Michael Schulze

TV Commercials Killer

Pause recording during ad breaks

Stop those annoying advertisement breaks spoiling your recordings with this ingenious circuit. Its basic ingredients are an overclocked Scenix SX28 microcontroller running some clever software, fast static RAM and a two-way 'code-learning' infrared remote control that obviates surgery to the family's precious VCR. Curious about the operation? Read on…

Anyone who has ever recorded a programme from a commercial broadcaster onto video or DVD will know the problem: either you have to stand over the recorder and press the 'STOP' button when the advertisements start (and then forget to press 'RECORD' when the programme continues), or you have to subsequently go through and edit out the breaks; or, of course, you can just put up with them. Alternatively, you can in some cases pay for services that are free of advertising.

How it's done

This advert killer, dubbed **'ViConti'** (for 'Video Continue', which, along with 'ViConti', is a registered trademark) uses the fact that, in general, a broadcaster's logo appears in the corner of the screen during normal programming, including feature films and the like, but that the logo disappears during advertisements (or commercial breaks as our US friends call them). The circuit must therefore:

- Determine whether the broadcaster's logo is present on the screen, and, if so, where;
- Monitor the picture to check when the logo disappears;
- send an infra-red command to the recording equipment (VCR or DVD) to pause recording;
- continue to monitor the picture and restart recording when the logo appears again.

Real-time image processing is relatively straightforward these days. PCs and special-purpose signal processors are up to the task, but these tend to be rather expensive. Run-of-the-mill microcontrollers are much too slow, and at this point this article would come to an abrupt end were it not for the Scenix (now known as Ubicom) SX28 microcontroller. Because of its incredibly high speed, this device provides a cheap alternative to signal processors in some applications.
The SX28 was originally specified to run at 50 MHz. Experiments have shown, however, that it can run without problems at at least 80 MHz. Meanwhile, 75 MHz devices are now standard, and a 100 MHz version was announced some time ago, although it is not yet available.

Now we have looked at the microcontroller, it is time for a quick look at the circuit diagram in **Figure 1**. The other important components are the familiar LM1881 video sync separator (IC9) and an ADC1175 analogue-to-digital converter. Also connected to the microcontroller are a 64k-by-4 static RAM type IDT61298 (IC5), for storing picture information, and an I^2C EEPROM. The RAM is controlled serially from the microcontroller using two fast counters (IC3 counting the pixels within a line, IC8 counting the lines of the picture). The ICS502 clock multiplier generates an 80 MHz clock for the microcontroller from the 20 MHz crystal. The A/D converter is clocked at 20 MHz.
The amplifier built around T1 and T2 raises the level of the video signal to be processed to about 2 V and the colour carrier is filtered out by L1, C5 and C6. The amplified signal is digitised to 8 bits by A/D converter IC2 and presented to port RC of the microcontroller.

The video sync separator extracts a line clock (BP) from the video signal, which is used to synchronise the microcontroller and drive the clamp circuit (IC7c and T3) in such a way as to ensure that the AC-coupled video signal has the correct DC offset applied. Also, the odd/even (O/E) signal is monitored by the software to enable the two video fields to be correctly assembled in the memory.
The infra-red signal that the microcontroller must use to stop the recorder during an advertisement break must first be learned from the recorder's remote control. This facility is provided by the circuit using a type SFH203A infra-red receiver. Originally the output of the receiver fed directly into a Schmitt trigger input, but here an ordinary NAND gate (IC7d) does the same job. The sensitivity of the receiver is deliberately low (the range is just a few centimetres), and so no amplification of the signal is required. After the Schmitt trigger circuit the signal needs to be taken to microcontroller input RB1 via the multiplexer constructed from IC7 and T4.
The two infra-red signals received from the remote control (for 'RECORD' and for 'PAUSE') are stored in serial EEPROM IC6, a 24C08. When needed, the signals are read from the EEPROM and sent out using infra-red transmitter diode LD1 (an LD271H).
The ViConti ad killer has no display. Instead, it shows its status using light-emitting diodes D3 and D4. These are driven using a single control signal, but nevertheless are capable of four different indications: both off, green on, red on, or both on.

Three steps

How does a human being recognise the broadcaster's logo? The characters and the style help, as does familiarity from, for example, television listings magazines. Unfortunately, our microcontroller doesn't read listings magazines, and doesn't have a *déja vu* input. How, then, can we get the microcontroller to recognise the broad-

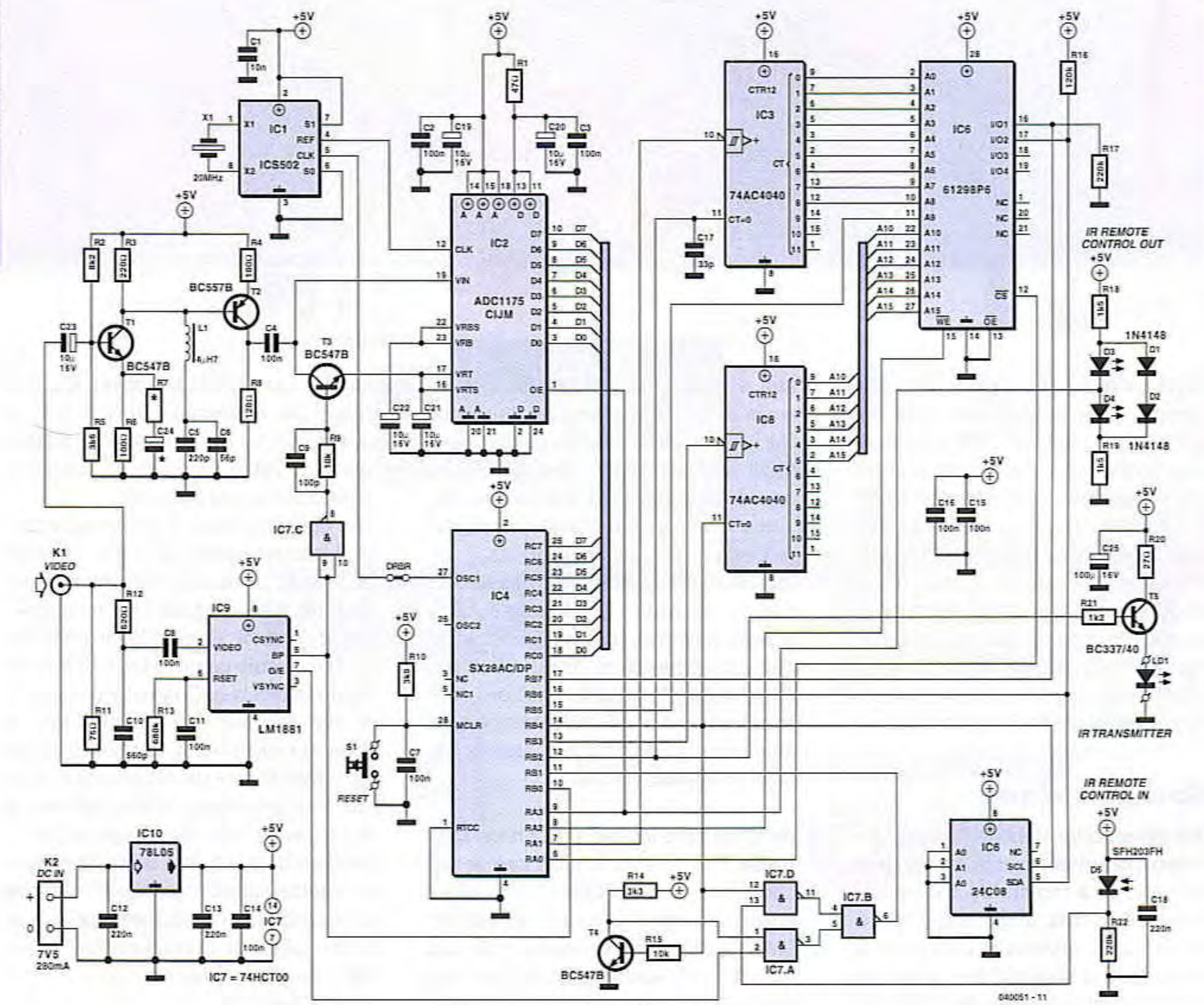


Figure 1. The ViConti advertisement killer uses an overclocked Scenix microcontroller.

caster's logo?

In general, the broadcaster's logo is the only static content in the picture, while the rest of the picture changes more or less quickly. Hence we simply need to check which area of the picture does not change other than at the moment when the logo is switched off for the advertisement break.

The logo must be recognisable on black-and-white televisions, and so it is sufficient to process a monochrome picture. In the area where the logo sits the pixels of the image always have the same brightness. In the digitised picture the samples always have the same value, whereas nearby samples will to some extent exhibit variations in brightness.

Phase 1

If we suppose that the broadcaster's logo has a minimum brightness of say 20 %, then we might expect that nearby pixels will sooner or later (and hopefully before the first advertisement break) fall below this threshold, depending on the nature of the programme. We can then deduce that they do not belong to the logo. This is enough to recognise the logo and to build up a black-and-white map of the image in the first bit of the picture memory. Black is represented by 0, white (i.e., logo) by 1. **Figure 2** shows part of an original picture along with the filtered logo.

The microcontroller loads the 8-bit pixel value from port RC, compares it with a threshold value, and, if it is recognised as not belonging to the logo, writes a black pixel into the corresponding position in the picture memory (in the first bit). The pixels in the memory are initialised to white.

Because there are only 136 bytes of internal RAM available in the SX28, an external memory is required. In order to help economise on port pins, two fast counters are added to drive the address lines of the memory.

The assembler program in the SX28 microcontroller, running at 80 MHz, takes precisely nine machine cycles (i.e., nine instructions) at 12.5 ns each to process each pixel (see **program snippet 1**).

From this we can calculate that in the visible part of each television scan line, which lasts approximately 52 μs, we have a total of about 460 pixel samples. The line-by-line synchronisation of the program with the incoming scan lines is achieved using an external interrupt obtained from the line signal from the LM1881.

Only a stripe of 128 lines from the upper third of the picture is sampled, since this is where the broadcaster's logo is generally located. At regular intervals a test is made to determine whether the white pixels (i.e., those considered as belonging to the logo) all fall within a reasonably-sized rectangle in either the upper right-hand or upper left-hand corner of the screen. If this is the case, the second phase begins.

PROGRAMM SNIPPET 1

```
  mov  W,#$02                ;Pixel counter: 460 Pixel per line!
  mov  $1E,W
  mov  W,#$CD
  mov  $1D,W                 ;Loop to check a picture line

:loopl setb   RA.0           ;Reset ext. memory write pulse?
  clrb RA.1                  ;Reset ext. pixel counter clock?
  mov  W,#$33                ;Threshold for logo search
  mov  W,RC-W                ;When W < RC then Carry Flag set
  rl   RA                    ;Ext. memory write pulse
                             ;when Carry not set
                             ;Ext. pixel counter clock
  decsz           $1D
  jmp  :loopl
  decsz           $1E
  jmp  :loopl
```

Figure 2. Above, a section of the original picture in black and white; below, the filtered logo.

Figure 3. Section of the original picture shown next to the stored version, which is compared with the logo below. Below, the white pixels represent the logo, the black pixels the reference pixels, and the grey pixels are neutral. The numbers show the threshold and reference values.

Phase 2

Since it is desirable to be able to react to the absence of the logo very quickly, ideally within a fraction of a second, we need criteria to determine whether or not the logo is displayed which are independent of changes in the picture content. Here again the black-and-white image must suffice. The logo stored in the picture memory consists of a few hundred pixels and has a characteristic average brightness value, which is usually rather different from the average value of the surrounding pixels. Typical values for the average brightness in the logo and for that in the surrounding pixels are determined in this phase before monitoring begins. The size of the logo is determined by a line-by-line search through the picture memory looking for the outermost white pixels. A border three pixels wide is added, and this gives the coordinates defining the 'logo rectangle', i.e. the area of the picture that is monitored for the presence of the logo. The number of white pixels that make up the logo is now counted, and exactly the same number of black pixels is added at random into the logo rectangle. The remaining pixels are set to neutral ('grey') by setting the second bit in the picture memory.

Phase 3

In phase 3 the picture is monitored in real time, that is, frame by frame. In each frame the average brightness of the logo pixels is calculated, as is the average of the same number of pixels not belonging to the logo, spread out over a representative area. If the difference between these two values falls below a preset threshold several times, this indicates that the logo has disappeared.

PROGRAM SNIPPET 2

```
  mov  W,$1D              ;Contains width of logo rectangle
  mov  $04,W

:loop4 setb   $04.4
  nop
  mov  W,RC               ;xxx Byte, read image from ADC into RAM
  mov  $00,W
  nop
  incsz       $04
  jmp  :loop4
```

PROGRAM SNIPPET 3

```
  mov  W,$1D
  mov  $04,W

:loop5 inc    RA          ;xxx Byte, read from RAM and add
  setb $04.4
  mov  W,$00
  snb  RB.6               ;Memory bit, ext. image memory
  jmp  :grey
  sb   RB.7               ;Memory bit, ext. image memory
  jmp  :blck
  add  $09,W              ;Add white
  snb  C_Flag
  incsz       $0A         ;Sum white in $0B, $0A, $09
  dec  $0B
  inc  $0B
  dec  RA
  incsz       $04
  jmp  :loop5
  ret

:blck  add    $0C,W       ;Add black
  snb  C_Flag
  incsz       $0D         ;Sum black in $0E, $0D, $0C
  dec  $0E
  inc  $0E
  dec  RA
  incsz       $04
  jmp  :loop5
  ret

:grey  jmp    :x3         ;Dummy for grey pixels
:x3    jmp    :x4
:x4           dec    RA
       incsz  $04
       jmp    :loop5
       ret
```

Figure 3 shows a segment from an original image, and, next to it, the stored version which is compared to the logo below. Here the logo pixels are shown as white, reference pixels as black, and neutral pixels as grey. The numbers show the threshold and reference values.

The bit samples from the infra-red remote control stored in the EEPROM which represent the command to stop recording are sent to the infra-red transmitter. With luck they are then detected by the recorder, which will stop recording.

The programme continues to be monitored. The average brightness difference threshold for detecting the return of the broadcaster's logo is now set somewhat higher. As soon as the logo is clearly detected, the infra-red transmitter sends out the command to continue recording.

In the monitoring phase the position of the logo in the television picture is known. Line by line, just the sequence of pixels belonging to the logo is selectively stored in the internal RAM of the microcontroller. This is carried out at exactly the same resolution as in phase 1, in precisely nine machine cycles per pixel (see **program snippet 2**).

In the remaining time until the end of the scan line the stored pixels are processed according to whether they are logo pixels (white), reference pixels (grey) or neutral pixels (grey). The various brightness sums are calculated (see **program snippet 3**).

When all the pixel brightnesses have been added together, the difference between the sums for logo pixels and reference pixels is calculated, and compared with a suitable threshold value which depends on the size of the logo. When frames which exhibit too small a difference are encountered several times in a row, the logo is judged to be absent and the infra-red command to stop the video recording is transmitted.

A software module to drive the I²C bus as a so-called 'virtual peripheral' is available from Scenix and has been modified here to drive the EEPROM to store the infra-red remote control command codes for the recording device to be controlled.

Since there are only two different infra-red commands to send to the recorder, we can offer a learning function. In

Note:

For correct operation of the TV ad killer the following are essential:

- **The broadcaster's logo must appear in the upper third of the screen, in a fixed position (as is generally the case).**
- **A good video signal is required. If the picture is not perfectly in sync, the logo can jitter (even though this may not be evident to the eye). The logo may then not be correctly recognised.**

learning mode, incoming pulses are sampled using an interrupt and quantised to a timebase. The count values are stored permanently in the EEPROM. In use, the values are fetched from the EEPROM and stored in RAM, so that the sequence of pulses for the code used can be reconstructed with good accuracy and sent out using the infra-red transmitter.

In use

A prerequisite for satisfactory operation is a very good video signal. If synchronisation is not perfect, the logo can jitter (even though this may not be evident to the eye). The logo may then not be correctly recognised.
When power is applied, or after the reset switch is pressed, the program runs in four stages. In the first stage the two infra-red command codes for controlling the recorder can be set.

LEARN IR CODES
turn on green LED
wait for IR signal

You now have approximately four seconds in which to press the required button (for example, 'REC') on the remote control. The infra-red transmitter on the remote control should be just a few centimetres from the receiver diode on the advertisement killer. If no infra-red signal is received within four seconds, the program jumps to 'FIND LOGO'; otherwise, it proceeds as follows:

IR signal recognised
green LED blinks
read IR signal
store code in EEPROM
turn off green LED
turn on red LED
wait for IR signal

The procedure for learning the 'PAUSE' code is the same, except that only 2 s is allowed. If no infra-red signal is received within 2 s, the program jumps to FIND LOGO. Otherwise it proceeds as follows:

IR signal recognised
red LED blinks
read IR signal
store code in EEPROM
turn off red LED

The record and pause commands (you can of course use any other remote control commands you choose) need only be programmed in the first time the unit is used, or if you wish to change the commands. Otherwise, simply turn the unit on and wait!

FIND LOGO
turn red and green LEDs on

The following procedure is executed for the first and second fields:

wait for interrupt at top of picture
wait for line interrupt
process lines 33 to 96

If no logo can be recognised in the external memory, the program remains in this loop; otherwise it jumps to:

ANALYSE LOGO

This process has already been described above, under 'Phase 2'.

MONITOR LOGO

The following procedure is executed for the first and second fields:

wait for interrupt at top of picture
wait for line interrupt
process lines 33 to 96

After processing, execution can proceed in three possible ways.

Logo present
green LED flashes periodically
return to MONITOR LOGO

Logo newly disappeared
red LED flashes periodically
proceed to TRANSMIT IR CODE 1

COMPONENTS LIST

Resistors:
(all metal film 0.25 W, 5 %)
R1 = 47Ω
R2 = 8kΩ2
R3 = 220Ω
R4,R6 = 100Ω
R5 = 3kΩ6
R7 = not fitted
R8 = 120Ω
R9,R15 = 10kΩ
R10 = 3kΩ9
R11 = 75Ω
R12 = 620Ω
R13 = 680kΩ
R14 = 3kΩ3
R16 = 120kΩ
R17,R22 = 220kΩ
R18,R19 = 1kΩ5
R20 = 27Ω
R21 = 1kΩ2

Capacitors:
C1 = 10nF 63V NPO
C2-C4,C7,C8,C11,C14-C16 = 100nF 63V X7R
C5= 220pF 63V NPO
C6 = 56pF 63V NPO
C9 = 100pF 63V NPO
C10= 510pF 63V NPO
C12,C13,C18 = 220nF 63V X7R
C17 = 33pF 63V NPO
C19-C23 = 10µF 16V E2.5-5
C24 = not fitted
C25 =100µF 16V E2.5-6

Semiconductors:
D1,D2 = 1N4148
D3 = LED, 5mm, red, low current
D4 = LED, 5mm, green, low current
D5 = SFH203FA (Infineon)

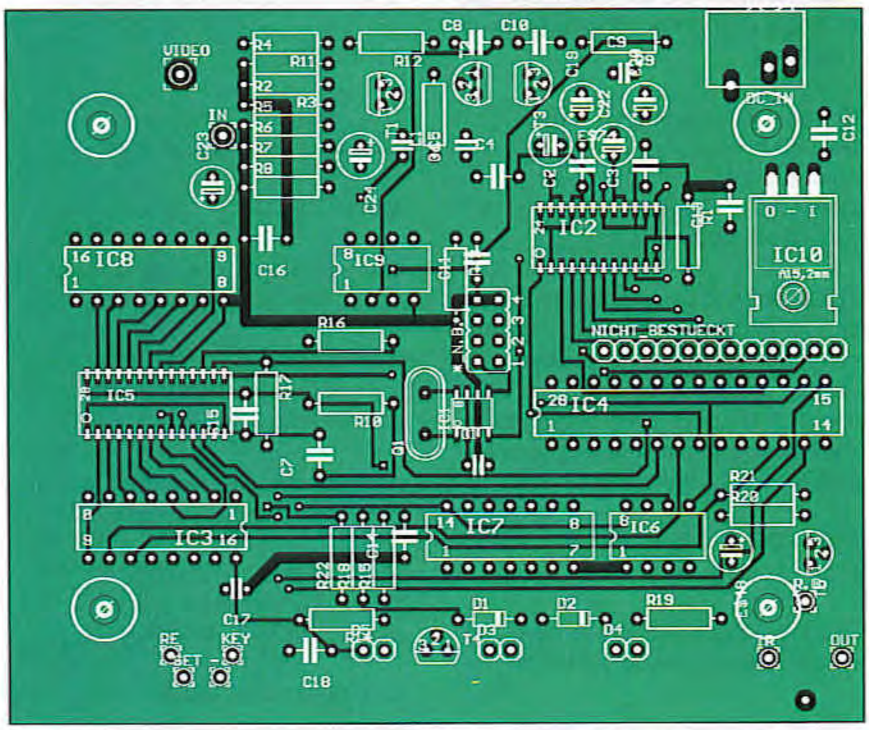
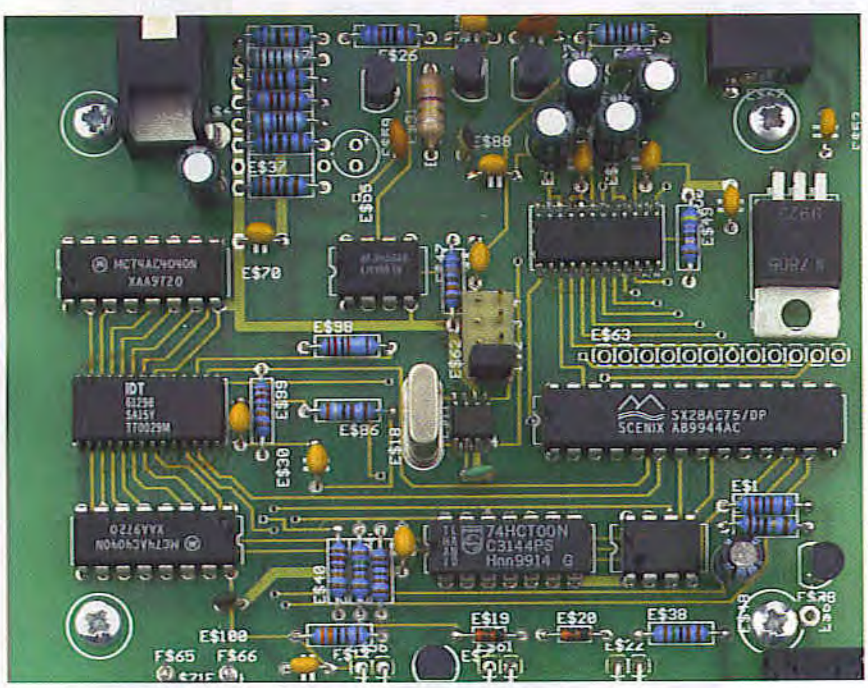


Figure 4. Component mounting plan for the double-sided printed circuit board. The trickiest character to deal with is the SRAM in a 28-pin SOJ package.



Logo newly appeared
green LED flashes periodically
proceed to weiter mit TRANSMIT IR CODE 2

TRANSMIT IR CODE 1
Fetch first learned infra-red code from EEPROM and transmit it three times
Proceed to MONITOR LOGO

TRANSMIT IR CODE 2
Fetch second learned infra-red code from EEPROM and transmit it three times
Proceed to MONITOR LOGO

Construction

The double-sided printed circuit board for the advertisement killer is designed to fit exactly in the suggested enclosure, without the need to wire any of the components using flying leads. Populating the board, as **Figure 4** shows, may present a few difficulties, since the clock multiplier, the A/D converter and the RAM are SMDs. The

T1,T3,T4 = BC547B
T2 = BC557B
T5 = BC337/40
LD1 = LD271-H (Infineon/Osram) *
IC1 = ICS502M (ICS)
IC2 = ADC1175CIJM (National)
IC3,IC8 = 74**AC**4040
IC4 = SX28AC/DP (Scenix, now Ubicom)*
IC5 = 61298P6 SOJ28-3 (IDT)
IC6 = NM24C08N08E (Fairchild)
IC7 = 74HCT00
IC9 = LM1881N08E (National)
IC10 = 78L05

Miscellaneous:

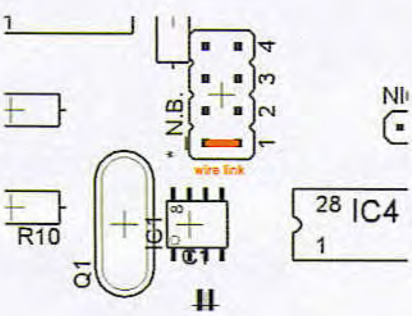
X1 = 20MHz quartz crystal (HC49U-H)
L1 = 4µH7
K1 = Cinch socket (Lumberg WBTOR 1)
K2 = mains adaptpr socket 2mm (Lumberg NEB/J 21R)
K3 = miniature jack socket (Lumberg KLBR2)
S1 = pushbutton with make contactt (Schurter 1301.9502, no cap)
Mains adaptor, 7.5-9 VDC, 300 mA
Enclosure (e.g., Woehr Bernic Desk Top Enclosure 2011S, www.woehrgmbh.de)

IR transmitter head
Jack plug (Lumberg KLS2SL)
IR sender LD271-H (Infineon), see above

Suggested supplier
Ing. Büro Schulze
Obere Ringstrasse 7
D-79859 Schluchsee
Germany.
Tel./fax: +49 7656 9173
Email Mschulze99@web.de

* hex code file **040051-11**, Free Download,

Figure 5. A jumper is fitted to the programming connector so that a clock is provided to the microcontroller.



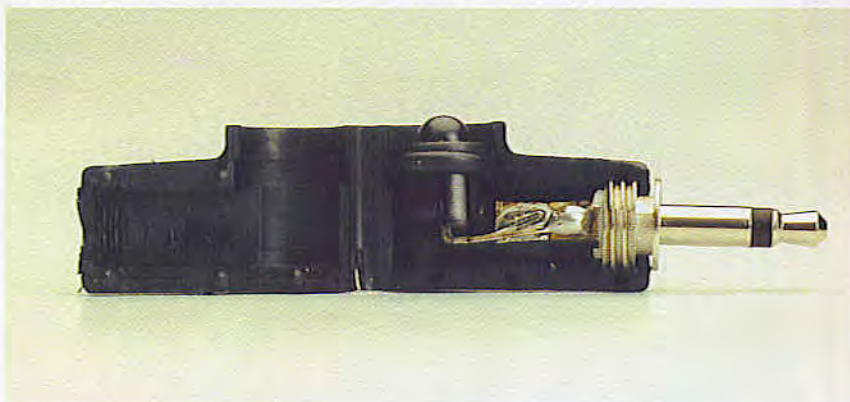


Figure 6. The infra-red transmitter unit is made from a right-angled jack plug, in which the diode is soldered in place of the usual cable.

first two ICs can be soldered using a fine-tipped iron and a steady hand, but the SRAM comes in a SOJ (small outline J-lead) package, whose pins curl under the IC itself. The following highly effective, if somewhat brutal, method is recommended.

1. First fix the device in position on the printed circuit board by carefully soldering two diagonally-opposite pins.

2. Solder all the remaining pins as quickly as possible, not worrying about any solder bridges that may be form between the pins. A normal rather than a fine-pointed bit is preferable, since it can be used more quickly.

3. Lay a length of solder wick loaded with flux across the soldered connections, and run a hot iron along it, over the pins. With luck all the excess solder will have been removed and the joints will have a satisfactory appearance. You must of course check that all the solder bridges have been removed. It is important that the wick holds enough flux and that the job is done quickly.

All the other components are of the normal leaded type and should not present any difficulties. Of course, you must observe the correct polarity for diodes, electrolytic capacitors, transistors and ICs.
The header in the middle of the printed circuit board is only required for in-circuit programming of the microcontroller (using the Parallax SX-Key). For normal operation simply fit a jumper in position 1 as shown in **Figure 5**: this ensures that the clock is provided to the microcontroller.

Finally, **Figure 6** shows how the infra-red transmitter is assembled. The transmitter diode is soldered to a jack plug and bent in such a way that the combination can be eased into a right-angled jack plug housing. The diode can be fitted with a plastic clip to ensure that it sits firmly in place. The whole arrangement can be rotated in the socket and so can be pointed accurately at the receiver diode in the recorder.

(040051-1)

Please note: This circuit has not been tested or post-engineered by the *Elektor Electronics* design laboratory. The use of the Viconti unit described in this article may not be legal in all countries.