Construction of the **touch tuning** circuit requires a bit of handiwork with a fret-saw. The printed circuit board consists of three sections; once separated, the two smaller sections are mounted perpendicularly to the main board to form a compact module.

The **gate dipper** is a useful little device that can be used to determine the resonant frequency of tuned circuits. Basically, it is the modern equivalent of the old grid-dip meter.

The 'heart' of the **strain gauge** is the 'stress absorber', which consists of a sheet of suitable metal onto which a bridge configuration of four electric-resistance strain sensors are bonded.

'I **played TV games**...' is a description of the TV games computer, written by a novice and for novices! One thing has become quite clear: with a little practice, even fairly sophisticated programs can be designed.

An important selling point of modern stereo tuners is the number of preset stations which can be selected. However for the home constructor, this is often a feature which must regrettably be foregone, being regarded in many designs as something of a luxury. The circuit described is intended to remedy that situation, by providing for up to 9 touch controlled preset stations. The only restriction is that the receiver must be varicap tuned.

It is often very useful to be able to match the values of capacitors and resistors and the only quick, effective way to do this is by using an **impedance bridge**. The circuit described is quite adequate for this purpose and it is also capable of measuring resistances between 100 Ω and 1 MΩ and capacitances between 100 pF and 1 μF.

Good news for SC/MP fans: two new records have been added to the ESS range. One contains the complete NIRL-E program; the other includes some games, a 'running script' program, 'tracer', 'disassembler' and 'biohythms'. Some further details on the latter programs are given here.

Given the fact that many types of capacitor — especially electrolytics — have a wide tolerance (20% is fairly common), it is often desirable to be able to measure capacitances both quickly and with a reasonable degree of accuracy. Of course a capacitance meter also enables one to measure the value of those piles of unmarked capacitors which end up at the bottom of one's junk box, or to test 'suspect' capacitors for potential faults — in short it represents a useful addition to the test gear of any constructor.

There are few projects which have not formed the subject of an article in Elektor at one time or another, however a strain gauge falls into that category. This in itself is perhaps slightly surprising, since there is a number of possible applications for such a device — a training aid for 'strength sports', measuring loads on cables, etc. or simple weighing purposes.

Writing your own programs for the TV games computer is fairly easy, provided you know the basic principles. The most important instructions are discussed this month, with some simple program examples.

---

*Selektor* .......................... 10-01

*Touch tuning* .......................... 10-06

*battery saver* .......................... 10-09

*Impedance bridge* .......................... 10-10

*New programs for the SC/MP* .......................... 10-12

*Digital rev counter* (A. Onde) .......................... 10-15

*Digifarad* (J. Guther) .......................... 10-16

*Short-interval light switch* .......................... 10-20

*p.c.b. for variable fuzz box* .......................... 10-20

*Gate dipper* .......................... 10-22

*Strain gauge* (W. v. Dreumel) .......................... 10-26

*I played TV games* .......................... 10-28

*Programmable sequencer* (C. Voss) .......................... 10-38

*Missing link* .......................... 10-41

*Market* .......................... 10-41

*Advertisers index* .......................... UK-30
What is a TUN?
What is a TUR?
What is the EPS service?
What is the TQ service?
What is a missing link?

Semiconductor types
Very often, a large number of equivalent semiconductor components exist with different type numbers. For this reason, abbreviated type numbers are used in Elektor wherever possible.

- '741' stands for μA741.
- LM741, MC741, CM741, RA741, SN74241, etc.
- 'TUP' or 'TUN' (Transistor, Universal, PNP or NPN respectively) stand for any low frequency silicon transistor that meets the following specifications:

| UCEO, max | 20V |
| IC, max | 100 mA |
| hFE, min | 100 |
| Pout, max | 1000 mW |
| ft, min | 100 MHz |

Some 'TUN's are: BC107, BC108 and BC109 families, 2N3855A, 2N3859, 2N3860, 2N3904, 2N3947, 2N4124. Some 'TUP's are: BC177 and BC178 families; BC179 family with the possible exception of BC159 and BC179; BC212, 2N3221, 2N3906, 2N4126, 2N4291.

- 'DUS' or 'DUG': These transistors are a DMOS silicon or Germanium respectively and stand for a diode that meets the following specifications:

| UR, max | 25V |
| IF, max | 100mA |
| IR, max | 1mA |
| Pout, max | 250W |
| GD, max | 50 pcs. |

Some 'DUS's are: BA127, BA217, BA218, BA221, BA222, BA317, BA318, BAX13, BAY161, NY104, NY1144.

Some 'DUG's are: OZ85, OA91, OA95, AA116.

- 'BC107', 'BC237', 'BC457': All refer to the same 'family' of almost identical better-quality silicon transistors. In general, any member of any one family can be used instead.

- 'BC107' families: 2N397, 2N398, 2N399, 2N407, 2N408.
- 'BC207' families: BC207 (8 - 9).
- 'BC208' families: BC208 (8 - 9).
- 'BC217' families: BC217 (8 - 9).
- 'BC237' families: BC237 (8 - 9).
- 'BC457' families: BC457 (8 - 9).

Resistor and capacitor values
When giving component values, decimal points and large numbers of zeros are avoided wherever possible. The decimal point is usually replaced by one of the following abbreviations:

- pico- (p) = 10^-12
- nano- (n) = 10^-9
- micro- (μ) = 10^-6
- milli- (m) = 10^-3
- kilo- (k) = 10^3
- mega- (M) = 10^6
- giga- (G) = 10^9

A few examples:
- Resistor value 2k7: 2700 Ω
- Resistance value 470: 470 Ω
- Capacitance value 0.000 000 000 0047 F
- Capacitance value 10n: this is the international way of writing 10,000 pF, or 0.01 μF, but since 1 n is 1 x 10^-9 farads or 1000 pF.
- Resistors are ½ Watt 5% carbon types, unless otherwise specified.

The DC working voltage of capacitors (other than electrolytic) is usually taken to be at least 60 V. As a rule of thumb, a safe value is usually approximately twice the DC supply voltage.

Test voltages
The test voltages shown are measured with a 20 kΩ instrument, unless otherwise specified.

Mains voltages
No mains (power line) voltages are listed in Elektor circuits. It is assumed that our readers know what mains voltage is standard in their part of the world.

Readers in countries that use 60 Hz should note that Elektor circuits are designed for 50 Hz operation. This will not normally be a problem; however, in cases where the mains frequencies are used for synthesis or some modification may be required.

Technical services to readers
- EPS service. Many Elektor articles include a lay-out for a printed circuit board. Some - but not all - of these boards are available ready-etched and predrilled. The 'EMC printed circuit list' in the current issue always gives a complete list of available boards.
- Technical queries. Members of the technical staff are available to answer technical queries (relating to articles published in Elektor) by telephone on Mondays from 13.30 to 16.45. Letters with technical queries should be addressed to: Dept. TQ. Please enclose a stamped, self-addressed envelope to cover any overtime costs. We usually reply by return of post.

Selling link. Any important modifications, additions to, improvements on or corrections in Elektor circuits are generally listed under the heading 'Selling Link' at the earliest opportunity.
New home-video standard from Philips

8 hours of TV from one cassette

The latest Philips and Grundig top-hit in the home-video field, the Video-2000 system, has received world-wide attention. One cassette, containing about 1000 feet of 1/2-inch tape, can be used to record eight hours of colour TV. One cassette costs about £20, so that one hour of TV program costs just over £2 to record. By way of comparison, the first colour video recorder used was over £200 of tape per hour. The picture quality hasn't suffered by this drastic cost reduction. All in all, it's not just another step forward — it's a giant leap! Obviously, Philips and Grundig hope that the new video cassette will be accepted as an international standard, and that it will prove the same long-term success as its predecessor: the compact audio cassette.

The Philips recorder, designed around this new cassette, is a beautiful piece of modern technology.

The first obvious difference between the Video-2000 system and older Philips systems (and JVC's VH-5 and Sony's highly-praised Betamax) is the narrow tape-track used. Until now, 1/2-inch tracks were fairly common; Philips and Grundig use less than half — the 1/2-inch tape in their cassette is used in both directions, since the cassette can be 'turned over'. It is used in the same way as the audio compact cassette — the only difference is that it is used to record television programs!

The new cassette can not be used on older video recorders, like the N1700. That particular machine will be going out of production in the second half of 1980; suitable cassettes will, however, remain available for some time to come. A new video recorder, the VR2200, is designed to use the new cassette. The price is expected to be about 30% more than that of the N1700 — around the £600 mark. Several new features are included in the new machine, and these deserve some further explanation.

Use of the tape

Before discussing the recorder itself, it is a good idea to take a look at the way in which it writes the program material on tape. Figure 1 shows where the information is written on the tape.

Since the tape can be 'turned over', the upper and lower half of this 'tape map' are mirror images. Starting from the outside, the first 650μm are used for a (mono) audio track. At a later date, if stereo sound ever gets off the ground for television broadcasting, this area can be divided into two 250μm tracks with a 150μm gap.

The next 4.85mm wide section is reserved for the video signal. This is recorded in a single track, like the audio signal; as in most video recorders, narrow (22.6μm wide) tracks are recorded at a slight angle in this section. In this particular case, the video tracks are angled at 3° with respect to the tape 'axis' — for clarity, an angle closer to 30° is used in figure 1... The final section of tape before reaching the centre (where the mirror image begins) is unused at present. This 300μm wide strip of tape can be used, at a later date, for various control signals.

All these sections are repeated on the other side of the centre-line, for what is called (in analogy with gramophone records) 'the other side of the tape'.

Video recording

Which is what the whole exercise is about... The video signals to be recorded run up to fairly high frequencies (approximately 4.8 MHz). In any type of recording, the 'detail' that can be written depends on how coarse or fine the writing implement is. In a tape recorder, the recording implement is the tape head; its 'gap width' determines the detail that it can write on the tape.

However, the 'space' required to record one period of a 4.8 MHz signal on tape depends on the speed with which the tape runs past the heads. The higher the speed, the longer a single period will be stretched on the tape. If the tape runs relatively slowly, a larger number of periods could theoretically be recorded on a small section of tape; however, the tape head is too 'blunt' to make this a feasible proposition. The result would be poor picture quality. Somehow, the speed of the tape relative to the head must be increased until the picture quality leaves little to be desired. If the tape is run at high speed past a stationary head, the picture quality can be quite good — but the recorder will 'eat' tape, both in feet of tape required per minute and in life expectancy of the tape... For this reason, it has become standard practice in video recorders to use a rotating head drum, incorporating two or more heads. This drum revolves at high speed, so that the heads move at high speed past the tape, even if the latter is transported relatively slowly.

The head drum is mounted at a slight angle with respect to the tape and very narrow video tracks are used, so that a fairly slow tape transport suffices to move the tape up sufficiently to write the next diagonal track adjacent to its predecessor.

All this may seem rather complicated, but it is basically similar to typing. Even if you type a lot of letters, it takes a while to fill the page — certainly if you use the minimum line spacing, so that each new line practically touches the one above. Something similar occurs in a video recorder; the main difference being that the tape is moved slowly and constantly, instead of jumping up 'one line at a time' like the paper in the typewriter. If you can visualise the paper moving up at a constant speed, so that it has just moved up one line by the time you start to type a new line (so that the lines slope down slightly), you have the principle of the video recorder.
Figure 2 illustrates how this system operates in practice. Two heads are mounted on the drum, and the latter is mounted at a slight angle with respect to the tape. As the tape is transported at a speed of 2.44 cm/s (just less than 1 inch/second, or about half the speed of an audio cassette recorder), diagonal tracks are written on it by the heads on the upper half of the drum. The lower half of the drum runs at a much lower speed, and takes care of the tape transport — it operates as a large diameter 'capstan'. The diameter of the drum is 85 mm and the upper half rotates at 25 revs per second, so that the two heads (K1 and K2) move at 5.08 m/s — or just under 17 ft. per second! The tape is 'wrapped around' half the circumference of the drum, so that as one head leaves the tape the other just starts to write on it. It will now be apparent how the tracks are recorded. The tape is almost stationary with respect to the upper half of the drum. One head can therefore record a track length equal to approximately half the diameter of the drum — very roughly, 100 mm or 3/8".

We said that the tape is almost stationary with respect to the head. To be more precise, the tape is transported over 22.6 cm as one full track is recorded. When the second head 'hits the track', the tape has moved up just far enough to enable this head to record its track parallel to the previous one, without overlapping. The final result is a series of diagonal tracks, as illustrated in figures 1 and 2.

A separate, stationary head is used to record the audio signal. This 'sound track' is located at the outer edge of the tape, as illustrated in figure 1. The erase heads are also stationary.

Vertical positioning of the video heads
With the extremely narrow video tracks written diagonally on the tape, positioning of the video heads during playback is obviously highly critical. Some way must be found to move the heads slightly until they are centred exactly on the corresponding tracks. A most intriguing solution has been found. The video heads are both mounted on a little piece of piezo-ceramic material. This is the material used in no-battery electric lighters: when it is compressed, a voltage appears across the ends, sufficient to draw a spark. However, it also works the other way: if a voltage is applied across the ends of the material, its shape will vary! The so-called PXE is used in this way in the video recorder.

By varying the voltage applied to it, the height of the heads can be adjusted.

Dynamic Track Following
It's all very well being able to vary the position of the heads, but first a control voltage must be derived in some way. Figure 3 again shows four video tracks with an exaggerated angle (the true angle being only 3°). Track f2 is written first, then f4 is recorded, and so on. And this is where it gets complicated. Tracks f2 and f3 are written by video head two; tracks f4 and f1 are recorded by head one. Simultaneously with the video signal, a 'pilot tone' is recorded on each track. Each head records two different (relatively low frequency) pilot tones alternately. Track f2 contains a 117 kHz pilot tone, recorded by K2 (video head 2); track f4, written by K1, includes a 164 kHz pilot tone; on track f3, the pilot tone frequency is 149 kHz; finally, the pilot tone on track f1 is
102 kHz. This cycle is repeated for the next set of four tracks, and so on.
During playback, the pilot tones are retrieved together with the video signal.
If the corresponding head (K1) is correctly positioned, a clean 164 kHz tone will be retrieved from track f4.
However, if the head is slightly high, some of the 149 kHz signal on track f3 will be mixed with this 164 kHz signal, producing a 15 kHz beat signal; if the head is low, a 47 kHz beat signal will appear (164 kHz (f4) - 117 kHz (f2) = 47 kHz). For the other head (K2), the opposite is true: if it is too high, a 47 kHz signal is produced; a 15 kHz beat signal corresponds to ‘too low’.
The amplitude of the beat signals is used as a basis for the ‘head height’ control signal.

Azimuth
The video heads are mounted in the head drum at a relative angle of 30°.
With respect to the tape, head one is mounted at 90° - 15° - 3° = 72°; head two is mounted at 90° + 15° - 3° = 102°.
This is illustrated in figure 4, where both heads are shown simultaneously with respect to the tape. There is good reason for the relative angle between the two heads.

As audio recorder enthusiasts will know, if a playback head is tilted slightly with respect to the recorded tape track, the high frequency response is drastically reduced. Correct ‘azimuth’ setting is essential for high quality playback. In this video recorder, the result is that a track originally recorded by K1 will only be ‘read’ by K2 with a severely reduced high-frequency response. In practical terms, this means that K2 will only see frequencies up to a few hundred kilohertz on K1’s tracks – K2 will reproduce K1’s pilot tones, but it will not reproduce the video signal! Cunning...

Head positioning during recording
During recording, accurate positioning of the heads is also required, to keep the tracks up against each other without overlap. To this end, one of the heads is fixed in an ‘average’ position; the height of the other is adjusted so that its tracks are correctly positioned.
A complete TV picture consists of 625 lines, written in two 312.5 line ‘frames’. Between each set of 312.5 lines, a short ‘vertical blanking’ interval is required. No video signal is recorded in this interval. One complete picture (two frames) is recorded 25 times per second... exactly the rotation speed of the head drum! Coincidence? Don’t you believe it. Each head records one frame, and two frames make one picture. Each track includes a vertical blanking interval, which can be used to record a control signal. In the VR 2020, a 223 kHz signal is recorded at this point, for 96 µs. Immediately after this, the head is switched to playback for a further 96 µs. The result of these manipulations is sketched in figure 5. Bearing in mind that the right-hand track is recorded first (the tracks themselves are recorded from lower left to upper right, but the tape movement is also from left to right) it will be
Switching transistors approach 1000 V Barrier

The switch mode power supply has a number of advantages over other conventional supplies. However, their use in high power circuits has been limited by the absence of transistors with sufficiently good characteristics to meet the heavy demands placed on them under high voltage switching conditions. An ideal switching transistor should have characteristics that include:

- very low VCE sat
- very low leakage current
- very good switching characteristics
- very good ruggedness
- good reliability

Characteristics that are difficult to obtain in high voltage high power transistors. The SGS-ATES MULTIPITAXIAL MESA technology, which was created to overcome these problems, gives an excellent compromise between these characteristics. In addition to this it gives the possibility of making complementary NPN-PNP high voltage, high power transistors, a feature impossible to find in other high voltage, very high power technologies.

In the Multipitaxial MESA technology a heavily doped N⁺ substrate is used as a foundation onto which is epitaxially grown a normally doped N type layer. On the N type layer is grown a second epilayer of lightly doped N⁺ type material. These two epilayer form the collector of the transistor. This type of collector construction gives extremely good ruggedness in Es/b conditions. On top of the collector is grown a third epitaxial layer which is to form the base of the transistor (into which an N⁺ type emitter diffusion is made). This epitaxial layer, in order to maintain the high voltage characteristics of the device whilst giving good switching times, must be of a P⁺ type material. However, if the emitter diffusion was made into the base as it stands, problems would arise in the stability of the transistor due to the very high electric field between the P⁺/N⁻ layer seen at the edge surface. Therefore an additional P⁺ diffusion is made into the base epitaxial layer that has the effect of widening the distance between equipotential lines at the surface thus reducing the surface electric field. The P⁺ diffusion does not of course reduce the intrinsic high voltage characteristic of the transistor.

Whilst the multipitaxial layer construction gives a breakdown value in the order of 1000 V it is known that when transistors are separated by mechanical
means, after diffusion, irregularities are
caused on the edge of the transistor
which dramatically reduce the collector/
base breakdown voltage. Obviously the
breakdown voltage of the device as a
whole is the breakdown voltage of the
weakest point, in this case the edge
surface between collector and base.

Therefore a method had to be found
which would allow separation of devices
without causing surface edge irregu-
larities.

The method found, whilst extremely
simple in concept, has had dramatic
effects in improving the collector/base
breakdown characteristics. In essence
the method used is to isolate each
transistor on the wafer by a deep
chemical edge. In this way it is possible
to achieve an extremely smooth edge on
the active part of the transistor in the
area of the base collector junction. It is
the cross section of the transistor after
the deep chemical etch that gives rise to
the name Mesa (after the mesas found in
S.W. United States and Mexico). The
depth of chemical etch is carried out after
the emitter diffusion and then, to
further enhance stability and preserve
surface cleanliness, glass passivation is
carried out on the channel formed. The
cut made to separate the transistors is
then made in the inactive area between
the dice.

Using this method it has been possible
to produce transistors with a V_{\text{max}} as
high as 900 V and with extremely good
reliability and ruggedness in high voltage,
high temperature conditions.

Typical electrical characteristics for
transistors constructed using the
multiemitter mesa technology are
shown in table 1.

Using these transistors it has been
possible to build switch mode power
supplies with a performance never
before possible. A typical example of
these transistors in a switch mode power
supply is shown in fig. 1. This circuit,
which uses two BUW 34's in the power
output stage, is capable of delivering up
to 400 W at 24 V or by using two
BUW 45's at 24 V.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>BUW 34</th>
<th>BUW 35</th>
<th>BUW 36</th>
<th>BUW 44</th>
<th>BUW 45</th>
<th>BUW 46</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_{\text{CBO(min)}}</td>
<td>500 V</td>
<td>800 V</td>
<td>900 V</td>
<td>500 V</td>
<td>800 V</td>
<td>900 V</td>
</tr>
<tr>
<td>V_{\text{CEO(min)}}</td>
<td>400 V</td>
<td>400 V</td>
<td>450 V</td>
<td>400 V</td>
<td>400 V</td>
<td>450 V</td>
</tr>
<tr>
<td>V_{\text{CE(sat(max))}}</td>
<td>1.5 V</td>
<td>1.5 V</td>
<td>1.5 V</td>
<td>1.5 V</td>
<td>1.5 V</td>
<td>1.5 V</td>
</tr>
<tr>
<td>t_{\text{on}}(typ)</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
</tr>
<tr>
<td>t_{\text{f}}(typ)</td>
<td>1.8 \mu s</td>
<td>1.8 \mu s</td>
<td>1.8 \mu s</td>
<td>1.8 \mu s</td>
<td>1.8 \mu s</td>
<td>1.8 \mu s</td>
</tr>
<tr>
<td>INL</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
<td>0.2 \mu s</td>
</tr>
</tbody>
</table>

NOTE: V_{\text{CE(\text{sat})}} is specified at I_c = 5 A, I_B = 1 A for BUW 34, 35, 36 and I_c = 10 A,
I_B = 2 A for BUW 44, 45, 46. Sw. on characteristics typified at V_{\text{CC}} = 250 V,
I_c = 5 A, I_B = 1 A for BUW 34, 35, 36 and V_{\text{CC}} = 250 V, I_c = 10 A, I_B = 2 A for BUW 44, 45, 46.
Many FM tuners employ varicap (variable capacitance) diodes. These are diodes which are especially designed so that their capacitance can be varied by means of a control voltage. If the varicaps are included in an LC circuit, the resonant frequency of the latter can thus be varied by altering the control voltage. In most tuner designs the control or tuning voltage is derived from a stabilised supply and is varied by means of a potentiometer. The main requirements of the tuning voltage are that it must be stable and affected as little as possible by fluctuations in temperature.

Preset tuning can be realised by using not one potentiometer, but a number of potentiometers connected in parallel, these being selected by switches (see figure 1). Only one switch may be closed at any given time; for example, when switch S3 is closed, switch S1 automatically opens. By adjusting the preset potentiometers so that each switch brings in a different station, a simple and effective preset tuning facility is obtained.

In the circuit described here, the basic design has been further refined, so that using only two switches a total of 10 preset stations can be selected. By employing touch switches, the need for interlocking switch assemblies is avoided, whilst the physical construction and appearance of the switches can be tailored to suit individual requirements.

Circuit

For a range of 87 to 104 MHz, the tuning voltage of most receivers must be capable of being varied from roughly 2 or 3 volts to approximately 30 volts. Thus it is clear that conventional CMOS switches cannot be used, since they are only capable of switching voltages of up to 15 V. However, as can be seen from the circuit diagram of figure 2, CMOS buffers N1...N4 are used to form a pair of suitable touch switches.

Under normal conditions the inputs of N1 and N3 are held high via R1 and R2. When one of the sets of touch contacts is bridged, the input of the corresponding gate is pulled down to ground (logic 0). The output of the gate is thus taken high, with the result that C1 or C2 rapidly charges up and the output of the succeeding buffer (N2/N4) goes low. Removing one's finger from the touch contacts takes the output of the first gate low again, causing the corresponding capacitor to discharge slowly via the parallel resistor. Thus each time one of

An important selling point of modern stereo tuners is the number of preset stations which can be selected. However for the home constructor, this is often a feature which must regrettably be foregone, being regarded in many designs as something of a luxury. The circuit described here is intended to remedy that situation, by providing for up to 9 touch controlled preset stations. The only restriction is that the receiver be varicap tuned.
the touch switches is operated a logic 0 is applied to the up or down input of IC1 (synchronous decade up/down counter). This IC counts the pulses applied to its inputs when the LOAD input is high, and transfers the result in BCD form to its outputs.

Upon switch-on the LOAD input of the counter is held low via capacitor C3, so that the counter outputs are reset i.e. also taken low. When the 'up' touch switch is operated, the counter increments by one, i.e. the number 1 appears in BCD at the counter outputs. If the up switch is touched a second time, the number 2 appears at the counter outputs, and so on. Touching the 'down' switch decrements the

Figure 1. This arrangement of potentiometers and switches represents a simple method of selecting preset stations. The only drawback is the need for an interlocking switch assembly.

---

Figure 2. Complete circuit for preset touch tuning. Using only two touch switches a choice of 9 preset stations can be selected.
Figure 3. The printed circuit board for the touch tuning circuit. Some initial work with a fret-saw is required to separate the board into 3 sections.

<table>
<thead>
<tr>
<th>Parts list:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistors:</td>
<td></td>
</tr>
<tr>
<td>R1, R2 = 2M7</td>
<td></td>
</tr>
<tr>
<td>R3, R4 = 10 M</td>
<td></td>
</tr>
<tr>
<td>R5 to R11 = 220 Ω</td>
<td></td>
</tr>
<tr>
<td>R12 to R21, R23 = 100 k</td>
<td></td>
</tr>
<tr>
<td>R22 = 1 M</td>
<td></td>
</tr>
<tr>
<td>P1 to P9 = 20-turn preset potentiometer (Piher), 50 k or 100 k</td>
<td></td>
</tr>
<tr>
<td>P10 = 10-turn potentiometer, 50 k or 100 k</td>
<td></td>
</tr>
<tr>
<td>Capacitors:</td>
<td></td>
</tr>
<tr>
<td>C1, C2 = 3n3</td>
<td></td>
</tr>
<tr>
<td>C3, C4, C5 = 100 n</td>
<td></td>
</tr>
<tr>
<td>Semiconductors:</td>
<td></td>
</tr>
<tr>
<td>D1, D2 = DUS</td>
<td></td>
</tr>
<tr>
<td>T1 to T10 = BC 556, BC 557</td>
<td></td>
</tr>
<tr>
<td>IC1 = 74LS192</td>
<td></td>
</tr>
<tr>
<td>IC2 = 74141</td>
<td></td>
</tr>
<tr>
<td>IC3 = 7447</td>
<td></td>
</tr>
<tr>
<td>IC4 = N1 to N4 = CD 4093</td>
<td></td>
</tr>
<tr>
<td>Display = HP 5082 - 7750 (common anode)</td>
<td></td>
</tr>
</tbody>
</table>
number on the counter outputs by 1. The outputs of the counter are connected to a BCD-decimal decoder/driver (IC2). Depending upon the BCD input data, one of the outputs of this IC will go low. The counter outputs are also connected to a BCD-7-segment decoder/driver, which in turn is connected to a 7-segment display. In this way the state of the counter (and the output of IC2 which is active) is clearly indicated.

When one of the outputs of IC2 goes low, the corresponding transistor is turned on. The emitter voltage of the transistor is determined by the position of the associated potentiometer wiper. Only a small saturation voltage is dropped across the transistor. The output voltage of the circuit (i.e. the tuning voltage for the varicap diodes) can thus be set by adjusting each potentiometer to give the appropriate voltage when the corresponding output of IC2 goes low.

Altogether 9 preset potentiometers are used, which means 9 preset stations. If no preset station is required (the counter output is zero) tuning through the FM band is accomplished by means of a conventional (ten-turn) potentiometer.

Construction

Construction of the touch tuning circuit requires a bit of handiwork with a fret saw. The printed circuit board, which is obtainable via the EPS service, consists of three sections, which before the components are soldered in place, must first be separated from one another. On one section of the board are four copper planes, which form the two pairs of touch contacts. A second section of the board is intended to accommodate the 7-segment display. A section is sawn out of the main board at the point where the display is to be mounted. The display board and the touch contacts are mounted perpendicularly to the edge of the main board, as shown in the accompanying photograph. Of course the individual is free to choose an alternative design for the touch switches if desired. The potentiometers used are 20-turn presets from Piher. The existing tuning potentiometer in the receiver can be used for the 10-turn potentiometer.

In conclusion

Since transistors are used as voltage switches, the circuit is slightly temperature dependent. However most tuners have fairly good automatic frequency control (AFC), which should ensure that this is not a problem. The supply voltage is 5 V, whilst the input tuning voltage should not exceed 30 V. When power is applied, the circuit automatically selects channel 0, i.e. the receiver can be tuned by hand. If one wishes a preset station to be selected immediately after switch-on, then the inputs of IC1 can be programmed to select another channel. For example, if pin 15 of the IC is connected to plus supply, channel 1 will automatically be selected.

Finally, it is perhaps worth remarking that if the display is not required, then R5 . . . R11, IC3, and the display itself can of course be omitted.

Photo 1. A section of p.c.b. is sawn out of the main board at the point where the display board, accommodating the 7-segment display, is to be mounted.

Basiclly the circuit is a simple timer. Pushbutton switch S1 is the start button for the die, roulette wheel, etc. When depressed, it causes capacitor C1 to charge up rapidly via D1. Transistor T1 is turned on, so that, via T2, the relay is pulled in, thereby providing the circuit of the game with supply voltage. When the switch is released, initially nothing will happen. C1 discharges via R1, R2 and the base-emitter of T1, however it takes several seconds until it has discharged sufficiently to turn off T1. When it does so, however, the relay drops out, cutting out the power supply to the die, etc.

With the component values shown in the circuit diagram, a delay of roughly 3 seconds is provided in which to read off the display. If that interval is too short (or too long), it can be modified as desired by choosing different values for C1 and/or R1/R2.

W. Jitschin
It is often very useful to be able to match the values of capacitors and resistors and the only quick, effective way to do this is by using an impedance bridge. The following circuit is quite adequate for this purpose and it is also capable of measuring resistances between 100 \(\Omega\) and 1 M\(\Omega\) and capacitances between 100 pF and 1 \(\mu\)F.

### Measuring resistance

Most readers will be familiar with the basic Wheatstone bridge circuit shown in figure 1, which represents the simplest way of measuring an unknown resistance. The bridge is formed by two pairs of resistors (voltage dividers) which are connected in parallel. As every reader will know (we hope), when two resistors are connected in series, the voltage dropped across each resistor is proportional to the value of that resistor. Thus if the resistors are connected as shown in figure 1 and we ensure that the ratio of \(R_a\) and \(R_b\) to \(R_x\) and \(R_c\) is the same, the voltages at points A and B must also be the same. To put it another way, for the bridge to be ‘balanced’ and the meter to read zero voltage between points A and B, \(R_a \times R_c\) must be the same as \(R_x \times R_b\). If now we make \(R_b\) variable and provide it with a calibrated scale, then by adjusting \(R_b\) until the meter shows zero deflection we can determine the value of the unknown resistance, \(R_x\).

### Measuring capacitance

Measuring capacitance is slightly more complicated than measuring resistance, however the basic principle involved is the same. A capacitor also possesses resistance to current flow, which is called its reactance, and like resistance is measured in \(\Omega\). Unlike a resistor, however, it is only meaningful to talk of a capacitor’s reactance to alternating current, since capacitors do not pass steady current at all. Furthermore, the reactance of a given capacitor is frequency-dependent, i.e. the greater the frequency of the voltage across it, the lower its reactance, and vice-versa. For this reason, we have to ensure that the supply voltage to our Wheatstone bridge is alternating and of constant frequency (it of course makes no difference to a resistor whether the voltage is AC or DC). Once that is the case, the reactance of the capacitor is determined solely by its capacitance. Thus if we replace the unknown resistance, \(R_x\), by the unknown capacitance, \(C_x\), and one of the fixed resistors in the bridge by a fixed capacitor, we can determine the value of \(C_x\) from the setting of the calibrated variable resistor, \(R_p\).

Since the capacitors are connected in series with a resistor, strictly speaking the meter is measuring impedance, hence the name, impedance bridge. When the variable resistor is adjusted for zero deflection on the meter, Wheatstone’s formula once again applies, i.e.: \(Z_k \cdot R_b = R_g \cdot Z_c\), where Z is the symbol for impedance (in \(\Omega\)).

### Circuit

The complete circuit diagram of the impedance bridge is shown in figure 2. As already explained, a resistance remains the same, regardless of whether the voltage source is steady or alternating. Thus we can choose an alternating supply voltage for the bridge. In order to be able to measure fairly small capacitance values, a reasonably high frequency (significantly higher than the main frequency) is required, and to this end a Wien bridge oscillator, formed by the circuit round op-amp A1, is used. When the gain of the op-amp is \(x\), the oscillator produces an alternating voltage with a frequency of roughly 1 kHz. The gain of the op-amp can be varied by means of P1, thus ensuring that the oscillator can always be started. Ideally P1 should be adjusted such that the circuit just oscillates and no more. If desired the oscillator output can be examined on an oscilloscope and P1 adjusted for as sinusoidal a waveform as possible, although this step is not strictly necessary. A2 functions as a buffer stage, delivering sufficient power to drive the bridge.

The Wheatstone bridge is clearly recognisable in the circuit diagram. If we compare it with the circuit of figure 1, it is apparent that resistor \(R_b\) is replaced by four different value resistors, each of which can be selected by the range switch, S1. Potentiometer P2 assumes the function of variable resistor \(R_p\) in figure 1. When the wiper of this potentiometer is turned hard up against the end stop such that no greater resistance can be measured, one simply has to select a larger value for \(R_b\). The fixed value capacitor in the bridge is formed by C8. This capacitor is connected in series with another potentiometer, P3. During the measurement procedure, when P2 is being adjusted for zero deflection on the meter, P3 is set for zero resistance. Once the measurement has been completed, the quality of the unknown capacitor (\(C_x\)) can be determined with the aid of P3.

How this is done is discussed in the section on using the impedance bridge. The voltage between points A and B in the circuit is measured by the differential amplifier A3. C6/R17 and C7/R15 ensure that only the 1 kHz alternating
impedance bridge

voltage appears across the inputs of A3. The output of A3 is fed via C9 to A4, which in conjunction with D6 provides a half-wave rectified voltage, suitable for driving the meter (which in fact displays the average value of the rectified signal).

Construction
It should not be difficult to construct the circuit using Vero-board or similar. If the circuit is mounted in the same box as the power supply, then care should be taken to place diodes D3 and D4, which stabilise the amplitude of the oscillator signal, at a reasonable distance from components which are liable to run warm. This point should not prove a serious problem, however, since the circuit only consumes some 20 mA.

Any readily available meter will prove suitable since it is not required to provide a reading which is accurate in absolute terms, rather it is a question of determining which setting of P2 gives the smallest deflection. The meter is being used to give a 'dip-reading'.

Using the impedance bridge

The general operation of the impedance bridge should be fairly clear from the foregoing description of the circuit. First of all however, the circuit must be calibrated. This is done by adjusting P1 until the oscillator starts. The oscillator can be checked by setting P4 to roughly the mid-position and connecting a wire link between the test terminals (Zx). When the oscillator starts the bridge will cease to be in a state of equilibrium (which is another way of saying that a potential difference exists between points A and B in the circuit). It may occur that the oscillator will stop after a short period; this simply means that P1 was not set to the optimal position and should be readjusted.

With P2 set for minimum resistance and S1 in position 4, P4 is then adjusted until maximum deflection is obtained on the meter. Diodes D6 and D7 are included to limit the current through the meter to an acceptable value; however if full-scale deflection cannot be obtained on the meter, an additional diode can be connected in series with D6/D7. Alternatively, should it prove impossible to limit the current through the meter sufficiently by means of P4, then D6 can be replaced by a wire link. Once the bridge has been set up, the next question is, how do we provide P2 with an accurately calibrated scale?

The simplest solution would be to print a suitable scale in this article. Unfortunately this is not really feasible, since P2 must be a linear potentiometer, and different types have a different effective electrical rotation. Furthermore the first and last sections of the potentiometer tracks are not completely linear, and the extent of the non-linearity varies from potentiometer to potentiometer. For these reasons it is better to experimentally determine a suitable scale oneself.

First of all, S2 is set to position R (measurement of resistance). S1 is then set to position 1 and a series of close tolerance resistors with values ranging from 100 Ω to 1 kΩ are mounted between the test terminals. For each resistor, P2 is adjusted until the bridge is balanced (i.e. minimum deflection on the meter). At the corresponding position of P2 a mark is drawn on the scale, accompanied by the first two figures of the resistor value separated by a full-stop. For example, if R2 equals 470 kΩ, one writes 4.7. For the different positions of the range switch, S1, the following multipliers give the correct magnitude of the values:

<table>
<thead>
<tr>
<th>Position</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 x 100 n</td>
</tr>
<tr>
<td>2</td>
<td>2 x 10 n</td>
</tr>
<tr>
<td>3</td>
<td>3 x 1 n</td>
</tr>
<tr>
<td>4</td>
<td>4 x 0.1 n</td>
</tr>
</tbody>
</table>

The calibration procedure need only be carried out for one range; thereafter the scale will also be correct for the other ranges.

To calibrate the scale for capacitors, S2 is set to position 2 and P3 adjusted for zero resistance. Close tolerance capacitors between 1 n and 10 n are then connected between the test terminals in turn, and P2 adjusted for minimum deflection on the meter. Once again the scale is marked at the corresponding positions of P2. For the value 1 n, switch S1 should be set to position 4; for larger values up to and including 10 n, position 3 is required. The scale will 'run' in the opposite direction to that for resistors, i.e. the scale will decrease from 10 down to 1 from left to right, whereas with resistors it increases from 1 to 10.

The multipliers for each position of the range switch are:

<table>
<thead>
<tr>
<th>Position</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 x 100 n</td>
</tr>
<tr>
<td>2</td>
<td>2 x 1 n</td>
</tr>
<tr>
<td>3</td>
<td>3 x 0.1 n</td>
</tr>
<tr>
<td>4</td>
<td>4 x 0.01 n</td>
</tr>
</tbody>
</table>

Circuit diagram of the impedance bridge.

Figure 1. The basic Wheatstone bridge. In order to measure capacitance, $R_c$ is replaced by a capacitor and the unknown capacitance inserted in place of $R_x$.

Figure 2. Complete circuit diagram of the impedance bridge.
new programs for the SC/MP

As usual, each program is preceded by a succession of 1200 Hz tones, to indicate the program number.

Program 1: Luna (R. Bayer)
This program simulates the landing of the LEM (Lunar Module) on the moon. The display gives information on the height above the surface, the rate of descent and the amount of fuel left in the tank.

The maximum thrust available from the engine is limited, so that leaving reverse thrust too late will result in a crash landing. The maximum permissible descent rate at the moment of touchdown is 01; if this is achieved, the display will alternate the final results with the message ‘landed’.

It is not an easy matter to control a LEM, and the result is that landings may well be rougher than intended. In that case, the message on the display will leave no doubt: ‘crashed’!

There is another way that things can go wrong. If too much thrust is used too soon, the fuel supply may run out before touchdown. This is indicated by an ‘F’ in the last digit (for ‘fuel?’), after which the speed will gradually pick up to the fatal moment of impact... ‘crashed’!

If the loudspeaker interface is included in the SC/MP system, the program will provide some suitable sound effects. The thrust that the motor is putting out can be recognised from the frequency of the rattle coming out of the loudspeaker. A crash landing is accompanied by a lot of noise, that can be interpreted as an explosion. The higher the speed at the moment of impact, the longer the racket will last.

Would-be astronauts have an option not available to their real-life counterparts. If it becomes obvious that things are getting out of hand, the landing can be interrupted by operating one of the other keys (other than 0...7). The original ‘Luna’ display then re-appears.

Good news for SC/MP fans: two new records have been added to the ESS range. One contains the complete NIBL-E program; the other includes some games, a ‘running script’ program, ‘tracer’, ‘disassembler’ and ‘biorhythm’.

Some further details on the latter programs are given here.

The keyboard is used to control the rate of descent. When the program is started (at address 0C00), the text ‘Luna’ appears on the display. The game is then started by operating any one of the keys on the HEX 1/O keyboard; the display now gives all relevant information. The first three digits (reading from the left) indicate the height of the LEM. The fourth digit is always off; the fifth and sixth give the present rate of descent. Finally, the seventh digit is off and the eighth gives the remaining fuel.

The power output of the engine can be controlled with keys 0...7. It is the intention to use the engine to brake down to a soft landing, but it should be noted that over-enthusiastic ‘reverse thrust’ can reverse the direction, so that the module starts to move away from the moon! A further point to watch is that operating the ‘0’ key shuts off the engine... after which it cannot be restarted! The initial thrust of the engine is set by the program to ‘2’, with the result that the LEM picks up speed towards the surface quite rapidly.
after which a new attempt can be initiated by operating any one of the keys.

Program 2: Battleships
(F. Schulte)
'Battleships' is normally a game for two players. In this program, the computer takes the role of one of the players. The game is played on a 64-square 'ocean', as shown in figure 1. In all, six ships take part in the engagement: two of three squares each, two of two squares and two of one square each. The ships may only be entered in horizontal or vertical direction, and they are not allowed to touch. When the program is started (at address 0C40), the word 'Ships' appears in the display. As soon as any key is operated, the computer draws in its own set of ships in its memory. It then invites its opponent to take the initiative: 'Fire'. The coordinates of the first square to come under fire can now be entered: first the line number and then the column number (or letter). The computer can reply in three ways:
1. If the shot landed on one of its ships, it will display 'Hit'. After a brief delay, it will invite a further try: 'Fire'.
2. If a ship is sunk, that is to say if all the corresponding squares have already been hit, this is indicated by the word 'Lost'; after a brief delay, this is again followed by 'Fire'.
3. A miss is indicated by the word 'Fail' The computer will follow this by a shot of its own, indicated as 'shot XY', where X and Y are the

the column number (or letter). The computer can reply in three ways:
1. If the shot landed on one of its ships, it will display 'Hit'. After a brief delay, it will invite a further try: 'Fire'.
2. If a ship is sunk, that is to say if all the corresponding squares have already been hit, this is indicated by the word 'Lost'; after a brief delay, this is again followed by 'Fire'.
3. A miss is indicated by the word 'Fail'. The computer will follow this by a shot of its own, indicated as 'shot XY', where X and Y are the

line and column numbers, respectively. The player can now answer in three ways:
1. A hit is recognised by operating the 'Down' key. The computer will reply immediately with 'shot XY'.
2. Operating the 'Up' key indicates that a ship is sunk. This, too, will be acknowledged with another shot.
3. A miss is indicated by operating any other key. The computer will tell you to get on with it, in that case: 'Fire'.
As soon as all ships of one of the sides are sunk, the word 'end' will appear on the display. After a brief delay, the program will reset and the word 'Ships' will appear.

Program 3: Keyplay
(F. de Brujin)
This game is known under a variety of names, 'NIM' being one of the most popular. It can be played with matches, sticks, coins, or... numbers. The rules are simple: each player in turn subtracts a number from the original; the one to get 0 as result, wins.

When the program is started, at address 0C00, the program will ask for a four-digit decimal number ('GE' = Give Entry); this is the number from which the players will subtract in turn. Next, the program will want to know the Limit ('L1'); this is the maximum number that may be subtracted at one time.
The human player is allowed to start. This is indicated by 'U' in the first display digit. A four-digit number can now be entered. If it is either 0 or more than the limit, the computer will refuse to accept it: it will display the word 'reject', followed by a repeated request 'U'. If a valid number is entered, the computer will perform the subtraction and display the result: 'Sxx', where xxxx is the remainder. It then calculates the number that it wants to subtract, and displays this with the prefix 'I'; finally, it performs this subtraction and again displays the result as 'Sxxxx'. It is now the human player's turn, and the game continues until the remainder becomes equal to 0. Depending on who reached this point, the display will indicate either 'I LOSE' or 'U LOSE'. The program can be re-started by operating the Halt/Reset key.

Program 4: Runtext
(R. Brinkmann)
This program can display up to 16 different lines of text, each consisting of up to 256 characters, as a 'running script' on the 7-segment displays.
The start address for the program is 0C00. Initially, 'runtext' appears on the display. One of the keys O...F is now used to select the desired one out of the sixteen texts. Even when a text is running, it is possible to switch over immediately to any other text, by operating the corresponding key.

The program consists of three parts:
1. A selection routine, that uses the Elbog LDKB1 routine to determine which of the texts is required. It places the start address of the text in pointer 2, and the length of the text in a memory location reserved for this purpose (as can be seen from the listing).
2. A display routine, that transfers the text (pointer 2) to the display (pointer 1). This routine also checks to see if a different text is required (key entry); as long as this is not the case, the text originally selected is repeated. The speed at which the text runs across the displays can be varied within wide limits by modifying the contents of addresses 0D48 and 0D57.
3. The text section, containing the texts in F-segment format. Each character is stored in one memory location (8 bits). The texts all start with seven spaces (00), so that a new text always starts on a blank display.

When this program is loaded from the ESS record, not only sections 1 and 2 (as given in the listing) are entered, but also several texts. For this reason, the memory is used up to and including location 0E33.

Program 5: Biorhythm (H. Prante)
A few years ago (in October 1971), Elektor published a program for calculating biorhythms on an HP65 calculator. Now, a similar program is available for the SC/MP system.
As usual, the program is started at address 0C00. Initially, the word 'today' appears; the date for which the biorhythm data are required should now be entered. The date should be entered in the following order: day, month, year (without '19'). This entry is immediately followed by the display 'birthday'; this date is entered in the same way.
The computer performs the necessary calculations and displays the results: three numbers, corresponding to the physical, emotional and intellectual rhythms. A new calculation can be performed after operating the Halt/Reset key.
The biorhythm theory was explained in the earlier article referred to above, but a brief reminder may be in order. The physical rhythm has a cycle of 23 days; the emotional cycle is 28 days and the intellectual cycle lasts for 33 days. The 'zero crossings' are critical days, and these include the half-way marks: 11th, 12th day for the physical cycle, 14th for the emotional and 16th/17th for the intellectual. The first half of each cycle is taken to have a positive influence; the second half is negative.

Program 6: Tracer (J. Fischer)
This program is a powerful extension of
the monitor software already available in the SC/MP system. The CPU routine in Elbug can only handle one breakpoint, and it must be reset every time it is used.

‘Tracer’ constitutes a much more powerful aid when de-bugging programs. It can be used to execute any other program in a ‘single-step’ mode. The program under test is thus executed instruction-by-instruction; between instructions, the contents of all registers can be examined (P1, P2, P3, Accu, Extension Register, Status Register). The display gives information on the position of the program counter and the following instruction, before actually executing it. If arrows are noticed at this point, it is possible to correct them before continuing the single-step scan.

The single-step mode can be executed in three ways:

1. **High Speed**: The program to be tested is executed at a rate of approximately one instruction per millisecond, until a specified address is reached. At this point, ‘Tracer’ automatically switches to the ‘Low Speed’ mode. The display is blanked during the High Speed mode.

2. **Low Speed**: The address and the corresponding instruction are displayed for approximately one second. The instruction is then executed, and the display is blanked for one second. This sequence is repeated until the point is reached where the change-over to ‘Manual Step’ is required. This will occur automatically at a specified address; however, it is possible to effect an earlier switch to the Manual Step mode by operating any of the keys during the Low Speed mode.

3. **Manual Step**: The next address and corresponding instruction remain visible in the display until one of the keys (any key except the CPU-routine key) is pressed. The address and instruction remain visible for about one second after the key is pressed; the instruction is then executed and, after a brief delay, the next address and instruction appear on the display. In all three modes, the keyboard and display remain available for input or output of data.

When ‘Tracer’ is started (at address 0000), the message ‘SS...’ appears on the display. Three addresses should now be entered, in the following order:

1. The ‘start address’ of the program to be tested;
2. The address at which the change-over to Low Speed is required;
3. The address at which the Manual Step mode must be initiated.

After the third address has been entered, operating any one of the keys starts the Tracer routine. The first section of the program will be run through in the High Speed mode, unless the second address is equal to the start address (11).

In the High Speed mode, the keyboard and display seem to function normally. In the Low Speed and Manual Step modes, this becomes rather more complicated.

In the Low Speed mode, the keyboard must be operated in the time that the instruction (and address) are visible on the display. In the Manual Step mode, the keyboard becomes operational when the command is given to execute the instruction; it remains available for approximately one second, until the display is blanked and the instruction is executed.

The time during which the display is blanked by Tracer (for one second after the instruction is executed) is used to show the display that the program under test would provide after that instruction is executed. However, it should be noted that the display is again used by Tracer before coming to the next instruction, so that all previous display data is lost and the ‘program display’ can therefore consist only of single digits.

Both the Low Speed and Manual Step modes can be interrupted to check the contents of all registers in the CPU. In the Low Speed mode, it is first necessary to switch over to Manual Step, by operating one of the keys. The display will then show the address and instruction that is about to be carried out. If the CPU-routine key is now operated, the display ‘CPU’ will appear. The keyboard can be used at this point to select one of the registers; the codes are the same as those used in the Elbug CPU routine: ‘1’ = Pointer 1, ‘2’ = Pointer 2, ‘3’ = Pointer 3, ‘5’ = Status Register, ‘A’ = Accu, ‘E’ = Extension Register.

There are various ways to leave the CPU routine:

- (Subtract) key: Tracer can be re-started.
- (Run) key: Return to High speed. Tracer now expects the entry of two addresses: one to indicate the point at which it must switch over to Low Speed and one which specifies the first address of the Manual Step mode.

Having re-started Tracer in either of these ways, all the facilities described above are available again.

**Program 7: Disassembler**

(F. de Bruijn)

A disassembler is a program that can be used to obtain listings (without comments, obviously) of programs in machine language. It is the opposite of an assembler program.

The listing can be obtained on a printer or an (Elek-) terminal. In the latter case, of course, no ‘hard copy’ of the print-out will be obtained.

The serial output signal for the printer or video display is available at flag 0. The transmission rate is 300 baud. This speed can be modified, if required, according to the following table:

<table>
<thead>
<tr>
<th>address</th>
<th>baud</th>
<th>baud</th>
<th>baud</th>
<th>baud</th>
</tr>
</thead>
<tbody>
<tr>
<td>159 B</td>
<td>97</td>
<td>64</td>
<td>25</td>
<td>86</td>
</tr>
<tr>
<td>159 D</td>
<td>17</td>
<td>96</td>
<td>03</td>
<td>01</td>
</tr>
<tr>
<td>15A7</td>
<td>89</td>
<td>70</td>
<td>50</td>
<td>31</td>
</tr>
<tr>
<td>15A9</td>
<td>08</td>
<td>62</td>
<td>01</td>
<td>00</td>
</tr>
</tbody>
</table>

The (1½ K) program offers the following facilities:

a) enter the ‘begin address’ of the program that is to be ‘disassembled’;
b) specify the begin and end addresses of a table;
c) mark a byte used by the program by entering ‘20’ at that point;
d) enter the number of consecutive lines to be disassembled.

The program is started at address 1000. When ‘D1...’ appears on the display, the begin address can be entered. This can be followed, if necessary, by specifying one table; in that case, the Block Transfer key must first be operated.

The next step is to specify the number of lines to be printed: note that this number must be entered in hexadecimal.

A suitable value, when using the Elek terminal for the display, is 0010. The maximum value is 00FF; this already makes for quite a lengthy print-out.

The program will start the print-out immediately after receiving this final entry; it will stop when the specified number of lines have been disassembled.

A further group of lines will be disassembled if the Halt/Reset key is operated.

If the program finds an instruction that it doesn’t recognise, it will print ‘?’. Jump instructions by means of the program counter are shown with the address to which the jump would be executed. The same applies to other instructions that use the program counter.
In most cars the engine speed (r.p.m.) is displayed on an analogue scale. However, a digital readout, using seven-segment displays, is also perfectly possible.

The circuit shown here provides a two-digit display calibrated in hundreds of r.p.m. per minute, i.e. 6000 r.p.m. will produce a readout of 60. There are two principal reasons for restricting the display to two digits. The first is quite simply that accuracy greater than this is not necessary, and secondly, a much longer gate time would be required otherwise, with the result that the counter would not be able to follow sudden changes in the engine speed.

The circuit is a modernised version of a rev counter published in an earlier issue of Elektor (see Elektor 1, December 1974). The input signal is derived from the contact breaker; the amplitude of the resulting pulse train being limited by zener diode D1 and then 'shaped' by T1 and the monostable N1/N2. The pulses are counted by IC3 (dual decade counter), whose outputs are connected to two BCD-to-7-segment latches/decoder-drivers. The reset pulse for the counters (i.e. the timebase signal) and the latch enable pulse are provided by a 555 timer (IC5).

The circuit has three adjustment points. Preset potentiometer P1 sets the width of the reset pulse. In the majority of cases it will be sufficient to set this potentiometer to the mid-position. However it may happen that the reliability of the circuit can be improved by choosing an alternative position. The latch period, and hence the rate at which successive measurements are displayed, is set by means of P2. Finally, P3 is used to calibrate the counter. This can be done using either a tone generator with a calibrated tuning scale, or else by using a mains frequency signal. In the former case the frequency of the input signal will depend upon the type of engine with which the rev counter is to be used. The counter is calibrated for a nominal r.p.m. of 6000, and depending upon the number of contact breaker pulses produced for each revolution of the engine in question, a signal of suitable frequency (see table 1) is fed to the input of the circuit and P3 adjusted until a readout of 60 is obtained. If a tone generator is not available, a low voltage signal of mains frequency (e.g. from a doorbell transformer) can be used. P3 is then adjusted until the appropriate readout is obtained (see table 1, 'revs at 50 Hz').
The range of digital test equipment is growing ever more extensive. Voltage, current, frequency, resistance, temperature—all these quantities are now commonly measured, and displayed, digitally. This not only applies to 'professional' applications, even the 'amateur constructor' has gone digital (see, for example, the 'universal digital meter', Elektor 45). Now it is time to add a digital capacitance meter to the range—the 'digifarad'.

The block diagram of the 'digifarad' is shown in figure 1. $C_X$ represents the unknown capacitance to be measured. Depressing the 'start' button momentarily closes the electronic switch, ES, so that $C_X$ is charged to a given voltage (Uc). When ES reopens, $C_X$ is discharged by a constant current source (I), with the result that the voltage on $C_X$ falls in a linear fashion. All other things being equal, this discharge rate is determined by the value of $C_X$. The voltage on the capacitor is monitored by a window comparator, formed by two op-amps and a set/reset flip-flop. For the period that $U_C$ remains within the upper and lower reference voltages (U1 and U2) of the 'window', the output of the comparator is low. This enables a three-digit counter, which counts the number of pulses from a clock generator. Thus the greater the capacitance of $C_X$, the longer $U_C$ takes to fall below the threshold voltage of the window comparator, and the more pulses counted by the counter. Finally, by varying the size of the constant current, I, we can arrange for capacitors of widely differing value to be measured in the same way.

The complete circuit diagram of the digifarad is shown in figure 2, and a pulse diagram is given in figure 3. The latter is not only useful in the (unlikely) event that trouble-shooting proves necessary; it is also a great help in the following explanation of the circuit. The various wave-shapes were measured at the corresponding points in the circuit.

It is not too difficult to relate the block diagram, given in figure 1, to the actual circuit shown in figure 2. The constant-current source, I, is formed by op-amp A1 and transistor T1. The size of the current is determined by the position of the range switch, S1 (see table 1). The op-amp varies the current through T1 and the selected range resistor so as to ensure that the voltage at the inverting input is always the same as the fixed reference voltage at the non-inverting input.

The electronic switch, ES, consists of transistor T2, which is turned on via the start button, S2, and flip-flop N3/N4. The voltage on $C_X$ is buffered by op-amp A2, and fed to the window comparator formed by A3 and A4, N1, N2, C1, C2, R18 and R19 form a set/reset flip-flop which is triggered by changes in the output state of the window comparator. When $C_X$ is fully charged, the outputs of A3 and A4 are both high.

Given the fact that many types of capacitor—especially electrolytics—have a wide tolerance (20% is fairly common), it is often desirable to be able to measure capacitances both quickly and with a reasonable degree of accuracy (e.g. when constructing precision timer circuits, matching the time constants of several RC networks, etc.). Of course a capacitance meter also enables one to measure the value of those piles of unmarked capacitors which end up at the bottom of one's junk box, or to test 'suspect' capacitors for potential faults—in short it represents a useful addition to the test gear of any constructor.

The circuit described here offers the advantages of a digital display, has 5 decade ranges, measuring from 1 nF to 9.999 µF, and is accurate to about 2%.

J. Guthrie
However when the voltage on \( C_x \) reaches the upper threshold of the 'window' (i.e. the voltage on the non-inverting input of A2 falls below that on the inverting input) the output of A2 goes low, with the result that the output of N2 also goes low, enabling the counter. As the unknown capacitance continues to discharge, the voltage on \( C_x \) will reach the lower threshold of the window, whereupon the output of A4 will go low, taking the output of N2 high and stopping the count.

In addition to turning on T2, the second flip-flop formed by N3 and N4 provides the reset and display enable signals for the counter. The display is inhibited during the count cycle, thus ensuring a stable readout. R20, C3 and the two diodes (D1 and D2) ensure that the two flip-flops assume the correct state upon switch-on. The clock-signal for the counter is provided by a 555 timer (IC3) connected as an astable multivibrator. The counter itself (IC6) is a single IC, type 74C928. It performs the 7-segment decoding, and drives the three LED displays via transistors T4...T6. The displays are of the common cathode type (e.g. HP 50B2-7760, DL 704, etc.). In all, four 'supply' voltages are required for the circuit: the reference voltage \( U_{ref} \) and the 16 V, 12 V and 5 V supplies. The obvious solution is to use IC's: one three-pin regulator (IC5) takes care of the 12 V supply, and a 'basic' 723 circuit (IC4, T3) provides all the other voltages, including the reference voltage.

**Construction**

Once again, printed circuit boards (available through the EPS service) reduce constructional problems to a minimum. Every single component, barring the mains transformer, is mounted on these boards—from supply circuit to displays. To increase the sense of achievement, three boards are required instead of one. A display board (figure 4c) is mounted behind the front panel, and the other two boards (figures 4a and 4b) are bolted together with spacers in a sandwich construction and mounted behind the display board. The display board contains the displays. Obviously, it also provides space for IC6, resistors R31...R37, switches S1, S2 and S3; furthermore the on/off indicator D8 and 'banana plug' connection sockets for the unknown capacitator \( C_x \). The upper board in the sandwich (figure 4b) is intended for the supply circuit (all except IC5) and the clock generator, IC3. Finally, the lower 'sandwich' board (figure 4a) must provide space for the remainder of the circuit. Rest assured: it does. The interconnections between the various boards are clearly marked with diagonal arrows.

![Figure 1. Block diagram of the digital capacitance meter. The unknown capacitor, \( C_x \), is discharged by a constant current, \( I \). The longer the discharge period the more pulses counted.](image1)

![Figure 2. Complete circuit diagram. Common cathode type displays are used.](image2)
Figure 3. In this pulse diagram, the letters A...I refer to the corresponding test points indicated in figure 2.

Table 1

<table>
<thead>
<tr>
<th>S1 position</th>
<th>measurement</th>
<th>scale</th>
<th>factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>current</td>
<td>range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1 µA</td>
<td>999 nF</td>
<td>1 nF</td>
</tr>
<tr>
<td>2</td>
<td>10 µA</td>
<td>9.99 µF</td>
<td>0.01 µF</td>
</tr>
<tr>
<td>3</td>
<td>100 µA</td>
<td>99.9 µA</td>
<td>0.1 µF</td>
</tr>
<tr>
<td>4</td>
<td>1 mA</td>
<td>999 µA</td>
<td>1 µF</td>
</tr>
<tr>
<td>5</td>
<td>10 mA</td>
<td>9.99 mF</td>
<td>10 µF</td>
</tr>
</tbody>
</table>
Final notes

The capacitance meter is as easy to use as a multimeter: switch it on with S3, select the desired range with S1, connect the unknown capacitor, press the start button (S2) and watch the result appear on the display. The measuring ranges are listed in the Table; the current listed in this table is the constant-current used to discharge $C_X$. If the capacitance value is completely unknown, it is a good idea to start in the highest range (position 5), and then switch back step-by-step until a useful reading is obtained.

The circuit contains only one calibration point, namely preset potentiometer P1. Calibration can be carried out with the aid of a close tolerance capacitor of a known value. Silvered mica capacitors, for example, typically have a tolerance of 1%.

One final remark. If a 12 V/1 A transformer is felt to be rather heavy, or if a smaller transformer happens to be available, resistors R31...R37 can be modified as required. Provided a slightly less brilliant display is considered adequate, the value of these resistors can be increased to 22 Ω; a 12 V/½ A transformer is then good enough.

Figure 4. The three printed circuit boards required. The main board (figure 4a) and the supply and clock generator board (figure 4b) are mounted in a sandwich construction, using spacers. They are then coupled to the display board (figure 4c) that is mounted behind the front panel.
short-interval light switch

Even in today's well-equipped modern houses there are various 'corners' where additional lighting is required. For dark cupboards, meter boxes etc. temporary lighting is usually sufficient, so that making a connection to the mains is hardly worthwhile; a simpler and cheaper solution is to use a battery-powered circuit which will light a lamp for a short period of time. As is apparent from the accompanying circuit diagram, such a circuit is by no means complicated. Using only one CMOS IC, three resistors and one capacitor, the circuit will switch on a lamp for a presettable interval.

The operation of the circuit is perfectly straightforward: when the button is pushed C1 charges up to the supply voltage. The outputs of the four parallel-connected inverters (N3...N6) are then low, so that the lamp will be off. When the button is released, C1 discharges via R1 until the input of N1 reaches half supply. The Schmitt trigger formed by N1 and N2 then changes state, with the result that the lamp is extinguished. The positive feedback resistor R3 ensures that the Schmitt trigger changes state very quickly.

With the resistor values shown in the circuit diagram, the lamp will remain lit for roughly 2.5 seconds per μF of C1.

Thus a 10 μF capacitor would give an interval of roughly 25 seconds.

The circuit can be powered by four 1.5 V cells connected in series. If a larger lamp is required, three 4.5 V cells connected in series can be employed. Alternatively, for really 'heavy-duty' applications, the four parallel-connected inverters can be replaced by a transistor, as shown in figure 2. The supply voltage should be matched to the voltage rating of the lamp and may lie between 4.5 and 15 V. The current through the lamp should not exceed 500 mA in that case.

The design for the 'variable fuzz box' was first published in the December 1978 issue of Elektor. Such is the popularity of this circuit, that we have decided to produce a printed circuit board for it.

The variable fuzz box is a special effects unit for guitarists, which by allowing the amplifier signal to be clipped in a variety of different ways (symmetrically, asymmetrically, soft, hard, etc.) offers a greater degree of control over the resultant sound. The circuit of the fuzz box was described in detail in the original article, hence will not be repeated here. However one connection to the original description has to be added: symmetrical clipping of the output signal produces only uneven (not even, as was stated) harmonics, whilst asymmetrical clipping generates both even and uneven harmonics in the output signal.

The alternative circuit diagrams of the fuzz box for symmetrical (figure 3 of original article) and asymmetrical (figure 4 of original article) power supplies are here combined into one (see figure 1). The circuit diagram contains a number of lettered connection points (a...h, j, k, m...w) which are marked on the printed circuit board shown in figure 2. The circuit diagram and accompanying parts list provides the relevant details on which connections should be made for either symmetrical or asymmetrical power supply requirements.

The current consumption of the circuit is less than 20 mA. A 741 can be used for IC1, however an LF 356 is a better choice.

Literature:
Variable Fuzz Box, Elektor 44, December 1978
1

Figure 1. Circuit diagram of the variable fuzz box for symmetrical and asymmetrical power supply stages.

2

Figure 2. Printed circuit board for the variable fuzz box. The one board is suitable for both versions of the circuit.

parts list

Resistors:
R1,R91,R101 = 10 k
R2,R3 = 100 k
R4,R5,R6,R7,R8 = 4k7

Capacitors:
C1 = 470 n
C2,C3 = 100 n
C41 = 22 μF/25 V
C52 = 10 μF/25 V
C62 = 2μF/40 V

Wire links:
asymmetrical supply voltage
+ 20 ... 30 V: points q and t
symmetrical supply voltages
± 10 ... 15 V: points q and k
C5 replaced by link
C6 replaced by link

Potentiometers:
P1,P2 = 4k7 (5 k) lin.
P3,P4 = 100 k lin.
P5 = 47 k (50 k) log.

Semiconductors:
IC1 = 741 or LF 356 (see text)
T1 = TUN
T2 = TUP
D1,D2 = DUS

Remarks:
*omitted in the case of
symmetrical supply voltages.
*replaced by wire link in the case of symmetrical supply voltages.
A grid-dip meter (the name harks back to the good old days of valves, which actually had a wire grid between their anode and cathode and the meter measured the grid current of the valve) is a useful little device which enables the resonant frequency of tuned circuits to be determined without having to make any electrical connection to the circuit in question. The grid-dip meter contains a coil, which forms part of a variable-frequency oscillator. The coil is held near the parallel-resonant circuit (the equipment containing the tuned circuit should be switched off for the purposes of the measurement). Series-resonant circuits can also be measured by shorting their inputs, so that a parallel-resonant circuit is obtained. The coil of the grid-dip meter is electromagnetically coupled to the resonant circuit. As the frequency of the oscillator approaches the resonant frequency of the LC circuit, so the oscillator becomes increasingly damped. This is registered by the meter, so that when the needle deflection is at a maximum, and the oscillator frequency coincides with the resonant frequency of the tuned circuit, the latter can simply be read off a calibrated scale.

The circuit of the gate dipper described here is based upon a device known as a lambda diode. As many readers may well never have heard of such a ‘beast’, it is worth devoting a little time to an explanation of this slightly unusual circuit element.

**Lambda diode**

If the term lambda diode is unfamiliar, the majority of our readers will have heard of tunnel diodes. These are diodes which exhibit a negative resistance over a certain portion of their voltage-current characteristic. The concept of a negative resistance may seem confusing, but in fact it is quite straightforward. As the
This voltage is smoothed by C4/R2 and fed to T3, which is connected as a source follower. Potentiometer P2 is adjusted such that a zero reading is obtained on the meter. If the coil, Lₓ, is brought near the passive tuned circuit which is to be measured, the negative voltage across D3 will fall as the oscillator is increasingly damped. This causes the source voltage of T3 to rise, thus causing a deflection on the meter. When the deflection is at a maximum, the value of C3 is an index of the resonant frequency of the tuned circuit under test. Due to the effect of the lambda diode, the behaviour of the meter needle is the opposite to that of other types of grid-dip meter, where the oscillator frequency is adjusted for minimum deflection (hence the term dip meter).

The grid-dip meter can also be used to check the operation of an oscillator. Once again the coil of the meter is held near the oscillator circuit, and C3 is adjusted until audible beat frequencies are obtained. These low frequency beat notes are not sufficiently smoothed to prevent them appearing at the source of T3, with the result that they are fed through to the output stage round T4 and T5, where they can be heard via a pair of headphones. P3 then functions as a volume control.

When checking the operation of tuned circuits in radio receivers, if the grid-dip meter is tuned for zero beat, then it is possible to modulate the r.f. signal (in accordance with the direct-conversion principle). The lambda diode oscillator then functions as a self-oscillating mixer stage. This fact allows the meter to be calibrated with a precise frequency scale (the procedure is described in detail in the section on calibration).

Construction

The track pattern and component overlay of the printed circuit board for the grid-dip meter is shown in figure 5. The coil, Lₓ, is not mounted on the board, but rather is connected to the
Figure 4. Complete circuit diagram of the gate dipper. FET T1 and the bipolar transistor T2 form the lambda diode. At first sight this configuration is different to that shown in figure 3. However from the point of view of AC currents, the base of T2 is connected to the drain of T1, and the gate of T1 is connected to the collector of T2. Thus the two circuits are equivalent for the purpose of AC.

Figure 5. Track pattern and component layout of the printed circuit board for the gate dipper (EPS 79514). The coil, Lx, is not mounted on the board, but rather is wound on a DIN loudspeaker plug, the socket of which is mounted on the case of the meter. In this way it is a simple matter to plug in different coils to obtain different measurement ranges.

Parts list:
Resistors:
- R1, R5 = 220 k
- R2 = 100 k
- R3 = 3 k
- R4 = 82 k
- R6 = 330 k
- R7 = 2 k
- P1 = 22 k lin
- P2 = 2 k lin
- P3 = 47 k log

Capacitors:
- C1, C5, C7, C13 = 22 n
- C2 = 47 p
- C3 = 220 p, variable
- C4, C10 = 100 p
- C6, C8 = 1 μ/10 V
- C9 = 10 μ/10 V tantalum
- C11 = 22 μ/6.3 V
- C12 = 1 n
- C14 = 10 μ/16 V

Semiconductors:
- T1 = BF 256B
- T2 = BF 451
- T3 = BF 256A
- T4, T5 = BC 549C
- IC1 = 78L05
- D1 . . . D4 = DUS

Miscellaneous:
- Lx, see text and table
- M1 = meter 225 μA (or less)
- S1 = on/off switch
- 8 DIN loudspeaker plugs
- 1 socket for loudspeaker plug
circuit via a plastic DIN loudspeaker plug. This provides the option of using several different coils to obtain different measurement ranges. The accompanying table lists the winding details for each coil and the corresponding frequency range.

The coils are wound on the plugs as far as possible from the metal terminals (see figure 6). If the coil is near any metal, eddy currents will cause energy losses which increase with frequency. The result is that after adjusting C3, the zero point of the meter will tend to drift. Admittedly, that is not such a disaster, since the meter is not being read, but merely used as an indicator to obtain the correct position for C3. However if the energy losses are severe enough, the meter will deflect to the point where no 'dip' is obtained.

The ends of the coil are fed through the inside of the plug and soldered to the terminal pins. The coil consists of only one turn is mounted directly on the pins, and the plastic cap is omitted. The socket for the plug is mounted on the case of the grid-dip meter and connected to the printed circuit board via short lengths of fairly thick wire. In this way it is a simple matter to interchange coils should a different measurement range be required. The variable capacitor, C3, is also mounted off-board and connected to the circuit via short, thick wiring. If the wires are too long, measurements above roughly 80 MHz are no longer possible.

Calibration and use

Before providing the gate dipper with a calibrated scale one must first know how to use it properly. P1 and P2 are set so that as positive a 'dip' as possible is obtained. The meter is here used essentially as an indicator, rather than as a measuring instrument. Thus at this stage P2 is adjusted not so much in order to set the zero point of the meter but rather to ensure that the needle remains within the scale range of the meter. Thus P2 can be adjusted to compensate for energy losses induced by metals in the vicinity of the coil etc.

As already mentioned, P1 in fact deter-

mines the biasing of the lambda diode, and hence the sensitivity of the circuit. The optimum setting of P1 can be determined as follows:

The wiper of P1 is turned fully towards the cathode of D1. The oscillator is then inoperative and the meter deflection at a maximum. Ensure that the needle is not hard up against the end stop, however (if necessary adjust P2 accordingly). Now turn the wiper of P1 in the opposite direction. At a certain point the needle deflection will decrease (the oscillator is now running). Continue to turn P1 until the deflection is at a minimum (here again it may be necessary to adjust P2). The meter range is then set between these two extremes by adjusting P2 (note, P2 will need to be readjusted when the coil, L, is changed).

To gain proficiency in using the meter it is advisable to practice with a tuned circuit whose resonant frequency is already known. At the same time one can experiment with different settings of P1 to obtain optimum sensitivity.

Once accustomed to using the meter, one can proceed to provide a calibrated scale for the variable capacitor C3. For this, the gate dipper is used as an AM demodulator. A length of wire (minimum 10 metres) which can be positioned either horizontally or vertically is used as an aerial. The latter is coupled to the coil of the grid-dip meter via a single-turn coil (see figure 7). One end of the coupling coil should be earthed (to for example a water pipe etc.). Capacitor C3 is then adjusted until a known AM station can be heard via the headphones. The oscillator frequency will then be the same as the carrier wave frequency of the transmitter. The scale for the variable capacitor can be calibrated simply by tuning into a number of different stations. If desired, higher frequencies can be calibrated by employing a number of tuned circuits of known resonant frequency.

The position of P1 at which reception is the strongest corresponds to the position which gives maximum sensitivity when using the circuit as a grid-dip meter. To facilitate tuning, it is recommended that a tuning capacitor with slow motion drive be used.

<table>
<thead>
<tr>
<th>No. of turns</th>
<th>Ø Cu wire</th>
<th>frequency range</th>
</tr>
</thead>
<tbody>
<tr>
<td>230</td>
<td>0.1 mm</td>
<td>374 kHz ... 871 kHz</td>
</tr>
<tr>
<td>110</td>
<td>0.1 mm</td>
<td>701 kHz ... 1616 kHz</td>
</tr>
<tr>
<td>47</td>
<td>0.2 mm</td>
<td>1536 kHz ... 4326 kHz</td>
</tr>
<tr>
<td>23</td>
<td>0.2 mm</td>
<td>2712 kHz ... 7224 kHz</td>
</tr>
<tr>
<td>12</td>
<td>0.8 mm</td>
<td>6777 kHz ... 21,2 MHZ</td>
</tr>
<tr>
<td>5</td>
<td>0.6 mm</td>
<td>12,6 MHz ... 45,6 MHz</td>
</tr>
<tr>
<td>2</td>
<td>0.8 mm</td>
<td>27 MHz ... 80 MHz</td>
</tr>
<tr>
<td>1</td>
<td>1.0 mm</td>
<td>50 MHz ... 150 MHz</td>
</tr>
</tbody>
</table>
Although the device described here is called a strain gauge, it is in fact being used to measure stress, i.e. the forces which are applied to it. Strain denotes the deformation of a material (change in form or bulk) as a result of the action of stress. However in all elastic materials (such as e.g. steel) there is a linear relationship between stress and strain, which is expressed by the following equation: $\delta = \varepsilon \cdot E$, where $\delta$ is the stress, $\varepsilon$ is the strain, and $E$ is a coefficient termed the modulus of elasticity. Every elastic material has its own modulus of elasticity which remains constant within certain limits of stress. Since strain is proportional to stress, it is thus possible to measure the one via the other.

The basic design of the strain gauge is shown in figure 1. An electrical signal is derived from a transducer. This signal is then amplified and used to drive an LED scale display. If one looks ahead to figure 3, it can be seen that the electronics involved are in fact extremely simple. The heart of the strain gauge is the stress absorber, the object upon which the forces to be measured actually act, and whose strain is measured. This part of the device cannot be bought, and must be made oneself.

### Stress absorber

As is apparent from figure 2, the object which bears the brunt of the forces to be measured is formed from a sheet of suitable metal, with a hole drilled in each end. The central portion is made narrower than the top and bottom, since it is at this point that the deformation of the metal is measured.

The amount of strain is actually measured by a special type of transducer called an electric resistance strain gauge. In its simplest form it consists of a grid of resistance wire cemented between two sheets of paper. The gauge is bonded to the metal, so that it undergoes the same deformations. The resultant changes in the length and cross-sectional area of the wire causes a proportional change in its resistance.

As figure 2 makes clear, four resistance strain gauges are mounted in a bridge configuration, two on the front of the stress absorber and two on the back. The changes in the resistance of the vertically oriented gauges (R2 and R3) are summed, whilst the horizontally oriented gauges provide temperature compensation. A further advantage of this arrangement is that flexing of the metal in the lateral plane will have no effect, since the bridge remains in equilibrium.

The bridge is provide with a stabilised supply voltage. A current of roughly 20 mA can flow through the strain gauges, and since they have a resistance of approximately 120 Ω, the voltage across the bridge is fixed at roughly 5 V.

### Circuit

The circuit of the strain gauge is shown in figure 3, and, as has already been mentioned, is fairly modest in dimension.

The low level output voltage of the measuring bridge must be considerably amplified before it can be displayed. This is performed by two 741 ICs, each of which contains two 741 type op-amps (it is of course also possible to employ four conventional 741's). A1 and B1 are connected as unity-gain amplifiers with high input impedance, so that the bridge is not loaded by the amplifier circuit.

The latter is formed by A2 and B2, which are connected as a differential amplifier with a gain of approximately 1000, adjustable by means of P2. Under quiescent conditions (no force applied to the gauge), P1 is adjusted for zero output voltage.

The display takes the form of a column of LEDs, which are driven by the well-known UAA 170 LED voltmeter IC. Depending upon the input voltage, this chip lights one of the LEDs D3...D18. The input is protected against negative and excessively large positive voltages by zener diode D2.

The power supply circuit is also quite straightforward. Two integrated voltage regulators (7812 and 7912) provide the + and -12 V rails, whilst the 5 V for the resistance bridge is obtained by the inclusion of two resistors (R9, R10) and a zener diode (D1).

### Construction

The amplifier, display driver and displays can easily be mounted on a strip of

W. van Dreumel
Veroboard, or similar. The stress absorber, however, is slightly more complicated, since it involves a certain amount of mechanical handiwork. The dimensions of the stress absorber will depend upon the type of material used and upon the desired measurement range. To obtain optimum sensitivity, the material should undergo as great a deformation as possible when under maximum load conditions. As can be seen from column 3 of table 1, the most suitable material from this point of view is hard brass, with duraluminium a good second. Column 2 of the table is used to calculate the cross-sectional area of the stress absorber (X x Y in figure 2). This is done by dividing the maximum permissible stress into the required range of forces to be measured. The ratio of X to Y can be chosen individually, however X should not be smaller than approximately 10 mm (because of the size of the strain gauges) and the overall shape of the stress absorber should remain similar to that shown in figure 2. The values of R6 and P2 in the circuit diagram are calculated on the basis of a stress absorber made of duraluminium and with a cross-sectional area of 20 mm².

Although electric-resistance strain gauges are not widely used by the amateur, various types are available commercially. For this particular application their dimensions should be in the region of 5 x 10 mm. Suitable types are (among others) the EA-XX-2508G-120 from Micro Measurements, the 3/120 LY 11 from HBM, and the PR9833 k/01 from Philips.

Calibration

Under zero load conditions P1 is adjusted such that the first LED in the scale lights up. A known weight is then suspended from the gauge and P2 adjusted until the corresponding LED lights up (obviously this will depend upon the measurement range chosen). If, for example, 10-turn potentiometers are used for P1 and P2, a fairly accurate scale can be obtained. For a variety of reasons, it is possible that the zero point of the scale may tend to fluctuate. However if P1 is mounted such that it is accessible externally, this should not present too many problems.

Table 1. The information contained in the table allows the suitability of various metals to be assessed, and the cross-sectional area of the 'stress absorber' to be calculated in each case.

<table>
<thead>
<tr>
<th>Material</th>
<th>Modulus of elasticity E; kg/mm²</th>
<th>Maximum permissible stress δ; kg/mm²</th>
<th>Maximum permissible stress δ%;</th>
</tr>
</thead>
<tbody>
<tr>
<td>hard brass</td>
<td>9000</td>
<td>42</td>
<td>0.46</td>
</tr>
<tr>
<td>duraluminium</td>
<td>7000</td>
<td>26</td>
<td>0.37</td>
</tr>
<tr>
<td>semi-hard brass</td>
<td>9000</td>
<td>24</td>
<td>0.27</td>
</tr>
<tr>
<td>hard aluminium</td>
<td>7000</td>
<td>14</td>
<td>0.20</td>
</tr>
<tr>
<td>sheet steel</td>
<td>21000</td>
<td>18</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Literature:

Linear Applications, National Elektor 12, April 1976.

Figure 3. The amplifier and display circuit consists of little more than 3 ICs. The symmetrical supply is provided by two voltage regulator ICs, whilst a couple of resistors and a zener diode provide the 5 V for the measurement bridge.
It is interesting to note that, by and large, our readers' comments and queries — yes, and problems, too — run parallel to our own. It is even more interesting that all our problems have been solved, as will be described. To make full use of a microprocessor, one should normally have access to the instruction manual. For the 2650, this is a 174-page book . . . Fortunately, the main points can be summarised rather more briefly.

I played TV games....

Everything you want to know about making software for the TV games computer, in two easy lessons . . .

In Elektor 48, April 1979, we described how to build a 'TV games computer'. Included was a brief explanation of how it works; the 'instructions for use' consisted of little more than the Read Cassette routine, so that the programs given on ESS records can be entered.

Apparently, however, the majority of our readers want more: they want to do their own programming. 'This will prove relatively easy', we said — and to prove it, the (sometimes fairly sophisticated) programs on the second ESS record for the TV Games computer were developed by a novice. The following article is based on the experience gained . . .

Addressing modes

When fetching or storing data, or jumping to and fro in a program, it is essential to specify the 'address' concerned. Obviously, in the TV Games computer, there are several different ways of doing this.

Absolute or relative

An 'absolute' address is simply the address itself. For instance, in machine language the instruction for 'Load Absolute into register zero' starts with 0C (more on this later!); if the data is to be fetched from address 0F00, the full instruction will therefore be 0C0F00. A 'relative' address, on the other hand, specifies a small jump in the program. Basically, the processor will calculate an 'absolute' address by adding the specified number (between -64 and +63) to the address that follows that particular instruction. As an example, if the instruction 'Load Relative into register zero, 2F' (in machine language: 082F) is located at the two address bytes 093E and 093F, the following address is 0940. The 'absolute' address corresponding to this instruction is therefore 0940 + 2F = 096F, and the data will be fetched from there.

The negative number required for a 'backwards' jump is entered as a '7-bit two's complement number'. In simple language, this means that you count down from 80 in Hex. For instance, if in the previous example the data was to be loaded from address 093D, the relative address would be 7D: the 'following address' 094D corresponds to 80, so 093F corresponds to 7F, 093E to 7E and 093D to 7D. The full instruction is therefore: 087D. Note that this way of specifying negative numbers means that FF . . . 3F are positive; 40 . . . 7F are negative; greater than 7F don’t exist.

All this may or may not seem simple in theory; in practice it has proved a source of endless programming errors . . . It is easier to miscalculate a relative address than to get it right! For simple programs, one may as well use 'absolute addressing' — the additional memory space required (the corresponding instructions are longer) is rarely a problem.

However, practice makes perfect, and as programs get more complex it becomes worthwhile to start using relative addresses wherever possible. As an aid to the beginner, one of the programs on the new ESS record contains a calculation routine for relative addresses — a useful check!

Direct or indirect

The two types of addressing explained above are both referred to as 'direct' address modes: data is transferred from or to the specified address. An alternative possibility is a two-step operation: specify an address where the desired address can be found. This is referred to as 'indirect' addressing.

Although both absolute and relative indirect addresses are possible, only the latter are useful in the basic TV games computer. A relative address is converted to an indirect relative address by adding 80. In the example given above, the 'load relative' instruction 082F was located at addresses 093E and 093F; the data was then fetched from address 096F. However, if the instruction is modified to 08AF (2F + 80 = AF) the
Table A.

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00F0</td>
<td>C0 60 50 CE</td>
<td>DATA</td>
</tr>
<tr>
<td>00F4</td>
<td>3D CE 50 60</td>
<td></td>
</tr>
<tr>
<td>00F8</td>
<td>C0 00 28 FF</td>
<td></td>
</tr>
<tr>
<td>00FC</td>
<td>63 FE 00 00</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td>7620</td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td>05C3</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>0004</td>
<td>0400</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>0005</td>
<td>CDF5F0</td>
<td>STRA, I-R1</td>
</tr>
<tr>
<td>0008</td>
<td>597B</td>
<td>BRNR, R1</td>
</tr>
<tr>
<td>0009</td>
<td>00E0</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>000D</td>
<td>0048F0</td>
<td>LODA, I-R1</td>
</tr>
<tr>
<td>0010</td>
<td>CDF7F0</td>
<td>STRA, I-R1</td>
</tr>
<tr>
<td>0013</td>
<td>5978</td>
<td>BRNR, R1</td>
</tr>
<tr>
<td>0015</td>
<td>0401</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>0017</td>
<td>C1FC00</td>
<td>STRA, R0</td>
</tr>
<tr>
<td>001A</td>
<td>0400</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>001C</td>
<td>C1FC01</td>
<td>STRA, R0</td>
</tr>
<tr>
<td>001F</td>
<td>9C1EE8</td>
<td>LODA, R0</td>
</tr>
<tr>
<td>0022</td>
<td>F420</td>
<td>TMI, R0</td>
</tr>
<tr>
<td>0024</td>
<td>9879</td>
<td>BCFR</td>
</tr>
<tr>
<td>0026</td>
<td>3F05CD</td>
<td>BSTA, UN</td>
</tr>
<tr>
<td>0029</td>
<td>1F0014</td>
<td>BCTA, UN</td>
</tr>
</tbody>
</table>

As things stand, these two instructions are unnecessary: the data in address FC1 is already 00 after the 'clear PVI' routine. However, other colours can now be selected by changing the data in the LODI instruction.

This slightly extended 'return to monitor' routine curers the problem of unwanted black squares down the left-hand edge of the screen (see text).

Table A. An illustration of what can be achieved with the instructions described in this article! The program is started at address 0000. If it works, proceed to Table B!

- The shorthand abbreviations given above are not mnemonic.
- They are simply a quick way to jot down what the instruction does.
- This binary section of program is executed as follows: First, the 'index register' R1 is loaded (LODI, R1 = Load Immediate, Register 1 more on this later) and 00 is loaded into Register 0. This is followed by the 'Store Absolute, Indexed to Register 1 with auto-decrement' instruction — incidentally, the value of mnemonics is clearly illustrated here: it is a lot quicker to write STRA, I-R1 than the mouthful full given above. At this point, the value in R1 (2D) is reduced by one and the result (2C) is added to the basic absolute address 1F80 (5F80 = 1F80 + 4000 for 'auto-decrement'). The value in R0 (00) is then stored in the resultant absolute address: 1F80 + 2C = 1FAC. One down, 44 to go! The next instruction, which will be explained in greater detail later, is 'Branch if Register 1 is Non-zero, Relative'. Since R1 is most definitely non-zero (it is still 2C at this point), the 'relative branch' is executed: the program 'jumps back' to the beginning of the previous instruction, as indicated by the arrow. This whole performance is repeated, storing 00 in progressively lower PVI addresses, until the data in contents of addresses 006F and 0070 will be used as the absolute address for this instruction: if the data stored at these addresses is 0A and 00, say, the 'load indirect relative, 2F' instruction will be carried out as if it read 'load absolute from 0A00'.

Once again, for simple programs it is easier, quicker and more reliable to use the corresponding 'absolute' instruction, and forget about the 'relative indirect' mode. As an aid to courageous novices, the calculation routine mentioned above actually gives two results: if the relative jump in the previous examples is calculated, the answer will appear as '2F Or AF' — for direct and indirect, respectively!

Indexed

In contrast to the 'relative' and 'indirect' addressing modes, 'indexed' addressing can prove extremely useful even in the simplest of programs. The basic idea is that the data stored in one of the registers is added to a specified 'absolute' address; the result of this addition is used as the absolute address for the instruction. The register containing the additional data for the address is referred to as the 'index register', and this register must be specified in the instruction. The data are always transferred to or from register zero when indexed addressing is used.

To specify the basic indexing mode, 0600 is added to the absolute address. Thus 066900 is not interpreted as 'load register one from absolute address 6900'; if we assume that the data already in register one is 0A, the instruction will be read as 'load register zero from absolute address 090A' — i.e. from 0900 plus the data in register one.

Two further extensions of this instruction make it invaluable: 'indexed with auto-increment' and 'indexed with auto-decrement', specified by adding 2000 or 4000 to the absolute address. In both cases, the final address is calculated in the same way — by adding the data in the 'index register' to the specified address. The following two instructions are equivalent: 0900 plus the data in register one. 2000 is added to the absolute address, however, before calculating the final address, the data in the index register are increased by one ('auto-increment') or one is subtracted from the data ('auto-decrement').

The value of this instruction is best illustrated in an example. Let us assume that we want to clear all 'background data' in the PVI. This means storing 00 in all addresses from 1F80...1FAC: 45 in all! Instead of using 45 individual 'store absolute' instructions, a single 'store absolute, indexed with auto-decrement' instruction can be used, with a bit of padding:

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0024</td>
<td>052D</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>0025</td>
<td>0400</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>0026</td>
<td>CDF5F0</td>
<td>STRA, I-R1</td>
</tr>
<tr>
<td>0029</td>
<td>597B</td>
<td>BRNR, R1</td>
</tr>
</tbody>
</table>

Table 1.

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0093</td>
<td>054E</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>0095</td>
<td>0600</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>0097</td>
<td>CDF5F0</td>
<td>STRA, I-R1</td>
</tr>
<tr>
<td>009A</td>
<td>597B</td>
<td>BRNR, R1</td>
</tr>
<tr>
<td>009C</td>
<td>0469</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>009E</td>
<td>C1FC06</td>
<td>STRA, R0</td>
</tr>
<tr>
<td>009F</td>
<td>045D</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>00A1</td>
<td>CDF5F0</td>
<td>STRA, I-R1</td>
</tr>
<tr>
<td>00A2</td>
<td>597B</td>
<td>BRNR, R1</td>
</tr>
<tr>
<td>00A4</td>
<td>48</td>
<td>HALT*</td>
</tr>
</tbody>
</table>

* Not the best way to end a program, as we shall see, but good enough for now!
R1 becomes zero. At this point, the BNR, R1, instruction does not result in a jump back, since R1 is zero, and the rest of the program is carried out.

For those who feel like trying out this program, it is more interesting to turn the background on instead of off. In that case, the background and screen colour must also be specified: ‘69’ in address 1FC6 gives yellow on blue. Furthermore, the objects will have to be cleared, since they are also used by the monitor program. A complete program is given in Table 1; the reason for starting at address 0003 (instead of 0000) will be given later.

While on the subject of indexed addressing, one final point should be noted. In general, this mode is available as a variation of all absolute addresses, with the exception of branch instructions. The only two indexed branch instructions, BXA and BSXA, will be discussed further on.

To or from register (zero)

Nearly all instructions involving transfer or manipulation of data require the use of a register. Obviously, the register to be used must be specified in the instruction.

In the examples already given, and Table 1 in particular, this principle is clear. The first byte of each instruction specifies the basic instruction and the register involved. For instance, the basic instruction for ‘Load Immediate’ is 04xx (where ‘xx’ is the data to be loaded); adding the number of the register to this gives the complete instruction: 04xx for Register 0, 05xx for R1, 06xx for R2 and 07xx for R3. In practice, this means that four variations exist for most instructions: one for each register. It also means that the second digit in an instruction specifies the register involved: 0, 4, 8 and C for register 0 (0003, for instance); 1, 5, 9 and D for register 1; and so on.

Finally, some instructions refer to data transfer or manipulation involving two registers, one of which is always register zero. The instruction ‘Load Register 0 from Register 1’, for instance, is 01. Similarly, ‘LODZ, R2’ (to use the mnemonic) is 02. It should be noted that in some cases, but not all (!), both registers can be specified as Register 0. This can sometimes be useful, as will be explained under ‘programming tricks’, next month.

Registers

We have already mentioned ‘registers’ several times. It is now time to take a closer look at them. To put it in a nutshell, a register can be visualised as a memory location inside the microprocessor itself. In the 2650, 8-bit registers are used; this means that they can store any data value from 00 to FF. In all, seven ‘general-purpose’ registers are available: register 0 and two ‘banks’ of three registers (R1, R2, R3 and R1', R2' and R3'). Of these seven, register 0 is always immediately available; at any given moment, however, only one of the register banks (R1...R3 or R1'...R3') is accessible. The other bank, and the

Figure 1.

Program Status Word

<table>
<thead>
<tr>
<th>bit:</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>function:</td>
<td>S</td>
<td>F</td>
<td>I</td>
<td>Not Used</td>
<td>Not Used</td>
<td>SP2</td>
<td>SP1</td>
<td>SP0</td>
</tr>
<tr>
<td>hex code:</td>
<td>80</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>08</td>
<td>04</td>
<td>02</td>
<td>01</td>
</tr>
<tr>
<td>S Sense</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I Interrupt Inhibit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit:</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>function:</td>
<td>CC1</td>
<td>CC0</td>
<td>IDC</td>
<td>RS</td>
<td>WC</td>
<td>OFV</td>
<td>COM</td>
<td>C</td>
</tr>
<tr>
<td>hex code:</td>
<td>80</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>08</td>
<td>04</td>
<td>02</td>
<td>01</td>
</tr>
<tr>
<td>CC1 Condition Code One</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC0 Condition Code Zero</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDC Interdigit Carry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS Register Bank Select</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I played TV games...
The program given so far in tables A and B will produce a stationary object and background.
The next step is to set the object in motion, by modifying the program from address 0941 on:

(093E) CC1FC6 STRA, R9
(0941) 0728 LODI, R3
(0943) 0653 LODI, R2
(0945) 0C1EB8 LODA, R9
(0948) F420 TMI, R9
(094A) 9002 BCFR
(094C) DB18 BDDR, R3
(094E) 0C1EBD LODA, R9
(0951) F410 TMI, R9
(0953) 9002 BCFR
(0955) FA8F BIRR, R2
(0957) F440 TMI, R9
(0959) 9002 BCFR
(095B) DA99 BDDR, R9
(095D) 0C1EBE LODA, R9
(095E) F420 TMI, R9
(0962) 9C0991 BCFA
(0965) FB00 BIRR, R3
(0967) CF1F0A STRA, R3
(096A) CE1F0C STRA, R2
(096D) 03 LODZ, R3
(096E) 38E3 BSTR, UN, Ind. (0994)
(0970) C3 STRZ, R3
(0971) 02 LODZ, R2
(0972) 3F0998 BSTA, UN
(0975) C2 STRZ, R2
(0976) 0502 LODI, R1
(0978) 0C1FCB LODA, R9
(0979) F440 TMI, R9
(097F) F977 BDDR, R1
(0981) 1F0945 BCTA, UN
(0984) E404 COMI, R9
(0986) 90B2 BCFR
(0988) D966 BIRR, R0
(098A) 4E00 COMI, R0
(098C) 902 BCFR
(098E) FD00 BDDR, R0
(0991) 17 RETC, UN
(0991) 0C1EB8 LODA, R9
(0994) F420 TMI, R9
(0996) 9C0945 BCFA
(0999) 3F05C0 BSTA, UN
(099C) 1F0014 BCTA, UN

After loading this program, it should be possible to move the object to and fro horizontally by operating the 'S' and '7' keys; vertical control is provided by the '2' and 'A' keys.

---

Table C.

The program given so far in tables A and B will produce a stationary object and background.
The next step is to set the object in motion, by modifying the program from address 0941 on:

(093E) CC1FC6 STRA, R9
(0941) 0728 LODI, R3
(0943) 0653 LODI, R2
(0945) 0C1EB8 LODA, R9
(0948) F420 TMI, R9
(094A) 9002 BCFR
(094C) DB18 BDDR, R3
(094E) 0C1EBD LODA, R9
(0951) F410 TMI, R9
(0953) 9002 BCFR
(0955) FA8F BIRR, R2
(0957) F440 TMI, R9
(0959) 9002 BCFR
(095B) DA99 BDDR, R9
(095D) 0C1EBE LODA, R9
(095E) F420 TMI, R9
(0962) 9C0991 BCFA
(0965) FB00 BIRR, R3
(0967) CF1F0A STRA, R3
(096A) CE1F0C STRA, R2
(096D) 03 LODZ, R3
(096E) 38E3 BSTR, UN, Ind. (0994)
(0970) C3 STRZ, R3
(0971) 02 LODZ, R2
(0972) 3F0998 BSTA, UN
(0975) C2 STRZ, R2
(0976) 0502 LODI, R1
(0978) 0C1FCB LODA, R9
(0979) F440 TMI, R9
(097F) F977 BDDR, R1
(0981) 1F0945 BCTA, UN
(0984) E404 COMI, R9
(0986) 90B2 BCFR
(0988) D966 BIRR, R0
(098A) 4E00 COMI, R0
(098C) 902 BCFR
(098E) FD00 BDDR, R0
(0991) 17 RETC, UN
(0991) 0C1EB8 LODA, R9
(0994) F420 TMI, R9
(0996) 9C0945 BCFA
(0999) 3F05C0 BSTA, UN
(099C) 1F0014 BCTA, UN

After loading this program, it should be possible to move the object to and fro horizontally by operating the 'S' and '7' keys; vertical control is provided by the '2' and 'A' keys.

---

The data contained in those three registers is 'in cold storage'. (The way in which one or other of these banks can be selected will be discussed below: see 'Program Status Word'). Any instruction referring to R1, R2 or R3 is performed only on that register in the selected bank — it has no effect on the corresponding register in the other bank.

Program Status Word

The 'Program Status Word' refers to two special-purpose 8-bit registers: the 'Program Status Upper' (PSU) and 'Program Status Lower' (PSL). Each bit in these registers has a special meaning, as illustrated in figure 1. Briefly, the most important points as they relate to the complete TV games computer are as follows:

- **Sense**: this bit is '1' for the duration of the vertical reset pulse, at the end of each 'frame'. It can be used, for example, to synchronise the program to the actual display on the screen.
- **Flag**: can be set, reset and tested at will, as an indication of some condition relating to the program—for instance, to distinguish between the first and following runs through a particular section in the program.
- **Interrupt Inhibit**: The PVI generates 'interrupts' at the end of each frame and each time an object is completed. If this bit is set, these interrupt requests are ignored; otherwise, program execution 'jumps' from wherever it happens to be to address 0903 and runs the program section that it finds there as a subroutine. Note that this can cause chaos, if one isn't aware of the mechanism; for this reason, it is advisable to start every program with the instruction '7520' (i.e. set Interrupt Inhibit).
- **Stack pointers**: These three bits are set and reset by the processor, to keep track of the 'subroutine levels'. The stack is eight levels deep, which means that the main program may branch to a subroutine, that may branch to a further subroutine, and so on up to eight times before starting to 'climb back up' by means of Return instructions. It is possible to modify the stack pointers deliberately, as part of a program, but this is unwise for beginners...

- **Condition Code**: These two bits are set by (the results of) several different instructions, as shown in the Instruction Set given elsewhere. For instance, if the data loaded into a register is 00, the condition code will also be set to 00. Most of the branch and return instructions can be made 'conditional', by specifying a particular condition code setting: in that case, a 'Branch on Condition True' instruction, for instance, will only be executed if the actual condition code at that point corresponds to the one specified. If the two don't correspond, the instruction is ignored.
- **IDC, WC, OVF, COM, C**: These five bits will be dealt with later; see: Arithmetic and Compare.
- **Register bank Select**: This bit is used...
Table 2. Load and Store

<table>
<thead>
<tr>
<th>description</th>
<th>example</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load register zero</td>
<td>LODZ</td>
<td>from R2 to R0</td>
</tr>
<tr>
<td>Load immediate</td>
<td>LODI</td>
<td>'xx' = data</td>
</tr>
<tr>
<td>Load relative</td>
<td>LODR</td>
<td>'yy' = displacement</td>
</tr>
<tr>
<td>Load absolute</td>
<td>LODA</td>
<td>'zzzz' = address</td>
</tr>
<tr>
<td>Store register zero</td>
<td>STRZ</td>
<td>to R1 from R0</td>
</tr>
<tr>
<td>Store relative</td>
<td>STRR</td>
<td>'yy' = displacement</td>
</tr>
<tr>
<td>Store absolute</td>
<td>STRA</td>
<td>'zzzz' = address</td>
</tr>
</tbody>
</table>

of the other three registers, 'C1', for example, transfers data from R0 to R1. Note that the instructions '00' and 'C0', for 'LODZ, R0' and 'STRZ, R0', don't exist.

Load immediate transfers the data given in the instruction to the specified register. '07CA' (= LODI, R3) loads the data 'CA' into register 3.

Load relative and Store relative refer to the relative addressing mode described earlier. Relative indirect addressing can also be used, as described earlier.

Load absolute and Store absolute are used when absolute or absolute indexed addressing is required.

Instruction Set

Several instructions have already been mentioned briefly; having laid the groundwork, it is now possible to examine the instruction set in greater detail.

Load and store

The principle of these instructions is obvious: data is transferred into (Load) or from (Store) a specified register.

Load Register zero and Store Register zero transfer data between R0 and one of the other three registers, 'C1', for example, transfers data from R0 to R1. Note that the instructions '00' and 'C0', for 'LODZ, R0' and 'STRZ, R0', don't exist.

Load immediate transfers the data given in the instruction to the specified register. '07CA' (= LODI, R3) loads the data 'CA' into register 3.

Load relative and Store relative refer to the relative addressing mode described earlier. Relative indirect addressing can also be used, as described earlier.

Load absolute and Store absolute are used when absolute or absolute indexed addressing is required.

In all cases, the two Condition Code bits are set according to the sign of the data transferred: they become 01 if the data is a positive number, 00 if it is zero and 10 if it is negative (i.e. 80...FF, corresponding to -128...-1).

The Load and Store instructions can be summarised as shown in Table 2.

(Subroutine)Branch

Normally speaking, a program is executed step by step: in other words, the instructions are carried out in the order in which they are stored in the memory. If a jump to a different section of the program is required, a so-called Branch instruction must be used.

There are two basic types of Branch instruction: those for a (main program) Branch and those for a Branch to Subroutine. In the former case, the main program itself jumps to a different point in the memory: a Branch to Subroutine, on the other hand, can be considered as an interruption in the main program: the main program is stopped at the branch-to-subroutine instruction, the subroutine (elsewhere in memory) is carried out, after which the main program continues at the point where it was interrupted. Several variations of both types of Branch instruction are available:

Branch (to Subroutine) on Condition True, Relative or Absolute. For each of these four basic instructions, a particular setting of the Condition Code bits can be specified; the branch will only be executed if the actual condition code corresponds to the one specified. For example, the basic instruction for Branch on Condition True, Absolute (BCTA) is '1Czzz', where zzzz is the absolute address to which we want to jump. As it stands, this branch instruction will only be carried out if the condition code is 00. Similarly '1Dzzz' and '1Ezzz' specify the condition codes 01 and 10, respectively. Finally, '1Fzzz' would seem to refer to a condition code 11, but this code doesn't exist. In fact the corresponding instruction is used for an unconditional branch: a branch that is always carried out, no matter what the condition code.

Branch (to Subroutine) on Condition False, Relative or Absolute. These four instructions are similar to those described above; the only difference is that the branch is executed if the actual condition code does not correspond to the one specified. The 'BSFA' instruction BCCzzz, for example, will cause a branch to subroutine if the condition code is either 01 or 10, but not if it is 00. Note that no 'unconditional' variations of these instructions exist: the corresponding codes 9Byy, 9Fzzz, BByy and BFzzz are used for other instructions.
specification. In this case, however, 01 is first added to (incremented) or subtracted from (decremented) the contents of the register, after which the branch instruction is only carried out if the new contents are non-zero. Note no 'Branch-to-subroutine' version of these instructions exists.

Zero Branch (to Subroutine) Relative, Unconditional. These two instructions are relatively useless in the TV games computer, since they specify a branch relative to address 0000: the start of the program!

In practice, these instructions will not be used often, since in most cases Clear, Masked or Preset, Masked instructions are more suitable. 'Clear Program Status Upper, Masked 40' (7440) will clear the 'flag' bit, without having any effect on the other bits in the PSU. Similarly, 'PPSL, RS' (7710) will select the second register bank.

Finally, any bit (or combination of bits) in each of the program status registers can be tested: 'Test Program Status Upper, Masked 40' (B440) will cause the Condition Code to be set to 00 if the 'flag' is set; otherwise the Condition Code will become 10.

Branch (to Subroutine) Indexed, Absolute, Unconditional. These two instructions are the only two indexed branch instructions that exist. The value in the index register (which <must> be R3) is added to the basic absolute address given, and the branch is executed to the resultant address.

Return from subroutine, conditional. As before, a condition code is specified as part of this instruction; if the actual condition code matches, the subroutine is terminated. An unconditional end of the subroutine is indicated by the condition code 11, so the instruction RETC, UN is 17. A variation on this instruction exists (RETE) that not only ends the subroutine, but also resets the Interrupt Inhibit bit. Not a good idea, until one has gained enough experience to start using the interrupt facility...

The complete set of branch instructions is summarised in Table 3.

Table 3. Branch (to subroutine)

<table>
<thead>
<tr>
<th>description</th>
<th>example</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On Condition True, Relative</td>
<td>(BCTR)</td>
<td>1Byy = unconditional</td>
</tr>
<tr>
<td>On Condition True, Absolute</td>
<td>(BCTA)</td>
<td>1Czzz = unconditional</td>
</tr>
<tr>
<td>On Condition False, Relative</td>
<td>(BCFR)</td>
<td>9Byy = unconditional</td>
</tr>
<tr>
<td>On Condition False, Absolute</td>
<td>(BCFA)</td>
<td>9Czzz = unconditional</td>
</tr>
<tr>
<td>On Register Non-zero, Rel.</td>
<td>(BRNR)</td>
<td>5Byy = unconditional</td>
</tr>
<tr>
<td>On Register Non-zero, Abs.</td>
<td>(BRNA)</td>
<td>5Czzz = unconditional</td>
</tr>
<tr>
<td>On Incrementing Register, Rel.</td>
<td>(BRR)</td>
<td>9Byy = unconditional</td>
</tr>
<tr>
<td>On Incrementing Register, Abs.</td>
<td>(BRIA)</td>
<td>9Czzz = unconditional</td>
</tr>
<tr>
<td>On Decrementing Register, Rel.</td>
<td>(BDRR)</td>
<td>9Byy = unconditional</td>
</tr>
<tr>
<td>On Decrementing Register, Abs.</td>
<td>(BDRA)</td>
<td>9Czzz = unconditional</td>
</tr>
<tr>
<td>Zero Relative, Unconditional</td>
<td>(ZBRR)</td>
<td>R3 only!</td>
</tr>
<tr>
<td>Indexed Absolute, Unconditional</td>
<td>(BXA)</td>
<td>9Fzzz = unconditional</td>
</tr>
</tbody>
</table>

Branch to Subroutine:

<table>
<thead>
<tr>
<th>description</th>
<th>example</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Condition True, Relative</td>
<td>(BSTR)</td>
<td>3Byy = unconditional</td>
</tr>
<tr>
<td>On Condition True, Absolute</td>
<td>(BSTA)</td>
<td>3Czzz = unconditional</td>
</tr>
<tr>
<td>On Condition False, Relative</td>
<td>(BSFR)</td>
<td>9Byy = unconditional</td>
</tr>
<tr>
<td>On Condition False, Absolute</td>
<td>(BSFA)</td>
<td>9Czzz = unconditional</td>
</tr>
<tr>
<td>On Register Non-zero, Rel.</td>
<td>(BSNR)</td>
<td>7Byy = unconditional</td>
</tr>
<tr>
<td>On Register Non-zero, Abs.</td>
<td>(BSNA)</td>
<td>7Czzz = unconditional</td>
</tr>
<tr>
<td>Zero Relative, Unconditional</td>
<td>(ZBSR)</td>
<td>R3 only!</td>
</tr>
<tr>
<td>Indexed Absolute Unconditional</td>
<td>(BSX)</td>
<td>8Fzzz = unconditional</td>
</tr>
</tbody>
</table>

Return from subroutine:

<table>
<thead>
<tr>
<th>description</th>
<th>example</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditional</td>
<td>(RETC)</td>
<td>14</td>
</tr>
<tr>
<td>And Enable Interrupt, Conditional</td>
<td>(RETE)</td>
<td>34</td>
</tr>
</tbody>
</table>
Table D.

With the complete program given so far (in tables A ... C), it is possible to get the object into your sights. Now, what about shooting it down??

First, modify the instruction in address 0962: instead of '9C9998', enter '9C999B'.

The existing program, from address 0999, is then extended as follows:

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>09BE</td>
<td>F800</td>
<td></td>
</tr>
<tr>
<td>0998</td>
<td>C0 C0</td>
<td></td>
</tr>
<tr>
<td>0990</td>
<td>C0 C0</td>
<td></td>
</tr>
<tr>
<td>0994</td>
<td>C0 C0</td>
<td></td>
</tr>
<tr>
<td>0999</td>
<td>C0 C0</td>
<td></td>
</tr>
<tr>
<td>099A</td>
<td>17</td>
<td>RETC, UN</td>
</tr>
<tr>
<td>0999</td>
<td>F480</td>
<td>TMI, R0</td>
</tr>
<tr>
<td>099D</td>
<td>9830</td>
<td>BCFR</td>
</tr>
<tr>
<td>099F</td>
<td>7702</td>
<td>PPSL, COM</td>
</tr>
<tr>
<td>09A1</td>
<td>E756</td>
<td>COMI, R3</td>
</tr>
<tr>
<td>09A3</td>
<td>392A</td>
<td>BCFR</td>
</tr>
<tr>
<td>09A5</td>
<td>E75B</td>
<td>COMI, R3</td>
</tr>
<tr>
<td>09A7</td>
<td>9A26</td>
<td>BCFR</td>
</tr>
<tr>
<td>09A9</td>
<td>E681</td>
<td>COMI, R2</td>
</tr>
<tr>
<td>09A8</td>
<td>9C22</td>
<td>BCFR</td>
</tr>
<tr>
<td>09AF</td>
<td>E866</td>
<td>COMI, R2</td>
</tr>
<tr>
<td>09A6</td>
<td>9A1E</td>
<td>BCFR</td>
</tr>
<tr>
<td>09B1</td>
<td>050A</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>09B3</td>
<td>049AA</td>
<td>LODA, I-R1</td>
</tr>
<tr>
<td>09B6</td>
<td>CD7F00</td>
<td>STRA, I/R1</td>
</tr>
<tr>
<td>09B9</td>
<td>5978</td>
<td>BRNA, R1</td>
</tr>
<tr>
<td>09B8</td>
<td>0564</td>
<td>LODI, R1</td>
</tr>
<tr>
<td>09BD</td>
<td>0C1FCB</td>
<td>LODA, R0</td>
</tr>
<tr>
<td>09C0</td>
<td>F440</td>
<td>TMI, R0</td>
</tr>
<tr>
<td>09C2</td>
<td>9879</td>
<td>BCFR</td>
</tr>
<tr>
<td>09C4</td>
<td>F977</td>
<td>BDRR, R1</td>
</tr>
<tr>
<td>09C6</td>
<td>C0 C0</td>
<td>2 x NOP</td>
</tr>
<tr>
<td>09C8</td>
<td>C0 C0</td>
<td>2 x NOP</td>
</tr>
<tr>
<td>09CA</td>
<td>C0 C0</td>
<td>2 x NOP</td>
</tr>
<tr>
<td>09CC</td>
<td>1F0900</td>
<td>BCTA, UN</td>
</tr>
<tr>
<td>09CF</td>
<td>0C1EBB</td>
<td>LODA, R0</td>
</tr>
<tr>
<td>09D2</td>
<td>F420</td>
<td>TMI, R0</td>
</tr>
<tr>
<td>09D4</td>
<td>9C9845</td>
<td>BCFR</td>
</tr>
<tr>
<td>09D7</td>
<td>3F65CD</td>
<td>BCTA, UN</td>
</tr>
<tr>
<td>09DA</td>
<td>1F08214</td>
<td>BCTA, UN</td>
</tr>
</tbody>
</table>

This leaves room for a section of program to be added later.

The compare instruction is basically similar, but it is both more precise and more versatile - and also more complicated to use. In this case, a data value is specified instead of a mask, and the condition code can be set in three ways: 01 for 'greater than', 00 for 'equals' and 10 for 'less than'. There are two main points to watch, when using this instruction: what is meant by 'greater than' (data in register greater than data specified, or vice versa; see the footnotes in the Instruction Set) and what type of comparison is required. With the 'COM' bit in the PSL set to 0, an 'arithmetic' comparison will be performed: all values from 00 to FF are treated as negative numbers (two's complement). If the COM bit is set to 1 (by means of the instruction 7702 = PPSL, COM) a 'logical' comparison will result: the data is treated as a positive 8-bit binary number.

Once the object is accurately (1) centered, it can now be 'shot to pieces' by operating the 'F' key.

No Operation

A surprisingly useful instruction, this! When the processor finds the code 'C0', it simply carries on to the next instruction. There are two cases where this can be particularly useful: to 'delete' instructions that prove unnecessary, or to 'leave a gap' into which further instructions are to be added at a later date.

Halt

This stops the processor, quite drastically. The only way to start it up again is either to operate the 'reset' key or provide an interrupt - provided the interrupt inhibit bit is not set. In general, this is not a good idea; in the TV games computer, a 'return to Monitor' instruction (1F0000 = BCTA, UN, for instance) will usually be more suitable.

Test under Mask; Compare

With all the conditional branching facilities available, it is obviously useful to have instructions that set the Condition Code. Basically, all types of data transfer to or data manipulation in a register do this; furthermore, the Test Under Mask Immediate (TMI) and Compare (COM) instructions set the condition code bits without altering the data in any way.

The TMI instruction is the easiest to use: a register is specified in the first part of the instruction ('F4' for register zero, 'F5' for R1, and so on) and a 'mask' in the second part. The mask simply specifies the bits to be tested: '81', for instance, is 1000 0001 in binary and so the first and last bits will be tested. If, in the data contained in the specified register, these two bits are '1s', the condition code will be set to 00; if not, CC will become 10. An example. If the data in R1 is 05, the instruction F501 (TMI, R1, 01) will set the condition code to 00 — the data, 05, is 0000 0101. By contrast, F581 will set the CC to 10: 0000 0101.

repeat from 0000

repeat if not

save status and

return to monitor

delay

leave some

more room

repeat from 0000

'TM' key?
A few tips

The instructions explained so far are sufficient for simple programs. The remaining facilities will be dealt with next month. Meanwhile, however, a few practical tips on how to program should prove useful.

First and foremost: remember to block the Interrupt facility if this is not required in the program! For the time being, it is advisable to start every program with the instruction ‘7620’ (PPSU, II).

There are several ways to end a program. Usually, one of the keys (‘PC’, for instance) is used to initiate a jump back to Monitor. A few variations are given at the end of Tables A . . . D. The jump to Monitor itself can be done in several ways. The shortest is to use a ZBRR instruction: ‘9800’ should do the trick, but we’ve never actually tried it.

A similar solution is ‘1F0000’, as mentioned earlier; this we have tried, and it usually works. Sometimes, however, for no apparent reason problems occur: in particular, a row of black squares or lines down the left-hand edge of the screen when the program is restarted. Without knowing why this happens, yet (maybe we will know more next month!), we can offer three solutions:

- return to monitor by means of the two instructions

0400 LODI, R0
1F0011 BCTA, UN

Note that, in this case, the original value in R0 is lost; for that matter, returning via address 0000 always causes the data in R0 to become 00, as several readers have noticed!

- similarly, but untried:
20 EORZ, R0
9B19 ZBRR

This has the advantage that if fits in the same memory space as ‘1F0000’, if the latter causes trouble.

- finally, if the value in R0 is to be stored:
3F00CD BSTA, UN
1F0014 BCTA, UN

Don’t ask us to explain this one — that would require an extensive discussion of the monitor software!

When it comes to the program itself, the first thing is to work out what you want to do. Obviously. For simple programs, this can usually be put into words quite easily. The program given in Table 1, for example, was originally specified as ‘clear objects; define background colour; load FF in all background bytes’. For more complicated programs, some more extensive advance plotting may be required — using a flow chart, say — but usually a complicated program can be ‘broken up’ into several simple routines. These can each be tried and tested individually, before ‘tacking them together’ to obtain the complete final program.

As each semi-complete routine is entered, it is highly advisable to store it on tape before the first test run. This lesson was learned the hard way, when one misplaced relative address caused ‘garbage’ to be stored at all sorts of awkward places throughout the program. The only solution was to laboriously re-enter the whole program . . .

When it comes to ‘de-buggin’ a program — it always does, no program works perfectly first time! — the ‘Breakpoint’ routine can be very useful. There are two points to watch, however.

In the first place, as mentioned in the original article, the Breakpoint address given must be the first address of an instruction. For instance, in the following section of program:

Table 5.

| 060A | LODI, R2 |
|      | LODA, R3 |
|      | TMI, R3 |
|      | BCFR, # |
|      | BDWR, R2 |

wait for VRLE

total delay: approximately 0.2 seconds

Table 6.

| 0900 | 7620 | PPSU, II |
| 0902 | 05AD | LODI, R1 |
| 0904 | 0400 | LODI, R0 |
| 0906 | CD5F00 | STRA, I-R1 |
| 0908 | 5978 | BRNR, R1 |
| 0909 | 0469 | LODI, R0 |
| 090D | CC1FC6 | STRA, R9 |
| 0910 | 0562 | LODI, R1 |
| 0912 | 04FF | LODI, R0 |
| 0914 | CD5F00 | STRA, I-R1 |
| 0917 | 060A | LODI, R2 |
| 0919 | 0F1FCB | LODI, R3 |
| 091C | F440 | TMI, R3 |
| 091E | 0879 | BCFR, # |
| 0920 | FA77 | BDWR, R2 |
| 0922 | 5970 | BRNR, R1 |
| 0924 | 40 | HALT |

clear objects and background

colour

load background delay

breakpoints can be specified at addresses 0000, 0002 and 0004, but not at 0001, 0003 or 0005! The second point to watch is that breakpoints modify the program at that point. If the breakpoint is found in the normal way, the original data will be restored automatically. However, if things really go wrong so that the reset key must be used to return to Monitor, it may be necessary to restore the data by hand!

The PVI and keyboard

The main points regarding the PVI were:

0000 7620 PPSU, II
0002 0400 LODI, R0
0004 0605 LODI, R2

0000 0002 LODI, R0
0004 0005 LODI, R2

...
Finally, what about adding a time limit? As follows:

- first, fill in the space in the program starting at address 0990:

<table>
<thead>
<tr>
<th>(0990)</th>
<th>F000</th>
<th>BDRR, R0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0991)</td>
<td>7710</td>
<td>PSRL, RS</td>
</tr>
<tr>
<td>(0992)</td>
<td>7710</td>
<td>PSRL, RS</td>
</tr>
<tr>
<td>(0993)</td>
<td>9802</td>
<td>BCFR</td>
</tr>
<tr>
<td>(0994)</td>
<td>0700</td>
<td>LODI, R3</td>
</tr>
<tr>
<td>(0995)</td>
<td>7510</td>
<td>CPSL, RS</td>
</tr>
</tbody>
</table>

- similarly, fill in the space starting at 0960:

<table>
<thead>
<tr>
<th>(0960)</th>
<th>F077</th>
<th>BDRR, R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0961)</td>
<td>7710</td>
<td>PSRL, RS</td>
</tr>
<tr>
<td>(0962)</td>
<td>0700</td>
<td>LODI, R3</td>
</tr>
<tr>
<td>(0963)</td>
<td>7510</td>
<td>CPSL, RS</td>
</tr>
</tbody>
</table>

- modify the data at address 0991; instead of '1F0945', the instruction becomes '1F09DD';
- at address 09D4, the instruction is modified to '9C0976' (instead of 9C0945).
- the program is extended, from address 09DD onward, as follows:

<table>
<thead>
<tr>
<th>(09D0)</th>
<th>1F0914</th>
<th>BCTA, UN</th>
</tr>
</thead>
<tbody>
<tr>
<td>(09D1)</td>
<td>7710</td>
<td>PSRL, RS</td>
</tr>
<tr>
<td>(09D2)</td>
<td>7604</td>
<td>BCFR</td>
</tr>
<tr>
<td>(09D3)</td>
<td>0619</td>
<td>LODI, R2</td>
</tr>
<tr>
<td>(09D4)</td>
<td>1808</td>
<td>BCTR, UN</td>
</tr>
<tr>
<td>(09D5)</td>
<td>CF1FC0</td>
<td>STRA, R3</td>
</tr>
<tr>
<td>(09D6)</td>
<td>F068</td>
<td>BDRR, R2</td>
</tr>
<tr>
<td>(09D7)</td>
<td>F802</td>
<td>BDRR, R3</td>
</tr>
<tr>
<td>(09D8)</td>
<td>1805</td>
<td>BCTR, UN</td>
</tr>
<tr>
<td>(09D9)</td>
<td>7510</td>
<td>CPSL, RS</td>
</tr>
<tr>
<td>(09DA)</td>
<td>1F0945</td>
<td>BSTA, UN</td>
</tr>
<tr>
<td>(09DB)</td>
<td>04FF</td>
<td>LODI, R0</td>
</tr>
<tr>
<td>(09DC)</td>
<td>CF1FC6</td>
<td>STRA, R0</td>
</tr>
<tr>
<td>(09DD)</td>
<td>7510</td>
<td>CPSL, RS</td>
</tr>
</tbody>
</table>

- if 'clock' stopped (R3 = 0), preset R2 for one-second count
- update score and increment R2
- duplicate R2 and decrement R3
- branch if R3 = 0
- repeat key check routine
- store V0 in score, make screen white ('you lose') and repeat via delay routine

all discussed in the original articles and the data supplied with the p.c. board.
One point, however, did not receive all the attention it deserves — at the time, we didn't realise how useful it was!
The 'VRL' bit, at address 1FCB, goes high at the end of each frame; it is reset at the end of the VRST pulse or when read. This means that it can only be read as '1' once for each frame. As an example, a simple 'delay' routine is given in Table 5. Basically, what happens is that the processor waits until it finds VRLE = 1; it then decrements the value in R2 and repeats the VRLE scan if R2 is not yet zero. The result is a delay, approximately equal to the value in R2 times the frame period (20 ms). By way of demonstration, this routine can be included inside the 'load background' loop in Table 1. The result is given in Table 6.

Finally, the keyboard. Each column corresponds to one address: 1EB8 for the column above the 'F' key, 1E89 and 1E8A for the next two columns, 1EB8 for the column that includes the 'reset' key (note that this key itself is not scanned in the keyboard layout suggested) and 1EBC... 1EBE for the last three columns. When reading the keyboard in this way, the four left-hand bits retrieved as data correspond to the four keys in the column and the other four bits are all ones. 'F' at address 1EB8, for instance, means that the

RCAS, WCAS and ESS

The Cassette routines were discussed in the earlier articles. Apparently, some readers have had problems loading the first ESS record, so a few words of advice may be appreciated... Assuming that a cassette recorder is used, the first thing to check is that programs can be stored on tape from the computer and retrieved without any problems. This can be done without ever loading a program: there is always some kind of data in the memory! The test sequence is as follows:

- operate the 'reset' key;
- operate the 'start' key ('1111' should appear);
- press the 'WCAS' key ('bEG =');
- enter 0900, followed by '4' ('bEG = 0900, End =');
- enter 0FF, followed by '4' ('End = 0FF, Sad =');
- enter 0900, followed by '4' ('

'Sad = 0900, FIL =' );
- enter 1, but not [4] ('FIL =');
- start the tape in the Record mode, and set the level to about half way;
- operate the '4' key.

Happily, the recording level meter should indicate approximately nominal full modulation during the first second or so after the '4' key is operated; it will then drop back slightly (to a few dB below full modulation). If this is not the case, the level setting can be corrected, after which the whole sequence described above will have to be repeated. Having found the correct level, it is wise to make a note of it, for future reference.
Having made a complete recording at correct level, the various addresses entered above will reappear on the screen. The test can now be concluded:
- operate the 'RCAS' key ('FIL = 1');
- enter the file number, '1' ('FIL = 1');
- press the '-' key (not '+?!').
The text 'FIL = 1' will jump to the top of the screen. The tape can now be played back, and the data recorded on it will be compared with the original data in the memory. During this time (approximately 36 seconds) two dots will flash below the '-' sign on the screen. At the end of this time, all the original data will reappear on the screen with the added line 'PC = 0900'. If this happens, all is well and the cassette interface is working.

In the unloved-for event that the check routine breaks off before the end of the recording, with the message 'Ad = 09BA', for instance, then something is wrong...

In our experience, moving the recorder further away from the TV set invariably cures the problem.

Next step. The ESS record. You would expect that recording it on tape and then playing it into the computer should work. In practice it does, most of the time, but sometimes the computer rejects the program for no apparent reason. (Message: 'Ad = ...'). Since the programs are on the record (with the exception of the missing 'Interrupt Inhibit' instruction in file 6, as mentioned earlier) it must be possible to load them. In one particularly stubborn case, the following solution was found. The output from the preamplifier, after

the tone and volume controls, is fed to the TV games computer. A 'high' file number is entered (8 or 9) in the RCAS mode, and the record is started. After some manipulations with the volume control, two dots will start to flash rapidly under the '+' sign, and the actual file number should also appear on the screen. The trick is now to manipulate the volume control (and, if necessary, the treble control) until the dots flash regularly and the second file number remains constant for the duration of each program on the record. Once this is achieved, the volume and treble controls are left strictly alone and each program in turn is loaded into the computer (this should work, now) and from there to the tape. From now on, the programs can be retrieved reliably from this tape. Rest assured, we are doing our utmost to make the second ESS record for the TV games computer easier to load...

ESS 006

...the second record with software for the TV games computer, which is what this article was to have been about. However, it's long enough as it is.

Some idea of the programs can be obtained from the photo's distributed liberally throughout these pages. One program converts the computer into a fairly comprehensive colour TV test pattern generator; the other can be considered as a programming aid. It contains routines for composing object shapes and background on the screen - so that you can see what you're doing --, the 'relative address calculation' mentioned earlier, and a routine for scanning a character set available in the monitor program as will be explained next month.

Full details of how to use these programs will be included with the record, which will be made available next month.

In conclusion

With the information given in this article, it is possible to write simple programs. Some examples are included on these pages. Now is the time to start practicing - next month we'll discuss the rest of the instruction set, and give some rather more complicated routines... After that, you will know as much as we do!
There are a variety of different ways in which the control voltages can be programmed and stored, e.g., via potentiometers, switches, sample-and-hold circuits, or digital memories. The method adopted here is to encode the voltage digitally and store it in a RAM. When the contents of the memory are read out, they are fed to a D/A converter, which provides an analogue signal suitable for feeding to the synthesizer VCOs. In addition to the pitch of the notes (i.e., their frequency), their relative length can also be programmed. The duration of each note can be selected in the ratio of 1:2:4:8. The block diagram of the programmable sequencer is shown in figure 1. The pitch (i.e., its position on the musical scale and its octave) and length of the note are set up in binary code on switches which are connected to the data inputs of the RAM (see figure 3). The address into which the data is stored is determined by an address counter. In actual fact, two address counters are employed, one of which (the 'subsidiary' counter) is clocked by the other ('main' counter). When the stored melody is to be played back, the address counter steps through each of the memory locations in turn. The data is read out and fed to the D/A converter, which provides the actual control voltages for the VCOs. During normal operation the circuit can store 16 sequences of 16 notes each, i.e., a combined sequence of 256 notes; with the aid of the reset circuit and the 'subsidiary' address counter, however, even longer (or indeed shorter) sequences are also possible. The note length is controlled by a D/A converter and VCO, the output of which varies the clock frequency of the main address counter. The analogue voltages from output A are fed to the synthesizer VCOs; at output B a gate pulse is generated to accompany each note. The gate pulse, whose width can be varied, is used to determine the start and duration of the envelope control voltage generated by the ADSR module of the synthesizer. The complete circuit diagram of the programmable sequencer is shown in figures 2a and 2b. Figure 2a contains the digital section of the sequencer, comprising the memory, address counter, and reset circuit, whilst figure 2b shows the D/A converters and output stages. Two 2101's, 256 x 4-bit RAMs, connected in parallel from the memory in which the digitally encoded control voltages are stored. The higher order addresses of the input data are set up on switches S2...S5. The flip-flop (IC11) interposed between the switches and the RAMs ensure that the new address set up on S2...S5 is only presented to the address inputs of the RAMs after the previous note sequence has ended.

The main address counter is formed by IC10. The counter is clocked, via IC6, by the analogue section of the circuit shown in figure 2b. This counter generates the 'low order' addresses.

C. Voss
i.e. it clocks from '0000' to '1111', whereupon the high order address is incremented by one (via S2...S5), before the counter resets and starts to cycle through another sequence of 16 addresses.

The reset circuit is formed by N12 and N13. When the data outputs a...f of the RAM all go high, N12 and N13 ensure that the binary counter is reset to zero. Thus the address containing the data word '111111' represents the reset address. Inverters I1...I4 form a NOR gate (the inverters all have open-collector outputs), so that only when the address counter resets (i.e. its outputs all go low), is IC11 clocked. This ensures that a new (high order) address cannot be presented to the address inputs of the RAMs before the previous note sequence has ended.

When S2...S5 are set to position c, the 'subsidiary' address counter (IC12) is connected to the address inputs of the RAMs. This counter is clocked by IC10, via I1...I5, so that it receives a clock pulse every time IC10 resets (i.e. every 16 addresses). Thus if all the outputs of IC12 are connected to the RAMs, the entire contents of the memory can be read out in sequence. Switch S1 determines the operating mode of the sequencer. In position a the switch blocks gate N10, with the result that the address counter is immediately inhibited. In position b the sequencer operates normally, whilst in position c the counter will stop once it reaches '0000'.

To actually program a sequence of notes into memory, pushbutton switch S9 is first pressed, resetting the address counter via N12 and enabling the RAMs. The information relating to the pitch and length of the note to be stored is then written into the RAMs by pressing S10. Each of the monostable multivibrators MMV1...MMV3 are now triggered in turn. The output pulse from MMV1 clocks the address counter (IC10) via N9. The pulse from MMV2 temporarily puts the RAMs into the write mode, so that the information present on the data inputs is in fact stored in memory. The output of MMV3 takes the output of N8 high, so that N13 is capable of recognising the reset code ('111111') on the data outputs of the RAMs.

The next note is written into memory in the same way; the input data is set up on the corresponding switches whereupon S10 is pressed and the data is written into memory. Once the desired sequence of notes is stored, pressing S11 writes the reset code into the memory by taking the inputs of N1...N6 low and hence the data inputs of the RAMs high. When N13 recognises the reset code, the address counter (IC10) is reset, so that IC11...I5, flip-flop FF2 is triggered and the RAMs are returned to the read mode. Schmitt trigger N21 ensures that FF2 assumes a definite state upon switch-on and that the RAMs are inhibited for a brief initial period. The digital-analogue converters and output stages of the circuit are shown in figure 2b. IC1...IC3 produce the analogue control voltages which determine the frequency of the notes, whilst the D/A converter round IC4 is used to control the length of the notes. Unlike the other two D/A converters (IC1/IC2), the output voltage increases in an exponential, not linear fashion.
That is to say, when the digital input signal increments by '01', the output voltage doubles. The output of IC4 is fed to a sawtooth generator formed by IC4 and IC5; this both clocks the main address counter (IC10 in fig. 2a) and, via the Schmitt trigger IC7, provides a variable-width gate pulse. In spite of the six adjustment points (potentiometers P1...P6), the circuit can be set up fairly simply, without the need for any special measuring equipment. The circuit is adjusted correctly when a change in the 'e' input from '0' to '1' causes the voltage at output A to increase by 1 V. This voltage can be adjusted by means of P3. P2 should be set such that the output voltage changes by 0.5 V when inputs 'b' and 'c' go high. Fine tuning is performed by means of P1. A change in state of input 'a' should correspond to a change of 1/12 V in the output signal. P5 should be adjusted such that the output frequency doubles when input 'g' goes high. P4 is used to compensate the offset voltages of IC4 and IC5. Finally, P6 determines the width of the gate pulse.
are connected in the same way as D11 and D12, further down the chain.

July/August 1979

**Circuit 25: linear thermometer**

In the circuit diagram, the indications 'IC1' and 'IC2' are transposed.

**Circuit 27: moisture sensor**

The sensor should be connected in parallel with R1, not in series with R2.

**Circuit 30: automatic heated rear windscreen**

To avoid confusion: pin 3 of IC6 is the clock input; it is indeed connected to supply common, as shown.

**Circuit E: servo amplifier**

The printed circuit board is reproduced at twice the actual size; the actual size is shown below.

---

**Circuit 80: programmable digital function generator**

IC1 and IC2 should be type 2101, not 2102.

**Circuit 90: µ-programmable speed controller for model railways**

In figure 1, the output pinning of IC2 is incorrect; from top to bottom, the pin numbers should read: 7 (=Q1), 6, 5, 4, 9, 10, 11 (=Q1). In figure 3, the supply voltage should be 5 V.

**Circuit 100: 256-note sequencer**

In figure 1a, pins 4 of IC1 ... IC3 should, of course, be connected to −15 V. In figure 1c, a 1 nF capacitor is shown between pin 3 of N38 and supply common. This capacitor should be connected between pin 4 of N24 and supply common.

**Circuit 106: chorosynth**

In figure 2, the connections to the left-hand keys should all be moved one position to the right. The first key is thus connected to the R1/R2 junction. Figure 3, R35 (between T10 and A3) is shown connected to +15 V; it should, however, be connected between the base of T10 and the inverting input of A3.

**Circuit 102: video pattern generator**

Pin 12 of N16 is shown connected via a diode to pin 7 of IC3b; it should, however, connect to pin 6. Furthermore, R26 is shown connected to the right-hand side of the Xtal, whereas it should be connected to the top of C1.

The author has suggested two improvements to this circuit: the resistor values for R38 ... R45 can be decreased by a factor of 10 (e.g. 47 kΩ instead of 470 kΩ), and a diode may be added between the output of N4 (cathode) and pin 13 of N12 (anode).
Amount may be purchased. Further details from Marshall's branches after October 12th 1979. Another significant point is that Marshall's have reduced prices on their top line products, resulting in very competitive prices. Two reply paid order forms are supplied in each catalogue to facilitate easy ordering and faster turn round on supply. The catalogue costs 50 p from any Marshall's branch or 65 p post paid from their head office. Marshall's (Head Office), Kingsgate House, Kingsgate Place, London NW6 4TA, Tel.: 01-624-8065.

Single-knob measuring bridge from Siemens

Siemens Ltd is marketing a single-knob resistance bridge that can be operated by one hand - a feature that greatly facilitates simultaneous note-taking. The bridge is available in two types: model M273-A1 comprising a Kelvin double bridge for measuring low resistance values (200 mohms to 2200 milli-ohms) and model M273-A2 with a Wheatstone bridge for medium resistance values. The low resistance model has built-in measuring lead compensation.

The bridge is balanced by an analog galvanometer read against a deviation scale. Both types are accurate to +/- 1% of the measured value (+/- 1.5% of the lowest range of model M273-A1) and can withstand a 2 kV voltage surge. The power source is two standard IEC R14 1.5 V cells. Alternatively, 2 or 6 V d.c. external power sources can be used which, generally, increase the reading accuracy. Both models are in a rugged moulded plastic housing. A leather case is available (extra) for heavy-duty use. Typical dimensions are 112 mm (4.41 in) wide x 84 mm (3.31 in) deep x 192 mm (7.5 in) long. The weight is 1.1 kg (2.41 lbs).

Siemens Limited, Siemens House, Windmill Road, SUNBURY-on-THAMES, Middlesex, TW16 7HS. Tel.: (09327) 85691.

High quality transient signal processors

Bryans Southern Instruments Limited of Mitcham, Surrey, have just announced the introduction of a new range of high-sampling frequency, high-resolution transient signal processors. Priced to suit modest laboratory budgets, these new units are known as the Series 523A. They offer a 10-bit resolution (i.e. 1 part in 1024), at the high sampling rate of 10 MHz, so that the shortest sampling interval is 100 nanoseconds.

When switched to dual timebase mode, the first part of the recording is recorded at timebase rate A, and the second part at rate B. The point at which the change is made, is set-up by the user on a preset trigger switch. It is calibrated in multiples of the sweep time, with a resolution of 0.05 from 0 to 0.95 of the sweep time. These dual timebase facilities can be used even when pretriggering is in operation - a unique feature for this type of instrument. Therefore, one can reference a short event of just a few microseconds during a trigger condition, many seconds in advance of a fast signal and still record the pre-trigger information. At the same time, the 523A maintains an exceptional high resolution of the time and amplitude of the signal. These transient signal processors are available in single- or dual-channel versions, each with 4096-word memories. They can be interfaced to many peripherals and computers by RS232C, IEEE488/1975 (G.P.I.B.), or the general purpose interface board. If required, up to seven instruments can be cabled together for digital output purposes.

Bryans Southern Instruments Ltd., Willow Lane, Mitcham, Surrey, CR4 4UL. Tel.: 01-640-3490.

New product range catalogue from Marshall’s

Marshall's announce the publishing of their new 1979/1980 product range catalogue on October 12th 1979. This edition contains many new products within its 60 pages, including an increased range of IC's, micro-power LCD clock modules, data and educational books etc etc. At the same time Marshall's are launching their new 'credit' card scheme, in conjunction with RETRA. This will enable customers to purchase goods on credit from any of the four Marshall's retail branches. Minimum monthly repayment is £5.00 and goods 20 times this

Four-colour plotter

An easy-to-use 11 by 17 in. (A3 size) microprocessor-controlled plotter that produces low-cost, high-quality multicolour graphic plots with data sent from virtually any computer or controller has been introduced by Hewlett-Packard. Interface is via RS232C/V24 asynchronous serial ASCII at any of eight switch selectable baud rates from 75 to 2400. Special design features also enable the plotter to be coupled into an existing computer terminal RS232C/V24 interface.

The new 7220A four-colour plotter generates character sets, dashed lines, and implements scaling and other high-level functions internally. Plotting colours are selected and changed under program control to produce high resolution graphic plots and overhead transparencies. There are seven colours available for clear film.

Character plotting speed of over two characters per second allows fully annotated graphs to be produced in minutes. A buffer with over 1100 bytes has been incorporated to store incoming graphic plot data so that 1/0 interrupts and data communications between computer and plotter are minimised. As an option, an additional 2048 bytes of buffer storage can be made available.

The plotter's built-in language contains two categories of instructions: device control and graphic instructions. The graphic instructions comprise more than 45 two-letter mnemonic instructions from the Hewlett-Packard Graphics Language (HP-GL) which equips the plotter with such capabilities as relative and absolute plotting, point digitising, labeling, character sizing, integer scaling and window plotting. No specialised programming experience is needed to use the new plotter.

For applications requiring unattended operation, the version of the new plotter features automatic page advance, an internal paper supply and paper cutter, and a detached paper tray to collect full- or half-page plots.

Hewlett-Packard Ltd., King Street Lane, Winnersh, Wokingham, Berkshire RG11 5AR

market WS1.5K1

The unit is quartz controlled (with access to the trimmer for fine adjustment), and includes an incandescent backlight feature. An alarm function is available to drive a beeper or some other external means of indication. With a running consumption of only 6 µA, the PC1M161A is suited to a variety of applications ranging from all types of consumer electronic equipment, to instrumentation, telephones, communications equipment etc. The unit requires only three momentary contact switches for setting etc. and its accuracy is within ± 5 minutes per year.

Ambit International, 2 Graham Road, Brentwood, Essex, Telephone: (0277) 227050.

market WS1.5K1

market WS1.5K1

market WS1.5K1

market WS1.5K1

market WS1.5K1
New, miniature, low-cost temperature recording spots

The temperature responsive tri- 
angle turns irreversible black from 
original white after having been 
exposed to its rated temperature for 
fractions of a second. Such 
single temperature spots, or multi-
temperature sequenced strips 
can record the maximum tem-
perature level of any surface to 
which they have been affixed over 
the entire application history with 
an accuracy of ±1% the price per 
spot can be as low as 2 pence.

Cobonics Ltd., 
Knaptown Mews, 
Seely Road, 
London SW17 9RL, 
Tel.: 01-672 4160.

(1231 M)

Miniature float switch for liquid-level sensing

A miniature float switch for sensing the level of noncorrosive liquids in vending, automotive and general industrial applications has been introduced by Hamlin Electronics. Designated the P219, the fully encapsulated switch measures only 15/16 inch (24 mm) 
diameter x 1 1/8 inche (45 mm) 
long (including mounting thread), 
and can switch a current of 0.5 A 
with a life of over 50 million 
operations. 
Maximum contact rating for the 
Hamlin P219 is 10 W, at it can 
switch voltages of up to 500 V. 
Hamlin Electronics Europe Ltd., 
Diss, 
Norfolk IP22 3AY, 
Tel.: (0393) 441112/3.

(1295 M)

Switching regulator power supplies

Now available from Amplicon 
Electronics Limited are four ad-
ditional models to their existing 
range of switching power supplies. 
These new models, designated 
RT153, RT154, RT303 and 
RT304, are designed to meet 
increasing micro processor appli-
cations. 
They employ isolated auxiliary 
outputs of 5 V and 12 V or 5 V 
and 15 V in addition to the main 
5 V output.

RT153 — 5 V @ 30 amps
12 V @ 5 amps
5 V @ 2 amps
RT154 — 5 V @ 30 amps
16 V @ 4 amps
6 V @ 2 amps
RT303 — 5 V @ 60 amps
12 V @ 5 amps
5 V @ 5 amps
RT304 — 5 V @ 60 amps
15 V @ 4 amps
5 V @ 5 amps

The RT153 and 154 models are 
packed in the standard 5 x 5 x 
(15) case size with the RT303 
and RT304 are in the standard 
5 x 8 x 10 (T30) case size, with 
combined total power ratings of 
150 watts and 300 watts respect-
ively.

Amplicon Electronics Ltd., 
16 Lion Mews, 
Hove BN3 5RA, 
Tel.: Brighton (0273) 720716.

(1292 M)

High power servo amplifier and motor 

The new SH3015 high power 
amplifier from Fairchild has been 
developed for applications re-
quiring high current and high 
output capability. It is able to 
supply up to 6 A continuously 
into a load between ±35 V. 
Notable features include internal 
compensation, programmable cur-
rent limiting and excellent stab-
ility when driving into resistive 
and inductive loads. 
The amplifier front-end incor-
porates a µA 741 operational 
amplifier with additional voltage 
and current gain stages so enabling 
it to meet the performance 
required for servo systems. The 
output is protected from voltage 
transients caused by inductive 
surges. Output current limiting is 
delivered by placing appropriate 
resistors between the supply pins 
and the respective current limit 
points. The case is electrically 
isolated.

Absolute maximum ratings in-
clude an internal DC power dissi-
pation of 70 W with a case tem-
perature of 25°C, input voltage 
differential 30 V and DC output 
current of 10 A.

Fairchild, 
Camera & Instrument (UK) Ltd., 
239 High Street, Potters Bar, 
Herts EN6 5BU, 
Telephone: (0707) 51111

(1231 M)

SEA COM — a new ‘do-it-yourself’ intercom for yachts

A new ‘do-it-yourself’ talk-back 

system for the private yachtman 

system has recently been introduced by 

Barkway Electronics. 

Sea Com is a low cost, point-to-
point intercom/loudhailer system 
which is as simple to install as a 
car radio and is being sold in 
‘do-it-yourself’ kits comprising a 
master unit and two speakers.

Barkway claim Sea Com is the 
only system of its kind on the 
market and envisage good sales 
throughout the world.

High quality sound and water-
proof equipment guarantee com-
mands and answers will be heard 
correctly, even in the worst 
possible conditions and Sea Com 
also features an alarm warning 
tone to alert other shipping 
in bad visibility.

The sub units can be fitted in 
the fore and aft positions of the 
boat, increasing safety at sea not 
only through clear sound but by 
cutting down the amount of 
movement necessary on board in 
person to person exchanges.

The system is designed for con-

tinuous operation and can be left 
on in the standby position for 
monitoring from lookout posi-
tions in bad visibility.

The Sea Com control, or master 
unit consists of a heavy duty 
water-tight aluminium case coated 
in ribbed nylon and the system is 

fitted with high output loud-

speakers, hand microphone, vol-
ume control, speaker selector 

switch and tone alert button.

The equipment has a power 
output of 10 watts and can 
operate on 12 or 24 volts D.C. 

from the ship’s batteries.

Barkway Electronics Limited, 
Barkway, Royston, 
Hertfordshire SG8 8EE 

England, 

Telephone: Barkway (0531) 84/666

(1282 M)

Wire twisting plier

The Milbar Safety Twist wire 


twisting plier is a versatile tool 

that will handle wire locking of 

nuts, bolts, screws and caps, 


together with twisting wires and 
cable in electrical and electronic 
work. Serrated plier jaws grip the 
wire to be twisted and sliding lock 
holds jaws in position while the 
wire is twisted simply by pulling 
on bail-race spiral knob. The plier 
has an overall length of 10½" with 
polished head and black oxide 
finish handle. Actuator rod 
extension is 5° with 3½ turns per 
pull.

Toolrange Ltd., 
Upton Road, 
Reading RG3 4JA, 
Tel.: (0734) 29446 or 22245.

(1291 M)
Gould Advance OS 3500
60 MHz oscilloscope

Gould Instruments Division has launched a new 60 MHz dualtrace general-purpose oscilloscope, the Gould Advance OS3500, featuring a wide range of measurement facilities normally found only on higher-bandwidth instruments. Among the special features of the oscilloscope are comprehensive triggering facilities with a trigger bandwidth of DC to 100 MHz, and an optional add-on digital measuring unit for accurate measurements of amplitude, time and frequency.

The Gould Advance OS3500 oscilloscope uses an 8 cm x 10 cm high-writing-speed cathode-ray-tube with an accelerating potential of 12 kV to give a bright, easy-to-read display. The instrument is designed for portability; measuring 32.5 cm wide x 18 cm high x 46.5 cm deep, and weighing 10 kg, and the carrying handle also functions as a fully adjustable stand.

The instrument has two input channels, Y1 and Y2, which provide maximum sensitivities of 2 mV/cm over the full 60 MHz bandwidth, and a special control circuit is incorporated to nullify thermal drift.

The wide range of operating modes available on the OS3500 includes comprehensive delayed timebase and triggering facilities. Varnier control of sweep delay time allows accurate timing measurements to be made, and the delayed timebase can be started by the main timebase sweep or triggered after a preset sweep delay. The main and delayed timebase controls are completely separate.

For the study of complex waveforms, an alternate timebase sweep mode is incorporated, which allows the main timebase intensified) and delayed timebase sweeps to be displayed simultaneously. This mode is selected by a simple pushbutton, and can be used with single or dual-channel operation. The advantage of this mode is that an immediate relationship is established between the main timebase signal and the detail being swept by the delayed timebase, and changes in the settings of either timebase do not involve the operator in any further adjustments.

Among the comprehensive triggering facilities is a trigger input function, activated by a simple pushbutton, which allows continuous display of the signal triggering the main timebase, whether the source is internal or external. When the instrument is operated in the dual-channel and trigger-view modes, three traces are displayed, and this facility simplifies the procedure of setting the trigger level for single-event signals as well as establishing the presence of a trigger signal under difficult measuring conditions. For measurements on a circuit with its own system clock, the oscilloscope can display and be triggered from this signal, while the two main input channels remain free to study other important information.

Another feature which simplifies the triggering of complex waveforms is variable trigger hold-off, which can be continuously adjusted up to approximately one sweep length of the main timebase on most ranges.

Gould Instruments Division, Roebuck Road, Hainault, Essex IG6 3UE, Telephone: 01-500 1000

Add-on digital measuring unit

The new Gould Advance DM3010 digital measuring unit from Gould Instruments Division is designed to increase the basic accuracy of the OS3500 dual-trace 60 MHz oscilloscope in the measurement of amplitudes and time relationships. Offered as a factory- or service-fitted option to the OS3500, the unit provides a 3½-digit digital-voltmeter facility via a separate floating input, as well as increased voltage and time accuracies when switched to operate with the oscilloscope.

For time measurements, a second 'bright-up' section of the oscilloscope's main timebase sweep is introduced and controlled from the DM3010. The period between the first and second bright-up sections is accurately displayed on the instrument's light-emitting-diode display. For amplitude measurements, a second complete sweep of the channel 2 signal is introduced, and the bottom of this signal is adjusted to coincide with the top of the basic display to provide an accurate digital readout.

Operated as an independent digital voltmeter with the separate floating input, the DM3010 measures voltages from 200 mV to 1000 V DC, with a resolution of 100 µV and an accuracy of ±0.15% of reading ± one digit. Resistance and current can also be measured. The combined accuracy of the DM3010 and OS3500 is ± 1% of reading ± two digits for time and ± 2% of reading ± two digits for amplitude measurements up to 5 MHz. Above 5 MHz, the accuracy is conditioned by the vertical amplifier roll-off to -3 dB at 60 MHz.

The additional accuracy offered by the DM3010 is of particular use in applications such as the measurement of digital-circuit time relationships, including memory timing and propagation delays. Phase and risetime measurements can also be made more precisely.

Gould Instruments Division, Roebuck Road, Hainault, Essex IG6 3UE, Telephone: 01-500 1000

Low-cost keyboard subsystem

Electronic Brokers' new low-cost Model 771 ASCII Keyboard is especially suited for use with the latest inexpensive video terminal and display boards. The combination of the 771 keyboard subsystem, and a video terminal board mounted in the user's mainframe, provides an attractive and versatile cost-saving alternative to conventional one-piece CRT terminals. Compact, reliable, and rugged, the 771 Keyboard is ideal for use in small business, word processing, or softwaare development applications for personal, business, scientific or educational microprocessor systems.

Standard features include full ASCII alphanumeric section; convenient cursor control and numeric pad; two-key rollover for key event rate; Upper & lower case plus control codes; TTY mode for upper-case only operation; Timed autorepeat on all keys; all modes standard parallel interface; detachable industry standard connector, non-glare keycaps; robust steel desktop enclosure.

The 771 Keyboard is supplied fully assembled and tested, with complete documentation, requiring only power and data connections to the user's system for operation. Supplied mounted in an all-steel desktop enclosure, finished in textured IBM blue and black, this Keyboard is a perfect complement to modern microprocessor hardware.

The KB771 is priced at a modest £45.60, with discounts for quantity.

Electronic Brokers Limited, 49/53 Pancras Road, London NW1 2BG, Telephone No: 01-837-7781.
Now, the complete MK 14 micro-computer system from Science of Cambridge

VDU MODULE. **£33.75**
Display up to 16K memory (32 lines x 16 chars, with character generator, or 4096 spot positions in graphics mode) on UHF domestic TV. Eurocard-sized module includes UHF modulator, runs on single 5V supply. Complete ascii upper-case character set can be mixed with graphics.

POWER SUPPLY. **£8.10** inc. p & p.
Delivers 8V at 600 mA from 220/240 V mains – sufficient to drive all modules shown here simultaneously. Scaled plastic case, BS-approved.

MK 14 MICROCOMPUTER KIT
**£46.55** inc. p & p.
Widely reviewed microcomputer kit with hexadecimal keyboard, display, 8 x 512-byte PROM, 256-byte RAM, and optional 16-lines I/O plus further 128 bytes of RAM.
Supplied with free manual to cover operations of all types – from games to basic maths to electronics design. Manual contains programs plus instructions for creating valuable personal programs. Also a superb education and training aid – an ideal introduction to computer technology.
Designed for fast, easy assembly; supplied with step-by-step instructions.

CASSETTE INTERFACE MODULE. **£7.25** inc. p & p.
Store and retrieve programs on any cassette recorder. Use for serial transmission down single line at up to 110 baud (teletype speed), e.g. over telehore line, and to communicate between two or more MK 14s.

PROM PROGRAMMER.
**£11.85** inc. p & p.
Use to transfer your own program developed and debugged on the MK 14 RAM to PROM (74557) to replace SC105 monitor for special applications, e.g. model railway control. Software allows editing and verifying.

Science of Cambridge Ltd
6 Kings Parade, Cambridge, CAMBS., CB2 1SN.
Tel: 0223 311488.

To order, complete coupon and post to Science of Cambridge for DELIVERY WITHIN 14 DAYS. Return as received within 14 days for full money refund if not completely satisfied.

To: Science of Cambridge Ltd, 6 Kings Parade, Cambridge, Cambs., CB2 1SN.

Please send me:
☐ MK 14 standard kit @ £46.55.
☐ VM 14 standard kit @ £46.55.
☐ Extra RAM @ £4.14 per pair.
☐ RAM I/O device @ £6.97.
☐ Installation character generator @ £33.75.
☐ VDU module without character generator @ £20.85.
☐ VDU module with character generator @ £33.75.
☐ Cassette interface module @ £7.25.
☐ PROM programmer @ £11.85.
☐ Power supply @ £8.10.
☐ Full technical details of the MK 14 System, with order form.
All prices include p & p.

I enclose cheque/MO/PO for £_____________ (total).

Name:

Address (please print):

Delivery within 14 days.

E/10/79