ELEKTOR ELECTRONICS
THE ELECTRONICS & COMPUTER MAGAZINE
FEBRUARY 1998
£ 2·75

FREE SUPPLEMENT
PC TOPICS

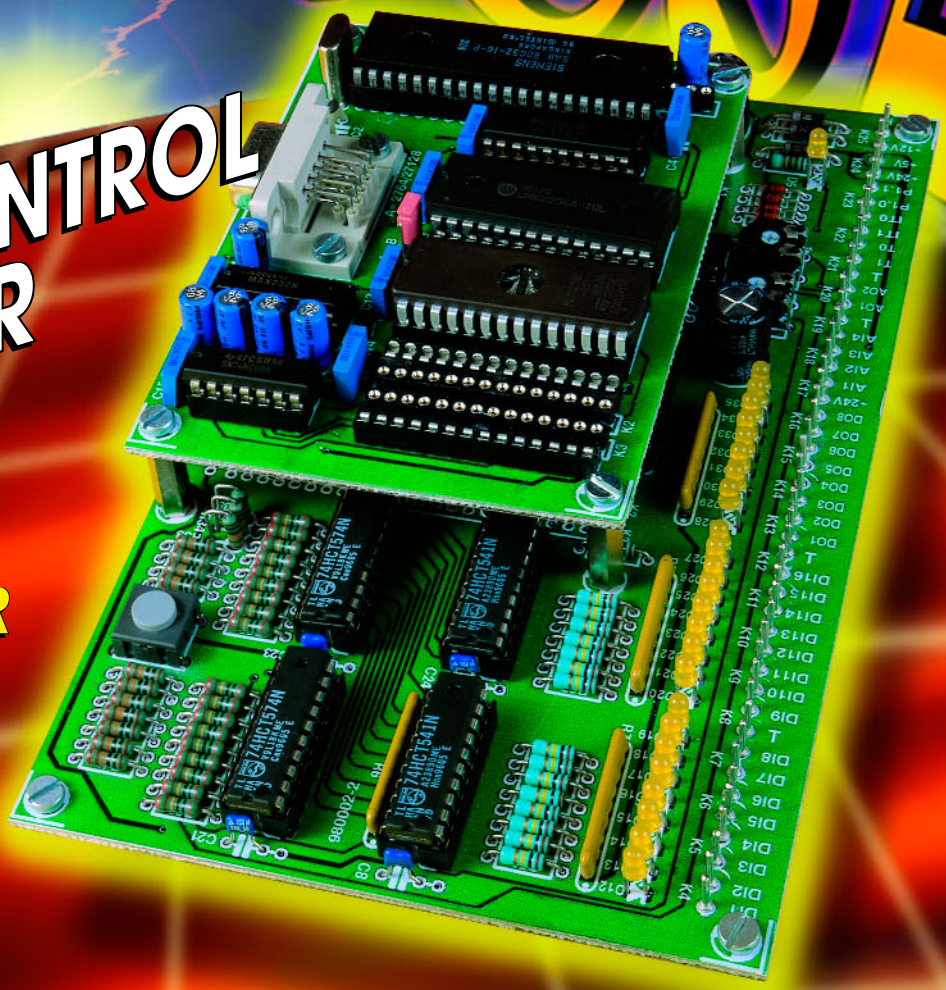
STAGE LIGHTS CONTROL WITH DMX512

80C32

80C32 BASIC CONTROL COMPUTER

VFO STABILIZER

AVC FOR PCs

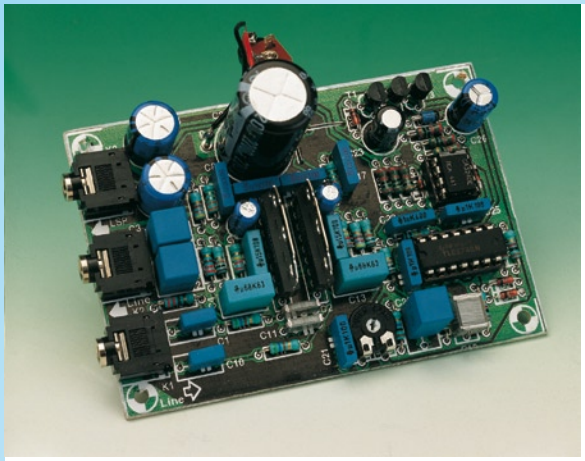


ELEKTOR ELECTRONICS

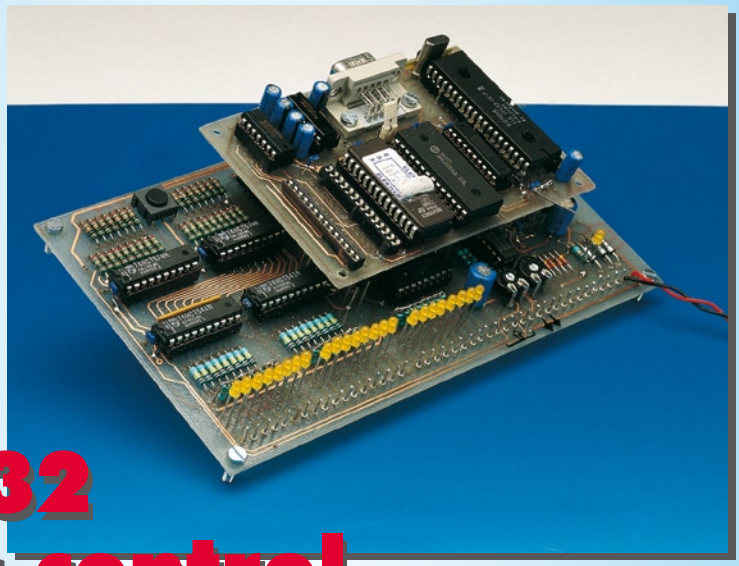
CONTENTS

February 1998 Volume 24
Number 263 ISSN 0268/4519

FREE 16-page SUPPLEMENT
PC Topics
Centrefold



AVC for PCs
24



30
80C32 BASIC control computer

18
Frequency display & VFO stabilizer

ABC AUDIT BUREAU OF CIRCULATIONS
CONSUMER PRESS

The term 'computer hobbyist' brings to mind the image of a lonely father sitting in a spare bedroom, typing cryptic machine-language code deep into the night. The wife is of course also lonely, which in the course of time results in the computer finding a place in the living room, in spite of its rather hideous appearance.
The space which thus comes free in the bedroom is soon filled, at first with children and after that with a second computer. And with the second computer comes the need for a network.

a simple PC network

a cable is all you need...

By Peter Smit

In addition to allowing communication via e-mail, interconnecting two or more computers has the advantage that files, printers, modems, disk drives, ZIP drives and CD-ROM drives can be shared. With the Internet boom it has become totally 'in' to run 3D games in multiplayer mode. With this option, multiple persons can participate in the game at the same time. They can play together in a three-dimensional maze or fly through space together. Some well-known games which have this option are Doom, Duke Nukem 3D, Quake, Outlaws, X-Wing versus T-Fighter and so on. Some of these run on a central server via the Internet, but they can also be run via a TCP/IP network, an IPX network, a modem or a null-modem link.

The type of link which is most appropriate for interconnecting individual computers depends on the application which we intend to use via the link.

Null-modem cable

A null-modem cable is a good choice when relatively small data volumes need to be transferred. This inexpensive serial link is only suitable for transferring small amounts of data and for playing some multiplayer games. The maximum speed which the RS232 port can achieve with such a cable is 115,200 baud. For serial data transmission with 1 start bit, 8 data bits and 1 stop bit, this results in an effective transfer rate of 10,250 bytes per second (36 MB/hour). For this data rate, both computers must be fast (486 or better) and should preferably be fitted with a type 16550 UART (Universal Asynchronous Receiver/Transmitter).

One can easily check whether such a chip is present by means of the program MSD (provided standard with DOS). When MSD is run, it displays the type of UART present for each COM port: 8250, 16450 or 16550. An 8250 can only manage 9600 baud. Starting with AT machines one finds only the faster types (16450, 82450 and 16550), which can handle up to 115,200 baud. Many 'snoop' programs report the slower 8250 when a 16450 is fitted. Only the 16550 has a

buffer. Modern internal modems and Pentium motherboards have a 16550 as standard. Multitasking operating systems, such as Windows 95, Windows NT and Linux, can experience timing problems if a UART buffer is not present. This shows up as lower data rates, loss of data packets etc.

It's advisable to disable the 16550's FIFO buffer for the port to which the mouse is connected. Problems such as the mouse pointer 'freezing' after a few movements can sometimes be caused by this buffer. In Windows95 the buffer can be disabled via the Control Panel/System/Device Manager/Ports/ COM1 (mouse)/Properties/ Settings/Advanced/Use FIFO Buffers. In case of communication problems the FIFO speed can also be somewhat reduced via this route. With Windows 3.11 one must include a line in SYSTEM.INI under the heading [386Enh]: 'COM1FIFO= 0' disables the FIFO buffer for the mouse connected to the COM1 port. Always make a copy of SYSTEM.INI before making any changes!

Now that we've dealt with UART pitfalls, let's return our attention to the null-modem cable. Such a cable provides a link between the serial (RS232) ports of two computers. The RS232 port was originally intended to be used for interconnecting a DTE (Date Terminal Equipment) and a DCE (Data Communication Equipment). A D25 cable between a computer and a modem is an example of such a DTE/DCE link.

A null-modem cable, by contrast, is used for a DTE/DTE link. It requires that certain leads be interchanged between the two connectors. The most important of these are TxD (Transmit Data) and RxD (Receive Data). These two leads plus a ground lead represent the simplest possible 'three-wire' null-modem cable. The only problem with a three-wire link is that there can be no hardware handshaking between the two computers.

In a three-wire cable, the connector pin for the signal which asks whether data can be sent (Request to Send) is connected directly to the pin which is intended to receive the answer from the other computer (Clear to Send). This results in a sort of 'narcissistic' connection: the computer wonders whether it can send data, while 'thinking' that it is talking to a second computer. And like a true narcissist it naturally provides the answer to its own question. In other words, "Can I send? Yes, I can always send and I decide that for myself."

Whenever the two computers do not have the same level of performance, the slower one must be the *host* machine, since the host machine determines the data transfer rate. If there are communication problems with the link, it's a good idea to check whether data transfers in the reverse direction work well. If this is the case, then there is a timing problem between the two computers. In order to properly solve such a problem we must use a connection with more than three leads, since this is the only way to have hardware handshaking (see **Figure 1**). Programs which only recognize hardware handshaking will thus not work with a three-wire null-modem cable.

A full null-modem connection consists of 7 leads. A 25-pin or 9-pin sub-D connector is used. A male connector is always used at the back of the computer, so that we must use two female connectors for the cable. Normally the mouse is connected to the COM1 port via a 9-pin connector, so that the 25-pin connector of the COM2 port is usually used for this sort of experiment.

There are several types of null-modem connections. The most expensive solution, which is also the most flexible, consists of two universal modem cables together with a null-modem adapter. A universal modem cable is a 25-lead cable which has a 25-pin male connector at one end and both a 25-pin female connector and a 9-pin female connector at the other end. A null-modem adapter is a small block fitted with two 25-pin female connectors. We connect the adapter between the two cables, using the single 25-pin connectors of the two cables. With this combination we can cope with all D9 and D25 COM ports. With two 1.8-metre cables, such a combination costs approximately £15.

For roughly £10 we can make do with a single universal modem cable and a null-modem adapter. However, the adapter has threaded posts which mate with the fixing screws of the cable connector. These are in the way if the adapter is to be plugged directly into the computer's COM-port connector. It's possible to dismantle the connector and remove these posts on one side of the adapter so that it can be plugged into the COM-port connector. However, this is not a particularly elegant solution, since the modified adapter cannot be secured to the connector.

One can also purchase a D25 serial female/female interconnection cable and modify it oneself according to Figure 1, at least if it does not have

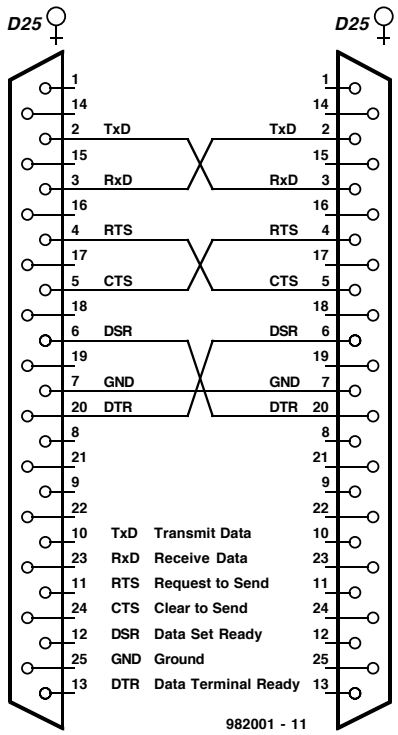


Figure 1. Wiring scheme for a full DTE/DTE null-modem cable.

moulded-on connectors.

The ease of use of a null-modem link is largely determined by the software used. The PC BIOS can (or at least could) only manage 19,200 baud. All MS-DOS null-modem software thus accesses the UART registers directly, rather than via the BIOS. This is the only way in which it is possible to achieve a 115,200 baud data rate.

The best-known interconnection software is LapLink, but Norton Commander also has a Link option. From MS-DOS 6.x onwards, Interlink provides a standard means for interconnecting two computers, and Windows 95 has the Direct Cable Connection option. For occasional use I prefer Norton Commander (V4.0). Since this program is anyhow often used to provide a user interface, it is natural to also use it for the link. Version 4.0 has the additional advantage that the contents of multiple directories can be selected concurrently and copied. With a bit of patience it is possible to transfer an entire hard-disk partition in one operation. Of course, at a data rate of 35 MB/hour this does not go particularly fast.

In Norton Commander one selects Menu/Right or Left/Link, by means of which the first computer is configured as the master and the second as the slave. (Pay attention to the selection of the correct COM ports.) The drives of the slave computer then appear at the master as a normal window within

which one can select, copy and delete items and create subdirectories. Norton Commander V4.0 also can be used with a parallel interconnection cable for a faster link.

An Interlink cable

If we wish to transfer data more quickly, then we will want to use a modified parallel cable. LapLink was the first to introduce this possibility, and Norton later followed suit in Norton Commander starting with Version 4.0. Microsoft adopted this idea starting with MS-DOS6.x in the form of Interlink. In Windows95 this option is called *Direct Cable Connection*. A disadvantage of the parallel link is the relatively short distance which can be bridged (a few metres). The greatest advantage of a parallel link is its speed. The maximum speed depends on the type of printer port and the sort of cable which is used.

Various types of parallel link cables can be used. First, there are standard 4-bit versions which work with LapLink, Norton Commander and Windows95. Then there is a modified 8-bit version for Norton Commander. An ECP cable, which is faster, can only be used with ECP/ECP ports. Finally, there is an 'intelligent' Universal Connection Module (UCM) which checks for itself which types of parallel ports are present and configures itself either in standard 4-bit mode or ECP mode. In standard 4-bit mode the transfer rate lies between 40 and 70 kbyte/s. In ECP mode the transfer rate can range up to 400 kbyte/s.

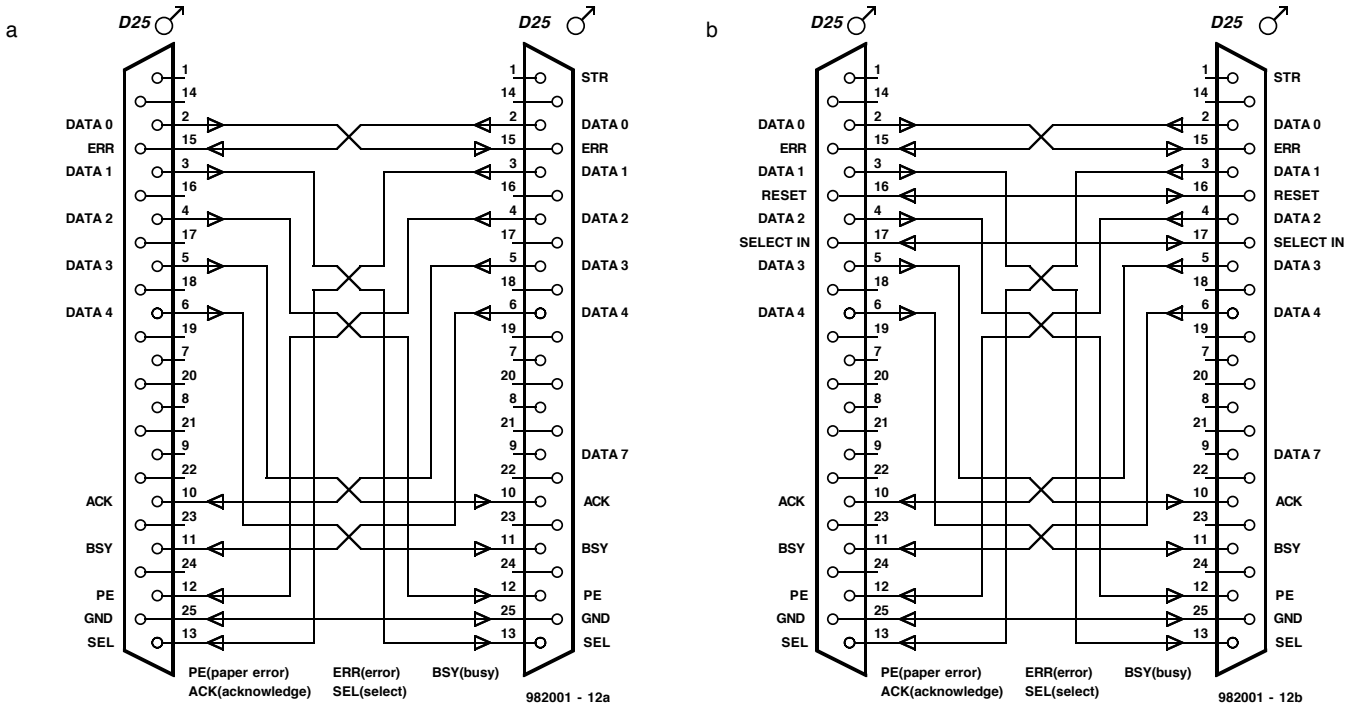
Officially, a Centronics port is bidirectional (data can be transferred in both directions). In practice, a more economical implementation has been used, since for years printers just swallowed data and only protested when they were out of paper. Modern printers are more mature and more inclined to 'talk back', which has resulted in the development of bidirectional parallel interfaces (PS/2, EPP and ECP).

The old-fashioned Centronics interface has 8 data lines which can only be used to send data. There are three registers for controlling the parallel port: an 8-bit data register (read/write), a 5-bit status register (read-only) and a control register (read/write). The status lines are 'misused' by the Interlink cable in order to allow data to also be received. If the incoming 5 status lines are cross-connected to 5 outgoing data lines at the 'other end', a 5-bit parallel link is created. The basic cable results from cross-connecting pins 2 through 6 with pins 11, 10, 12, 13 and 15; pin 25 serves as the ground lead (see **Figure 2a**). If better screening is desired, then additional ground lines are used (pins 18 and 24). This cable thus allows 5 bits to be transferred in parallel. One of these is used for handshaking, leaving a 4-bit data stream.

The control register contains a bit which allows a bidirectional parallel port to be enabled or disabled. If one were to interconnect the outputs (data lines) of two standard parallel ports, there's a good chance that one set of outputs would not survive the experiment. Moreover, it's not possible to read in data via a standard parallel port: what is read from the data register is not the data presented to the port from an external source, but the data which were last written to the register by the computer itself.

With a true bidirectional port, the output lines are placed in a high-impedance state when the bidirectional mode is enabled (control bit set). Data which are written to the data register remain in a buffer without being transferred to the output data lines. When the data register is read, the current status of the port's data pins is read. From this it is apparent that a bidirectional port does not support duplex operation. The bidirectional control bit serves to make the port act as either an output or an input. By properly using one or more status lines for handshaking, the two interconnected ports can be correctly switched between send and receive modes, so that true 8-bit data communication is possible. In order to use a parallel port in this manner, one must use an 8-bit ECP cable; a standard 4-bit cable cannot be used. An ECP cable interconnects all 8 data lines and cross-connects various status lines (see **Figure 3**). If such a cable is used with a non-bidirectional port, damage to the outputs (data lines) can result.

Figure 2. Three types of parallel-interface cables between two computers: a simple 4-bit cable (a), a 4-bit cable for Windows 95 (b) and an 8-bit cable for Norton Commander (c).



The modern Extended Capabilities Port (ECP) is bidirectional and has an extended control register which (among other things) allows the operating mode (SPP, EPP or ECP) to be configured. The ECP port uses an interrupt (IRQ7 for LPT1, IRQ5 for LPT2) to capture the data stream; it has a FIFO buffer and DMA support, it has a decompression mode and it handles its own handshaking. It is thus better suited for multitasking operating systems and much faster for the Direct Cable Connection. The transfer rate ranges between 200 and 400 kbyte/s. The transfer rate of a normal 4-bit cable is also increased when it is connected between two ECP ports. In order to achieve the highest data rates (200 to 400 kbyte/s) a special ECP or UCM cable must be used.

The way that a UCM cable works can be gleaned from the various information scattered about the Internet. There is software in C++ which is able to recognize whether a specific port is standard or ECP. An electronic switch which is normally in 4-bit mode can be switched to ECP mode via a software-generated status signal.

As soon as a parallel port's ECP mode is enabled (via the BIOS), the port uses an interrupt (7). This frequently results in an interrupt conflict if a sound card is installed. If the ECP mode is to be used, then the sound card must be configured to use a different interrupt (5).

A search via the Internet yielded six different descriptions of standard 4-bit Interlink cables. We have combined two of these plus the Norton Commander cable into a single figure (see **Figure 2**).

The simple version (**Figure 2a**) works with LapLink, Fastlynx, Ebox, XTLink and MS-DOS6.x Interlink. The version for Windows95 (**Figure 2b**) has two extra leads which interconnect pins 16 and 17.

The only connection scheme which supports 8-bit data transfers with standard parallel ports is the Norton Commander Link option (V4.0 and V5.0). Norton Commander uses three additional bits in the control register to allow the remaining 3 data bits to be read. A true 8-bit link results from the added connections (pins 1/7, 7/1, 9/16, 16/9 8/14 and 14/8) without using bidirectional ports (**Figure 2c**). In ECP mode, the control register cannot be used to read data. The 8-bit Norton Commander cable thus cannot be used with a parallel port operating in the ECP mode.

An important difference between the Windows95 cable and the Norton Commander cable relates to pin 16. This can either be interconnected with pin 16 or cross-connected with pin 9. One must chose between an 8-bit cable version which only works with Norton Commander or a 4-bit Norton Commander version which can also be used with Windows95. The Windows95 version is the most compatible. Unless you plan to work only with Norton Commander, I would advise you to choose the Windows95 version.

The ECP cable can only be used with bidirectional ports. For those of you who wish to experiment with an ECP cable, a diagram from Microsoft is included (see **Figure 3**). Remember that a standard port can be damaged if an ECP cable is used with it! In a working situation in which two specific computers are always connected to each other via ECP ports, this does not present a problem. On the other hand, something can always go wrong, such as for example if someone just wants to quickly copy a file to his 486 notebook.

In a situation in which various persons work with a variety of computers (including notebooks), use a UCM cable. Such a cable is fast, safe and problem-free in use. With Windows95, a UCM cable approaches the capabilities of a true network with regard to functionality and speed. There is even ODI software available for such cables, which allows them to be used in a Novell or Lantastic network. The UCM cable is less expensive than a separate network adapter attached to the parallel port of a PCMCIA network adapter. This is also a good option for connecting a notebook to a computer which is itself connected to a network. The notebook then receives a network connection via the UCM cable. For more information refer to the Parallel Technologies Internet site (*http://www.lpt.com/*). If the flexibility of the UCM cable is not necessary, then the money (£50) can be better invested in a network. A network is significantly faster and can easily be extended.

(982001)

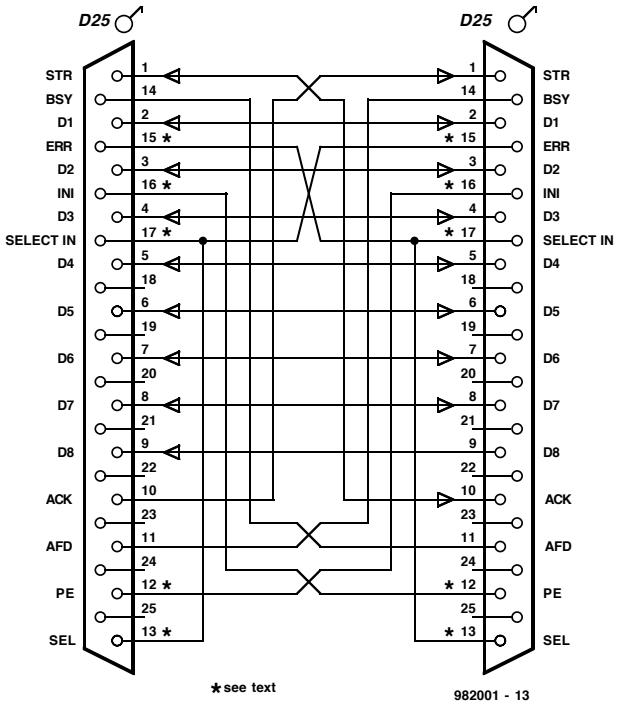
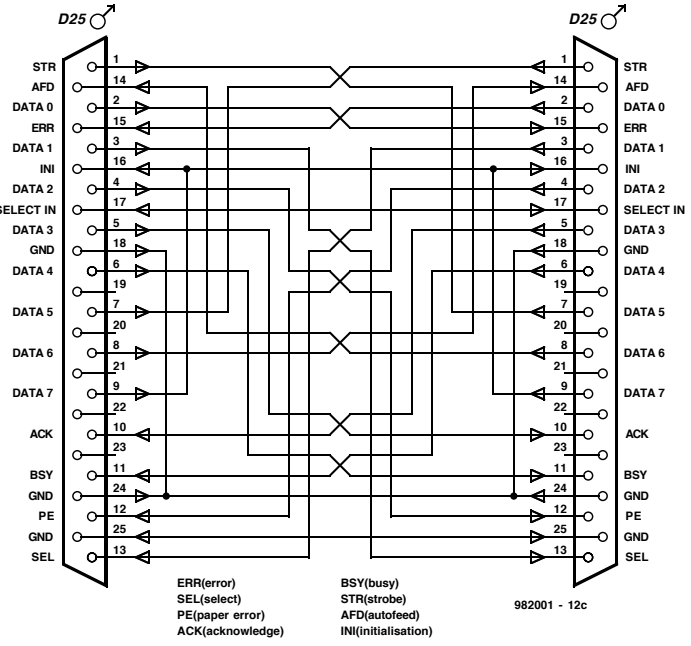


Figure 3. A special cable is required for bidirectional parallel ports (ECP). Don't forget that this cable must never be used with standard parallel ports!

C

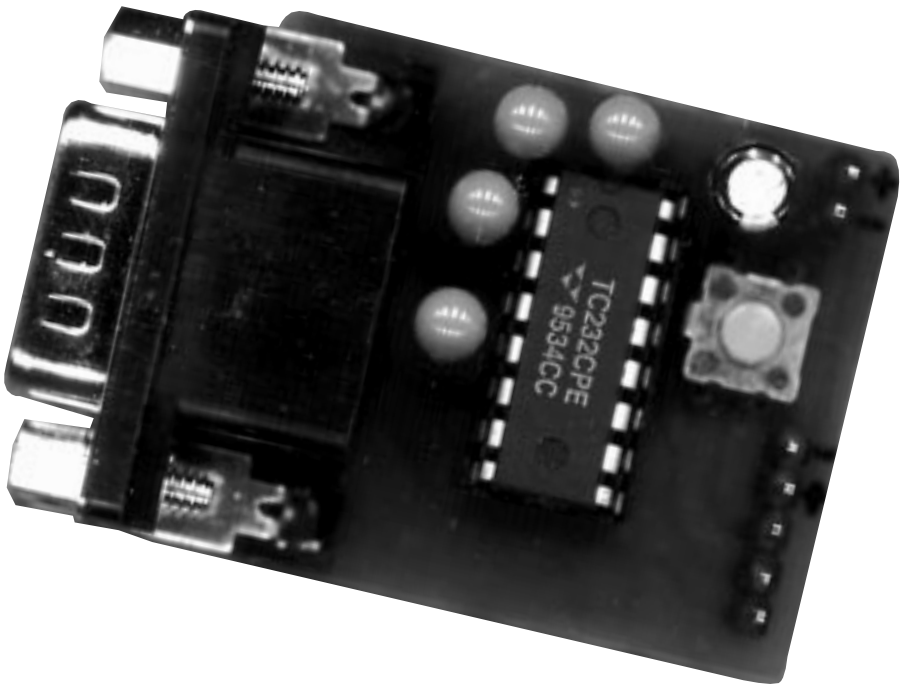


The Motorola 68HC11 microcontroller is packed with features and consequently requires very few parts to make a minimum system configuration. In principle, such a system does not need an RS232 interface because that is usually only necessary to be able to load programs into the controller's on-chip EEPROM memory. For the actual use of the program, the interface is not required. The RS232 interface described here is therefore built on a separate little board, and may be used at any time for programming and troubleshooting 68HC11 controller applications.

Design by J. Dietrich

RS232 interface for 68HC11

with a program loader for small systems



An 68HC11 Processor Board was published in the April 1994 issue of *Elektor Electronics*. The 68HC11 is supplied in a 52-pin PLCC case. It features up to 40 freely programmable input/output lines distributed across five ports, an EEPROM memory with a size of up to 2 kBytes, and a small RAM memory. Because the EEPROM is used as program memory, there is no need to burn EPROMs. The serial interface available on the chip allows ports, RAM or EEPROM cells to be read and written.

As illustrated in **Figure 1**, a minimum system using the 68HC11 requires only eight additional components. Even this small amount of external parts is sufficient to enable interrupt programs to be run and the eight A-D converter inputs to be employed.

The interface described here is built once only on a separate board, and migrated to future 68HC11 applications when and where necessary. It is only connected to the target system (here, the minimum system) for diagnosis and programming jobs.

Circuit and circuit board

The circuit diagram of the interface is given in **Figure 2**. Actually, it consists of just one IC, the unavoidable MAX232 RS232 driver/level converter which is capable of creating a fully RS232 compliant interface with symmetrical inputs and outputs, all on basis of a single 5-volt supply. Although the output line levels at pins 7 and 14 are at about ±9 V only, that is still within the RS232 specification.

The interface is connected to a PC using a cable with 9-pin sub-D sockets at both ends. Wires 2 and 3 in the cable should be crossed, all others travel pin-to-pin. The link to the target system (the 68HC11 processor board) is made via a 5-way cable connected to boxheader K2. This cable also carries the 5-V supply voltage for the interface. Connector K1 is only intended for situations where a 5-V supply is connected to the interface board, and the controller board is powered by the inter-

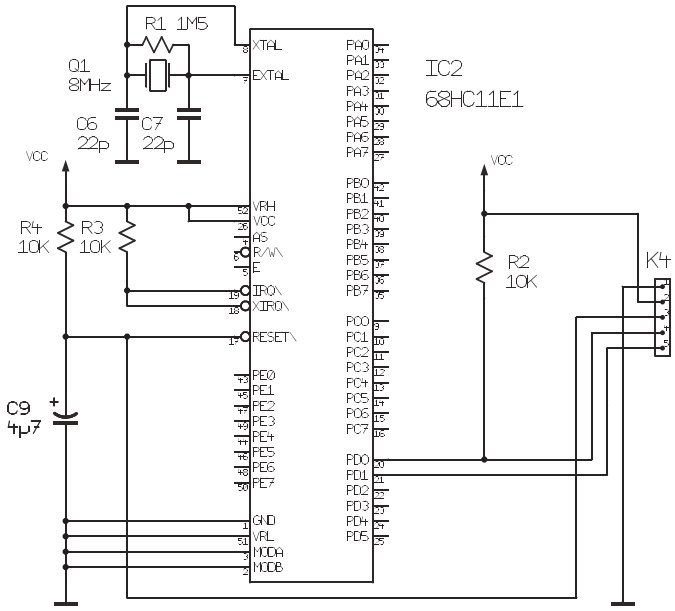


Figure 1. Apart from the microcontroller, a minimum system based on the 68HC11 requires only eight components.

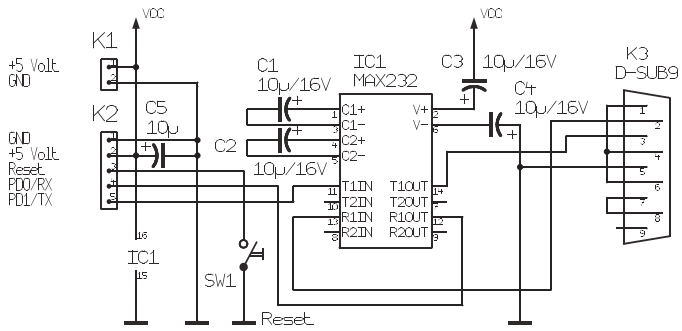


Figure 2. The RS232 interface is only built once, separate from the microcontroller system, and may then be used with all future experiments and applications of the 68HC11. In general, it is only connected-up for program development or debugging work.

face, via K2 and the 5-wire cable.
A push-button on the interface board allows the microcontroller to be reset. Without the interface board, the controller uses its own, internal, reset logic which does not require a push-button.
The artwork for the single-sided printed circuit board is given in **Figure 3**. In view of the small number of parts, a components list is not given. As usual, watch the polarity of the vertically mounted electrolytic capacitors before soldering them on to the board. Capacitor C5 decouples the supply voltage, and is safely rated at 10 V.

Software

The software you will need to get going with the 68HC11 processor board and the present interface is available free of charge from Motorola. The author used a packed file called ELEKT494.ZIP (66,304 bytes) which was downloaded from the Motorola BBS in Munich, Germany, telephone (+49) 89 92103111. This file may be found in subdirectory /mc68hcxx/m68hc11. The zip file contains, among others, an assembler with documentation, as well as programs for checking (MINIBUG) and programming EEPROMs (EEPROGIX). The list with short descriptions of all files held on the mailbox is called ALLFILES.BBS, and may be found in the directory /info.
How is the software transferred to the microcontroller? For test purposes, the software may be moved to the RAM area starting at address0. This is done by means of the ORG (originate) statement. 68HC11 versions are available which, like the -E1, have an equal amount of RAM and EEPROM. For program testing in RAM, enter the following:

1. Produce the assembler file using a simple word processor (pure ASCII file with the extension .ASC). Example: TEST.ASC.
2. Launch the assembler
   ASMHC11 TEST.ASC;b=256 (for CPUs with 256 bytes of RAM)
   ASMHC11 TEST.ASC;b=512 (for CPUs with 512 bytes of RAM)
   - Three files are generated with the following extensions:
   TEST.S19 for EEPROM programming
   TEST.LST ASCII documentation file
   TEST.BOO for direct copying into the RAM memory
3. Configure the PC's serial port in a DOS window (example: COM2)
   MODE COM2:1200,N,8,1
4. Copy the program into the 68HC11 RAM
   COPY TEST.BOO/B COM2:
   Once received the program is immediately executed.

Programming the on-chip EEPROM also requires a modification to EEPROGIX, which comes with the Elekt494 file:

1. Modify EEPROGIX as follows:
   ...
   LDS #$FF
   LDX #$1000 offset for control regs
   CLR $35,X !! insert this line!!
   CLR SCCR1,X initialize SCI for 8 data bits, 9600 bd
   LDD #$30
   ...
2. Launch assembler by typing ASMHC11 EEPROGIX.ASC;B=256

Next, program the EEPROM:

1. Produce the program using Editor, then assemble it.
2. Configure the PC serial port in DOS window
   MODE COM2:1200,N,8,1
3. Copy the once modified program into RAM
   COPY EEPROGIX.BOO/B COM2:
4. Close the DOS window, and launch HyperTerminal (Windows95)
   select Direct Cable Connection on COM2
   configure as 9600 bits/s, 8 bits, no parity, 1 stop bit, hard ware handshaking protocol
   Do File → Settings → ASCII Configuration, and set a character delay of 5, click on OK to leave the menu.
5. Type an upper-case I in the terminal window
   (selects internal programming of the

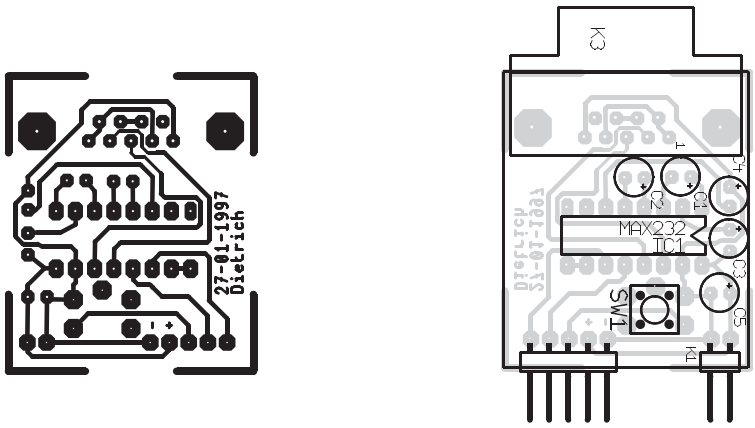


Figure 3. Track layout and component mounting plan of the small interface board.

EEPROM)

6. Transmit (upload) the relevant S19 file using Transmit Text File.
   Example: TEST.S19.

To be able to start the EEPROM resident program without the interface connected to the HC11 board, fit a jumper on contacts 4,5 of header K4. The program is then automatically launched when the supply voltage is switched on. Like EEPROGIX, the previously mentioned program MiniBug is transferred into the controller RAM using the COPY command and a bit rate of 1200/s. Next, the serial connection to the board may be established using HyperTerminal and a bit rate of 9600/s.

Finally

As already hinted at, the 68HC11 comes in three basic flavours:

MC68HC11A1	(256 bytes RAM, 512 bytes EEPROM)
MC68HC11E1	(512 bytes RAM, 512 bytes EEPROM)
MC68HC11E2	(256 bytes RAM, 2048 bytes EEPROM)

The MC68HC11 is also available from second source Toshiba under the type designation TMP68HC11E1T.
The modifications to the EEPROM programmer software (EEPROGIX) are only necessary if you use the -E versions, because only these have a special protection byte for the EEPROM. An important point to keep in mind for your own applications and experiments is that this byte **must** be erased before any writing is done to the EEPROM in an -E version.

(972019)

Following the introduction of the USB (Universal Serial Bus) and in its wake FireWire (IEEE 1394), almost all hardware and software vendors in the computer world have ranged themselves behind the serial-bus concept. USB has thus become *the* standard for connecting peripheral devices to the PC. Thanks to this development, there is finally a single interface standard which allows all computer devices to be connected to each other. In the meantime, manufacturers of consumer electronics have also embraced the new FireWire standard.

USB and FireWire

connecting all digital devices via a single cable

The problem is a familiar one: you purchase a new piece of equipment for the PC, search for matching cables, get all tangled up in the nest of cables behind the PC, and then have to look for a suitable driver. After that the correct interrupt and DMA channels must be assigned. Add to this the fact that every device needs its own mains power connection, and the chaos is complete. Even the computer manufacturers themselves find this all a bit too much.

The Universal Serial Bus shows that a better, and above all more user-friendly approach is possible. The USB offers a completely integrated 'Plug & Play' solution for all devices. For the user this means that any given device can be connected to the computer at any arbitrary time (switching off the power is not even necessary!). The operating system scans the USB every few seconds and responds appropriately whenever a new device is detected. Memory is automatically allocated and a suitable interrupt is assigned. A maximum of 127 devices can be connected to the bus, and smaller devices can draw their operating power directly via the bus interface.

In the meantime, Microsoft has announced that Windows98 will fully support the USB protocol. There are even rumours that a new OEM version of Windows95 will appear with USB support.

There are two different categories of connection available within USB: one with a maximum data rate of 1.5 Mbit/s and the other with a maximum data rate of 12 Mbit/s. The slower variant relates to relatively slow devices such as mice, barcode readers, card readers and keyboards. The faster variant is intended for data communication with video recorders and audio systems, as well as for use with high-performance peripheral devices such as hard-disk drives.

Cabling

The USB uses a daisy-chain cabling technique. This means that the devices are connected to the cable in sequence, one after the other. Thus in theory the computer need have only one USB interface to allow up to 127 peripheral devices to be connected.

Of course, the USB has its limitations. The maximum extent of the cable is 5 metres. Longer distances can be achieved by using *hubs*, which are buffer/splitter devices. A hub can be

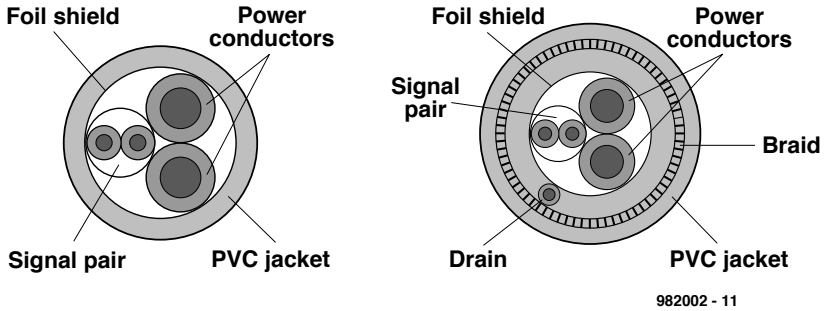


Figure 1. The construction of a standard USB cable and of a version with extra screening.

incorporated in a peripheral device, but it can also be included in a link as a stand-alone buffer and/or splitter. If a device includes a hub, then it is easily possible to incorporate two or more functions in a single enclosure. For example, one could imagine a keyboard with a built-in card reader, touchpad and/or mouse. A USB link can be extended a maximum of seven times, which means that its a total length can reach up to roughly 40 metres.

The USB link uses a 4-lead cable (see **Figure 1**). The two thinner wires are used for data communication, while the two thicker wires provide power to the connected device interfaces. A special, screened version of the cable is available for critical applications. The power leads can supply a total current of up to 500 mA at 5 V. This means that small devices, such as modems and card readers, no longer need to have their own, separate power supplies. The connection to the cable is made via a compact 4-way plug, which measures only 12 mm × 4.5 mm.

FireWire: a step further

Applications for the USB are limited to data streams with a maximum rate of 12 Mbit/s. For many applications related to the PC this is more than adequate. However, there are conceivable applications for which significantly higher data rates are desirable. One example would be the loss-free (uncompressed) distribution of digital television signals. For such applications, a new communication standard has been developed: IEEE 1394, commonly known as FireWire. FireWire is platform-independent, and its performance far outstrips that of other existing interfaces, including even the most advanced SCSI versions. The most important characteristics of the FireWire interface are low cost, ease of use thanks to a compact connector, complete 'Plug & Play' capability and especially high performance. The currently-used I/O interfaces (Centronics, RS232, Ethernet, SCSI and so forth) can thus be replaced by a better system, and communications between the computer and a wide variety of peripheral devices, or directly between individual devices, can be considerably simplified. In terms of performance, FireWire lies between USB (maximum 12 Mbit/s) and fibre-optic links (1 Gbit/s). With this level of performance FireWire could displace existing networking standards such as Ethernet (10 Base-T), but in practice FireWire is presently too expensive for this. In addition, Ethernet has emerged from the computer world, while FireWire is clearly destined to play a dominant role in the arena of modern consumer electronics products for digital communications. The configuration shown in **Figure 2** clearly illustrates the mutually complementary nature of USB and FireWire and demonstrates how they can be used to extend each other. Each has its own particular region of application. In the illustrated configuration, USB is used for devices directly connected to the PC, while FireWire forms the digital link between a D-VHS recorder, a set-top box and a digital television.

The performance level of FireWire presently lies at around 100 Mbit/s, with speeds of 200 Mbit/s and 400 Mbit/s approaching feasibility. Just as with USB, a simple connector is used which can be attached with a minimum of effort.

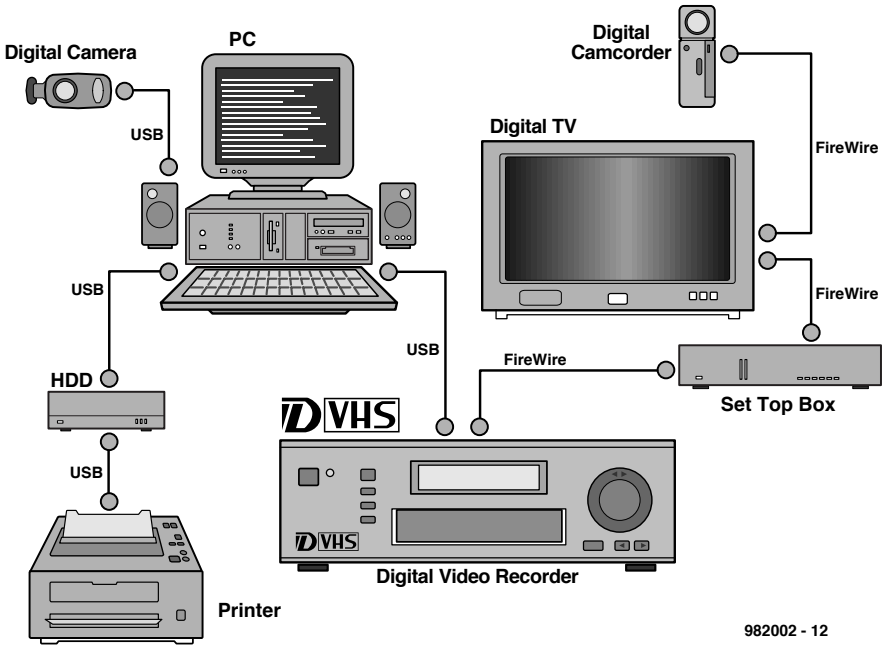


Figure 2. A configuration in which both USB and FireWire play a role.

The protocol

In a significant departure from existing interfaces, FireWire allows for *isochronous* data transmission in addition to the existing *asynchronous* data transmission (what this all means is explained in the following text). In addition, the interface specification allows FireWire to be used in both cable and backplane environments (a backplane provides the internal connections between the various components of a computer). The protocol is thus usable both inside and outside of the computer enclosure. Depending on the specific electronic components used, data rates of 25 to 50 Mbit/s are possible on the backplane, and as much as 400 Mbit/s is possible with cable. A maximum of 16 hubs may be included in the cable system, and the separation between any two hubs can be a maximum of 4.5 metres. The maximum total extent of the cable is thus 72 metres.

Figure 3 illustrates a configuration in which two computers and a number of I/O devices communicate with each other via the IEEE-1394 bus. Note that the serial bus is also used on the backplane.

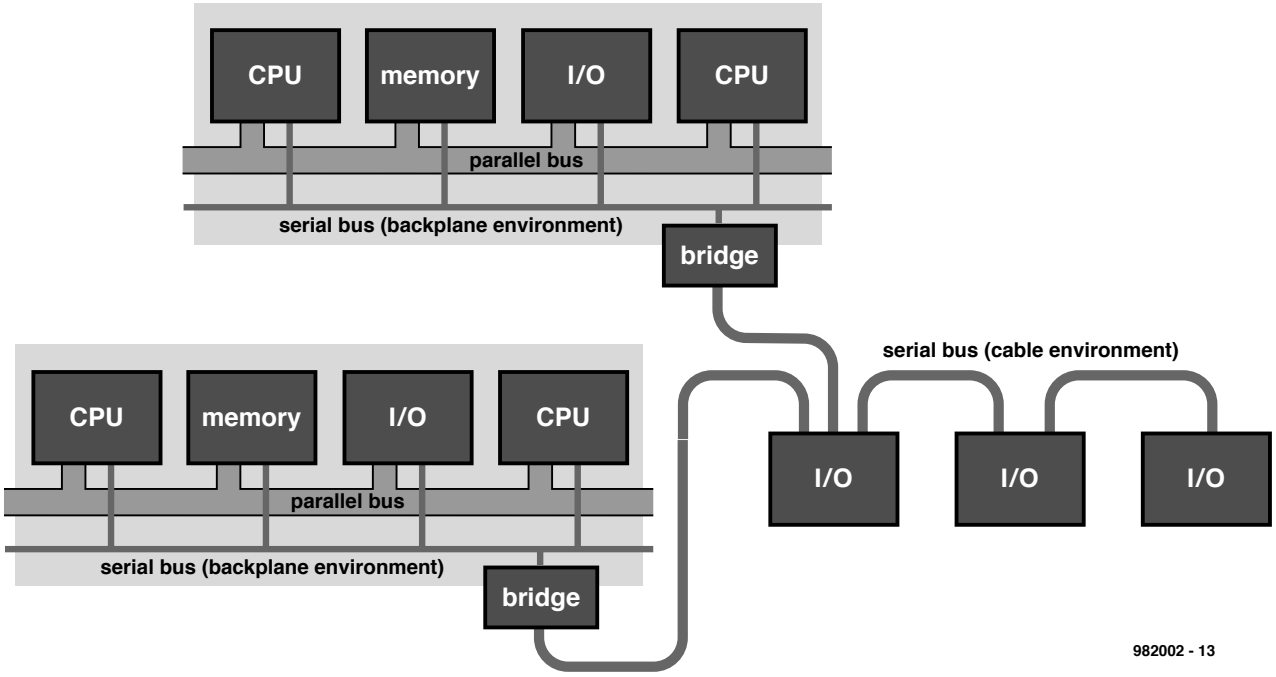


Figure 3. Two computers communicate with each other and with peripheral devices via FireWire. Note that FireWire is used for both the cable connection and the backplane.

Since FireWire is based on high data rates, it places unusually severe demands on the cabling material. While USB works with a 4-lead cable, FireWire uses a 6-lead cable. This is illustrated in **Figure 4**. Two leads are reserved for distributing electrical power. The allowed dc voltage may lie between 8 V and 40 V, with a maximum total current of 1.5 A. The signal lines are implemented as two individually-screened twisted pairs.

The model

The specification of the FireWire protocol is based on three *layers*: the Transaction Layer, the Link Layer and the Physical Layer. These are depicted in **Figure 5.**

Transaction Layer

The Transaction Layer manages data transfers between two devices via the Serial Bus. The system recognizes three types of transactions: *read* (data is transmitted from a device to the main system), *write* (data is transmitted from the main system to a device) and *lock* (data is transmitted from a device to the main system, which in turn sends the processed data back to the device). The bus supports the IEEE 1212 standard, which uses 64-bit addressing. The topmost 16 bits of the address are treated as an identification code (node ID) within each device. The 16 bits of the node ID are further divided into a 10-bit bus ID and a 6-bit offset ID. Since the highest possible address (all one's) is reserved for special applications, an actual system configuration can have up to 1023 buses, each of which can have up to 63 independent device connections (nodes).

Link Layer

The Link Layer looks after delivering information packets according to a half-duplex protocol. Each individual packet is sent via a process called a *subaction*. Two types of subaction are possible:

- *asynchronous subaction*, in which an arbitrary amount of data plus some Transaction Layer information is sent to a specific node (device address), following which a confirmation (acknowledgement) is returned from the destination device;
- *isochronous subaction*, in which a variable amount of data is sent at regular intervals, with simplified addressing and without confirmation from the destination device.

Each subaction can have up to three distinct phases:

- *arbitration sequence*: a device which wants to transmit data sends a bus access request to the Physical Layer. If the device already controls the bus as the result of a just-completed subaction, it receives immediate access to the bus.

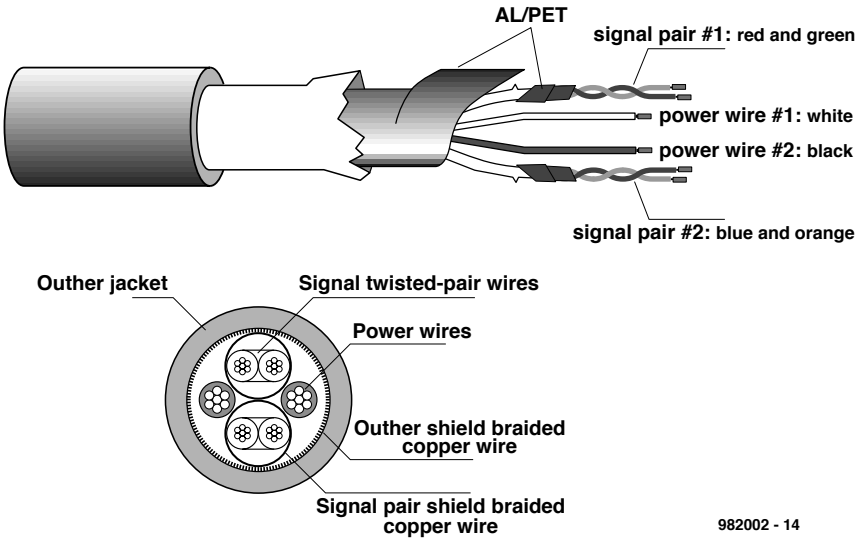


Figure 4. The construction of a cable which is suitable for FireWire.

- *data packet transmission*: the source device transmits a data packet containing coded speed, format and transaction information, the addresses of the source and destination devices and the data. Isochronous packets contain a short channel identification code (ID) in place of the addresses of the sender and receiver.
- *acknowledgement*: a uniquely-addressed destination device will return a confirmation code which indicates that the data was properly received and contains information regarding the action taken on receipt of the data. Isochronous subactions and asynchronous broadcast subactions do not require acknowledgement.

All asynchronous subactions are normally separated from each other by short intervals during which the bus is in an 'idle' state; these are called *subaction gaps*. An additional gap occurs in the interval between the completion of the data packet transmission and the receipt of the acknowledgement. The length of this gap depends on the physical system configuration. **Figure 6a** depicts the organization and timing of asynchronous subactions. Gaps also occur between isochronous transmissions; these are called *isoc gaps* and are depicted in **Figure 6b**.

Physical Layer

The Physical Layer has three functions. First, it translates the logical levels of the Link Layer into electrical signals for the cable, and in the reverse direction it translates signals from the cable back into proper logical levels for the Link Layer. In addition, it determines the actual electrical and mechanical configuration of the connection. Finally, the Physical Layer acts as a sort of referee which grants devices access the bus when they wish to send data.

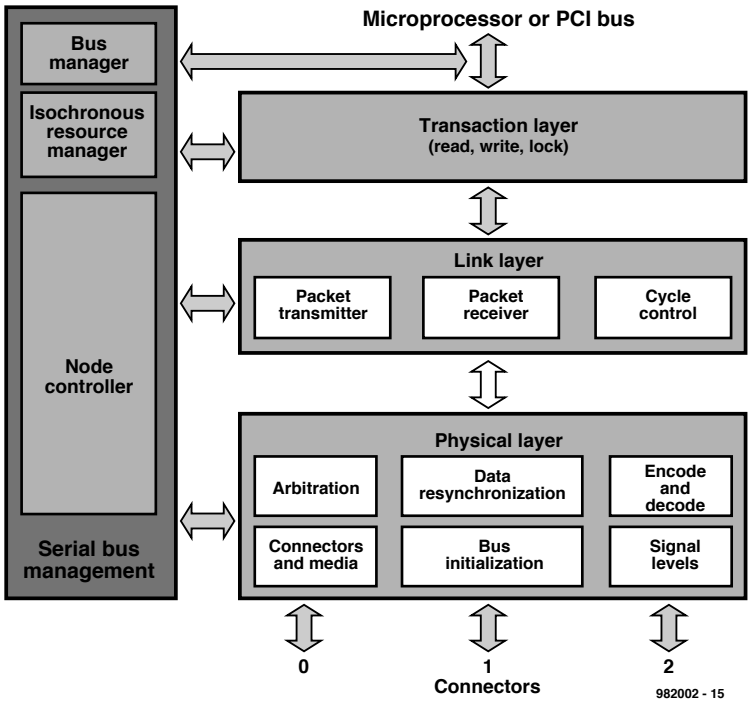


Figure 5. The FireWire protocol has three layers which communicate with each other.

The future is what counts

USB and FireWire have a lot to offer. Although the first equipment to use these interfaces (both computers and peripheral devices) is already appearing on the market, it's still too early to proclaim that the serial bus interfaces will be a great success. A few years will be necessary before this can be judged. Readers who wish to follow further developments in this area can keep a good eye on them via the Internet. Information about USB can be found at *http://www.usb.org*, and FireWire has its own site at *http://www.firewire.org*.

(982002)

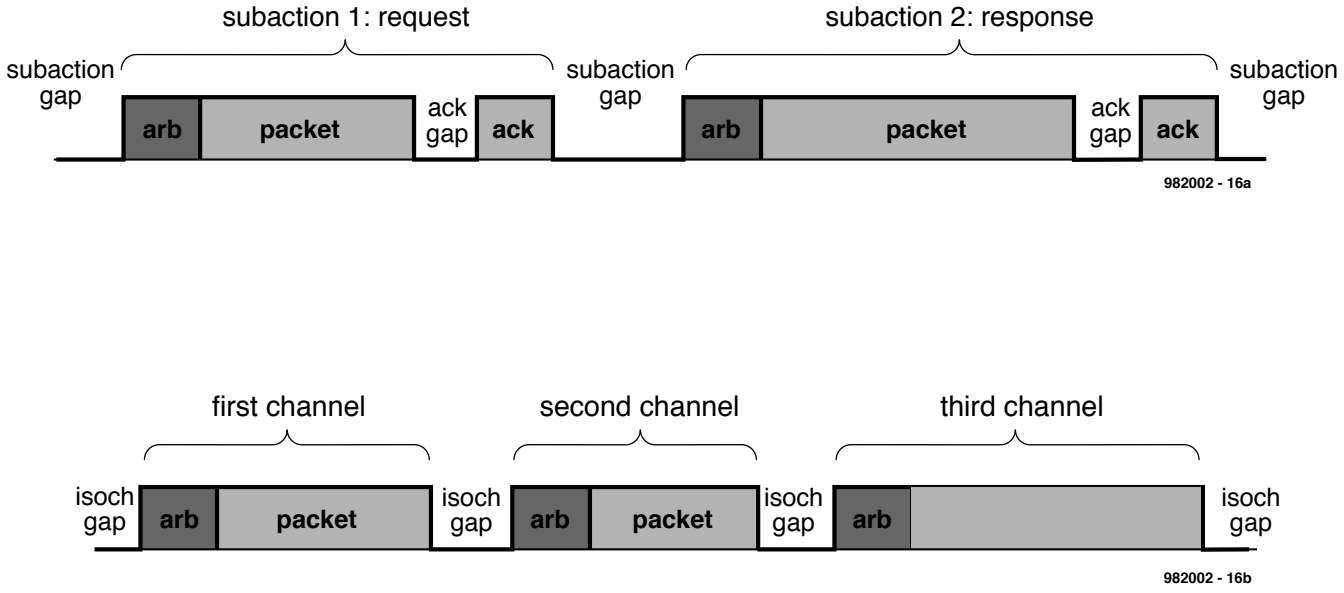


Figure 6. This timing diagram illustrates the difference between data transmissions using asynchronous subactions (a) and isochronous subactions (b).

We have reason to believe that this is a unique and novel circuit. It measures ambient light intensity by means of the serial port on your PC, without any kind of external power supply. A Visual BASIC demonstration program is available for the circuit. This program is easily modified to meet personal requirements.

Design by B. Oehlerking

Light intensity measurement with a PC

no external power supply required

COMPONENTS LIST

Resistors:
R1,R2,R3 = 10kΩ
R4 = 12kΩ

Capacitor:
C1 = 1μF MKT

Semiconductors:
D1,D3 = 1N4148
D2 = low-current LED, red
D4 = low-current LED, green
T1 = BC557C
T2 = BC547C
IC1 = TLC7555

Miscellaneous:
Small encapsulated solar cell, 0.45 V, e.g. Conrad 198030
9-pin sub-D socket, IDC type
10-way boxheader
10-way IDC socket
2-3 m of 10-way flatcable

Figure 1. Circuit diagram of the light intensity meter. Note that the sensor is a small solar cell.

The operating principle of the circuit shown in **Figure 1** is the time it takes for a capacitor to be charged to a certain voltage (threshold level). The capacitor, C1, is supplied by a constant-current source consisting of D2, R2, R1, T1 and a solar cell acting as the sensor. The result is a capacitor charging voltage which rises linearly. The CMOS timer IC type 7555 (whose internal diagram is shown in **Figure 2**) operates as a comparator, comparing the voltages between pins 2/6 with the supply voltage between pins 4/8. If the voltage between pins 2 and 6 reaches about 2/3 of the supply voltage, the chip output changes from high to low (active).

The crux of the circuit is that it does not require an external power supply. Its supply voltage is 'stolen' from the PC's RS232 port, GND providing the ground level, while the positive supply level is created with the aid of the DTR (data terminal ready) line. The RTS (request to send) line is used to charge the capacitor as the measurement signal travels to the PC via the CTS (clear to send) line. The length of the time interval between discharging and

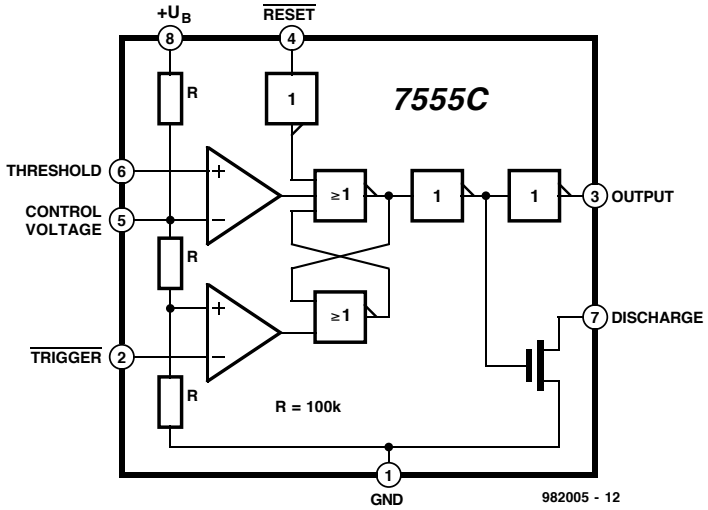


Figure 2. Internal structure of the TLC7555 CMOS timer used in the circuit.

Table 1.

DTR	RTS	D4	D2	CTS
off = –10 V, on = + 10 V		red	green	off = 0 V, on = + 10 V
off	off	off	off	off
on	off	off	on	off
off	on	on	off	off
on	on	off	on	on

charging is inversely proportional to the measured light intensity.

Figure 3 shows the flowchart of a suggested program which measures and evaluates the time interval. Admittedly, a software loop is not an ideal solution since it makes the program dependent on the speed of the computer used. Unfortunately, Visual BASIC offers no alternative when it comes to time measurement at reasonable accuracy. The program you find on the disk with order number **986002-1** (see Readers Services page) is only intended as an example to help you write your own software for the interface.

Because most RS232 interface lines are at ±10 V, diode D1 is needed as a polarity reversal protection. Diode D3 prevents a too negative potential at the base of T2.

Construction of the circuit should be straightforward. The few parts that go into the circuit are easily fitted on a small piece of veroboard or stripboard. The solar cell mentioned in the parts list (450 mV) has two screw connections at the back, which double as mechanical fasteners and electrical contacts. The interface is connected to the PC by way of 2 to 3 metres long 10-way flatcable (of which only four wires are used), which is connected to the circuit using an IDC (clamp-on) connector, and to the PC's RS232 port via a 9-way sub-D socket.

The program offering a test mode in which DTR and RTS may be controlled separately, the hardware may be tested with relatively simple means. The test program also reads the status of the CTS line. The consequences of the different levels of DTR and RTS for the LEDs and the CTS line are listed in **Table 1**.

(982005-1)

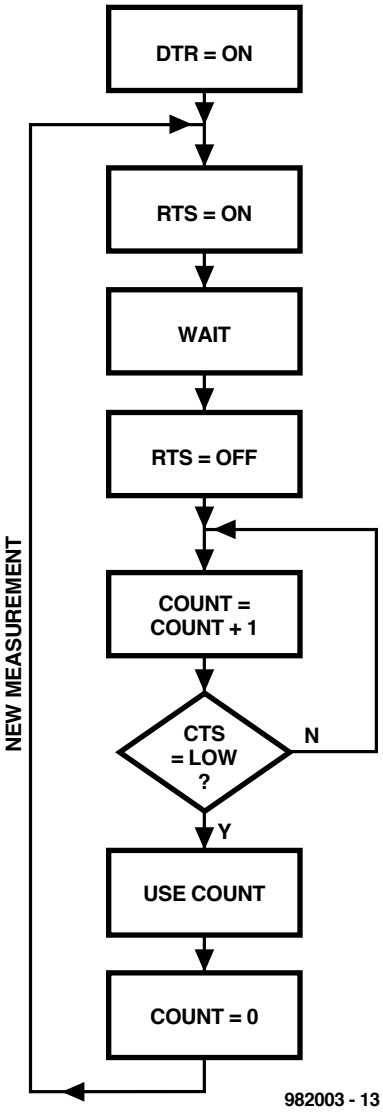
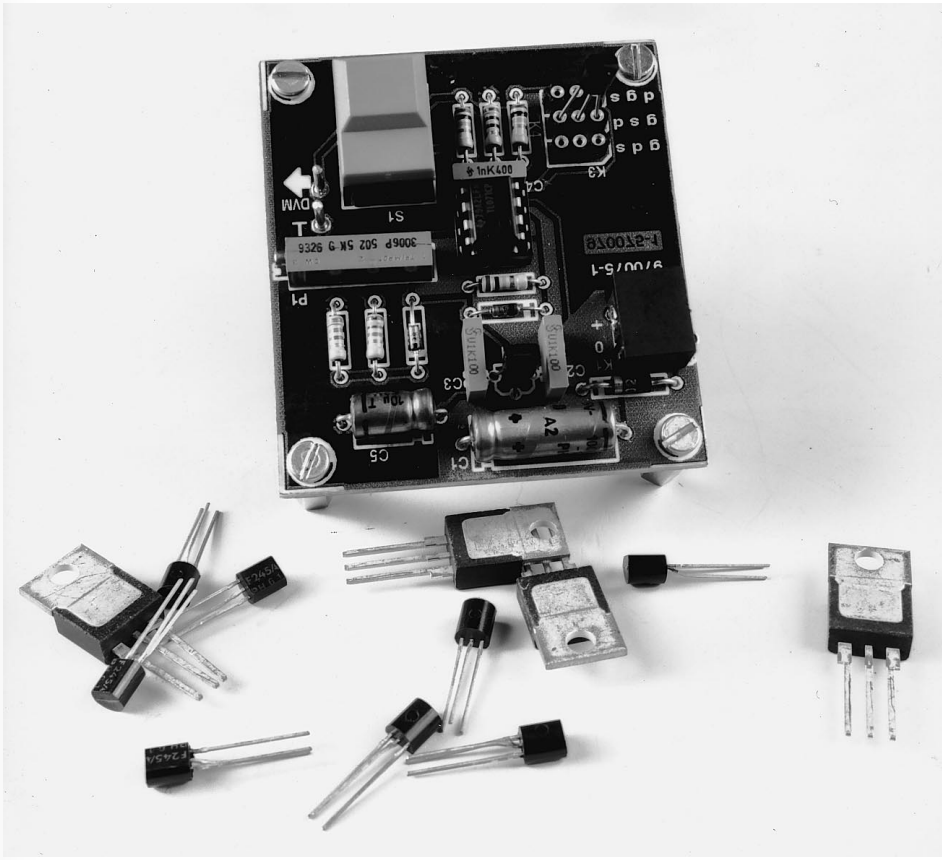


Figure 3. Structure of the demo and test program written in Visual BASIC. The full program is available on disk.

JFET tester

FET matching and testing made simple

Many designers shy away from using JFET transistors although numerous circuits would benefit from using these devices. Whenever very high input impedance, high operating frequency and relatively low noise are paramount design requirements, a JFET may be a good choice. This article aims at removing some of the general hesitation about using JFETs by describing a simple tester that allows you to measure two crucial electrical parameters of these devices.



Design by M. Frankowski

Although the staggering amount of data you can obtain from a FET's datasheet may cause the odd apoplectic fit with some of you, it can be upheld that there are two absolutely vital selection criteria (or 'electrical parameters') which help to identify an unknown JFET, or find so-called *matched devices* from a batch of JFETs (more about this further on).

The present tester is only suitable for *n-channel* small-signal JFETs (junction field-effect transistors). This does not detract from its usefulness however because chances are pretty small that you will ever encounter a p-channel JFET in your lifetime!

MEETING THE JFET

Although most of the background theory relevant to JFETs may be found in any reasonable electronics textbook, it may still be useful to present a brief recap in this article.

The circuit symbol of the n-channel JFET is shown in the basic connection diagram in **Figure 1**. The in-going arrow at the G (gate) denotes the n-channel version. Although the symbol of a JFET is different from that of a bipolar transistor, it may help you think of the D (drain) electrode as the collector, the S (source) electrode as the emitter, and the G (gate) electrode as the base. Unlike an n-p-n transistor, however, the operating range of the parameter called 'gate-source voltage' (V_{GS}) is negative. In other words, the gate is made negative with respect to the source.

Now, let's first take a moment to explain the notation of the various parameters symbols you will find in the datasheets and this article. The above example, V_{GS}, should be read as follows: *Voltage* (V) between *Gate* and *Source* (GS). Note that the *index* 'GS' is printed smaller and lower than the capital letter V. Similarly, the symbol I_D means 'current (I) in drain (D) channel', and V_{DS} means 'voltage (V) between drain (D) and source (S)'. Once you are comfortable with the basics of this notation system, you will have little difficulty in unravelling the meaning of the various symbols used

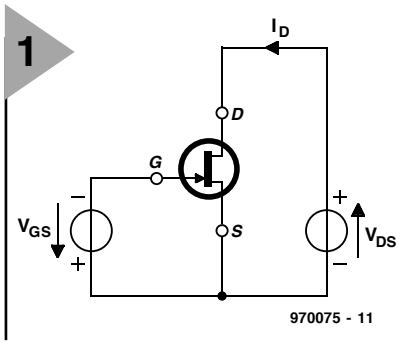


Figure 1. Basic JFET connection circuit.

to describe the electrical parameters of JFETs (and other transistors).

PINCH-OFF VOLTAGE

The first vital JFET parameter is called the *pinch-off voltage*, symbol $-V_{GS(p)}$ or simply $V_{(p)}$. Unfortunately, due to fabrication techniques, this parameter is subject to relatively large tolerance. In other words, the actual $-V_{GS(p)}$ spec of the JFET you have available may differ considerably from the value specified in the datasheets. The pinch-off voltage is the gate-source voltage at which negligible drain current flows. Hence the name: this voltage pinches off the current flow in the drain-source channel to virtually nought. The remaining current is due to leakage, and usually defined as 1 nA, 10 nA or even 10 μA by the manufacturer. The pinch-off voltage is stated assuming that V_{DS} is held constant. Here, too, opinions differ: some manufacturers state $-V_{GS(p)}$ at V_{DS} = 10 V, others at V_{DS} = 15 V. It makes little difference, however, at long as you know that V_{DS} is a constant value, or nearly so. The JFET tester described here measures $-V_{GS}$ at I_{DS} = 10 nA, and does not use a constant voltage for V_{DS}. Yes, it can be done! Referring back to the graph in Figure 2, you can see that the I_D curve for values of $-V_{GS}$ approaching the $-V_{GS(p)}$ value (like $-V_{GS}$ = 4 V) runs virtually straight from V_{DS} = 2 V onwards. In other words, I_D remains virtually constant as long as V_{DS} is between, say, 3 V and 15 V. So, the error caused by the non-constant value of V_{DS} in the test circuit is negligible, because V_{DS} is always in the range where I_D is virtually constant. That's why JFETs make great constant-current sources!

DRAIN SATURATION CURRENT

The second important JFET parameter is the value of the current through the drain-source channel when $-V_{GS}$ is at 0 V, and V_{DS}, at 15 V (usually!). When these two conditions are satisfied, the drain current will remain practically constant at a certain maximum value. In other words, the drain-source channel is *saturated*; it will not pass more current. The symbol used for the drain saturation current is $I_{D(ss)}$. Like I_D, $I_{D(ss)}$ derates (worsens) with increasing temperature, as illustrated in **Figure 3**.

PRACTICAL CIRCUIT

Admittedly, that was rather a lot of theory to wade through. High time to discuss how the two measurements mentioned above are performed in practice. Let's look at the circuit diagram in **Figure 4**.

Pinch-off voltage measurement (S1 not pressed)

This measurement is performed when S1 is not actuated. The pinch-off voltage is indicated on the DVM (digital voltmeter) connected to the output of the tester. Imagine a JFET is inserted in the D.U.T. (device under test) sockets. A reference voltage of 100 mV is created with R1-D2-P1-R2, and connected to the non-inverting input of opamp IC2. The inverting input is connected to the positive supply rail by a 10-MΩ resistor (R3), as well as to the FET under test, which is included in the feedback path to the opamp output. Since the gate of the JFET is at ground potential, the voltage at the source is controlled to make the gate more negative than the source ($-V_{GS}$). Because the opamp will attempt to make the voltages at its inputs equal, it supplies a gain at which the voltage across R3 will equal 100 mV. Resistor R5 then drops 10 nA × 100 kΩ = 1 mV. In this way, $-V_{DS}$ of the JFET is controlled until I_{DS} equals 100 mV/10 MΩ = 10 nA. The result is that $-V_{GS(p)}$ appears on the DVM. The error sources in this measurement are I_{D3} (10 pA), the input bias current of the TL071 opamp (<200 pA) and its input offset voltage (<10 mV).

Some JFETs have a pinch-off voltage between 0 and –2 V. To enable these devices to be tested also, a voltage regulator is used in the negative supply rail. By supplying the opamp with 15 V rather than 12 V (as the rest of the circuit), the range of V_{GS} is extended to about +2 V.

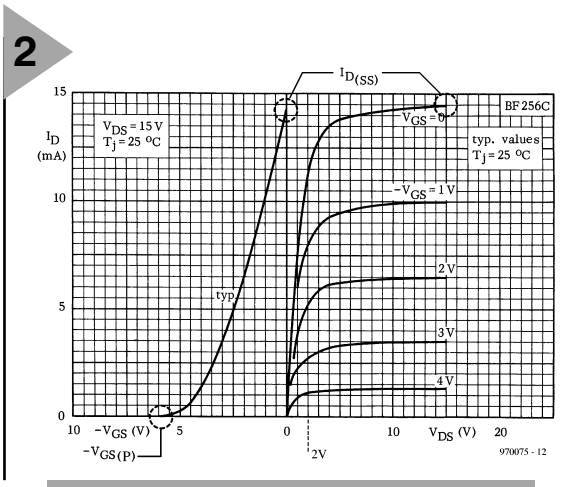


Figure 2. The pinch-off voltage $-V_{GS(P)}$ and the drain saturation current, I_D, are easily determined if you have a transfer characteristic graph like this available (example: BF256C JFET; source: Philips Components).

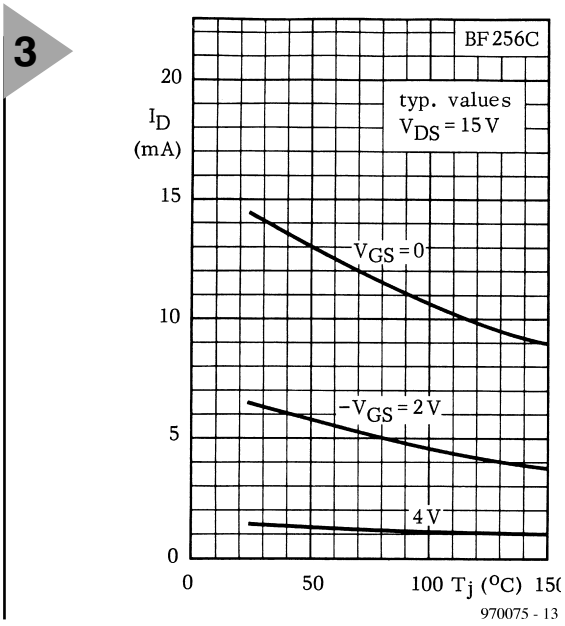


Figure 3. Drain current (I_D) derating as a function of junction temperature (T_j).

Drain saturation current measurement (S1 pressed)

This is a much simpler measurement.

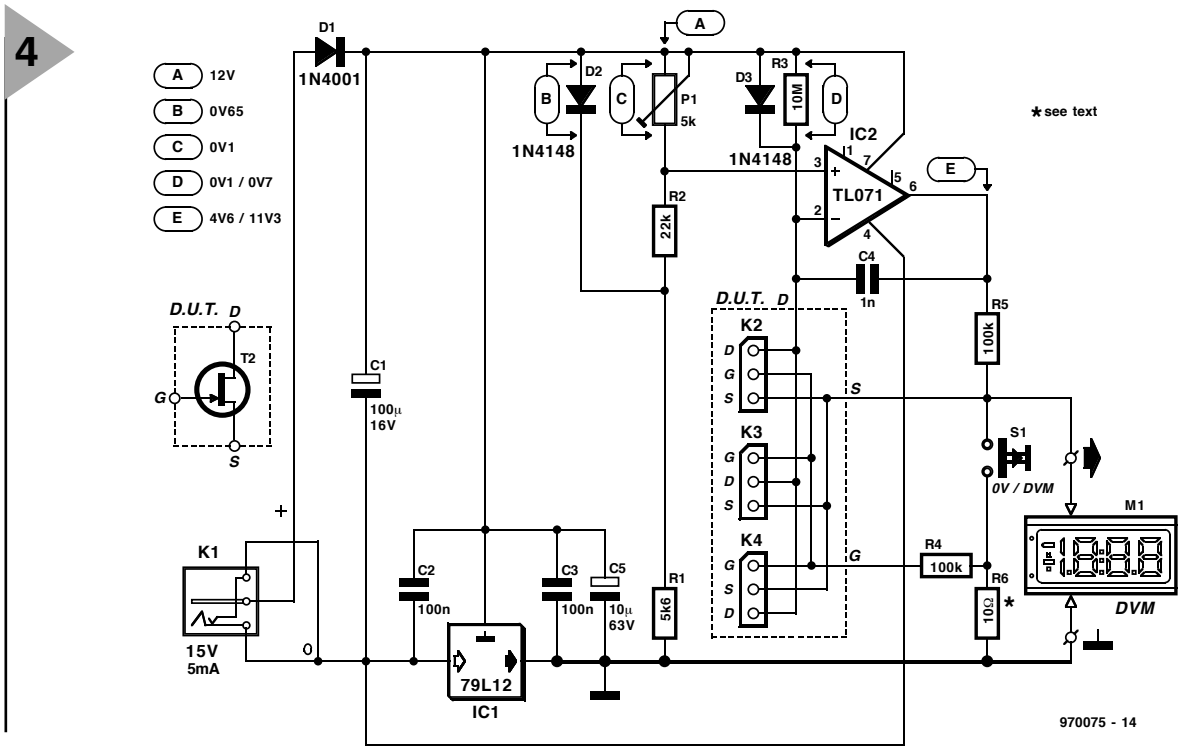


Figure 4. Circuit diagram of the JFET tester.

Pressing S1 causes the source to be connected to the gate, so that V_{GS} = 0 V. Although the reference voltage at the +input of IC2 remains 100 mV, D3 now drops its normal forward bias voltage of about 0.7 V. Consequently the opamp pulls its output to (practically) the positive supply level. Resistor R5 then carries (12−0.7) V/100 kΩ or a little more than 100 µA. This current also flows through R6, where it is added to the drain saturation current. As this will be in the mA region, the error is, we feel, insignificant for all practical intents and purposes.

Because R6 has a value of 10 Ω, the $I_{D(ss)}$ indication on the DVM is in (mA×10). Mind you, you are measuring a voltage that indicates a current! True, a value of 1 Ω would have been more logical because then the readout is simply in mA. A higher output voltage is used, however, to enable the DVM to be switched to a higher range with resultant higher accuracy (in general!). Owners of 4.5-digit DVMs may use a 1-Ω resistor in position R6. Whatever DVM you use, the accuracy of the tester will benefit from the use of a 1% (close-tolerance) resistor for R6.

The voltages indicated in the circuit are typical. The first voltage applies when S1 is not pressed, the second, when S1 is pressed. Unless otherwise indicated, measurements are with respect to the ground rail, i.e., the 0 V input of the DVM.

The circuit may be powered by an inexpensive mains adaptor with an output voltage of about 15 VDC. Because the tester will only draw a few tens of mA when S1 is pressed to perform a drain saturation current measurement,

COMPONENTS LIST

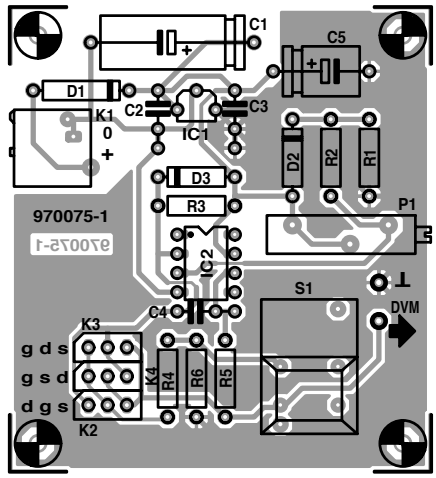
Resistors:
R1 = 5kΩ6
R2 = 22kΩ
R3 = 10MΩ
R4,R5 = 100kΩ
R6 = 10Ω (1% if necessary)
P1 = 5 kΩ 10-turn cermet preset, vertical

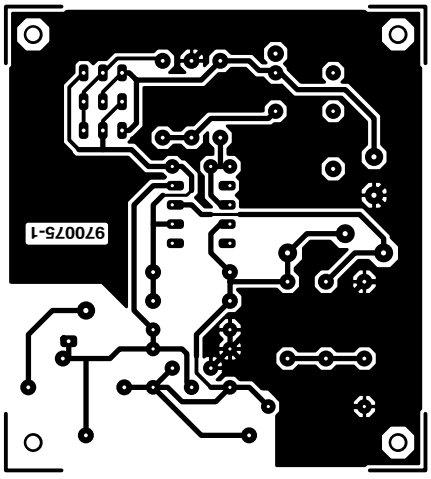
Capacitors:
C1 = 100µF 16V
C2,C3 = 100nF
C4 = 1nF
C5 = 10µF 63V

Semiconductors:
D1 = 1N4001
D2,D3 = 1N4148
T1 = device under test (JFET)
IC2 = TL071CP
IC1 = 79L12

Miscellaneous:
M1 = digital voltmeter (DVM).
K2K3,K4 = 3 rows of 3 IC socket pins, turned sockets.
S1 = Digitast press-key, 1 make contact (ITT-Schadow).
K1 = mains adaptor supply socket, flat model.

Figure 5. Copper track layout and component mounting plan of the PCB designed for the JFET tester (board not available ready-made).





even very low power adaptors may be used. Any small adapter rated at 12 VDC should be suitable because it will typically supply at least 15 V when only a few milli-amps are drawn. When S1 is not pressed, the current consumption is about 5 mA.

CONSTRUCTION AND ADJUSTMENT

A ready-made printed circuit board is, unfortunately, not available for this project. The artwork to make your own board is, however, shown in Figure 5. Look carefully at the component overlay to make sure each and every polarized part is fitted the right way around. There are three D.U.T. sockets on the board to accommodate different pin-outs of the JFETs you want to test. Make sure you know the pin-out from a datasheet, else neither measurement will make any sense at all, and you may destroy the device under test.

Preset P1 is adjusted until it drops exactly 100 mV (use a 10-MΩ DMM for this adjustment).

Finally, a suggested front panel layout is shown in Figure 6.

HINTS AND KINKS

If you want to do some testing on known JFETs, we recommend the BF245 and BF256 series to start with . Another commonly used JFET (particularly outside Europe) is the 2N5486.

The highest drain saturation current that can be measured by the circuit is about 40 mA. If you see anything above this value on your DVM display, the measurement is probably not valid.

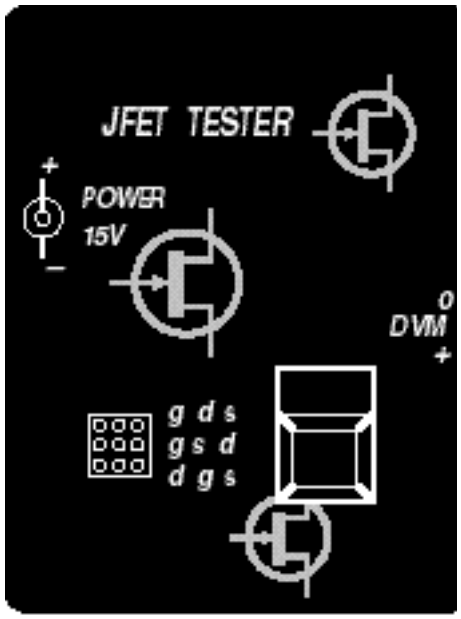
Take the internal resistance of your DMM into account when measuring the voltages indicated in the circuit; at some junctions, even 10 MΩ may be a relatively 'heavy' load.

MATCHING, WHAT'S IT FOR?

Some (audio) amplifier stages of the differential type use JFETs which have to be 'matched' for optimum performance. The two elementary tests carried out with the aid of the present tester should enable you to pick two JFETs with almost equal electrical properties from a batch of, say, ten.

(970075-1)

6



970075-F

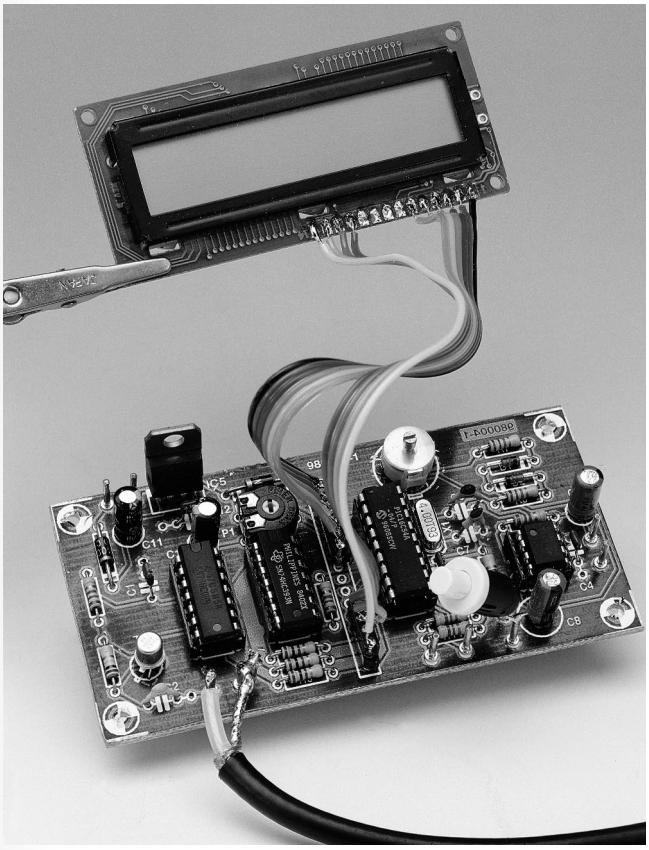
Figure 6. Suggested front panel layout for the JFET tester.

Frequency display and VFO stabilizer

eliminates frequency drift in home-brew and 'surplus' HF receivers

One of the most frequent topics for discussion among the many 'Home Brewers' on the amateur bands is the difficulty of building a VFO that is stable enough to be used on all of the HF bands. The relatively simple circuit described here will stabilize the frequency of an HF VFO, and provide a digital frequency display. The display will allow for the different frequency offsets required for USB and LSB, and the fact that the VFO frequency may be above the signal frequency on some bands, and below it on others.

By Eamon Skelton, EI9GQ



The circuit uses a microcontroller to count the frequency of an HF VFO (variable frequency oscillator), add or subtract the IF (intermediate frequency) offset, and display the signal frequency on a standard Hitachi intelligent LCD display module. The VFO frequency is stabilized by sending a correcting voltage to a varicap diode in the VFO. The circuit is based on a 'Huff & Puff' stabilizer that I have been using for several years.

The circuit design philosophy was to keep the hardware as simple as possible, and to use inexpensive and readily available components. This was made possible by using a microcontroller chip which was programmed to take care of most of the complex functions like counting, arithmetic, and driving the LCD display module. Two versions of the unit were built, one using the 8031 microcontroller, and this version using the PIC 16C54. I have tested the circuit at various frequencies between 8 MHz and 80 MHz.

CIRCUIT DESCRIPTION

The circuit diagram of the VFO stabilizer is shown in **Figure 1**. The circuit is basically a Frequency Locked Loop (FLL). As many of you will be aware, the PIC16C54 microcontroller is a complete computer on a chip with 512×12 bits of ROM, 32 bytes of RAM, a clock circuit, 12 I/O pins, a real time clock/counter, and several other useful features. For a full description of the PIC16C54 and its instruction set, see the Microchip data book or Microchip data sheet DS300151.

The RF signal picked up from the VFO is amplified and digitized by T1, then buffered by IC1b. The BSX20 is a fast switching transistor for use up to 200 MHz. The amplified signal is gated by IC1c and IC1d. A 100-ms gate pulse is generated by a software delay loop in the PIC. The square wave pulses from IC1d are counted by IC2 which is configured as an 8-bit counter.

The output of IC2b is connected to the RTCC input of the PIC through a 1.2kΩ resistor (R6). The PIC has a real-time clock/counter (RTCC) which can count pulses applied to the RTCC input (pin 3). The RTCC register is only eight bits wide giving a maximum count of 255. If the PIC's internal prescaler is set to divide by 256 the

maximum count is 65,535, effectively making a 16-bit counter. With a 1-ms gate time this would allow the counter to count up to 65.535 MHz but the resolution would be 1 kHz which is not good enough for our purposes.

The 74HC393 counter chip increases the count to 24 bits, or 16,777,215. With a 100-ms gate time this will allow a maximum count frequency of 167.77721 MHz and a resolution of 10 Hz, that is, if you can find logic chips that are fast enough. One problem with this arrangement is that it is not possible to read the least significant 8 bits directly from the 393 counter.

This problem is overcome by sending pulses to the counter input through gate IC1d. By counting the number of pulses it takes to make the counter overflow it is a simple matter to calculate the value in the '393 at the time the gate was closed. As the PIC internal prescaler can not be read directly, a similar method is used to calculate the value in the internal prescaler. Pulses are applied to the prescaler input by pin 13 of the PIC (RB7) until the prescaler overflows. The most significant 8 bits can be read directly from the RTCC register. This may seem like a strange way of reading the count but it is quite easy to implement in software and makes the circuit hardware very simple.

Now that we have the count result stored in the PIC, the IF offset must be added or subtracted; the result is the signal frequency in binary. This number is converted first to BCD, then to ASCII and finally sent to the LCD display module. A new count takes place about 9 times every second, the display is updated every second count, or just over 4 times a second. Updating the display more often than this causes the last digit to flicker, updating less often makes the display 'sluggish' when you tune quickly across the band. Before the first digit of the frequency is displayed a test is done to find out if it is a zero. If it is then a blank space is displayed instead, giving automatic leading zero suppression. Whether the IF offset is to be added or subtracted is determined by the state of the ADD/SUBTRACT (+/−) input, which is linked to an input pin on the PIC. Whether the offset is for USB (upper sideband) or LSB (lower sideband) is determined by the state of the OFFSET input pin. Suggested circuits for controlling the ADD/SUBTRACT (+/−) and OFFSET inputs are shown in **Figures 2 and 3** respectively. Mind you, these are just examples, the exact configuration of the switches depends on the requirements of your HF receiver.

So far the circuit is just acting as a frequency counter and display. At the end of each count/display cycle, the counters are reset and the cycle is repeated again. No attempt is made to control the frequency of the VFO. The control voltage at the output of the integrator is set at about 2.5 volts and remains there until the LOCK/UNLOCK button, S1, is pressed.

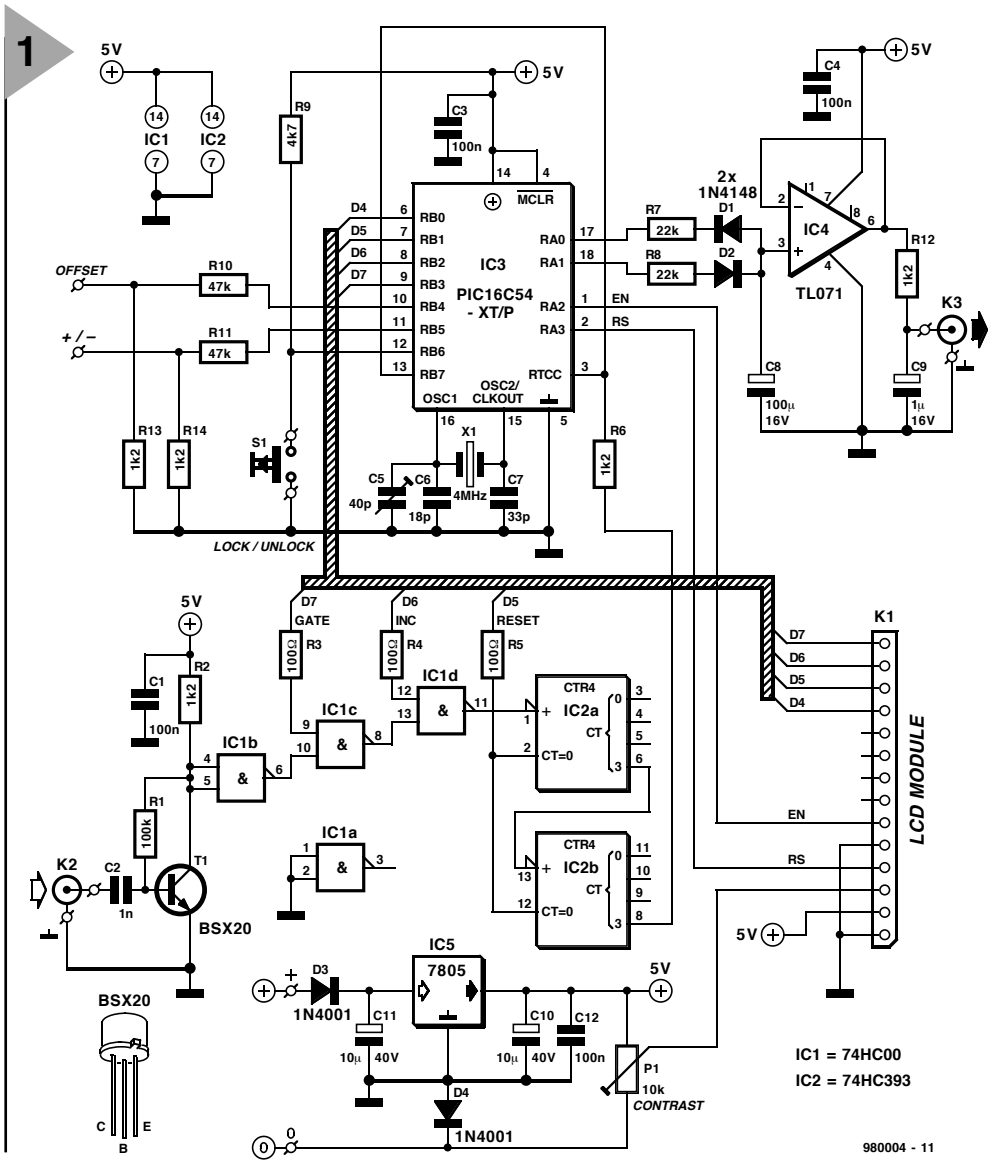


Figure 1. Circuit diagram of the Frequency Display and VFO Stabilizer.

How the VFO is controlled

When you find a frequency that you want to stay on, press the LOCK/UNLOCK button. After the button is pressed there is a 100 ms delay, then the result of the most recent count is stored in three registers in the PIC. The result of all subsequent counts are compared with this value. If the current count is less than the stored value, the VFO has apparently drifted lower in frequency; a positive pulse is sent to the integrator (IC4) to correct the error. If the current count is greater than the stored value then a negative correction pulse is generated.

The width of the correction pulse depends on the degree of VFO drift. If the error is less than 10 Hz then a very short pulse of about 2 ms duration is generated. Greater frequency errors result in longer correcting pulses: 20 Hz = 4 ms, 30 Hz = 6 ms, 40 Hz = 8 ms and so on. This results in much tighter control of the VFO than can be achieved with a conventional 'Huff and Puff' circuit.

When the circuit is in locked mode, the LCD display readout changes: 'MHz' disappears from the display and is replaced by the 10-Hz digit. To the right of this is the error level display which is shown as E0 to E9. E0 means that the error is less than 10 Hz, E1 is an error level of 20 Hz, E9 is an error level of 100 Hz. If the error level is greater than 9, a '9' is still displayed. The last character on the display is the correction direction indicator: '>' indicates a positive pulse, '<' indicates a negative pulse. A low error level indication of 0 or 1 and a continuous rapidly alternat-

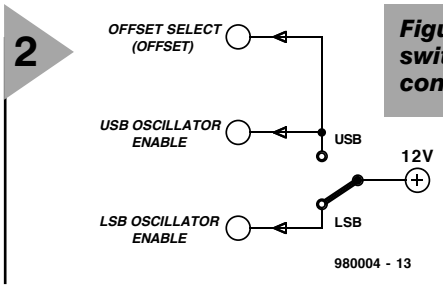


Figure 2. Suggested method of switching the ADD/SUBTRACT (+/−) control input on the board.

ing up/down indication means that all is well, and the VFO frequency is within the control loop bandwidth of the system.

The varicap diode control of the VFO should be arranged so that the maximum frequency change is approximately ±3 kHz. This should be sufficient range to keep a moderately stable VFO locked for hours or even days!

When you need to change frequency, press the LOCK/UNLOCK button again. This puts the device in unlocked mode, the integrator output is set at 2.5 volts and the display changes back to 6 digits followed by 'MHz'. To lock to a new frequency simply press the LOCK/UNLOCK button again.

THE PROGRAM

Various interesting options are available as regards the control software which resides in the PIC microcontroller. To enable you to make your choice, we first tell you what's available for this project, and then make an important statement.

The items available for this project are (1) a **ready-programmed PIC** (for 10.7 MHz IF), (2) a **ready-made PCB** and (3) a **diskette** containing the **source code files** for the PIC control program. For prices, order numbers and other relevant information, please refer to the Readers Services pages elsewhere in this issue.

And now, a serious note. Although this frequency display/VFO stabilizer is fairly easy to build, you should realize that *considerable experience may be required to establish the link with the VFO in your receiver*. Before building this project, you should, therefore be positive about the following points

1. The receiver is a heterodyne (mixer) design. The VFO frequency is between 8 and about 80 MHz.
2. The VFO has varicap control allowing a tuning range of ±3 kHz to be produced by a control voltage swing of 5 V (2.5 V = centre tuning).
3. The VFO signal can be 'tapped' in a safe way (preferably by inductive coupling) and has a level of at least 100 mV_{pp}.

Most experienced radio amateurs (and not only those who actually transmit!) will be able to come to terms with these conditions, if necessary with the help of a fellow ham.

Back to your options! Here's what you can do.

1. I have a receiver with an IF of 10.7 MHz.

Simply order the PCB, the ready-programmed PIC and the source disk supplied through our Readers Services.

2. I have a receiver with an IF other than 10.7 MHz.

Order the PCB (**980004-1**) and the source code disk (**986006-1**) as separate items. Purchase a PIC16C54, and get hold of a an assembler and a PIC programmer. Edit the source code as explained in the README file, and then program your own PIC.

3. I have a receiver with an IF of 455 kHz or 7.8 MHz.

Do the same as under 2. The necessary files are on the disk.

4. I can make my own PCBs and program my own PICs.

Order the diskette only (**986006-1**). Make your own PCB using the artwork shown in this article. Burn your own PIC for the IF you require. Tell your friends about it.

A PIC programmer can be built from one of the many published designs, or a commercially made unit can be purchased from one of several companies advertising in this magazine. The source code file on disk may be edited using any ASCII word processor. Details on modifying the IF offset are also available. Examples are available for 10.7 MHz, 455 kHz and 7.8 MHz.

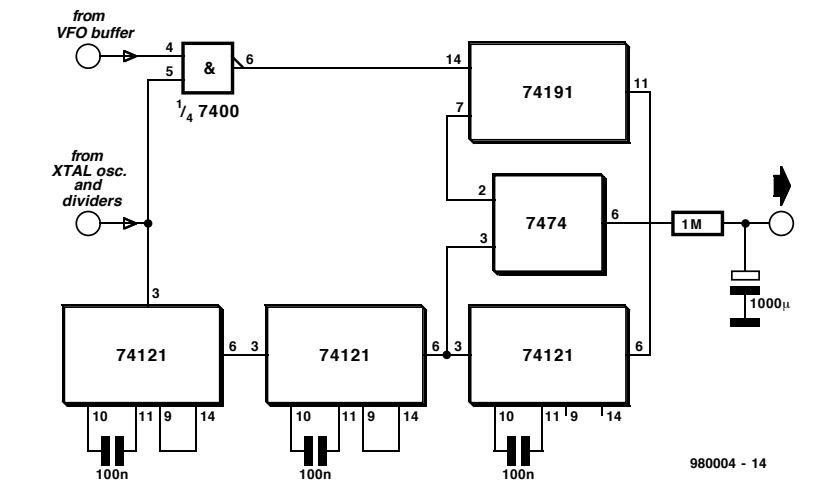
CONSTRUCTION

If you use the PCB layout shown in **Figure 4** and a ready-made board, construction of the circuit is fairly easy. Check the orientation of all polarized components (electrolytic capacitors, diodes, ICs, transistor T1). Use sockets

Huff and Puff

Although this circuit is generally referred to as the 'huff and puff' stabilizer among English-speaking hams, it should really be called the PA0KSB VFO stabilizer after its inventor, the Dutch radio amateur Klaas Spaargaren, PA0KSB. The circuit can act as an outboard enhancement with any reasonable VFO, keeping the output frequency stable within a couple of hertz without adding parasitics and other whistles to the VFO output.

A crystal oscillator, whose output frequency need not be a round value, is followed by a divider cascade which open a gate for, say, 1 second. Next, a binary counter counts the VFO cycles within this gate period. On closing the gate, the last counter digit is compared to '8'.If it is smaller, the 2^3 output of the counter is reset to 0. If it is greater, the output is made logic 1. The counter output is applied to a D-bistable. If the gate signal drops to 0, the first one-shot is triggered. Next, the second one-shot clocks the 1 or the 0 into the D-bistable. Finally, counter is reset by the third one-shot. If the counter signal was a 0, the $\overline{Q}$ output of the bistable goes high, charging the capacitor and so causing the VFO frequency to go up. By contrast, a counter signal of 1 causes the frequency to go down. In this way, the VFO frequency 'puffs' at a rate of a few hertz around the stabilization point at which the counter detects an '8' as the last digit. Over the years, the basic design by PA0KSB was enhanced and followed by several variants using a raster of about 40 Hz, allowing virtually continuous tuning.



3

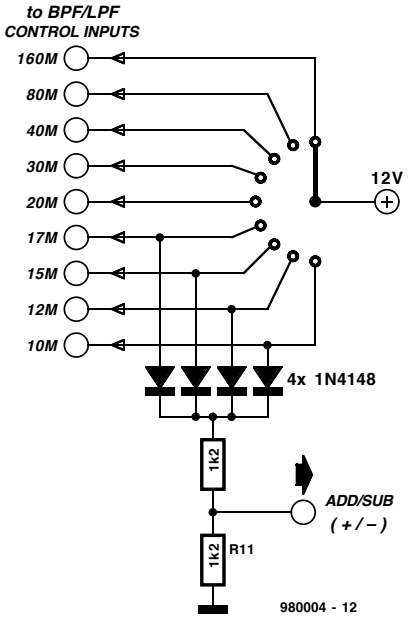


Figure 3. Method of switching the OFFSET select input on the board. Note that USB/LSB selection is only required on SSB receivers.

for the ICs if you want to experiment with different logic IC families, LS, ALS, HC etc. If you use 74HC or 74ALS series chips for IC1 and IC2 it will not be necessary to have a heat sink on the 5-V regulator. The circuit was tested with HC ICs, and worked reliably up to about 50 MHz. Higher input frequencies should be possible if you use 'ALS ICs.

The type of opamp used for IC4 is quite critical, the ADOP07CN gave very good results. If you do not have an ADOP07CN available, the TLO71CN also works quite well.

THE DISPLAY

Any general-purpose 1-line 16-character display that uses the Hitachi HD44780 chip should be suitable (the author used a type 16166 LCD display module). Some of these displays have LED or electroluminescent backlighting built in. Do not spend large sums of money on these displays, they are often advertised for less than £5.00.

TESTING

When the unit is first powered up, adjust preset P1 for best contrast on the LCD display. Connect your VFO to the input (C2), set the ADD/SUB input high to add the IF offset, or low to subtract the IF offset. Set the OFFSET switch for USB (high) or LSB (low). The display should show the approximate frequency of the VFO plus or minus the IF offset.

The best way to calibrate the counter accurately is to tune your receiver to a frequency standard signal or a station of known frequency accuracy, then adjust C5 until the displayed frequency is correct. The capacitor values shown in the circuit diagram worked fine with several 4-MHz crystals from the author's junkbox. If the crystal you are using requires different capacitor values it may be necessary to change the value of C6.

In the author's HF transceiver, the circuit was put in a small box made from copper clad glass fibre board, this box was mounted on top of the VFO. The RF input and control output connections were made with miniature (RG174 or similar) coaxial cable. A short length of ribbon cable is used to connect the LCD module to the PCB. The OFFSET select input may be connected to the USB/LSB switch on the front panel of the rig. The ADD/SUBTRACT (+/−) input may be connected to the band switch of the rig using four diodes (see Figure 2). If you only need to subtract the IF offset, you can connect the add/subtract (+/−) input to ground.

(980004-1)

Components list

Resistors:
R1 = 100 kΩ
R2,R6,R12,R13,R14 = 1kΩ2
R3,R4,R5 = 100Ω
R7,R8 = 22kΩ
R9 = 4kΩ7
R10,R11 = 47kΩ
P1 = 10kΩ preset H

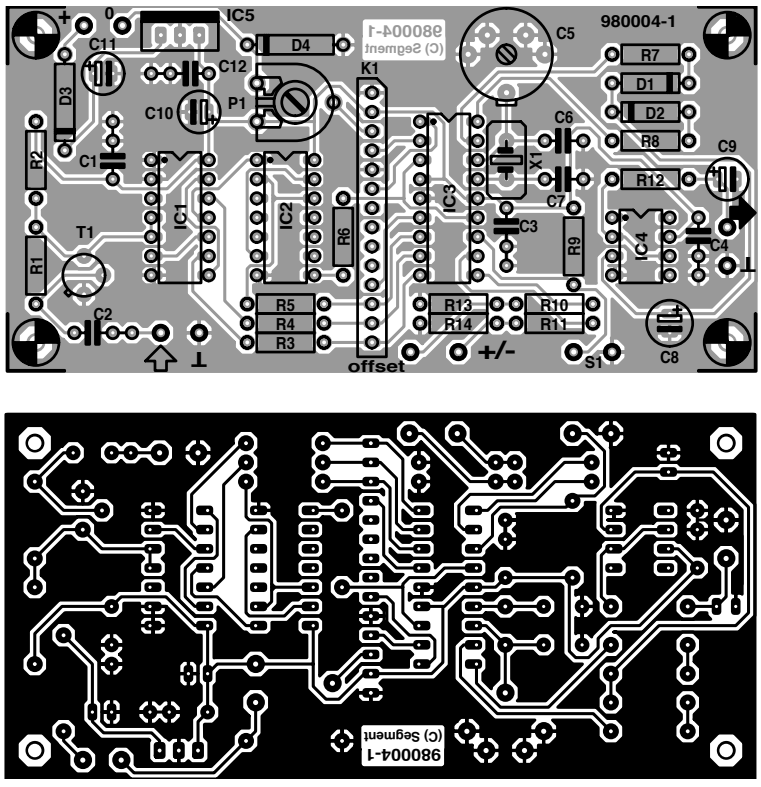
Capacitors:
C1,C3,C4,C12 = 100nF
C2 = 1nF
C5 = 40pF trimmer
C6 = 18pF ceramic
C7 = 33pF ceramic
C8 = 100μF 16V radial
C9 = 1μF 16V radial
C10,C11 = 10μF 40V radial

Semiconductors:
D1,D2 = 1N4148
D3,D4 = 1N4001
T1 = BSX20
IC1 = 74HC00 (see text)
IC2 = 74HC393 (see text)
IC3 = PIC16C54-XT/P (order code 986502-1)
IC4 = TL071CP
IC5 = 7805

Miscellaneous:
X1 = 4 MHz quartz crystal
S1 = push-button, 1 make contact
K1 = 14-way SIL header
LCD module 1×16 characters
PCB only, order code 980004-1
Disk only, order code 986006-1
PIC only, order code 986502-1

Figure 4. Copper track layout and component overlay (board available ready-made through the Readers Services).

4



AVC for PCs

limits differences in sound level

An annoying phenomenon (not restricted to PCs) is that each and every programme that produces sound does so at a different level. This means almost constant adjustment of the volume control to ensure audibility of one programme and protection of the ear drums with another one. The control circuit described in this article is designed to obviate this nuisance: it constantly monitors the signal even at the output of the soundcard and adjusts it when required. Use of the circuit is not restricted to PCs; it may also be used as a dynamic limiter in existing audio equipment.

Brief specification

Power output	*1.2 W*
Maximum input	*1 V*
Compression	*10:1*
Supply line	*12 V, 6 VA*
Output load	*8 Ω (LSP); 10 kΩ (line)*
Input sensitivity	*280 mV (gain line in to out = 0 dB; distortion at output = 1%)*
	120 mV (gain line in to out = max; distortion at output = 1%)
Line in to LSP out (input voltage = 200 mV)	
THD+N	*0.25% (2×0.5 W)*
Signal-to-noise	*70 dB for 0.5 W output at maximum gain*
Channel separation	*>45 dB*
Line in to line out (input voltage = 200 mV; no loudspeaker connected)	
THD+N	*0.047%*
Signal-to-noise	*80 dB*
Channel separation	*>73 dB*

Design by T. Giesberts

In general, the signal levels in current audio equipment are equalized and standardized (although there are still some exceptions). Software manufacturers do not seem to know or care about this. Anyone who has ever opened two different sound programs will know of the quite different levels various effects often have. This is obviously an annoying situation and one which makes the constant adjusting of the volume control a necessity.

The present circuit offers a solution to this problem. It consists of a dynamic compressor with a control range of 10:1 which ensures that very loud and very soft sound passages are attenuated or amplified respectively. This results in a much narrower dynamic range of audio signals which makes adjusting the volume control a much less frequent necessity. It proves that something that appears difficult in software can be easily achieved by a small electronic circuit.

DESIGN

The block diagram of the automatic volume control (AVC) circuit is shown in **Figure 1.** The stereo audio signal at the output of a sound card used in a multimedia PC is applied to the line input. The active part of the circuit consists of two integrated amplifiers that contain a variable preamplifier and a compact output amplifier.

The signal from the output amplifier is freed from any direct voltage and then applied to a discrete rectifier. After the rectified signal has been processed, it is used to control the amplification factor of the preamplifiers. The control circuit is based on a number of discrete operational amplifiers. The design is an OR-type, so that the sound channel (lefthand or righthand) with the highest peak signal level determines the amplification factor of

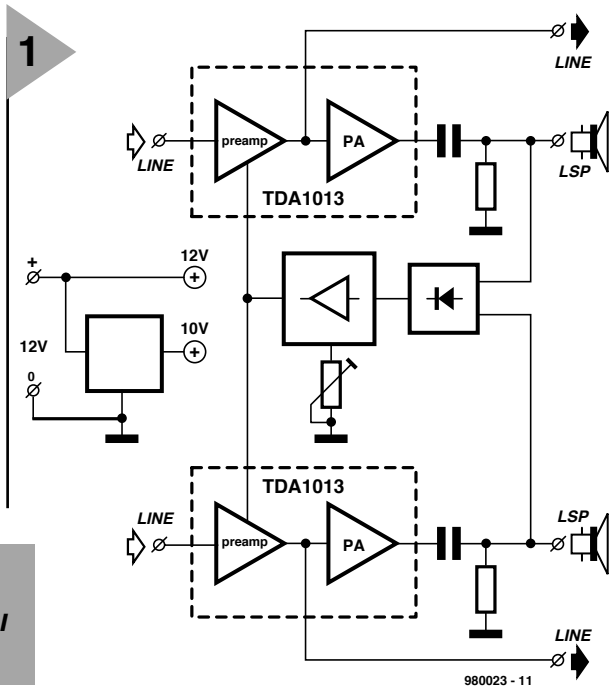


Figure 1. Block diagram of the automatic volume control circuit for PCs.

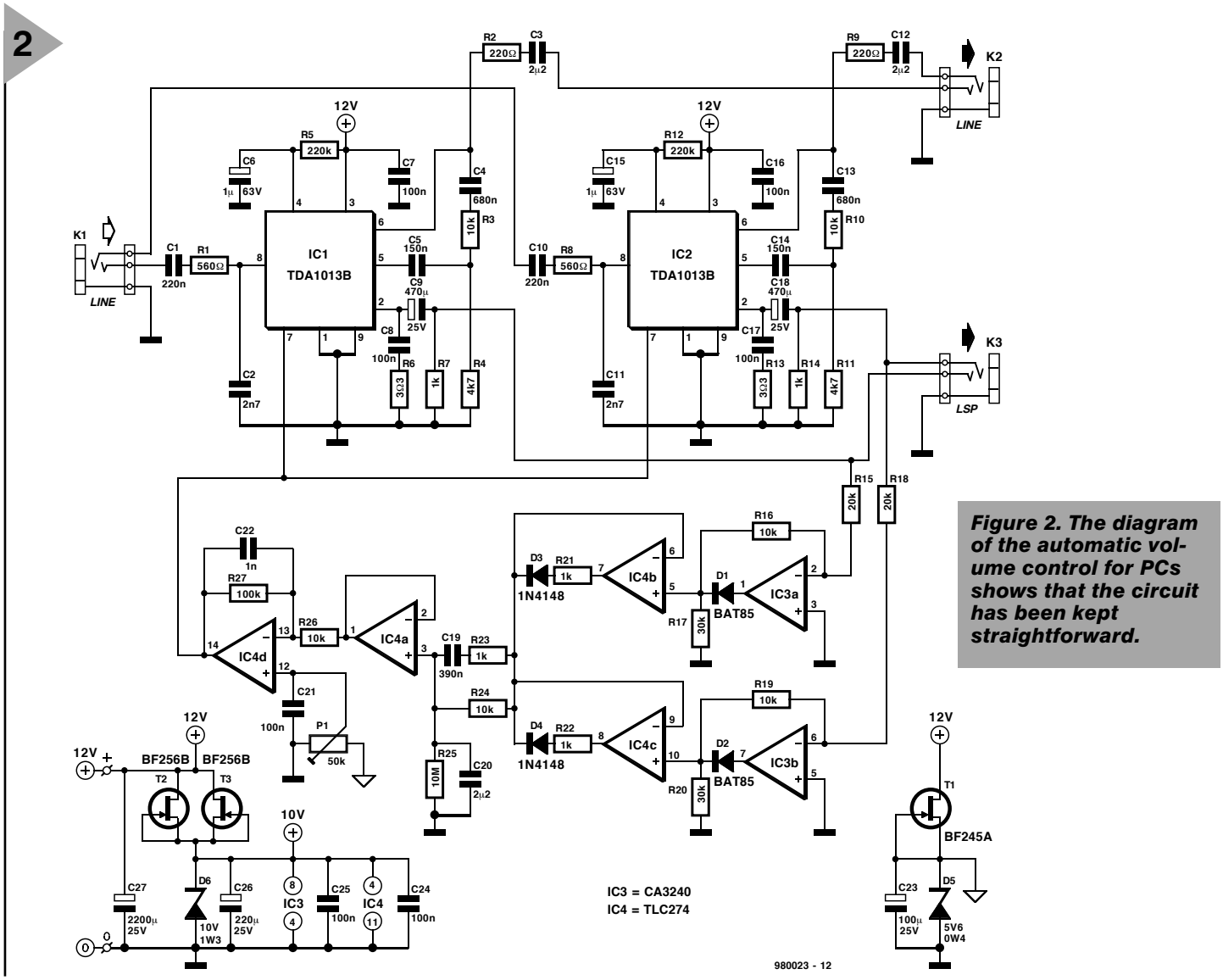


Figure 2. The diagram of the automatic volume control for PCs shows that the circuit has been kept straightforward.

Parts list

Resistors:
R_1, R_8 = 560 Ω
R_2, R_9 = 220 Ω
R_3, R_{10}, R_{16}, R_{19}, R_{24}, R_{26} = 10 kΩ
R_4, R_{11} = 4.7 kΩ
R_5, R_{12} = 220 kΩ
R_6, R_{13} = 3.3 Ω
R_7, R_{14}, R_{21}, R_{22}, R_{23} = 1 kΩ
R_{15}, R_{18} = 20 kΩ
R_{17}, R_{20} = 30 kΩ
R_{25} = 10 MΩ
R_{27} = 100 kΩ
P_1 = 50 kΩ (47 kΩ) preset

Capacitors:
C_1, C_{10} = 0.22 µF
C_2, C_{11} = 0.0027 µF
C_3, C_{12}, C_{20} = 2.2 µF metallized polyester (MKT), pitch 5 or 7.5 mm
C_4, C_{13} = 0.68 µF
C_5, C_{14} = 0.15 µF
C_6, C_{15} = 1 µF, 63 V, radial
C_7, C_8, C_{16}, C_{17}, C_{21}, C_{24}, C_{25} = 0.1 µF
C_9, C_{18} = 470 µF, 25 V, radial
C_{19} = 0.39 µF
C_{22} = 0.001 µF
C_{23} = 100 µF, 25 V, radial
C_{26} = 220 µF, 25 V, radial
C_{27} = 2200 µF, 25 V, radial

Semiconductors:
D_1, D_2 = BAT85
D_3, D_4 = 1N4148
D_5 = zener diode 5.6 V, 400 mW
D_6 = zener diode 10 V, 1.3 W
T_1 = BF245A
T_2, T_3 = BF256B

Integrated circuits:
IC_1, IC_2 = TDA1013B
IC_3 = CA3240E
IC_4 = TLC274CN

Miscellaneous:
K_1–K_3 = 3.5 mm stereo audio socket for board mounting
PCB Order no. 980023-1 (see Readers Services towards the end of this issue)

3

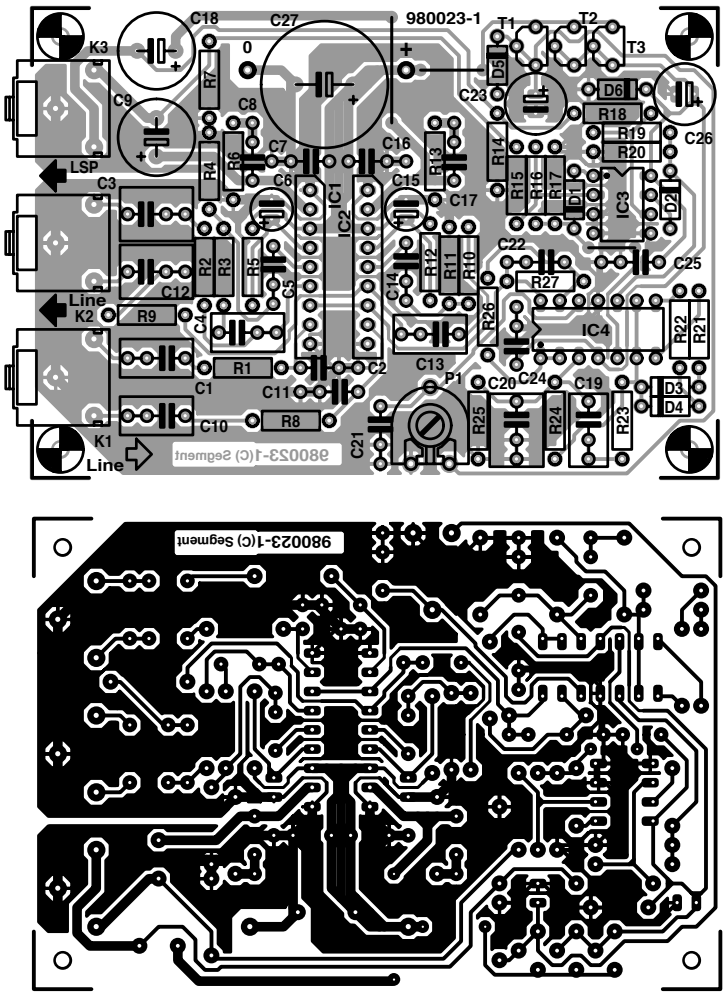


Figure 3. The printed-circuit board for the automatic volume control.

the stereo preamplifier.

The values of various components in the control circuit are chosen to ensure a fast attack time and a long release time. This ensures that short-duration signal peaks are effectively suppressed, whereupon the circuit recovers (relatively) slowly from the damping action.

Power for the circuit is derived from a standard 12 V mains adaptor.

CIRCUIT DESCRIPTION

In the circuit diagram in **Figure 2**, the preamplifier-output amplifier combination is contained in IC_1 and IC_2. This type of IC is a compact 4 W audio amplifier with integral voltage-controlled volume control. The range of the logarithmic volume control is 80–90 dB with control voltages between 6.5 V (+5 dB) and 2.0 V (–80 dB).

The control voltage is applied to pin 7 of the IC. The line output is at pin 6, which is linked via a capacitor to pin 5, the input of the output amplifier..

The circuit has three stereo terminals: line in, line out, and power out.

The power output is 2.5 W for a loudspeaker impedance of 8 Ω and a supply line of 18 V, which is sufficient for most applications.

The analogue input signal at the line input, K_1, is applied to pin 8 of IC_1,raised in the preamplifier and output via pin 6. The transfer between pins 8 and 6 depends on the control voltage at pin 7.

The line signal is attenuated and its level made suitable for inputting to the output amplifier by networks R_3-R_4 and R_{10}-R_{11}. Assuming a supply line of 12 V, the output amplifier is driven fully ($P_{o(max)}$ = about 1.2 W into 8 Ω) by an input signal of 90 mV.

RC networks are provided at the inputs (R_1-C_1-C_2 and R_8-C_{10}-C_{11}) and the line outputs (R_2-C_3 and R_9-C_{12}).

The output amplifier outputs are provided with large electrolytic capacitors, C_9 and C_{18}.

The supply lines are decoupled by C_7 and C_{16}.

Filters R_6-C_8 and R_{13}-C_{17} ensure that the amplifiers remain stable at high frequencies.

RECTIFICATION AND REGULATION

The audio signal to be rectified is taken from the loudspeaker terminals and applied to IC_{3a} and IC_{3b}. The following description is based on IC_{3a}.

Negative signals are inverted by the op amp and amplified by a factor that depends on the ratio R_{15}:R_{16}. In the present circuit, this is –2, that is, attenuation. With positive signals, the op amp is overdriven and its output negative. Diode D_1 is then cut off and half the input voltage is available at its cathode [R_{17}/(R_{15}+R_{16}+R_{17})]. This means that the op amp behaves as a full-wave rectifier/amplifier, whose amplification is the same (0.5) for both

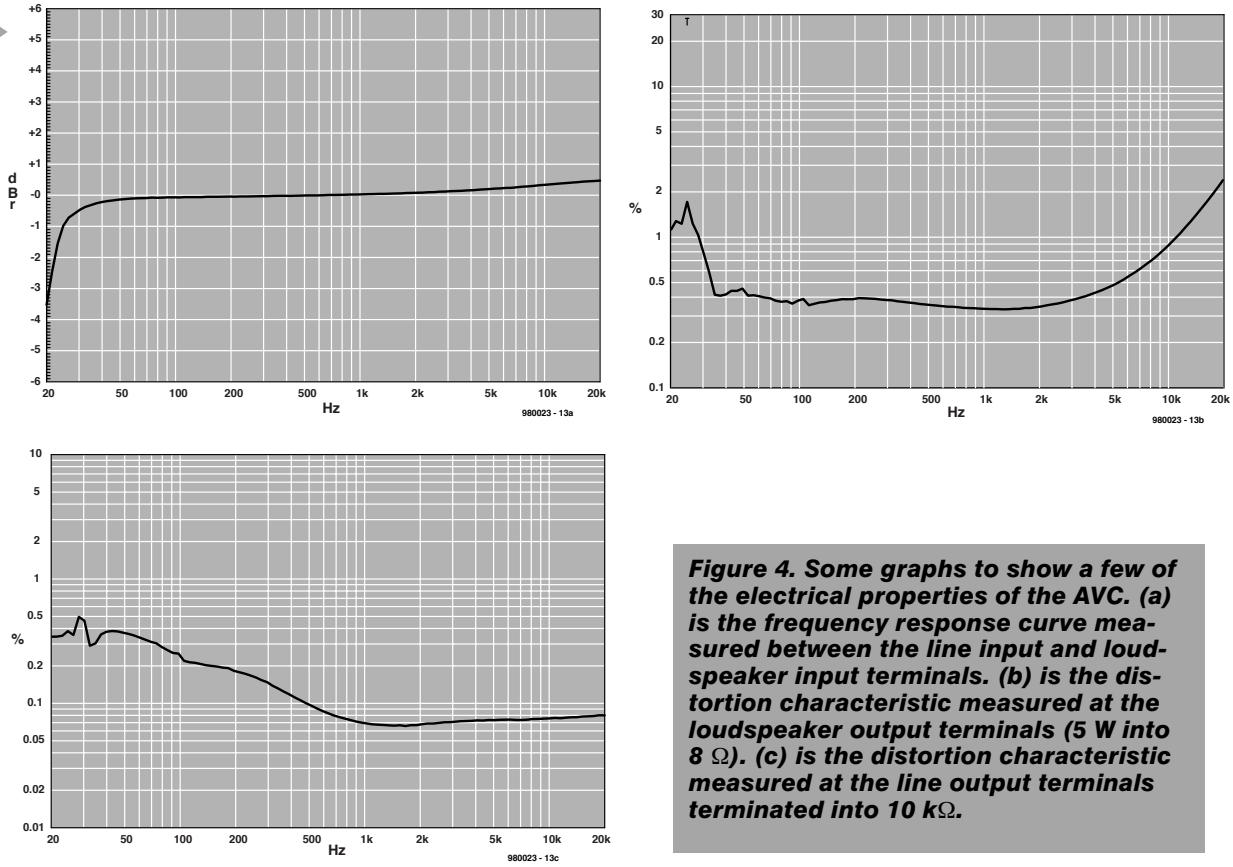


Figure 4. Some graphs to show a few of the electrical properties of the AVC. (a) is the frequency response curve measured between the line input and loudspeaker input terminals. (b) is the distortion characteristic measured at the loudspeaker output terminals (5 W into 8 Ω). (c) is the distortion characteristic measured at the line output terminals terminated into 10 kΩ.

halves of the input signal.

Operational amplifiers IC_{4a} and IC_{4b} are half-wave rectifiers whose outputs are interlinked by diodes D_3 and D_4. Because of these diodes, the output with the highest potential determines the extent to which capacitor C_{20} is charged via resistor R_4. Network R_{23}-C_{19} has been added to ensure that fast signal fluctuations are passed on very rapidly.

Capacitor C_{20} is discharged slowly via resistor R_{25}, so that the control circuit returns to its default setting when no or a smaller input has been applied for some time. The potential across C_{20} is buffered by IC_{4a}, while IC_{4d} ensures that the (fixed) default level is added to the signal. The resulting control signal is applied to the control input (pin 7) of IC_1 and IC_2.

With component values as specified, the compression is 10:1; in other words, a 20 dB change at the input results in a 2 dB change at the output.

The setting of P_1 depends on the signal level at the input of the circuit. Since this level varies largely from one sound card to another, the design provides a wide control range.

SUPPLY LINES

As mentioned earlier, the circuit is powered by a standard 12 V mains adaptor, which is applied directly to the output amplifier. All other circuit elements are supplied with a regulated 10 V potential. This voltage is produced with the aid of current source T_2-T_3 and zener diode D_6.

The reference voltage of 5.6 V is produced with the aid of current source T_1 and zener diode D_5.

CONSTRUCTION

The circuit is best built on the printed-circuit board shown in **Figure 3** (see Readers Services towards the end of this issue). Start the construction with placing audio sockets K_1–K_3, the three wire bridges, and all solder pins, and follow these with first the passive components, and then the active ones. Mind the polarity of the electrolytic capacitors, diodes, transistors, and ICs. After it has been fitted, set the preset to minimum volume (anticlockwise).

Solder the output leads from the standard mains adaptor from which power is derived to the relevant pins on the board. If the board is to be housed in an enclosure, a plug-and-socket arrangement should be used for linking the output from the adaptor to the board.

Check that the output voltage of the adaptor does not rise above 18 V with small loads.

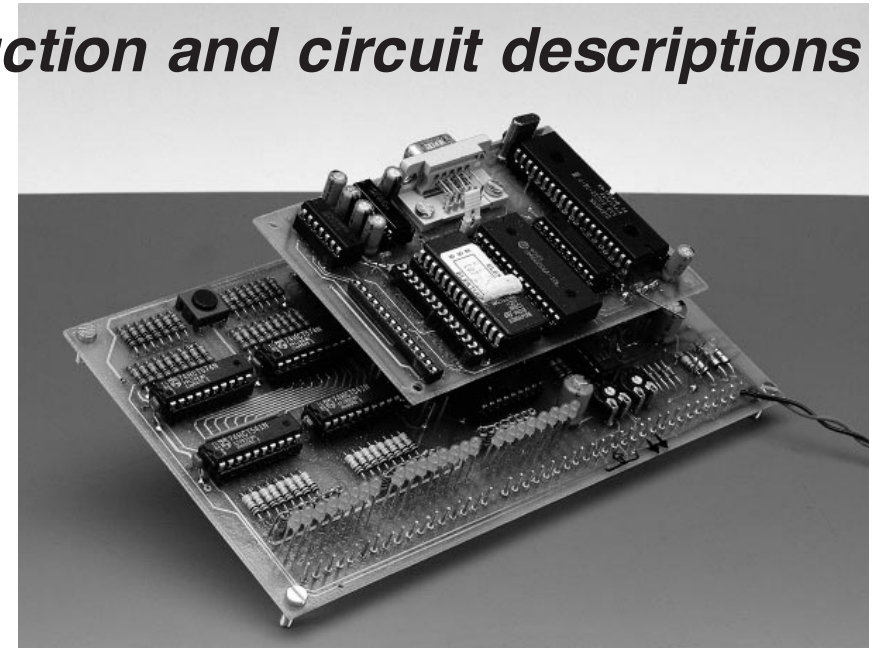
When all is connected, the circuit can be tested. Passive loudspeakers may be linked directly to the LSP output terminals, but active ones should be connected to the line output terminals.

Finally, connect a sound source, for instance, the line output of a sound card or the output of a Walkman™ to the input of the circuit and adjust P_1 for the desired volume. From then on, any fluctuations in the signal input level will be minimized automatically.

[980023]

80C32 BASIC control computer

Part 1: introduction and circuit descriptions



Faced with the ominous task of having to cajole an existing microcontroller board into managing a control function, many of you will recognize the problem of having to develop a more or less complex interface board. Another problem may be the complexity and general hassle of everything to do with assembly language. The computer described in this article may be programmed in MCS-51 BASIC, and has been designed specifically for control purposes.

Design by H.-J. Böhling

The 80C32 BASIC control computer consists of two boards which are interconnected via three single-row pinheaders. The smaller of the two boards (10×8 cm) is the actual control computer. The other board is considerably larger at 10×16 cm (Eurocard size), and contains all input/output circuits and 45 solder pins, or, if so desired, screw terminals.

The control computer runs the MCS-51 BASIC interpreter, and has been designed for maximum flexibility and optimum price/performance ratio. It can also be used in stand-alone mode. For this purpose a reset network (R1-C12) is provided, which may only be fitted if the computer is used in stand-alone mode.

Many options are open to those of you wishing to realize their own applications using the present control computer: for instance, the three pinheaders for the inter-board connections are arranged in a 0.01-inch raster which allows a piece of Veroboard or general-purpose stripboard to be used as a carrier.

The control computer was developed with low cost and simple construction in mind. Because of this, the address decoding is not handled by a costly PAL but by common-or-garden 74HCT logic. The same goes for nearly all I/O functions. With the possible exception of the type μPD7002 A/D converter from NEC, the add-on multifunction board does not contain esoteric components either.

THE 80C32 MAIN COMPUTER BOARD

The design of the 80C32 BASIC control computer follows well-trodden paths. In addition to the microcontroller type 80C32 (IC1) with its low-address latch (IC20) the main board also contains three memory ICs: a 32-kByte static RAM (IC3), an EPROM (IC4) with a capacity of up to 32 kBbytes for the 8-kByte BASIC interpreter and your own BASIC extensions, as well as up to 16 kBytes of EPROM (IC5) to hold your BASIC programs. The address decoding for IC5 is handled by a 74HC00 (IC6). No address decoding is required for memories IC3 and IC4 because each of them occupies a block of 32 kBytes in the lower memory range. That enables them to be addressed in a simple manner using address line A15 to control their (active-low) chip select inputs.

Jumper JP1 enables pin 27 of IC4 to be connected to A14 (for a 27256 EPROM), or to + 5 V, so that 16-kByte as well as 8-kByte EPROM may also be used in this position. Position IC5 can only accommodate 16-kByte or 8-kByte EPROMs, because the upper address range is required for the I/O range, and only 16-kByte EPROMs may be used in any case as the BASIC program memory.

The 80C32 is a ROM-less CMOS version of the 8052 for which the MCS51-BASIC interpreter was originally written (by Intel). This interpreter requires a rather special memory structure, as illustrated in **Figure 1**.

The main board, of which the global structure is shown in **Figure 2**, provides a battery backup supply for the RAM chip, IC3. Normally, the + 5 V supply line (pin 1 of K2) is connected through to pin 1 (VRM) of K1. This connection, as well as that between IC3 and A15, has to be broken. To make the board go into low-power mode, the $\overline{CS}$ input (pin 20 of IC3) also has to be connected to pin 14 of K2. Because the multi-function does not make use of this option, the relevant connections remain intact.

The last noteworthy component on the main board is the MAX232 converter for the RS232 interface (IC7). This chip and the associated 9-way sub-D connector forms the gateway to the PC (or terminal/console) on which you develop your programs for the control computer.

MULTIFUNCTION EXTENSION BOARD

The circuit diagram of the multifunction extension board is given in **Figure 3**. It accommodates the power supply, the address decoding circuits for the I/O range, the watchdog timer, the digital inputs and outputs, as well the analogue inputs and outputs. The operation of each of these sub-circuits is described in the following sections.

Power supply

The power supply is open to modifications to meet your specific requirements. All that is required really is a smoothed direct voltage of at least + 12 V at about 150 mA which has to be applied to terminal block K25. Note that a capacity of 150 mA is only sufficient for the control computer and all LEDs. If K24 is not connected, driver IC8 (for the digital and analogue outputs) is supplied at just 11 V instead of 24 V. A higher voltage (up to about + 30 V) should only be connected to K24 if higher driver output voltages are required, for example, to control 24-V relays which are often used in industrial equipment. Diodes D1 and D3 afford protection against reverse polarity.

Watchdog timer

The watchdog timer consists of a monostable (IC4d) with a period of about 10 s, using R4 and C11 as the timing elements. If no reset signal is available, then C11 will discharge across R4. This can only be prevented by permanent recharging by means of signals from CPU port P1.5, which are transformed into suitable pulses by IC4e and IC4f, using C10 and R1.

The listing of the *Background Clock with Watchdog Timer Reset* program (available on the project floppy disk) shows how an on-time interrupt may be used to generate watchdog-feed pulses. Diode D4 prevents capacitor C11 from discharging across the output of IC4e. A reset can also be brought about by pressing S1. Fitting jumper JP1 disables the watchdog timer. LED D5 goes out when a reset arrives. Diodes D36 through D43, in combination with resistors R75 and R76, protect the CPU inputs against voltage surges.

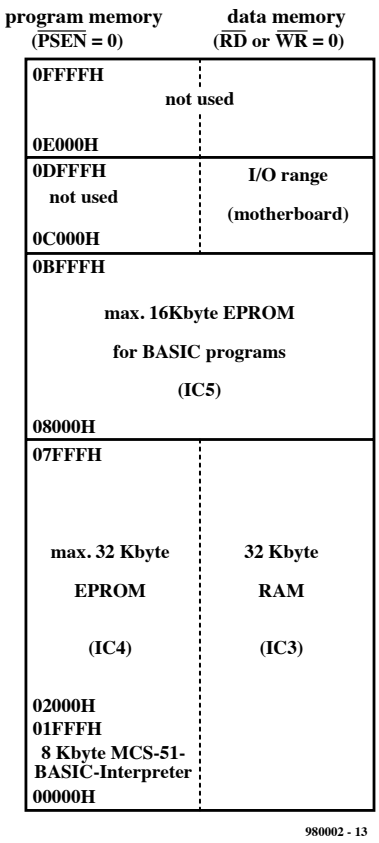
Address decoding

The address decoding of the I/O (input/output) range relies on a 74HCT139 (IC3). This simple circuit selects the 8-kByte large address range between $0C000_H$ and $0FFFF_H$, and also supplies four I/O select signals ($\overline{IO1}$ through $\overline{IO4}$).
This creates a cluster of 16 I/O addresses between $0C000_H$ and $0C00F_H$ (see table below).
The above address block is repeated from address $0C010_H$, because 512 mirror images appear in the 8-kByte large I/O range.

16 digital inputs

The 16 digital (TTL-compatible) inputs are created with the aid of two 8-bit input drivers type 74HCT541 (IC5 and IC6). Resistors R59-R74 act as protection devices. The LEDs, D12-D27, not only indicate the status of the input signals (high/low) but also pull the inputs high via their series resistors R7 and R8. This is useful when nothing is connected to the inputs. Consequently, the inputs are active-low. If you want a particular line to be permanently low, simply tie it to 0 V (ground). Although this 'inverse logic' has be taken into account when you start programming the computer, it has a definite advantage in that no + 5-V supply is needed outside the control computer. The example program called *DO and DI Test* on the project diskette shows how the inputs are interrogated.

1 ***Figure 1. This rather special memory structure is required by the MCS51-BASIC interpreter.***



8 digital outputs

The 8 digital outputs are beefed up by an ULN2803 (IC8), an integrated darlington driver with open-collector outputs capable of switching loads of up to 50 V at 0.5 A. Each output of the ULN2803 has a built-in suppressor diode which allows inductive loads like stepper motors and relay coils to be controlled without problems. If you really want to go up to 50 V with the

Select signal	*Address*	*Read*	*Write*
$\overline{IO1}$	*0C000H*	*digital inputs 1-8*	*digital outputs 1-8*
	0C001H	*digital inputs 1-8*	*digital outputs 1-8*
	0C002H	*digital inputs 1-8*	*digital outputs 1-8*
	0C003H	*digital inputs 1-8*	*digital outputs 1-8*
$\overline{IO2}$	*0C004H*	*digital inputs 8-16*	*analogue output 1*
	0C005H	*digital inputs 8-16*	*analogue output 1*
	0C006H	*digital inputs 8-16*	*analogue output 1*
	0C007H	*digital inputs 8-16*	*analogue output 1*
$\overline{IO3}$	*0C008H*	*not used*	*analogue output 2*
	0C009H	*not used*	*analogue output 2*
	0C00AH	*not used*	*analogue output 2*
	0C00BH	*not used*	*analogue output 2*
$\overline{IO4}$	*0C00CH*	*ADC status register*	*ADC status register*
	0C00DH	*ADC high data*	*not used*
	0C00EH	*ADC low data*	*not used*

control voltage, watch the voltage rating of C12, and operate the D-A (digital-analogue) converter at + 8 V!

LEDs D28-D35 indicate the logic states of the outputs. The 8-bit output word is written into D-latch IC7 (a 74HCT574).

The clock signal is created by combining the $\overline{WR}$ and $\overline{IO1}$ signals in wired-OR gate D6-D7 and NOR gate IC4a.

As with the digital inputs, inverse logic applies to the digital outputs: a logic 1 at the input results in a logic 0 at the associated output. Consequently, load currents are 'sunk', i.e. any loads connected to the digital outputs have to be permanently connected to the positive rail of their supply, the digital outputs of the ULN2803 acting as switches to ground.

4 Analogue inputs

The four analogue inputs on the multifunction extension board are connected to an ADC (analogue-digital converter) type μPD7002 (IC1), which uses an internal multiplexer to select one of the four input channels. For simplicity's sake, the 2.5-V reference voltage is derived from the supply voltage by potential divider R18-R19. The input voltages also arrive at the ADC by way of potential dividers (R10-R17). The values of the resistors in the potential dividers determine the input resistance as well as the ADC range. The following equations apply, for example, to channel 0:

$R_{in} = R10 + R14$

$V_{in} = V_{ref} (R10+ R14)/R14$

As you can see, using 200-kΩ resistors creates an input resistance of 400 kΩ and a measurement range of 5 V.

The $\overline{EOC}$ (end of conversion) interrupt output of the ADC chip is not used here. If you want to use it anyway, connect it to an interrupt input of the 80C32 CPU. By omitting the associated diode (D42 or D43), the INT0 or INT1 interrupt input of the CPU is then exclusively assigned to the ADC's $\overline{EOC}$ signal.

An A-to-D conversion is launched via the Control Register. The end of it is reported in the Status Register. Next, the value may be read in the Data Registers. In this set-up, the bits have the following meanings (see table below).

The program called *D/A and A/D Conversion Test* should enable you to examine the behaviour of the A-D converter using plain old BASIC.

Control Register (write)

D0, D1	*ADC channel address*
D2	*input flag*
D3 = 0	*8-bit conversion*
D3 = 1	*12-bit conversion*

Status Register (read)

D0, D1	*ADC channel address*
D2	*output flag*
D3	*8-bit or 12-bit converted*
D6 = 0	*Busy (working on conversion)*
D7 = 0	*EOC (conversion finished)*

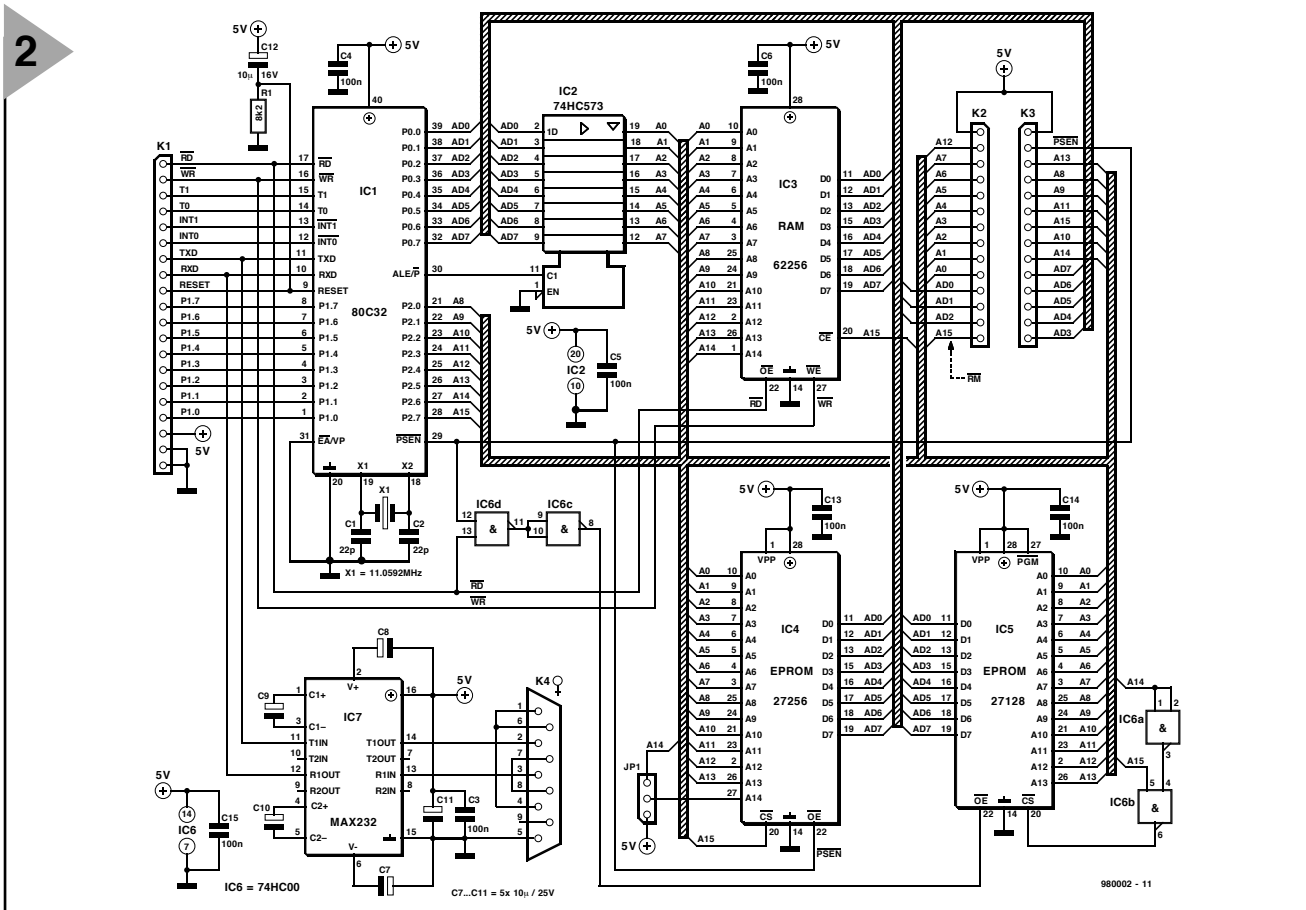
Low-Data Register (read)

D0-D3	*= 0*
D4-D7	*bit 9 - bit 12 (with 12-bit conversion)*

High-Data Register (read)

D0-D7	*bit 1 - bit 8 of conversion result*

Figure 2. Minimum, yet stand-alone, configuration of the 80C32 control computer.



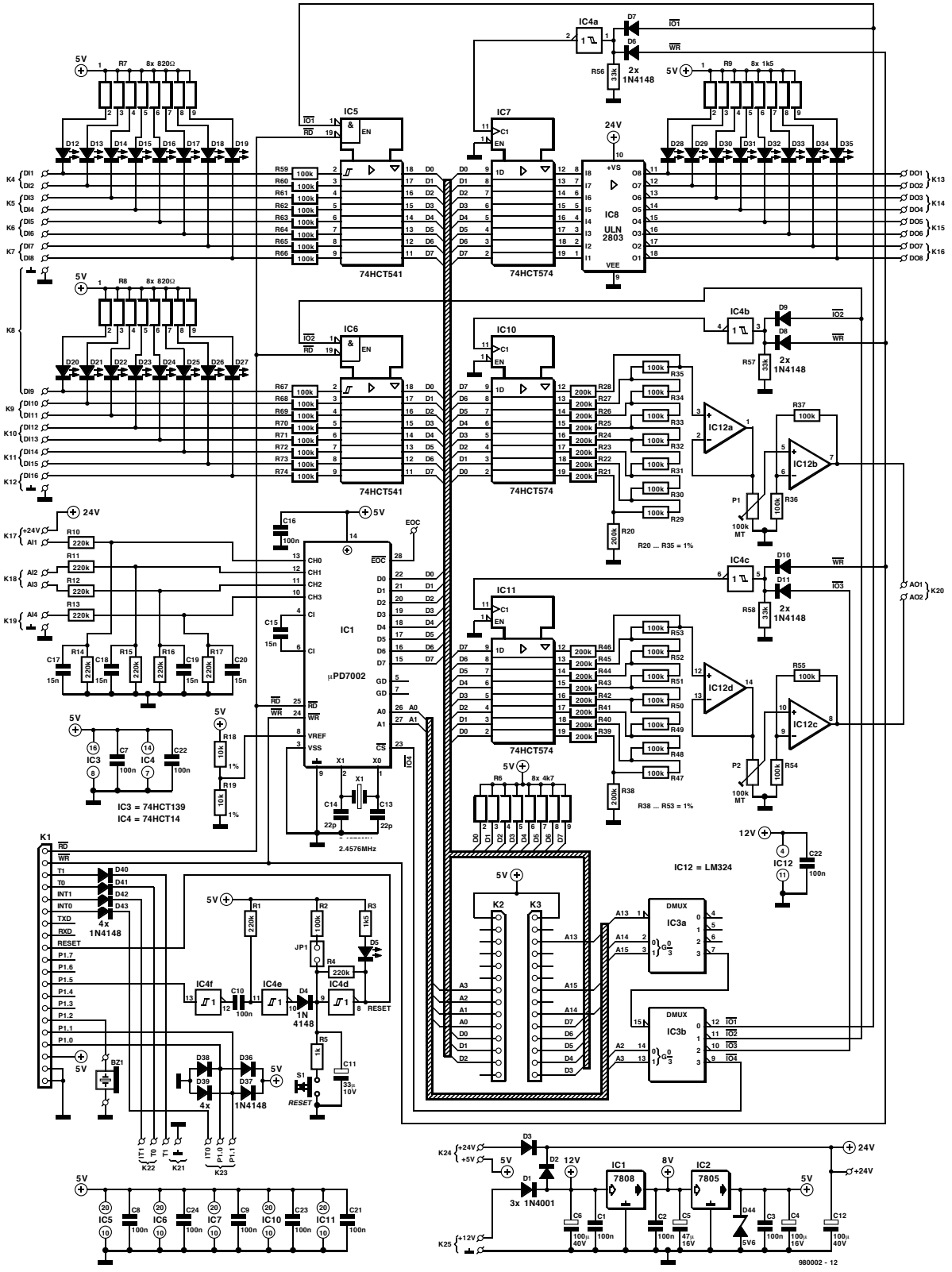
2 analogue outputs

The two analogue outputs work in much the same way as the digital outputs. An 8-bit word is applied to an R-2R network (R20-R35 and R38-R53) rather than an output buffer. Impedance converters IC12a and IC12d buffer the resulting output voltages. Next come adjustable output amplifiers (IC12b and IC12c) which give you accurate control over the output signal levels before they leave the control computer. If necessary, multiturn presets may be used where really accurate output level settings are required.

The output amplifier may be supplied with either an internal voltage of + 8 V, or an external voltage of + 24 V applied by way of terminal block K24. In the first case, install the short wire link under IC12. The long wire is used if you intend to connect a 24-V external supply.

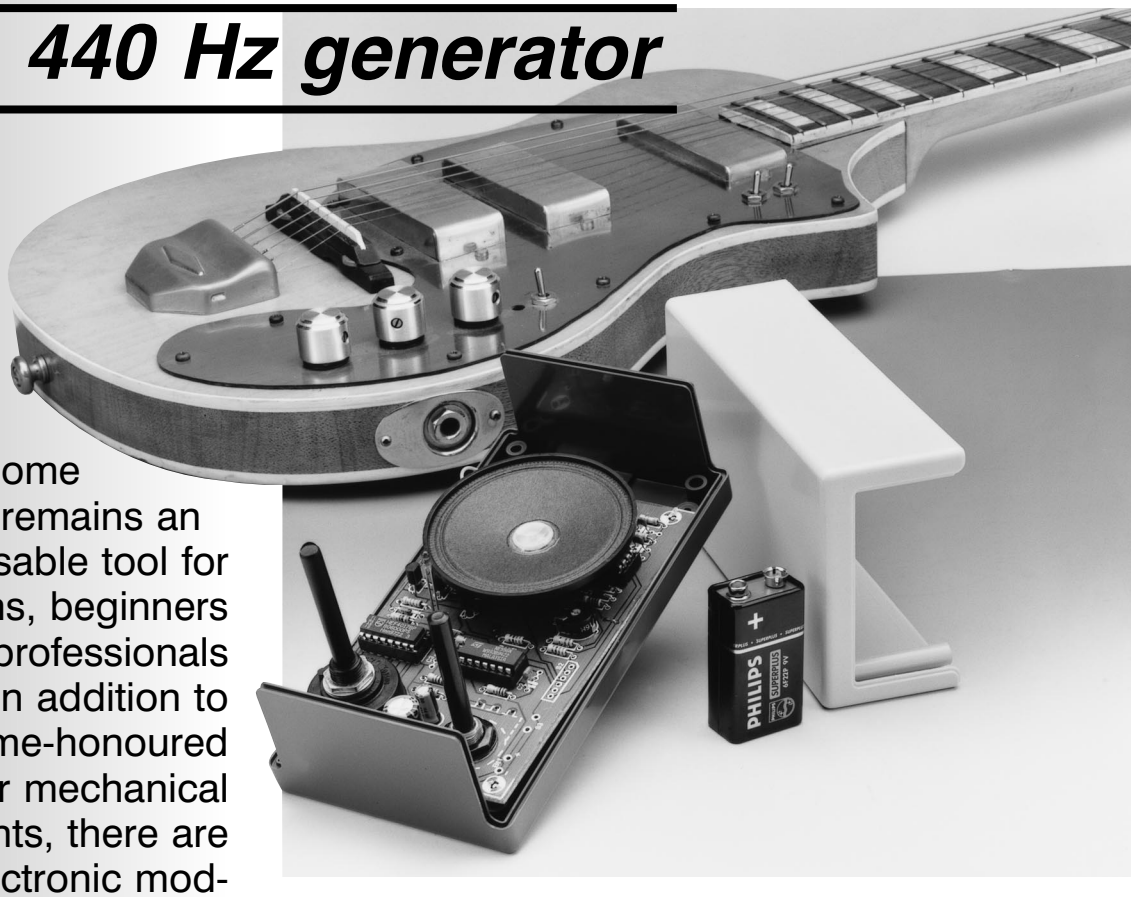
(980002-1)

Figure 3. Circuit diagram of the multifunction extension board.



simple electronic metronome

with 440 Hz generator



A metronome is and remains an indispensable tool for musicians, beginners and professionals alike. In addition to the time-honoured triangular mechanical instruments, there are now electronic models on the market. This article describes a simple DIY metronome that may prove useful to beginners in music.

A metronome is an apparatus for sounding an adjustable number of beats per minute and therefore for fixing the tempo of a composition. The idea of the clockwork model patented by Maelzel seems to have been appropriated from the Dutch inventor D N Winkel.

The one most commonly used is a pyramidal wooden instrument at the front of which a perpendicular steel strip about 3.5 in long by 0.5 in wide is pivoted. The principle is that of a double pendulum (an oscillating rod weighted at both ends). The upper weight is movable along the steel strip and according to its position on the rod the number of oscillations per minute can be made to vary between 40 and 208. The rod beats (or ticks) as it swings back and forth. Maelzel's graduated scale, fixed to the case, gives speed of oscillation. A composer who wants, say, 78 crotchet (US: quarter-note) beats in a minute will write M.M. (Maelzel metronome) ♩ = 78′.

The electronic metronome described here has a useful feature in emphasizing the first beat in a bar. This is done acoustically by increased volume as well as electrically by the lighting of an LED. Another useful feature is the provision of a generator for tuning string instruments.

CIRCUIT DESCRIPTION

Metronome

The generator producing the beats consists of the two halves of a Type 556 CMOS timer, IC_1. Section IC_{1a} is configured as an astable multivibrator., whose frequency can be adjusted with P_1 between 60 and 250 beats per minute. If the lower figure is too fast, the value of C_1 may be increased slightly. On the other hand, the upper limit of 250 beats may be raised (although this is unlikely to be required for music applications) by reducing the value of R_1 to about 1 kΩ.

To ensure a regular, stable frequency, C_1 should preferably be a tantalum capacitor, but if need be, a good-quality electrolytic may be used.

The output of IC_{1a} triggers the other section, IC_{1b}, a monostable multivibrator, via C_2. The monostable generates pulses of constant width in the rhythm of the clock frequency.

The output if IC_{1b} is split into two: one part is applied to the clock input

* In music and audio engineering, a third is a melodic and harmonic interval, taking three steps in a scale (major or minor) counting top and bottom notes. So, *major third* (C up to E), *minor third* (C up to E_b), and *diminished third* (C# up to E_b).

Design by F. Hueber

of decade counter IC_2, and the other to low-frequency output amplifier T_1-T_2 via R_5.

Outputs Q_0–Q_8 of the counter are successively enabled and actuated, that is, in this case, they are changed from logic low to logic high (0 to 1). This is particularly important as regards Q_0 (pin 3), since this is held low for the longest period of time to keep transistor T_3, which functions as a switch, on.

Potential divider R_5-R_6 is then actuated, which results in the pulses arriving from IC_1 being applied at only half amplitude. Output Q_0 goes high only at the first beat in a bar, when the potential divider is not actuated, whereupon the relevant pulse arrives at T_1-T_2 at full strength. Consequently, this beat is rather louder than the others and also sounds a little different. At the same time, the high level at Q_0 causes D_3 to light briefly. This LED also serves as a battery indicator: when the battery voltage drops below about 6 v, zener diode D_3 ensures that the diode remains extinct.

The number of beats in a bar (US: measure) is set with S_1. Depending on the setting of this switch, the counter is reset after 3–8 pulses. In the time signature, which is placed on a sheet of music immediately after the clef sign, the numerator in these fractions indicates the number of beats in a bar and the divisor the value of each beat. Thus, a time signature of 3/4 means that there are three beats in each bar and that the value of each beat is a crotchet or quarter note.

Transistors T_1 and T_2 together form a simple push-pull output stage that operates with virtually no quiescent current. The sound level may be adapted to individual taste by changing the value of R_9 between the output stage and the loudspeaker. Note, however, that the sum of this resistor and the ohmic value of the loudspeaker must not drop below 20 Ω to prevent overloading of the transistors.

Diodes D_1 and D_2 ensure that the output signal is a nearly pure sine wave. The 'nearly' is caused by the inevitable cross-over distortion resulting from the output stage operating without quiescent current. The output frequency is that of the international concert pitch, that is, the tuning-note A (=440 Hz). but may be altered slightly, if desired, with P_3. The actual frequency-determining components are R_{13}, R_{14}, C_5, and C_6.

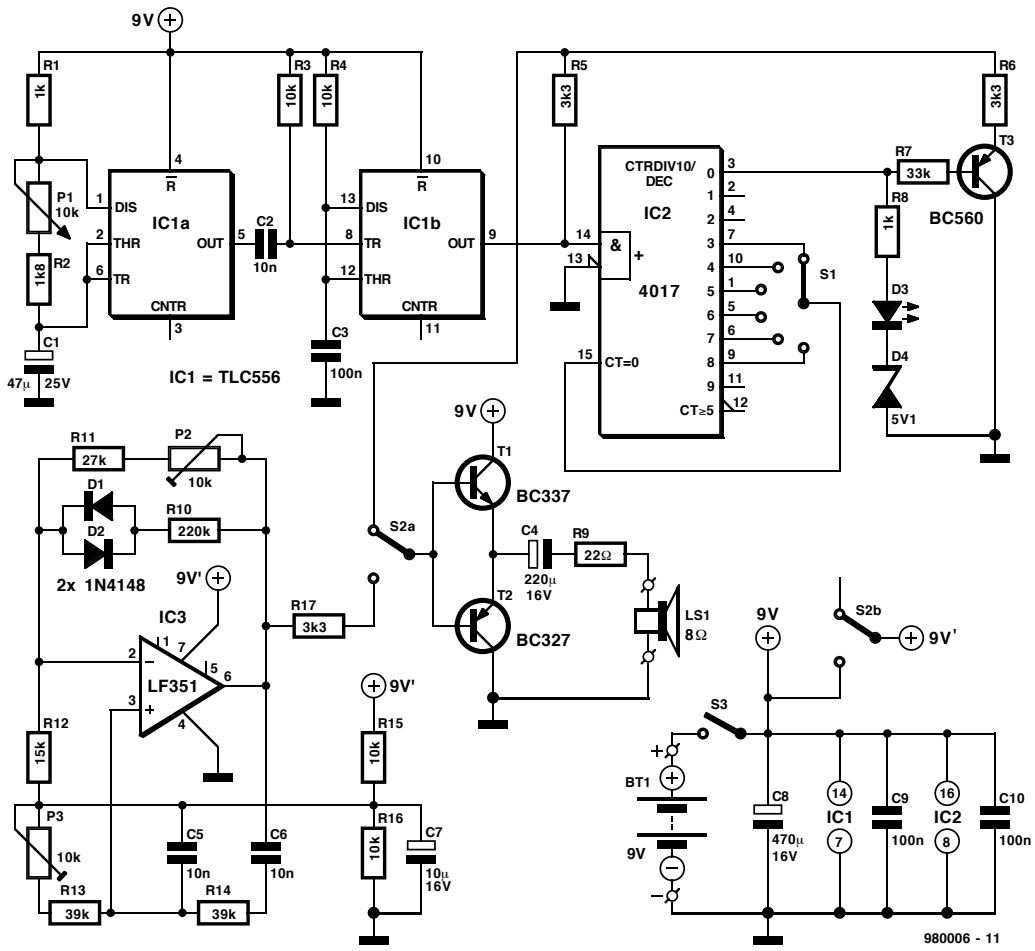


Figure 1. The metronome proper is based on IC_1 and IC_2, while the Wien bridge oscillator, based on IC_3, provides the tuning frequency.

440 Hz generator

The push-pull output stage can be switched from the metronome proper to the output of frequency generator IC_3. This is a classical Wien bridge oscillator whose output level is set with P_2. The type of op amp used is not important: almost any type, even a 741, will do.

Power supply

Power is supplied by a 9-V dry battery. Switch S_3 is the on/off selector. Switch section S_{2b} arranges power to be supplied to oscillator IC_3 when the unit is switched to 'tuning'.

With average use, the battery will last quite a long time. The metronome circuit draws a current of about 8 mA and the frequency generator one of around 15 mA.

CONSTRUCTION

The metronome/generator is best built on the printed-circuit board shown in **Figure 2**, which is, however, not available ready made.

Rotary switch S_1 and preset P_1 may be fitted directly on the board.

The space at the right-hand side of the board is intended to house the

2

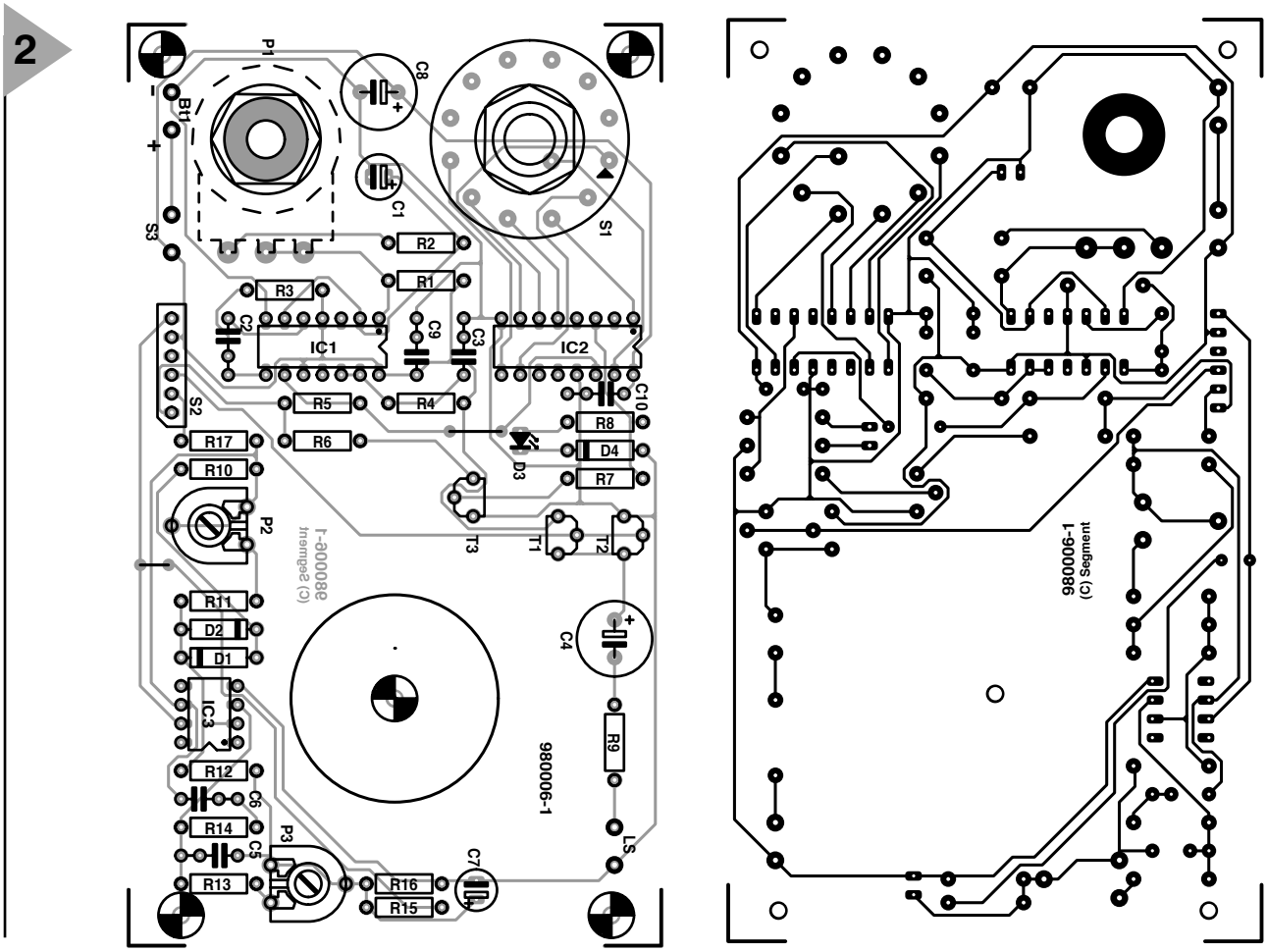


Figure 2. The printed-circuit board for the metronome is designed to accommodate the potentiometer and the rotary switch.

Parts list

Resistors:
R_1, R_8 = 1 kΩ
R_2 = 1.8 kΩ
R_3, R_4, R_{15}, R_{16} = 10 kΩ
R_5, R_6, R_{17} = 3.3 kΩ
R_7 = 33 kΩ
R_9 = 22 Ω
R_{10} = 220 kΩ
R_{11} = 27 kΩ
R_{12} = 15 kΩ
R_{13}, R_{14} = 39 kΩ
P_1 = 10 kΩ, linear
P_2, P_3 = 10 kΩ preset

Capacitors:
C_1 = 47 µF, 25 V
C_2, C_5, C_6 = 0.01 µF
C_4 = 220 µF, 16 V, radial
C_7 = 10 µF, 16 V, radial
C_8 = 470 µF, 16 V, radial

Semiconductors:
D_1, D_2 = 1N4148
D_3 = LED
D_4 = zener diode, 5.1 V, 400 mW
T_1 = BC337
T_2 = BC327
T_3 = BC560

Integrated circuits:
IC_1 = TLC556
IC_2 = 4017
IC_3 = LF351

Miscellaneous:
S_1 = 1-pole, 12-position rotary switch for board mounting
S_2 = double-pole change-over switch
S_3 = single-pole switch
BT_1 = 9-V battery with connecting clips
LS_1 = loudspeaker, 8 Ω
Enclosure, 150×80×55 mm

loudspeaker. If desired, a suitable hole can be cut in the space for the loudspeaker magnet.

The finished board and battery are best housed in a suitable enclosure, such as the one specified.

TUNING

Tuning the frequency generator with P_3 to 440 Hz is, of course, best done with the aid of a suitable frequency meter. If such an instrument is not available, the A-note struck on a well-tuned piano may be used to compare the generator output with (this needs a critical ear, of course!).

FINALLY

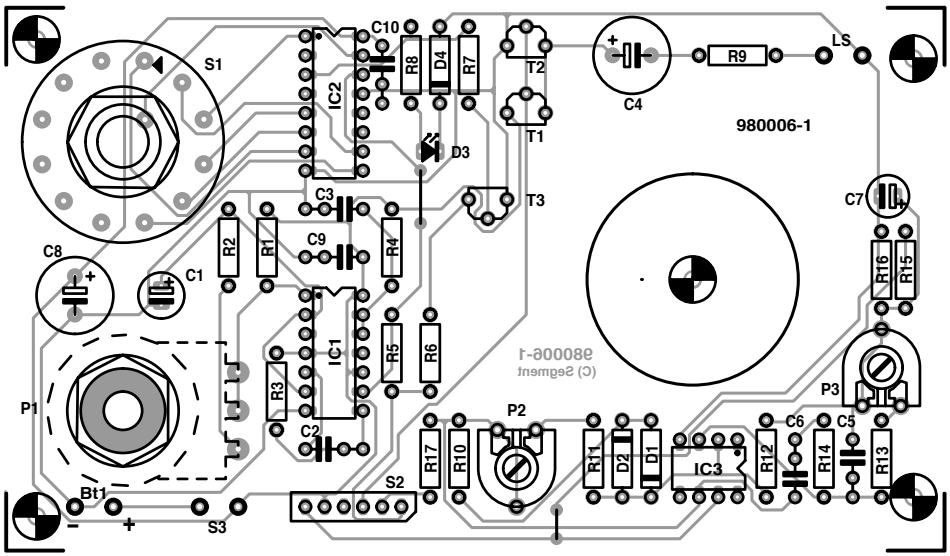
Bear in mind when selecting a value for R_9 and setting P_2 that the sound from the loudspeaker is much louder in the enclosure than when it lies on the table! [980006]

Corrections & Updates

Simple Electronic Metronome

February 1998 — 980006-1

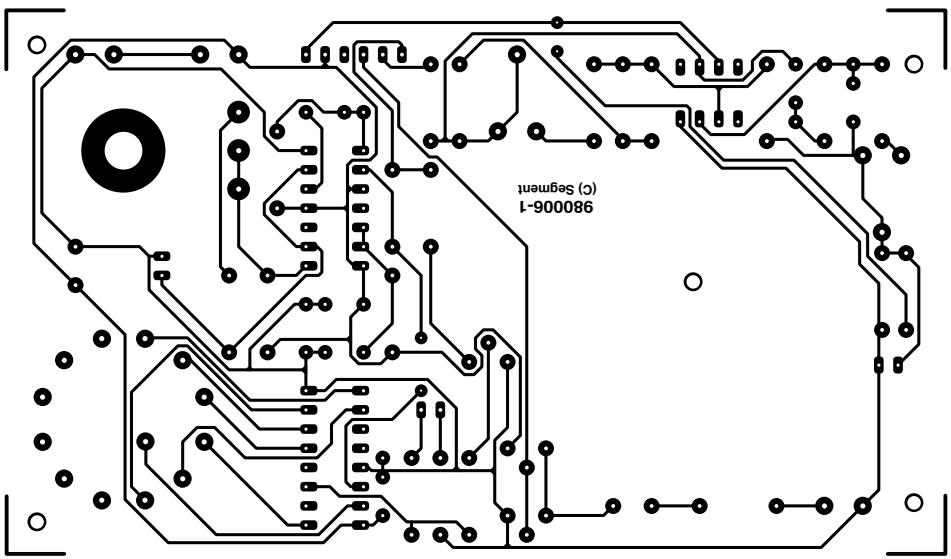
The PCB artwork for this project was not properly reproduced by our image setter. The correct drawings are given here.



Prize Contest

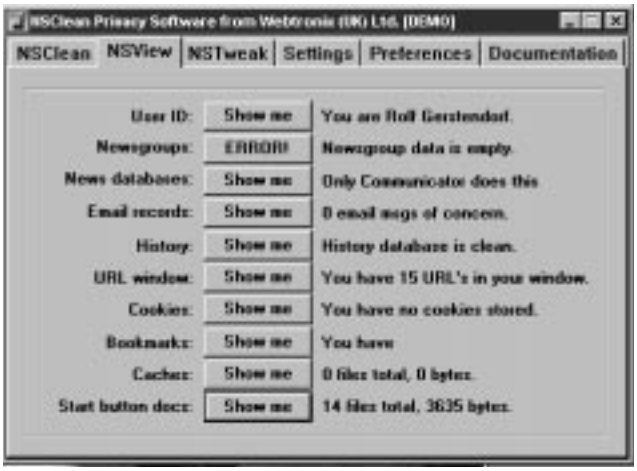
February 1998, page 9.

We apologize for a few errors in the final instalment of the 'you can Win this magnificent prize' contest on page 9 of our February 1998 issue. At the right-hand bottom of this page it is stated: '... the correct answer to question 3 and stick ...'; this should have read: '... the correct answer to question 4 ...'. Also, the months on the four stickers were incorrect: they should have read 'February 1998' (Ionization circuit: March 1998) and not January 1998. Obviously, these stickers will be accepted as if they had carried the correct month.



no more peeping Toms on the Internet

New and very unwelcome on the Internet are traders who secretly collect personal data on web users. The tiniest amount of personal information seems to be sufficient these days to enable these peeping Toms to reach final conclusions about your interests and preferences. Despite massive protests of data protection authorities, there is brisk trade in databases compiled in this way. As a result, the unwitting web user is bombarded with megabytes of unsolicited information (junk mail) from less reputable Internet traders.



Browsing the Internet invariably leaves traces on your own PC as well as on the network server of your ISP (Internet Service Provider). For example, each site you visit is recorded in the History file, and you make your e-mail address known to the server with each download. Along with this address comes a lot of personal data that belongs with your account. Web users will create bookmarks, compile URL lists and 'favourites' folders. An increasingly popular way of collecting data is by means of cookies. These are small files containing individual data, which are stored on the web user's PC for easy retrieval by Internet traders. Mind you, cookies can be very useful, for instance, to keep a permanent record of, say, your client number with a certain supplier, or other salient information. In this way, cookies avoid the hassle of typing personal data over and over again as you come back often to a certain web site.

Unfortunately, this information also lends itself to illegitimate use, and there is a disturbing increase in the number of cases in which this has actually happened. Although all modern web browsers allow users to reject cookies, actually doing so is annoying because the repeated warnings tend to slow you down when time is at a premium (always remember your phone bill). In some cases, it is even impossible to access a web site without accepting a cookie.

UK-based Internet provider Webtronix now offers a program called NSClean (for Netscape) or IEClean (for Internet Explorer) which records all access to the various memories (hard disk and cache) as you surf the Internet, and destroys all traces again the moment you log off. The home page of Webtronix may be found at http://www.webtronix.co.uk/wsc.htm. The programs are capable of detecting the various targets areas (History folder, URL window, cookies, bookmarks, etc.), allowing targeted clearing or retaining of individual areas (including individual entries).

Another interesting feature of NSClean and IEClean is the in-built alias function. At the press of a button, you can change your e-mail address into a fictitious one, for as long as the PC is on-line. In this way, you do not (unwittingly) disclose your identity when accessing an ftp site.

NSClean and IEClean cost £29.95 for the 16-bit Win3.x version, or £34.95 for the 32-bit Windows95/NT versions. Credit card owners may download the programs straight away. Obtaining this software by ordinary mail is also possible, P&P is then £5.

(985013-1)

introduction to digital signal processing

Part 2
Sampling and digital filters

After last month's brief excursion into sampling, this instalment takes a look at the effects sampling has and then takes the first steps into digital filter technology.

Shannon's sampling theorem (1949) states that, in a pulse-coded system, two samples per cycle will completely characterize a band-limited signal, that is, the sampling rate must be twice the highest-frequency component. In practice, the sampling rate is at least five times the highest frequency.

It will be seen that the theorem is invalidated when the highest frequency is more than half the sampling rate.

SOUND PROGRAMME

MUSICG1 <return> generates a sound programme of 60 single tones, each separated from the next and preceding one by a semitone. The lowest frequency is 40 Hz and the highest 14 kHz, that is, the programme spans a range of more than five octaves. The tones are sampled by MUSIC1.WAV at a rate of 44.1 kHz. The conditions of the theorem are fulfilled, which is verified by a good audible sound.

UNDERSAMPLING

If the sound generated in the previous paragraph is sampled at a rate of only 11,025 kHz, that is 1/4 of the original rate, we speak of undersampling (see **Figure 3**). It is executed by DWNSMPL1.EXE, whereby MUSIC1.WAV is converted to a file MUSIC2.WAV. The undersampling factor is indicated by DWNSMPL1\inp = MUSIC1.WAV\ out = MUSIC2.WAV\factor= 4 <return> . This means that only tones lower than 11,025/2= 5512.5 Hz can be reconstructed properly. Sampling of the higher-frequency tones results in a phenomenon known as aliasing. This gives rise to a tone erroneously taking on the identity of an entirely different frequency when recovered.

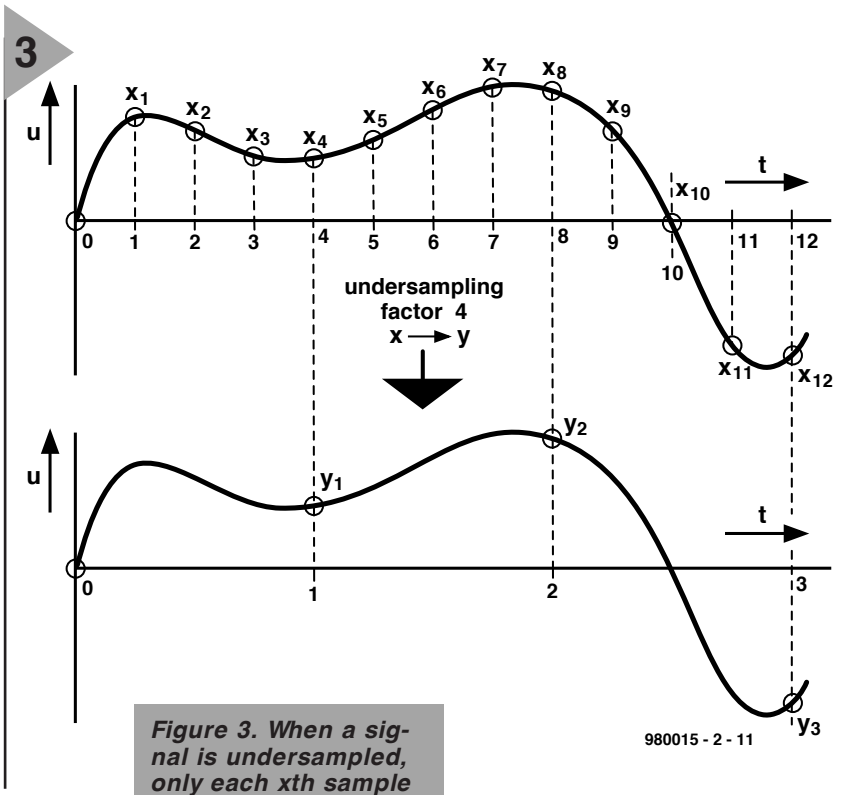


Figure 3. When a signal is undersampled, only each xth sample is taken.

ALIASING FREQUENCIES

Aliasing frequencies are not random: they can be accurately predicted. If, for instance, a sinusoidal signal of frequency $f_0 < f_s/2$ is sampled at a rate equal to f_s, definite sample values are obtained. Each signal of frequency $m(f_s - f_0)$ or $m(f_s + f_0)$, where $m = 1, 2, 3, \ldots$, generates the same sample values associated with f_0 – see **Figure 4**.

After sampling has taken place, these frequencies cannot be distinguished from one another. To prevent

Aliasing

The effect of aliasing may be compared to that seen on a cinematographic or video film when a spoked wheel of a vehicle turns at such a speed that successive samples (frames of the film) catch the wheel at slightly earlier or later positions. Between one frame and the next a spoke turns to almost the same position as formerly occupied by an adjacent spoke. The result is to make the wheel appear to rotate much more slowly or even bakcwards.

this situation arising, an analogue-to-digital converter (ADC) is usually preceded by a low-pass filter that suppresses the aliasing frequencies.

LOW-PASS FILTER

The low-pass filter used to suppress aliasing frequencies must be a digital type. Digital filtering is completely different from analogue filtering. Analogue filtering processes signals in the frequency domain, whereas digital filtering does so in the time domain. So, if a certain frequency domain response is required, it is necessary to convert this response into the equivalent time domain. So, let us see what happens when we try to use an analogue filter (see **Figures 5** and **6**).

During a sampling interval $\Delta T = t_{k+1} - t_k$, input voltage u changes but little, but attains the value u_k. The output voltage will not change much either, so that an almost constant current $i = (u_k - v_k)/R$ flows through resistor R. At the onset of the sampling interval, the potential across the capacitor is v_k. It is charged by i during the sampling period to attain the potential

$$v_{k+1} = v_k + i\Delta t/C = v_k + (v_k - u_k)/RC\Delta t.$$

If we solve this for u_{k+1}, we obtain

$$v_{k+1} = rv_k + u_k(1-r), \text{ where } r = 1-(\Delta t/RC.$$

This is the calculation prescription for the first digital filter. The program is on the CD-ROM under the title LP1.EXE; the source code, LP1.PAS is given in **Figure 7**.

The example shows that not all programs for digital signal processing need to be long and tedious. It is, of course, admitted that most of the routine work has been done by program library SIGLIB.PAS.

The filter is tested by processing file MUS1.WAV (also listen to it!) on the CD-ROM. This is done by calling up

1p1 \r= 0.995 \scale= 10 \inp= mus1.wav \out= tmp.wav < return> .

Listening to the resulting signal makes the distinction between tmp.wav and the original signal very clear.

Some experiments may be carried out with the filter. For instance, try out several values for r, but note that this should not be greater than unity to prevent the filter becoming unstable.

Just listening to a filter's performance is, of course, not the best test. For a proper test, a couple of test signals are needed to analyse, say, the frequency response in the time and frequency domains. These signals may be derived from a pulse generator.

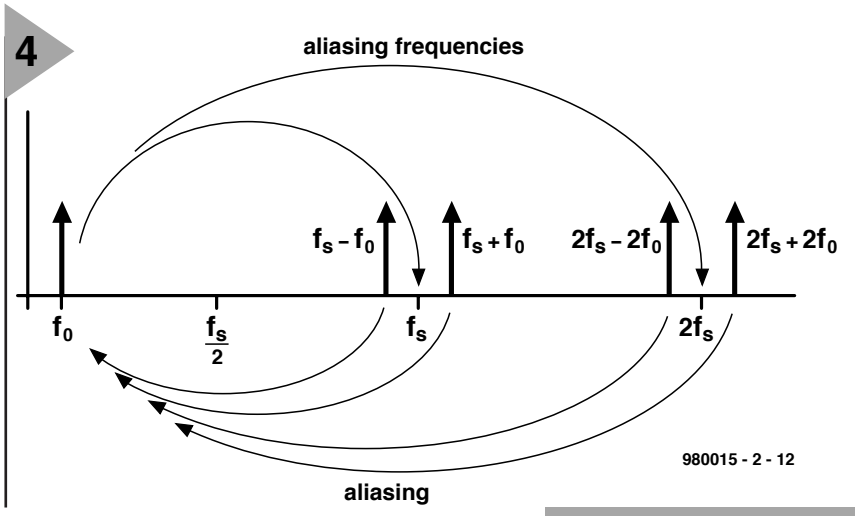


Figure 4. Aliasing causes equivalent frequencies to be produced from dissimilar samples.

PULSE GENERATORS

The CD-ROM contains a couple of pulse generators. The first and simpler is indicated by PULSE1.EXE. This generates a very brief pulse that has only one sampling value which is not zero: the value of all others is zero. The position and amplitude of the pulses can be set by relevant parameters. This elementary signal is very important and will be used frequently in experiments later in this series of articles.

Another pulse generator is found under STEP1.EXE. This generates a simple step signal with predestined amplitude and position. For instance, calling up DO XLP1.SPP generates the signals shown in **Figure 8**. That at the top represents the reaction of filter tmp.wav to pulse1.wave, and that at the bottom, tmp.wav, the slowly rising response to step1.wav.

As before, this experiment may be repeated with various values of r. For instance, what happens when $r = -0.9$? However, this kind of response does not give a very clear picture of the performance of the low-pass filter. A better one is obtained by the use of a sweep generator, which is also available on the CD-ROM.

When DO XLP2.SPP < return> is called up, a sweep signal of 1–1000 Hz is generated (top of **Figure 9**) and applied to the low-pass filter. The amplitude of the output signal (at the bottom) drops with rising frequency.

Another possibility is to apply white noise to the filter and view the output signal. But for that purpose, it is necessary to first analyse and examine signal spectra.

Figure 5. An RC network used as an analogue low-pass filter.

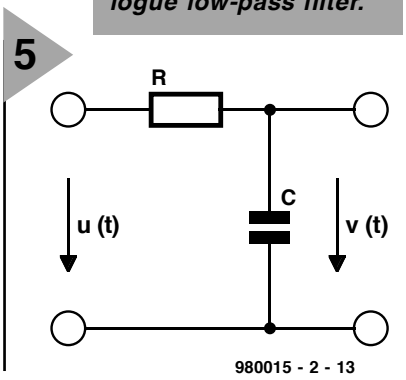
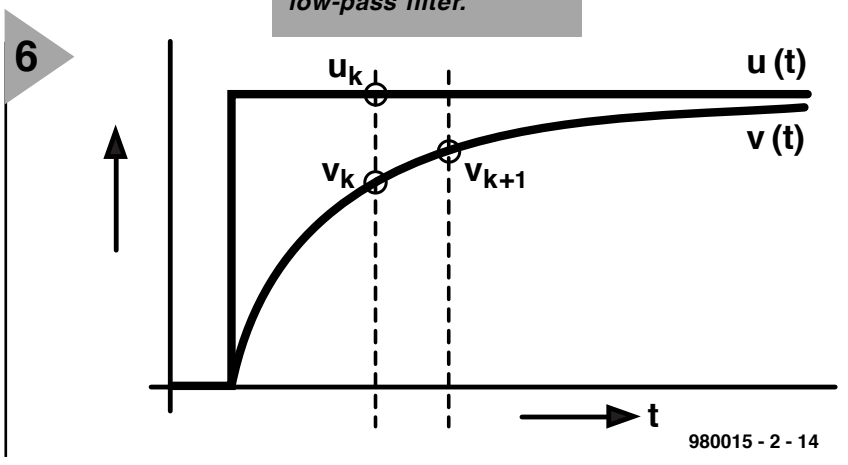


Figure 6. Step response of the digital low-pass filter.



```
001 program lp1 ;
002 uses dos,crt,graph ;
003
004 {$I SIGLIB.PAS }
005
006 var k:int ;
007     y,scale,r:float ;
008
009 begin
010 start('simple lowpass') ;
011 scale:=1.0              ; setparreal('\scale=',scale) ;
012 r:=0.95                 ; setparreal('\r=',r) ;
013 inpfn:='pulse1.wav'     ; setparstring('\inp=',inpfn) ;
014 outfn:='tmp.wav'        ; setparstring('\out=',outfn) ;
015
016 openinp(inpfn) ;
017 openout(outfn) ;
018
019 y:=0 ;
020 for k:=1 to nsamples do
021  begin
022  output(scale*y) ;
023  y:=r*y+(1-r)*input ;
024  if (k mod 2000)=0 then write('.') ;
025  end ;
026 stop ;
026 end.
```

Lines 1–5 form the head of the program and declare that program library SIGLIB.PAS will be used.

Lines 6–7 stipulate the requisite variables when line 10 calls up the initialization of SIGLIB.PAS

Lines 11–14 determine the parameters; if required (owing to procedures in SIGLIB.PAS) the actual parameters called for by the program.

Lines 16–17 actuate the data input and output of the WAV files.

Line 19 initializes the value of the filter. The procedure in lines 20–25 executes the actual filter operation.

In line 22, the actual output value of the filter is written in the output data file.

Line 23 contains the actual filter algorithm.

Line 26 closes all files and thus the program.

Figure 7. Pascal source code for the digital filter.

SPECTRUM ANALYSER

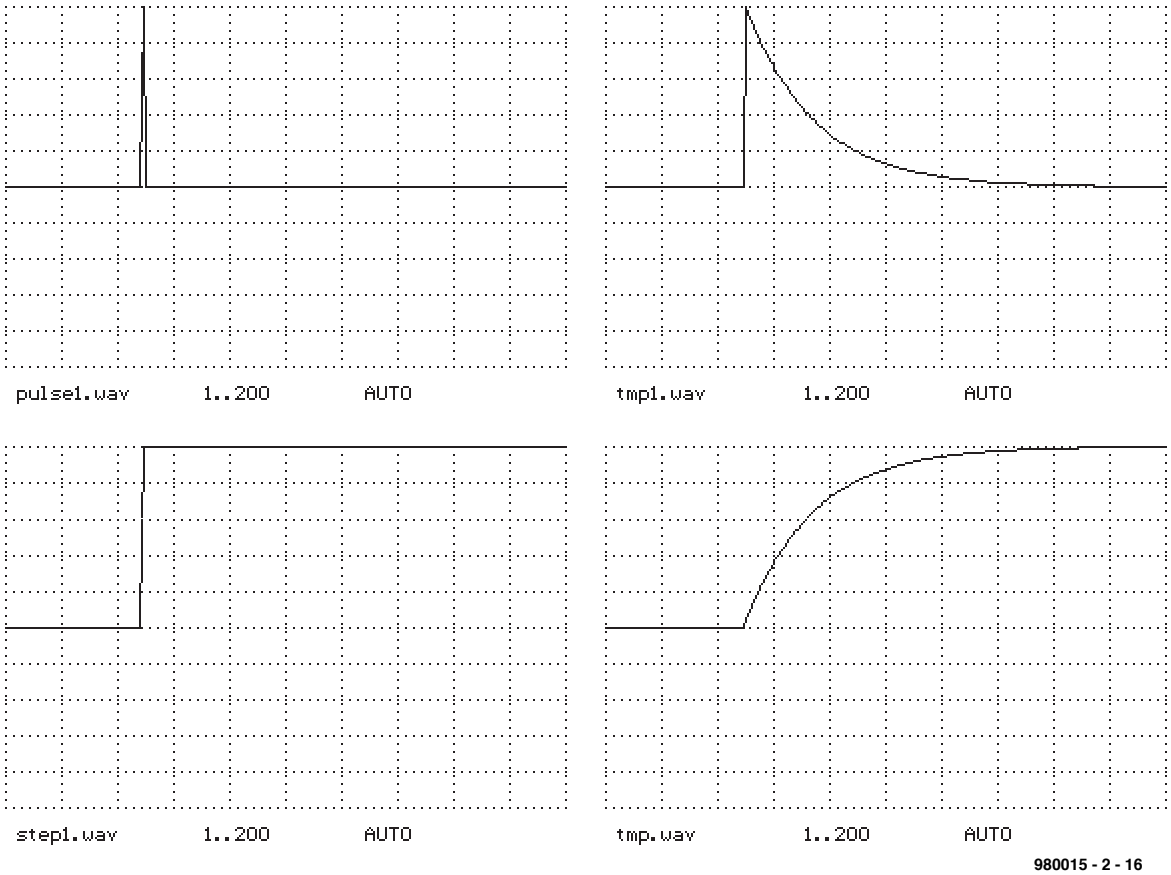
An introduction to DSP would not be of great value without a facility to view, examine and analyse signal spectra. Therefore, the CD-ROM contains program SPEC1.EXE (source code SPEC1.PAS) which enables the spectra of wave files to be calculated and viewed.

To calculate a spectrum, SPEC1.EXE uses a discrete Fourier transform (DFT), which derives $n/2$ amplitudes that are associated with frequencies 0 Hz (d.c.) to $f_s/2$ from n values In the present program, n= 4096. The amplitude levels are shown on a logarithmic scale to obtain a wider range of values.

Experiment XSPEC1.SPP generates two signals of frequency 193.7988 Hz (tmp1.wav) and 196.4905 Hz (tmp2 .wav) respectively from 4096 samples at a rate of 44.100 samples/s. If the DFT is applied to these signals, the spectra shown in **Figure 10** are obtained.

It appears as if the first signal (tmp1.wav – top left) consists of only

Figure 8. Pulse and step response of the low-pass filter.



9

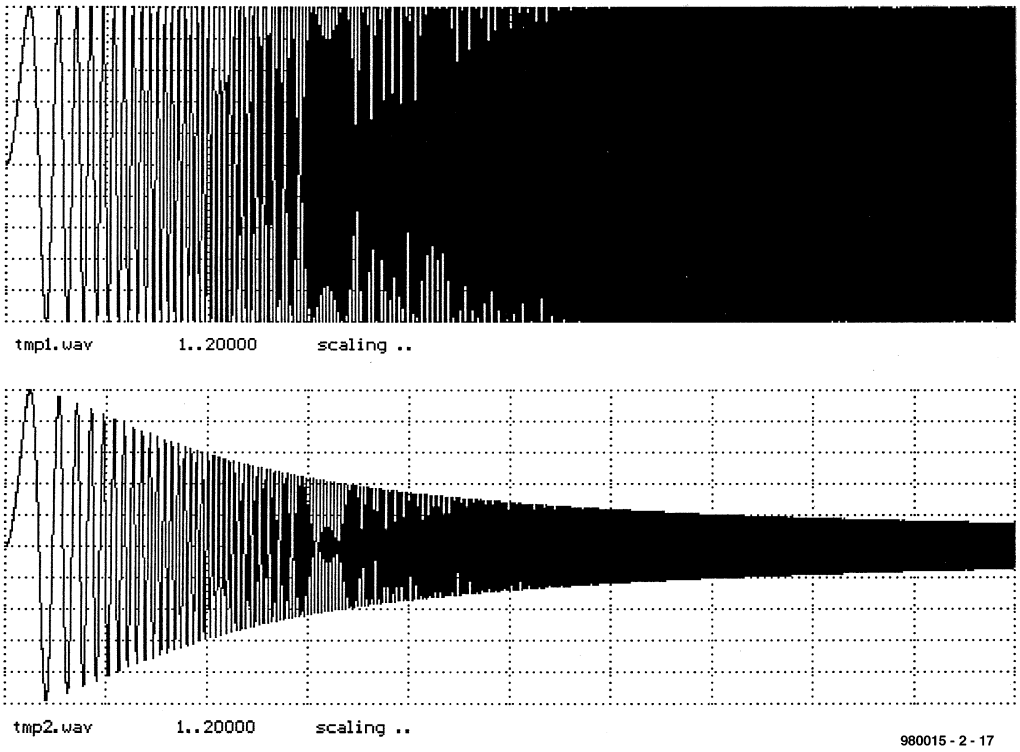


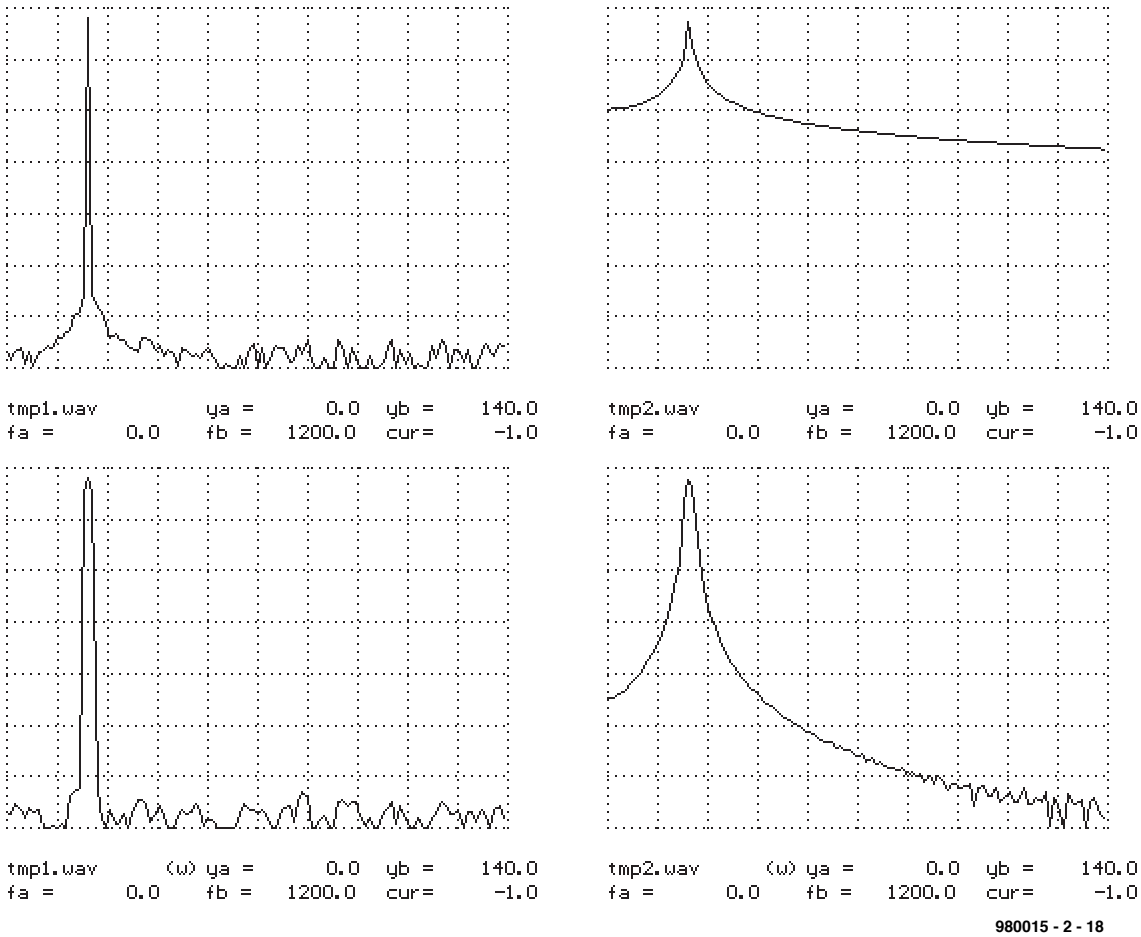
Figure 9. Passage of a signal from a sweep generator through a low-pass filter.

Figure 10. Spectra of sinusoidal signals: at the top without the window function, and at the bottom with it.

one frequency. On the other hand, the second signal (tmp2.wav – top right) has a 20 dB peak, but the spectrum at either side does not drop off very much.

How does the DFT conclude that there are some of a great many frequencies in the signal? The answer lies in **Figure 11**, which shows that in the case of signal tmpl.wav a whole number of complete cycles fits exactly in the region enclosed by the 4096 sampling points. This is not so in case of the other signal. The value of

10



11

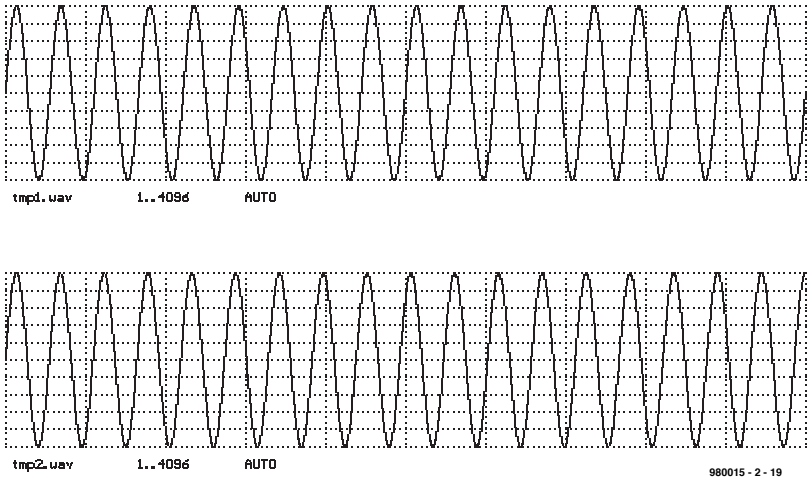


Figure 11. Signals may fit well or not so well in the window of the spectrum analyser.

this signal at the left is 0, while at the right 18.5 cycles fit in the 4096-point region. Because of this, all possible oscillations are required (and thus shown up by the DFT) to display the signal.

WINDOW FUNCTION

Obviously, the situation in the previous paragraph is not satisfactory, since the DFT should generate a line-shaped spectrum of signal tmp2.wav also. This is obtained when the input signal is multiplied (top of **Figure 12**) by a window function (at the centre). This signal then undergoes a DFT and results in that displayed at the bottom.

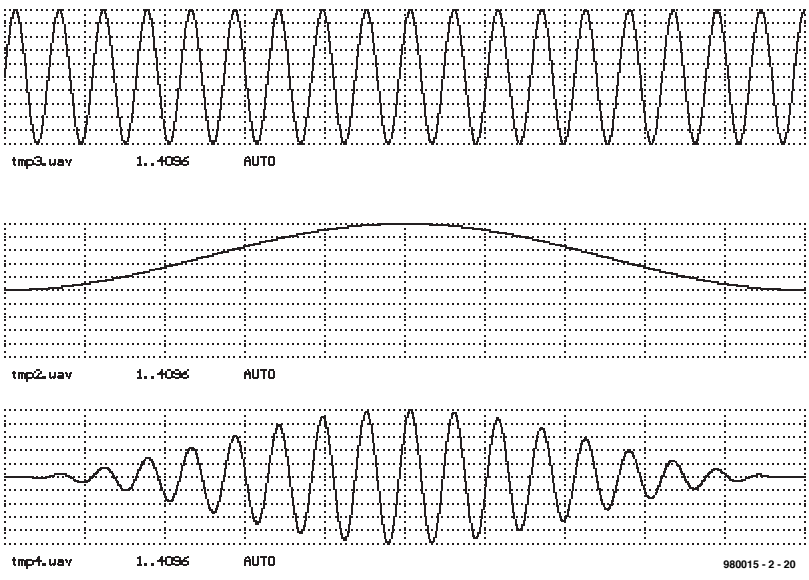
The window function ensures that the signal becomes compatible with the DFT. The spectra of the signals subjected to the window function are shown at the bottom of Figure 10. It will be seen that the spectrum of tmp2.wav is now clearly line-shaped, but that the line produced by tmp1.wav has broadened slightly – this is the price to be paid for the window function. The spectrum analyser on the CD-ROM gives a choice of working with or without the window function. The window function will be met again when filter synthesis will be discussed.

Next month's instalment will continue with the subject of filtering and also deal with the spectrum analysis of some real signals.

[980015-2]

Figure 12. Windowing is simply the multiplication of the signal with a window function.

12



APPLICATION NOTE

AM/FM antenna impedance matching IC

The U4253BM and U4254BM ICs are integrated AM/FM antenna matching circuits in bi-CMOS* technology. They are intended particularly for car applications and may be used with windscreen, roof and bumper antennas. The U4254BM chip has a lower noise figure than the U4253BM and a different AM amplifier stage with two outputs. Apart from the additional AM output (pin 11 which is not connected in the U4253BM), the pinouts of the two devices are identical.

** Integrated circuits that contain both bipolar and CMOS transistors. This combination allows the advantages of both processes to be exploited.*

A TEMIC Semiconductors Application

FUNCTIONAL DESCRIPTION

The U4253BM/U5254BM impedance matching circuit (see block diagram in **Figure 1**) compensates for cable cable losses between the antenna and the car radio which is usually placed far away from the antenna.

The FM amplifier provides excellent noise performance. External components are used to adjust the gain and

1

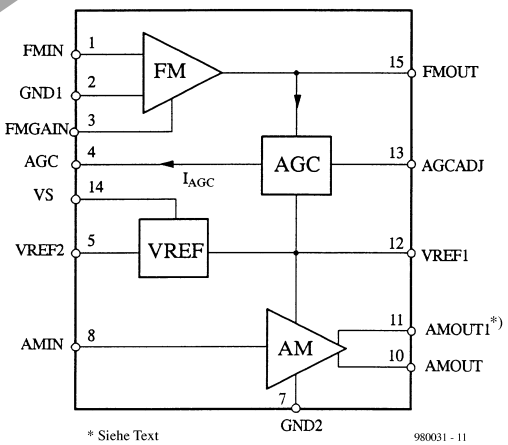


Figure 1. Block diagram of the AM/FM antenna impedance matching IC.

the input-output matching impedance. Therefore, it is possible to adjust the amplifier to various impedances (usually 50, 75 or 150 Ω).

To protect the amplifier against input overload, an Automatic Gain Control (AGC) is included on the chip. The AGC observes the a.c. voltage at the FM amplifier output, rectifies this signal, and delivers direct current to dampen the input antenna signal via an external p-i-n diode.

The threshold for the AGC is adjustable. Simple and temperature-compensated biasing is possible owing to the integrated voltage reference V_{Ref2}.

The AM part consists of a buffer amplifier. The voltage gain of this stage is about unity. The input resistance is 470 kΩ and the input capacitance less than 10 pF. The output resistance is 125 Ω. An excellent dynamic range is achieved owing to the CMOS source follower stage.

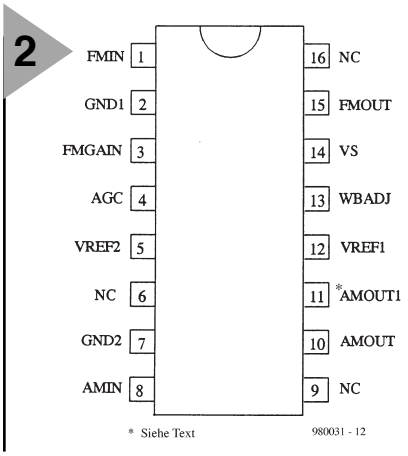


Figure 2. Pinout of the AM/FM antenna impedance matching IC.

PIN DESCRIPTION

The pinout is shown in **Figure 2**. Note that pins 6, 9, 11 (U4253BM only), and 16 are not connected.

FMIN (pin 1) (**Figure 3a**)
FMIN is the input of the FM amplifier. It is the base of a bipolar transistor. A resistor or a coil is connected between FMIN and VREF2.

GND1 (pin 2)
To avoid crosstalk between AM and FM signals, the circuit has two separate ground pins. GND1 is the ground for the FM part.

FMGAIN (pin 3) (**Figure 3b**)
The direct current of the FM amplifier transistor is adjusted by an external resistor which is connected between FMGAIN and GND1. To influence the a.c. gain of the amplifier, a resistor is connected in series with a capacitor between FMGAIN and GND1. The capacitor has to be a short at frequencies ≥ 100 MHz.

AGC (pin 4)
Direct current flows into the AGC pin at high FM antenna input signals. This current has to be amplified via the current gain of an external p-n-p transistor that feeds a p-i-n diode. This diode dampens the antenna input signal and

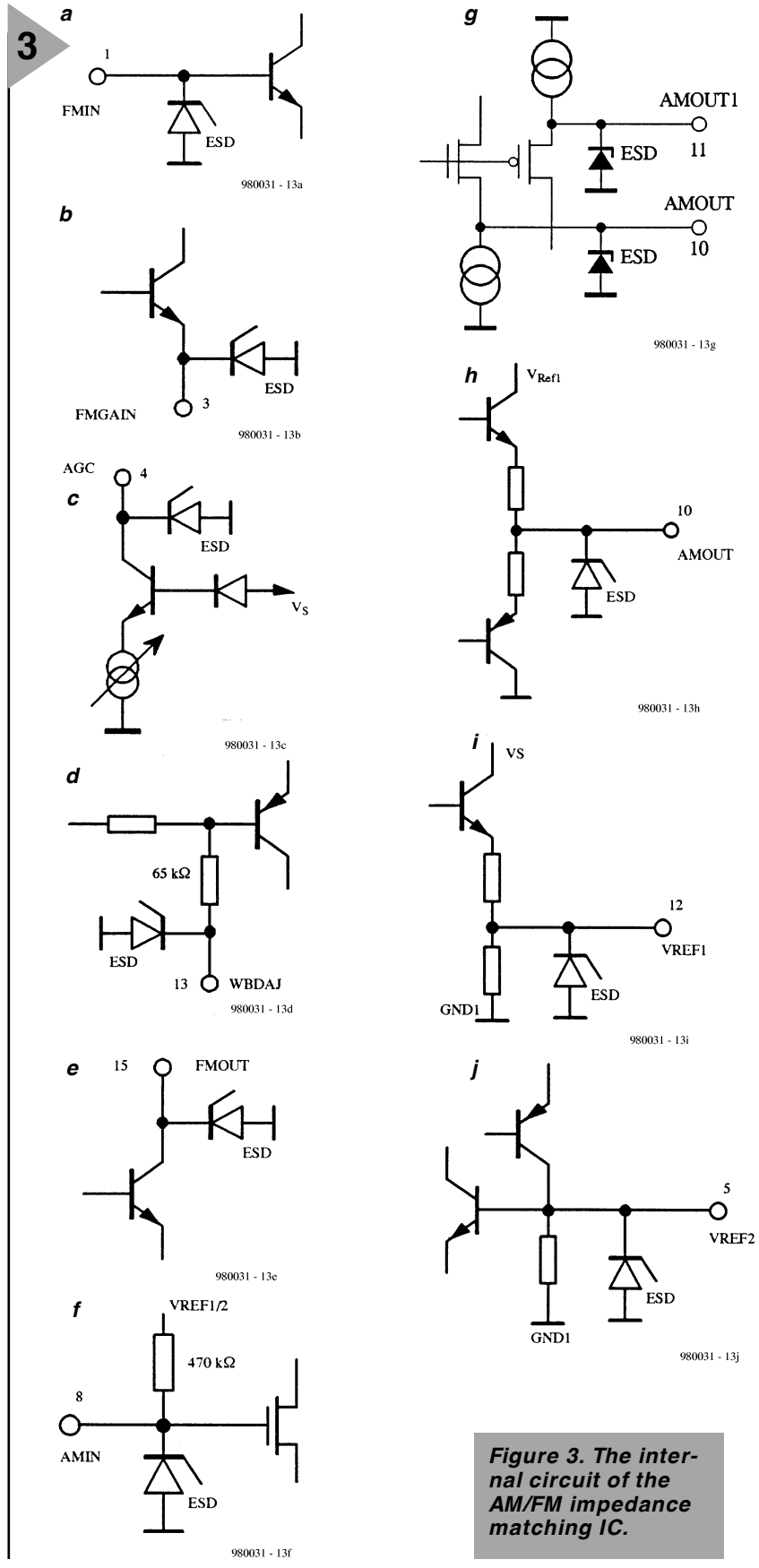


Figure 3. The internal circuit of the AM/FM impedance matching IC.

4

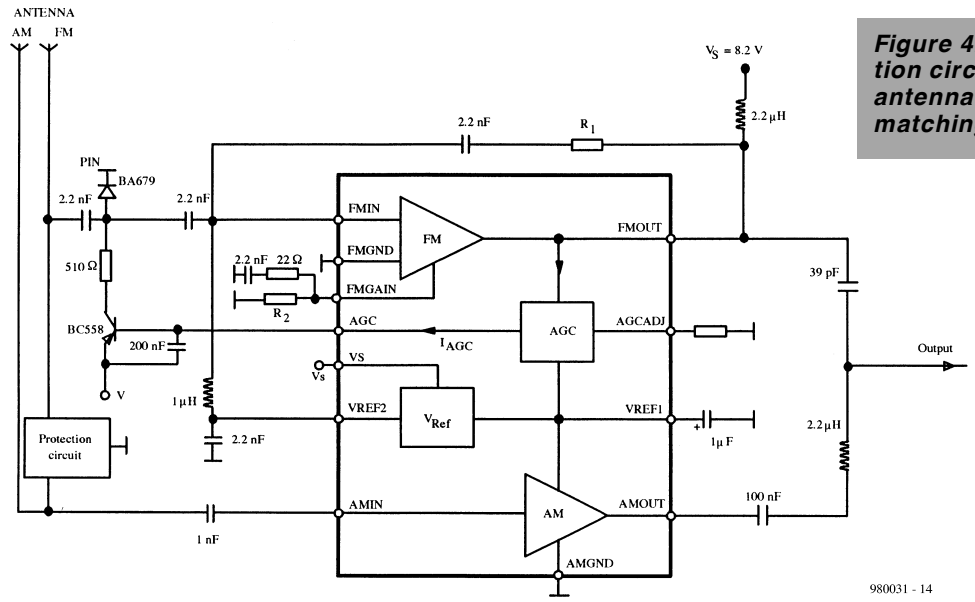


Figure 4. Typical application circuit of the AM/FM antenna impedance matching IC.

Table 1. Electrical Characteristics		V_S = 8 V, T_{amb} = 25°C, unless otherwise specified				
Parameters	***Test Conditions / Pins***	***Symbol***	***Min.***	***Typ.***	***Max.***	***Unit***
Supply voltage	*Pin 14*	V_s	*7.2*	*8*	*8.8*	*V*
Supply currents	*Pin 14*	I_s		*4*		*mA*
Reference voltage 1 output, (I_{12} = 0)	*Pin 12*	V_{Ref1}		*5.5*		*V*
Reference voltage 2 output, (I_5 = 0)	*Pin 5*	V_{Ref2}		*2.6*		*V*
Temperature dependence of V_{REF2}	*Pin 5*	$V_{ref2}/\Delta T$		*–1*		*mV/K*
AM amplifier						
Input resistance	*Pin 8*	R_{AMIN}		*470*		*kΩ*
Input capacitance	*Pin 8*	C_{AMIN}			*10*	*pF*
Output resistance	*Pin 10*	R_{OUT}		*200**		*Ω*
Voltage gain	*Pin 10 / Pin 8*	*a*		*0,85*		
Output noise voltage (rms value)	*Pin 10, pin 8 to ground via 15pF; B = 6 kHz, 150 kHz to 300 kHz, 500 kHz to 6.5 MHz*	V_{N1} V_{N2}		*–2* *–6*		*dBµV* *dBµV*
2nd harmonic	*Pin 10, pin 8 to ground via 15pF; f_{AMIN} = 500 kHz, Output voltage = 110 dBµV*		*–60***			*dBc*
FM amplifier						
Supply current limit	*I_{AGC}, I_{AGCADJ} = 0 A, Pin 15*	I_{15}		*33*	*35*	*mA*
Input resistance	*f = 100 MHz Pin 1*	R_{FMIN}		*50*		*Ω*
Output resistance	*f = 100 MHz Pin 15*	R_{FMOUT}		*50*		*Ω*
Power gain	*f = 100 MHz Pin 15/ Pin 1*	*G*		*5*		*dB*
Output noise voltage	*Pin 15 f = 100 MHz, B = 120 kHz*	V_N		*0*		*dBµV*
3rd order output intercept	*f = 100 MHz Pin 15*			*132*		*dBµV*
AGC						
AGC input voltage threshold	*f = 100 MHz Pin 15 pin 13 grounded; AGC threshold DC current is 10 µA at Pin 4*	V_{th1}		*96*		*dBµV*
AGC input voltage threshold	*f = 100 MHz Pin 15 pin 13 not connected; AGC threshold DC current is 10 µA at Pin 4*	V_{th2}		*106*		*dBµV*
AGC output current	*AGC active*	I_{AGC}			*1.2*	*mA*

* 125Ω for U4254BM

** –65dBc for U4254BM

protects the amplifier input against overload. The maximum current which flows into the AGC pin is about 1 mA. In low-end applications, the AGC function is not necessary and the external components can then be omitted.

VREF2 (pin 5) (**Figure 3j**)
For the d.c. biasing of the FM amplifier, a second voltage reference circuit is integrated. Because of temperature independence of the collector current, the output voltage has a negative temperature coefficient of about $-1\ mV\ K^{-1}$. To stabilize this voltage an external capacitor of a few nF to ground is recommended.

GND 2 (pin 7)
This pin is the ground connection for the AM amplifier.

AMIN (pin 8) (**Figure 3f**)
The AM input has an internal bias voltage. The direct voltage at this pin is $V_{Ref1}/2$. The input resistance is about 470 kΩ and the input capacitance is smaller than 10 pF.

AMOUT (U4253BM only)
(pin 10) (**Figure 3h**)
This pin is the output of the AM amplifier in the U4253BM. The direct voltage at this pin is almost $V_{Ref1}/2$. The output impedance is about 200 Ω. The output capacitance is smaller than 10 pF.

AMOUT and **AMOUT1** (U4254BM)
(Pins 10 and 11) (**Figure 3g**)
The buffered AM amplifier consists of a complementary pair of CMOS source followers. The transistor gates are connected to AMIN. Pin 10 is the NMOS transistor's source, whereas pin 11 is the PMOS transistor's source. Owing to the two different direct voltage levels at these pins, they have to be connected together via an external capacitor of about 0.1 µF. This technique enables an excellent dynamic range to be achieved.

VREF1 (pin 12) (**Figure 3i**)
VREF1 is the stabilized voltage for the AM amplifier and the AGC block. To achieve good noise performance at LW frequencies, it is recommended that this pin is connected to ground via an external capacitor of about 1 µF.

WBADJ (U4254BM)
AGCADJ (U4253BM(
(pin 13) (**Figure 3d**)
The threshold of the AGC can be adjusted by varying the direct current at this pin. If the pin is connected directly to GND1, the threshold is set to 96 dB µV at the FM amplifier output. If a resistor is connected between this pin and GND1, the threshold is shifted to higher values with increases resistance. If the pin is open, the threshold is set to 106 dB µV.

VS (pin 14)
This pin is linked to the positive supply voltage (7.2–8.8 V).

FMOUT (pin 15) (**Figure 3e**)
The FM amplifier output is the open collector of a bipolar r.f. transistor. It should be linked to VS via a coil.

TYPICAL APPLICATION

A typical application circuit is shown in **Figure 4**. In this, it is assumed that the antennas for reception of long-wave and medium-wave (LW/MW) signals and very-high-frequency (VHF) signals are of different lengths. The antennas are linked to the circuit via a protection circuit.

Whereas the AM antenna is linked directly to the input of the AM section via a 0.1 µF coupling capacitor, the FM antenna is connected to the FM section via a p-i-n diode attenuator. Before and after the attenuator coupling capacitors block any direct voltage.

The antenna impedance is matched to the input impedance of the FM section by resistors R_1 and R_2. The value of these resistors for various antenna impedances is given in **Table 4**.

The 1 µH choke between VREF2 and the input of the FM section prevents the antenna signal being short-circuited by the 0.0022 µF decoupling capacitor to ground.

The 2.2 µH choke following the 0.1 µF coupling capacitor at the AM output presents a very low impedance to AM signals, but a very high impedance to VHF signals.

[980031]

Table 2 – Features

- *High dynamic range for AM and FM*
- *Integrated AGC for FM*
- *High intercept point 3rd order for FM*
- *FM amplifier adjustable to various cable impedances*
- *High intercept point 2nd order for AM*
- *Low noise output voltage*
- *Low power consumption*

Table 3 – Absolute maximum ratings

Reference point is ground (pin 2 and 7)

Parameters	Symbol	Value	Unit
Supply voltage	V_S	8.8	V
Power dissipation, P_{tot} at T_{amb} = 85 °C	P_{tot}	460	mW
Junction temperature	T_j	150	°C
Ambient temperature	T_{amb}	–40 to + 85	°C
Storage temperature	T_{stg}	–50 to + 150	°C
Electrostatic handling	$\pm V_{ESD}$	2000	V

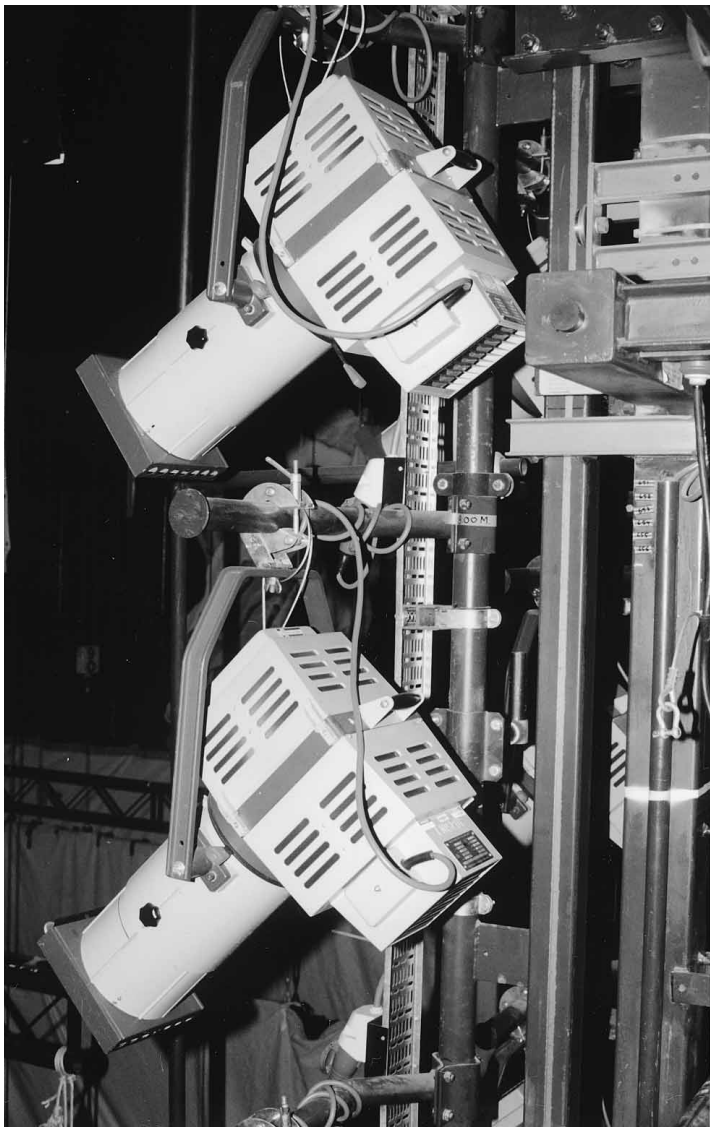
Table 4 – Cable impedance vs resistor values

FM cable impedance (Ω)	R_1 (Ω)	R_2 (Ω)
50	150	22
75	270	33
100	390	51
125	470	86
150	620	160

TEMIC U.K. Ltd.
Easthampstead Road
Bracknell, Berkshire RG12 1LX
United Kingdom
Tel: 44 1344 707 300
Fax: 44 1344 427 371

stage-lighting control

with DMX512 protocol



Before the computer age, it required quite a number of assistants to control the stage lighting in a theatre. Today, however, even here the computer has proved its usefulness as a tool. With the DMX512 (***D***igital ***M***ultiple***X*** for 512 units) standard, a computer can control the entire lighting system, including ancillary functions such as the colour filters and dimmers. In this setup, a simple interface cable enables the computer to control up to 512 separate lighting units.

By our Editorial Staff

Some of us theatregoers may remember the large theatre lights that were manually operated. Each light required at least a pair of hands – a labour-intensive and therefore costly affair. When electronic control units, and later the computer, became available, many theatres adopted analogue lighting control systems that were much simpler to operate, and therefore more cost-effective.

However, over the past ten years or so, digital control systems controlled by computer have become the norm. In most smaller theatres, the set of instructions developed for the United States Institute for Theatre Technology (USITT), code-named DMX-512, has been adopted. This is an efficient, yet simple, digital protocol, accepted in many parts of the world, which enables all aspects of the stage lighting to be controlled by a computer.

IN TIMES GONE BY...

In the early years of (analogue) electronic control units, an analogue signal was required for each control channel,

that is, each lighting function that had to be controlled. What's more, a separate cable or pair of wires in such a cable was needed for each of these functions. This cable had to distribute linear control voltages of 0–10 V according to an internationally accepted set of rules. Evidently, although this is a practical setup when there are not all that many lights involved, it becomes cumbersome and costly when many lights are to be controlled, because it results in many or very thick cables. However, for the technicians involved, the use of low voltages means that it is a safe system that can be checked with a simple multimeter.

With the increase of technical facilities whereby modern projectors fulfil more and more functions, the analogue system becomes more and more impracticable. Each piece of equipment needs several channels, which makes the cabling ever more complicated. Today, some lighting units provide 25–30 functions.

In larger theatres, these difficulties with analogue systems led in the early 1980s to the introduction of digital control systems, which in the mid-1980s resulted in the USITT adopting the digital DMX-512 standard. In this system, each pair of wires can control up to 512 functions. In practice, of course, the number is restricted to 32. However, each unit being controlled may use several channels at a time, so that a fair amount of the total available capacity is used.

Although the DMX-512 protocol is not new, it may not be known to many readers. Moreover, a successor is already being developed: the DMX-B. Fortunately, the new set of standards is backwards compatible with the DMX-512, so that older equipment remains usable.

EFFICIENT AND EFFECTIVE

The DMX-512 Standard was last modified and officially laid down as a norm by the USITT in 1990. It is based on the much more widely used RS485 Standard. A basic layout of a DMX-512 link between transmitter and receiver, both of which use the same driver, is shown in **Figure 1**. Interconnection is via a symmetrically controlled pair of wires.

The data are transmitted asynchronously serially over the wires. The settings are sent sequentially, that is, first the level for dimmer 1, then that for dimmer 2, and so on, until the levels for all connected dimmers (up to 512)

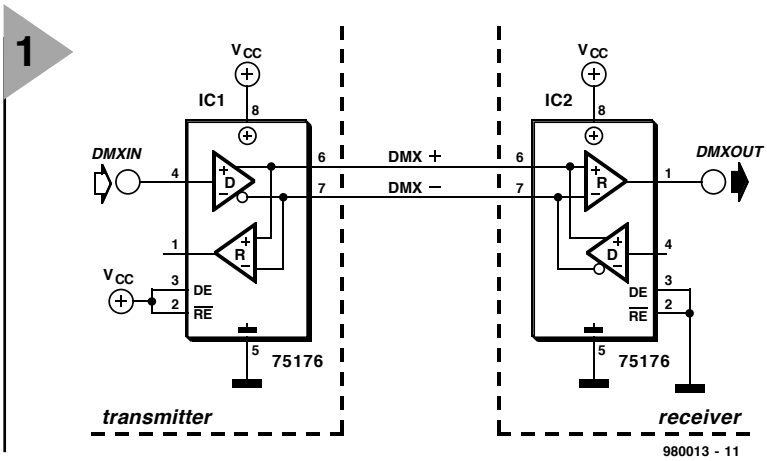
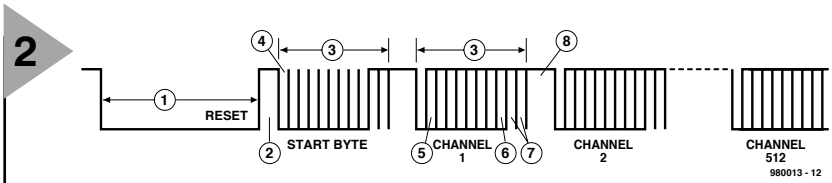


Figure 1. Basic layout of a simple transmitter and receiver used in the DMX-512 system. The driver is a standard driver for the RS485 system and is readily available.



no.	name	min.	typ.	max.	unit
1	reset		88	88	µs
2	mark	8	-	1	µs
3	frame	43.12	44.0	44.48	µs
4	start bit	3.92	4.0	4.08	µs
5	LSB	3.92	4.0	4.08	µs
6	MSB	3.92	4.0	4.08	µs
7	stop bit	3.92	4.0	4.08	µs
8	mark (between frames)	0	0	1.00	s
9	mark (between trains)	0	0	1.00	s

Figure 2. The timing diagram shows the maximum width of a data block with which up to 512 lighting units can be controlled. In this setup the repetition rate is 44.1 Hz.

How to make a terminator

Terminators are of great importance to ensure good and reliable communication, but, owing to their small size, are easily mislaid. It is, therefore, useful, to be able to make one at low cost.

Remove the hood from an XLR plug and solder a 120 Ω, 0.25 W resistor between pins 2 and 3, and replace the hood. That's all! The photograph shows a completed terminator.

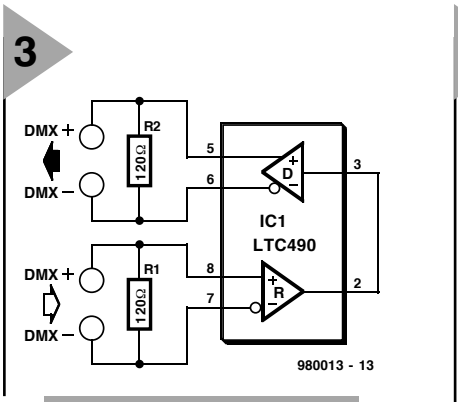


Figure 3. When lines longer than about 1 metre are needed, a bus repeater as shown will be essential.

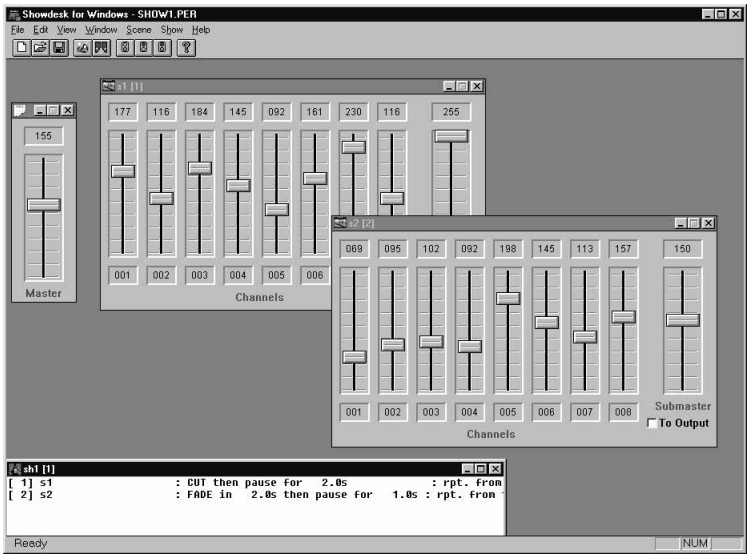


Figure 4. To get a better idea of the facilities provided by the DMX-512 standard, download a demo version of ShowDesk from web site http://www/fpfltd.com

have been sent.

The reset pulse is followed by a mark signal that indicates the onset of communication. Normally, the mark signal must be at least 8 μs long, but there are systems which are able to recognize a mark-to-break width of 4 μs; these are coded DMX-512/1990 (4 μs).

The onset of a cluster of bytes is marked by a reset signal that is followed by a start byte. This is followed by the brightness data for the first dimmer in the shape of an 8-bit value 0–255, that is, 00_H–FF_H. The relationship of this value with respect to the present brightness setting is a matter for the dimmer itself. For instance, the manufacturer of the dimmer may give it a control curve according to which its brightness increases or decreases. In the case of lamps whose brightness must change rapidly, use may be made of a shutter; the DMX-code then arranges the opening and closing of the shutter.

Each DMX-512 instruction consists of a start bit, eight data bits, and two stop bits (one frame). In the quiescent state, the level on the communication line is high (mark), whereas the active level is low (break or space). The break signal itself is not less than 88 μs wide, a time that corresponds to two frames. The system recognizes the break as a reset signal, whereupon all current operations are discontinued.

The next step is the transmitting of a number of n+1 frames, each of which contains the setting of one of the connected dimmers. The first transmitted frame, that is, the start byte, marks the onset of the series of commands and has a fixed value 00_H. This indicates that the settings refer to the dimmers. Because of this start code, standards that use a different start code for controlling other types of unit may be added at a later date. For this reason, the connected dimmers must not react when the communication is

Figure 5. A modern control room from where the lighting manager controls the stage lighting resembles a computer centre.

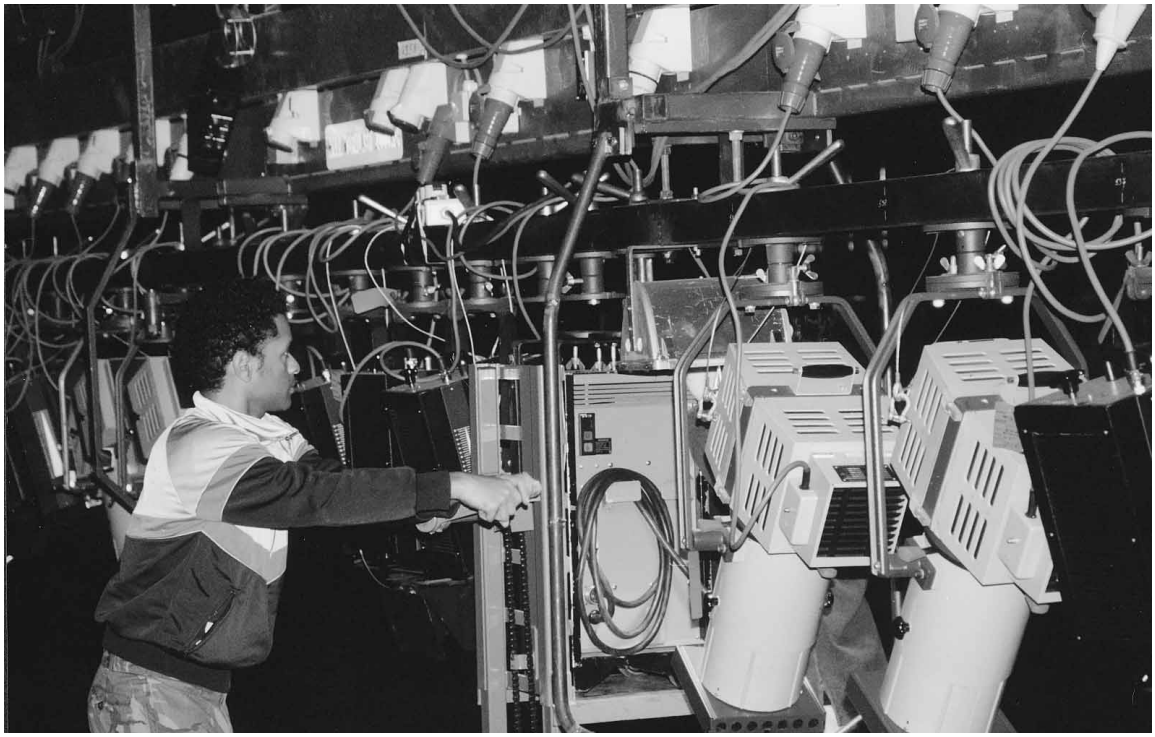


Figure 6. Modern theatre lights may have more than 25 functions. For each of these functions, the light uses a DMX address.

begun with a start code other than 00_H.

TIMING

As mentioned earlier, the DMX-512 standard supports up to 512 dimmers; a minimum is not stipulated. After the data has been transferred to the final dimmer in the chain, the data line returns to the quiescent state (mark). The next reset signal indicates that a new transfer of data is imminent. It is imperative that two sequential setting instructions are separated by an interval (pause) of not less than 1196 μs.

The data rate in the DMX-512 standard is 250 kbit/s. Since one bit lasts for 4 μs, a complete instruction, including the stop and start bits, takes 44 μs. The timing diagram in **Figure 2** shows a complete sequence of 512 bytes, the data stream required for the theoretical maximum of 512 dimmers. When all times are added, the maximum time duration is 22,668 μs, which corresponds to a repetition rate of 44.1 Hz. From this, it is clear that the use of the maximum number of dimmers restricts the speed of operation.

The DMX-connection allows 32 lighting units to be linked to the bus. There is no limitation as to the number of addresses that each of these units can handle.

CABLES AND ALL THAT

The cables carry rectangular-wave signals at a frequency of 250 kHz maximum. Each signal may contain components at frequencies up to 2.5 MHz. This means that in the DMX system cables must be used that are quite different from the ones used in analogue systems. No longer can standard cable with simple connectors be used: specific types of cable with corresponding connectors are imperative.

As mentioned earlier, the system is based on the RS485 interface, which is an improvement of the earlier RS422 system. The improvements make possible more connections to the bus and additional space for more masters. The latter facility is not used in the DMX512 system, but the former enables applications within a network. Although the RS485 standard limits the length of the cable, exceeding the specified length within reason will not create difficulties: distances of up to 1 metre (3.3 ft) are perfectly usable, provided that the final unit in the chain is terminated correctly into an impedance of 120 Ω.

If larger distances need to be spanned, a bus repeater should be used. The circuit of such a repeater is shown in **Figure 3**. Note that both the input and output are terminated into 120 Ω. The DMX-512 standard does not specify the electrical isolation.

SOME LIMITING VALUES

It is important that the driver can handle signal levels between 1.5 V and 5 V at a common-mode potential between –7 V and +12 V.

The leakage current at the output should not exceed 100 μA during an output signal.

The input impedance of the receiver must be not lower than 12 kΩ, while the output load must not exceed 60 Ω.

Short-circuit currents of 150 mA to earth and 250 mA to the positive supply line are permissible.

This article is based on information available in the relevant Internet information from Soundlight (http://www.soundlight.de/techtips/dmx512/dmx512.htm).

[980013]

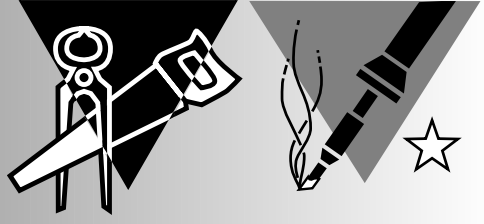
Pinout of connectors used

5-way AXR (XLR) plug

pin	*function*
1	*earth (screen)*
2	*DMX-*
3	*DMX+*
4	*n.c. (may be linked to DMX–)*
5	*n.c. (may be linked to DMX+)*

3-way AXR (XLR) plug

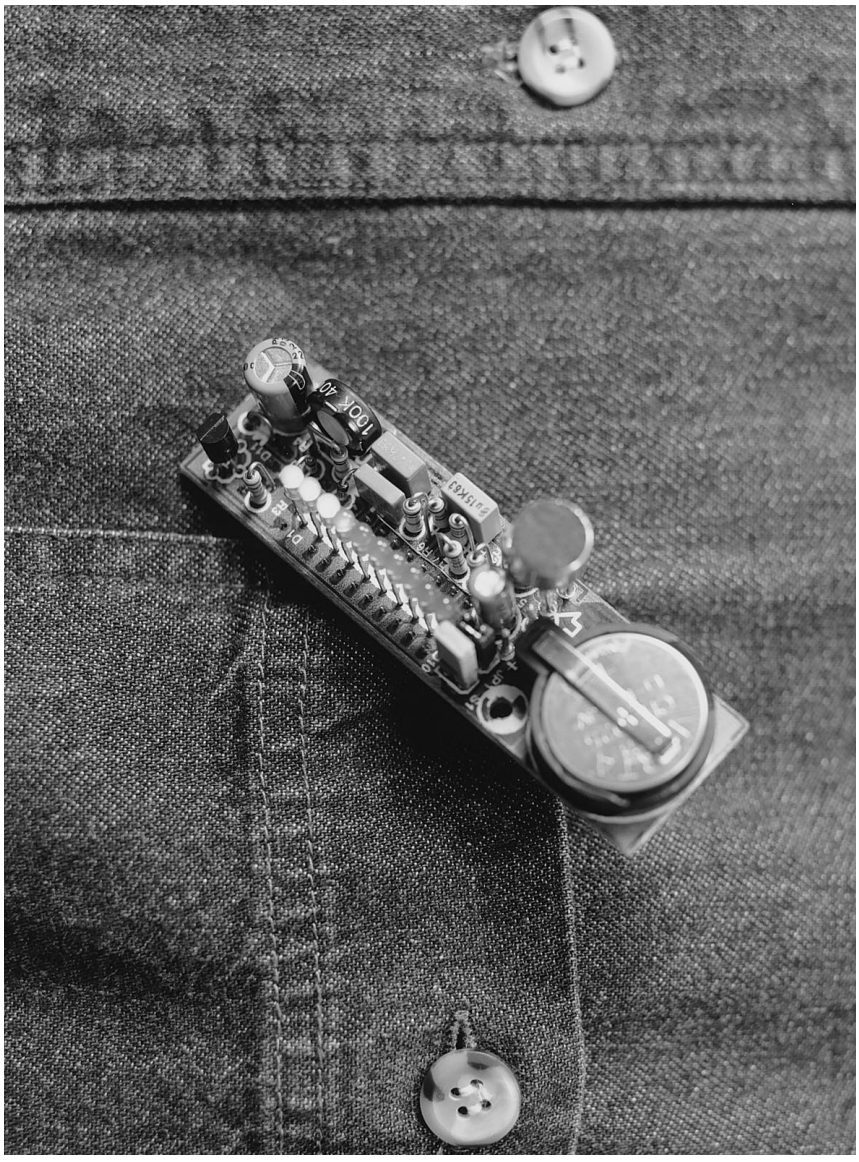
pin	*function*
1	*earth (screen)*
2	*DMX-*
3	*DMX+*



functional trinket

reacts to sound pressure

Strictly speaking, a trinket cannot be functional, but the title of this article is apt. It is a kind of miniature VU meter whose LED bar fluctuates in rhythm with ambient sound. Owing to its modest dimensions, it can easily be worn as an adornment which, in a disco or at a party, will, no doubt, draw the attention of many.



Design by T. Giesberts

Nowadays, not many things surprise us any more. In this age of high technology, we are used to all kinds of new discoveries and developments, and technical ingenuity. Mobile telephones, portable CD players, watches with built-in alarm: what is there left to impress us with? Not only satellite TV, but also a radio-controlled vehicle on the planet Mars are accepted as a commonplace.

Of course, this is true not only in the world of science and technology, but also in other spheres of human interest. It is not easy to dream up something really new or innovative – something that draws a spontaneous reaction of "Fancy that" or "How do

1

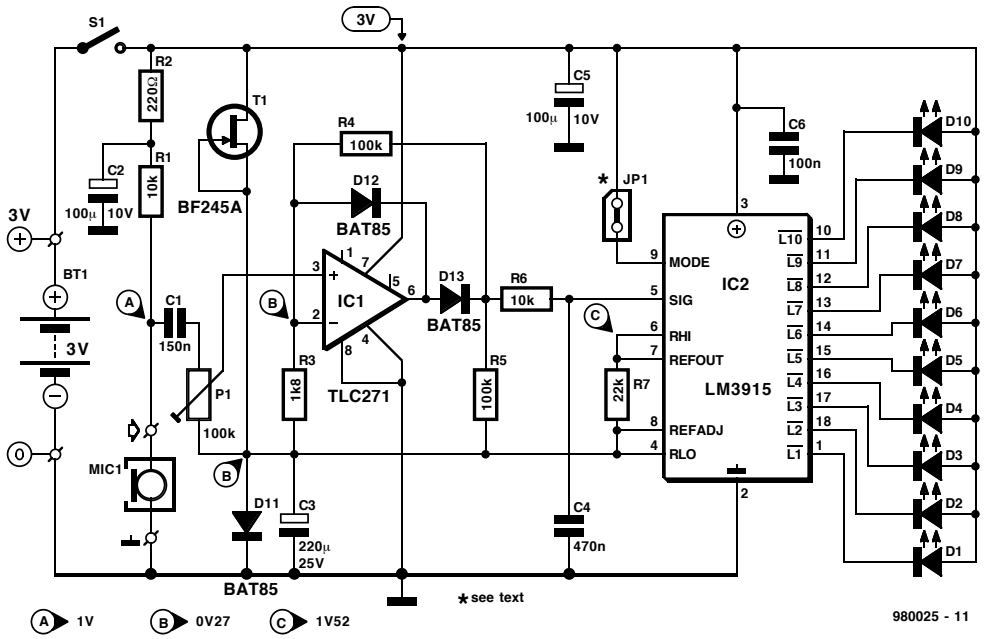


Figure 1. The circuit consists of a microphone, an amplifier/rectifier, and a display of ten LEDs.

they do it?"

The trinket described in this article is designed specially for lovers of music and dance. It is intended to enable constructors to build something that sets them apart from others. It is not exactly hi-tech, but rather a combination of technology and music. It is an adornment that, by means of a moving point of light, reacts to the sound pressure of music to which it is exposed. It may, nevertheless, also fulfil a useful function: when the wearer of the trinket notes that the LED indication is constantly at maximum, it is time to put the ear plugs in, because the sound pressure is then clearly no longer healthy.

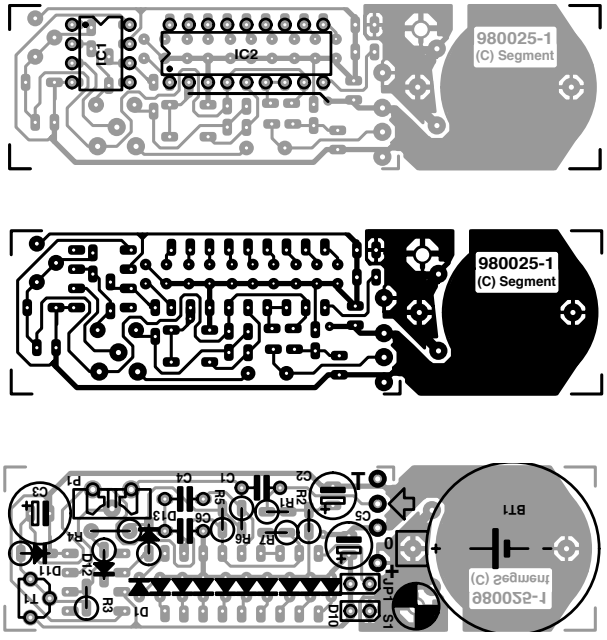
Figure 2. Placing the ICs at the track side of the board meant that the dimensions could be kept small.

DESIGN

The aim of the design is to make variations in detected ambient sound, music or speech, visible by means of a moving LED or bar of LEDs, not unlike the way a VU meter works. At the same time, it has been kept tiny (75×20 mm or 3×0.8 in)to enable it being worn as a brooch .

Clearly, a microphone is needed, and fortunately electret types are available in very small sizes. Also, tiny LEDs for the display are readily available today. The remainder is some electronics to convert the picked up signals to a stepwise varying direct voltage for driving the LEDs. A final requirement is that the electronics can operate for a long

2



Parts list

Resistors:
R_1, R_6 = 10 kΩ
R_2 = 220 Ω
R_3 = 1.8 kΩ
R_4, R_5 = 100 kΩ
R_7 = 22 kΩ
P_1 = 100 kΩ preset, upright

Capacitors:
C_1 = 0.15 µF, pitch 5 mm
C_2, C_5 = 100 µF, 10 V, radial
C_3 = 220 µF, 25 V, radial
C_4 = 0.47 µF, pitch 5 mm
C_6 = 0.1 µF, pitch 5 mm

Semiconductors:
D_1–D_{10} = LED, 3 mm, high efficiency
D_{11}–D_{13} = BAT85
T_1 = BF254A

Integrated circuits:
IC_1 = TLC271CP
IC_2 = LM3915N

Miscellaneous:
JP_1 = jumper
S_1 = miniature on/off switch or jumper (see text)
BT_1 = 3 V lithium battery Type CR2025 or CR2032 with holder for board mounting
MIC_1 = electret microphone, dia. ≤ 10 mm
PCB Order no. 980025-1 (see Readers Services towards the end of this issue)

time from a small, low-voltage battery.

The final design is shown in **Figure 1**. The sound is picked up by a tiny electret microphone MIC_1. It has a diameter of about 10 mm (0.4″) and contains an integrated impedance equalizer that also functions as amplifier. The supply line to this amplifier is set by R_1 to just under half the supply voltage (test point A). The output of the microphone is applied to sensitivity control P_1 via capacitor C_1.

The signal at the wiper of P_1 is applied to operational amplifier IC_1. This stage has a twofold function: amplifier and, in conjunction with D_{12} and D_{13}, single-phase rectifier. Its amplification is determined by the ratio R_3:R_4: with values as specified, it amounts to ×6. This results in a direct voltage at its output (pin 6) that varies in accordance with the strength of the signal picked up by the microphone. This voltage is averaged to some extent by network R_6-C_4 to prevent very rapid fluctuations, which would lead to an unstable display.

The display is formed by a bar of ten LEDs that are controlled by the well-known Type LM3915 driver (IC_2). The driver comprises a voltage reference source, an accurate potential divider and ten comparators, each of which can control an LED directly. The potential difference between two successive LEDs corresponds to a sound pressure difference of 3 dB.

The direct voltage applied to pin 5 of IC_2 can be converted to a single, wandering LED (dot mode) or to a fluctuating bar display (bar mode) via pin 9. In the dot mode, the contacts of jumper JP_1 must remain open; in the bar mode, they should be closed.

3

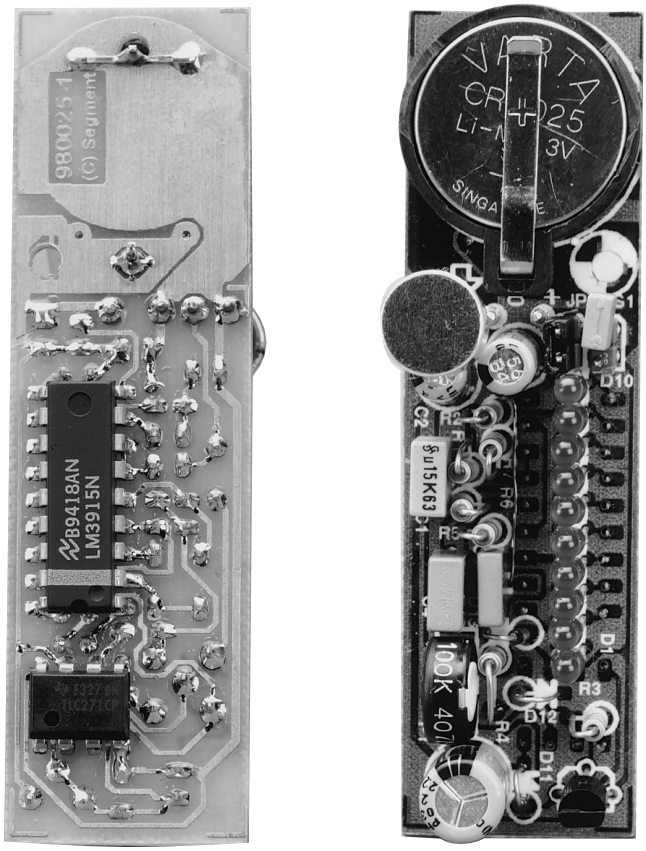


Figure 3. Both sides of the completed prototype board. Soldering the ICs in place requires a steady hand an a fine-tipped soldering iron.

CONSTRUCTION

The design of the printed-circuit board in **Figure 2** is a compromise between small size and ease of construction. This has been accomplished by accommodating IC_1 and IC_2 on the track side of the board instead of as normal on the component side. Note that soldering these components requires a small soldering iron with a very fine tip.

The circuit is powered by a 3 V lithium cell that is fitted on to the board with the aid of a specially available holder. Note that the +ve terminal must point upward.

The microphone is soldered directly to the pins marked with an input arrow.

On/off switch S_1 may be replaced (as it is in the prototype) by a jumper, which is smaller than a switch.

Note that D_2–D_{10} are all placed in the same direction, but D_1 the other way around.

FINALLY

The completed prototype board is shown in **Figure 3**. A clip to enable the board to be fastened to a lapel or similar may be soldered or glued at the underside near the battery (if soldered, take care not to cause a short-circuit).

There are, of course, other possibilities of construction: for instance, the LEDs may be clustered together away from the board (which can then be hidden in, say, a breast pocket). The two sections can then be linked by a mini cable. Ingenious readers can, no doubt, think of different constructions.

The LEDs specified have a small outer edge. This edge must be filed or cut away with a sharp knife to enable these diodes to be placed close together on the board.

Although problems are highly unlikely, the voltage level at three test points is given in the circuit diagram to facilitate faultfinding.

Depending on the type of microphone used, it may be necessary to change the value of R_1 to obtain a voltage of 1 V at test point A. If this is done, it is advisable to check the offset at the output of the op amp across pins 4 and 5 of IC_2 (without microphone). If this is higher than 100 mV, it is advisable to replace IC_1.

The sensitivity of the circuit with a standard electret microphone and P_1 set to maximum is arranged to give a full display for a sound pressure input level of 100 dB. Note that a sustained input at this level is dangerous for your hearing. If the sensitivity is considered insufficient, it may be increased by giving R_3 a value of 1.5 kΩ or even 1.2 kΩ.

BATTERY

The circuit draws a current of about 6 mA with all LEDs out, about 12 mA in the dot mode, and up to 22 mA with all LEDs on in the bar mode. Since a Type CR2025 battery has a capacity of 120 mAh, and a CR2032 one of 170 mAh, the trinket will continue to light even when the party lasts until the early hours – certainly in the dot mode.

[980025]

COMMANDS		
Command	**Function**	**Example(s)**
RUN	Execute a program	RUN
CONT	CONTinue after a STOP or Control-C	CONT
LIST	LIST program to the console device	LIST LIST 10-50
LIST#	LIST program to serial printer	LIST# LIST# 50
NEW	erase the program stored in RAM	NEW
NULL	set NULL count after carriage return-line feed	NULL
RAM	evoke RAM mode, current program in READ/WRITE memory	RAM
ROM	evoke ROM mode, current program in ROM/EPROM memory	ROM ROM 3
XFER	transfer a program from ROM/EPROM to RAM	XFER
PROG	save the current program in EPROM	PROG
PROG1	save baud rate information in EPROM	PROG1
PROG2	save baud rate information in EPROM and execute program after RESET	PROG2
FPROG	save the current program in EPROM using the INTELligent algorithm	FPROG
FPROG1	save baud rate information in EPROM using the INTELligent algorithm	FPROG1
FPROG2	save baud rate information in EPROM and execute program after RESET, use INTELligent algorithm	FPROG2
STATEMENTS		
Statement	**Function**	**Example(s)**
BAUD	set baud rate for line printer port	BAUD 1200
CALL	CALL assembly language program	CALL 900H
CLEAR	CLEAR variables, interrupts and Strings	CLEAR
CLEARS	CLEAR Stacks	CLEARS
CLEARI	CLEAR interrupts	CLEARI
CLOCK1	enable REAL TIME CLOCK	CLOCK1
CLOCK0	disable REAL TIME CLOCK	CLOCK0
DATA	data to be read by READ statement	DATA 100

POP	POP argument stack to variables	POP A,B,C
PWM	PULSE WIDTH MODULATION	PWM 50,50,100
REM	remark	REM DONE
RETI	return from Interrupt	RETI
STOP	break program execution	STOP
STRING	allocate memory for strings	STRING 50,10
UI1	evoke User console Input routine	UI1
UI0	evoke BASIC console Input routine	UI0
UO1	evoke User console Output routine	UO1
UO0	evoke BASIC console Output routine	UO0
OPERATORS – DUAL OPERAND		
Operator	**Function**	**Example(s)**
+	addition	1 + 1
/	division	10 / 2
**	exponentiation	2**4
*	multiplication	4*4
–	subtraction	8–4
.AND.	logical AND	10.AND.5
.OR.	logical OR	2.OR.1
.XOR.	logical exclusive OR	3.XOR.2
OPERATORS – SINGLE OPERAND		
Operator	**Function**	**Example(s)**
ABS()	absolute value	ABS(–3)
NOT()	ones complement	NOT(0)
INT()	integer	INT(3.2)
SGN()	sign	SGN(–5)
SQR()	square root	SQR(100)
RND	random number	RND

READ	read data in DATA statement	READ A
RESTORE	restore READ pointer	RESTORE
DIM	allocate memory for arrayed variables	DIM A(20)
DO	set up loop for WHILE or UNTIL	DO
UNTIL	test DO loop condition(loop if false)	UNTIL A= 10
WHILE	test DO loop condition	WHILE A= B
END	terminate program execution	END
FOR-TO-{ STEP}	set up FOR-NEXT loop	FOR A= 1 TO 5
NEXT	test FOR-NEXT loop condition	NEXT A
GOSUB	execute subroutine	GOSUB 1000
RETURN	return from subroutine	RETURN
GOTO	go to program line number	GOTO 500
ON GOTO	conditional GOTO	ON A GOTO 5,20
ON GOSUB	conditional GOSUB	ON A GOSUB 2,6
IF-THEN-{ ELSE}	conditional test	IF A= B THEN A= 0
INPUT	INPUT a string or variable	INPUT A
LET	assign a variable or string a value (LET is optional)	LET A= 10
ONERR	ONERR or GOTO line number	ONERR 1000
ONTIME	generate an interrupt when TIME is equal to or greater than ONTIME, argument-line number is after comma	ONTIME 10, 1000
ONEX1	GOSUB to line number following ONEX1 when INT1 pin is pulled low	ONEX1 1000
PRINT	PRINT variables, strings or literals, P. is shorthand for PRINT	PRINT A
PRINT#	PRINT to software serial port	PRINT# A
PH0.	PRINT HEX mode with zero suppression	PH0.A
PH1.	PRINT HEX mode with no zero	PH1.A
PH0.#	PH0.# to line printer	PH0.# A
PH1.#	PH1.# to line printer	PH1.# A
PUSH	PUSH expression on argument stack	PUSH 10,A

MCS-52 BASIC Language
Quick Reference
80C32 BASIC Control Computer
Instruction Set 2

DATASHEET 2/98

ELEKTOR ELECTRONICS

LOG()	natural log	LOG(10)
EXP()	e (2.7182818) to the x	EXP(10)
SIN()	returns the sine of argument	SIN(3.14)
COS()	returns the cosine of argument	COS(0)
TAN()	returns the tangent of argument	TAN(.707)
ATN()	returns the arctangent of argument	ATN(1)
OPERATORS – SPECIAL FUNCTION		
Operator	**Function**	**Example(s)**
CBY()	read program memory	P.CBY(4000)
DBY()	read/assign internal data memory	DBY(99)= 10
XBY()	read/assign external data memory	P.XBY(10)
GET	read console	P.GET
IE	read/assign IE register	IE= 82H
IP	read/assign IP register	IP= 0
PORT1	read/assign I/O port 1 (P1)	PORT1= 0FFH
PCON	read/assign PCON register	PCON= 0
PCAP2	read/assign RCAP2 (RCAP2H:RCAP2L)	RCAP2= 100
T2CON	read/assign T2CON register	P.T2CON
TCON	read/assign TCON register	TCON= 10H
TMOD	read/assign TMOD register	P.TMOD
TIME	read/assign the real time clock	P.TIME
TIMER0	read/assign TIMER0 (TH0:TL0)	TIMER0= 0
TIMER1	read/assign TIMER0 (TH1:TL1)	P.TIMER1
TIMER2	read/assign TIMER0 (TH2:TL2)	TIMER2= 0FFH
STORED CONSTANT		
PI	pi = 3.1415926	PI

MCS-52 BASIC Language
Quick Reference
80C32 BASIC Control Computer
Instruction Set 4

DATASHEET 2/98

ELEKTOR ELECTRONICS