VOLT 50VA TRANSFORMER
2	15VAC IN ⌋	
3	TRANSF. CT. ⌉	
4	TRANSF. CT. ⌋	
5	15VAC IN ⌉	
6	15VAC IN ⌋	

⌋ PCB LINKED

SKT5	FUNCTION	TO
1	21VDC UNREG OUT NOT USED	
2	0V	1/10
3	0V	
4	0V	
5	DIRECT DC OUTPUT NOT USED	
6	DIRECT DC OUTPUT NOT USED	

SKT6	FUNCTION	TO
1	PSU SWITCH IN	3/2
2	LAMP SUPPLY OUT ⌉	7/11
3	LAMP SUPPLY OUT ⌋	7/12
4	0V LAMPS ⌉	7/1
5	0V LAMPS ⌋	7/2
6	+12 VOLTS OUT	2/1

THY 0 TURNS ON BULB 0 FIRST
THY 1 TURNS ON BULB 1 SECOND
THY 2 TURNS ON BULB 2 THIRD

SKT7	FUNCTION	TO
1	0V	6/4
2	0V	6/5
3	INPUT TO THY 12	2/9
4	INPUT TO THY 13	2/10
5	INPUT TO THY 14	2/7
6	INPUT TO THY 15	2/8*
7	INPUT TO THY 11	2/4
8	INPUT TO THY 10	2/3
9	INPUT TO THY 9	2/6
10	INPUT TO THY 8	2/5
11	LAMP SUPPLY (+V)	6/2
12	LAMP SUPPLY (+V)	6/3
13	INPUT TO THY 7	1/2
14	INPUT TO THY 6	1/3
15	INPUT TO THY 5	1/4
16	INPUT TO THY 4	1/5
17	INPUT TO THY 0	1/9
18	INPUT TO THY 1	1/7
19	INPUT TO THY 2	1/8
20	INPUT TO THY 3	1/6

* ONLY USED ON 16 ONE DIRECTION SET UP

Inter-board connections

circuitry when the lid is removed to change the message. The one described in this article has separate boxes connected with a cable. To make transporting it easier, plugs and sockets were used to connect the cables between the two boxes; this is expensive but is a useful addition.

Board I: Electronic Control Board (Sequence Board)

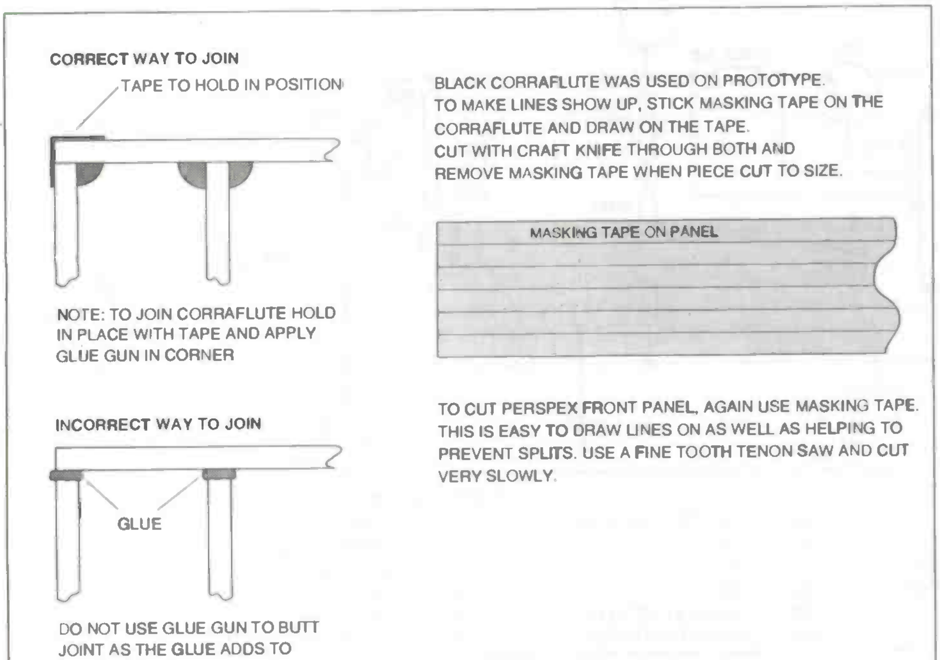
An oscillator built around IC3b and using an external potentiometer RVI 220kn as the resistive element provides the running speed of the unit (R1 is there to prevent the potentiometer from shorting out the IC at minimum resistance). Pulses from this oscillator are fed to a counter, IC1. This is a rather complex counter but several of its features are not used; the features used in this project are binary/denary.

The counter is selected to binary, a "1" on pin 9 which gives it a count of 16 volts rather than 10. The direction pin 10 is used to change the direction of the build up of lights and is driven by a stat (see later in text). The outputs A to D build up in binary from all zero's to all 1's. This is fed to a 4 to 16 line decoder, IC2. It is enabled by holding pin 23 down to 0V and not engaging the latch facility i.e.. holding pin 1 high via R3. This IC has 16 active outputs i.e. they are normally low (OV) and switch to a high one at a time giving an output according to the inputs on A-D. The outputs are from 0-15 - a little confusing but this is because zero out of the counter is decoded to an output on the decoder - this activates lamp 1, output 1, controls lamp 2 and so on.

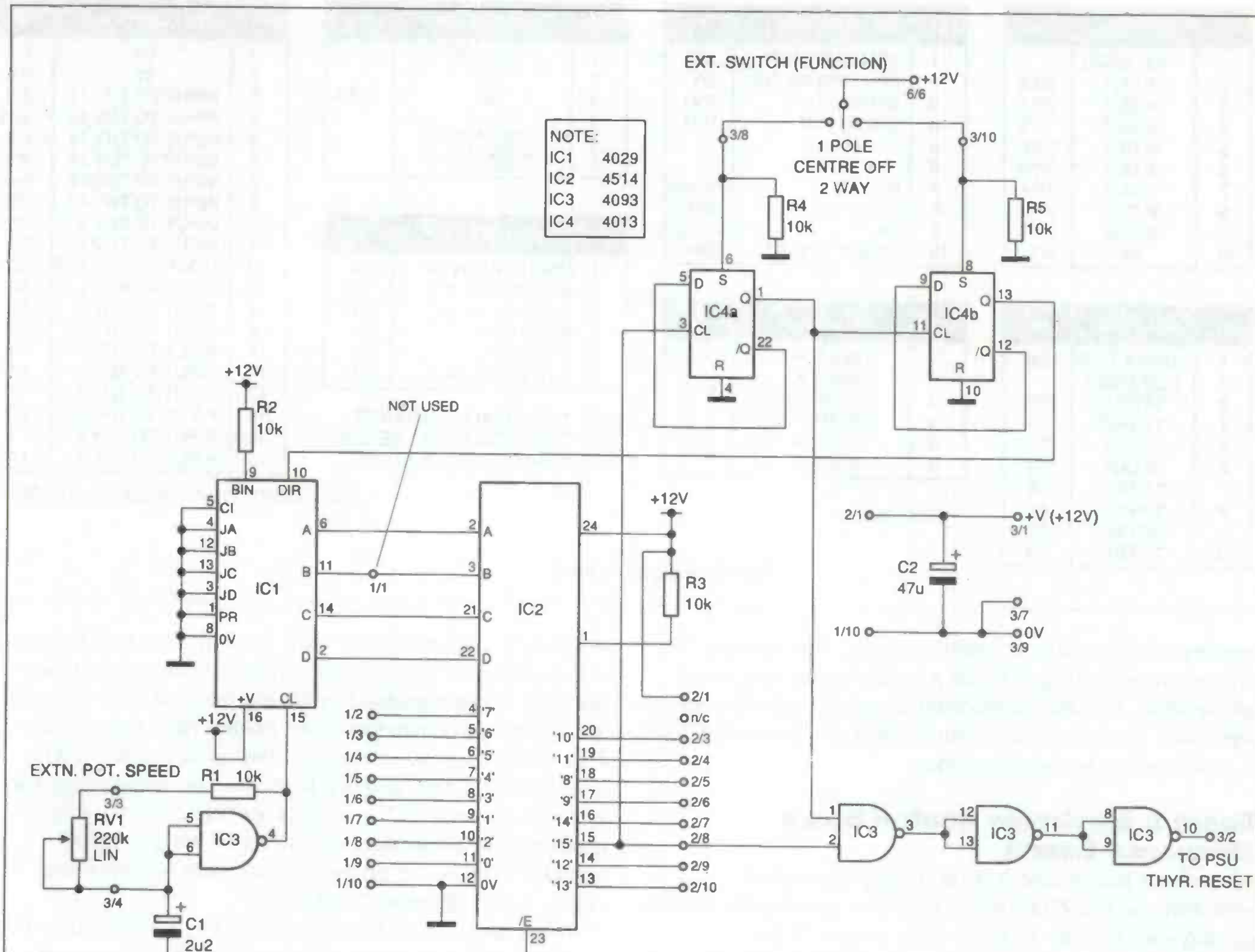
These are taken off the control board to the driver board via sockets 1 and 2. The 16th output, output 15 is taken to the clock input of a "D" type stat wired as a ÷2. The first time pin 15 - the 16th output of the decoder - goes high, it clocks the Q of the D type to the Q output remains in this state until the clock receives another pulse i.e. the decoder has gone through the complete count again this time the Q changes state again. The effect of this is to divide by two the signal present at pin 15 of the decoder.

Now the direct output from pin 15 is ANDed with the divide by two signal from the D type. This ANDing is done in IC3a because it is a NAND gate and the output only goes low when the two inputs are high. As there are two unused gates they are placed in series. This saves grounding their inputs as unused gates but has no electronic effect. This ANDed output is fed to the output switch to switch off the lamp supply on the power supply board. The other stat in IC4 is used as another divide by 2 using the already divided by 2 signal from the decoder, pin 15. The output of this is fed back to the direction pin on the counter, making a total ÷ 4.

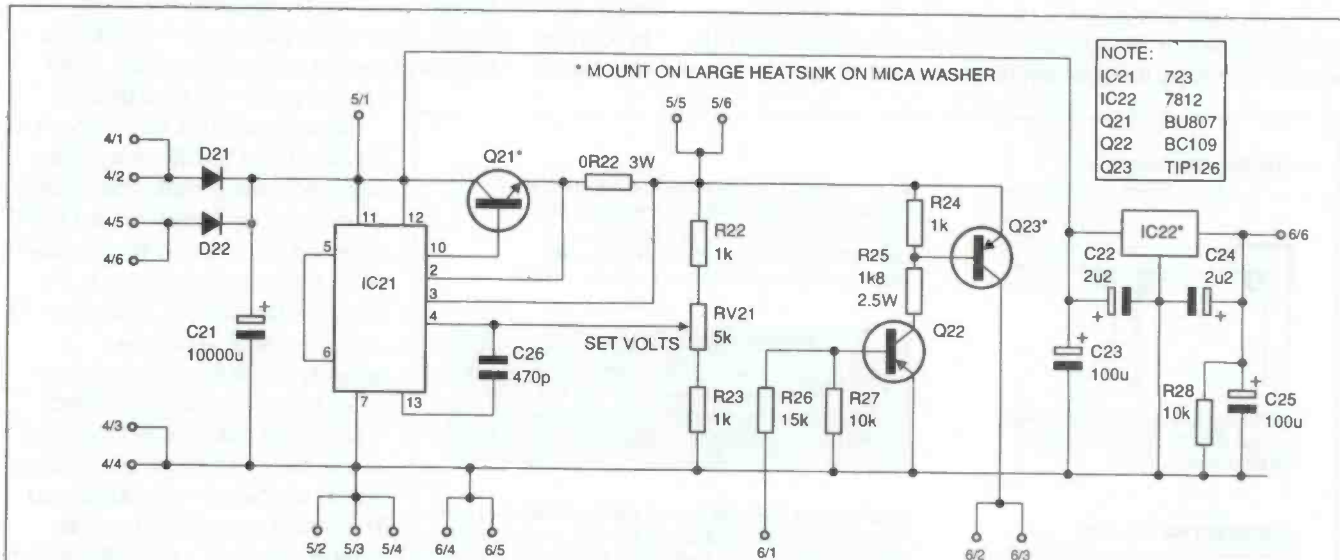
The combined effect is to illuminate the bulbs in turn until they are all on. They stay on for another complete scan 0-15 when they are



Construction details



Electronic control circuit

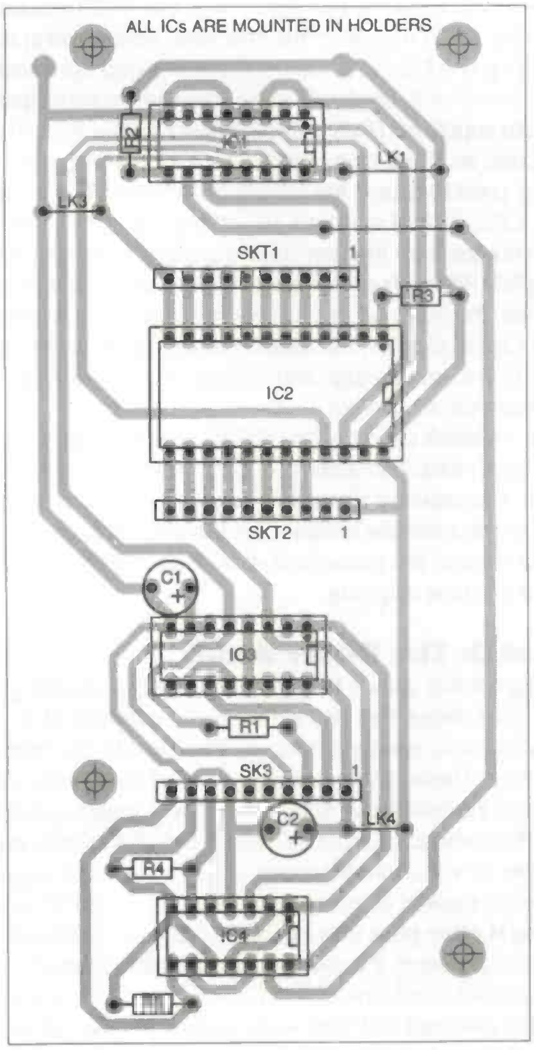


SKT PINOUTS:

4/1	AC IN - 12V	5/1	+VOLTS UNREG (15V)	6/1	1 VOLT SWITCH INPUT
4/2	AC IN - 12V	5/2	0V	6/2	OUTPUT VIA SWITCH
4/3	AC CENTRE TAP	5/3	0V	6/3	OUTPUT VIA SWITCH
4/4	AC CENTRE TAP	5/4	0V	6/4	0V
4/5	AC IN - 12V	5/5	+VOLTS OUT REG	6/5	0V
4/6	AC IN - 12V	5/6	+VOLTS OUT REG	6/6	12 VOLTS REG OUTPUT

[LINKED ON PCB

Power supply and switch circuit



Control PCB component layout

all reset. Now the bulbs illuminate from the other direction and stay on for a complete scan before resetting and starting all over again, making a total of 64 pulses from the oscillator for a complete sequence.

Both of the D type stats have their sets brought out to a switch. If the set of a stat is taken high, the Q output goes high and stays high, overriding the clock and D signals. If IC4a set is taken high, there is no holding the light on for one scan and the direction freezes. If IC4b is taken high, the hold still works but the direction is always the same.

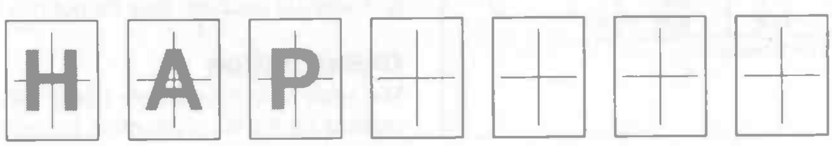
Although all the inputs and outputs to this board can be connected directly with wires, the use of plugs and sockets aids fault-finding and any possible modification or repair is highly recommended as the board can be completely disconnected and worked on without the restriction of wires connected to other boards etc.

Board 2: The Power Supply Board

Firstly the bulbs used set the voltage required. The ones used in our model were a cheap string of 20 lights sold as Christmas tree decorations. They consisted of 20 lights so, with a 240 supply, the voltage of each bulb must be 240 over 20 = I2V: to produce long life in the bulbs they must be run slightly under this voltage.

Because the supply is going to be subjected to very light loads i.e. one bulb on, to heavy loads i.e. all 15 bulbs on, a regulated supply is necessary to maintain this voltage. Each bulb, depending upon type, may require up to 150MA - a total of around 2.25 amps may then be required, although not continuously. The regulator must be variable to allow the voltage to the bulbs to be adjusted to just below its rated voltage i.e. a 12 volt bulb should be run at around 11.5 volts across the bulb. Allowing for voltage drops in the switching darlington the supply is set to around 12 volts.

To produce this supply, a transformer of 15-0-15 Is required at, at least, 50Va. As we have already seen, the maximum load Is not continuous and is averaged out when only one or two bulbs are on. From here, the 15-0-15 is passed on to the PSU PCB. As only three connections of the 6-pin socket are required they are doubled up to reduce the current on each pin. From here, two large diodes, D21and D33, convert the AC into DC which Is smoothed by C21. A 723 regulator is used to stabilise the supply; this is a standard regulator circuit. As well as regulated, the supply is variable by adjusting VR2I and current is limited by use of R21. This will reduce the output voltage to maintain a maximum of about 0.65 volts across it,. This then acts as a current limit of around 3 amps. A power darlington Is used as the main series pass transistor - BV 807, TR21. These are both rugged and reliable and easier to mount than a T03 style of transistor (note that its metal heatsink is connected to its collector and so must be mounted on the heatsink with an insulating set). To allow the PCB to be used as a standard supply in other projects, the

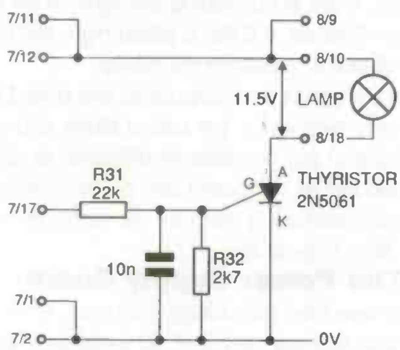


USE FRONT PANEL WITH THE 15 WINDOWS AS THE MASTER TO MAKE A TEMPLATE. THIS IS THEN USED BEHIND THE FILM WHEN MAKING THE MESSAGE TO ENSURE THE LETTERS ARE IN THE MIDDLE OF THE WINDOWS.

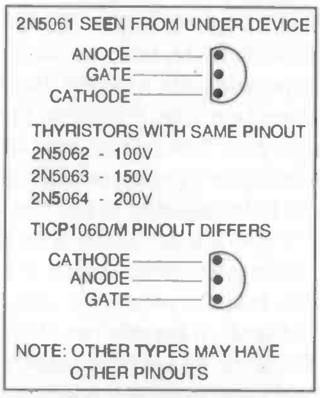
THE MESSAGE IS BUILT UP USING A STENCIL SO ALL THE LETTERS ARE THE SAME SIZE AND STYLE. GAPS BETWEEN WORDS CAN BE LEFT BLANK OR THE BULB BEHIND IT CAN BE REMOVED.

THE FILM USED IS DRAFTING FILM CUT TO SIZE OR PCB MASTER FILM OR SIMILAR. SEVERAL MESSAGES CAN BE MADE TO SUIT TIME OF YEAR

The message film



THYRISTOR SWITCH CIRCUIT
15 OR 16 REQUIRED - SEE TEXT
15 ON BI DIRECTIOINAL SYSTEM
16 ON SINGLE DIRECTION



LAMP NO.	INPUT SKT.	OUTPUT SKT.
0	7/17	8/18
1	7/18	8/17
2	7/19	8/16
3	7/20	8/15
4	7/16	8/14
5	7/15	8/13
6	7/14	8/12
7	7/13	8/11
8	7/10	8/8
9	7/9	8/7
10	7/8	8/6
11	7/7	8/5
12	7/3	8/4
13	7/4	8/3
14	7/5	8/2
15*	7/6	8/1

* NOT REQUIRED ON 15 LAMP SET UP

SUPPLY	INPUT SKT	OUTPUT SKT.
0V	7/1	—
0V	7/2	—
#+11.5V	7/11	8/9
#+11.5V	7/12	8/10

ACROSS BULB

Thyristor switches

variable DC output is provided in socket 5, pins 5 and 6 with 0V on pins 2, 3 and 4 (note that this output is not required in this project). The unregulated supply across the reservoir capacitor is also taken out. This can be used to ensure there is at least 4 volts difference between this and the output voltage to maintain stability on heavy loads (not used in this project).

Where this supply differs from most is the inclusion of TR22 and TR23; these form a power switch which, normally, is on i.e. the output in the collector of TR23 is around 11.5 volts. This is due to TR22 being turned on by a signal from the control board. This in turn drives TR23 but, when TR22 is turned off, TR23 is also turned off reducing the output to zero, necessary to reset the thyristors. Note that TR23 requires an insulating set to mount on the heat sink. When testing and setting up this board by itself, a link is required from socket 6, pin 1 to socket 5, pin 5 otherwise there will be no output to the bulb supply as TR22 and TR23 will both be turned off, i.e. it requires an active high signal to turn it on. A separate 12 volts is used to supply the control board logic. This is catered for in IC22, a 7812 standard 12 volt regulator, the input of which comes from the unregulated supply across the reservoir capacitor. Although all the power devices are mounted on a heatsink, the 7812 will not get hot because of its minimal load in the CMOS logic on the control board. Again, all the inputs and outputs are via plugs and sockets to allow testing, modifications, repairs etc.

The heatsink used on the PCB is a piece of aluminium angle 3mm thick and 35mm x 35mm x 140mm long, size not critical. It is essential that even this is bolted to a metal back panel to minimise the temperature increase while in use. A couple of extra fins bolted onto the exterior of the back panel will aid this (see diagram).

Board 3: The Driver Board

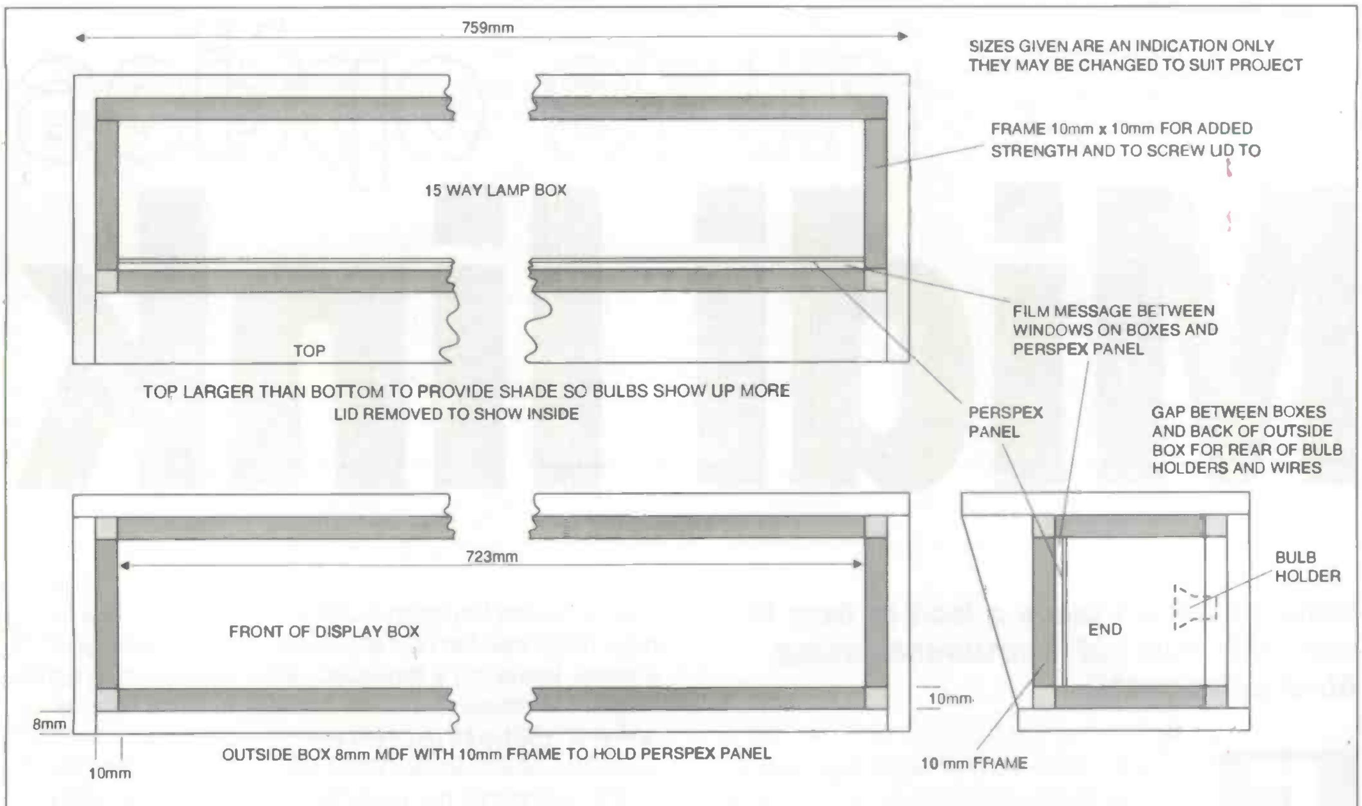
Although at first glance this board looks a little daunting, a closer look shows that the same circuit consisting of 4 components is repeated 15 times. The drive to the lamps is via a thyristor. These devices, once turned on by a positive drive signal on the gate input, stay on until the supply is removed. R31, R32 produce a potential divider to limit the drive current from the 12 volts from the control board. The small capacitor C31 helps prevent false triggering. The values of R31 and R32 and the I4 other pairs may need to be altered if different thyristors are used. It is recommended that if different thyristors are used, one of them is wired and the values of R31 and R32 checked that they work before the other 14 are wired. If a less sensitive type is used R32 may be removed altogether and if an even less sensitive type is used, the value of R31 may be reduced accordingly. Do not reduce below 8K2 as this may overload IC2. Once one of the thyristors is working reliably, the others can be wired using similar values if different from the ones shown in the drawing.

Again, all the inputs and outputs are taken in and out on sockets to aid fault-finding and modifications or repairs. The input socket is a 20-way 0.1 type while the output is an 18-way 0.156. This prevents them from being swapped during construction.

Great care must be taken when connecting these plugs and sockets because they do not run in sequence.

Construction

The small lamp boxes were made out of corriflute, a fantastic material similar in construction to corrugated cardboard but made out of plastic. It is reasonably cheap and is very easy to work with. It can be joined either by tape or glue from a glue gun. The 15 boxes each contain a coloured bulb mounted in its holder at the rear of the box and a foil reflector behind (take care the foil does not short out the bulb). The holders were cut from the Christmas tree string of lights with about 6" (about 1500mm) of cable. These were then joined to longer wires to connect to the driver board. Care must be taken not to allow these to short. Heatshrink sleeving is useful here to protect the joints. These corriflute boxes, each the same size, allows the message to be made up from a template. I used the front



Main display box

panel of the boxes i.e. one piece with 15 oblong holes cut in it as the template. The oblong cutouts were traced onto a piece of card. This was then placed under the transparent film when making the message, thus ensuring that the letters on the film are in the centre of the cutout of each box. Great care must be taken to ensure it is the right way round. Common sense should prevail here.

The boxes containing the bulbs are placed in a larger box leaving room to slide the message film between the boxes and the plastic front panel. The outer box is made from MDF board 3/8" (10mm) thick. The size is determined by the size of the 15 boxes and if they are each only 1/4" different to the sizes shown, the outer box must be nearly 4" different to accommodate this - so each one must be measured and built to the size required rather than working to a drawing. To enhance its appearance, the MDF box is covered in sticky-back wood effect plastic - yes, I used to watch Blue Peter.

The control box, if built separately, should be big enough to lay each of the three boards flat with the mains transformer mounted far enough away from them so as not to risk shorting the terminals that normally protrude. All mains connections should be shrouded and, if in any doubt, a qualified engineer's help should be sought.

The box must have a metal rear panel and the heatsink should be bolted to it. Extra fins should be added on the outside to help dissipate the heat (see diagram). Nearly all the heat is produced by the series pass regulator transistor TR21. This is because TR21 has voltage across it as well as current through it, whilst the switching transistor TR23 is either on or off and therefore does not produce so much heat.

In the prototype (yes, I actually made one), the front panel, rear panel and base were three separate pieces bolted together which enabled the project to be laid out and gave better accessibility while constructing and working on the project. Doing it this way means that more people can work on it at the same time.

Again, the actual size is not critical but safety must always be at the top of the list when deciding - if in doubt, go for a larger rather than smaller box.

The whole project uses reasonably priced components. The cable connecting the two boxes could have a plug and socket at each end but this increases the overall cost. A cheaper method is to have a long lead on the display box and mount the 18-way housing for plug 8 on this lead. It is slightly less convenient but much cheaper and still retains the boxes as completely separate units, making moving them around much easier.

PARTS LIST

Off PCB pieces

- 15-0-15 volt 50Va transformer (Maplin DH 315)
- 1 pole 2 way centre off switch (toggle)
- 2 pole on/off mains switch
  220kn linear pot
- 1 piece of aluminium angle approx
  30 x 30 x 160mm long x 3mm thick
- To make 10 and 20 way sockets
  2 strips of pins (Maplin JWS9P)
- 5 x housings to suit 10 way (Maplin PY94C)
- Terminal pins, 5 packs of 10 (Maplin YW2SC)
- 3 x 6 way plugs (Maplin JZ01B)
- Pins for above (Maplin JY9 1 Y)

Fibre optics Midi link

Robert Penfold takes a look at how to link MIDI musical instruments using fibre-optic cable

Fibre-optic cables are now beginning to play a major role in the world of telecommunications, but they have had less impact in the home and office. Optical cables are used to some extent in computer network systems and digital audio gear, but they remain relatively rare outside these niche markets. In truth, fibre-optic cables have no real advantages in many day- to-day applications and would simply provide lower performance at higher cost. However, there are some areas where they have one or two potential advantages, and for the home-constructor fibre-optics have a strong novelty value. Although fibre-optic communications tends to be associated with expensive hi-tech systems, surprisingly simple and inexpensive circuits can be used in applications that involve the interchange of data at relatively slow rates.

The system described in this article is for use with MIDI (musical instruments digital interface) systems. MIDI is basically just an ordinary asynchronous serial communications system, much like a standard RS232C computer interface. It uses a 5 milliamp current loop system and an opto-isolator at each input, rather than the RS232C positive and negative signal voltages. Interfacing a fibre-optic system to this type of signal presents no difficulties though. The MIDI baud rate is quite high at 31250 baud, but this baud rate is well within the capabilities of inexpensive fibre-optic LEDs and photocells.

This system has two potential advantages over ordinary wire cables. One of these is simply that this system will work reliably at up to at least 30 metres, which is twice the maximum cable length that should be used with normal MIDI interfacing.

The second advantage is that an optical cable radiates no electro-magnetic interference. This can be an important factor in situations where there are long cables, some carrying digital signals and the others carrying audio signals. Digital noise is very good at finding its way into audio signal paths. Even with a fibre-optic cable running alongside a sensitive microphone cable there is no possibility of any cross-coupling.

How It Works

With equipment of this type it is possible to use a modem style f.s.k. (frequency shift keying) technique, or a very simple d.c. system. With modern optical cables and photocells it is only necessary to resort to an f.s.k. system if a very long operating

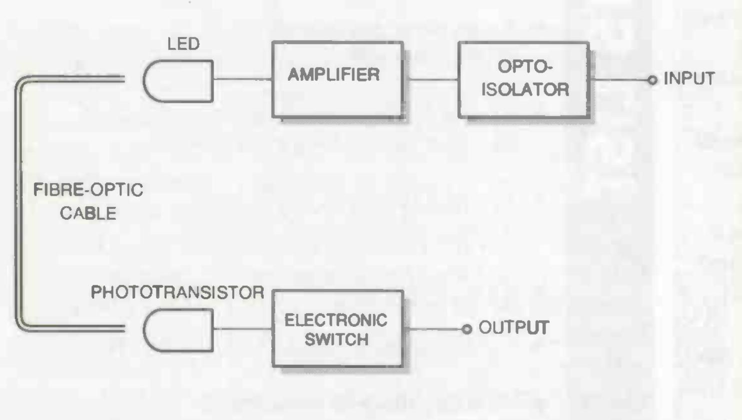


Fig.1. The MIDI fibre-optic link block diagram

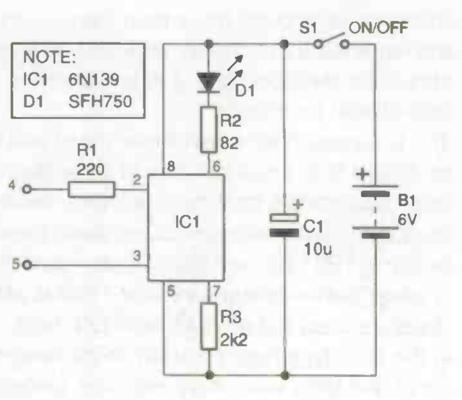


Fig.2. The fibre-optic transmitter circuit

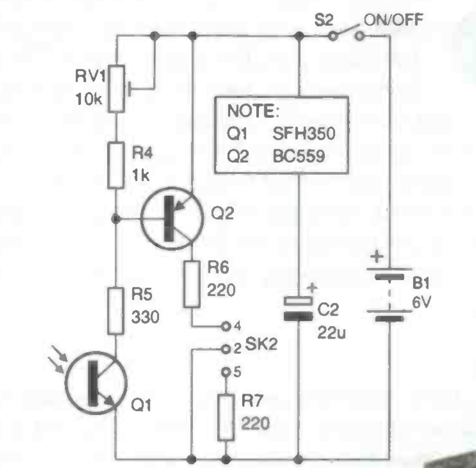
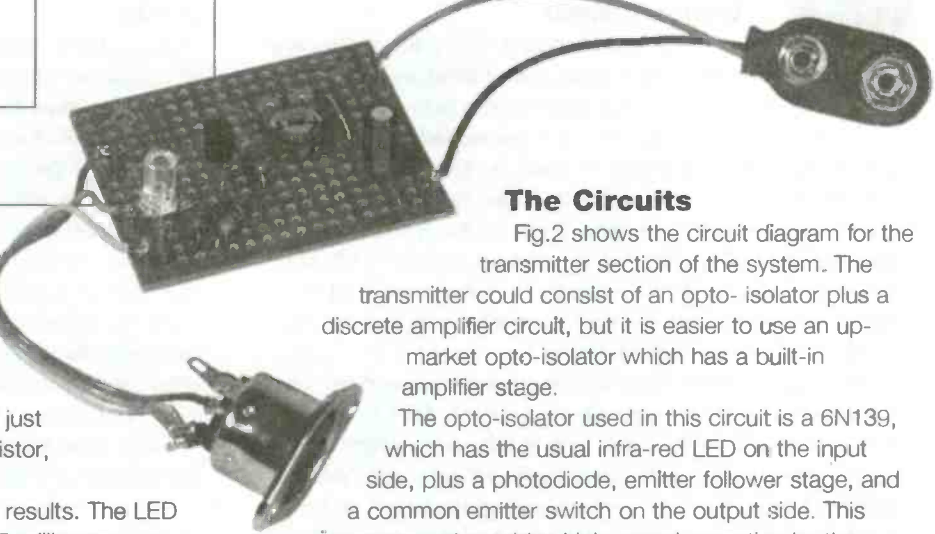


Fig.3. The fibre-optic receiver circuit



range is required, which is highly unlikely in a MIDI application. A simple d.c. link is therefore used in this case.

Fig.1 shows the block diagram for the MIDI fibre-optic link. The transmitter could just consist of a LED and a current limiting resistor, but the limited drive current from a MIDI output would probably not give very good results. The LED would only have an "on" current of about 5 milliamps, whereas a current about ten times higher than this is needed in order to guarantee good performance. The LED at the transmitter end of the system is therefore driven via an amplifier which provides a suitably high drive current.

The amplifier is driven with the MIDI input signal via an opto-isolator stage. The latter is to a large extent superfluous in this case, since the fibre-optic link is effectively a sort of giant opto-isolator! However, the MIDI specification stipulates that all items of MIDI equipment should have an opto-isolator input stage, and MIDI outputs are designed to drive the LED in an opto-isolator. It is advisable to use an opto-isolator input stage, if for no other reason than ensure good compatibility with the MIDI output which drives the transmitter circuit.

At the receiving end of the system there is just a phototransistor driving an electronic switch. When the LED at the transmitter is switched on, the phototransistor has a relatively high leakage current, and this is used to turn on the electronic switch. When the LED at the transmitter is switched off, the phototransistor has only a minute leakage level, and the electronic switch is turned off. The switch controls the LED in the opto-isolator at the MIDI input of the unit driven from the receiver circuit, and this LED switches on and off in sympathy with the one at the transmitter.

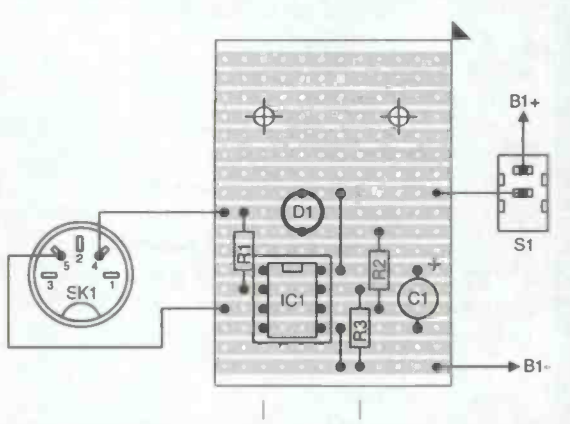


Fig.4. The transmittercircuit layout and wiring

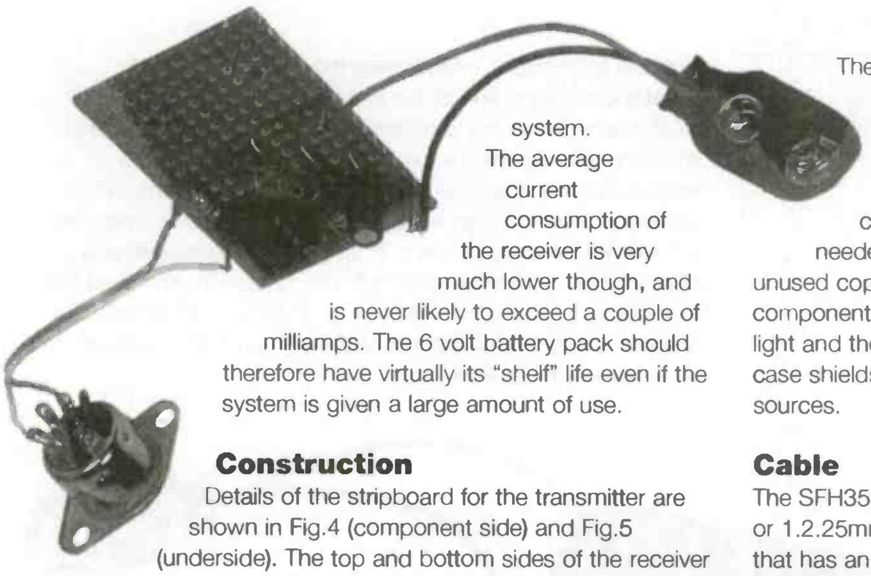
The Circuits

Fig.2 shows the circuit diagram for the transmitter section of the system. The transmitter could consist of an opto- isolator plus a discrete amplifier circuit, but it is easier to use an up-market opto-isolator which has a built-in amplifier stage.

The opto-isolator used in this circuit is a 6N139, which has the usual infra-red LED on the input side, plus a photodiode, emitter follower stage, and a common emitter switch on the output side. This arrangement provides high-speed operation by the standards of opto-isolators, and the 6N139 can actually handle baud rates of up to about ten times the MIDI baud rate. It also provides good efficiency, with an output current at around ten times the 5 milliamp input current. R2 is the current limiter for D1 which is a special fibre-optic LED. R1 provides part of the current limiting for the LED at the input of IC1. The remaining current limiting is provided by resistors in the MIDI output circuit that drives SK1. R3 is the load resistor for IC1's emitter follower stage. The current consumption of the circuit depends on the amount of MIDI activity the unit handles. Under standby conditions the current consumption is negligible, but it rises to a maximum average level of about 20 milliamps with a continuous stream of data. In normal use, the average consumption is unlikely to be more than a few milliamps, and the four HP7 size batteries have an extremely long operating life.

The receiver circuit appears in Fig.3. TR1 is the phototransistor, and this is a special fibre-optic type. Its base terminal is externally accessible, but in this circuit it is left unused. TR1 drives a p.n.p. common emitter switching transistor (TR2), which in turn drives the output socket via current limiting resistors R6 and R7. When TR1 is in darkness, its leakage level is negligible and TR2 is turned off. When TR2 receives a pulse of light from the optical cable, its leakage level increases to a few hundred microamps, and TR2 is biased hard into conduction. This arrangement is similar to the one used in the 6N139 opto-isolator, and it provides a switching speed that is more than adequate for the present application. VR1 controls the sensitivity of the circuit, and it is adjusted to give an output signal that accurately retains the mark-space ratio of the input signal.

Like the transmitter circuit, the current consumption of the receiver depends on the density of the data fed through the



system. The average current consumption of the receiver is very much lower though, and is never likely to exceed a couple of milliamps. The 6 volt battery pack should therefore have virtually its "shelf" life even if the system is given a large amount of use.

Construction

Details of the stripboard for the transmitter are shown in Fig.4 (component side) and Fig.5 (underside). The top and bottom sides of the receiver board appear in Figs. 6 and 7 respectively. The transmitter board measures 12 holes by 17 copper strips. Cut out a board of this size using a hacksaw, and then file any rough edges to smooth finish. Then drill the two 3.3 millimetre diameter mounting holes, which will take 6BA or metric M3 mounting bolts. Next, make the four breaks in the copper strips, using either the special tool or a hand-held twist drill of about 5 millimetres in diameter.

The board is then ready for the components and the two link-wires to be fitted. The cathode (+) lead of D1 is the shorter lead. Use single-sided solder pins at the points where connections to off-board components will be made. Virtually any small plastic case will accommodate the board and other components. Fit the component board on the rear panel of the case, and SK1 and S1 on the front panel. A 2.3 millimetre diameter hole for the fibre-optic cable is drilled in the front panel directly in front of D1. The cable is a push-fit into D1 (and TR1 in the receiver) and it may tend to pull out of place unless it is clamped or glued in place. If it is clamped in place, be careful not to clamp it too hard, or the inner filament might be damaged.

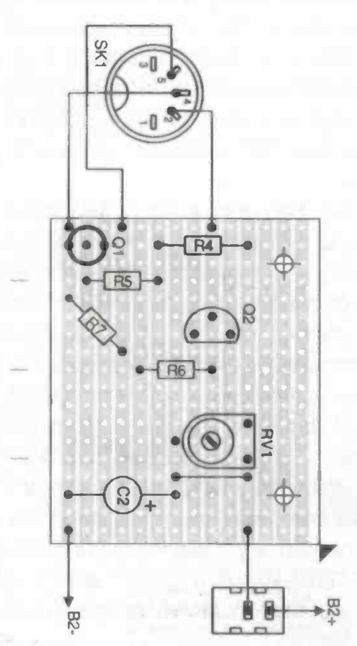


Fig.5. The receiver board component layout and wiring

The stripboard for the receiver measures 18 holes by 14 strips. Construction of the receiver follows along much the same lines as construction of the transmitter. The shorter lead of TR1 is the emitter leadout, and the middle one is the collector leadout. The base lead of TR1 is not needed in this circuit, but it is connected to an otherwise unused copper strip in order to secure TR1 firmly to the component panel. Note that TR1 is sensitive to extraneous light and the receiver might not function properly unless the case shields TR1 reasonably well from any strong light sources.

Cable

The SFH350 and SFH750 are designed for use with 1/2.2mm or 1.2.25mm fibre-optic cable. In other words, with a cable that has an inner filament which is one millimetre in diameter, and an overall diameter of 2.2 or 2.25 millimetres. As supplied, the cable will probably have rather roughly cut ends which will not provide an efficient coupling. It is easy to cut the ends properly, and it requires nothing more than a sharp modelling knife and a cutting board. Simply trim a few millimetres from each end of the cable, using plenty of pressure so that each cut is completed in a single stroke. Make the cut reasonably perpendicular to the cable. Also, make sure that the cable is fully pushed home into TR1 and D1, or only a weak coupling will be obtained.

Even using an old fibre-optic cable that was not particularly efficient with the red light from the SFH750 I found that an

PARTS LIST

Resistors (0.25 watt 5% carbon film)

- 2 82R
- R3 2k2
- R4 1k
- R5 330R

Potentiometer

- VR1 10k min hor preset

Capacitors

- C1 10u 25V radial elect
- C2 22u 16V radial elect

Semiconductors

- IC1 6N139 opto-isolator
- TR1 SFH350
- TR2 BC559
- D1 SFH750

Miscellaneous

- B1,2 6 volt (4 x HP7 size cells in holder - ● 2 off)
- SK1,2 5 way 180 degree DIN (2 off)
- S1,2 p.s.t. min toggle (2 off)
- 0.1 inch stripboards having 12 holes by 17 strips, and 18 holes
- by 14 strips, battery connector (2 off), 8-pin DIL IC holder,
- 1/2.2mm fibre-optic cable, wire, solder, etc.

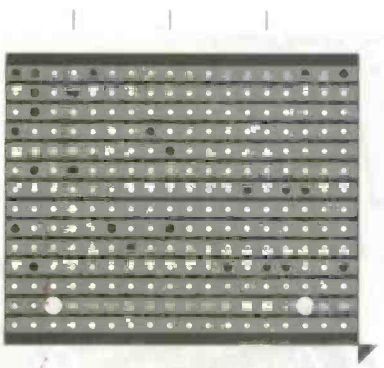


Fig.6. The underside of the transmitter board

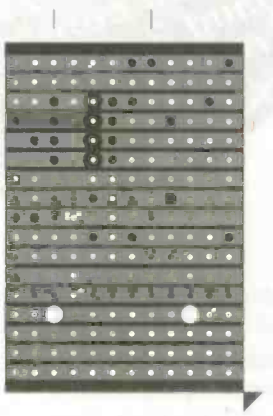
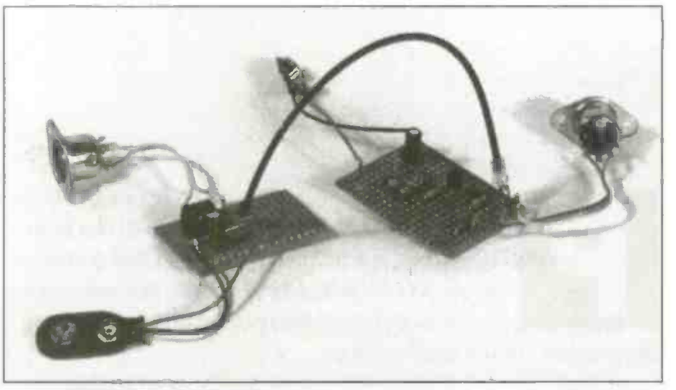


Fig.7. Underside of the receiver board

wavelength of the SFH750, and a much greater operating range should be possible using a cable of this type. I have not tested the system over a range of more than 30 metres though. Do not take fibre-optic cables through tight bends as this can seriously damage the inner filament. Most fibre-optic cable have a minimum bend radius of about 15 to 25 millimetres.

In Use

SK1 on the transmitter connects to the MIDI OUT socket of the master unit via a short MIDI cable. SK2 on the receiver connects to the MIDI IN socket of the slave device via another short MIDI cable. A suitable setting for VR1 can be found by trial and error. Any setting that gives reliable operation is suitable. Provided a strong signal is reaching the receiver, there will be a wide range of settings that give reliable results. Simply set VR1 at roughly the middle of this range.



Stamp Duty

Roland Oliver looks at using the parallax basic stamp and your pc's RS232 serial port 2400 baud 3 wire serial relay controller

PICs - or, by their real name, Peripheral Interface Controllers - have been around for some time now. They are present in many of the common-or-garden mice that we use with our ever more powerful personal computers and in many embedded control applications.

In their 16/17/1 8cxx clothing they are programmable in machine code via various development tools which also vary in price from the sublime to the ridiculous, depending what you want to pay. C compilers are available for the so inclined making things easier.

In this article we are going to use one of the newer derivations of the PIC, the Basic Stamp. Manufactured by Parallax Inc, Rocklin, CA, USA, the Basic Stamp is produced in three forms - the original Basic Stamp with the processor, memory and voltage regulator mounted discreetly on a pcb,

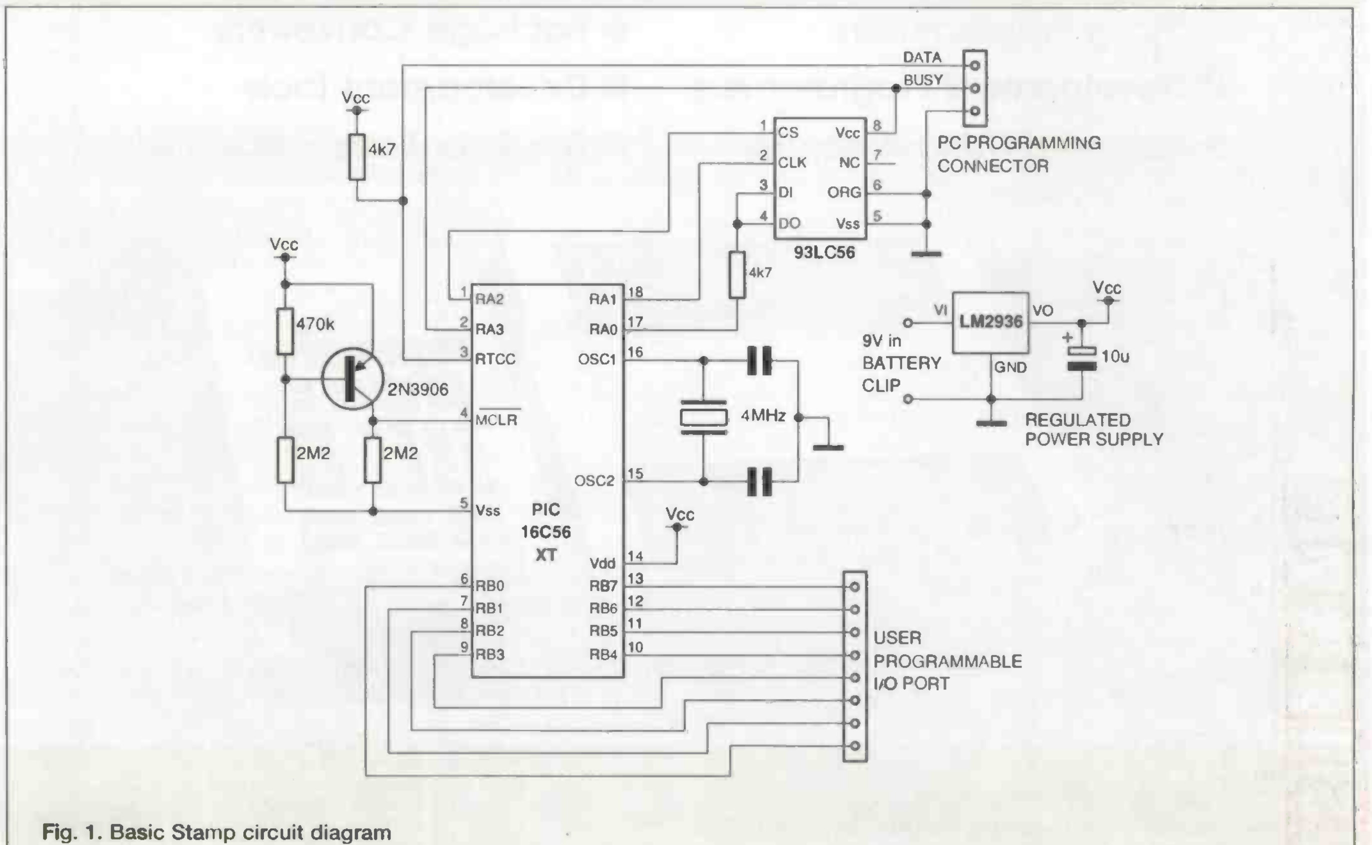


Fig. 1. Basic Stamp circuit diagram

complete with PP9 battery clips, 14 pin sip and prototyping area. The second style of Basic Stamp is hybridised on its own pcb and is terminated with a 14 pin male sip, which is soldered into the stamp motherboard . The motherboard also has a prototyping area (larger than the original Stamp), 14 pin sip and battery clips, part number for this Basic Stamp is BS1-IC.

The third type of Basic Stamp is the 24 pin dil version - its part number is BS2-1C.

The Parallax Basic Stamp is about as basic as you are going to get with any processor. Programming is carried out in Parallax PBASIC- an interpreter developed by Parallax to run on a PIG 16C56 Micro controller.

PBASIC is available complete with a parallel interface cable, extensive notes, programming manuals, built in Editor and free Technical support ready to run on any IBM compatible PC. Note that this development kit does not include the Basic Stamp itself. These must be purchased separately. The price of the development kit is roughly £99 In the UK and $99 from Parallax in the US . The Basic Stamps themselves run at £29 in the UK and $39 in the US.

Basic Stamp topology

Referring to photo of the Basic Stamp we see it is mounted on a roughly 4 x 6.5 cm board with header (SIP) pins with which to connect the 1/0 and two battery clips to connect the ppt 9 volt battery. A further three pins are provided to connect to the parallel cable which in turn connects to LPT 1 on the PC (arrows on the cable connector and the board make sure you are connected the right way round).

Figure 1 shows the circuit diagram for the Basic Stamp. Communications with the PC are carried out via the PC programming connector connected to pin 3 (RTCC) of the Stamp with handshaking on pin 2 (RA3). Power up reset is applied to pin 4 (MCLR) resetting the stamp to the beginning of the programme in memory. A 4MHz resonator is connected to pins 15 and 16 ,OSC 1-OSC 2 to serve as a clock.

Connected to pins 1 - 17 - 18 is a 93IC56 serial EEPROM of 256 Bytes capacity pin 17 (RA0) being the data 1/0 (bi directional), pin 18 (RA1) is the clock to sync the serial data line to the EEPROM Pin 1 (RA2) is the chip select for the EEPROM and is held high during read/write cycles, pin 8 (Vcc) of the EEPROM is also held high via pin 3 (RA3) of the Micro Controller during read writes to supply power to the EEPROM.

The 1/0 for the Basic Stamp consists of pins 6 through to pin 13 (RB0 - RB7) or 8 pins in total. All eight pins are pin programmable as either input or outputs and can sink a maximum of 25 ma or source 20 mA when used as outputs. However, parallax recommends a maximum chip total of 50 ma when sinking and 40 mA when sourcing.

All the above refers to the original BASIC stamp which we are using in this project. The 051 is the hybridised version of the original and contains all its features, including the Voltage regulator, but will fit in a much smaller PCB footprint thanks to its sip construction.

The Basic Stamp Type BS2-IC as mentioned above is a 24 pin DIL device, and has all the characteristics of the other two Basic Stamps plus the advantage of 16 1/0 pins.

As I have not had the chance to review this latest addition to the Parallax range I cannot comment on it in detail but looking at its specifications - 2048 bytes of programme space, enough for 600 lines of code, 20 MHz clock and serial port baud rates up to 9600, it sure gladdens the heart of a old hardware Techi ,(it even has a real time clock interface).

PARTS LIST

Power supply parts

- BR1 GI WO4-9027 full wave bridge rectifier
- C1 1000 micro farad 16 VDC electrolytic capacitor
- C2 0.1 microfarad mylar or equivalent capacitor.
- voltage regulator LM 7805 +5VDC regulator

PARTS LIST

Basic stamp based RS232 relay driver

- IC1 75492 relay driver
- D1-d4 1N4001
- LED1—4 general purpose LEDs
- R1—4 870 ohm quarter watt resistors
- R5 22 K quarter watt resistor
- REL 1—4 rs 349-648
- DB 9 male connector (pcb right angled mount type)
- 14 pin stripline 0.1 inch type
- 2 female 10 pin connectors (stripline) available from RS
- length of ribbon cable 10 core (to suit), see text.

Basic Stamp `basic's & hardware'

I grew up in a teletype world where you loaded the bootstrap loader from a paper tape, fitted to the teletype into a DEC pdp 8 with 4k of memory and you thought you were in heaven with such a powerful machine. However, times have changed and `booting up' means nothing more than switching on the PC and waiting for the `C:' prompt or Bill Gates' Windows Programme Manager. How boring.

Oh for the days when the exhilaration of debugging a programme coupled with "Am I reading this hardware right?" and then finally it comes good.

Stamp Basic is like that. You get hooked on its capabilities, and try to get more and more out of it.

Looking at the Basic Stamp you first think its too basic to do anything but very rudimentary stuff like switching on a LED etc, but on further observation you find it contains a lot more than expected.

Instructions such as SERIN, SEROUT, PWM, SOUND, PAUSE, OUTPUT, LOW HIGH, TOGGLE - the list goes on. We are

going to use SERIN and SEROUT as the main highlights of this project.
The Basic Stamp has a built in UART (universal asynchronous receiver transmitter) which can be assigned to any pins of the Stamp by issuing the appropriate SERIN pin, baud rate, (qualifier, qualifier), variable, variable - for instance:

```
SERlN 7,n2400, (">00"), B0, B1
```

would tell the Basic Stamp to walt at the serial port pin assigned - in this case, pin 7 for a 2400 baud signal transmitted from the FC (let's say COM1) with the ASCII characters >00 embedded as a lead in code,. If the lead in code was not transmitted, the Basic Stamp would sit and wait for the next set of characters that matched the lead in code. Once the lead in code was detected the Basic Stamp would then store the next two ASCII characters transmitted into registers BO and B1 for later processing - its that simple!

If half a dozen Stamps were hung on a net then the lead in code becomes useful as it uniquely identifies each individual Stamp. For instance, STAMP 2 could have a lead in code of ">01" and so on - all it takes is to programme each Stamp with its own address at programme time.

Of course, any good programme sending serial data has a feedback or error checking loop (somebody may have pulled the serial plug out of the PC connected to the Stamp) In this case we would retransmit the data we received back to the PC from the Stamp using the SEROUT pin, baudrate, (data, data, ...) ,for Instance SEROUT 6,n2400, (">00",B0,B1 ," RECEIVED",13, 10) would transmit the lead in code the contents of B0 B1, the ASCII characters "RECEIVED" and finally a carriage return ,linefeed, to the PC.

This, of course, is not foolproof error checking but is good enough at this Baud rate and the distances involved.

Now that we have got the data into a couple of registers (BO B1),and we have established how to format a reply to the PC's com port, we can start organising the Stamp to do some work, which in this project is to control four relays connected to the Stamp via a Driver (IC1 see fig 2).

Relay controller criteria

The criteria for this project was based on the premise that I would be able to use any old communications programme on my PC to communicate to the Stamp and, as most comms programmes in Terminal mode communicate in ASCII, the Stamp programme would have to allow for this.

I required that once a relay was switched on by a command from the PC then it stayed on until a further turn off command was transmitted from the PC (except under power failure condition when the relays would switch off anyway and remain switched off until the PC commanded them on again).

First panic set in - "how am I going to convert all of those ASCII characters to Hex or binary to get the ones and zeroes out on the pins without running out of memory - I only have 256 bytes?". Well, once the panic died down, it proved simple - a lookdown table is another one of the Stamp's instructions which we will attack In just a while.

First thing I needed to do was to get the Stamp to recognise a lead in code, I decided to use >00 . This code is not strictly needed, but as I had grander ideas of setting up a net based on a RS 422/RS485 multidrop , then eventually I would need the lead in codes so that the net could broadcast to all the Stamps, and only the Stamp addressed would respond.

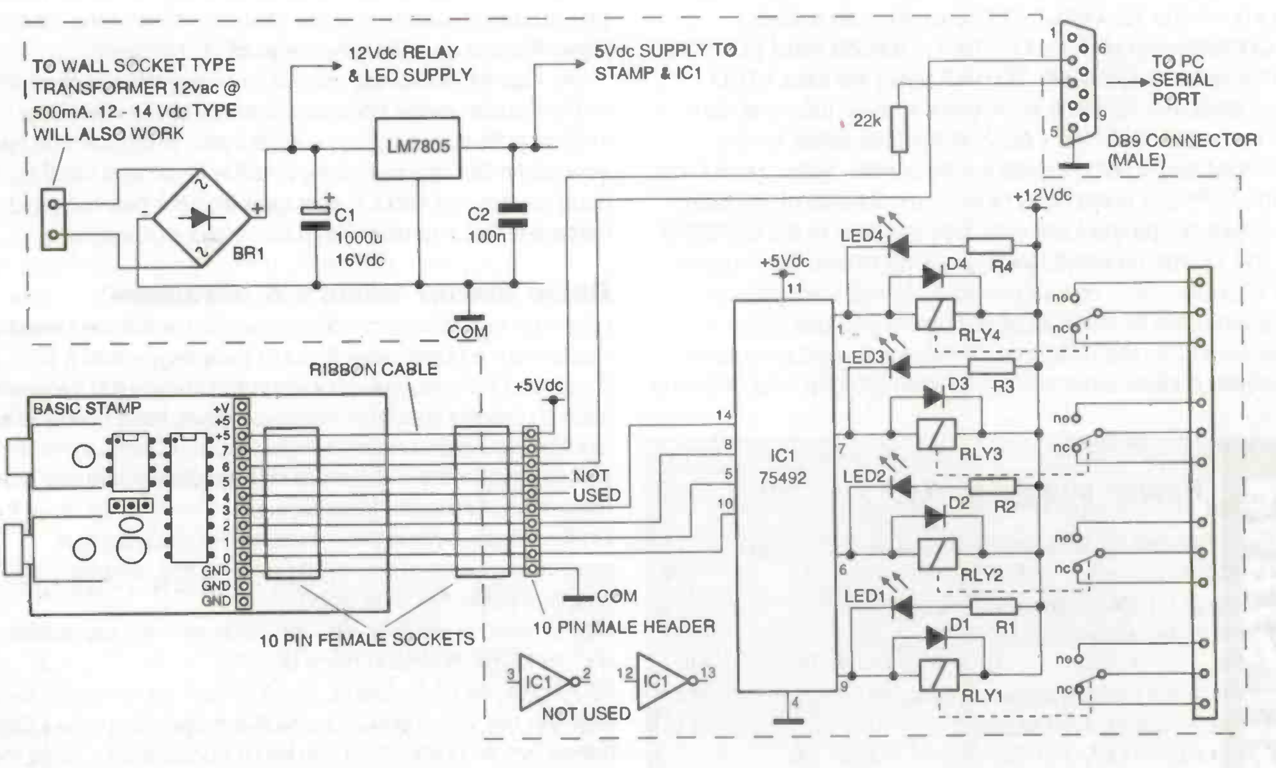


Fig. 2. Basic Stamp RS232 reley driver circuit diagram

The next step was to assign a register to hold the switch on/switch off character so B0 was chosen for this. The characters chosen to signify switch on and switch off were K (uppercase) for ON and L(uppercase) for OFF.

After this, all I had to do was assign register B1 to the data the Stamp was going to see coming from the PC com port. This data of course contains the ASCII code for the particular relay I wished to switch on, or off. If I also wished to have the facilities to switch on and off multiple, or all relays in one go I could also do this with the one data byte.

Pin 13 (bit 7) of the Stamp was then selected to use as the serial input (see circuit diagram fig 2). Note the strangely simple connection from pin 3 on the DB9 connector to pin 7 on the Stamp and the lack of a RS232 line receiver - only a 22k resistor limiting the current, apparently Parallax included clipping diodes on all the inputs so the Stamp will accept the + and - RS 232 signals with ease.

Pin 12 (bit 6) of the Stamp was chosen as the serial output and is connected directly to the DB9 connector pin 2. You may ask where the RS232 line driver is. Well,

Parallax say in their manual "The Stamp's serial output uses 0-5 volt levels, so it may not drive RS232 devices without additional circuitry .We have found however, that most PC serial ports will work without additional circuitry".

Do you think they are telling you porkies here? Well they are not the three totally different PCs I tried the project on, all worked well at 2400 Baud.

However, if you intend to use the project over long lines then you should consider using line driver/receivers such as the MAX 232.

Finally, pins 6 through 9 are used to drive the relays (see fig 2) via a 75492. The 75492 is really a MOS digit driver that I found in my parts box, which is capable of sinking the relay currents at the required 12 vdc so I pressed it into service for this project.

D1 through D4 are 1N4001 diodes to protect the Driver IC from relay back EMF,LEDS 1 through 4 and their associated resistors give visual indication which relays are switched on. The relays used are 12vdc types RS part number 349-658.

This brings us to another Basic Stamp instruction, PINS. First you set up the direction you want the data to flow either in or out. The instruction for this is DIRS. This can be in decimal hex or binary - just add the appropriate prefix symbol nothing for decimal, $ for hex, and for binary, pick the pins you wish to use as input they are signified by a `0', then pick the pins you wish to make outputs signified by a `1'.

In our case we would start the programme with a line that looks like this DIRS =%00001111. This will set up the pins in binary, RB0—RB3 will act as outputs RB4—RB7 will be inputs.

After setting up the DIRS we just issue the statement (let's say) PINS = $0F and all the relays switch on.

Serial relay controller software

I remember not so long ago been asked to design and build a Relay controller similar to this project, using a 1015 UART all in hardware and one of the constraints was that it had to fit in a box not bigger than 4 x 4 x 2 inches.

After looking at the size of the UART chip, the baud rate generator and all the support devices needed, it looked like an impossible task and indeed it was. It ended up on a piece of 8 x 4 inch strip board, and that did not include the Transformer or Relays.

The moral of this is why use complicated hardware when with a little software you can do it better with a Basic Stamp.

The whole programme for the Serial relay controller fits in just 13 lines of code, so there is plenty of scope left to expand on the basic theme.

Referring to figure 3 Software listing, notice the familiar DIRS statement setting up the pins. Line 2 declares a label; this is signified by the colon following the word loop; this is followed by a statement to zero out b3 separated by a colon from the now familiar SERIN (Multiple statements can be included in Stamp Basic use a colon to do the separation.

Line 3 tells the program to gosub to label `txout' on line 12. This line retransmits the contents of b0 and b1 back to the PC to signal that is has received the transmission, line 13 returns the programme back to line 4.

As I mentioned earlier, a lookdown table is used to convert the ASCII characters received in bl back to a binary number the Pins can handle.

The Stamp handles the lookdown table like so; first give the Stamp the Instruction LOOKDOWN, then the register it is going to find the data, in our case b1. After this, give it a series of ASCII numbers (48, 49) (the table) Finally assign it a register to store the result b3.

In our case, during operation, the lookdown instruction will look through all the numbers in the table, until it finds a match for the ASCII character It has stored in b1; if it does not find a match it ignores the line and b3 stays set at `0'.

If it does find a match it counts which number it was from left to right within the table. If it is the first number in the table b3 will store the number `1' ,if it is the fourth number in the table b3 will store the number 4, and so on.

Notice that the numbers in the table are the ASCII numbers 0 through 9 and uppercase A through F, or 0 -15 in binary; b3 will be used later in the programme to operate the relays indirectly.

Line 5 looks to see what character was transmitted into b0 and directly analyses the ASCII character, hence the quotation marks around it. If the character "k" is not found ,the programme moves to line 6.

If the character "k" is found (switch on) then the programme jumps to the label `swon' line 8. At this point the PINS instruction comes into action. The outputs on the lower four bits of the Stamp will assume the result of the OR operation, based on the present state of the lower 4 bits (pins 6, 7, 8, 9) ORED with b3 (remember our LOOKDOWN table stored b3), note the OR symbol:

As can be seen from the line we have just looked at, the OR operator will switch on further relays as instructed by the last PC transmission but will mask off any that are already switched on, leaving them in the on state.

Line 11, GOTO returns the programme to the `loop' label line 2, and the processor waits for the next character to be transmitted from the PC.

In the above we assumed that the Stamp had received the "k" character in b0. However, it could easily have been the "L" character (switch off).

Figure 3. Basic stamp serial relay driver, basic programme listing.HEXRS.TXT

```
1Dirs =%00001111 `Set up the pin directions
2loop : b3=0 : serin 7 ,n2400, (">00") ,b0 ,b1
`Zero b3,set pin 7 ser input
3gosub txout `GOSUB to label txout and  Return the
data to the PC
'Select the Hex ASCII code and put it in b3

4      lookdown b1, (48, 49, 50, 51, 52, 53, 54,
```

```
55, 56, 57, 65, 66, 67, 68, 69, 70), b3
5      if b0 = "K" then swon `If lead in char in b0
is "K" then goto label swon
6      if b0 = "L" then swoff 'If lead in char in
bo is "L" then goto label swoff
7      goto loop
8      swon: pins = pins : b3 `OR the PINS with b3
and set PINS
9      goto loop
10     swoff : pins = pins&/b3 `AND/NOT the PINS
with b3 and set PINS
11     goto loop
12     txout : serout 6,n2400,(13, 10, ">00", b0
,bl," received", 13,10)
13     return
NOTE Line numbers are for reference only and are
not included in the programme, Stamp Basic uses
labels as line identifiers.
```

The programme in this case would have analysed line 6 ,and moved to the label 'swoff' line 10. Here we go again with the PINS instruction except this time we do an AND/NOT operation, then the result is put on the lower 4 bits (pins 6,7,8,9) of the Stamp.Again note the AND/NOT symbol &/.

This again checks the relay output pins and makes sure that only the relays instructed in the last transmission will be switched off and the rest will stay on.

Line 11, GOTO the label 'loop' instructs the programme to return to line 2 and wait for another transmission.from the PC.

Talking to the Stamp from the PC

Well that's your department, as I do not know what communications programme you are using. You may wish to write a BASIC programme, or in some other language; I can only give you the codes you use to switch the relays on and off. Here are some tips though.

First thing is to make up a suitable transmission cable with female DB9 connectors at either end, the PC transmits out on pin 3 and pin 3 on the Stamp end is set to receive so connect pin 3 to pin 3 on the DB9 connectors.

The PC receives data on pin 2 and the Stamp end is set to send data on pin 2 so, again, connect pin 2 to pin 2 on the DB9 connectors.

We also need a ground or earth. This will be pins 5 at both end of the DB9 connectors.

NOTE: You may have to experiment some with the above. With some comms programmes, the PC may refuse to send out data until the handshaking lines have been either Jumpered or disabled in Software. In this case, you have no alternative but to read the appropriate manual, to find out how to do it.

Set your baud rate to 2400,8 data bits, no parity,1 stop bit put your comms into terminal mode; Teletype will do just fine. Make sure you do not have a modem selected; this contains lead in characters for the modem itself. Turn on the local Echo so you can see what you are typing to avoid confusion.

If you have Windows, you have it made. Go into terminal and set up the above, and you are online.

Codes to send to the Stamp

it does to switch . The relay off the data byte is transmitted as ASCII but formatted in Hex for transmission.

If you wished to switch on relay '1' and leave all the others off you would type >00K1 and relay 1 would switch on. If you already had relay 1 switched on and wanted to switch it off, you would type in >00L1 and the relay would switch off.

As can be seen from Figure 3, the ASCII characters matching the Hex numbers 0-9 and (uppercase) A-F. This is to enable you to switch on any or all of the relays with one byte of data instead of having to switch on individual relays with separate transmissions

Example: Typing in >00KF would switch on all the relays and as the same applies to switching off relays >00LF would switch them all off.

Individually the relays can be switched on and off

using using the following codes

```
Relay 1 >00K1 (on) ,>00L1 (off).
Relay 2 >00K2 (on) ,>00L2 (off).
Relay 3 >00K4 (on) , >00L4 (off).
Relay 4 >00K8 (on) , >00L8 (off).
welcome datacomp
```

When typing in the codes for transmission, type exactly as it appears above; do not include spaces ,if you type in lower case the Stamp will ignore the transmission.

What can I use it for?

The Basic Stamp is a quick and easy method of getting your project up and running fast. After that, you could machine code it into one of the cheaper Pics for production use. Although I can see it being used in its own right (SIP Version) in limited production situations such as programmable thermostats, serial transmission humidity detectors, and portable equipment applications.

Maybe I will get to try the 24 pin version next.

Designing a Pic Micro controller based project

PART 3

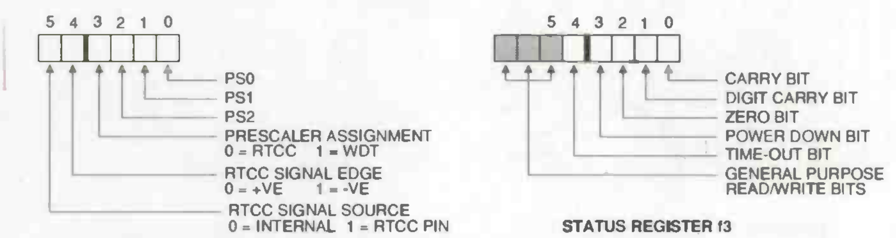
In part 3 of this short tutorial series, Bart Trepak continues his construction of a PIC based alarm clock

Last month's article dealt with the instruction set for the PIC and developed the circuit for the digital alarm clock. This month, we will deal with developing a flowchart and writing a programme to generate a time base for the clock and also discuss the use of an assembler programme.

Timebase

The most important part of the clock is the timebase which the clock counts to determine when each second, minute or hour has passed. This can be any regular waveform such as the 50Hz mains or that obtained from a crystal oscillator. This circuit uses a crystal oscillator making it independent of the mains supply, but the same basic programme could be used for a 50Hz system, taking into account the much lower frequency. This part of the programme will be tackled first but, before we do this, it is perhaps useful to look at how this would normally be done when using a microprocessor and then how it can be done using the PIC15C54.

In a microprocessor system, a timebase can be generated in two ways. One way is to write the programme in such a way that all programme paths take the same length of time to execute. No matter which path the programme takes, due to, say, someone pressing a switch or not pressing it, the same number of steps are executed resulting in the same time being taken to execute the programme. This can be done by breaking down longer programmes into shorter sections and padding out short sections with NOP instructions or other short sequences. Thus, if it takes, for the sake of argument, 100uS to execute each section, then this can be used as a time base. A counter can be initially set to 100 (decimal) and decremented each time such a segment of code is executed.



PS2	PS1	PS0	RTCC RATE	WDT RATE
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Fig.2a. Pictorial representation of data memory in PIC

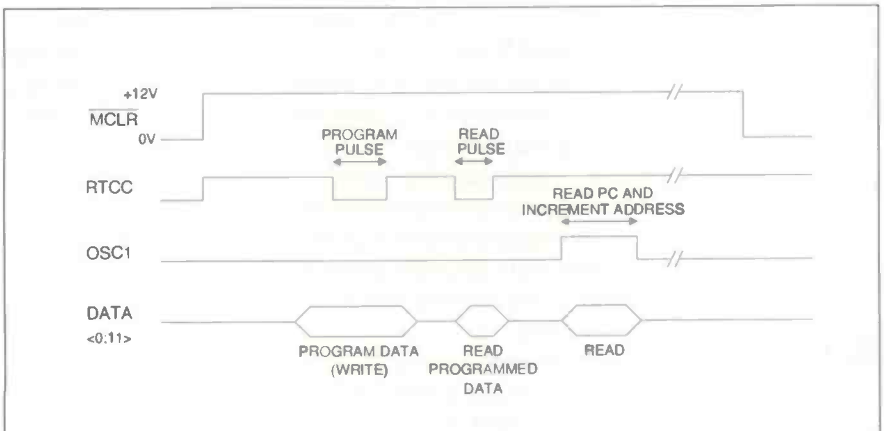


Fig.3. I/O registers

allowing the RTCC counter to be incremented at a much lower rate than the crystal frequency to provide a slower time base.

If a 3.2768MHz crystal is used in the oscillator and the prescaler division ratio set at 256 by suitably programming the OPTION register, the RTCC register will be incremented at a frequency of 3200Hz (i.e. 3276800 divided by 4, divided by 256). By loading the RTCC register initially with 224 decimal, the zero state (equivalent to a decimal count of 256) will be reached after 256-224=32 cycles, giving a period of 32/3200=0.01Sec or 10mS. Therefore, as soon as the RTCC zero is detected, the processor can deal with this and then happily execute the rest of the programme safe in the knowledge that another RTCC zero would not occur for another ten milliseconds. During this time, it would need to execute the whole programme, including displaying all the digits, reading the switches, adjusting all the counters, comparing them to the alarm time set and switching on the buzzer if required. As long as this took less than 10mS, it could get back to reading the RTCC before it reached zero again. This may sound like a tall order until it is realised that a 3.2768MHz oscillator, even when divided by four, will result in an instruction time of 1.2uS, thus enabling some 8,000 instructions to be executed in this time!

It would therefore reach zero every 10mS which could be further counted (divided) to obtain seconds, or half seconds or whatever. Quite mind-boggling, especially in a complicated programme with many different paths and, if the processor has a counter/timer, this is often used instead.

With this method, the counter/timer is loaded with a number and then decremented, either in response to a signal at an input pin (such as the 50Hz mains frequency) or by an internal signal obtained from the micro-processor clock which, if a crystal oscillator is used, would also be a regular frequency. Since this goes on independently of the processor, the processor can get on with the other tasks that it has to do, such as driving the displays, scanning keyboards etc. When the counter reaches zero after some required period of, say, 0.01 second - which of course depends on the number originally loaded into the counter and the frequency at which it is decremented - an interrupt is generated. This causes the processor to suspend execution of the current programme and go to a subroutine which would increment another counter which after so many counts, 100 in this example, would increment the seconds counter and also the minutes and hours counters if required before resuming the original programme.

Although the PIC16C54 series micro controller has a counter/timer (RTCC f01), it does not support any interrupts, so a slightly different approach must be used. The RTCC counter can be incremented automatically by the crystal oscillator or by a signal to the external RTCC input pin. Here it is also loaded with a suitable value but it must be read each time it is incremented to ensure that the zero state is not missed. Otherwise, the counter would simply overrun and start counting again leaving no indication that the zero point had been reached and passed. From this it would seem that there would be very little time to run any other programme because each read instruction would increment the counter and would need to be followed by another read and another to ensure that the zero state had not been missed. However, if the original number loaded into the counter was, say, 100, then after a zero was detected we could execute, say, 90 or so instructions before returning to the programme to read the RTCC counter and still be sure that the next zero had not been missed. (Note that the RTCC register can only be incremented and not decremented by the clock so to count to 100 it would need to be initially loaded with 156 decimal so that it reached 256, the equivalent of zero since the maximum count is 255, after one hundred counts.) The PIC also has a prescaler which can be set to give a division ratio of up to 256,

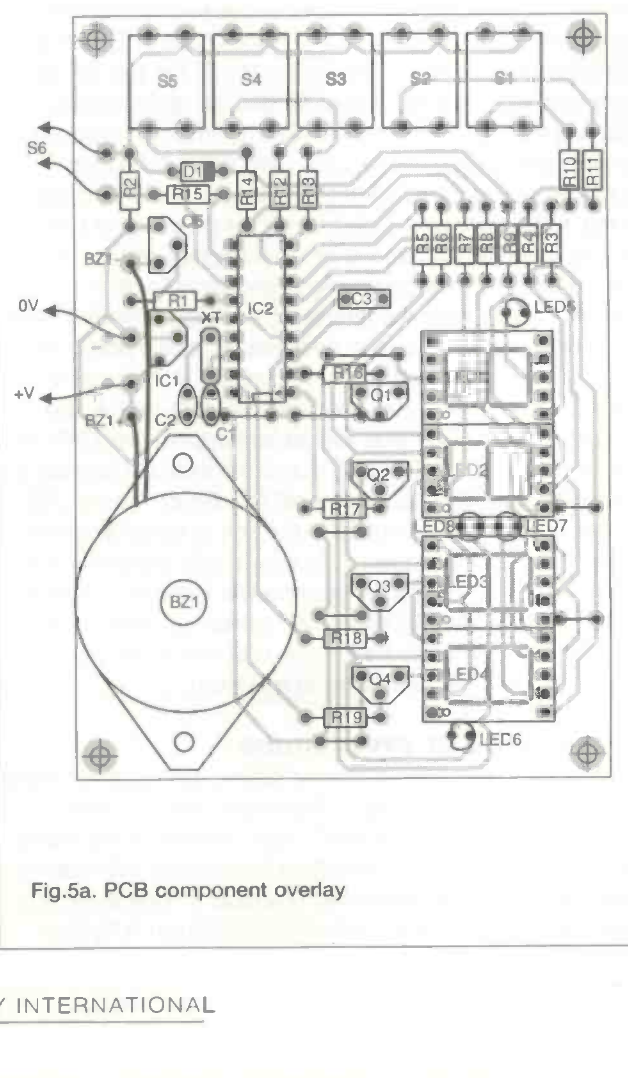


Fig.5a. PCB component overlay

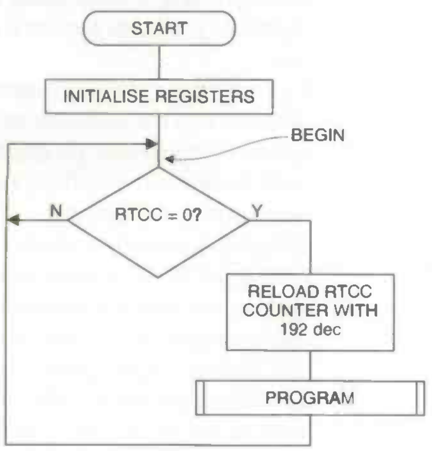


Fig.6. Clock timebase flowchart

After each 10mS period, it is an easy matter to decrement another counter which, if it is initially loaded with 50 decimal, will reach zero every 0.5 Seconds (to flash the colon) and this can be further divided to give seconds, minutes and hours. Note that a 10mS period is chosen because this will also determine the frequency with which the display is refreshed and lower frequencies could cause the display to flicker since each digit is only on for a fraction of the 10mS period.

Timebase flowchart

A basic flowchart can now be drawn for this part of the programme and this is shown in Fig 6. Starting at the point marked BEGIN, the RTCC counter is checked to determine if it has reached zero. If it has not (N), it goes back to BEGIN and checks again. Eventually (once every 10mS) the test will be successful and the programme will take the other branch, reloading the RTCC counter with 224 decimal and executing the part of the programme marked PROGRAM. This will eventually consist of the various counting operations, display driving, keyboard reading etc but, for the purposes of testing this part of the programme, it simply complements port A and then goes back to the beginning as shown in fig 6b.

Before this programme can work, the microcontroller must be set up to run it. When it is powered up, all the ports are configured as inputs and, if it is required (as in this routine) to connect an oscilloscope or frequency meter to port A, then this port must first be made into an output port. The OPTION register must also be set up to assign the internal prescaler to the internal instruction clock and set the division ratio to 256. These functions are performed in the part of the programme marked INITLSE and shown in fig 6a. As with the part marked PROGRAM, this will also be expanded as in the final version the initial conditions at power up of the various other counters (not yet defined) will also need to be specified in this part of the programme when we come to add them.

The Assembler programme

The flowchart must now be converted into a programme using the instructions shown in fig 3. The micro-controller itself, however, does not "understand" these instructions any more than it understands the flowchart as it deals only with data in binary form (machine code) which consists of 1s and 0s. A programme written using the mnemonics shown in fig 3. (called the source code) will therefore have to be converted into this binary form at some stage before it can be loaded into the EPROM on the chip and executed. This could be done by hand, by converting each instruction into its binary equivalent by looking it up in a table, for example. All these 1s and 0s could then be loaded into a programmer (bit by bit or typed in, depending on your programmer) and a chip blown. This would be a very tiresome and error prone process which, thankfully, is not necessary because special programmes called Assemblers are available which do the job quickly and with no errors.

An Assembler programme for the PIC series of micro-controllers is available from Microchip Technology which runs on a standard IBM or compatible PC and converts the source code into machine code. The mnemonics (or source code) are simply typed into the computer using any standard word processor programme which can produce ASCII code and the Assembler programme produces the machine code version ready to load into the programmer. To do this, however, the mnemonics have to be written in a special format so that the Assembler programme can operate on it and the rules for this are shown (and explained) in fig. 7, together with a typical listing.

Assembler programmes from other suppliers may have different features and different protocols to those discussed here. Even upgraded and later versions from the same supplier may also have changed in some details. The reader should, therefore, consult the manual supplied with the particular Assembler programme before use and regard what follows as typical of most Assemblers but not neccesarily correct for your particular one.

Before going on to write the mnemonics for the timebase programme, it would be useful to discuss some of the advantages of a typical Assembler programme. This would have a number of features which are very helpful to the programmer beyond simply converting the mnemonics into machine code. Often in a programme, instructions will refer to registers such as the RTCC (01) or PORT B (06) as well as other registers which the programmer may wish to invent such as MINCTR (ie. minute counter) which would be assigned to one of the general purpose registers such as 09 for example.

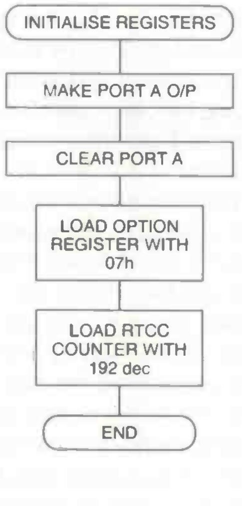


Fig. 6a. Clock initialise routing flowchart

In a complicated programme, where many registers are being used, it would be difficult to follow the operation if all the registers had to be referred to as 01 or 09 and the programmer would need to keep referring to a list to know which register was which and what it was doing to be able to follow the programme. The assembler programme therefore allows such registers to be defined before the programme is run so that instead of typing 01 or 09 the programmer may write RTCC or PORTB or even MINCTR and leave the programme to interpret this as required.

Many of the often used symbols such as the special function registers RTCC, PORTA, STATUS and so on together with words like “zero”, “w” and “same” can be assigned values such as 01, 05 and 03 or 0, 1. As these would always be the same for every programme using the PIC (the RTCC register is always register 01 for example), to save having to type them in at the beginning of every programme these can be stored in a separate list which can be included in the programme listing by using the “include” statement. An example of such a list (PIC.H) is shown in FIG. 8 where for example RTCC is equated (equ) to 01h and PC equ 02 and so on. Programmes may now therefore be typed using instructions such as MOVF PORTA,w instead of MOVF 05,0 which makes the whole listing much easier to understand and the assembler will substitute 05,0 or the binary equivalent of this in the instruction when it is run. Thus although from the timebase listing shown in FIG. 9 it may appear that RTCC, PORTA etc. have not been defined, this has in fact been done by the “include PIC.H” statement.
Another extremely useful feature is the ability to use labels to mark points in the programme. In FIG. 7, for example, there is a point marked BEGIN to which the programme has to go to if the RTCC register has not reached zero. Normally, the programmer would use a “goto k” instruction in this situation where k is the address (in binary or hex) of the point in the programme where BEGIN is located. While this may be easy to work out for the timing programme now, as more instructions are added to the INITIALISE routine the point marked BEGIN will move to another address and the k in the “goto” instruction will need to be changed. With many “goto” instructions to keep track of, only some of which would need changing as new instructions were added, there would be many mistakes and finding them would be extremely difficult. Instead, the assembler allows the programmer to write “goto BEGIN” and as long as there is only one unique label called BEGIN, the programme will alter the address for all the “goto” instructions affected as more lines are added and the programme re-assembled.
This is not the end of the story and some assembler programmes offer other features such as a Macro capability. Macros allow solutions to frequently used functions to be designed and stored as self-contained programmes. These may be added to a programme as required without having to re-define register names and avoid the tedium of re-typing the same programme sequence each time it is used in a new application. A discussion of these is beyond the scope of this article but they are described in the manual supplied with the programme

CHOOSING YOUR OPTIONS

We are now in a position to convert the flowchart to a source code programme using the mnemonics and a solution to the timing programme is shown listed in FIG. 9 using the layout outlined above which will allow the assembler programme to

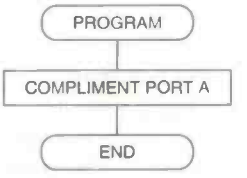


Fig.6b. Clock program flowchart

interpret it correctly and convert it into object code (the binary 1s and 0s) required to programme the device. This can now be explained in detail and compared to the flowchart of FIG 6. The INTLSE routine starts by setting up the I/O ports which, as mentioned, are all inputs at switch on. This is done by first loading the W register with a binary number (putting 0s in the bits which are to be programmed as outputs) using the “movlw” instruction. This moves the literal number (specified as 00h in this instance which is of course 0000 0000 in binary) into the W register and this is transferred to the TRISA register using the “tris PORTA” instruction to make port A an output. Note that in the case of port A, only the lower four bits are used since the port is only four bits wide, so that the hex number could just as well have been X0h (where X is any valid hex digit). This does illustrate an important point as far as the Microchip assembler is concerned however and it is that all hex numbers must start with a numeral so that F0h, A5h, C3h etc. would not be acceptable and would generate an error and so would have to be written as 0F0h, 0A5h or 0C3h while numbers such as 9Fh or 5Bh would be interpreted correctly.
If the port A were to remain as an input, then of course, nothing would need to be done while if a mixture of inputs and outputs were required on a port, then the literal loaded into W would have a number with 0s corresponding to outputs and 1s corresponding to inputs in it. These I/O lines may change during the course of a programme as some subroutines may require certain lines which were outputs to function as inputs or vice-versa so the “tris” instructions may be used again at the beginning of some subroutines. Even if none of the lines are required to change, it is often a good idea to redefine input and output ports at intervals during the execution of a programme because in noisy environments, the TRIS registers could become corrupted with disastrous results as far as the programme is concerned.
The rest of this routine is similar and simply loads different values into W using the “movlw” instruction and transfers them to the required register such as the OPTION and RTCC registers using the appropriate instructions “option” (which loads the OPTION register) and “movwf” (which moves the value in W into the register specified).
As mentioned in PART1 of this series, the OPTION register is a special register which allows the programmer to define how the RTCC counter will work. By setting or clearing certain bits in this register the prescaler may be set for a division ratio from 1:2 to 1:256. The signal source for the RTCC counter (ie internal clock or external pin) as well as the signal edge (positive or negative going) on which the RTCC is to be incremented can also be defined. Setting bit 3 enables the internal prescaler to be assigned to the Watch Dog Timer. In this project, a division ratio of 1:256 is required with the internal signal source and the prescaler assigned to the RTCC register and from FIG 2a. this translates to a value of XX0X

0111 in binary (where X is don't care) so a value of 07h will give the required results.

The timing routine starts at BEGIN and uses the "movf RTCC,w" instruction which causes the zero flag to be set in the STATUS register if RTCC=0 without actually changing the value in the register. This flag is then tested using the "btfss STATUS,2" instruction which skips the next instruction if the zero bit is set. If it is not set (ie. RTCC is not zero) the next instruction "goto BEGIN" is executed which takes the programme back to BEGIN to check the RTCC for zero again. If RTCC is zero, then the "goto BEGIN" instruction is skipped and the RTCC is re-loaded with the value .224 by the instructions "movlw .224" which moves the binary equivalent of 244 into the W register and "movwf RTCC" which moves the contents of the W register into RTCC. After this, the rest of the programme PROGME (called PROGRAMME in the flowchart) is executed.

In this case, as mentioned, the PROGME routine consists of simply complimenting output port A so that it can be monitored on an oscilloscope or frequency meter to test that the timebase is operating correctly which is done by the instruction "comf PORTA" followed by "goto BEGIN" which will return the programme back to BEGIN to check the RTCC for zero again. Once the programme has been checked, this part will be deleted and replaced by other routines for updating counters, driving displays, reading the key board and doing all the other operations required.

The "ORG 1FF" statement at the end of FIG. 9 sets the programme origin and this is the address 1FFh which is the point that the PIC16C54 starts from following a power on reset. The first thing that the PIC does after this is to increment the Program Counter (which overflows to zero) and the first instruction is therefore fetched from the EPROM at address 0. This is the "goto START" instruction shown on FIG. 9 and causes the programme to jump to the location of the label "START" which in this instance happens to be the next instruction anyway and seems pretty pointless. In this

```
This file defines labels for ports, reset
vectors, registers etc. for the standard PIC
parts. It is saved under the name PIC.H
;
;
PIC54                         equ          1FFH
    ;define reset vectors
PIC55                         equ          1FFH
PIC56                         equ          3FFH
PIC57                         equ          7FFH
;
RTCC                  equ          01H
;define registers
PC                    equ          02H
STATUS                equ          03H
FSR                   equ          04H
;
PORTA                 equ          05H
PORTB                 equ          06H
PORTC                 equ          07H
;
w                     equ          0H
same                  equ          1H
```

Fig 8 Program Include Statement

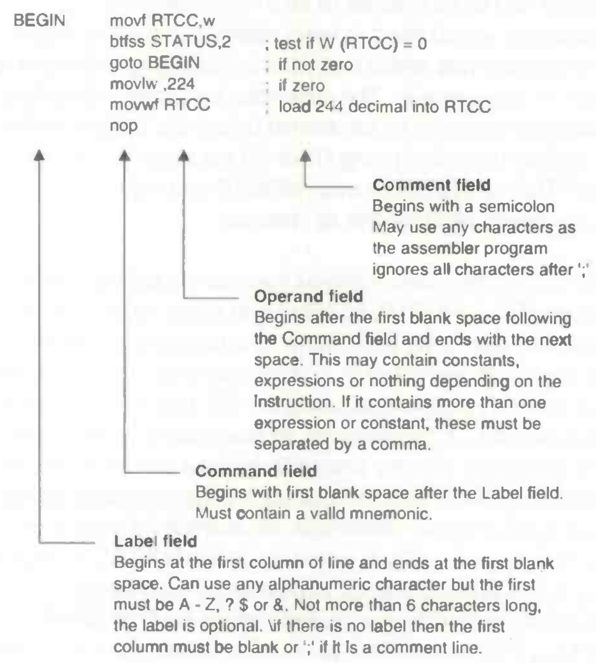


Fig.7. Source code fields

simple programme it is and the initial "goto START" instruction could be left out but with more complex programmes, the start of the programme will in all probability not be at the beginning especially when subroutines are used as we shall see later. Finally the "END" statement is required to tell the assembler that there are no more instructions to be executed. Anything after the END statement will be ignored by the assembler.

FROM PAPER TO SILICON

As mentioned earlier, the micro-controller does not understand the mnemonics that have been used so far and the programme has to be re-written in binary with 1s and 0s and this is the job of the assembler programme which is supplied with the chip programmer for the PIC or can be downloaded from Microchip Technology's bulletin board via a modem. All that needs to be done is to type the programme of FIG. 9 into a PC using a standard word processor programme and stored in a file called say CLOCK.ASM (the .ASM extension presumably stands for "assembler code" but it is in fact a standard ASCII file).

Once this has been done the assembler programme is run and this goes through the CLOCK.ASM file and checks that there are no errors such as "goto BENIG" which tells the programme to branch to a label which does not exist, or two or more identical labels have been used. It also checks that correct extensions have been used on instructions which specify a file register address and destination and that all subroutine calls end with a return "retlw" instruction. What it cannot do is tell if the programme is logically correct and does what the programmer intended it to do. That is up to him!

When this programme has run (which takes a few seconds depending on your machine) it will show any errors encountered on the screen. Errors are specified by the line number where they occur together with a brief description and how serious the error is eg. fatal or just a warning that something may be wrong. These have to be corrected before the next step can be taken and will involve going back into the CLOCK.ASM file, making the necessary corrections and re-running the assembler programme again.

When the assembler has been run and no errors have been detected, the programme will have generated a number of files called CLOCK.LST, CLOCK.OBJ and various other output files. Some of these are required for use with the simulator programme and will not be discussed here. The .LST file is a copy of the .ASM file with the location or programme counter and object code (the code that will be loaded into the EPROM in the micro-controller) also shown together with the line numbers. This makes it possible to locate any errors which are reported when the assembler programme is run but it is the .ASM file which needs to be corrected. Most mistakes are usually simple typing errors and are easily found. At the end of the .LST file there may also be a cross reference table listing every symbol used and which lines it appeared on. The .OBJ file contains the object code (the 1's and 0's) which can be loaded directly into the chip using a suitable programmer. This file can be output in different formats for use with Microchip's and other manufacturer's programmers.

Microchip also supply a software simulator which enables the programme to be executed or single stepped one instruction at a time on the host PC or compatible computer while selected registers are displayed on the screen. This is useful in finding some faults in a programme which would not be apparent from the logic. It is also possible to change inputs at various stages in the programme execution to see what effect these have and show up any faults which may then be apparent. Another useful feature is the ability to set "breakpoints" in the programme to make the programme stop when it reaches a particular address in its execution, enabling the user to single step the programme from that location while executing sections of the programme known to be working quickly. The use of this programme will not be discussed but it is very easy to use and can be very useful in finding tricky faults when the processor does not seem to do what the programmer intended. Using the simulator and observing how each register changes following each instruction will usually reveal the source of the trouble.

A suitable programmer is now required to programme the chip with the code generated by the assembler. There are many programmers on the market and some have even appeared as constructional articles in hobby magazines. A low cost programmer, the PICSTART 16B is also available from Microchip Technology which is a small printed circuit board with a socket to accept an 18 or 28-pin micro-controller. This is powered by a small d.c. supply and connected to a serial port of the IBM compatible PC via a cable (all supplied). As well as this, the pack contains the Assembler and Simulator software together with data books manuals and a few sample chips all for around £120. The software supplied allows the assembled machine code (in this case the CLOCK.OBJ file) to be sent to the programmer module and a chip programmed. During this procedure, the Oscillator configuration fuses must be set to "XT" and the WDT fuse set to disable the Watch Dog Timer function which is not used in this design. The details for doing this will depend on the programmer used so the user is referred to the manual supplied with the programmer.

Assuming that the printed circuit board shown in FIG. 5a has been constructed, the programmed chip can be plugged into the socket and any port A pin checked to see if it has a 50Hz signal on it which would prove that, so far at any rate, the programme is working correctly. Alternatively, the chip itself may be connected up to a crystal as shown in the circuit diagram on a breadboard and the operation checked.

Sooner or later however, the displays and keyboard will need to be checked so it is probably as well to build the circuit to begin with on a printed circuit board and perform all the tests on this.

The chip can now be erased using a UV EPROM eraser and the next new piece of code developed for the clock. This will in turn be added to the current programme and the process repeated so that the full programme for the clock will begin to take shape.

```
;
     LIST P=16C54              ;f=inhx16
     INCLUDE ìPIC.Hî
                               ;
     goto START
                               ;
START       nop
                               ;
                               ;INITIALISE ROUTINE

INTLSE      movlw 00h          ;ie 000 000
     tris PORTA                ;make PORTA o/p
     movwf PORTA               ;make PORTA 0000
                               ;
     movlw 07h                 ;ie 0000111
     OPTION                    ;int signal, L to H,
                                presclr 256 to RTCC
     movlw .224                ;move 224 decimal into
     movwf RTCC                ;RTCC - gives 10mS with
                                3.2768mhz xtal
                               ;
BEGIN      movf RTCC, w
     btfss STATUS, 2           ;test if w (RTCC) = 0
     goto BEGIN                ;w (RTCC) not zero
                               ;
     movlw .224                ;if RTCC=0 move
     movwf RTCC                ;224 dec. into RTCC
PROGME      comf PORTA, same   ;switch port A on
                                and off
                               goto BEGIN
                               ;
                               ;
     ORG 1ffh
                               ;
     END
```

FIG. 9. Code for CLOCK.ASM TIMEBASE (seeFlowchart 6)

Next Month...

Next month's article in this series will deal with LED display driving and counting techniques and introduce some of the other instructions available to the programmer.

8088 Interrupt based control computer PART 2

The first part of a new computer project designed by R. Grodzik based around the popular 8088 processor and specifically designed for multitasking control applications

Last month we looked at the overall designof a multitasking 8088 processor based single board computer. In this issue we continue by first looking in greater depth at one of the key features of a multitasking system, the hardware and software interrupts

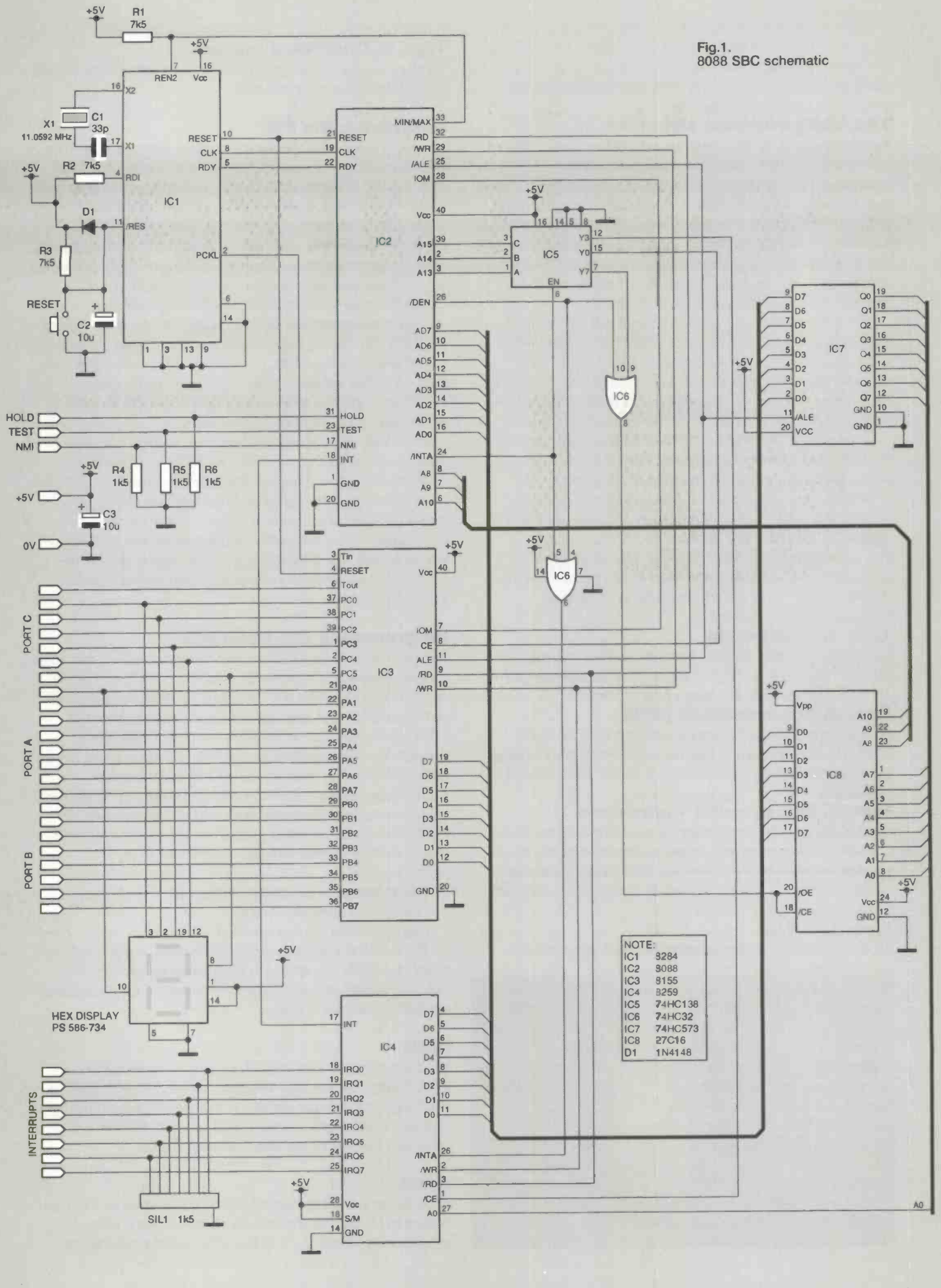
Interrupts can occur by hardware (external) or software (internal) means. The hardware interrupt system is controlled by the 8259 peripheral controller. This chip is a complex IC and contains several internal registers which have to be initialised in order for the PIC to function. In a PC, the EPROM resident software known as the BIOS, runs on switch-on and is quite transparent to the user. This software locates all the hardware peripherals attached to the system and allocated an interrupt number (TYPE) to each. For example, the TYPE 9 interrupt activates when a key is pressed or released, a TYPE 14H occurs on the serial port. All these interrupts are termed 'hardware interrupts' since they are activated on the receipt of a positive going logic pulse on one of the 8259 PIC interrupt lines (known as IRQ lines on the IBM expansion slot).

Associated with the 8088 range of processors are internal 'software interrupts' which occur when certain conditions are met inside the processor. These are sometimes error conditions which occur when the software is liable to 'crash' i.e. a TYPE 0 interrupt occurs if a numerical division by zero is attempted. A pocket calculator will usually indicate an E or error condition. Both hardware and software interrupts can be evoked at any time in software by the use of the 'INT <TYPE>' instruction - useful in checking the effects of an external hardware interrupt without one actually occurring in hardware.

As previously mentioned, the PIC supplies the identity of the hardware interrupt by sending an 8-bit <TYPE> number to the 8088. This TYPE number is internally multiplied by 4 by the 8088 to provide a 'pointer' (address) in external memory RAM which holds the start address of the interrupt service routine program. Since the start address consists of a CODE SEGMENT word and an INSTRUCTION POINTER word (CS:IP) - each pointer occupies 4 bytes. These 4 bytes are located in the bottom of the 8088 memory map (0000-01FFH). For example, a TYPE 6 interrupt will generate a double word pointer of 0018H - an address location containing the IP offset pointer and 001AH - the next word location containing the CS of the interrupt service routine. When a TYPE 6 interrupt occurs, a double word VECTOR address of 001AH/0018H addresses external RAM and the 8088 'picks up' the CS:IP start address at this location in RAM, at which point the 16-bit CODE SEGMENT register and the 16-bit INSTRUCTION POINTER are loaded with these new values. The old CS:IP register contents are saved on stack, the interrupt program runs until a RETI (return from interrupt) software instruction is encountered, at which point the old CS:IP values are reloaded into their respective registers and the main 'background' program re-commences execution. Similarly, a software interrupt will generate an interrupt TYPE number which is again multiplied by 4 to provide a vector address in memory which contains the double word start address of the interrupt service routine.

In summary, if the background program is interrupted by a TYPE 6 hardware or software interrupt at address C123:4567H and, for example, the interrupt service routine for a TYPE 6 interrupt occurs at address F000H:1234H (CS:IP), RAM location 0018H must be loaded with 1234H and RAM

Fig.1.
8088 SBC schematic



location 001AH must contain F000H. When the interrupt is generated, execution begins at address F000H:1234H. On a IRET instruction, the CS:IP registers are popped from the stack and loaded with C123:4567H, and the background program recommences execution.

The 8088 interrupt structure

Since at switch-on RAM contents are indeterminate, EPROM-based software has to load the relevant locations in RAM with the interrupt vectors of any hardware or software interrupt that may occur during the execution of the main 'background' program. This operation is described as the loading of interrupt vectors into the interrupt vector table -the interrupt vector table being the first two RAM page locations in memory. Since this information is lost on power-down, this process has to be repeated whenever the system is powered-up.

The 8088 SBC contains a 8155 peripheral i.c. which contains a programmable IO port, a timer and 256 bytes of RAM. This RAM is selected by the Y0 line of the 74138 decoder, allocating addresses 0000 to 00FFH for this area of RAM. Since this area is also used as a STACK only the lower addressing range is used to form the interrupt vector table.

The vector addresses for software interrupts are located at the bottom of memory, specific address locations which are unchangeable for the 8088/86 range of processors:

The first five locations of the interrupt pointer table are reserved for dedicated 8088 processor functions as follows:

Type 0, divide by zero.

This interrupt activates when a DIV or IDIV instruction produces a quotient that is too large to be contained in the results register. If this occurs, the 8088 automatically generates a type 0 interrupt.

Type 1, single-step

Usually this points to an IRET instruction. If the trap flag has been previously set, a single instruction of the user program will be executed, returning to the type 1 pointer.

Type 2, non-maskable (NMI)

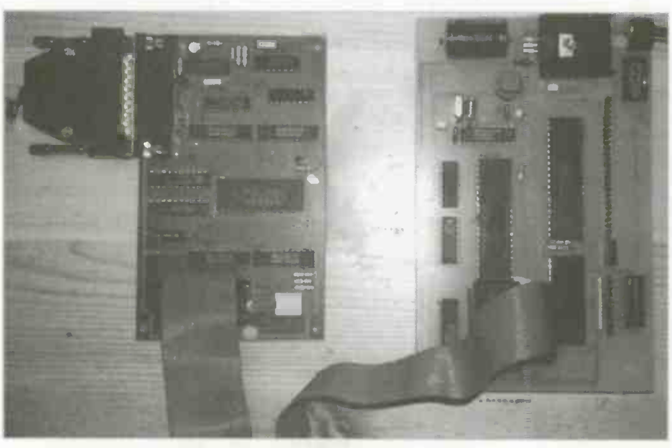
Is reserved for interrupts generated by the NMI pin of the 8088. The IF flag status is ignored and therefore this interrupt will always be acknowledged, since it cannot be 'locked-out' or disabled.

Type 3, one byte INT instruction

Used to implement a breakpoint function in a system. If byte CCH (INT 3) is inserted in a program, execution of the program will stop when CCH is encountered, and a type 3 interrupt routine can then be used for example to examine the contents of the CPUs registers.

Type 4, INTO.

An interrupt will be issued conditionally, where the INTO instruction is executed if the OF (overflow) flag is set.



Type 6, Undefined opcode.

Interrupt occurs if an illegal OPCODE is generated - by a faulty assembler of corruption of data.

Interfacing the PIC

The 8259 is a device specifically designed for use in real time, interrupt driven micro-computer systems. The bi-directional data bus (D0-D7) interfaces directly with the system data bus - control words, status information and interrupt vector information is transferred on this bus. The 8 interrupt lines of the 8259 labelled IR0 - IR7 can be programmed as level (high) or edge (low to high transition). When an external device raises any of these lines high, an interrupt (active high) signal is issued from the INT -pin17 output of the 8259 PIC.

This interrupt request signal is applied to the 8088 through the INTR (pin 18), - a logic active high input. Once the interrupt is recognised by the 8088, the CPU completes its current instruction and issues an interrupt acknowledge pulse from its INTA output (pin 24) to the 8259. Upon receiving an INTA signal from the 8088, the 8259 releases an 8-bit pointer onto the data bus which is read by the CPU. The INTA active low signal is also routed to pin 6 of the 74138 address decoder which tri-states the Y outputs, thereby preventing any data bus contention from other peripheral devices. The pointer value is derived from a combination of initial software programming of the 8259 and by a 3-bit value dependent on which IRQ line was activated. The CPU then uses this value to determine which interrupt service routine is executed for the relevant interrupting device.

Programming the 8259 PIC

The PIC needs to be initialised and programmed, so that each external interrupting device can be allocated its own pointer number. Each interrupt line on the 8259 will, when asserted, be allocated its own interrupt service routine. Additionally, individual IR lines can be prioritised and also disabled by software masking and thus preventing specific external devices from generating an interrupt. Since the 8259 is a fairly sophisticated device, only the basic programming details have been included, which should suffice for most purposes. The 8259 is similar to any other programmable peripheral device in that it contains internal registers which have to be programmed.

The sequence of programming the PIC is as follows:
Loading the command words.
Loading further control words, if required.

Three ICW (initialisation command words) words-ICW1, ICW2 and ICW4 are written to the internal registers by the CPU. An additional operational control word (OCW1) specifies which IRQ lines are enabled.

ICW1

The eight interrupt lines (IR0-IR7) of the 8259 will generate eight different pointer type numbers, which will each point to eight separate interrupt vectors in the vector table at four byte intervals. Level (high) or edge (+ve)triggering of interrupts is controlled by bit 3 of the ICW1 command byte.

ICW2

TThe vector pointer generated by the 8259 when an interrupt is asserted is dependent on a combination of the initial programming of bits 3 - 7 of the ICW2 register and by the contents of bits 0 - 2. Bits 3 - 7 of the ICW2 word remain common to all eight vector numbers, since they are written into the ICW2 register during the initialisation of the 8259. Bits 0 to 2 of the pointer type are generated internally by the hardware of the 8259-an interrupt on IR0 generates 0, 0, 0, an interrupt on IR7 generates 1, 1, 1. So for example. If the ICW2 was initialised as 20H, interrupt IR0 will generate a type 32 (20H) pointer, (the contents of ICW2 (0 0 1 0 0 0 0 0), whereas IR7 will generate a type 39 (27H) pointer - (ICW2 contents 0 0 1 0 0 1 1 1).

ICW4

TThe programming of the ICW4 register determines the action taken by the 8259 at the commencement and conclusion of the interrupt service routine. Interrupt requests are ordered in priority from 0 through 7, with an interrupt issued at IR0 having the highest priority. For a nominal mode of operation the ICW4 register is loaded with 13H:when an interrupt is acknowledged, further interrupts of a lower priority will be locked out until the INTA pulse is issued, which will automatically re-enable the interrupt mechanism once the IRET instruction of the interrupt service routine has been executed.

OCW1

An operational command word (OCW1) can be written to the PIC at any time during the execution of a program. Masking of individual interrupt lines of the PIC is achieved by writing a logic 1 at the corresponding IR positions of the OCW1 word, disabling the interrupt sources of that particular line. Writing FFH to this register will disable all of the interrupt lines.

Initialising the PIC

The example below shows how the 8259 may be initialised. Four command words-ICW1, ICW2, ICW4 and OCW1 are written into the control registers of the 8259. ICW1 is written into PIC base address 6000H, the remaining registers are written into at address 6001H.

```
ICW1 =17H   ;edge triggered  interrupt
 ;level triggered interrupt ICW1 = 1FH
ICW2 =20H   ;bit3 to bit7 of pointer
ICW4 =13H   ;Fully nested mode
OCW1 =00H   ;all interrupts  enabled
 PIC EQU 6000H ;8259 peripheral base address.
 MOV AL, ICW1         ;Write control words to PIC
      MOV DX, PIC          ;A0 = 0
      OUT DX, AL
      MOV AL, ICW2
      MOV DX, PIC+1        ;A0 = 1
      OUT DX, AL
      MOV AL, ICW4
      MOV DX, PIC+1        ;A0 = 1
      OUT DX, AL
      MOV AL, OCW1
      MOV DX, PIC+1        ;A0 = 1
      OUT DX, AL
```

Once the 8259 operational command and control words have been written, the device is active and will respond to an interrupt. The address of the interrupt service routine (interrupt vector) for each interrupting device has to be pre-written into the interrupt vector table. An interrupt on the IRQ0 line of the 8259 will cause a TYPE 32 (20H) pointer which, when multiplied by 4, the 8088 will result in a doubleword vector address found at locations 80H and 82H in RAM. The following program shows how the vector table is programmed for a base pointer of TYPE 32.

This model program will assemble to 2 Kbytes of ROMable code which will run on the 8088 board. On reset, the 8088 boot address (FFFF0H) will generate address location 07F0H of the EPROM area, at which a jump to start of EPROM (address FF800H) instruction will cause execution of the program. The assembler directive 'page , 132 will allow the listing (FILENAME. LST) to printed in a small font 132 column format.

```
;INTERRUPTS
page , 132
CODE SEGMENT
ASSUME CS:CODE
;MAIN PROGRAM
 CLI                          ;Disable interrupts
MOV SP, 0FFH                       ;Stack Pointer at RAM
top
CALL VECTORS                       ;Load interrupt
vector
CALL INITIALISE                    ;Initialise 5259
STI                           ;Enable interrupts

;Background Program

AGAIN:    HLT                      ;Wait for interrupts
 JMP AGAIN                    ;Repeat

                              ;Load vector table
VECTORS PROC NEAR
CODE_SEGMENT EQU 0FF80H       ;CODE SEGMENT

RAMTABLE =ICW2                ;ICW2 base TYPE 32
pointer
;Interrupt pointers for IRQ0-IRQ7
IRQ0 = RAMTABLE               ;TYPE 32 POINTER
IRQ1 = RAMTABLE +1            ;TYPE 33 POINTER
IRQ2 = RAMTABLE +2            ;TYPE 34 POINTER
IRQ3 = RAMTABLE +3            ;TYPE 35 POINTER
IRQ4 = RAMTABLE +4            ;TYPE 36 POINTER
IRQ5 = RAMTABLE +5            ;TYPE 37 POINTER
IRQ6 = RAMTABLE +6            ;TYPE 38 POINTER
IRQ7 = RAMTABLE +7            ;TYPE 39 POINTER

;Load vectors into table

MOV WORD PTR DS:[IRQ0*4], OFFSET ISR0
MOV WORD PTR DS:[IRQ0*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ1*4], OFFSET ISR1
MOV WORD PTR DS:[IRQ1*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ2*4], OFFSET ISR2
MOV WORD PTR DS:[IRQ2*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ3*4], OFFSET ISR3
MOV WORD PTR DS:[IRQ3*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ4*4], OFFSET ISR4
MOV WORD PTR DS:[IRQ4*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ5*4], OFFSET ISR5
MOV WORD PTR DS:[IRQ5*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ6*4], OFFSET ISR6
MOV WORD PTR DS:[IRQ6*4+2], CODE_SEGMENT

MOV WORD PTR DS:[IRQ7*4], OFFSET ISR7
MOV WORD PTR DS:[IRQ7*4+2], CODE_SEGMENT

RET
VECTORS ENDP

INITIALISE PROC NEAR          ;Initialise 8259 PIC

ICW1 =17H                     ;edge triggered
interrupt
                              ;level triggered = 1FH
ICW2 =20H                     ;bit3 to bit7 of pointer
ICW4 =13H                     ;Fully nested mode
OCW1 =00H                     ;all interrupts  enabled
```

```
OCW2 = 080H
;Rotate interrupts

PIC EQU 6000H       ;8259
peripheral base
address.

MOV AL, ICW1        ;Write
control words to PIC
MOV DX, PIC         ;A0 =0
OUT DX, AL
MOV AL, ICW2
MOV DX, PIC+1       ;A0 =1
OUT DX, AL
MOV AL, ICW4
MOV DX, PIC+1       ;A0 =1
OUT DX, AL
MOV AL, OCW1
MOV DX, PIC+1       ;A0 =1
OUT DX, AL
MOV AL, OCW2
MOV DX, PIC         ;A0 =0
OUT DX, AL

RET
INITIALISE ENDP
  ;Interrupt service
  routines for IRQ0 -
IRQ7

ISR0:      ;Start of
  'Interrupt service
   routine'
           ;USER CODE
GOES HERE
IRET       ;Return from
   interrupt.
ISR1:      ;User
  programs to service
   Interrupts
IRET
ISR2:
IRET
ISR3:
IRET
ISR4:
IRET
ISR5:
IRET
ISR6:
IRET
ISR7:
IRET
ORG 07F0H
JMP FAR PTR BOOT
ORG 0800H
CODE ENDS
CODEBOOT SEGMENT AT
0FF80H
ASSUME CS:CODEBOOT
ORG 0000H
BOOT PROC FAR
BOOT ENDP
CODEBOOT ENDS
END
```

Since each interrupt service routine (ISR0 - ISR7) is located at address CS:IP, the vector table in RAM occupies four addresses for each pointer. Procedure VECTORS loads the CS:IP values for each interrupt type in successsive RAM locations starting at address 0080H. Procedure INITIALISE loads the 8258 control registers and initialises the PIC. Any hardware interrupts will now cause the relevant interrupt service routine to be activated.

Interrupts are ordered in priority from highest - IRQ0, to lowest -IRQ7. Thus, if two or more Interrupts occur simultaneously, the higher priority interrupt will be serviced. Operational control word OCW2 automatically rotates the priority of interrupts so that even if all eight IRQ lines are activated at the same time, all of them will be serviced in sequential order.

Programming the I/O port

The 8155 consists of two 8-bit ports (port A and port B) and an additional 6-bit port (port C). Port C port lines PC0 - PC3 are connected to the D0 - D3 pins of an intelligent HEX display (0 - F) for a nibble input of 0000 - 1111. Line PC5 controls the RH decimal point and line PC4 is used to blank (logic high) the display. All ports can be configured as outputs or inputs, but are not bit addressable. The port command register is located in IO space at address 0000, to which a command word is sent to configure the ports. Command register bit assignments are shown below. So, for example, to program ports C and B as outputs and port A as an input the following byte is written to the command register:

```
D7 D6 D5 D4 D3 D2 D1 D0
_ _ _ _ _ _ _ _
X  X  X  X  1  1  1  0
```

PORT_COMMAND REGISTER BIT ASSIGNMENT:

0000H	D7	D6	D5	D4	D3	D2	D1	D0
PORT C INPUT					0	0		
PORT C OUTPUT					1	1		
PORT B INPUT							0	
PORT B OUTOUT							1	
PORT A INPUT								0
PORT A OUTPUT								1

The following procedure will write the display the contents of the CL register (lower nibble) on the intelligent display

```
DISPLAY PROC NEAR
PORT_COMMAND             EQU 0000    ;COMMAND
                          register address
PORT_C                   EQU 0003    ;Port C
                          address
PORT_B                   EQU 0002    ;Port B
                          address
PORT_A                   EQU 0001    ;Port A
                          address
MOV DX, PORT_COMMAND
MOV AL, 00001100B        ;Port C output port
OUT DX, AL
MOV DX, PORT_C
MOV AL, 5
OUT DX, AL               ;Send 5 to hex display
RET
DISPLAY ENDP
```

The procedure can simply be called by instruction 'CALL DISPLAY'

Programming the 8088 board

The MASM assembler was used in programming examples. Since it does not produce romable code (pure binary code) which is sent to an EPROM emulator connected to the board's EPROM socket, a small batch file (8088 . BAT) needs to be run. The procedure is as follows:

*The source code is written using any ASCII (DOS) text editor such as 'EDIT' 'RPED' etc. The file is then saved as FILENAME. ASM, the ASM extension signifying that the file is text source code.

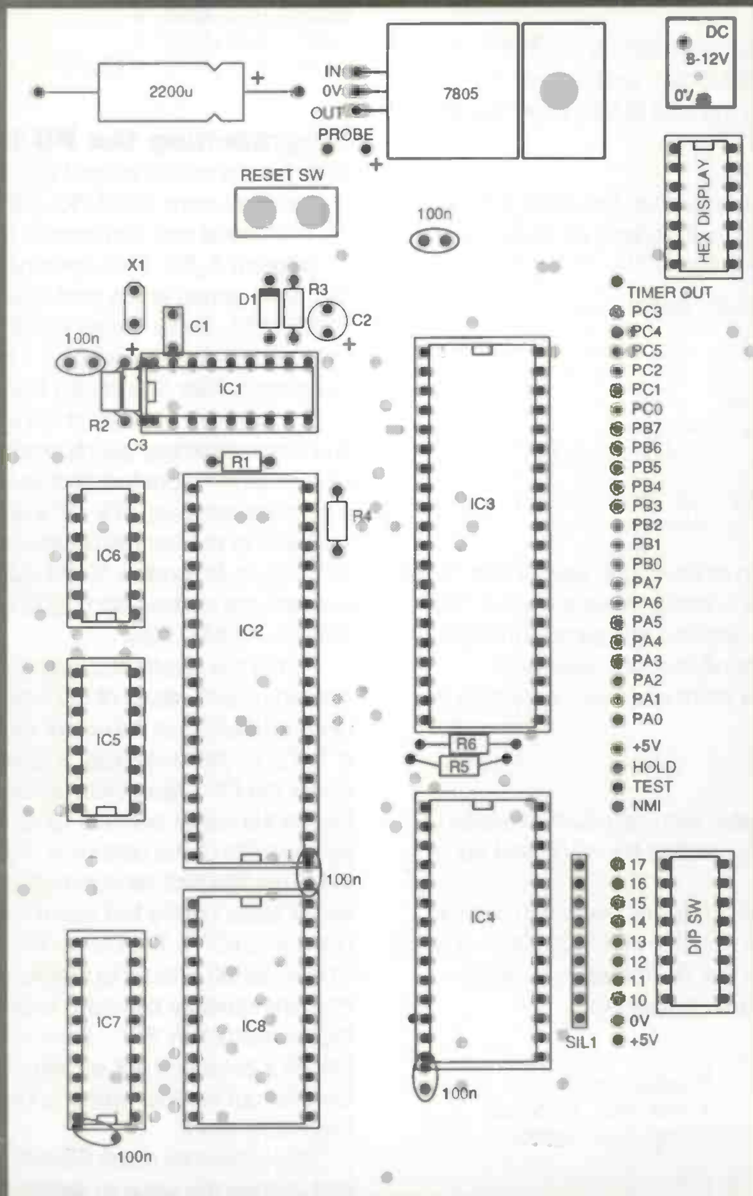


Fig.2. Component overlay for 8088 SBC

*The (8088. BAT) batch file is run, :I. E. 8088 FILENAME ENTER producing a listing (FILENAME. LST) and the resultant object code (FILENAME. COM) which is 2Kbyte in size.

*The code is then sent to the 8088 board using DOS command COPY FILENAME. COM LPT1 ENTER if the emulator is connected to the parallel printer port. If connected to the serial port (COM1), COPY FILENAME. COM COM1 ENTER will download the code via the serial link.

```
;8088. BAT
ECHO ASSEMBLE AND LINK %1
IF NOT EXIST %1. ASM GOTO END
MASM %1, %1, %1;
IF ERRORLEVEL 1 GOTO END
LINK %1, %1, %1;
IF NOT EXIST %1. EXE GOTO END
EXE2BIN %1. EXE %1. COM
ECHO OFF
ERASE %1. EXE
ERASE %1. OBJ
:END
```

Note. File Exe2bin. exe will be found in the DOS directory.

The timer

The 8155 contains a 14-bit countdown timer whose input at pin 3 is a square wave signal at 1. 8432 MHz from the 8284 clock generator. The output of the timer at pin 6 can be programmed in 1 of 4 modes:

*A single square wave
*Continous square wave
*A single pulse on terminal count
*Continuous pulses.

To program the timer IO address 0005 is used to configure the mode and to load the MSB (6 bits) count length. IO address 0004 is used to load the LSB (8 bits) count length. See diagram.

To program the timer for x N division, the count (N) value is loaded into the timer's count length register as shown below.

```
0005H M2    M1   T13   T12   T11   T10   T9   T8

      MODE      ---    MSB OF COUNT ---

M2  M1
-   -
0   0  Single square wave
0   1  Continous square wave
1   0  Single pulse on terminal count
1   1  Continous pulses
0004H     T7   T6   T5   T4   T3   T2   T1   T0
            ---      LSB OF COUNT     ---
```

Bits T0-T13 specify the length of the count, bits 14 and 15 will specify the timer mode. For a continuous square-wave M2 will be 0 and M1 will be 1. In addition, the command register at address location 0000 controls the timer start/stop sequence. Bits 6 and 7 of the command word determine the timer command as follows:

00 No operation
01 Stops the timer
10 stops the timer after terminal count is reached
11 starts the timer by loading the mode and count length.

The following program demonstrates the use of the timer in providing a frequency division of 3FFEH (16382). This will result in a square wave 1/16383 of the Tin frequency (1. 8432 Mhz/16382 =112 hz), at the Tout output pin.

```
TIMER PROC NEAR
 COMMAND EQU 0               ;COMMAND REGISTER ADDRESS
 TIMER_HIGH EQU 5            ;TIMER MSB ADDRESS
 TIMER_LOW EQU 4             ;TIMER LSB ADDRESS

 TIMER_COMMAND =11000000B;START COUNTER
 HIGH_COUNT =01111111B    ;MODE 01, MSB COUNT 3FH
      ;Continuous square wave
 LOW_COUNT =11111110B     ;LSB COUNT FEH

 MOV DX, TIMER_LOW           ;LOAD TIMER
 MOV AL, LOW_COUNT
 OUT DX, AL
 MOV DX, TIMER_HIGH
 MOV AL, HIGH_COUNT
 OUT DX, AL

 MOV DX, COMMAND
 MOV AL, TIMER_COMMAND       ;START TIMER
 OUT DX, AL
 RET
TIMER ENDP

8088 SBC MEMORY MAP:
FFFFF......,
FFFF0.......reset          2 K BYTE EPROM AREA
FF800.......

06000.......8259 PIC ADDRESS
06001.......

00FF
      STACK
      VECTOR TABLE         256 BYTES RAM
0000
```

```
I/O Addresses:
0005.......TIMER HIGH BYTE/MODE
0004.......TIMER LOW BYTE
0003.......PORT C
0002.......PORT B
0001.......PORT A
0000.......PORT COMMAND
```

Programming the PC for Interrupts.

While we are on the subject of interrupts, and the fact that the PC uses the same 8259 PIC, the following text describes how the PC's serial port can receive data on an interrupt basis.

Program S_INT. PAS demonstrates how incoming serial data is streamed to the hard disk in real time into file 'SAMPLES'. Since it is an ASCII text file it may easily be inported into any spreadsheet - for example into Lotus 123 using the familiar 'File Import Numbers' commands.

The program consists of the main PAscal program S_INT and three assembler subroutines which are first converted to OBJ code format before final compilation into an EXE file by the Pascal compiler. The PC's serial USART needs to be initialised to provide the necessary communications protocol of 9600 baud: Procedure 'COM. ASM', : the USART'S internal registers are enabled for RS232 communications by Procedure 'ENABLECOM1. ASM'.

Part of the initialisation process in the Pascal program is to write the CS:IP values of the interrupt service routine ('INTHANDLER') to the vector table (variable 'INTERVECTOR'). A TYPE 12 (0CH) interrupt is issued whenever a serial byte enters the PC's serial port. Since the PC's BIOS has already loaded the vector pointers for all interrupts into the interrupt pointer table at the bottom of memory (0000:0000 onwards), the users interrupt service routine address is loaded into the vector table, having first saved the original pointer in variable ('ORGVECTOR'). The OCW1 PIC register - PC's port address 21H is loaded with 00 to enable all interrupts. Once the interrupt structure has been enabled by setting the interrupt flag by instruction 'STI' - inline code FAH, a serial byte entering the PC's serial port will activate the COM1 interrupt and cause the interrupt service routine to be executed: in this case Procedure RSINT.

This procedure reads the byte received by the PC's USART and assigns this value to variable 'HEX' by 'passing it over the stack' to the Pascal environment. A sample number associated with each sample increments for every byte and is also written to text file 'SAMPLES'. Finally, if the ENTER key is pressed, the original pointer is written into the vector table and the program aborts.

```
PROGRAM S_INT(INPUT, OUTPUT);

{$f+}
{$I-}
{$L COM}                  (*External . obj file*)
{$L ENABLECOM1}          (*External . obj file*)
{$L RSINT}                (*External . obj file*)
USES dos, crt;
TYPE
DIGITS=0 . . 255;          (*data 00 - FF*)
VAR
ORGVECTOR:POINTER;   (*Pointer in interrupt table
                         for serial*)
                       (*interrupt*)
INTERVECTOR :ARRAY [$00. . $FF] of POINTER
ABSOLUTE $0000:$0000;
NUMBER:INTEGER;          (*Sample number*)
SAMPLES:TEXT;            (*ASCII text file*)
HEX:DIGITS;              (*serial data variable*)
PROCEDURE RSINT(VAR HEX:DIGITS);EXTERNAL;
PROCEDURE COM;EXTERNAL;
PROCEDURE ENABLECOM1;EXTERNAL;
```

```
PROCEDURE MESSAGE;
BEGIN
CLRSCR;
WRITELN('Ready for data');
WRITE('Press enter key to exit to DOS');
END;


(*Main data acquisition program*)
 (*Interrupt service routine*)


PROCEDURE INTHANDLER;INTERRUPT;
BEGIN
RSINT(HEX);                  (*Fetch data*)
WRITE(SAMPLES, HEX:5);       (*Write ASCII
                                  data + sample*)
                                 (*No. *)
WRITELN(SAMPLES, '   ', NUMBER);        (*to hard
disc*)
NUMBER:=NUMBER+1;            (*increment sample
number*)
PORT[$20]:=$20;              (*issue end ofinterrupt*)
                             (*command to PIC*)
END;


                             (*Initialise variable*)
BEGIN
ASSIGN (SAMPLES, 'SAMPLES');  (*Open textfile
'samples'*)
REWRITE (SAMPLES);
NUMBER:=1;                    (*Initialise sample No.
                               to 1*)
ORGVECTOR:=INTERVECTOR [$c];  (*get interrupt
                                  vector for*)
                             (*COM1, ($b COM2)*)
INTERVECTOR[$c]:=@ inthandler;    (*Place CS:IP
                                 offset of*)
                             (*interrupt service*)
                             (*into vector table*)
COM;                         (*Define RS232 protocol*)
ENABLECOM1;                  (*Initialise USART
registers*)
PORT[$21]:=0;                (*Enable all IRQ
interrupts*)
                             (*keyboard, video,
                              harddisc and*)
 welcome datacomp(*serial port*)
MESSAGE;
REPEAT                       (*Wait for serial
interrupt*)
UNTIL KEYPRESSED;            (*Exit to DOS if enter
key*)
(*pressed*)
INLINE($FA);
CLOSE(SAMPLES);              (*Close data file*)
INTERVECTOR [$c]:=ORGVECTOR;  (*Restore original
interrupt
                             (*vector*)
END.


;Initilise COM1 (COM2) port for 9600 baud, 1 stop
bit, 8 data
bits, no parity.
CODE     SEGMENT BYTE PUBLIC
          ASSUME CS:CODE
PUBLIC  COM
   COM  PROC FAR
         PUSH BP
         MOV BP, SP
;Main  Program begins


MOV AL, 0E3H ;9600 baud
MOV AH, 0
MOV DX, 0     ;com2=1
             ;com1=0
INT 014H     ;BIOS interrupt

;Main program ends
```

```
             POP BP
             RET  2
      COM    ENDP
CODE         ENDS
             END


;Initialise USART registers
CODE     SEGMENT BYTE PUBLIC
          ASSUME CS:CODE
PUBLIC  ENABLECOM1
ENABLECOM1    PROC FAR


          PUSH BP
          MOV BP, SP
          MOV DX, 03F9H           ;Enable Rxd bit in
control register
          MOV AL, 1               ;02F9H = COM2
          OUT DX, AL
          MOV DX, 03FCH           ;Enable USART
          MOV AL, 0BH             ;02FCH = COM2
          OUT DX, AL
          POP BP
          RET  2
ENABLECOM1    ENDP

CODE     ENDS
          END


;Fetch serial data byte and pass to Pascal
variable HEX
CODE    SEGMENT BYTE PUBLIC
ASSUME CS:CODE
PUBLIC RSINT
RSINT   PROC FAR
          CLI


          PUSH BP         ;Save BP contents
          MOV BP, SP       ;Copy stack pointer to
base pointer
          MOV DX, 03FDH  ;Usart base address (COM1)
02FDH=COM2


          LDS SI, [BP+6] ;Point SI to stack
location for data

SUB DX, +5                ;Usart data buffer address
IN AL, DX                 ;Fetch data
         MOV BYTE PTR [SI], AL ;Place data in stack
         POP BP                ;readjust stack
pointer


         STI
         RET 4                 ;and return
RSINT   ENDP
CODE    ENDS
         END
```

A floppy disk is available from the author which contains 'Boardmaker' PCB files for both the 80188 and 8088 SBC's, A86 assembler and source code example software. Available from:

Mr. R. Grodzik (Micros)
53 Chelmsford Road
Bradford BD3 8QN
West Yorkshire.

Price £12. 50 P&P inc. welcome datacomp

Practically SPEAKING

BY TERRY BALBIRNIE

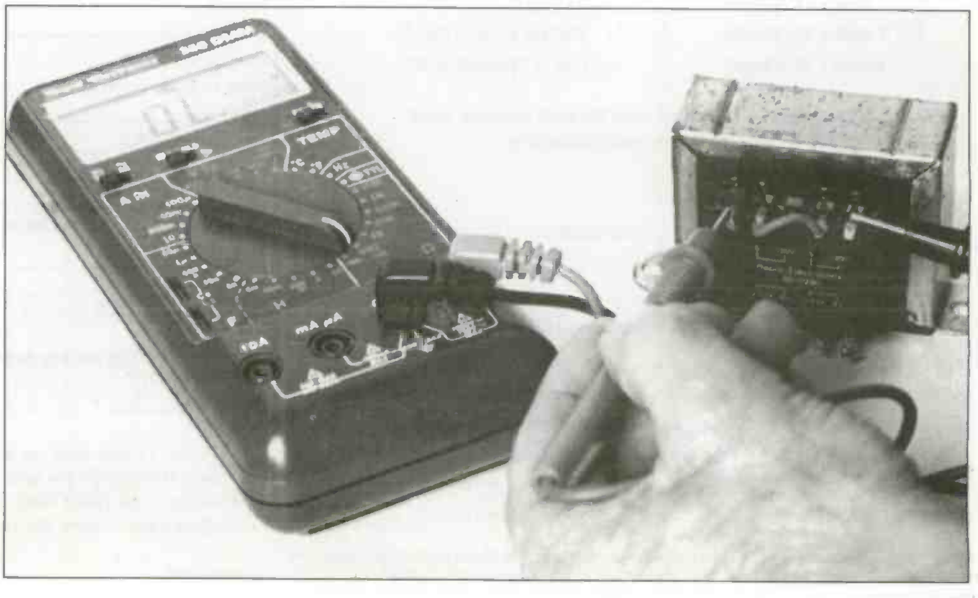
Last month, we looked at repairs to mains-operated appliances found around the home. This time we shall look at electronic equipment where a low-voltage supply is derived from the mains using a step down transformer

For some tests, the equipment will need to have its inner circuitry exposed and for it to be plugged in to the mains. It is therefore essential to shield any unprotected mains connections so that it is impossible to touch them. If in doubt about your ability to do this safely, confine yourself to those tests which may be carried out with the equipment isolated from the mains.

Most tests involve using a multitester set to either a low resistance range (or buzz test) or to voltage. Current measurements must be taken with great care and only with the necessary knowledge to interpret the readings. One of the worst things which can be done with a multitester is to leave it set to a high (often unfused) current range and to apply the probes direct to the mains. An enormous current will flow, possibly ruining the meter and burning out the test probes. For this reason, it is best to store a multitester set to a high voltage range.

Regard any voltage over 20V as potentially dangerous. With damp hands, it is possible for this to send a lethal current through the body. Remember, it is also possible to receive a dangerous shock from a capacitor which is charged up to any voltage greater than this. Certain pieces of equipment - notably photoflash units - charge a capacitor to several hundred volts using a battery-operated circuit. This type of equipment needs special care since it can deliver a lethal shock.

As mentioned in last month's Practically Speaking, blown fuses are probably the most common reason for a piece of equipment. failing to work. Fuses should be removed to test them and the multitester used on a low-resistance range. A good fuse will indicate almost zero resistance. Inspect the fuse in the standard UK mains plug first, then check other fuses inside the equipment or those in panel fuseholders. Do not assume that a blown fuse is the source of trouble. It often indicates a fault elsewhere and this must be investigated.



Practical example

One common source of trouble is the plug-in power supply used for rechargeable tools, laptop computers, games, cordless telephones and similar devices. These are not treated well by their owners. The plug at the equipment end is often removed by pulling on the wire itself. This causes the output wire to snap or break from its soldered connections. The wire is often twisted mercilessly and the device often knocked harshly or dropped on the floor.

Damaged output plugs are easily replaced and joints re-soldered. With the appliance unplugged from the mains, check both input and output wires for continuity by applying the probes between the ends of each. A wire often breaks near one end where most bending occurs. By cutting off the last few centimetres, it may be re-connected giving the unit a new lease of life. If the break is near the middle of the wire, it should be replaced with a length of the same type. If the plastic case is cracked or the transformer is rattling around loose, the unit must be discarded. Never tape over cracks and splits or attempt to stick components down inside.

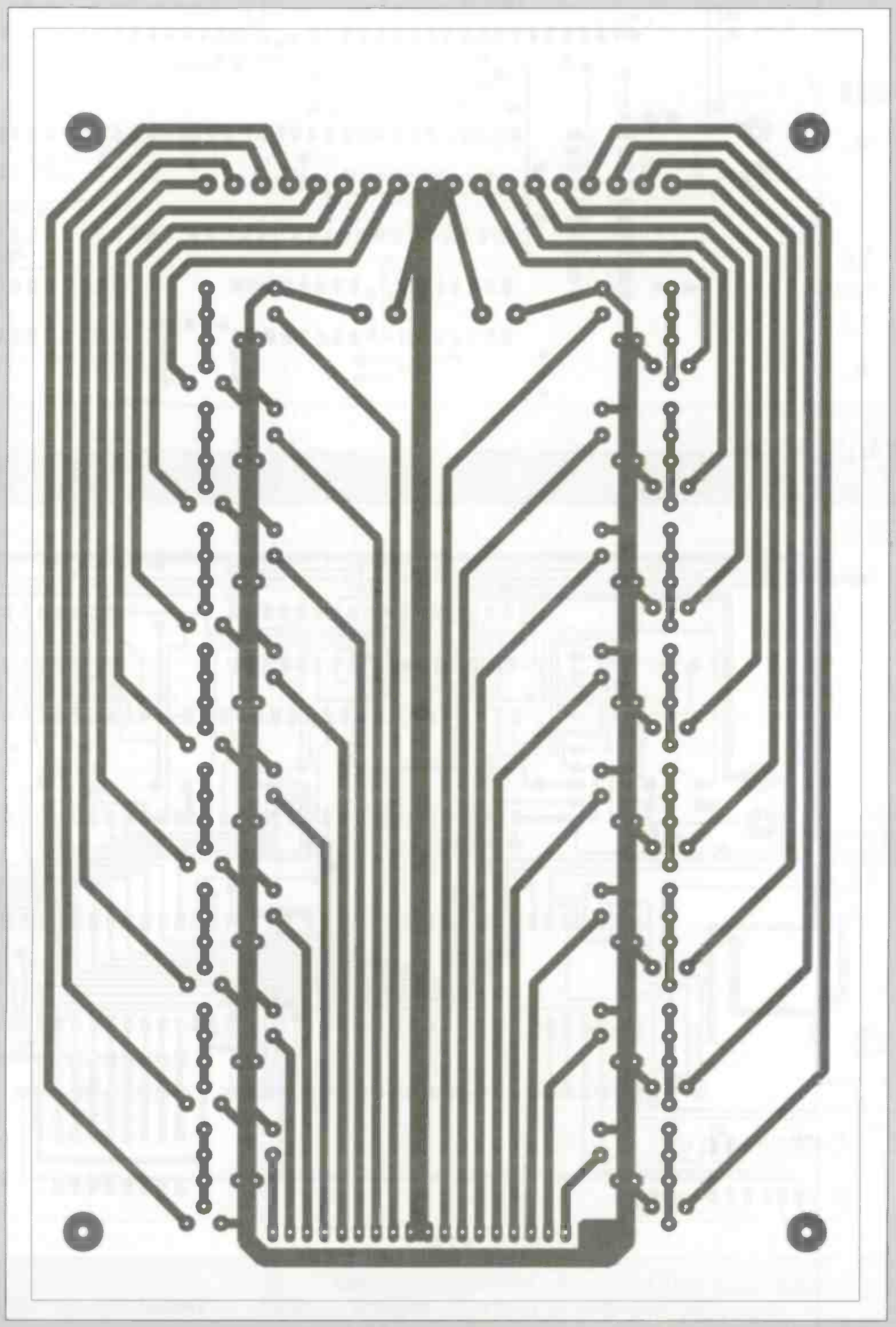
Often power supplies have a "polarity reversal" plug and socket somewhere along the length of the output wire. This is often pulled out by the owner and re-connected in the opposite sense. The equipment fails to work and it is reported as faulty when nothing else is amiss.

If a fault persists after checking wires and fuses for continuity, test the switch if there is one. To do this, remove the wires from it and check using the multitester set to a low resistance range. It should show near-zero resistance when on. If there is a step-down transformer, check that an a.c. voltage exists across the transformer secondary winding. If it does not, check that mains voltage exists at the primary. This must be done very carefully indeed using the multitester set to an a.c. range greater than 240V. If mains voltage exists at the primary but there is no output at the secondary, the transformer has failed. Depending on the nature of the equipment, it may not be worthwhile replacing it. If no mains supply exists at the primary, the fault probably still lies in the input lead, a fuse, connector or switch, so check more carefully.

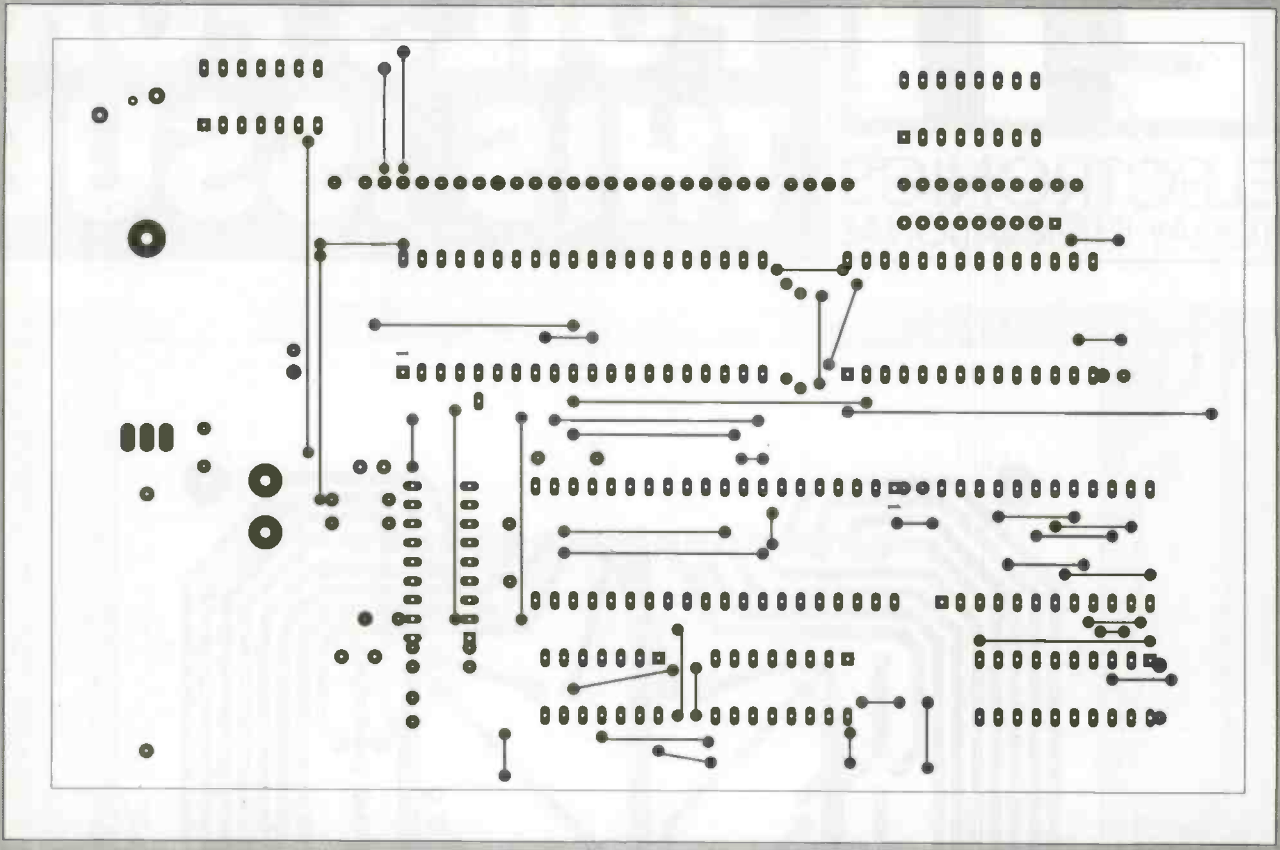
If the fault persists, look for obvious signs of trouble such as overheating. However, in the absence of a circuit diagram and specialist knowledge, it may be as well to leave the repair to the professionals.

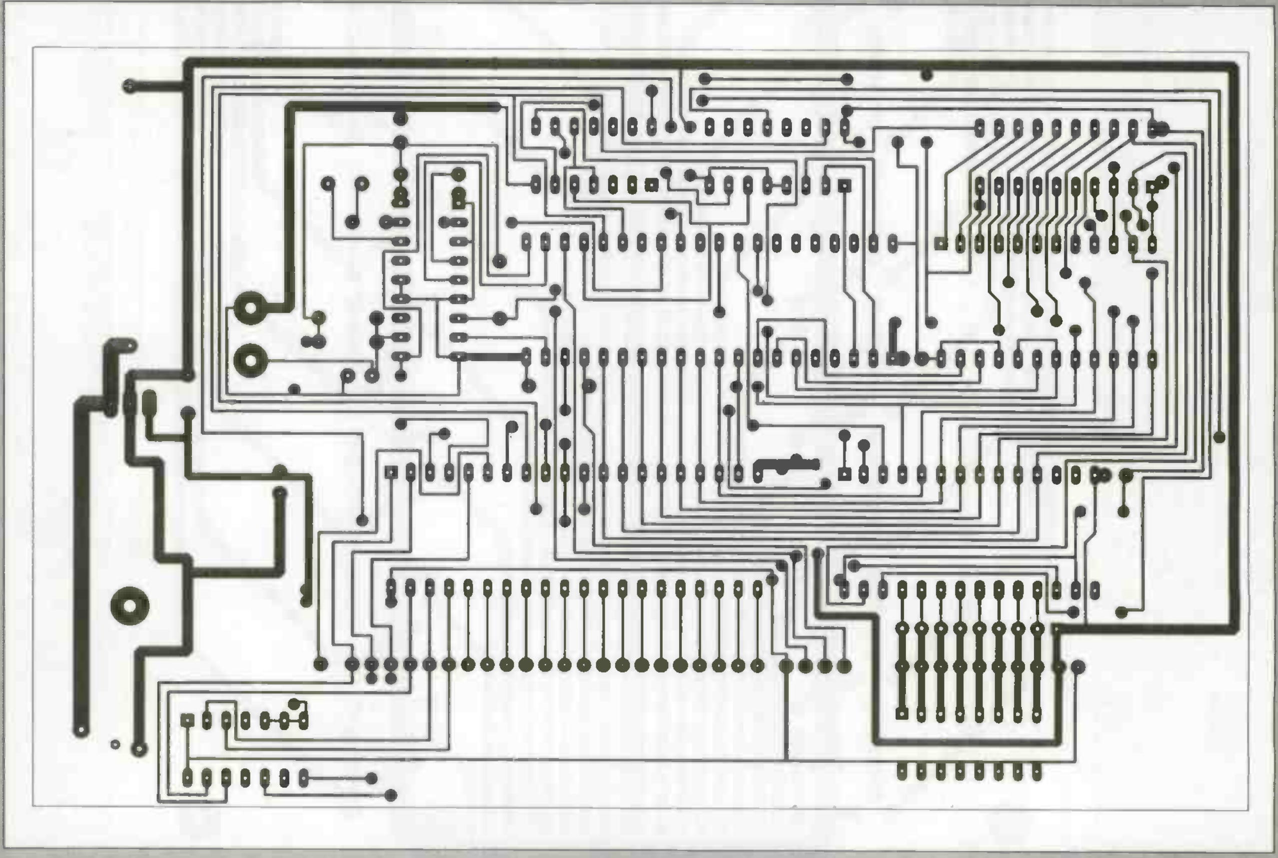
ADVERTISERS INDEX

ETI ELECTRONICS TODAY INTERNATIONAL FOILS FOR THIS ISSUE

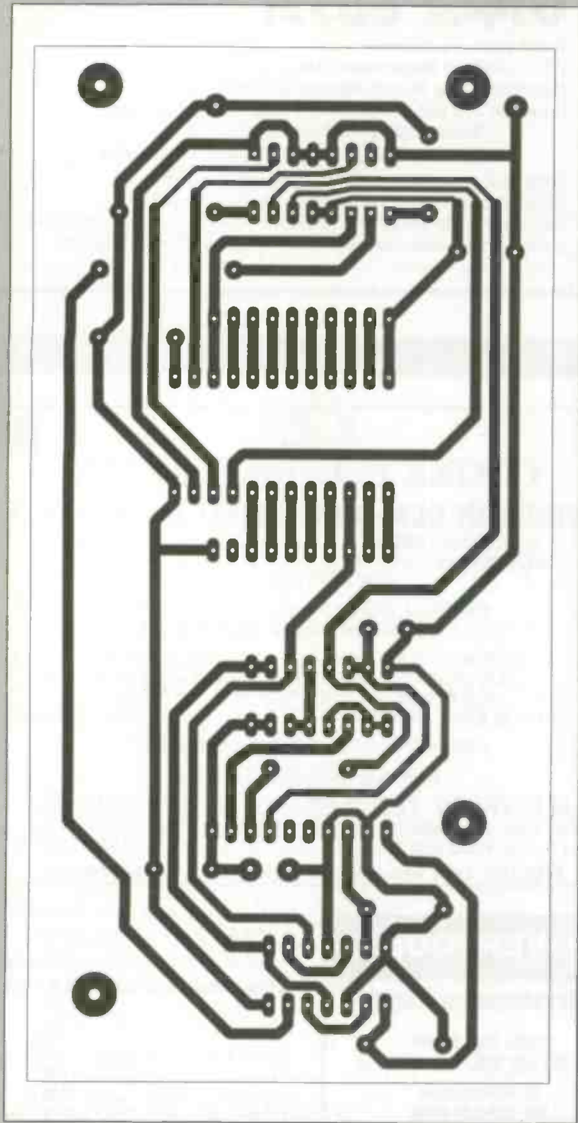


HELLO MUM DRIVE BOARD

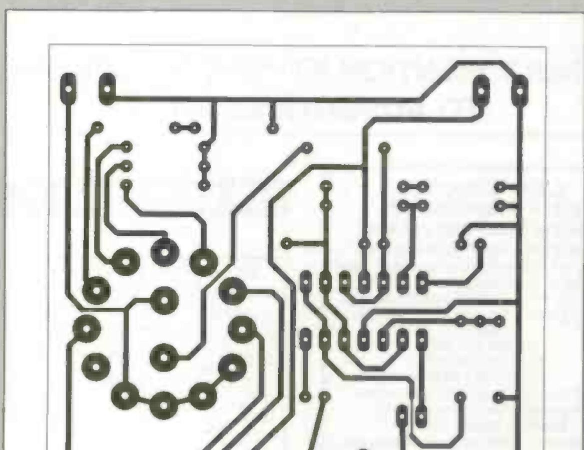




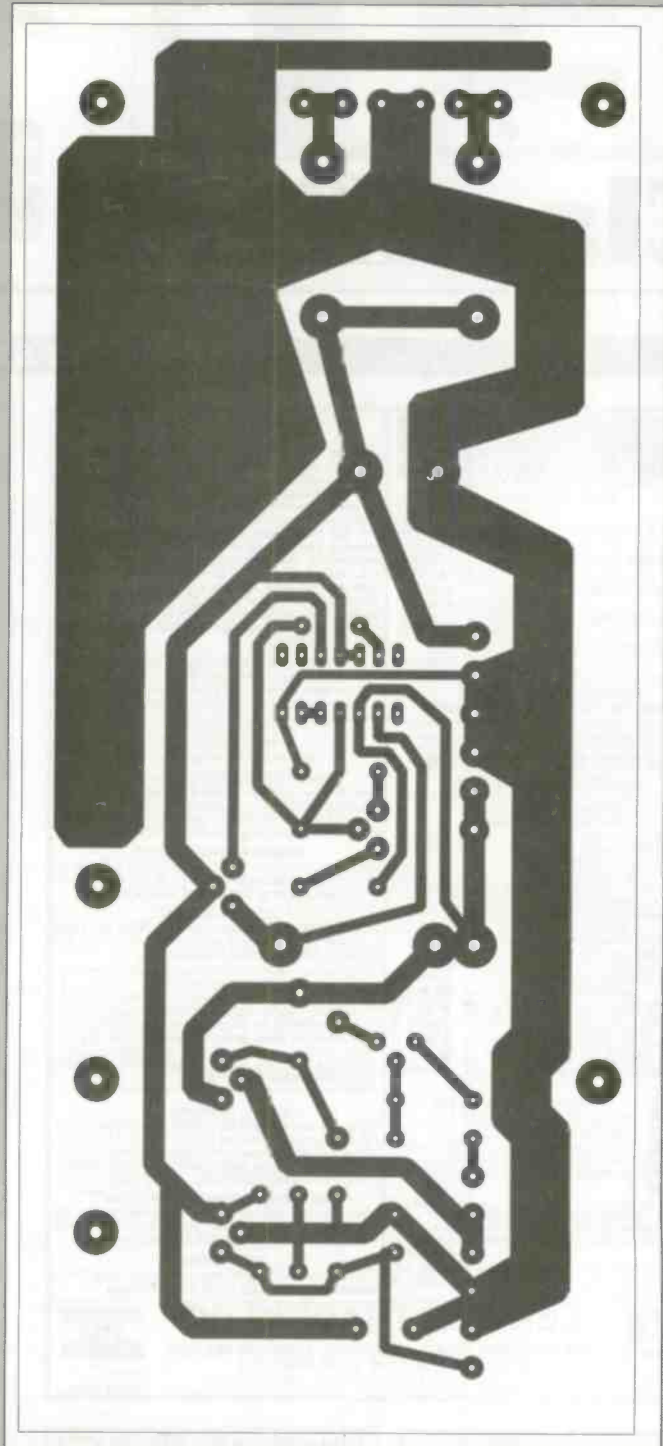
8088 Interrupt based control computer



HELLO MUM CONTROL BOARD



DRY BATTERY TESTER



HELLO MUM CONTROL BOARD

ETI Classified

James Gale
01442 66551

Send your requirements to:
ETI Classified Department, Nexus, Nexus House, Boundary Way, Hemel Hempstead, HP2 7ST
Lineage: 75p per word (+ VAT) (minimum 15 words)
Semi display: (minimum 2.5cms)
£10.50 + VAT per single column centimetre
Ring for information on series bookings/discounts.
All advertisements in this section must be pre-paid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

VISA Access

FOR SALE

VARIABLE VOLTAGE TRANSFORMERS

INPUT 220/240V AC 50/60
OUTPUT 0-260V

	Price	P&P
0.5KVA 2.5 amp max	£31.90	£6.00
	(£44.53 inc VAT)	
1KVA 5 amp max	£41.15	£7.00
	(£56.58 inc VAT)	
2KVA 10 amp max	£59.40	£8.50
	(£79.78 inc VAT)	
3KVA 15 amp max	£78.65	£8.50
	(£102.40 inc VAT)	
5KVA 25 amp max	£139.15	
	(Plus Carriage)	

Buy direct from the Importers. Keenest prices in the country

COMPREHENSIVE RANGE OF TRANSFORMERS-LT-ISOLATION & AUTO

(110-240V Auto transfer either cased with American socket and mains lead of open frame type. Available for immediate delivery.

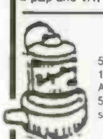
WIDE RANGE OF XENON FLASHTUBES
Write/Phone your enquires

ULTRA VIOLET BLACK LIGHT FLUORESCENT TUBES

4ft 40 watt £14.00 (callers only) (£16.45 inc VAT)
2ft 20 watt £9.00 (callers only) (£10.58 inc VAT)
12in 8 watt £4.80 + 75p p&p (£6.52 inc VAT
9in 8 watt £3.96 + 50p p&p (£5.24 inc VAT)
6in 4 watt £3.96 + 50p p&p (£5.24 inc VAT)

230V AC BALLAST KIT
For either 6in, 9in or 12in tubes £6.05 + £1.40 p&p (£8.75 inc VAT)
The above Tubes are suitable for Forged Bank Note detection, security marking etc.
Other Wave Lengths of U.V. TUBE available please telephone your enquines

400 WATT UV LAMP
Only £38.00 + £4.00 p&p (£49.35 inc VAT)
160 WATT SELF BALLASTED BLACK LIGHT MERCURY BULB
Available with BC or ES fitting, Price inc VAT & p&p and VAT £25.55



12V D.C. BILGE PUMPS
500 GPH 15ft head 3 amp £19.98
1750 GPH 15ft head 9 amp £34.55
Also now available 24V D.C. 1750 GPH 15ft head 5 amp £35.55. All designed to be used submerged. PRICES INCLUDE P&P & VAT

SUPER HY-LIGHT STROBE KIT
Desiged for Disco, Theatrical uses etc.
Approx 16 joules. Adjustable speed £50.00 + £3.00 p&p (62.28 inc VAT)
Case and reflector £24.00 + £3.00 p&p (£31.73 inc VAT)
SAE for further details including Hy-Light and Industrial Strobe Kits.

"BOFFINS SPECIAL"- UNIQUE OFFER
Surplus Precision Medical Unit, Internally in excellent condition. Designed primarily to eject a precise controllable amount of fluid from a medical syringe (latter not supplied). Contains the following removable components: Dual Micro Processor Boards and EPROMS Escap Precision 12V DC Motor with 300:1 Gear Box and optical encoder coupled to a precision threaded drive mechanism. Mains supply with 6 x 1.5V Ni-Cad A.A. cells back-up. L.C.D. Digital read-out 17mm high with legends. Audible warning.
These are sold for the dismantling of the exceptional quality components, regret no Circuits available.
Ridiculously low price: £20.00 + £4.00 p&p (£28.20 incl VAT)

5KVA ISOLATION TRANSFORMER
As New Ex-equipment fully shrouded Line Noise Suppression, Ultra Isolation Transformer with terminal covers and Knock-out cable entnes. Pnmary 120/240V Secondary 120/240V 50/60 Hz. .005 pF Capacitance. Size L.37 x W.19 x H.16cm Weight 42 Kilos
PRICE £120.00 +VAT
ex-warehouse. Carnage on request

240V AC CENTRIFUGAL BLOWER
New Manuf. Surplus Skeleton Blower suitable for mounting inside an enclosure to cool equipment. Overall size 130x110x85mm. Outlet 80x35mm. Impellor 60mm dia x 60mm long. £14.10 Incl.P&P & VAT

SEWING MACHINE MOTOR
Brand new 220/240v AC/DC SEW-TRIC 2 lead Brush Motor. Size L. 100mm x H. 70mm x W.55mm. Spindle. 1/4in. dia x 1in. long. £14.10 incl. P&P & VAT

GEARED MOTORS
71 RPM 20lb inch torque reversable 115V AC input including capacitor and transformer for 240V AC operation. Price inc VAT & p&p £27.73

SOLID STATE EHT UNIT
Input 230/240V AC, Output approx 15KV. Producing 10mm spark. Built-in 10 sec timer. Easily modified for 20sec, 30 sec to continuous. Designed for boiler ignition. Dozens of uses in the field of physics and electronics, eg supplying neon or argon tubes etc. Price less case £8.50 + £2.40 p&p (£12.81 inc VAT) NMS

EPROM ERASURE KIT
Build your own EPROM ERASURE for a fraction of the price of a made-up unit kit of parts less case includes 12in 8 watt 2537 Angst Tube Ballast unit, pair of bi-pin leads, neon indicator, on/off switch, safety microswitch and circuit £15.00 + £2.00 p&p (£19.98 inc VAT)

WASHING MACHINE WATER PUMP
Brand new 240V AC. fan cooled. Can be used for a vanety of purposes. Inlet 1¼in. outlet 1 in. dia. Pnce includes p&p & VAT. £11.20 each or 2 for £20.50 inclusive

MICROSWITCH
Pye 15 amp changeover lever microswitch, type S171. Brand new price 5 for £7.05 inc VAT & p&p

SERVICE TRADING CO
57 BRIDGMAN ROAD, CHISWICK, LONDON W4 5BB
FAX 0181-995 0549 0181-995 1560
ACCOUNT CUSTOMERS MIN, ORDER £10
Showroom open Monday/Friday
Ample Parking Space

VISA ACCESS

COOKE INTERNATIONAL

SUPPLIER OF QUALITY USED TEST INSTRUMENTS,

ANALYSERS, BRIDGES, CALIBRATORS, VOLTMETERS, GENERATORS, OSCILLOSCOPES, POWER METERS. ETC
ALWAYS AVAILABLE

ORIGINAL SERVICE MANUALS FOR SALE
COPY SERVICE ALSO AVAILABLE

EXPORT, TRADE AND U.K. ENQUIRIES WELCOME
SEND FOR LIST OF EQUIPMENT AND MANUALS
ALL PRICES EXCLUDE VAT AND CARRIAGE
DISCOUNT FOR BULK ORDERS SHIPPING ARRANGED

OPEN MONDAY TO FRIDAY 9AM - 5PM

COOKE INTERNATIONAL
ELECTRONIC TEST & MEASURING INSTRUMENTS
Unit Four, Fordingbridge Site, Main Road, Barnham, Bognor Regis, West Sussex, PO22 0EB Tel (+44) 01243 545111/2 Fax (+44) 01243 542457
WANTED TEST EQUIPMENT & MANUALS TO PURCHASE

LIVERPOOL

PROGRESSIVE RADIO

87/93 Dale Street
Tel: 051 236 0982 051 236 0154

47 Whitechapel
Tel: 051 236 5489
Liverpool 2

'THE ELECTRONICS SPECIALISTS'
Open: Tues-Sat 9.30-5.30

SWC SCIENTIFIC WIRE COMPANY

ENAMELLED COPPER WIRE
TINNED WIRE SILVER
PLATED COPPER WIRE
SOLDER EUREKA WIRE
NICKEL CHROME WIRE
BRASS WIRE LI TZ WIRE
BIFILAR WIRE MANGANIN
WIRE TEFZEL WIRE NICKEL
SAE BRINGS LIST 18 RAVEN RD LONDON E18 1HW
FAX 081 559 1114

KINDLY MENTION ETI WHEN REPLYING TO ADVERTISEMENTS

TURN YOUR SURPLUS TRANSISTORS, ICS ETC INTO CASH immediate settlement.
We also welcome the opportunity to quote for complete factory clearance
Contact:
COLES-HARDING & CO
Unit 58, Queens Road, Wisbech, Cambs PE13 7PQ
BUYERS OF SURPLUS INVENTORY ESTABLISHED OVER 20 YEARS
Tel: 01945 584188 Fax: 01945 475216

SMART CARD PRODUCTS
Smartcards, Readers/Writers, OEM couplers, Evaluation kits and more.
EPSILON ELECTRONICS
Brynsengvn. 1A, 0667 Oslo, Norway TEL/FAX +4722640810

LEN COOKE ENTERPRISES
For the best value in Used Electronic Test Instruments
We buy, sell and service oscilloscopes, signal generators, frequency counters, spectrum Analysers, Power meters, logic testers, etc. Spare parts available for most Textronic scopes.
Tel: 081-813-9946
Fax: 081-574-2339
Mobile: 0374 759984
Mail order address: Unit 5, Southall Enterprise Centre, Bridge Road, Southall, Middx. UB2 4AI
We engineer what we buy, we support what we sell.

SOFTWARE

PC SCIENTIFIC & TECHNICAL PROGRAM LIBRARY
Low cost specialist software and
GOOD QUALITY PROGRAMS THAT WORK
are our forte.
We have the largest scientific and technical software library as well as the best mainstream programs available anywhere.
Software supplied on CD ROM or Floppy.
Discover the true gems of shareware with our 250,000 word book "The Encyclopedia of Shareware". Over 3000 programs listed, and described in 178 different categories.
For your copy send £2.50 or phone/fax your order. Major credit cards accepted. You also receive a money saving special offer voucher.
PDSL, Winscombe House, Beacon Rd, Crowborough Sussex TN6 1UL Tel 01892 663298, Fax 01892 667473

THIS SPACE COULD BE WORKING FOR YOU! FOR DETAILS OF ADVERTISING RATES CALL JAMES ON 01442 66551 OR FAX YOUR ADVERTISEMENT ON 01442 66998

ASTRA *Desk Top Accounting*
Overture - Invoicing, Sales, Purchases, Nominal, VAT, etc
Premier - plus Stock, Jobs, EPOS, Barcodes, Serial No, etc
Professional - plus Multi-Currency/Depts/Location, etc
S&S Systems Ltd, Bretton Court, Manor Road, Wales Sheffield, S31 8PD, UK. (Software distributors and Apple dealers)
Tel: (01909) 773399 • Fax: (01909) 773645 • Int Code: (+44 1909)

COMPONENTS

EPROMS PLDs + MICRO CONTROLLERS copied or programmed. We supply devices/convert discrete logic to PLDs. PLDs for ETI PC Post Card supplied at £10/P.C.B. £12. PO Box 1561, Bath 01225 444467.

BOOKS & PUBLICATIONS

THE BUG CONSTRUCTOR'S MANUAL
All you will ever need to know about Bugs. Over 40 Bug circuits for every application. Telephone, High Power, Crystal Controlled and UHF bugs and many pages of useful information. Many previously unavailable circuits. This is the very latest book, and the "ONLY" book of its kind in the UK. A valuable asset for the hobbyist and professional. Introductory offer price only **£9.95** inc. P&P.
For immediate despatch, send cheque or P.O.'s payable to:
GAINSFORD ELECTRONICS
71 Gainsford Road, Southampton SO19 7AW.

PRINTED CIRCUIT SERVICES

For the VERY best in:-
:Photoplotting
:Printed Circuit Design
:Screen Printing Metal Panels
:Graphic Reproduction
It's GOOD to talk to....
R P PHOTOGRAPHIC
tel: 0181 - 777 4407

PLANS

ELECTRONIC PLANS, laser designs, solar and wind generators, high voltage teslas, surveillance devices, pyrotechnics and computer graphics tablet. 150 projects. For catalogue, SAE to Plancentre Publications, Unit 7, Old Wharf Industrial Estate, Dymock Road, Ledbury, Herefordshire, HR8 2HS.

DIY PLANS!! Windgenerators, petrol generators, Inverters, portable welding generators, engine crane, electric bike, solar projects, boats and the famous Homebuilt Dynamo. Plus books on sterling engines, mechanical wonders, the writings of Thomas Tesla, electromagnets etc Hundreds of exciting projects. 2x1st stamps for descriptive catalogue. Jemmett Engineering 8 Hallam Gardens Pinner Middlesex HA54PR.

PRINTED CIRCUIT BOARDS

PCB's, PROTOTYPES AND PRODUCTION, Artwork copied or raised from your draft designs or schematics. Quoteline 01232 473533 or write to P Agar, 36 Woodcot Avenue, Belfast, BT5 5JA

FAX YOUR ADVERT ON 01442 66998

COURSES

Start training now with the specialists for the following courses. Send for our brochure — without obligation or Telephone us on 01626 779398 ETI 9/95

Name ______________________

____

____

____

☐ Telecomms Tech C&G 2710
☐ Radio Amateur Licence C&G
☐ Micro-processor
☐ Introduction to Television

Radio & Telecommunications Correspondence School
12 Moor View Drive, Teignmouth, Devon TQ14 9UN

Commercial Reality

EUREKA!!!
See your ideas come to life using this exciting home study course.
For free details write to:
Commercial Reality (Ref ET9), Triteq
St. Mary's House, 40 London Road,
Newbury, Berks. RG14 1LA
or Telephone: 01635 873775 (24hrs)

READERS PLEASE NOTE that some telephone dialling codes may recently have changed. If you experience any difficulty in contacting advertisers, ring our advertising department on 01442 66551

ETI

ELECTRONICS TODAY INTERNATIONAL, CLASSIFIED ADVERTISEMENT DEPARTMENT, NEXUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST

Rates: Lineage 75p per word + VAT minimum 15 words.
Semi-display £10.50 per single column cm plus VAT. No reimbursements for cancellations.
All ads must be pre-paid.

Name ....

Address ....

.... Daytime Tel. No: ....

Signature .... Date ....

PLEASE DEBIT MY ACCESS/BARCLAYCARD No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiry Date ....

FOR SALE **COMPONENTS** **PLANS** **OTHER – PLEASE STATE**

Open Forum

This month in Open Forum we shall not be looking at the state of the electronics industry in general but at the state of hobby electronics, and in particular the role that ETI plays.

I am sure that every reader is acutely aware that the whole nature of hobby electronics is changing very rapidly, a change that is mirroring the equally rapid changes in the electronics industry as a whole. The main driving force behind this change is that circuitry is being replaced by software, where there was once a board full of ICs and discrete components there is now a single microcontroller chip and perhaps a half dozen other components.

In industry this has enormous advantages in terms of reduced costs, ease of updating designs, and greatly improved reliability. But it also means that the circuits used in different applications are all starting to look remarkably similar, the difference between them being in the software rather than the hardware.

The trend towards embodying the functionality of a design in software rather than in hardware is something which a magazine like ETI cannot ignore. We live in the real world, and in that world people no longer build crystal sets, or construct devices using simple logic gates, masses of transistors etc. Today we live in a world where electronics equals microcontrollers, programmable logic arrays, and highly integrated special purpose ICs.

For the reader of ETI this trend has two effects; on the one hand it actually makes it easier to build more complex applications, but on the other it removes a lot of the electronics from a project and replaces it with computer programming. Hence the fact that magazine is increasingly filled with pages of code listings.

I personally think that we are at a transition point, where the electronics enthusiast, in the old sense, is disappearing and being replaced by the enthusiast who designs and builds electro-computing systems. Someone who is as heavily into computing as electronics, and who sees electronics as simply a means of practically implementing what is primarily a software design.

With the actual circuitry becoming more and more just a matter of plugging together standard building blocks (such as the PIC based Basic programmable microcontroller system), the function of a magazine like ETI has to be one of providing those building blocks, and also providing a source of ideas on how they can be used for inventive and useful applications.

I could well be wrong, but it is important for ETI, for the hobby electronics industry, and for you the readers, that we decide exactly where we are all going. One interesting pointer to the future is the success of the new BBC TV programme Eureka. It is evidence of an increasingly popular interest in invention and engineering, particularly where they can be applied to the solution of everyday problems. The enthusiast is no longer viewed as being rather strange, but as a pioneer, a builder of the future.

So I would like to finish off this piece with an appeal; if you are working on an unusual idea, or an invention which you think would be of interest to other readers of ETI, be it an electronically controlled home, a back yard spaceship or a thought controlled computer, why not let us know?. Who knows, it could be the start of something big.

Next Month...

In the October 1995 issue of Electronics Today International we will be featuring an interesting range of new projects. The main project in this issue is the ETI Basic programmable microcontroller

Amongst the other projects in this issue, there is one from Terry Balbirnie that will interest all parents of young children - a low intensity child's night light with micropower consumption. For all music buffs there is a 16 channel MIDI mixer from Tom Scarff.

Dr Pei An shows how to build a computerised radio control system that will allow a PC to remotely control any device within a range of several hundred yards. From David Geary there is a timed isolator system which should prevent mains equipment being accidentally left on. From Tim Parker there is a versatile upgradable bench power supply, and Bart Trepak continues his practical look at designing a project around the PIC microcontroller.

In the main feature article in next month's ETI, Dave Clarkeson takes a look at the technology behind some of the current advances in electron microscopy. Terry Balbirnie takes a practical look at the use of solar panels.

ETI

ELECTRONICS TODAY INTERNATIONAL

EDITORIAL

Editor ***Nick Hampshire***
Sub Editor ***Eamonn Percival***
Editorial Assistant ***Wendy Butler***

CREATIVE

Designer ***Nadia Ahdout***
Technical Illustration ***John Puczynski***
Photography ***Manny Cefai***

ADVERTISEMENT SALES

Display Sales
James Gale
Advertisement Copy Control
Marie Quilter

MANAGEMENT

Divisional Director
Terry Pattisson
Production Administrator
Theresa Davis
Group Editor
Stuart Cooke
Group Advertisement Manager
Claire Jenkinson
Markrting Managmer
Jason Doran
Copy Sales Manager
David Pegendam

ABC CONSUMER PRESS

ISSN 0142-7229

Subscription rates-UK £25.80 Europe £34.70 Sterling Overseas £36.20 US Dollars Overseas $54.00

Published by Nexus Special Interests, Nexus House, Boundary Way, Hemel Hempstead HP2 7ST. Telephone (01442) 66551. UK newstrade distribution by SM Distribution Ltd, 6 Leigham Court Road, London SW16 2PG. Telephone 0181-677 8111. Overseas and non-newstrade sales by Magazine Sales Department, Argus House, Boundary Way, Hemel Hempstead, HP2 7ST, Telephone (01442) 66551. Subscriptions by Nexus Subscription Services, ETI, Queensway House, 2 Queensway, Redhill, Surrey RH1 1QS. Telephone (01737) 768611.
US subscriptions by Wise Owl Worldwide Publications, 4314 West 238th Street, Torrance, CA90505 USA. For Visa/Mastercard orders in USA - Telephone (310) 375 6258 Fax (310) 375 0548. Pacific Time: 9am-9pm Weekdays, 10am-6pm Weekends. Typesetting and origination by Ebony, Liskeard, Cornwall. Printed by Wiltshire Ltd. Bristol.

NEXUS Special Interests

Nexus House, Boundary Way,
Hemel Hempstead HP2 7ST
Telephone (01442) 66551
Fax (01442) 66998


MAPLIN
ELECTRONICS AND BEYOND
FULL COLOUR GUIDE TO ELECTRONIC PRODUCTS
NEW
1996 CATALOGUE
Bigger, Brighter and Better Value than ever before. Over 1180 Full Colour pages with hundreds of Brand New Products at Super Low Prices
on sale from 1st September
only £2.95
from all branches of WHSMITH and other major retail outlets.
Order by post including P & P £3.45
01702 554161
TEST GEAR
PCB EQUIPMENT
COMPUTER ACCESSORIES
ELECTRONIC COMPONENTS
TOOLS
BOOKS
Adventures with ELECTRONICS
INVESTIGATING DIGITAL ELECTRONICS
ELECTRONIC CIRCUITS
VISIT YOUR LOCAL MAPLIN STORE TODAY
To place your order and to discover a world of electronics and beyond
Belfast 357-359 Lisburn Road. Birmingham Sutton New Road, Erdington. Bradford 28 Valley Road, Hamm Strasse. Brighton 65 London Road. Bristol 302 Gloucester Road. Cardiff 29-31 City Road. Chatham 2 Luton Road. Coventry 12 Bishop Street. Dudley Unit 7, Pedmore Road. Edinburgh 126 Dalry Road. Glasgow 264-266 Great Western Road. Ilford 302-304 Green Lane. Leeds Carpet World Building, 3 Regent Street. Leicester Office World Building, Burton Street. Liverpool Edge Lane, Fairfield. London 146-148 Burnt Oak Broadway, Edgware; 107-113 Stanstead Road, Forest Hill; 120-122 King Street, Hammersmith. Manchester 8 Oxford Road; 169 Cheetham Hill Road, Cheetham Hill. Middlesbrough Unit 1, The Forbes Building, Linthorpe Road. Milton Keynes Unit 2, Office World Building, Snowdon Drive, Winterhill. Newcastle-upon-Tyne Unit 4, Allison Court (The Metro Centre), Gateshead. Northampton 139 St. James Road. Nottingham 86-88 Lower Parliament Street. Portsmouth 98-100 Kingston Road. Preston Unit 1, Corporation Street. Reading 129-131 Oxford Road. Sheffield 413 Langsett Road, Hillsborough. Slough 216-218 Farnham Road. Southampton 46-48 Bevois Valley Road. Southend-on-Sea 282-284 London Road, Westcliff-on-Sea. Stockport 259-261 Wellington Road South. Stoke-on-Trent 39-45 London Road. Plus many NEW stores opening soon. Tel: (01702) 552911 for details of your nearest store.