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

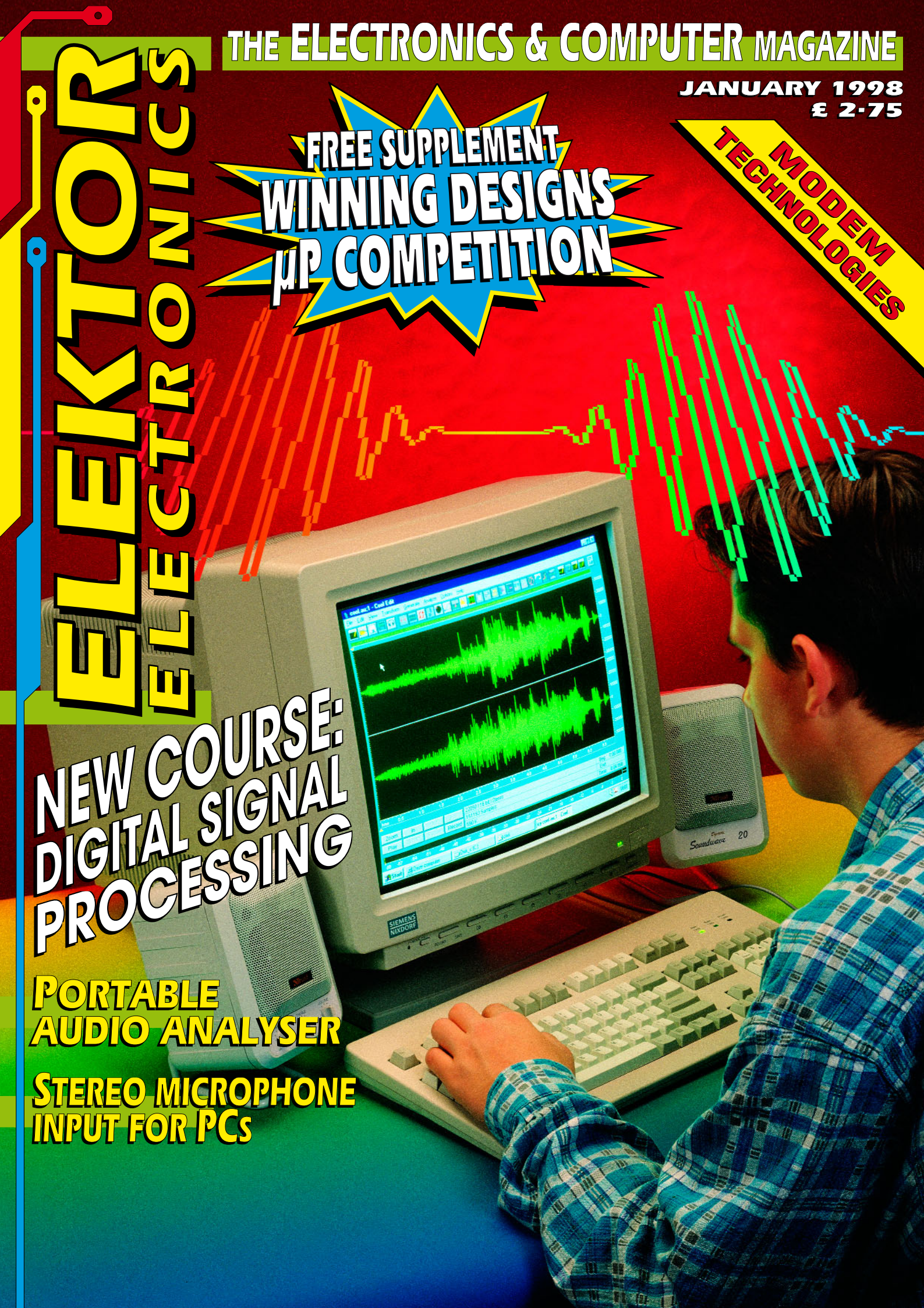
FREE SUPPLEMENT
WINNING DESIGNS
µP COMPETITION

MODEM TECHNOLOGIES

NEW COURSE: DIGITAL SIGNAL PROCESSING

PORTABLE AUDIO ANALYSER

STEREO MICROPHONE INPUT FOR PCs



ELEKTOR ELECTRONICS

CONTENTS

January 1998 Volume 24
Number 262 ISSN 0268/4519

16-page SUPPLEMENT on the International Microprocessor Competition 1997
Centrefold

Portable sound-pressure meter 14

20 Introduction to Digital Signal Processing

42 Stereo microphone input for PC

ABC AUDIT BUREAU OF CIRCULATIONS
CONSUMER PRESS

When he received his prize for the 1995 Elektor Electronics Circuit Design Competition, the winner of the international first prize, Laurent Lamesch, had his next project planned already. It was going to be an IC tester with microprocessor control, suitable for stand-alone operation. The fantastic results, as it turned out, not even Laurent himself was able to guess: he won the International First Prize in Elektor's 1997 Microprocessor Competition.

International Microprocessor Competition 1997

The Winner and his Prize: Laurent Lamesch (left) receives one box after the other from Jury representative Harry Baggen.

The number of entries received for the 1997 competition exceeded that of the 1996 Competition, which called for 'software' only. This came as a surprise because in 1995 the Jury members were convinced that 'microprocessor designs' would be too thin a base for a competition!

The number of entries received from readers of the four participating national editions of Elektor in Germany, France, Holland and the rest of the world (coverage of the UK magazine) was also surprising, in particular, if you look at the circulation figures of the individual magazines. As compared with the readers of the German and French editions, the representation of those taking the English and Dutch magazine is not very strong. National preferences are also noticed when it comes to popularity of the various microcontrollers: Motorola's 68/68HC family, for instance, has more fans in Holland and France than elsewhere. By contrast, the 8051 (MCS-51) series and its spin-off controllers are evergreens in Germany and the UK. The winner, however, is the PIC family which seems to be popular around the globe. There is heavy competition, however, from Atmel's AVR series!

The International Prize Winner

When Laurent Lamesch casually mentioned his intentions of designing an IC tester three years ago, he was fairly sceptical about the feasibility of such a project. Consequently, he was no less surprised than the Jury. Because all entries receive a grade between 1 and 10 from five Jury members, the overall winner is not known until the last moment when all points are added up. Looking at the total scores, it was clear that the 80C535 based IC tester was way ahead of the competition.

Laurent Lamesch had originally intended the project for personal use only, and started it as a side activity (although the basic ideas had rooted long ago). The Competition announced in the July/August 1997 magazine provided the motivation to finish the design relatively quickly.

Laurent Lamesch is a 26 year old Luxembourger who works as an electronics design engineer in the automotive industry. At the age of 17 he started to read Elektor and take an interest in electronic design. He received his formal education in electrical engineering at the ETH in Zürich. Whereas digital electronics represents a professional as well as personal interest area, Laurent Lamesch is also active as a bike rider and a sound engineer at a local radio station, Radio Ara (103.3 or 105.2 MHz in Luxembourg, sound samples also available at **http://www.ara.lu**).

The prize, a complete Microchip PIC Development System donated by Arizona Microchip (USA/France) and representing a value of £2310 should be very useful to Laurent Lamesch. Interestingly, his thesis was a PIC-based application, written and produced without the luxury of such a powerful development system...

(982015-1)

This article supplies a condensed description of a stand-alone IC tester for SSI (small-scale integration) logic ICs (with up to 24 pins) from the well-known 74xx and 40xx series. The elementary building blocks that make up the design are an 80C535 microcontroller, a large EPROM, an LCD display, a small keyboard and an RS232 interface. The latter allows the tester to receive new IC test vectors produced by a test vector compiler which runs on a PC. Another DOS program allows new test vectors to be tested using the same RS232 interface, so you don't have to re-program the system EPROM or dig up an EPROM emulator.

By Laurent Lamesch (Luxembourg)

IC tester

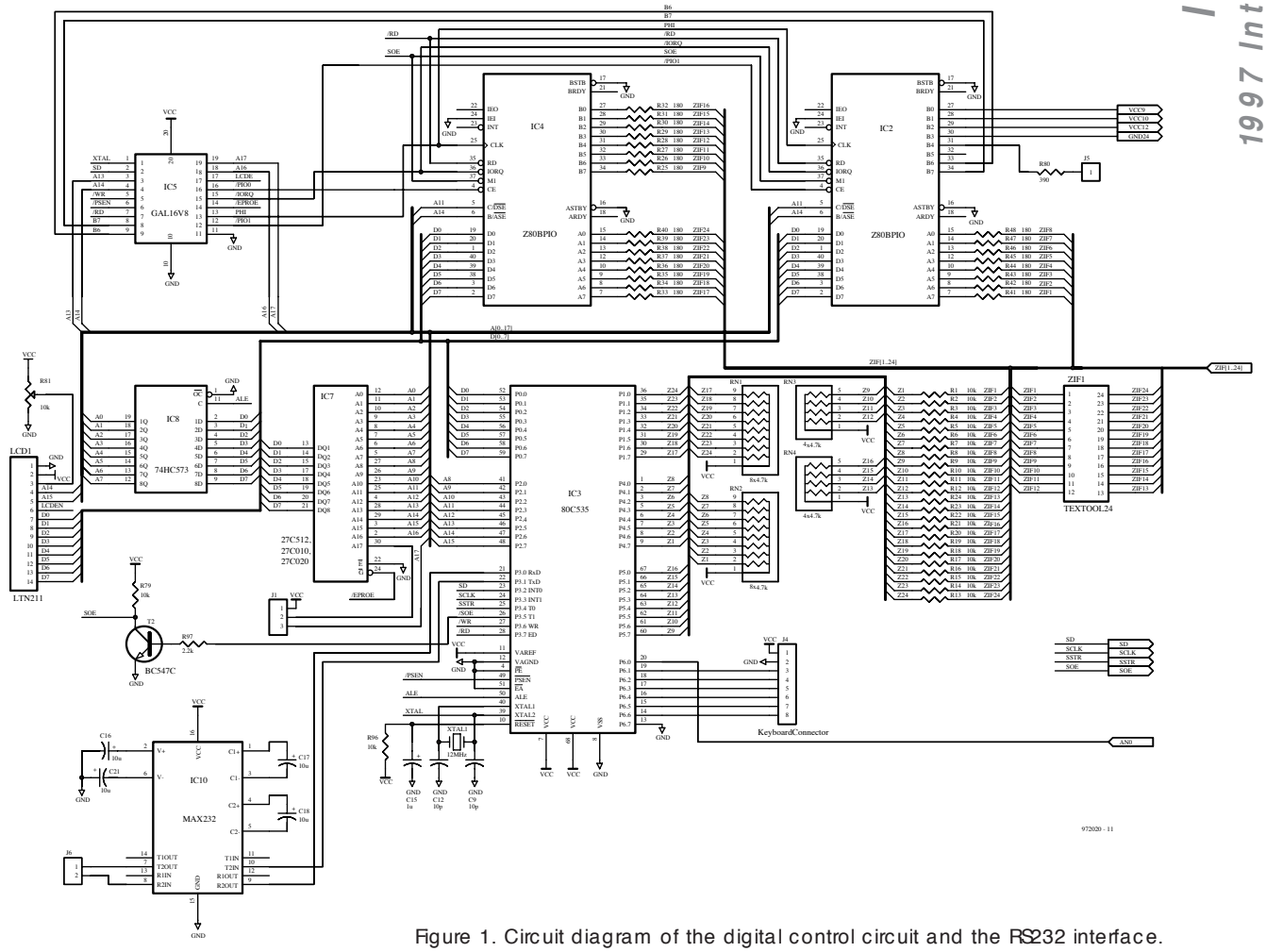


Figure 1. Circuit diagram of the digital control circuit and the RS232 interface.

All input pins of the device under test (DUT) inserted in the Textool zero-insertion force socket, ZIF1, may be pulled to the logic high (H) or logic low (L) level by means of the outputs on the Z80PIOs and current limiting resistors. The same PIOs also enable the logic states of the DUT outputs to be checked, while ports 1, 4 and 5 of the 80C535 are used to detect which DUT pins represent a high impedance. Furthermore, the supply voltage pins of the DUT may be con-

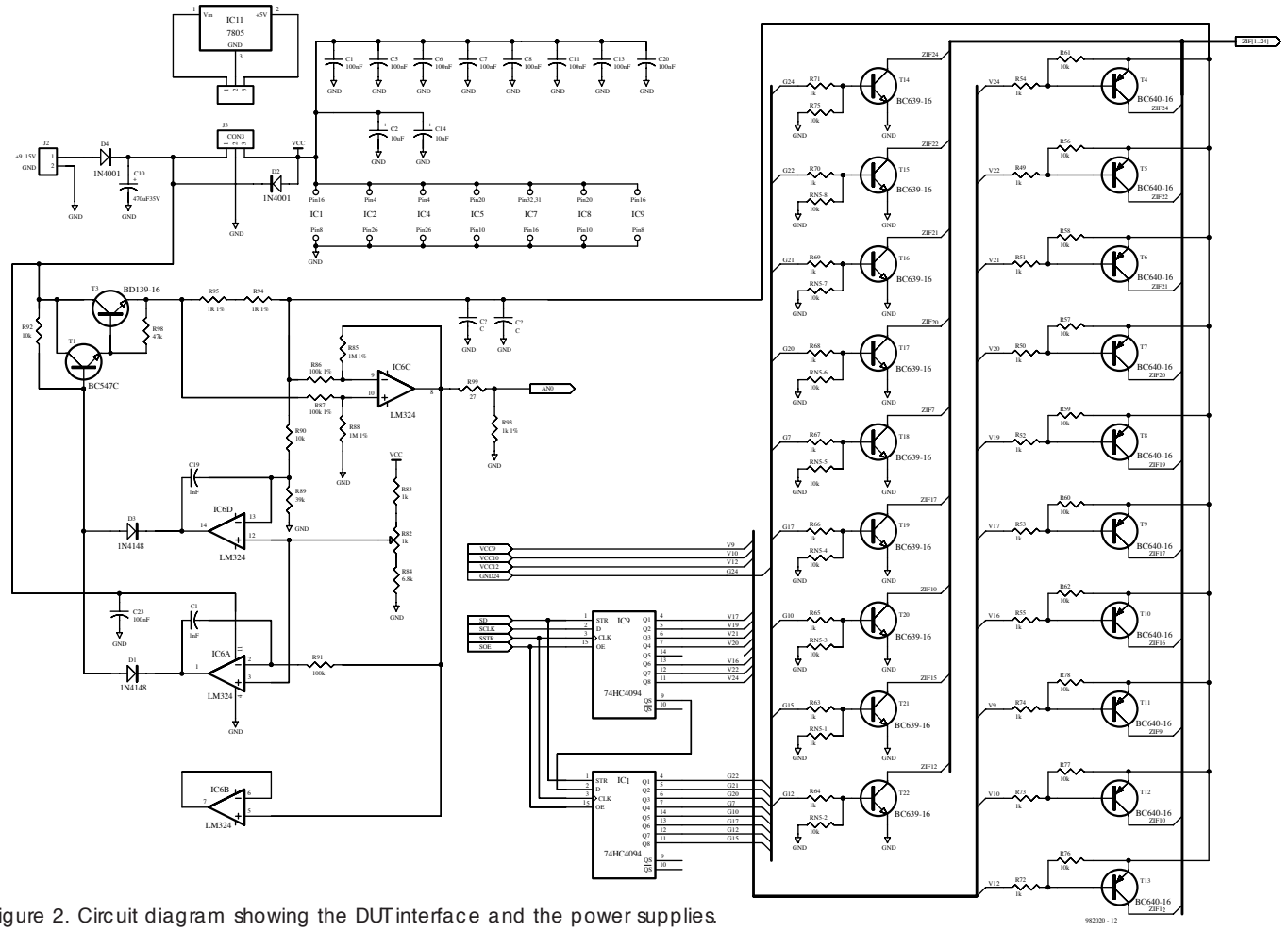


Figure 2. Circuit diagram showing the DUT interface and the power supplies.

nected to GND or a current-limited voltage source via switching transistors.

This voltage source (built around IC6) supplies 5.2 V, and its output current is limited to about 0.2 A. The actual output current is converted into a proportional voltage for measurement by the controller.

Depending on the size of the test vector files, the control program of the IC tester is contained in a 27C512, 27C010 or 27C020 EPROM. If multiple banks are present inside the EPROM, then the selection between the 64-kByte chunks is accomplished by outputs B6 and B7 of IC2. The 80C535 also controls an LC display and a keyboard with 6 keys. A GAL, IC5, looks after the address decoding, and also generates the PHI signal for the Z80PIOs.

The reason for using the Z80PIO to control and monitor the DUT inputs and outputs is that this chip is the only widely available 16-bit parallel port IC of which all port line directions are individually controllable, while the output drivers for all port lines consist of push-pull circuits.

Jumper J1 selects between a 28-pin and a 32-pin EPROM in position IC7. When a 28-pin EPROM is used, J1-1 is connected to J1-2. When a 32-pin EPROM is used, connect J1-2 to J1-3.

A 9-way sub-D socket is connected to J6. This enables you to connect the IC tester to the RS232 serial port on your PC. Ground for the serial interface is taken from J2. The pin connections of the sub-D socket are as follows:

9-way sub-D	J6	J2
2	1	
3	2	
5		2

Adjustment

The only adjustment in the circuit is the DUT supply output voltage. This is set to 5.2 V ±0.05 V using preset R82.

Operation

The tester is operated using six keys labelled Enter, Escape, dn (scroll down), up, dn2 (fast scroll down), and up2 (fast scroll up). The up and dn keys have an auto-repeat function which causes the repeat rate to be automatically increased as the key is held depressed.

LED D5 lights to indicate that the IC under test is being powered, and should not be removed from the ZIF socket.

Pressing the escape key takes you to the main menu. There, the following functions may be selected:

1. **Test IC:** the user picks an IC from an IC library, and the DUT is checked for correct operation. The test may be repeated. If indicated by the test vectors, the current consumption of the IC under test is measured and displayed.
2. **Identify:** this allows you to identify the type number of an unknown IC. If the GND and Vcc pins are unknown, only those test vectors are used that have the GND and Vcc pins at the same positions. The GND/Vcc pin entry is optional. Next, you can select the libraries that have to be scanned.
3. **Retest IC:** once an IC has been tested or identified, it may be tested again without having to pick it from the libraries.
4. **Trace:** all test vectors and the response of the DUT to these vectors appear in succession on the LC display.
5. **Options:** here, you can define global options.
6. **Info:** information on version and copyright.
7. **Self Check:** the IC tester hardware

may be checked using this function and a voltmeter.

8. **Remote Mode:** connect a PC to the RS232 interface and debug test vectors using the DOS program TVCHK.EXE.

The up/dn keys are used to scroll one item up or down. The up2/dn2 keys do the same, but then five items at a time. The ent(er) key is used to confirm a selection. Esc, finally, jumps to the main menu.

Test vector compiler and debugger

ICVC.EXE is the test vector compiler which generates a Test Vector Binary File from the Test Vector Source File. The Binary File has to be appended to microcontroller program, ICT.BIN, before an EPROM may be programmed for the IC tester. File appending is achieved with the simple 'copy /b ...' command.

ICTVC.EXE is launched by typing ICTVC scrfile.TVC at the DOS prompt. So, for instance, ICTVC VECT.TVC. It generates the following files:
TVC.OUT: Test Vector Binary File
ERR.OUT: Error report
LIST.OUT: List file containing information on the source file, the binary file, a copy of the source file with line numbers, plus, for each line, the bytes that were generated from this line.
TMP.OUT: temporary file, used by ICTVC.EXE.
If an error occurs during the compilation phase, the relevant error reports are written to ERR.OUT. Error reports are not displayed on the PC monitor.

TVCHK is a shell program which calls ICTVC.EXE, and enables the generated test vectors to be debugged. This program has to be launched using the COM port to which the IC tester is connected, or the address and interrupt line of this COM port, as an appended switch. So, for instance, TVCHK2.

The batch file **EDT.BAT** is used to launch an external text editor (word processor). If another editor than EDIT.COM is used, this batch file has to be modified accordingly. The batch file launches the editor and conveys the first batch file parameters. The Test Vector Source File is relatively large, so your external editor may not be able to load it entirely. If that happens, you should create a file which only contains the new test vectors (if new test vectors are being debugged). Once the new vectors have been debugged, they may be appended to VECT.TVC without using the TVCHK program. the option 'Compile tv source code' is then used to compile the complete file.

A file called **SMALL.TVC** is supplied for practicing. Select option 2 (Edit & compile tv source file) in the TVCHK program and then enter SMALL.TVC instead of VECT.TVC.

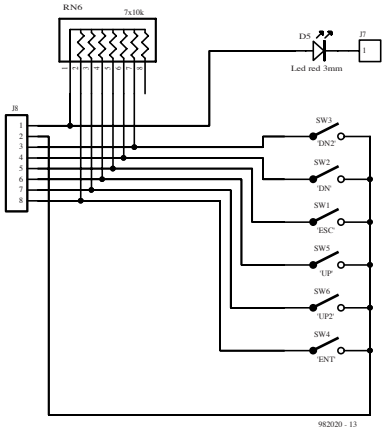


Figure 3. The keyboard circuit.

Building this project

As already mentioned, this article presents a condensed description of a

Index of IC test vectors in VECT.TVC

Library: 74xxx
74:00
74:01*
74:02
74:03
74:04
74:05
74:06
Parent: 74:05
74:07
74:08
74:09
74:10
74:11
74:12*
74:13
74:14
74:15*
74:16
74:17
74:18*
74:19*
74:20
74:21
74:22*
74:24*
74:25
74:26
74:27
74:28*
74:30
74:31*
74:32
74:33
74:34*
74:35*
74:37
74:38
74:39*
74:40
74:42
74:45
74:46*
74:47
74:48*
74:49*
74:51 St,S*
74:51 LS,L
74:54*
74:55*
74:73
74:74
74:75
74:76
74:83
74:86 -C,-L
74:86 C,L*
74:90
74:92
74:93
74:95A,B
74:100
74:107
74:109
74:119
74:125
74:126*
74:128*
74:131*
74:132
74:133*
74:136*
74:137
74:138
74:139
74:140*
74:147
74:145*
74:148
74:150
74:151
74:153*
74:154
74:155
74:156*
74:157
74:158*
74:159*
74:160
74:161
74:162
74:163
74:164
74:165
74:166
74:168*
74:169
74:170
74:173
74:174
74:175
74:180*
74:184*
74:185*
74:190
74:191
74:192
74:193
74:194
74:237*
74:238*
74:239
74:240
74:241
74:242*
74:243
74:244
74:245
74:247*
74:248*
74:249*
74:250*
74:251
74:253
74:257
74:258*
74:259
74:260*
74:266
74:273
74:280
74:283
74:290*
74:293
74:299
74:323
74:347*
74:348*
74:352*
74:353*
74:365
74:366*
74:367
74:368*
74:373
74:374
74:375
74:377*
74:386*
74:390
74:393
74:412
74:425*
74:426*
74:445*
74:447*
74:465*
74:466*
74:467*
74:468*
74:518*
74:519*
74:520*
74:521
74:522*
74:533*
74:534*
74:540*
74:541
74:563*
74:564*
74:573
74:574
74:576*
74:580*
74:590
74:591*
74:592
74:595
74:596*
74:620*
74:621*
74:622*
74:623*
74:638*
74:639*
74:640
74:641*
74:642*
74:643*
74:644*
74:645
74:646
74:647*
74:648*
74:649*
74:668*
74:669*
74:670
74:682
74:683*
74:684
74:685*
74:688
74:689*
74:699
74:746**
74:747**
74:756*
74:757*
74:758
74:759*
74:760*
74:762*
74:763*
74:810*
74:811*
74:1000*
74:1002*
74:1003*
74:1004*
74:1008*
74:1010*
74:1020*
74:1032*
74:1034*
74:1035*
74:1036*
74:1240*
74:1244*
74:1245*
74:1640*
74:1645*
74:2240*
74:2241*
74:2244*
74:2540*
74:2541*
74:7245*
74:7266*
74:7540*
74:7541*
Library: 40xxx
4001
4002
4009*
4010*
4011
4012
4013
4014*
4015
4016
4017
4019
4020
4021
4022*
4023
4024
4025
4027
4028
4029
4030
4040
4041*
4042
4043
4044*
4049**
4050**
4051
4052
4053
4056
4060
4066
4067
4068 -RCA*
4068 RCA*
4069
4070
4071
4072
4073
4075
4076
4077
4078 -RCA*
4078 RCA*
4081
4082*
4093
4094
4099*
40014*
40102
40103
40105
40106
40160
40161
40162
40163
40174
40175
40192
40193
40194
4502
4508
4510
4511
4512
4514
4515*
4516*
4518
4520
4522
4526
4529
4539
4543*
4555
4556*
4584
4724

** Test vector for this IC not yet verified with a correctly operating IC.*
*** This IC has not been fully tested using the IC test vectors.*
: Any TTL family identifier, except if the type number has a suffix.
-X Test vectors not valid for X family device of this IC (e.g. 74:86 -C means not for 74C86).
X Test vectors apply only to X family device of this IC (e.g. 74:86 C means for 74C86 only).

fairly large project which consists of hardware, software and extensive documentation. Unfortunately, the overall size of the project as submitted by the author is such that it is not possible to reproduce it in full in this supplement. After all, we want to show you some of the other prize-winning entries, too!

The IC tester project will be described as a project for home construction in a future issue of *Elektor Electronics*, complete with a printed circuit board, a ready-programmed EPROM and tested project software on disk.

Meanwhile, the design (as we received it) will be included on a Contest Compilation CD-ROM which we hope to publish by the end of January 1998. (982020-1)

Development tools used

IC Tester firmware:	Metalink 8051 Cross Assembler V1.2h
	Signetics HEX2BIN.EXE
PC programs:	Borland Turbo C++ V3.0
Schematic layout[1]:	Protel Advanced Schematic V3.5
PCB layout[2]:	Easytrax V2.06
GAL assembler:	c't Galio

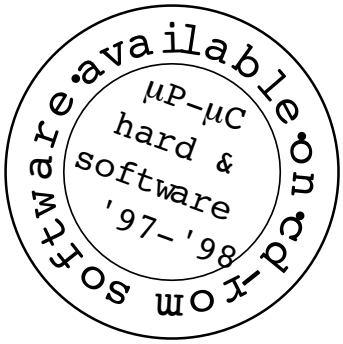
[1] cosmetic changes by Elektor
[2] layouts not reproduced in this article

FIRST NATIONAL PRIZE UK

1997 International Microprocessor Competition

PIC on the Rocks is a system for PIC users that allows them to do in-circuit debugging of PIC programs. It uses a simple serial connection with a host PC running Windows95 or NT. The PIC program the user wants to check also needs to embed a small piece of telemetry code. This code enables the PIC side of the telemetry link. The project software includes source code for both the PIC and the host PC. It also comprises a manual in Word format and some examples.

By Ben de Waal (USA)



PIC on the Rocks

The PIC micro controller range from Microchip has become one of the leading micro controllers used by engineers and hobbyists to solve simple and medium complexity embedded problems. Personally I have used PICs in pagers, Virtual Reality peripherals, home security systems and even hamster treadmill measurements.

One of the major advantages of PICs is that they are relatively inexpensive. The basic development tools are also fairly cheap and this makes it easy for the hobbyist (*Elektor Electronics* reader) to develop software for it. The major drawback is that PIC debugging tools and In Circuit Emulators (ICE) are very expensive. This creates the common scenario that average Joe Hobbyist has the development tools but also has to burn and erase endless number of times to get the solution 'just right'. Most of the time he tries to figure out 'what goes on' inside the PIC since he cannot afford or justify the cost of an ICE.

This project addresses the PIC debugging issue by providing an almost non-intrusive solution for PIC telemetry (remote debugging). It allows you to pause PIC execution, look at the internal registers (files), edit them and continue till the next 'break point', etc.

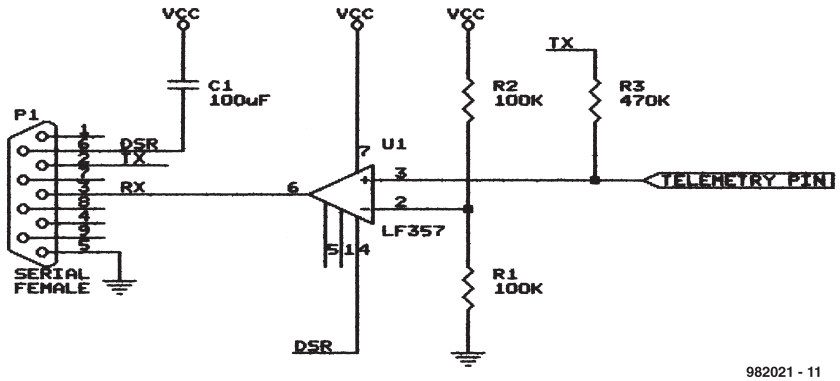


Figure 1. The opamp solution. Note: depending on the serial cable, you may have to swap pins2 and 3 as well as pins4 & 6. DSR should be an output.

fairly large project which consists of hardware, software and extensive documentation. Unfortunately, the overall size of the project as submitted by the author is such that it is not possible to reproduce it in full in this supplement. After all, we want to show you some of the other prize-winning entries, too!

The IC tester project will be described as a project for home construction in a future issue of *Elektor Electronics*, complete with a printed circuit board, a ready-programmed EPROM and tested project software on disk.

Meanwhile, the design (as we received it) will be included on a Contest Compilation CD-ROM which we hope to publish by the end of January 1998. (982020-1)

Development tools used

IC Tester firmware:	Metalink 8051 Cross Assembler V1.2h
	Signetics HEX2BIN.EXE
PC programs:	Borland Turbo C++ V3.0
Schematic layout[1]:	Protel Advanced Schematic V3.5
PCB layout[2]:	Easytrax V2.06
GAL assembler:	c't Galio

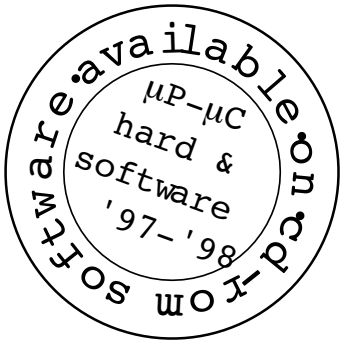
[1] cosmetic changes by Elektor
[2] layouts not reproduced in this article

FIRST NATIONAL PRIZE UK

1997 International Microprocessor Competition

PIC on the Rocks is a system for PIC users that allows them to do in-circuit debugging of PIC programs. It uses a simple serial connection with a host PC running Windows95 or NT. The PIC program the user wants to check also needs to embed a small piece of telemetry code. This code enables the PIC side of the telemetry link. The project software includes source code for both the PIC and the host PC. It also comprises a manual in Word format and some examples.

By Ben de Waal (USA)



PIC on the Rocks

The PIC micro controller range from Microchip has become one of the leading micro controllers used by engineers and hobbyists to solve simple and medium complexity embedded problems. Personally I have used PICs in pagers, Virtual Reality peripherals, home security systems and even hamster treadmill measurements.

One of the major advantages of PICs is that they are relatively inexpensive. The basic development tools are also fairly cheap and this makes it easy for the hobbyist (*Elektor Electronics* reader) to develop software for it. The major drawback is that PIC debugging tools and In Circuit Emulators (ICE) are very expensive. This creates the common scenario that average Joe Hobbyist has the development tools but also has to burn and erase endless number of times to get the solution 'just right'. Most of the time he tries to figure out 'what goes on' inside the PIC since he cannot afford or justify the cost of an ICE.

This project addresses the PIC debugging issue by providing an almost non-intrusive solution for PIC telemetry (remote debugging). It allows you to pause PIC execution, look at the internal registers (files), edit them and continue till the next 'break point', etc.

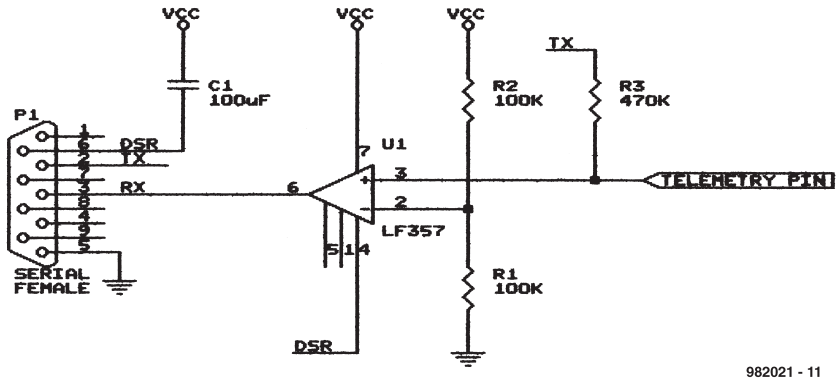


Figure 1. The opamp solution. Note: depending on the serial cable, you may have to swap pins2 and 3 as well as pins4 & 6. DSR should be an output.

The best part is that the total cost for the hardware components can be less than one dollar!

The telemetry system works in two possible ways. The first enables you to place break point instructions at selected areas in your source code. You program the PIC, place it in-circuit and whenever the PIC reaches the break point, it will declare itself to the PC host and start a telemetry session. From the PC you can examine and control the current state of the PIC and let it continue until the next break point is reached. This solution allows full speed execution (especially great in timing sensitive applications – 80% of them are) but also allows a view into the heart of the PIC.

The second approach is a variation of the first. Basically the set-up is the same but the communications pin is set up to interrupt the PIC. This way, instead of placing break points, you can actually manually break it at desired locations. This only works for PICs with interrupt capabilities.

How it works

You select one pin on either port A or port B (or other on PIC17Cxx series) as being the telemetry pin. This pin will talk to the host using standard serial communication protocol. This pin is available for general use while not in telemetry mode but some restrictions apply.

The telemetry code was designed for the PIC16C71 and is placed at 0x380 by default. With little work you can modify it to work with any unit in the PIC family.

The code that you are debugging stays exactly the same, except that you include a couple of lines at the end of the program, as shown in **Listing 1**.

Remember to substitute the correct values for your application into the constants defined. If you have trouble with high baud rates, select a higher clock or lower the baud rate. Keep in mind that the PIC has to emulate a UART.

You are also free to choose any pin. Depending on whether you need interrupt capabilities, you might be restricted by the choice.

Apart from the simple hardware interface you have to provide (described below), you are ready to go. A break point is signalled by a **call OTRTELEMETRY** instruction. The example in **Listing 2** shows how to use this.

Manual interrupt configuration (method 2, described above) requires you to save the *W* and *STATUS* registers in special places. This lets the host know where to look for them when you examine them. Usually you save these registers anyway as a first step in servicing interrupts.

The telemetry pin you select should be configured as a negative edge interrupt pin. The host 'break' command will signal the interrupt. A typical ISR is shown in **Listing 2**. The directory **\OTR\EXAMPLE** contains an example application using the 'break point' approach.

```
OTR_CLOCK_SPEED     = .4000000      ; PIC clocks at 4 MHz
OTR_BAUDRATE        = .9600         ; desired COMMs speed (8 data, 1
                                      stop)
OTR_PORT            = PORTB         ; desired telemetry COMMs port
OTR_PIN             = 0             ; desired telemetry COMMS pin

    include "\OTR\INC\71OTR.ASM" ; exact path may vary
```

Listing 1. Code snippet to place at end of application program.

```
ISR_START:
; save W & STATUS
    movwf  OTR_SAVE_W
    swapf  STATUS,W
    movwf  OTR_SAVE_STATUS

    < code to identify proper interrupt comes here only if other
      types of interrupts can also occur - depends on your appli-
      cation >

    < You may need to save the FSR & other registers that the OTR
      will corrupt - depends on whether you use them or not >

; initiate telemetry
    call  OTR_TELEMETRY_INT

    < Restore the FSR or other registers >

; restore W & STATUS
    swapf  OTR_SAVE_STATUS,W
    movwf  STATUS
    swapf  OTR_SAVE_W,F
    swapf  OTR_SAVE_W,W
; done
    retfie
```

Listing 2. Interrupt Service Routine (ISR) to enable telemetry on interrupt.

Options

Figure 1 shows an opamp solution. I have used this successfully with JFET input rail-to-tail type opamps. The opamp swings to + 5 V (PIC supply voltage usually) and is less noise immune than a swing to + 10 V or more. If you have a + 12V supply handy, please use it. Please see considerations below on the DSR line.

Figure 2 shows my favourite p-n-p transistor solution. Usually you need to play with the resistor values in order to get it to work just right given wire lengths and serial port power capabilities and transistor beta. The values shown work good enough with an acceptable margin. Keep in mind that the output also swings to + 5V resulting in a lower noise margin. It is more tricky to make it swing to + 12 V since we turn

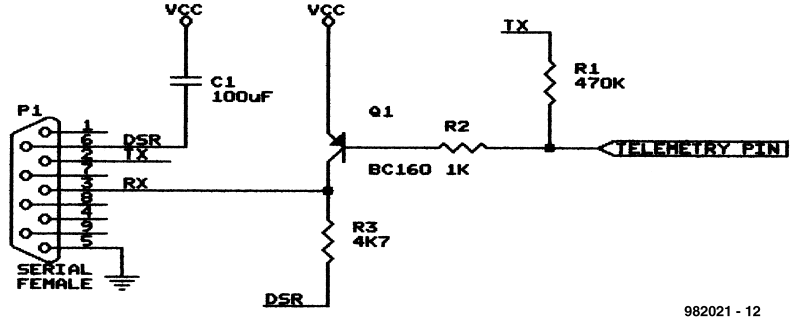


Figure 2. PNP transistor solution. See note with Figure 1.

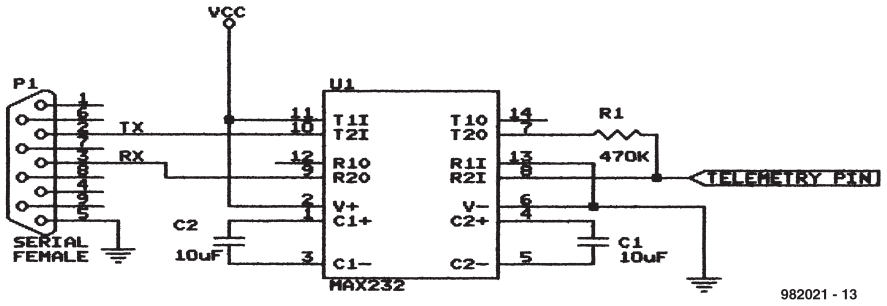


Figure 3. Formal solution using a MAX232. See note with Figure 1.

the p-n-p transistor off with the PIC. Please see considerations below on the DSR line.

Figure 3 shows the formal way using a MAX232 (or similar). **Note** that the MAX232 is a special logic NOT gate. This means that the standard code will send the wrong signals (inverse). To correct this you have to modify the telemetry code at all the places indicated with 'MAX232' in the source code.

Host program

The host program functions very much like the *DEBUG* program under *DOS*. Only a subset of DEBUG's commands are implemented. I have provided the source code to allow easy modification and extension.

The host program establishes communications with the PIC. When the PIC does not respond, the host assumes it is not in telemetry mode. This is because the PIC is busy with other things (like executing your program!). When the PIC goes into telemetry mode, it reports itself to the host. The host will then enter command mode.

Command mode is identified by a '>' prompt with the cursor flashing next to it. From here you can issue commands from the keyboard.

The complete **Command Summary** may be found in the documentation file. The program accepts and outputs all numbers only in hexadecimal.

There is no hidden memory translation. This means that when you read file 3 (the *STATUS* register), you will get its current value (somewhere in the telemetry code), not the value it had when the PIC suspended execution. The *W* and *STATUS* registers are relocated to addresses *0x2d* and *0x2c* respectively. Modifying those files will indirectly modify the registers.

You may modify any PIC data memory you want from the host program. Nothing inhibits you from changing a port pin from input to output, or to initiate an A to D conversion, reprogram the timer, etc. The only restriction is that the actual telemetry pin bit will always read as a 0. Any attempt to manually modify it will be ignored.

Considerations

While using the PIC telemetry code, please keep the following in mind:

- The stack must have 3 or more entries available. This is especially important to consider when you use the manual interrupt method since you do not exactly control where it breaks.
- The break point instruction will corrupt the *FSR* and *PCLATH* files. If you are using these you have to back them up before calling telemetry.
- Interrupts must be disabled before starting telemetry since the serial communications is time sensitive. Failure to do so will result in telemetry errors and a possible 'PIC system crash'.
- The code occupies just short of 128 program locations. The default address is at 0x380.
- The code uses 4 memory locations (0x2c through 0x2f). They are available as general memory while telemetry is not performed but will be corrupted during telemetry.
- The code does observe the watchdog so you may have it enabled during telemetry.
- You cannot use the RC oscillator on the test PIC since it is too inaccurate (unstable) for proper serial communications. It would probably also be too slow for a decent baud rate.
- To single step you may put break commands between each instruction of the code portion you want to investigate.
- The code was designed for a *PIC16C71*. You may have to fiddle with it a little to make it work on the other PICs in the family.
- The PIC restricts reading and writing certain memory locations. These restrictions still apply for telemetry. Look at the PIC data sheet for more information.
- Certain functions will not halt when PIC goes into telemetry mode. This include the Timer, an A to D conversion in progress, etc.
- Sometimes the DSR line will not properly drive low (−13 V). This will impair communications. If you have a ±12 V power supply, I suggest you use circuit 1 (with the op-amp) and drive pin 7 to + 12 and pin 4 to −12. Do not connect the DSR line. Otherwise, use the MAX232. (982021-1)

Development tools used	
MS Visual C++ 4.2	PC host program
Microchip MPASM	PIC examples & OTP code
Microchip PICSTART-16B	Burning ceramic PIC16C71/JW units

All prize-winning entries on CD-ROM

over 100 projects as supplied by the Contest participants

treasure-trove for microprocessor enthusiasts

The five projects presented in this month's Supplement represent only a tiny fraction of the vast number of microprocessor-related projects we received as entries for our 1997 Microprocessor Design Contest.

As the staff of each of the four language groups (Dutch, French, German and English) awarded more than 25 prizes donated by our advertisers,

the total number of prize-winning designs is over 100! All these designs have been collected and transferred on to a CD-ROM which will be available through our Readers Services by the end of January 1998. The tile of the CD-ROM is 'μP-μC Hardware/Software 97-98', order code 986001-1, price £16.50. Users of this CD-ROM should note that all projects are supplied *as is*, that is, they have not been extensively tested by our laboratory as is usual with projects submitted by free-lance authors. Also, some of the file formats used by the Contest competitors may be unusual. In all cases, however, a contact address is provided, and we encourage information exchange between authors and CD-ROM users. A 'readme' file on the CD-ROM provides the name of each project and its author, along with its location in a sub-directory on the CD-ROM.
As a matter of course, all projects come complete with unabridged source code files (PIC, 8051/32, 8751, Atmel AVR, 68HC11, Z80 and many others), so here's a unique opportunity for you to 'learn the tools of the trade'. Certain authors have also supplied the associated assemblers, compilers or debuggers, or information on how these may be obtained. Some projects are ready-to-go and suitable for direct reproduction, while others may serve as a source of inspiration for your own applications.

Highlights on the CD-ROM include: IC Tester (Int.), Gyroscope (F), Cable Tester (G), PIC on the Rocks (UK), Penelope Robot (UK), AVRc Parallel Programmer (UK), Solar Controller (G), Intelligent Cable Tester (F), Video DVM (UK), PIC Simulator (G), Speech Controller (G), Monitor Refresh Meter (NL), Locomotive Decoder (NL), 3-Phase Sinewave Generator (F).

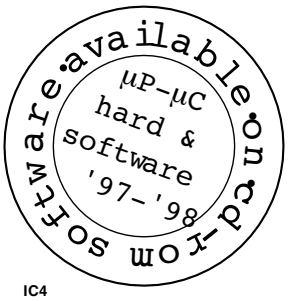
FIRST NATIONAL PRIZE GERMANY

1997 International Microprocessor Competition

Cables for information traffic are usually marked by a large number of wires, which have to be connected in the proper order. If all wires have the same colour, or if the cable ends are at a considerable distance from each other, a lot of patience is needed to establish the wire allocation.

by U. Reiser (Germany)

Cable Tester



The cable tester described here is capable of performing a continuity test on 12 wires at a time, and find out which wire ends belong together. This is achieved by installing a transmitter device at one end of the cable, and a receiver, at the other. The transmitter provides 12 outputs which supply different pulse codes. The receiver detects the code and indicates it on a 7-segment LED display. Apart from the wire under test, another connection has to be present between the transmitter and the receiver (cable braid, screening, or a reference wire). To help you find this additional connection, the receiver features an acoustic continuity tester.

Transmitter and receiver

Both the **transmitter** and the receiver are based on a RISC microcontroller (PIC) type 16C84, which was chosen because it slashes the component count in these circuits. Outputs RA0 through RA3 and RB0 through RB7 of the

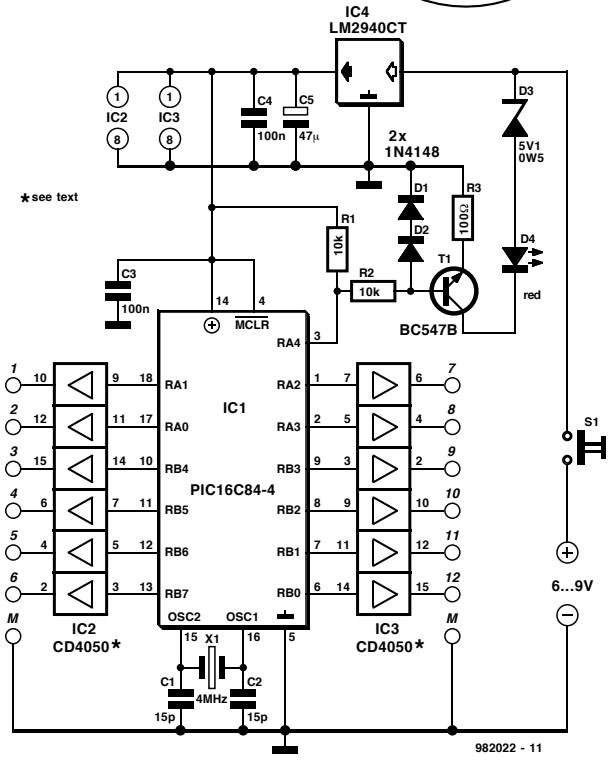


Figure 1. Circuit diagram of the transmitter.

the total number of prize-winning designs is over 100! All these designs have been collected and transferred on to a CD-ROM which will be available through our Readers Services by the end of January 1998. The tile of the CD-ROM is 'μP-μC Hardware/Software 97-98', order code 986001-1, price £16.50. Users of this CD-ROM should note that all projects are supplied *as is*, that is, they have not been extensively tested by our laboratory as is usual with projects submitted by free-lance authors. Also, some of the file formats used by the Contest competitors may be unusual. In all cases, however, a contact address is provided, and we encourage information exchange between authors and CD-ROM users. A 'readme' file on the CD-ROM provides the name of each project and its author, along with its location in a sub-directory on the CD-ROM.
As a matter of course, all projects come complete with unabridged source code files (PIC, 8051/32, 8751, Atmel AVR, 68HC11, Z80 and many others), so here's a unique opportunity for you to 'learn the tools of the trade'. Certain authors have also supplied the associated assemblers, compilers or debuggers, or information on how these may be obtained. Some projects are ready-to-go and suitable for direct reproduction, while others may serve as a source of inspiration for your own applications.

Highlights on the CD-ROM include: IC Tester (Int.), Gyroscope (F), Cable Tester (G), PIC on the Rocks (UK), Penelope Robot (UK), AVRc Parallel Programmer (UK), Solar Controller (G), Intelligent Cable Tester (F), Video DVM (UK), PIC Simulator (G), Speech Controller (G), Monitor Refresh Meter (NL), Locomotive Decoder (NL), 3-Phase Sinewave Generator (F).

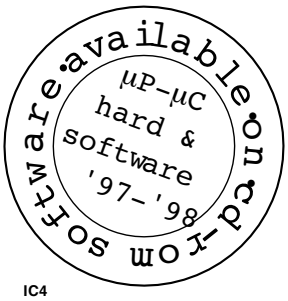
FIRST NATIONAL PRIZE GERMANY

1997 International Microprocessor Competition

Cables for information traffic are usually marked by a large number of wires, which have to be connected in the proper order. If all wires have the same colour, or if the cable ends are at a considerable distance from each other, a lot of patience is needed to establish the wire allocation.

by U. Reiser (Germany)

Cable Tester



The cable tester described here is capable of performing a continuity test on 12 wires at a time, and find out which wire ends belong together. This is achieved by installing a transmitter device at one end of the cable, and a receiver, at the other. The transmitter provides 12 outputs which supply different pulse codes. The receiver detects the code and indicates it on a 7-segment LED display. Apart from the wire under test, another connection has to be present between the transmitter and the receiver (cable braid, screening, or a reference wire). To help you find this additional connection, the receiver features an acoustic continuity tester.

Transmitter and receiver

Both the **transmitter** and the receiver are based on a RISC microcontroller (PIC) type 16C84, which was chosen because it slashes the component count in these circuits. Outputs RA0 through RA3 and RB0 through RB7 of the

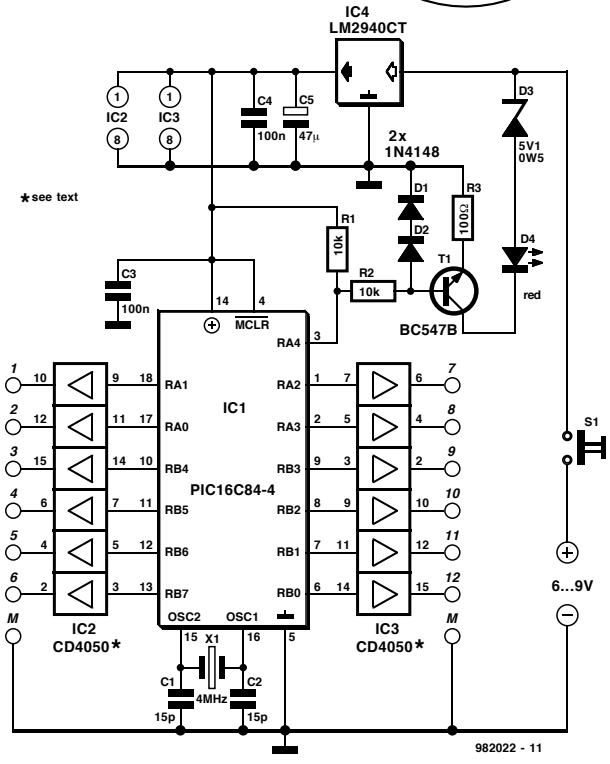


Figure 1. Circuit diagram of the transmitter.

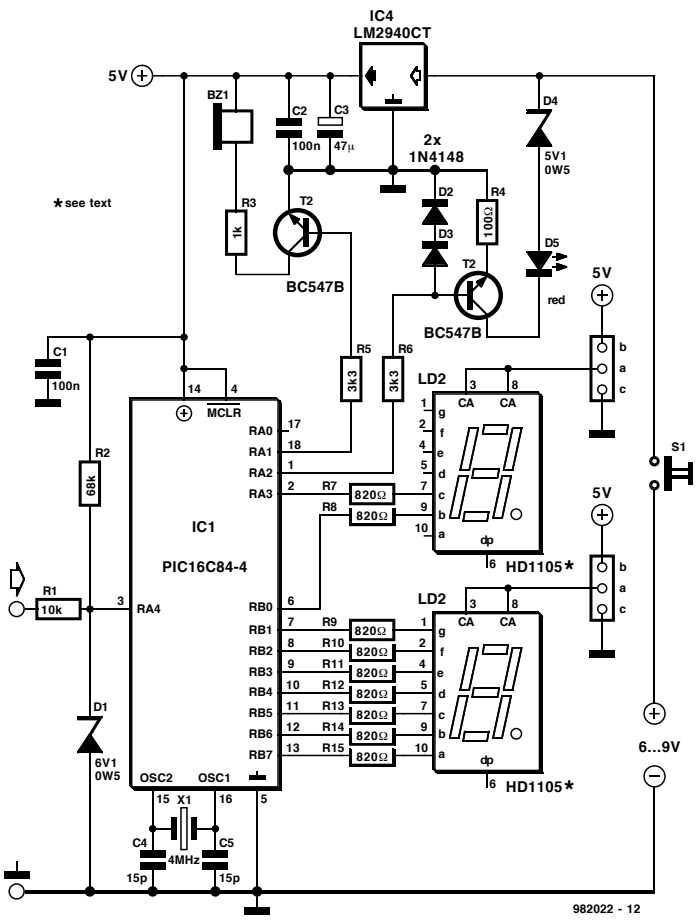


Figure 2. Circuit diagram of the receiver.

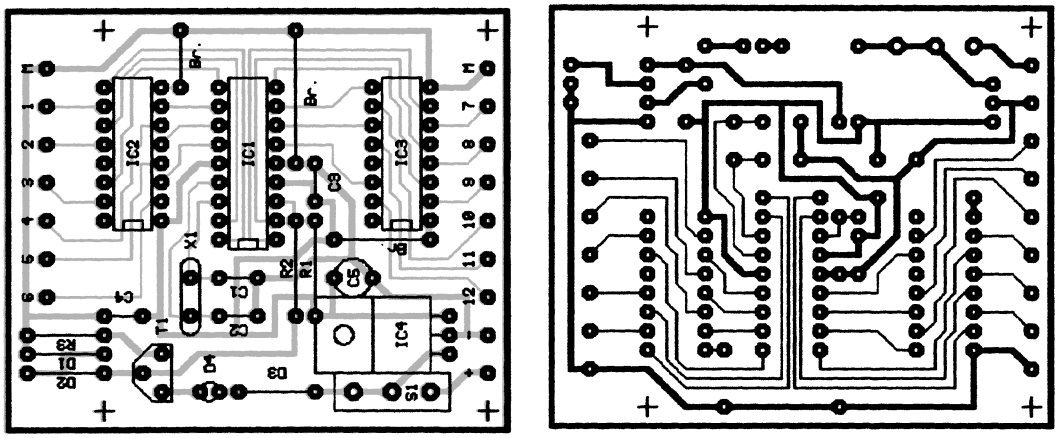
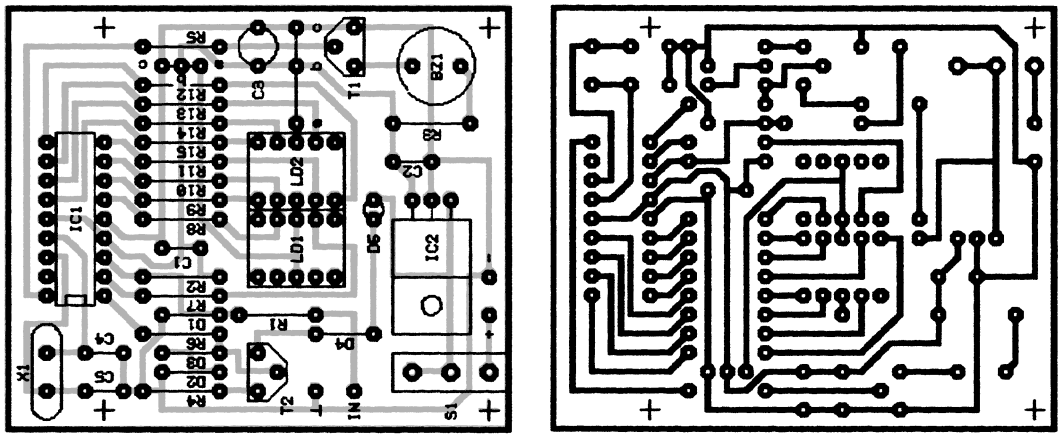


Figure 3. PCB copper track layouts and component overlays.



transmitter shown in **Figure 1** supply 'active low' pulses with a length of 1 ms, the number of pulses being dependent on the port line number. Logic high pauses, also with a length of 1 ms, are inserted between the active-low pulses. To make sure the transmitted code sequences are properly copied at the far end of the cable, a run-in start bit is transmitted (1 ms low/1 ms high), as well as a stop bit, which is a 4-ms long pause (high). The total length of a pulse sequence is 29 ms.

The PIC outputs are connected to drivers (type 4049 using the TRANSM49 program, or type 4050 if you use the TRANSM50 version). These buffers ensure that sufficient drive current is available for relatively long cables. By way of constant-current source T1, output RA4 (RTTC) causes the 'power on' LED, D4, to flash at a rate of 0.5 s (i.e., during the 'high' pause of the test routine). The zener diode makes it impossible for the LED to light when the battery voltage drops below the zener voltage. In this way, the LED also acts as a useful 'low-battery' indicator. The transmitter operates off a regulated 5-V supply, with only the 'on' indicator being powered directly by the battery.

In the **receiver (Figure 2)**, the transmitter signal is evaluated by a PIC16C84. The signal picked up by the test probe is applied to Schmitt-trigger input RA4 via a current-limiting resistor, R1. R2 pulls the non-used input high, while D1 eliminates input voltage surges. The PIC is constantly busy evaluating the received pulse trains. If a particular sequence is recognised twice in a row, the relevant wire number appears on the two-digit LED display. If the test probe remains logic high longer than 3 s, the display is cleared and the 'power on' LED flashes (current saving mode).

Pulses (1 ms low, 1 ms high) are recognised by means of three level measurements. In standby mode (input high) the software executes a delay loop. If a low level is detected, the signal level is checked again during the pulse time (after 0.776 ms), to

COMPONENTS LIST
TRANSMITTER

Resistors:
R1,R2 = 10kΩ
R3 = 100Ω

Capacitors:
C1,C2 = 15pF
C3,C4 = 100nF (5mm)
C5 = 47μF 10V

Semiconductors:
D1,D2 = 1N4148
D3 = zener diode 5v1, 0.5W
D4 = low current LED, red, 3mm
T1 = BC547B
IC1 = PIC16C84-04
IC2,IC3 = 4049 or 4050
IC4 = LM2940CT

Miscellaneous:
X1 = 4MHz quartz crystal
S1 = miniature switch, on/off, PCB mount
Case (Conrad 522864-77)
14 test prods (Conrad 737691-77)
9-V PP3 battery with clip-on lead

COMPONENTS LIST
RECEIVER

Resistors:
R1 = 10kΩ
R2 = 68kΩ
R3 = 1kΩ
R4 = 100Ω
R5,R6 = 3kΩ3
R7-R15 = 820Ω

Capacitors:
C1,C2 = 100nF (5mm)
C4,C5 = 15pF
C3 = 47μF 10V

Semiconductors:
D1 = zener diode, 6V 0.5W
D2,D3 = 1N4148
D4 = zener diode, 5V1, 0.5W
D5 = low current LED, red, 3mm
T1,R2 = BC547B
IC1 = PIC16C84
IC2 = LM2940CT

Miscellaneous:
X1 = 4MHz quartz crystal
S1 = miniature switch, on/off, PCB mount
LD1,LD2 = HD1105
Bz1 = d.c. buzzer, 140Ω
Case (Conrad 522864-77)
2 test prods (Conrad 737691-77)
9-V PP3 battery with clip-on lead

make sure the first pulse was not caused by interference. After a second period of 0.776 ms, the software checks that a logic high level is present. If so, the pulse is considered valid. Next, it is registered in a counter, and then, after a 'high' period of at least 2 ms, compared with the result of the previous count. If the result is a match, the wire number is sent to the LED display. If not, the previous result is overwritten by the current one.

Apart from the pulse interrogation and the high-level monitoring, the low duration at the IN input is also under constant examination. If the input remains low longer than 48 ms, the display is cleared, D5 is switched off, and RA0 supplies a 2.5-kHz signal to T1. The transistor, in turn, drives a piezo sounder. This is the previously mentioned continuity tester function of the circuit. Meanwhile, the software examines the input every 0.4 ms. If a high level is detected, the routine is left.

The transmitter and the receiver are each powered by a 9-V PP3 battery. Both circuits incorporate low-drop voltage regulators which enable the batteries to be used until they are almost 'flat'.

Construction

The two copper track layouts and component mounting plans are shown in **Figure 3**. The only point to note in the construction of the transmitter is that the assembly code file matches the buffer IC you want to use (4049 or 4050). The receiver board allows you to use either common-cathode or common-anode 7-segment displays. In the first case, you fit wire jumpers a-c, else, a-b. Here, too, two different programs are available (recei**ano**.asm and recei**cat**.asm), and your choice must match that of the displays fitted on the board.

Having finished the soldering work, you may connect the circuits to a benchtop supply and check that the supply voltage is 5 V behind the low-drop regulators. You may also check the operation of the 'Low-Battery' function by turning down the supply voltage. If the 'on' LED starts to flash after you switch on the transmitter, you may safely assume that this circuit is functional. The different pulse trains supplied by the transmitter outputs are easily observed if you have an oscilloscope.

At power-on, the receiver first performs a test routine on the displays. If everything works as it should, the following display segments are switched on one after another, at 0.6 s intervals: 0b, 0c, 0a, 0d, 0g, 0e, 0f, 1b, 1c. The piezo buzzer sounds when the last segment lights. The 'on' LED also lights all the time. As a further test, you should connect the test probe to ground, whereupon the buzzer should sound.

The circuits are fitted in ABS (strengthened plastic) cases (101×60×26 mm) having a battery compartment. In the transmitter case, drill one 6-mm hole for the miniature toggle switch, fourteen 1-mm holes for the outputs plus ground, and one 3-mm hole for the LED. The test wires are made from short lengths of light-duty flexible wire and miniature test prods or crocodile clips. Inside the transmitter case, each wire should have its own strain relief in the form of a knot.

In the receiver case you should drill holes for the switch and the LED. Also, a 20 by 13 mm rectangular clearance is required for the 7-segment displays. Two additional 1-mm holes are required for connecting wires to the test probe and the ground connector. As with he transmitter, these wires should have knots acting as strain reliefs at the inside of the case. (982022-1)

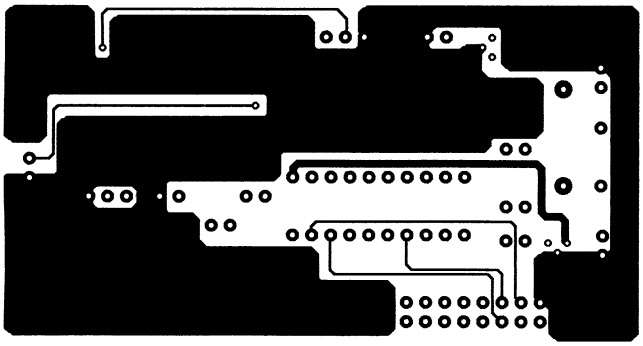
In not a few cases, PC users complaining about headaches and eye strain are using a too low picture refresh rate on the PC display. The circuit discussed here measures the refresh rate using a photodiode stuck on the display screen, and indicates the value on an LC display. The instrument also allows the horizontal (line) frequency to be measured. This is achieved by inserting the refresh meter between the VGA card and the display. Both the horizontal and the vertical picture frequency of the VGA signal are then indicated on the LCD. To save current, the refresh meter is provided with an automatic shut-down function.

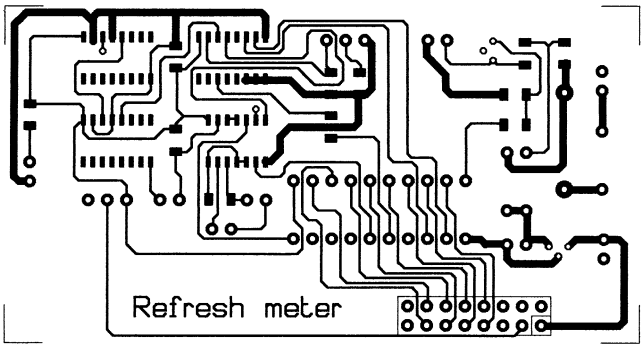
By H. Vos (Netherlands)

Display Refresh Meter

Figure 1. The circuit diagram of the display refresh meter contains just a few ICs.

The circuit is built around a member of the new AVR RISC processor family from Atmel. The development software used for the project may be found on the Atmel's Internet site, *www.atmel.com*. The package consists of an assembler (WAVRASM v. 1.11), a debugger (AVR Studio v. 1.01 for Windows95/NT) and the AVR development board software (v. 1.15). The microcontroller may be programmed by way of an SPI bus (serial programmable interface). The development board and the associated software may be obtained from Atmel's sales offices.





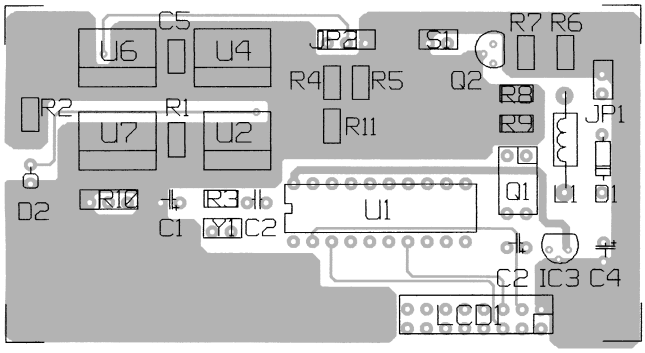


Figure 2. The PCB artwork shows that the design is partially based on SMA components.

The hardware

The controller type AT90S1200 from Atmel offers 15 I/O lines, 1 Bkytes of Flashable program memory, 64 bytes EEPROM and a variety of infernal devices including a timer and a watchdog timer. In the present design, the essential elements are the 8-bit timer and the external interrupt input. The timer is used as a 1-second timer to count pulses. The external interrupt input is used as a counter input. The clock frequency in the circuit is set at 32.768 kHz. This value results in low power consumption, and is convenient when it comes to calculating the divisor for the 1-second timer. Using the 4051 data selector, the desired input signal is selected: this may be optical (D2) or the horizontal or vertical sync pulse supplied by the VGA card. The horizontal frequency is first divided by 100 in U5 and U6 before it is applied to the INT0 input of the AVR chip. The actual counting operation is performed by an interrupt routine which is launched by the signal applied to the INT0 pin.

RC combination R1-C1 arranges the power-on reset of the microcontroller. The power supply may strike you as somewhat unusual. If the battery is connected-up and S1 is not pressed, the circuit is off. The base of Q2 is pulled to ground by R8, causing Q1 and Q2 to be switched off. Press S1 and FET Q1 starts to conduct, enabling the circuit to receive its supply voltage via regulator IC3. Shortly afterwards, the control program executed by the micro pulls the PD6 pin logic high, causing Q2 to conduct and take over the function of S1. In other words, the circuit remains on when S1 is released. If the software does detect an input signal which remains present for about 30 seconds, the PD6 line pulled low. As a result, Q2 is switched off, and with it the supply voltage.

About the software

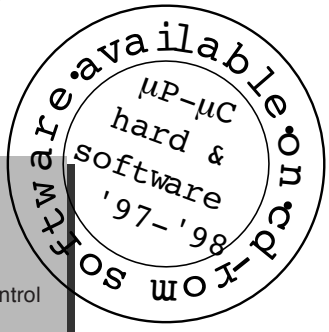
The program loaded in the AVR micro is fairly simple. When the circuit is switched on, a routine is launched that scans the inputs (optical and vertical sync) for signal activity. Each input is monitored 250 ms. If a signal is found at one of these inputs, the system switched to that input, and the frequency of the measured signal appears on the LC display. This continues until the input signal disappears. If no signal is detected for half a minute, the circuit is automatically switched off.

(982023-1)

This article describes a circuit which the Jury found particularly interesting because it employs a novel component: a vibration piezo-electric Gyrostar® sensor from Murata®. Here, the gyroscopic sensor is coupled to a PIC16C71, the combination acting as a real gyroscope which may be used for stabilising a model car, vessel or plane.

By P.L. Destin (France)

PLD Gyroscope Version II



Technical characteristics

- True closed loop for rotation control
- Auto-calibrate at power-on
- Remote control for gain (gyroscopic sensitivity)
- 'Clutch disengage' mode with direct retransmission and no restrictions to rotation control
- Watchdog with reset and automatic return to neutral when radio signal fails
- Axial-vibration resistant
- Low power consumption
- High reliability without risk of wear and tear
- Aluminium case protects against dust and water ingress, also reduces electrical noise
- Low cost and weight

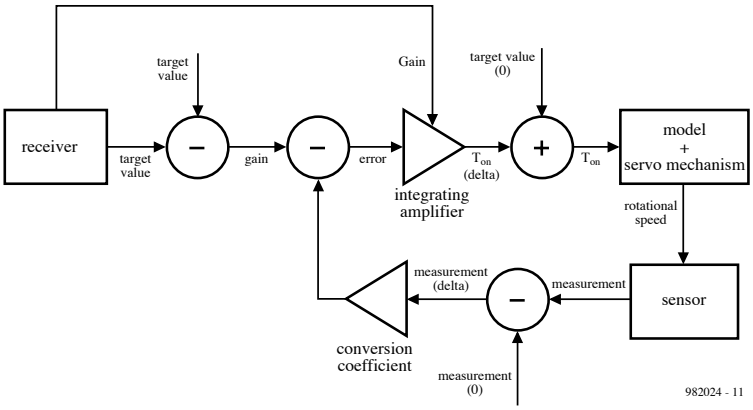


Figure 1. Basic structure of the regulation loop applied in the gyroscope.

The operation of the present gyroscope is based on a regulation loop as shown in **Figure 1**. One channel of the radio control receiver supplies the target value of the rotation. The other channel supplies the gain control information. The target and gain data consist of pulses with a variable length (1.5 ±0.4 ms). The loop compares the variation of the target value with the variation of the measurement result supplied by the Gyrostar rotation sensor. Depending on the difference between these two values ('error' signal), a pulse is produced with a length T_{on} (1.5 ±0.4 ms) whose function is to control the amount of rotation of the radio-controlled model.

The schematics

Figure 2 shows the schematic diagram of the PLD Gyroscope. Apart from the components already mentioned (PIC U2 and Gyrostar U4), there's not a lot that goes into this design: one voltage regulator, U1, and one opamp, U3! The voltage regulator is a low-drop type from National Semiconductor. The circuit comprises a number of 3-pin sock-

ets, which either serve to receive the servo-control connector, or provide the link with the voltage doubler module. Potentiometer P1 enables deviations in the sensor output voltage to be compensated (±1 V in the stand-by state). To adjust the pot, apply power to the gyroscope and turn the wiper until a voltage of about 2.5 V is measured at the opamp output (pin 6).

The schematic shown in **Figure 3** is that of a voltage doubler which may be required in case the present circuit is used in or on a model which is powered by a 4.8-V battery pack. The voltage doubler is not required if your R/C model uses a battery voltage of 6 volts or more. Basically, the circuit contains just a 555 (U1) and three pairs of sock-

ets which convey the servo, target and gain signals.

Construction

Obviously, you have to start by programming the PIC using a programmer for the PIC16C71. The 'watchdog' programming option has to be activated. To avoid problems caused by vibration and shock, it is best not to use IC sockets. Some of the solder spots on the board serve to effect through-contacting between the two board sides. To make sure you don't interchange the 'gain' and 'target' connections, it is recommended to cover the 'gain' plug in a length of heat-shrink tubing. For the following receiver brands, the ground and supply have to be reversed: Robbe, Graupner, Futaba and Lextronic. No problems are expected with the following brands: Multiplex, Space, Sanwa and Simprop.

The complete electronics, including the gyroscope and the (optional) voltage doubler, are fitted in an aluminium case with 5 cm long side panels. The gyroscope circuit board is secured to the bottom panel of the case using double-sided adhesive tape or velcro . The voltage doubler board is secured likewise to the cover panel. Except the jumper, the discrete components are firmly secured to the board by means of Neoprene glue (vibration!).

Installation

It is recommended to follow these steps in the indicated order:

1) Test the R/C model and the radio control without the gyroscope. Make sure all controls work as specified, from the transmitter right through to all servo actions. Pay special attention to the direction controls in case your model is a car or a boat, and to the tail rudder on helicopters and planes.
2) Install the gyroscope in the rotation axis of the model, with U2 and U4 being positioned in parallel with this axis.
3) Connect the gyroscope circuitry. Make sure you don't interchange any cables!
4) Switch on the transmitter.
5) Set the rotation and gain controls to their neutral positions.
6) Secure the model on to a stable support.
7) Without starting the propulsion or lift motor(s), power up the complete receiver installed in the model.
8) Wait a couple of seconds before moving the model or any controls on the transmitter.
9) Operate the direction control on the transmitter, and check that the relevant actuator on the model responds properly (wheels, rudder, tail compensation flap).
10) Carefully increase the gain, and pivot the model in your hand. Check that the direction control on the model (wheels, rudder, tail compensation flap) responds by moving in the opposite direction.

If everything seems to work so far, you may proceed with the final test. If not, you may

- remove the four screws and open the gyroscope case;
- remove the 'invert' jumper in the gyroscopic sensor;
- re-assemble the case by fitting the four screws of the gyroscope;
- start again at step 4 above.

11) Start the propulsion or lift motor(s), slightly increase the gain, and perform a final test.

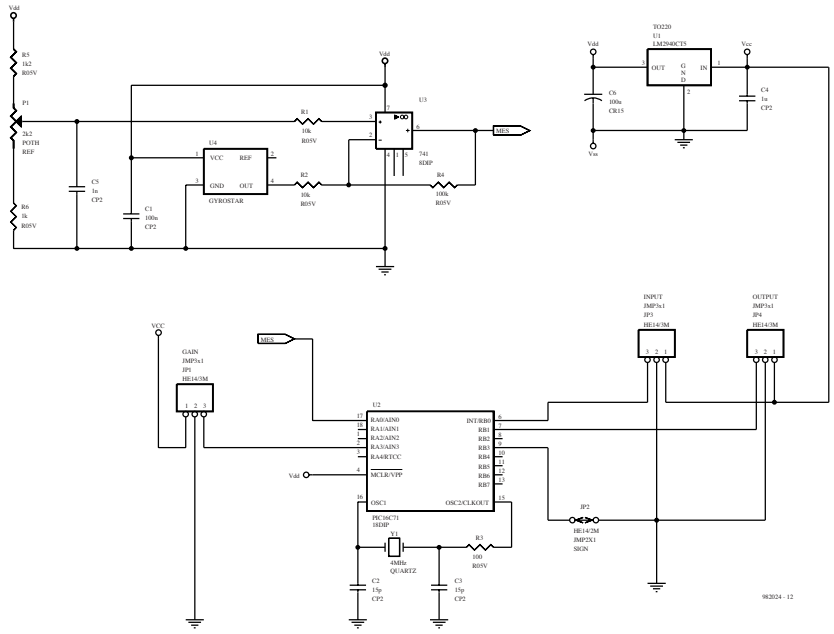


Figure 2. Circuit diagram of the gyroscope. A Murata 'Gyrostar' sensor is used.

Practical use

Getting started

1) Switch on the transmitter.
2) Set the rotation and gain controls to neutral.
3) Without starting the main motor, power up the complete receiver installed in the model.
4) Wait a couple of seconds before moving the model or any controls on the transmitter.

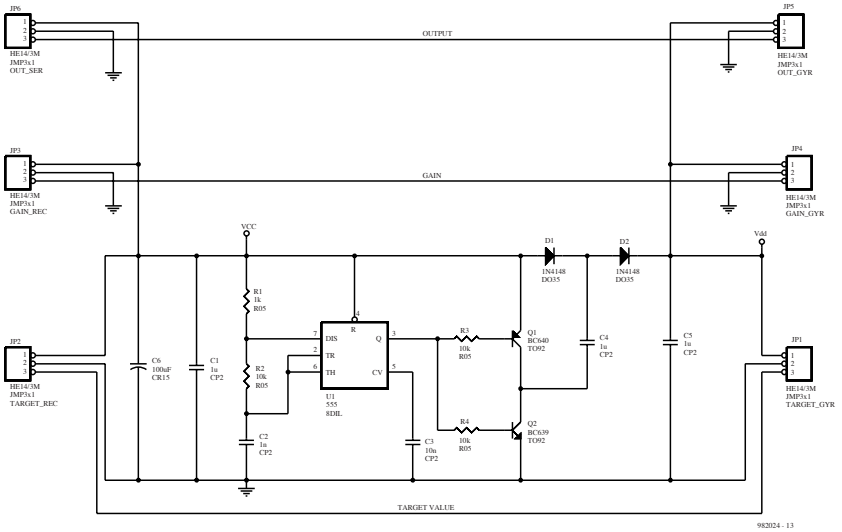


Figure 3. Circuit diagram of the voltage doubler. You only need this cct if the battery voltage in your R/C model is 4.8 V.

5) Operate the direction control on the transmitter, and check that the relevant actuator on the model responds properly (wheels, rudder, tail compensation flap).
6) Start the main motor.
7) Carefully increase the gain.

Direct control
If you set the gain control to its initial value (i.e., neutral) or a lower value, the gyroscope copies the original command without gyroscopic compensation. This allows, among others, to drive a model car or boat in reverse.

Gain control
Increasing the gain allows you to make the model accelerate and raise the force the model can exert against external perturbation (wind, current). By contrast, reducing the gain reduces the 'briskness' of the model, in particular, a boat's tendency to tack. Properly adjusted, the gyroscope achieves a compromise between these two situations.

Debugging
In case the circuit does not function as it should:
- Check that the transmitter works properly.
- Check that the transmitter battery is fully charged.
- Check that the receiver battery is fully charged.
- Switch off and restart the complete receiving system.

In the specific case of direction control problems:
- Verify the correct installation of the gyroscope.
- Temporarily change over to two other R/C channels for the direction and the gyroscope gain.
- Temporarily try out the use of another servo motor as the direction actuator.

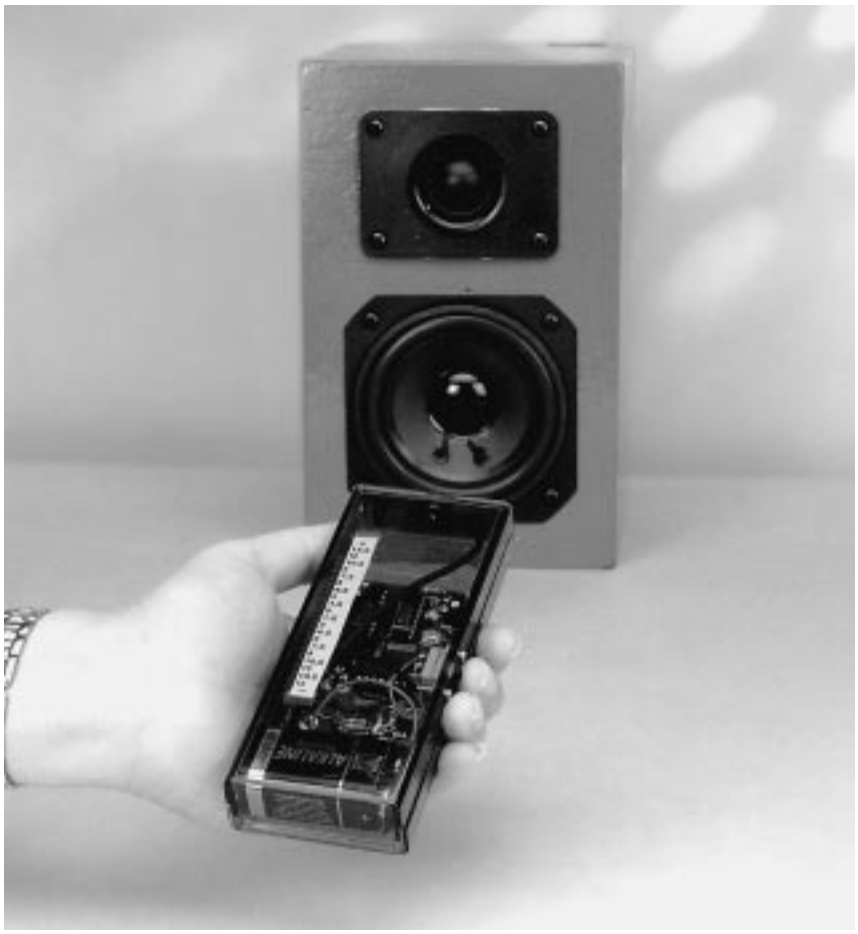
If no improvement is noted, there's no alternative but to check the gyroscope itself.

One final remark: the author advises that the circuit and the associated PIC resident program have been tested successfully on a radio-controlled model car. Lacking funds and a suitable model, he was unable to perform any testing on a model aeroplane or helicopter. What about *your* pioneering spirit? (982024-1)

portable sound-pressure meter

sound analysis with LED indication

Few audio enthusiasts possess, or have access to, equipment required for accurately measuring the performance of a loudspeaker or the acoustics of a given hall or room. The unit described in this article is an instrument that does not give the performance of a professional meter but, in conjunction with a test CD, makes possible fairly accurate sound level measurements that enable the frequency response of an acoustic system to be ascertained. Moreover, its small size makes it a very handy unit to carry about.



Most home workshops have facilities for measuring current, voltage and resistance as well as other generated electrical signals with the aid of an oscilloscope and a function generator. However, equipment for measuring the frequency response of a loudspeaker, consisting of at least a sweep-oscillator system, a level recorder and a standard microphone, is in most cases not available.

Because of this lack, many amateurs would like a simple, inexpensive instrument with which the performance of a loudspeaker in a given hall or room can be assessed. Such an instrument, which, of course, cannot have the accuracy of a professional meter, is presented in this article. It consists of a simple-to-build sound pressure meter, equipped with a 20-LED display and has a resolution of 1.5 dB. In conjunction with a suitable test CD, it forms a very useful, compact and affordable miniature sound-level meter.

** 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♭), and diminished third (C# up to E♭).*

Design by H. Baggen

1

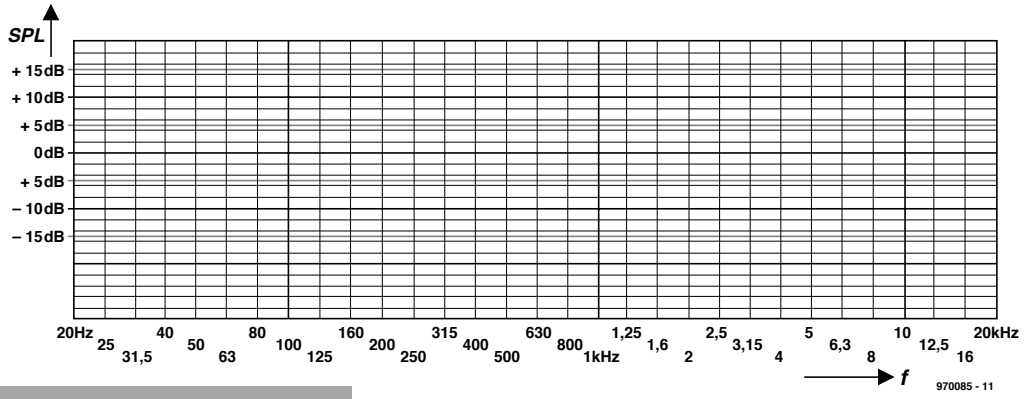


Figure 1. Model of a suitable graph paper to trace the measured frequency response of an acoustic system.

TEST SIGNALS

The circuit in **Figure 2** is, in essence, a fairly accurate sound level meter intended for carrying out relative, not absolute, measurements. Absolute sound levels are not of great import for determining the frequency response of a loudspeaker. What is of import is how the level of one sound compares with that of another, and that is a relative measurement.

In professional sound measurements, the sound source normally consists of an amplifier fed by a sweep-oscillator. Such a system is, however, not cheap and in the present circuit a much less expensive source, a test CD, is used.

Most test CDs contain 30 discrete third-octave signals in the range 20 Hz to 20 kHz. When the output level produced by each of these signals is measured in identical conditions and plotted on graph paper as shown in **Figure 1**, a somewhat coarse, but nevertheless usable, frequency characteristic is obtained of the loudspeaker being tested.

It should be noted that the 30 signals available from the CD are recorded from a sweep-oscillator system generating third-octave warbled sine waves or noise (since, for measurement purposes, noise is much more to music). The frequency of a swept signal is not constant but swings between two values that are separated by 1/3 octave, that is, a third*. Third-octave noise is pink noise filtered to such a degree that only the frequencies at intervals of a third are retained.

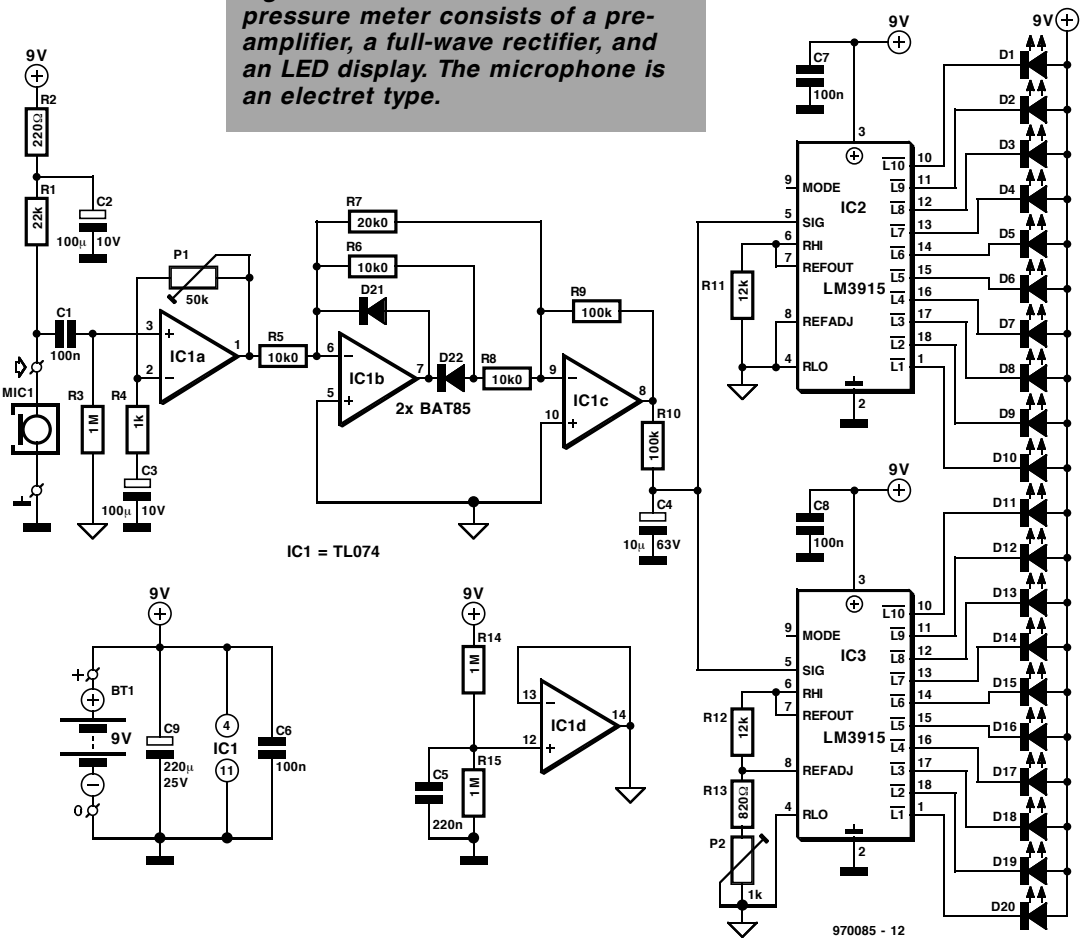
Third-octave noise signals are used to eliminate the effects of the hall or room in which the loudspeaker is tested. Since many frequencies are generated simultaneously, the standard (test) microphone registers their mean level and this results in the averaging of the room (hall) resonances, which makes them less obtrusive.

CIRCUIT DESCRIPTION

Designing a circuit that picks up sounds and displays their relative level with reasonable accuracy is not very difficult – see Figure 2. This circuit consists of three sections: a microphone preamplifier (IC_{1a}), a full-wave rectifier

2

Figure 2.The circuit of the sound-pressure meter consists of a pre-amplifier, a full-wave rectifier, and an LED display. The microphone is an electret type.



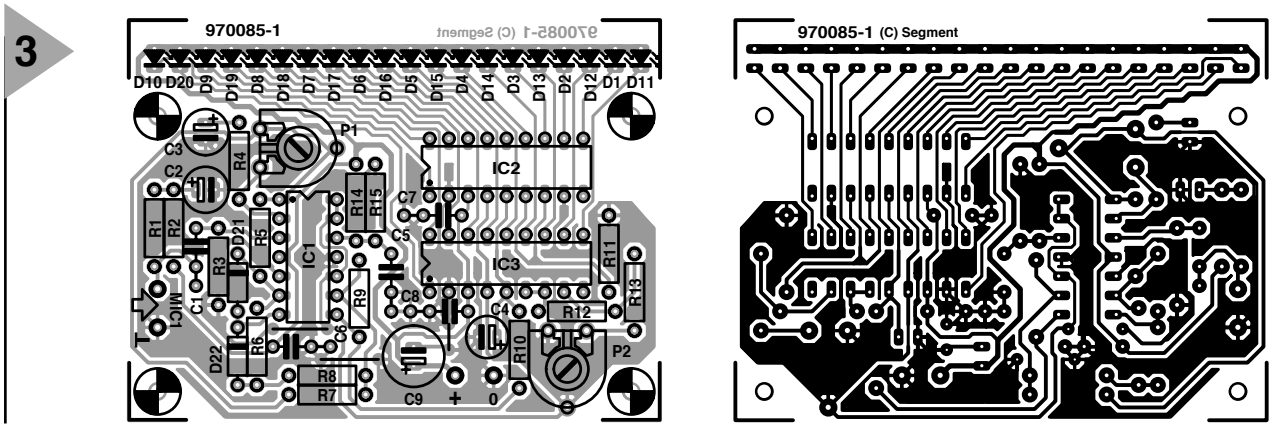


Figure 3. Printed-circuit board for the sound-pressure meter. Special care is needed when mounting the LED display.

Parts list

Resistors:
R_1 = 22 kΩ
R_2 = 220 Ω
R_3, R_{14}, R_{15} = 1 MΩ
R_4 = 1 kΩ
R_5, R_6, R_8 = 10.0 kΩ, 1% (but see text)
R_7 = 20.0 kΩ, 1% (but see text)
R_9, R_{10} = 100 kΩ
R_{11}, R_{12} = 12 kΩ
R_{13} = 820 Ω
P_1 = 47 kΩ (50 kΩ) preset
P_2 = 1 kΩ preset

Capacitors:
C_1, C_6, C_7, C_8 = 0.1 µF
C_2, C_3 = 100 µF, 10 V, radial
C_4 = 10 µF, 63 V, radial
C_5 = 0.22 µF
C_9 = 220 µF, 25 V, radial

Semiconductors:
D_1–D_{20} = LED, 3 mm, high efficiency
D_{21}, D_{22} = BAT85

Integrated circuits:
IC_1 = TL074CN
IC_2, IC_3 = LM3915N

Miscellaneous:
MIC_1 = electret microphone with rubber surround (e.g., MCE2000 from Monacor †)
BT_1 = 9 V battery with terminal clips
1 off single-pole on/off switch
Case: as desired - see text
PCB Order no 970085-1 (see Readers Services towards the end of this issue)

† Monacor
Inter-Mercador GmbH & Co, KG
Postfach 448 747
D-28286 Bremen
Germany
Telephone + 49 421 78650
Fax + 49 421 488415/488416

(IC_{1b} and IC_{1c}), and the LED display (IC_2, IC_3, and D_1–D_{20}). Op amp IC_{1d} is used for creating a virtual earth at half the supply voltage.

The microphone in this application must meet certain requirements, of course, even though it is used in a low-budget version of a sound level meter. It must, for instance, be fairly linear, otherwise the circuit cannot perform very well. It is therefore out of the question to use microphones of unknown origin with vague or uncertain properties. On the other hand, an expensive standard (test) microphone is a superfluous luxury. A good, linear electret microphone is a very good compromise between these extremes. The prototype instrument uses a type that is linear within ± 2 dB over the

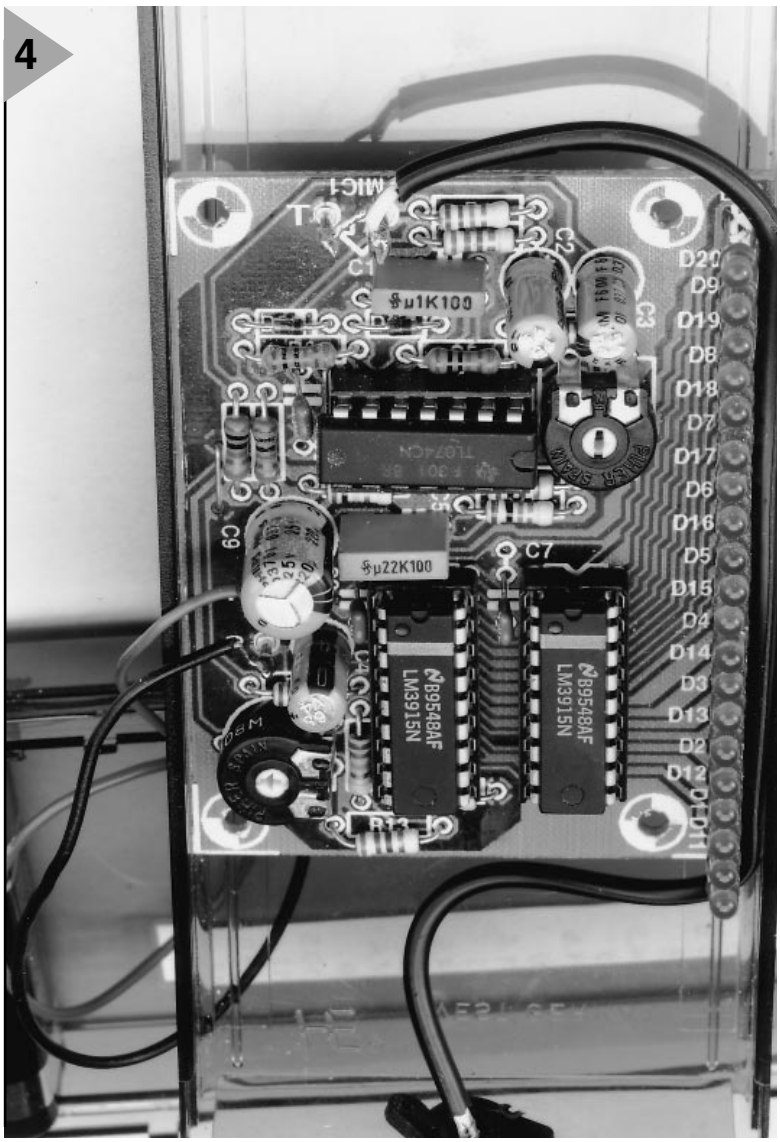


Figure 4. Inside view of the prototype sound-pressure meter in translucent case.

20 Hz to 20 kHz range.

The integral amplifier of microphone MIC$_1$ is held at about half the supply voltage with the aid of R$_1$. Network R$_2$-C$_2$ decouples the supply line.

The microphone signal is applied to preamplifier IC$_{1a}$ via C$_1$. The cut-off frequencies of networks R$_3$-C$_1$ and R$_4$-C$_3$ are sufficiently low to result in a measurement error at 20 Hz of not more than 0.1 dB. The amplification, and thus the sensitivity of the microphone, is set with P$_1$.

The output of the preamplifier is applied to a conventional full-wave rectifier, IC$_{1b}$ and IC$_{1c}$, So as to obtain a reasonable sensitivity without degrading the bandwidth of the instrument, IC$_{1c}$ provides additional ×5 amplification. This results in an enhancement of the accuracy of the measurements at high frequencies.

To ensure a stable display, the rectified signal is differentiated by network R$_{10}$-C$_4$ (which constitutes a large time constant) before it is applied to the LED display.

The display is driven by the well-known Type LM3915 drive (IC$_2$). This IC contains a voltage reference source, a precise potential divider, and ten comparators, each of which can drive an LED directly. The level of the input voltage to the driver is displayed by the LED array in ten steps of 3 dB each.

Of course, 3 dB-steps are rather coarse and the resolution has, therefore, been enhanced by the addition of a second driver, IC$_3$, whose reference has been shifted by 1.5 dB. This is done by making the potential at the REFADJ(ust) pin (8) ×1.1885 higher than that at the corresponding pin of IC$_2$. This makes D$_{11}$ the top of the decibel scale, followed by D$_1$, D$_{12}$, D$_2$, D$_{13}$, and so on. In other words, the LEDs driven by IC$_2$ and IC$_3$ are interlaced. This method has a slight drawback in that during measurements two LEDs light simultaneously: the correct test result lies somewhere between them. However, it was found that the operator quickly gets used to this.

POWER SUPPLY

Since the meter is intended for use as a portable instrument, the power supply must be battery-operated. The current drain is not greater than 19 mA, so that a 9-V battery will give about 100 hours service under normal conditions.

To ensure that op amps IC$_{1a}$–IC$_{1c}$ remain within their common-mode range for as long as feasible, all three are powered by half the supply voltage. This is effected with the aid of a fourth op amp, IC$_{1d}$, and potential divider R$_{14}$-R$_{15}$, which is decoupled by C$_5$. The output of the op amp, pin 14, constitutes a virtual earth at half the supply voltage above the real earth.

The reference pins of IC$_2$ and IC$_3$ are also connected to the virtual earth.

CONSTRUCTION

The instrument is best built on the printed-circuit board shown in **Figure 3**. The length of the board is determined by the dimensions of the display and the transparent case specified. It is, nevertheless, compact so that great care is required during soldering.

Potentiometer P$_1$ has intentionally been connected the wrong way around, that is, it has to be turned clockwise to reduce the amplification.

Note that although E96 type resistors are specified for R$_5$–R$_8$, E24 types may also be used if unavoidable. Their values should then be 11 kΩ instead of 10.0 kΩ or 22 kΩ instead of 20.0 kΩ as the case may be.

The prototype is housed in a translucent case, which has the advantage of not needing a cutout for the display. But, of course, any suitable or available case may be used, as long as the battery and finished board can be fitted comfortably inside.

Do not omit the rubber surround supplied with the microphone when fitting this on to the case. This surround damps spurious vibrations and makes the microphone less susceptible to reflections in or of the case.

The display must, of course, be given a suitable scale of ± 15 dB. The scale should have a 0 at its centre and this may be placed halfway between D$_5$ and D$_{16}$ as shown in **Figure 5**. The other markings at 1.5 dB steps are placed accordingly.

SHIFTING THE REFERENCE OF IC$_3$

Preset P$_1$ is set according to the test CD used, and this will be discussed in the next section.

Preset P$_2$ sets the 1.5 dB shift of the reference voltage to IC$_3$, for which an accurate digital voltmeter is needed. Measure the potential, U_{REF1}, across pins 2 and 7 of IC$_2$ and then connect the voltmeter across pins 2 and 7 of IC$_3$. Adjust P$_2$ until a meter reading of 1.1885U_{REF1} is obtained.

USAGE

Usable test signals to obtain a frequency characteristic as described earlier may be obtained from the following test CDs: *The Test* (Stax, AXCD 92001); *Compact Test* (Pierre Verany, PV 784031); *Hi-fi Check* (Stereoplay); and *CD-2Check* (Monacor 30.0180).

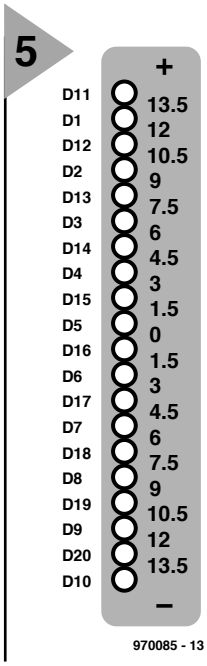
When the frequency response is being measured, it is advisable to place the loudspeaker well away (≥ 1 metre) from reflecting walls or other objects. Set the volume of the audio system to a level where the test signals are well above the ambient noise level at all times. When a suitable level has been found, adjust P$_1$ on the sound-pressure meter to obtain a 0 reading on the LED display.

Bear in mind that at frequencies below about 200 Hz, effects of the room or hall are so strong that the measured levels say hardly anything about the performance of the loudspeaker being tested. This may be checked by holding the microphone right in front of the loudspeaker. It will be found that a number of peaks and troughs measured earlier (at the normal test distance of 1 metre) disappear.

An impression of the acoustic performance of the room or hall may be obtained by repeating the response measurement at a distance of 3–4 metres from the loudspeaker. At that distance, there is no question any longer of a flat response!

[970085]

Figure 5. Example of two usable scales for the sound-pressure meter. The total measuring range spans 30 dB.



DC-DC Step-Up Converter

January 1999, 980073

Electrolytic capacitors C4 and C9 are shown with the wrong polarity in the circuit diagram, Figure 1. The relevant PCB lay-out is all right.

Portable Sound-Pressure Meter

January 1998, 970085

The text alongside Figure 5 should be changed to read: Measure the potential U_{REF1}, across pins 4 and 7 of IC2 and then connect a voltmeter across pins 4 and 7 of IC3.

Tachometer

October 1998, 980077

In the parts list and the circuit diagram, the value of R1 should be corrected to read 2.2 kΩ.

Introduction to digital signal processing

A new six-part course

Digital signal processing (DSP) is largely concerned with the application of Fourier and *z*-transform techniques to a wide variety of scientific and engineering problems. Since it involves a transformation of sampled data into the discrete frequency domain, the methods find use in both signal analysis and signal processing. Although DSP is very much a technique of the late 20th century, it is based on Nyquist's sampling theorem, which dates back to 1928. Although DSP techniques are highly mathematical, this new six-part course will be based on worked-out examples and experiments. For these, only a PC with soundcard and CD-ROM drive are necessary.

Box 1. Features

- *All examples may be executed directly*
- *Various wave files for analysis*
- *Variety of programs for many applications*
- *All source codes on the CD-ROM written in TURBO-PASCAL*
- *Batch files for complex experiments*
- *Simulation of complex transfer systems*
- *Construction of hardware not necessary*

This six-part course will endeavour to make the most essential basics of digital signal processing accessible to the reader by examples and experiments. All the reader needs is a middle-of-the-road PC equipped with sound-card and CD-ROM drive.

By M. Ohsmann

SEVERAL LEVELS

The simplest of these is simply to carry out the experiments without fully understanding the fundamentals underlying them.

The next one is to carry out new experiments at the hand of the program supplied The course includes helpful suggestions for these.

A further level is for readers to study the mode of operation of the program at the hand of the source texts and the explanations given. Following this, they can write their own program(s) for signal processing and design their own experiments.

Finally, they may try to transfer the course material to a specific signal processor. It should be noted, however, that this requires much experience of programming since there are a number of obstacles in the way, such as scaling, differential arithmetic, initialization of hardware components not covered in the course (since they do not really pertain to signal processing). Those wishing to go into this direction should read relevant publications which will be mentioned at the end of the course.

The contents of the course are given in **Box 3**. As mentioned in the summary, DSP is highly mathematical, but the course will avoid as much mathematics as possible.

SOFTWARE

The signal processing programs presented and used in the course are contained as .EXE files on a CD-ROM which will be available through the Readers' services from February 1998 onwards. These programs will enable readers not only to carry out the set exercises, but also to analyse and produce their own data.

The contents of the CD-ROM are shown in **Box 4**.

Installation

The software installation instructions are contained in file INSTALL.DOC on the CD-ROM mentioned earlier. Basically, the programs are copied in a specific directory and added, when required, to the long WAV files for the relevant part of the course. The CD-ROM also contains some helpful suggestions how best to get acquainted with the software.

Source codes

Most programs are written in TURBO-PASCAL 5.0 and their source code is included on the CD-ROM to enable readers to extend and modify them. The method of operation of a program is frequently explained in the course on the basis of the source code to clarify a particular concept.

Experimentation files

Many experiments require the inputting of a series of programs in a prescribed order. For example, signals are generated, modified (with the aid of filters), and then displayed. It is, for instance, possible to reconstruct the complete response of an SSB transmitter with the aid of file SSB RECEIVERS. Since a single experiment frequently

Box 2. Parameters of wave files

Sampling rate	*44 100, 22 050, or 11 025*
Bits per sample	*16*
Format	*uncompressed*
Channels	*mono(phonic)*

Box 3. Planned course content

- ✓ *Wave files,*
- ✓ *sampled signals,*
- ✓ *sampling theorem,*
- ✓ *aliasing,*
- ✓ *downsampling,*
- ✓ *recursive low-pass and band-pass filters,*
- ✓ *signal generators,*
- ✓ *filter analysis with sweep frequency,*
- ✓ *spectrum analyser,*
- ✓ *discrete Fourier analysis,*
- ✓ *fast Fourier analysis (FFT),*
- ✓ *echo generation,*
- ✓ *step response and frequency response of filters,*
- ✓ *filter design,*
- ✓ *finite impulse response (FIR) filters,*
- ✓ *noise signals,*
- ✓ *filter analysis with noise signals,*
- ✓ *periodic signals,*
- ✓ *Fourier synthesis,*
- ✓ *frequency modulation,*
- ✓ *amplitude modulation and demodulation,*
- ✓ *phase modulation,*
- ✓ *quadrature process,*
- ✓ *wireless facsimile,*
- ✓ *RDS (radio data services) modulation.*

Box 4. Programs on the CD-ROM

Signal generators	
SIN0	*sine wave generator*
SIN1	*sine wave generator*
PULSE1	*pulse generator*
STEP1	*step-wave generator*
NOISE1	*white-noise generator*
FMSWEEP1	*sweep generator*
MUSICG1	*sound-program generator*
Filters	
SINFIL1	*simple band-pass filter*
BANDP1	*simple band-pass filter*
BUTTER1	*digital Butterworth filter*
LP1	*simple low-pass filter*
ECHO1	*echo generator*
FIRFIL1	*general-purpose finite impulse response (FIR) filter*
SPECFIL1FIR	*filter synthesis*
Modulation, demodulation, mathematics	
DWNSMPL1	*downsampling*
SUM1	*weighted sum of two signals*
MUL1	*product of signals (mixing function, etc)*
AMGEN1	*amplitude modulation, synchronous demodulation, mixing*
FMGEN1	*frequency modulator*
SCHMITT1	*Schmitt trigger function*
SHORT1	*signal window spacing*
RTTYRX1	*decoding of serial facsimile data*
Analysers	
INFO1	*general information, average value, signal energy*
SCOPE1	*multi-channel oscilloscope*
SPEC1	*multi-channel spectrum analyser*
Miscellaneous	
SHELP	*help function for the PASCAL/EXE program*
SPP	*preprocessor for experimentation files*

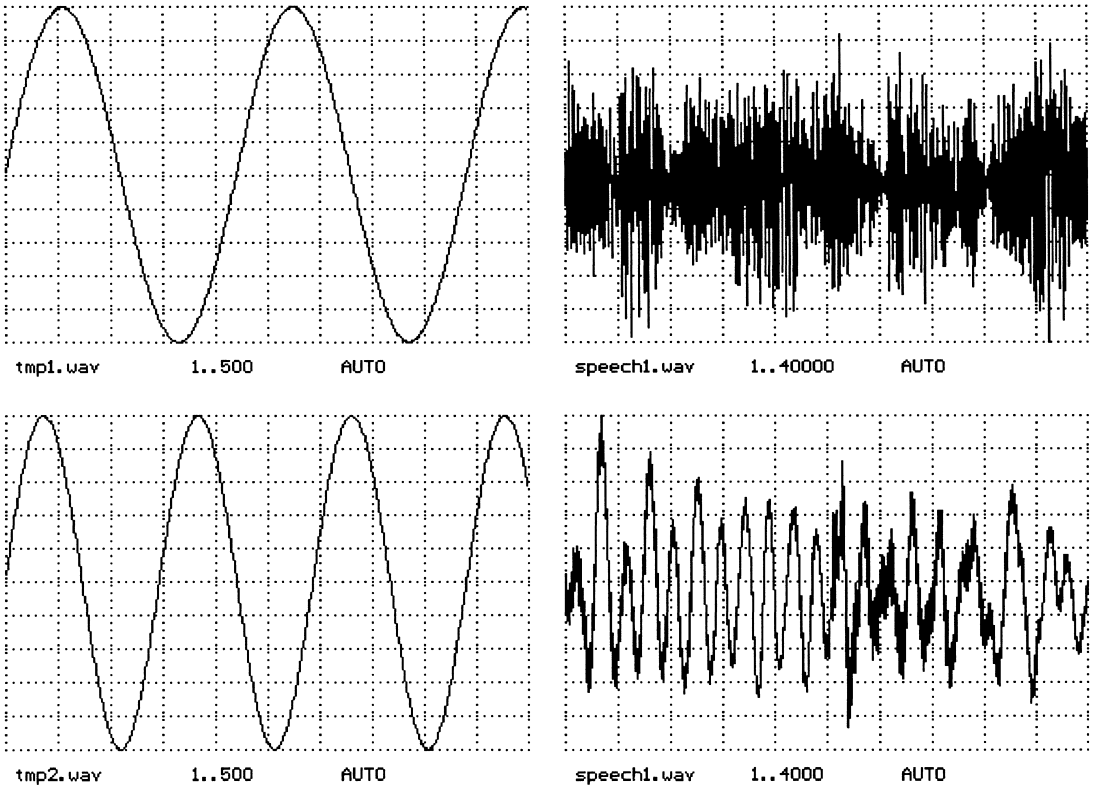


Figure 1. Generated waveforms are displayed as on an oscilloscope.

needs a number of files, there is a simple facility of accessing them all at once by means of a preprocessor.

Preprocessor SPP

The preprocessor program SPP enables the complete sequence of operations in an experiment to be contained in a single file. Basically, this program does nothing but combine a number of files necessary for an experiment by means of a number of instructions.

For example, to carry out the experiment described in file TEST3.SPP, all that is needed to be input is DO TEST3.SPP <return>. Try this out when the CD-ROM has become available. The result is shown in **Figure 1.** Other simulation programs are started in a similar manner.

SHELP function

The most important parameters of many programs may be listed as a help by inputting SHELP program name <return>.

SOURCE MATERIAL

If readers wish to conduct various experiments in signal processing, they need signals that can be processed. The CD-ROM contains a number of audio signals that enable all experiments to be carried out. It is, however, also possible for signals recorded by readers themselves to be processed. All such signals are processed in wave files. In the present course, wave files (ending in .WAV under DOS or Windows) form the central means of exchanging data between files. Various parameters of the data are stored in the header of the wave files. In the present course, the parameters shown in **Box 2** are used.

Sound reproduction

Readers who wish to listen to experiments conducted in the course, rather than look at a curve on the monitor, are offered the opportunity of doing so. For this, a suitable program, many of which are available on sound-cards or as freeware or shareware on the Internet, is needed. Preferably, such a program should run under DOS. When it is enabled, a request has to be made: please play back the file SPEECH1.WAV copied from the CD-ROM. Thereupon, a text with music should be heard.

Recording

Readers who wish to record their own signals, for instance, from a receiver or from an electronic circuit, have to enable a program to make this possible with the sound-card. This program is normally available on the sound-card or as shareware on the Internet. The recording of wave files is not really necessary for the present course, since all requisite files are provided on the CD-ROM.

Nevertheless, readers are advised to become acquainted with how to prepare a wave file, since this will enable them to resolve many measurement problems with the course programs. Such files will extend the operating range of their PC appreciably. It is, of course, important to arrange the recording parameters so that the programs can execute the files.

FIRST EXPERIMENT

The CD-ROM contains a program SIN1.EXE for generating a wave file that contains a sinusoidal signal. Start with SIN1 <return>. The default setting ensures that a 2-second long sine wave of 1000 Hz is generated. The amplitude is 10 000. This signal is loaded into file SIN1.WAV, whereupon the file can be played back. This gives readers the opportunity of generating a sinusoidal signal for test purposes.

The parameters of the program allow other sinusoidal signals to be generated as well. For instance, if a 500 Hz signal with an amplitude of 5 000 and a sampling value of 100 000 at a sampling rate of 11 025 is to be generated, readers should access the program with SIN1 \scale= 5000 \f0= 500 \n= 100000 \fs= 11025 \out= sin2.wav <return>. The data are loaded into file SIN2.WAV.

Readers should try to generate a variety of signals and listen to them. Operation of the program will be explained later.

SOME THEORY

Digital signal processing is based on Shannon's theorem (1949) that tells us how often we have to monitor a time-

varying signal if we wish to reconstruct it from its digital samples. The theorem is a development of Nyquist's 1928 theorem which states that two samples per cycle will completely characterize a band-limited signal; in other words, the sampling rate must be at least twice the highest-frequency component.

We shall first have a look at how an analogue-to-digital converter (ADC) works. Such a converter can operate in three fundamentally different ways. Some operate directly by converting an analogue voltage into a digital one by multiple comparators (flash converters); some operate indirectly by using a local digital-to-analogue converter (DAC) to generate a voltage equal to the unknown input voltage; and a third is called an integrating ADC which converts an unknown voltage into a period of time and then measures the time.

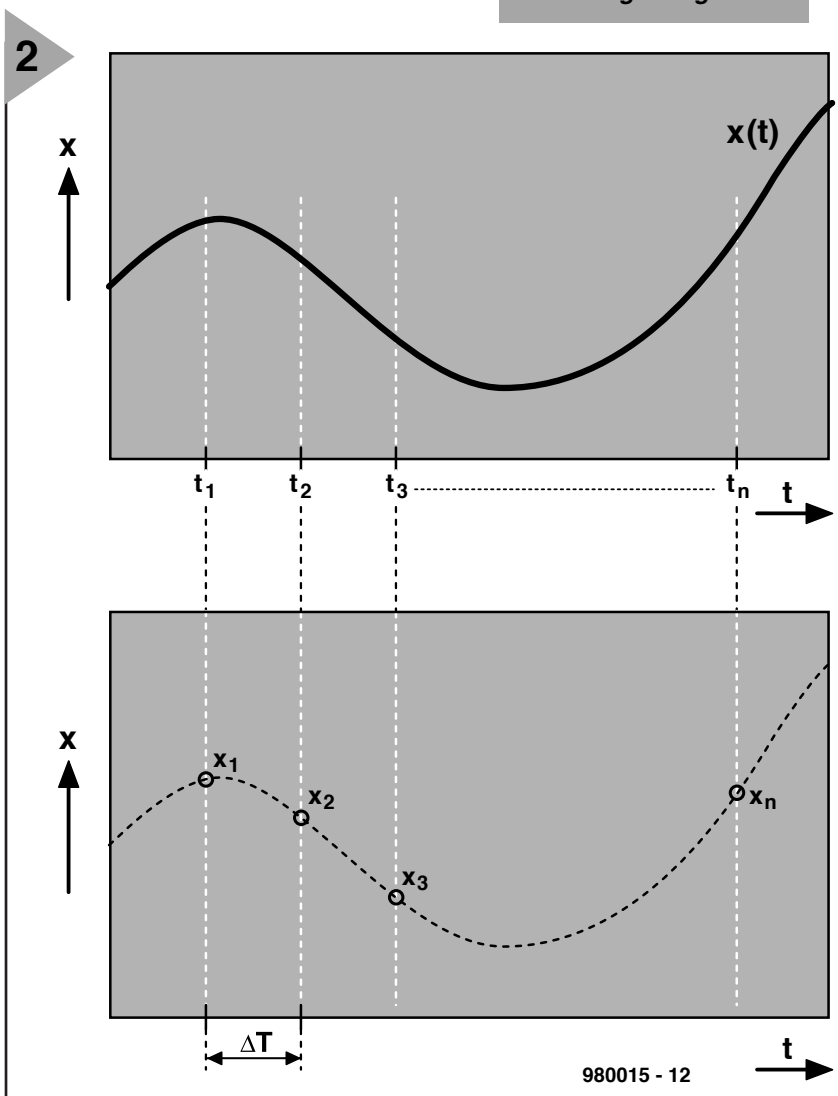
In analogue technology, a signal x has a certain value at a given time t. The function of the voltage with respect to time is $x(t)$ – see **Figure 2**. In order to carry out computations on the signal, it is sampled. This is done by approximating the value of the signal at a number of sampling points in time. Normally, the distance between any two successive sampling points in time is the same. The number of samples per second is called the sampling frequency or sampling rate.

At a point in time t_n , the signal has a value $x(t_n)$, which from now on will be called x_n.Thus, the time-constant signal $x(t)$ has become a time-varying signal. This signal is represented by a series of numbers, similar to the way temperature is recorded by the Meteorological Office. Temperature varies all the time, however slightly, but the value broadcast by the weatherman is an average over a period of time.

In next month's instalment, we will discuss the effects of sampling further and will also have a first look at digital filter technology.

[980015-1]

Figure 2. Sampling of an analogue signal.



Electronic Handyman

Part 2

Architecture of the AVR microcontroller

The steady progress in the field of microcontrollers and microprocessors has produced new, powerful devices which come in standard IC cases yet easily match the computing power of first-generation PCs. The best known representatives of this new group are the well-established PIC devices from Microchip. However, these are now facing serious competition from Atmel's AVR chips. In this article we have a look at the AT90S1200, the first member of this new controller family.

By Dipl. Ing. Bernard C. Zschocke

Apart from a processor core, the typical microcontroller also has a memory element, I/O lines and special functions like counters, timers or serial interfaces, all integrated on the chip. Furthermore, there's often a clock oscillator, RESET logic and a watchdog function. Many controller families also enable external program memory and/or data memory to be connected. That is, however, not generally possible with controllers in a 20-pin DIL enclosure. The internal program memory seems to be the bottleneck in these cases. In the early days of microcontroller technology, controller memory could only be programmed at production time, and it was not possible to correct errors. Later, controllers arrived which could be programmed by individual users, using special programming equipment. These programmers obviously reduce cost whenever prototypes or small production runs are involved. PIC controllers, too, (with a few exceptions) are basically OTPs (One Time Programmable devices).

However, even OTPs are too expen-

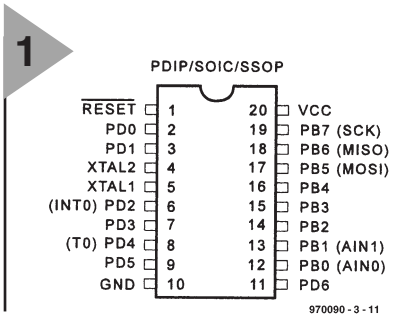


Figure 1. Pinout of the AT90S1200.

Table 1. Status-Register (SREG)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I	T	N	S	V	N	Z	C

Flag	Function
I	Global Interrupt Enable
T	Bit Copy Storage, for BST and BLD instructions
H	Half Carry Flag
S	Sign Bit, S = N ∧ V
V	Overflow with two's complement
N	Negative Flag, indicates negative result
Z	Zero Flag, indicates result is zero
C	Carry Flag

sive, and that's where Atmel comes in with its AVR controllers which may be programmed more than a thousand times using only four lines (SCL, MOSI, MISO and RESET), and some PC software to download the object code via the RS232 port. If suitable hardware is used (as, for example, in the Electronic Handyman), then it is even possible to leave the controller in its application circuit!

THE PROCESSOR CORE

Remarkably, the processor core of the AVR series of microcontrollers has **no** accumulator register for arithmetic functions. Instead, *register-to-register arithmetic* is applied. As long as enough registers are available, that reduces the number of instructions for, say, an addition from three to one (load accumulator, add, store accumulator results). Otherwise, the processor core is MCS-51 compliant, and capable of executing all the usual instructions. The AT90S1200 instruction set may be found on this month's *Datasheets*. The status flags required for the arithmetic functions are listed in **Table 1**.

Pipeline technology allows the processor core to execute an instruction cycle in one clock period. For the one-cycle instructions, that results in a throughput of one million instructions per megahertz of clock frequency (1 MIPS/MHz). Unfortunately, setting and clearing a port pin requires two clock cycles. Thanks to the static structure of the processor, the clock frequency may be made as low as you want, enabling you to achieve an (almost) directly proportional reduction in current consumption.

Further features of the AVR series include a Flashable program memory, an on-chip EEPROM (size depends on controller type) and on-chip RAM in the more extensive models. As with all controllers, there are also different I/O modules. The block diagram of the AT90S1200 (which is also used in the Electronic Handyman) is shown in **Figure 2**. The CPU core comprises a program counter, stackpointer, instruction register/decoder, general-purpose registers, ALU and status registers. Below these elements you find the logic for programming the Flash memory.

MEMORY MODEL

A short discussion of the memory model used by AVR controllers may be useful before tackling the various special functions and the interfaces with the real world. AVR controllers use the Harvard memory architecture, in which program memory and data memory are separated. Depending on external configurations, AVR controllers are capable of addressing up to 8 Mbytes of program memory and an equal amount of data memory. The SRAM data memory has a width of eight bits, the program memory, 16 bits. The first 32 bytes of the data memory cover the Register File (registers R0 through R31; register R30 duplicates as the Z register for indirect addressing). Atmel also refers to these registers using the term 'special functions' just like the timers and the I/O registers. The term *Special Function Registers* as we know it from the MCS51 would have been more appropriate. Access to registers R0 through R31 is possible using register instructions as well as via the LDS or SDS instructions. Note, however, that access by way of the latter two instructions requires three instruction cycles and two words in the program memory. The same goes for the I/O registers: here, access is possible through the IN and OUT instructions (1 cycle; 1 word each), or (again) via the LDS and STS instructions. However, because the AT90S1200 has no internal data memory besides the register file and the I/O registers, the STS and LDS instructions are not available on this controller. In fact, the

Figure 2. Architecture of the AT90S1200.

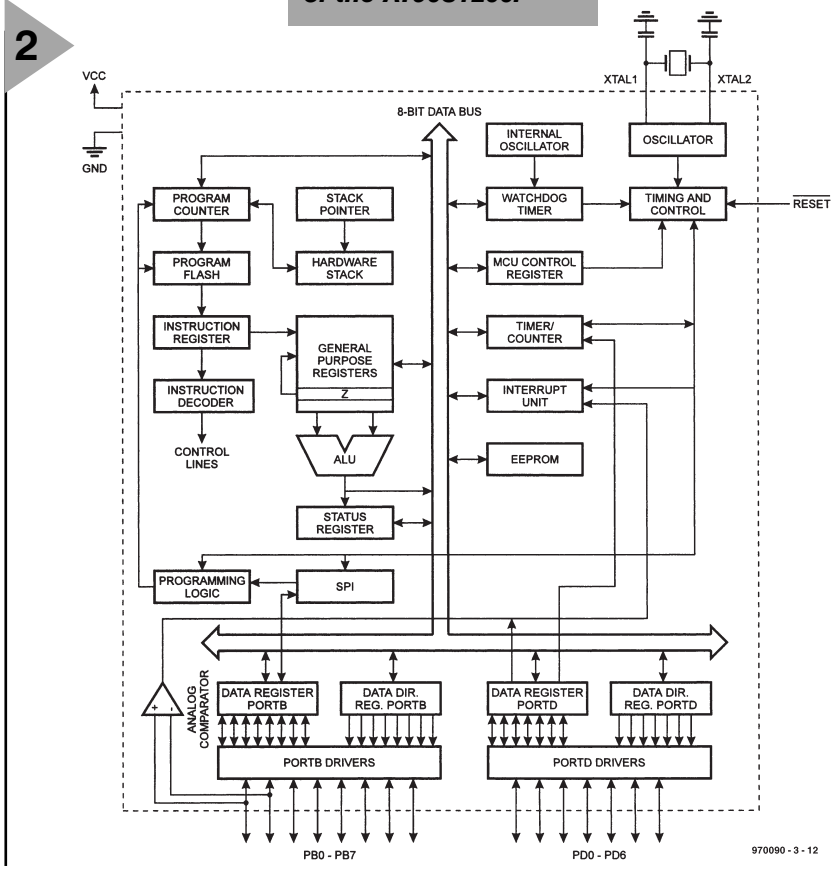


Table 2. Interrupt Vectors			
Vector #	**Prog. Address**	**Source**	**Meaning**
1	$000	RESET	hardware and Watchdog RESET
2	$001	INT0	External Interrupt 0
3	$002	Timer0, OVF0	Timer/Timer Overflow
4	$003	ANA COMP	Analogue Comparator

Table 3. Interrupt Enable/Disable Control Bits			
Bitname	**Bit #**	**I/O Register**	**Source**
INT0	6	GIMSK	External Interrupt 0
ACIE	3	ACSR	Analogue Compare Interrupt
TOIE0	1	TIMSK	Timer 0 Overflow

Table 4. External Interrupt Triggering		
ISC01	**ISC00**	**Interrupt triggered by**
0	0	low level at INT0 pin
0	1	—
1	0	falling edge at INT0 pin
1	1	rising edge at INT0 pin

Table 5. Direction Control Bits in I/O Registers			
DDBn DDDn	**PORTPBn PORTPDn**	**I/O**	**Comment**
0	0	Input without pull-up	Port pin in high-Z state
0	1	Input with pull-up	Port pin supplies current when switched to ground (GND)
1	0	Output	Port pin supplies 0 when at low impedance
1	1	Output	Port pin supplies 1 when at low impedance

AT90S1200 supports only 89 of the 120 instructions handled by the other family members. Although another instruction, MUL (which multiplies a register with a register) is provided, it is not implemented in currently available devices. One explanation for the absence of 31 AVR instructions in the AT90S1200 could be a reduced chip area and, consequently, lower production costs.

In addition to program and data memory, AVR controllers also contain EEPROM (size depending on controller type), whose contents are indirectly accessible through the I/O registers EEAR, EEDR and EECR. To get access, you load the EEAR register with the EEPROM address you want to read or write. For writing you then load the data into the EEDR register, and set the EEWE bit (bit 1 in the EECR). The write operation is finished when the EEWE bit is reset. Reading involves setting the EERE bit after indicating the address. If the EERE bit is cleared again, the EEDR register contains the retrieved data.

PROGRAM EXECUTION

In addition to the usual sequential program execution, the AVR controllers also support subroutine calling and interrupts (which are absent in a number of PIC devices). The AT90S1200 offers a triple hardware stack for the subroutines. The other AVR controllers have a stack pointer which enables a stack to be set up in SRAM. Remarkably, though, is the total absence of stack overflow and stack underflow registers with an associated interrupt vector. Arguably, these would have been very useful for writing error-tolerant and error-detecting programs.

Table 2 shows the interrupt jump addresses. The RESET interrupt initialises all I/O registers. The vector number corresponds with the priority level assigned to the interrupt.

To enable an interrupt, you set the I bit (bit 7 in the SREG). Once an interrupt has been serviced it is cleared by the hardware, and then set again using the RETI instruction (*return from interrupt*) at the end of an interrupt routine. Because of this sequence, only one interrupt can be serviced at a time, except if the interrupts are enabled again within an interrupt routine. Note, however, that each interrupt causes one return address to be stored on the stack! The various interrupts may also be individually enabled or disabled. After a RESET, they are all disabled. **Table 3** shows the I/O register bits that may be used to enable (bit = 1) or disable (bit = 0) interrupts.

A number of events may cause an external interrupt to be applied to port PD2. The event selection is on the basis of bits ISC00 (bit 0 in the MCUCR I/O register) and ISC01 (bit 1 in the MCUR I/O register). This is explained in **Table 4**. To prevent accidental clearing of an interrupt, external interrupt 0 should be blocked before changing ISC00 and ISC01. The other interrupts are discussed with the relevant I/O functions.

I/O REGISTERS

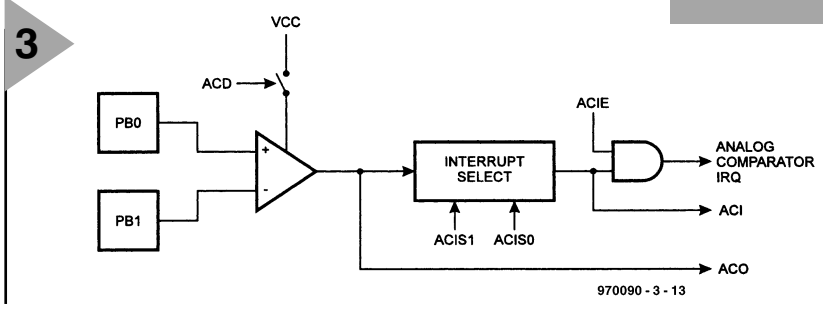
The processor communicates with the real world through its input/output (I/O) lines. Eight I/O lines are bundled into an I/O register. The 20 pins of the DIL case allow a maximum of 15 I/O lines. Consequently, in the AT90S1200, I/O register PB has eight lines, while PD has only seven. I/O registers PA and PC are reserved for devices with more pins (for example, the AT90S4414 or AT90S8515).

As opposed to the well-known controllers in the MCS-51 series, an additional direction bit is available for each I/O register. These bits are called DDRB and DDRD for Port B and Port D respectively. Bit 0 of DDRB controls bit 0 of Port B, and so on. The functions are shown in **Table 5**.

Three addresses are internally provided for each port: one for the direction register, one for the data register and one for reading the Port pins. See **Table 6**.

Reading a Port state should always be done via the address of PIND and PINB. The contents of the Port Data register only matches the state of the Port pins when these are set to 'output'. A programming tip: if you want to change the port pin direction, you should first read the current value of

Figure 3. Internal structure of the analogue comparator.



the data or direction register before changing the relevant bit. With error-tolerant programming in mind, this is done to make sure the other port pins are not affected. Particular care should be taken when changing direction bits in an interrupt routine. This is best avoided!

TIMERS AND COUNTERS

Almost any microcontroller will sport some kind of on-board timer or timer/counter. Timers and counters are always mentioned together because these functions are usually implemented by a *single* hardware counter. The only difference concerns the clock source. In timer mode, this source is an internal clock oscillator. In counter mode, the source is the external input (here, T0). With many controllers, including the AT90S1200, it is possible to reduce the clock frequency with the aid of a prescaler. Bits CS02, CS01 and CS00 determine the manner in which the counter is clocked (see **Table 7**).

The current counter state (value) may be read or set in I/O register TCNT0. The counter starts when it is set by the next instruction. When an overflow occurs (counter state changes from $FF to $00), TOV0 (bit 1 in the TIFR I/O register) is set, and an interrupt is generated (if enabled).

ANALOGUE COMPARATOR

The block diagram of the on-chip analogue comparator is shown in **Figure 3**. Using the ACD bit (bit 7 in the ACSR register) the comparator is switched on, and its output may be read via the AC0 bit. Bits ACIS0 and ACIS1 determine when the ASCI flag is set and an interrupt is generated, provided, of course, it is enabled using the ACIE control bit (see **Table 8**). When switched on, the analogue comparator unfortunately draws rather a lot of current.

WATCHDOG TIMER

A watchdog is capable of detecting when a program has crashed. All you have to do is start the timer and reset it at regular intervals (here, with the aid of the WDR instruction). If no reset occurs during a certain TimeOut period, the timer triggers a RESET. The WDR instruction must **NEVER** be used inside an interrupt routine. Use it once only, and, in general, in the topmost program loop, because there the microcontroller state is accurately known.

In AVR microcontrollers, the watchdog is clocked at about 1 MHz by an associated (RC) clock oscillator. The WDE bit (bit 3 in the WDTCR register) is used to start the watchdog timer, while WDP2, WDP1 and WDP0 determine the length of the TimeOut period per the information shown in **Table 9**.

Table 6. Port Register Addressing

Name	I/O Register Address	Access	Value after Reset	Function
PINB	$16	Read	-	State of Port B pins
DDRB	$17	Read/Write	$00	See Table 5
PORTB	$18	Read/Write	$00	See Table 5
PIND	$10	Read	-	State of Port D pins
DDRD	$11	Read/Write	$00	See Table 5
PORTD	$12	Read/Write	$00	See Table 5

Table 7. Timer/Counter Control by I/O Register TCCR0

CS02	CS01	CS00	Function
0	0	0	Stop, timer/counter off
0	0	1	Timer, count rate = clock rate
0	1	0	Timer, count rate = clock rate ÷ 8
0	1	1	Timer, count rate = clock rate ÷ 64
1	0	0	Timer, count rate = clock rate ÷ 256
1	0	1	Timer, count rate = clock rate ÷ 1024
1	1	0	Counter, clock = falling edge at pin T0
1	1	1	Counter, clock = rising edge at pin T0

CURRENT-REDUCTION MODES

In many applications, a microcontroller is given the momentous task of waiting for an event to occur, without wasting too much (battery) power. The AT90S1200 features two 'sleep' modes which may be started using the SLEEP instruction, but only if bit 5 in the MCUCR register was set beforehand. If the SM bit (bit 4 in MCUCR) was reset, however, the processor enters its Idle mode, which means that the CPU is halted while the timer/counter, the watchdog and the interrupt system keep working. The CPU and the program are restarted by an interrupt. The analogue comparator should be switched off if it is not required to restart the CPU. If, on the other hand, the SM bit was set when the SLEEP instruction is encountered, the external clock generator shuts down and the device enters the Power-Down mode. In case the watchdog was switched on, it causes a RESET when the TimeOut period has elapsed. Otherwise, the CPU and the program may only be restarted via an external RESET or an external, edge-triggered, interrupt.

(970090-3)

Previous articles in this series:

1. Electronic Handyman (Part 1), *Elektor Electronics* December 1997
2. Programmer for Electronic Handyman & AT90S1200, *Elektor Electronics* December 1997.

See also:

Datasheets,
Elektor Electronics January 1998
Handyman website,
http://www.zschocke.com/handyman

Table 8. ACSR Register Organisation

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACD	-	ACO	ACI	ACIE	-	ACIS1	ACIS0

Table 9. TimeOut Period setting using WDP0, WDP1, WDP2

WDP2	WDP1	WDP0	TimeOut period in clock cycles
0	0	0	16 k (= 16 ms)
0	0	1	32 k
0	1	0	64 k
0	1	1	128 k
1	0	0	256 k
1	0	1	512 k
1	1	0	1024 k
1	1	1	2048k (= 2,048 ms)

modern modem technology

ever-faster modems?

When modems were young (a long time ago), they were used in teleprinter circuits at the sedate speed of 45.45 signals (also called elements or pulses) per second. One element per second is one baud (based on the old 5-unit digital Baudot code) and may consist of several bits of data. The Baudot code was replaced by the ASCII code many years ago.The low speeds of these early modems sufficed for many years – until these devices came to be required for linking other machines such as computers or computers and telephone exchanges. Owing to these different demands, over the past ten years or so, the speed of modems has been raised and raised again, but now, owing to the limitations of the analogue sections of the telephone network, the limit of 33.6/56 kbit/s seems to have been reached. But is this really so?

By our Editorial Staff

A modem is a device inserted between a data terminal equipment (DTE) such as a computer and the transmission medium (telephone wire, fibre cable, radio). The modem, a contraction of *mo*dulator-*dem*odulator is often referred to as the data communication equipment (DCE).

The mechanical and electrical interface of a modem is defined by the RS232D/CCITT V.24 standard for serial communications between computers and peripheral devices. Modems are manufactured to CCITT (International Telegraph and Telephone Consultative Committee) recommendations, also known as the V-series, and range from early V.21 devices running at 200

[1] In modern data communications, the symbol replaces the baud as the unit of transmitted data. Like the baud, a symbol may (and frequently does) represent more than one data bit.

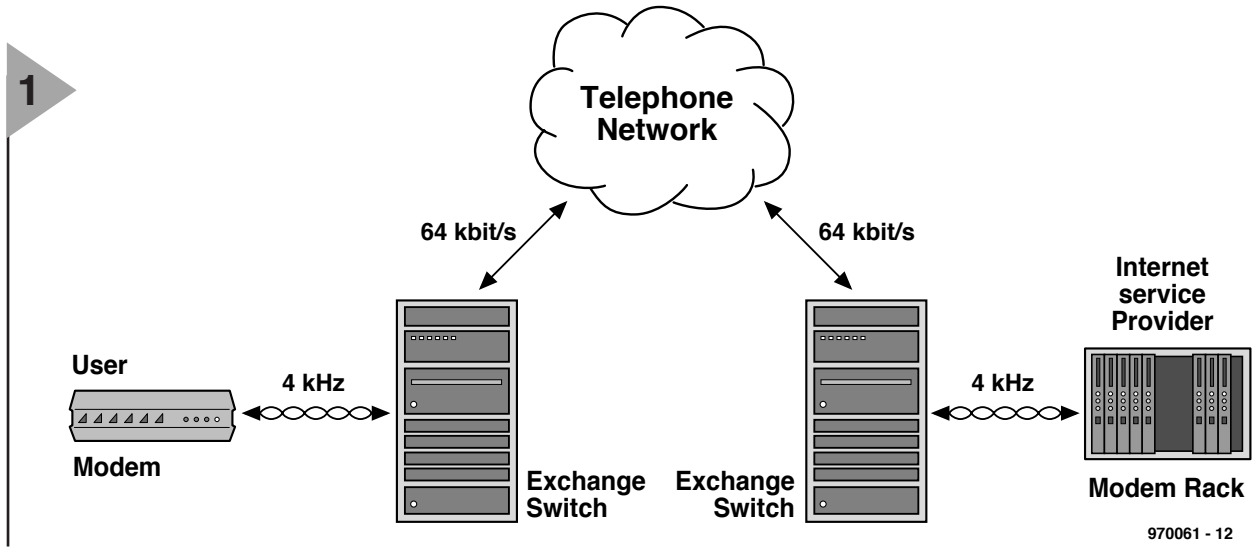


Figure 1. The old telephone channel, from modem user to service provider. This network, which has a bandwidth of 4 kHz, provides 33.6 kbit/s modems the maximum capacity.

bauds to the modern V.42 and the developing V.fast running at 33.6/56 kbit/s.

A modem has to

(1) modulate an outgoing baseband signal on to a carrier for transmission through a telephone line or via radio link;

(2) demodulate an incoming modulated signal from the telephone line or radio link to recover the input baseband information.

In this article, only the telephone line as transmission medium will be considered, but the principles apply also to other links.

Today, the consumer has a choice between an analogue and a digital telephone line. Transmission rates attainable via an analogue line are 33600 bit/s from the user to the telephone exchange (upstream) and 56000 bit/s from the telephone exchange to the user (downstream). Note that the baud rate is equal to the bit rate only when each signal event (or pulse) represents one bit condition, as in a binary system.

MODULATION

There are a great many types of modulation, a number of which are variants of the basic methods. Although the analogue and digital basebands are dissimilar, the modulation techniques, in general, are not.

The techniques employed range from two-tone frequency-shift keying (FSK), through variants of phase shift keying (PSK), differential PSK (DPSK) to multilevel quadrature amplitude modulation (QAM).

In FSK, the carrier is switched between two frequencies. One of these is assigned the logic value 0 and the other, the binary 1. This technique is fairly simple, but requires a large bandwidth, resulting in low data density.

Phase shift keying, sometimes called phase reversal keying (PRK) is a single-frequency method where the data bit stream causes a change of carrier phase. The phase shift is defined and can therefore vary only within certain fixed values. A binary 0 results in no phase change, while a logic 1 causes a 180° phase shift. Intermediate values are obtained by combinations of bits, such as (in 4-PSK) 00, 01, 10, 11.

Quadrature amplitude modulation may be analogue or digital. In the analogue variant, two quadrature versions of the same carrier frequency are amplitude modulated and added to produce a third carrier signal containing both modulations. In the digital variant, which is based on PSK and the amplitude variant, a phase shift as well as an amplitude shift are possible. Moreover, in this method, two carrier frequencies are used, each of which carries half of the data transmission.

BANDWIDTH

The restrictions imposed by the analogue telephone line are difficult to overcome. They relate mainly to the bandwidth of the channel, and the noise added to any signal passing through it.

The usable bandwidth of an analogue telephone line extends roughly from 200 Hz to 4 kHz. This is an artificial restriction, imposed to allow the network to carry many telephone calls at once, It is, however, universal and

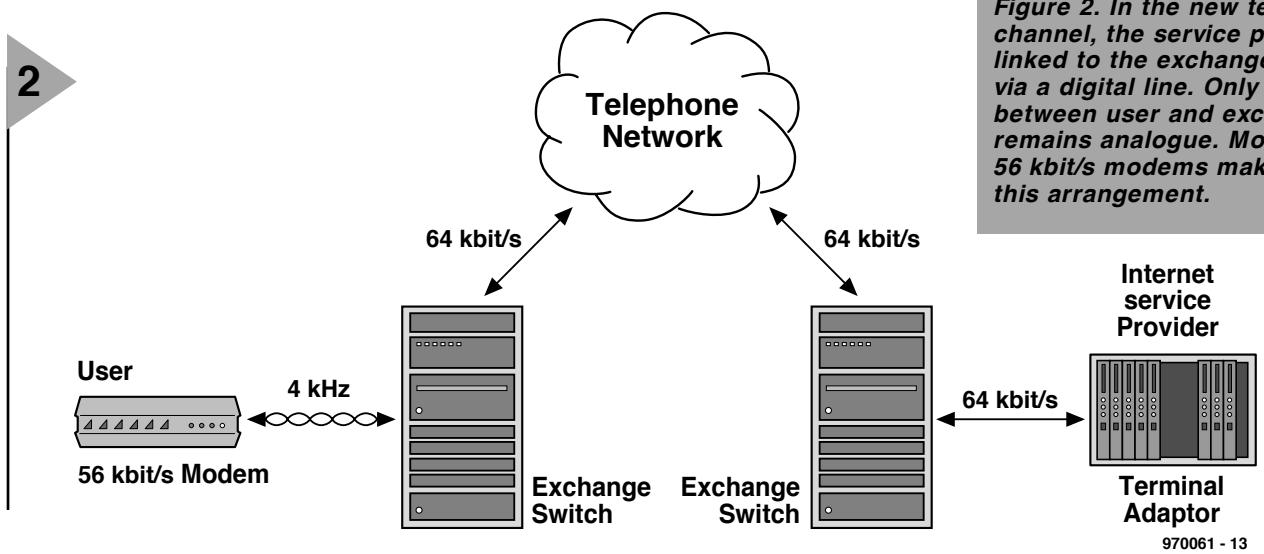


Figure 2. In the new telephone channel, the service provider is linked to the exchange switch via a digital line. Only the link between user and exchange remains analogue. Modern 56 kbit/s modems make use of this arrangement.

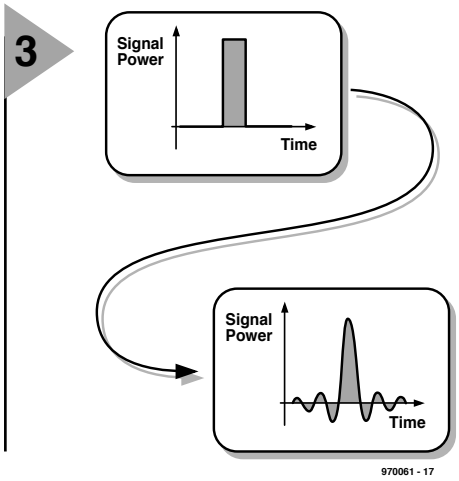


Figure 3. Since an analogue telephone line is intended for the transmission of analogue speech signals, digital data can be easily corrupted owing to the narrow bandwidth of only 4 kHz.

limits the speed at which symbols* can be transmitted.

Not all V-series recommendations pertain to full duplex operation. Those that do enable the automatic return to a lower bit rate if the line quality (noise!) demands this (see Table).

DATA RATES

Not so long ago, the maximum data rate was 28800 bit/s, attained with symbol rates of 3200 per second and with up to nine bits per symbol.

Within the recent past, modern communication technology made possible a data rate of 33600 bit/s, but with the available bandwidth of analogue telephone lines, the end appeared in sight.

However, by some ingenuity and the fact that audio signals are quantized before being transmitted, this rate has been further increased (to 56 kbit/s), but only, as stated earlier, as far as the downstream (service provider or telephone exchange to user) section is concerned. The rate over the upstream section (from user to telephone exchange or service provider) remains 33.6 kbit/s, at least for the time being. This applies only to analogue lines between telephone exchange and user. If a digital line has been installed between these two locations, the data rate is currently 128 kbit/s (modems for this rate are readily available).

Much of the increase in data rates is thanks to the modernization of the telephone network in many countries, which, at least as far as the UK is concerned was started 20-odd years ago and, to all intents and purposes, is now complete.

The telephone networks of yesteryear were intended for the transmission of analogue voice signals over open wire or twisted pair unshielded copper cables. From the 1960s on, in some countries, notably the UK and the USA, there have been significant developments in solid-state electronics which made the use of digital circuits and exchanges possible.

Today, telephone exchanges are connected via digital lines that can carry 32 channels of 64 kbit/s each (a total of 2.048 Mbit/s). At the exchange, it can be determined whether the user has an old-fashioned analogue line or a modern ISDN (Integrated Services Digital Network) line. In case of an analogue line, the signals coming from the user are sampled at the input to the exchange at a rate of 64 kbit/s with a resolution of 8 bits. This means that the section between user and exchange behaves like a near-ideal line able to cope with 64 kbit/s. The new modems use this property to operate at a downstream rate of 56 kbit/s. In the light of the foregoing, it should thus be noted that this rate is only possible in areas where the telephone network itself is a digital one.

Obviously, where a downstream rate of 56 kbit/s is possible, the (Internet) service providers gratefully use it.

NON-COMPATIBLE STANDARDS

As so often in the history of technology, the consumer is again confronted with two non-compatible standards. US Robotics use the X2 system, whereas Lucent (formerly the Bell Telephone Company) and Rockwell use the K56flex standard.

The first problem encountered by the designers of faster modems was the inter-symbol interference (ISI). When symbols are transmitted through a channel, they become blurred in time. This causes the end of one symbol to overlap the beginning of the next, resulting in corruption of the data. This interference is most pronounced when the data rate approaches the limit of the available bandwidth. **Figure 3** shows the effect when a pulse is input to a transmission line. The receiver detects a completely different wave than the digital pulse transmitted by the sender.

The problem of ISI is tackled in both systems by making use of a data rate of 8000 pulses/s, which is equal to the sampling frequency of the analogue telephone line. Each pulse contains eight bits of data, so that a theo-

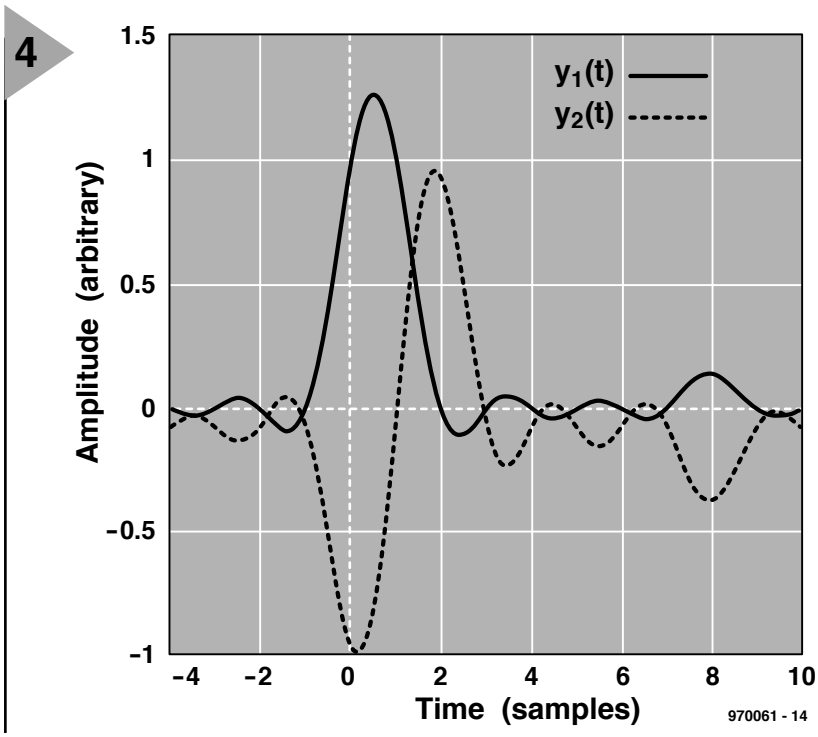


Figure 4. Waveform due to two pulses as seen by the receiver when Lucent's/Rockwell's K56flex standard is used. Note that they have a value only at their own sampling point and at sampling point 8. At all other sampling points their value is zero.

retical bandwidth of 64 kbit/s is available. However, the systems use a different technique to approach this theoretical value as closely as feasible.

It all comes down to the necessity of minimizing any interference. This is not easy and takes up a sizeable chunk of the available processor capacity. In both systems, use is therefore made of controlled ISI, in which a data pulse cannot interfere with an adjacent pulse. Thus, controlled ISI is interference that occurs only with one other pulse. The coding is arranged so that prior calculation can determine which pulse will be affected by the interference. The system is thus aware of this and can take measures to cope with it. This technique ensures that the bandwidth of the channel is used to its greatest extent. The price of it is a more complex receiver. However, this can easily be resolved nowadays by the use of inexpensive digital signal processors (DSPs).

Although both systems use controlled ISI, they use a different technique of coding, which makes them utterly incompatible.

5

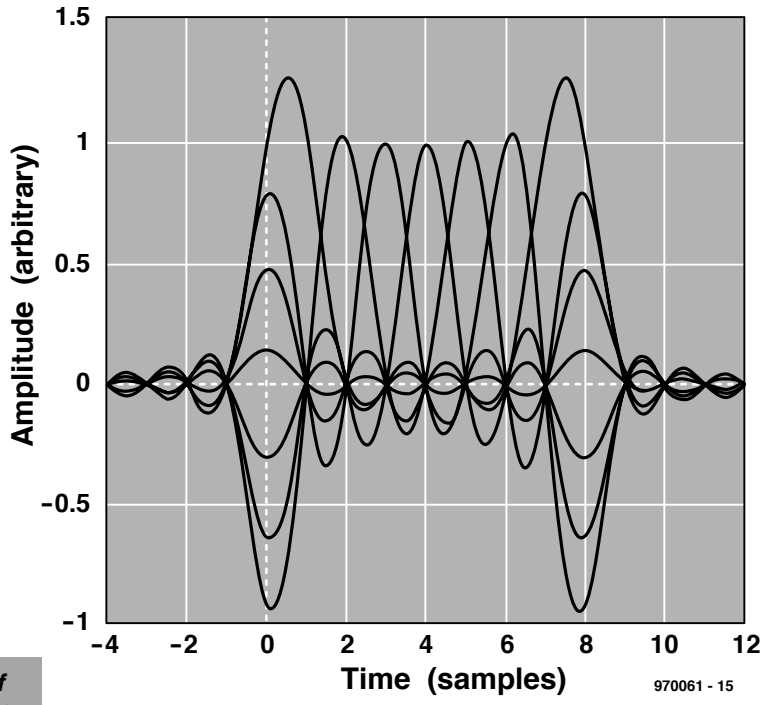


Figure 5. A full set of seven pulses as seen by the receiver when Lucent's/Rockwell's standard is used. It is evident that the level at the eighth sampling point is not defined.

The K56flex standard

In the K56flex standard, the bits are transmitted in packets of eight. Of these, seven are transmitted in such a way that the interference occurs in the time reserved for the eighth bit. Therefore, the transmitting of the eighth bit makes no sense since it would be corrupted and so cause interference itself. This means that only 7/8 of the maximum capacity is available.

The shape of a received wave when the K56flex standard is used is shown in **Figure 4** (for clarity's sake, only two pulses have been transmitted here). In this example, the amplitude of both pulses is normalized; the coding usually takes account of the signal amplitude (at a resolution of eight bits). Note that for y1(t) the signal has a value at sampling times 1 and 8, whereas for y2(t) this is at sampling times 2 and 8.

The two waves represent the first two bits from a group of eight. So, the pulses do not interfere with one another, but would do so with the eighth bit (which has not been transmitted).

The transmission of a group of seven pulses results in the diagram shown in **Figure 5**. Here again, only one pulse can have a value at a given sampling time. The eighth pulse may be considered to have a non-defined value.

As it is known that the sampling rate is 8 kHz and that eight bits can be coded per pulse, it can be calculated that the digital bandwidth is 64 kbit/s. Of this, only 7/8, that is, 56 kbit/s, is used.

The X2 standard

In the X2 standard[2], all available capacity is used, and there is therefore no pause in the data stream. This results in data pulses corrupting adjacent pulses. The data is extracted from this corrupted data stream by the use of a Viterbi equalizer. This equalizer analyses the quality of the telephone line before data are transmitted. To do this, one of the two modems sends a test signal that is analysed by the other. Any distortion, echo, and noise can be determined from this analysis. Then, the roles are reversed and a signal transmitted in the opposite direction, whereupon a second analysis is carried out. The results of the analyses are processed by a Viterbi algorithm[3] and used to program the equalizer. In this way, the distortion can be anticipated and taken account of in the receiver.

This technique in which the hardware anticipates the quality of the telephone line is not new and is also used in modern modems.

QUANTIZATION NOISE

When an analogue signal is quantized (digitized), some accuracy is lost owing to the conversion from a continuous single level to, say, 256 possible levels. This conversion error is considered to be a form of noise since it has a similar effect on the signal quality. During the transmission from provider to user only one analogue-to-digital converter (ADC) is in use and the quality of this must therefore be of a high standard.

The three manufacturers have tackled this aspect in an identical manner. As long as the output signal of the ADC is identical to the requisite value, the quantization noise is minimal. How quantization noise arises is shown in **Figure 6** (upper sketch); the lower sketch shows that when the reference point is chosen well, virtually no quantization noise ensues.

Since the ADC and the digital-to-analogue converter (DAC) are both 8-bit units, choosing the correct reference point is not difficult. A sample is taken when the input level of the data pulse is equal to one of the steps of the converter. This ensures that the sampling error, and thus the quantization noise, is a minimum. If the measured line characteristics are also taken into account, selecting the reference point is straightforward.

Once the quantization noise has been minimized, attention must be paid to other sources of noise. Since these normally do not reside in the modem, it is not easy to take appropriate measures to nullify or at least minimize them. The use of a compander usually enables the effect of such sources to be reduced drastically.

---

[2] X2 is also a CCITT Recommendation covering international user facilities in public data networks.

[3] Information about the Viterbi algorithm may be found at web site http://docs.dcs.napier.ac.uk/docs/get/ryan93a/document.html

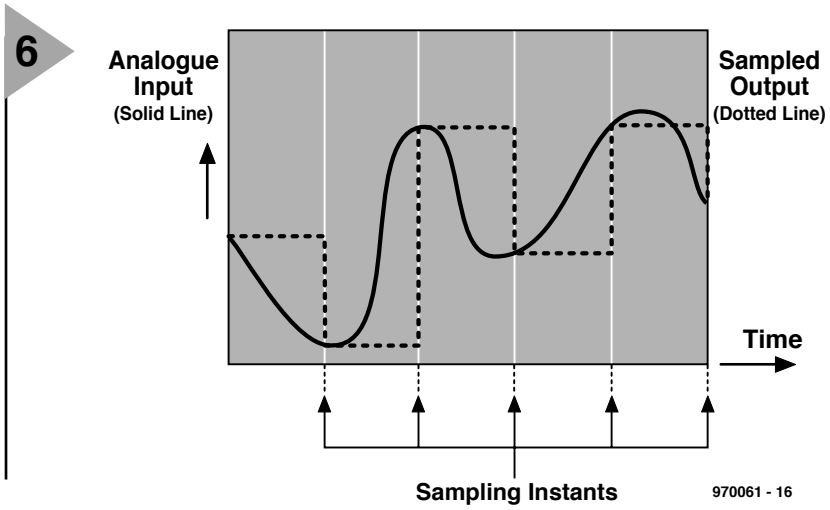


Figure 6. Correct sampling minimizes the generation of quantization noise.

V-series (CCITT) recommendations

	symbol/s	bit/s	modulation	remarks
V.17	2400	9600/14400	QAM	telefax
V.21	300	300	FSK	
V.22	600	1200	PSK	
V.22bis	600	2400	QAM	
V.23	1200/75	1200/75	FSK	
V.26bis	1200	2400	PSK	
V.27ter	1600	4800	PSK	
V.29	2400	9600	QAM	
V.32	2400	9600	QAM	
V.32bis	2400	14.400	QAM	
V.34	3429	28.800	QAM	
V.34+	3429	33.600	QAM	

FSK = frequency shift keying
PSK = phase shift keying
QAM = quadrature amplitude modulation
Note 1: bis and ter are terms with Latin roots meaning two and three respectively. Recommendations with these terms in the number have 2 or 3 data rates, the system normally starting up in the higher rate but falling back to the lower speed under noisy conditions.
Note 2: the number of bits per symbol is readily deduced from the symbol rates and bit rates.

Figure 7. A number of manufacturers have 56 kbit/s modems available. These normally use chipsets from Lucent, Rockwell or US Robotics.

WHAT OF THE FUTURE?

Currently, 56 kbit/s modems are the fastest available for use on an analogue telephone line, provided the network is a digital one. At the time of writing (late 1997), it is not known which standard will be adopted by the CCITT. Based on past experience, it is, however, fairly certain that users will be able to buy an upgrade kit.

The performance level of these modems approaches the limits imposed on it by the maximum capacity of the telephone network.

Users with an ISDN connection can use 128 kbit/s modems which are already available as a look in most catalogues and magazines will show (a typical example is the Motorola BitSURFR). Another advantage of an ISDN line is that enables connections to be made much more rapidly.

It is not likely that further improvements in the data rates can be made until the infrastructure of the telephone network has been changed.

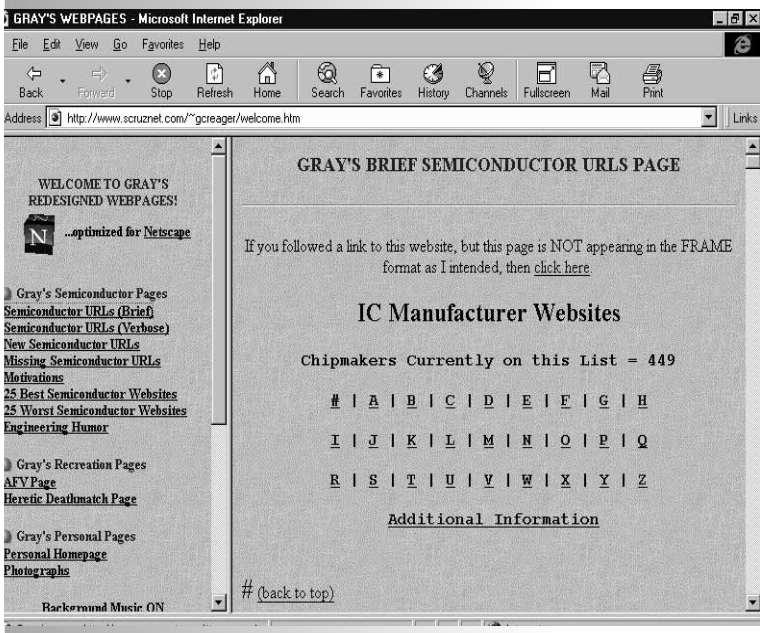
One possibility is, however, that in the future users connected to a cable network will be able to gain access to a digital communication system via a cable modem. Such networks already exist, but are not yet accessible to the majority of consumers.

Fast access to the Internet is also possible via a satellite link, but here again, these are not (yet) available to the general public. Also, the cost of such a link is prohibitive.

A final possibility is a separate data line, a so-called digital subscriber line (DSL) to enable fast digital data connections to be made. In particular, the ADSL (asymmetric digital subscriber line) provides much faster and more powerful facilities for the Internet than the traditional dial-up link. But again, these are fairly expensive.

All in all, it would appear that most consumers will just have to wait for the current limitations of the telephone line to disappear. Until then, all they can do is use their 33.6/56 kbit/s modem to best advantage. [970061]

electronic engineering (EE) Virtual Libraries



The Internet is a great medium when it comes to locating datasheets, application notes, and other documents for electronic design engineering.

The Internet, it has been said, is a gigantic *virtual library*. Motorola, Intel, Philips, and indeed every major chip manufacturer have made their data libraries available via the World Wide Web. White papers, book reviews, technical application notes, datasheets, overviews, and 'Frequently Asked Questions' (FAQ) reports all await a simple click of the mouse.

Unfortunately, however, finding this wealth of documentary design information is not easy. Web URL's (uniform resource locators) can be as arcane as *http://www.semiconductors.philips.com/designerindex/*
and it is far from obvious that **all FAQ's** are stored at *ftp://rtfm.mit.edu*.

Contributions by Jason McDonald

This article, therefore, explains the easiest ways to search for EE-related documents in the vast Internet library, and it provides some locations for the best EE-related repositories. Throughout, enter the World Wide Web URL (uniform resource locator) in the top bar of your browser such as Internet Explorer to access each resource.

FAQs: FREQUENTLY ASKED QUESTIONS REPORTS

FAQs are basic Internet reports answering 'Frequently Asked Questions' and as such span the gamut from microcontrollers, the PowerPC microprocessor, the Ada language, C++, and Object-Oriented Programming to more general topics such as Star Trek, AIDS, and even Unitarianism. Fortunately, FAQs lucky enough to be officially sanctioned are archived by MIT at *ftp://rtfm.mit.edu/pub/ usenet-by-hierarchy/*.
Click on *sci* and then *electronics* to access electronics-specific FAQ's.

A more modern way to locate FAQs is to search via Oxford University at *http://www.lib.ox.ac.uk/search/search faqs.html*
Simply enter key words such as 'electronics' or 'Motorola' to identify documents which interest you. Also have a look at the **Electronics Engineer's ToolBox** published by CERA Research at *http://www.cera2.com/ebox.htm*.

GETTING DATA FROM SEMICONDUCTOR MANUFACTURERS

In their quest to better serve design engineers, every major semiconductor manufacturer now serves their data libraries via the World Wide Web. In some cases, you simply search, click and retrieve an HTML version of a datasheet or application note. In others, you must first acquire the free Adobe Acrobat reader from Adobe (*http://www.adobe.com*) in order to read the portable document format (.pdf). And in still others, you can use the Web to order information by fax in a few minutes.

The ultimate list of URLs for **semiconductor manufacturers** is posted and maintained by Gray Creager at *http://www.scruznet.com/~gcreager/welcome.htm.*

TECHNICAL REPORT SERVERS

Technical report servers provide free access to governmental, university, and non-profit research agencies in the electronics and computer science areas. One of the biggest and best is The Collection of Computer Science Bibliographies, searchable by key word and located at *http://liinwww.ira.uka.de/bibliography/index.html*.
To date, the bibliography contains nearly 750,000 references and has served over 2 million inquiries since February 1995. Indiana University's Knowledge Base at *http://sckb.ucssc.indiana.edu/kb/*
contains about 4,500 entries that answer computing questions asked by Indiana University users. You can either search by key word, or use a subject tree. Try, say, 'TCP/IP'. At a more advanced level, tap into the resources of NASA at *http://techreports.larc.nasa.gov/cgi-bin/NTRS*. As a US government agency, NASA makes available its wealth of technical knowledge via this Web server.

Specific to the electronics area, Eg3 Communications offers a search engine at *http://www.cera2.com/ghindex.html*. Try, say, 'BASIC Stamp'.

The Networked Computer Science Technical Reports Library (NCSTR) at *http://cs-tr.cs.cornell.edu/* is another repository strong in computer science, and the On-line CS Techreports — a master listing of Internet Computer Science repositories is located at *http://www.cs.cmu.edu/afs/cs.cmu.edu/user/jblythe/Mosaic/cs-reports.html*.

(985005-1)

ELEKTOR ELECTRONICS

AT90S1200
Instruction Set Summary 1

AT90S1200
Microcontroller with 1K bytes Downloadable Flash

Manufacturer:
European headquarters: Atmel UK Ltd.,
Coliseum Business Centre, Riverside way,
Camberley, Surrey GU15 3YL.
Internet: www.atmel.com. Email: avr@atmel.com

Features
- Utilizes the AVR® Enhanced RISC Architecture
- 89 Powerful Instructions - Most Single Clock Cycle Execution
- 1K bytes of In-System Reprogrammable Downloadable Flash
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 64 bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 32 x 8 General Purpose Working Registers
- 15 Programmable I/O Lines
- VCC : 2.7 - 6.0V
- Fully Static Operation, 0 - 16 MHz
- Instruction Cycle Time: 62.5 ns @ 16 MHz
- One 8-Bit Timer/Counter with Separate Prescaler
- External and Internal Interrupt Sources
- Programmable Watchdog Timer with On-Chip Oscillator
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Programming Lock for Software Security
- 20-Pin Device
- Selectable On-Chip RC Oscillator for Zero External Components

Brief Description
The AT90S1200 is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S1200 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.
The AVR core combines a rich instruction set with the 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

8-Bit AVR® Microcontroller with 1K bytes Downloadable Flash

AT90S1200

ELEKTOR ELECTRONICS

AT90S1200
Instruction Set Summary 3

Mnemonic	Operands	Description	Operation	Flags	# Clocks
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC+ k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC+ k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC+ k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC+ k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC+ k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC+ k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC+ k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC+ k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC+ k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC+ k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC+ k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC+ k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC+ k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC+ k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC+ k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC+ k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC+ k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC+ k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
LD	Rd,Z	Load Register Indirect	Rd ← (Z)	None	2
ST	Z,Rr	Store Register Indirect	(Z) ← Rr	None	2
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+ 1) ← Rd(n), Rd(0) ← 0	ZC,NV	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+ 1), Rd(7) ← 0	ZC,NV	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+ 1) ← Rd(n),C←Rd(7)	ZC,NV	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ←C,Rd(n) ← Rd(n+ 1),C←Rd(0)	ZC,NV	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+ 1), n= 0..6	ZC,NV	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ←Rd(7..4),Rd(7..4) ←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1

AT90S1200
AT90S1200 Instruction Set Summary 2
DATASHEET 1/98
ELEKTOR ELECTRONICS

Mnemonic	Operands	Description	Operation	Flags	# Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd · K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • (FFh - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)= 0) PC ← PC + 2 or 3	None	1 / 2
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)= 1) PC ← PC + 2 or 3	None	1 / 2
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)= 0) PC ← PC + 2 or 3	None	1 / 2
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)= 1) PC ← PC + 2 or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC ← PC + k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC ← PC + k + 1	None	1 / 2

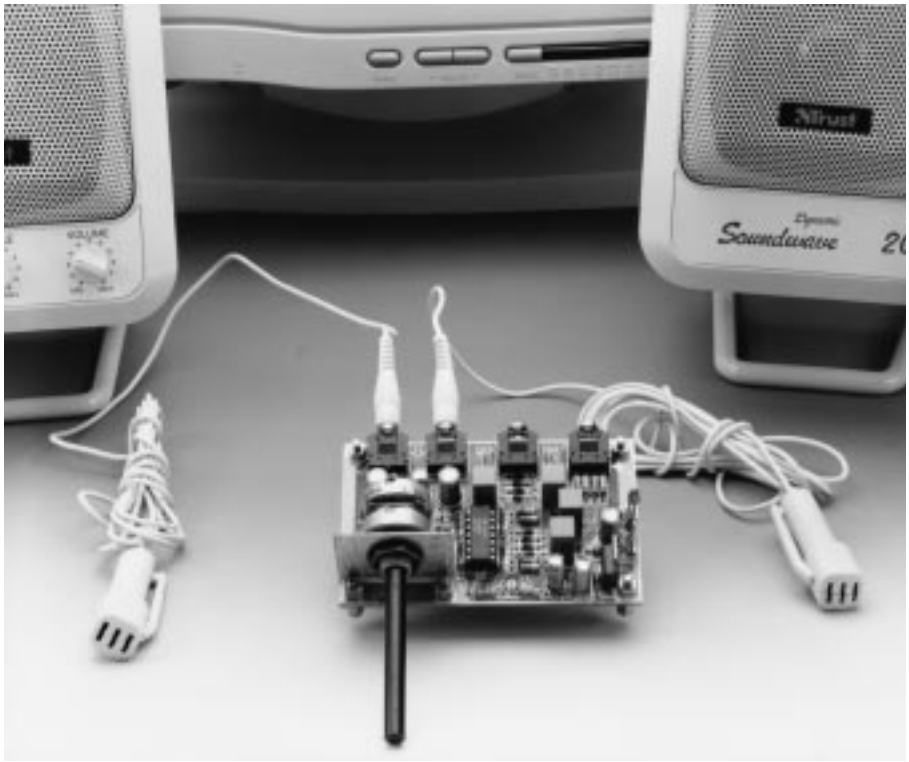
AT90S1200
AT90S1200 Instruction Set Summary 4
DATASHEET 1/98
ELEKTOR ELECTRONICS

Mnemonic	Operands	Description	Operation	Flags	# Clocks
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		-Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
GLOBAL INSTRUCTIONS					
NOP		No Operation	None	1	
SLEEP		Sleep	(see specific descr. for Sleep function)	None	3
WDR		Watch Dog Reset	(see specific descr. for WDR/timer)	None	1

All instructions use one word, except CALL, JUMP, LDS and STS (two instructions).
Rd = Register r0-R31 or R16-R31 (depending on instruction).
Rr = R0-R31.
b = constant (0-7), may be a constant expression.
s = constant (0-7), may be a constant expression.
P = constant (0-31/63), may be a constant expression.
K = constant (0-255), may be a constant expression.
k = constant, value rane depends on instruction. May be a constant expression.
q = constant (0-63), may be a constant expression.

stereo microphone input adaptor for PC

useful extension of sound card



There are occasions when it is desirable or even essential for a stereo microphone to be connected to a personal computer. Unfortunately, the microphone input of most sound cards used in PCs is monophonic. This article describes a simple adaptor to convert the mono(phonic) input into a stereo(phonic) input. It may also be used to provide a cassette deck with a stereo(phonic) microphone input.

Design by T.Giesberts

Nowadays, there is not the sharp dividing line of yesteryear between consumer audio, TV, video, and computer, equipment. In fact, today it is sometimes difficult to decide where one ends and the other begins. The audio installation may be used to reproduce the sound of a film on the video recorder; the CD-ROM drive of a PC may be used to play an audio CD; and the PC may be used for processing audio and video signals.

The PC can serve not only for the reproduction of the simple sounds that support certain software, but also for processing complex music signals. In that case, the audio signals are first quantized via the sound card and subsequently processed as desired. In fact, hard-disk recording is no longer a novelty.

Unfortunately, the microphone

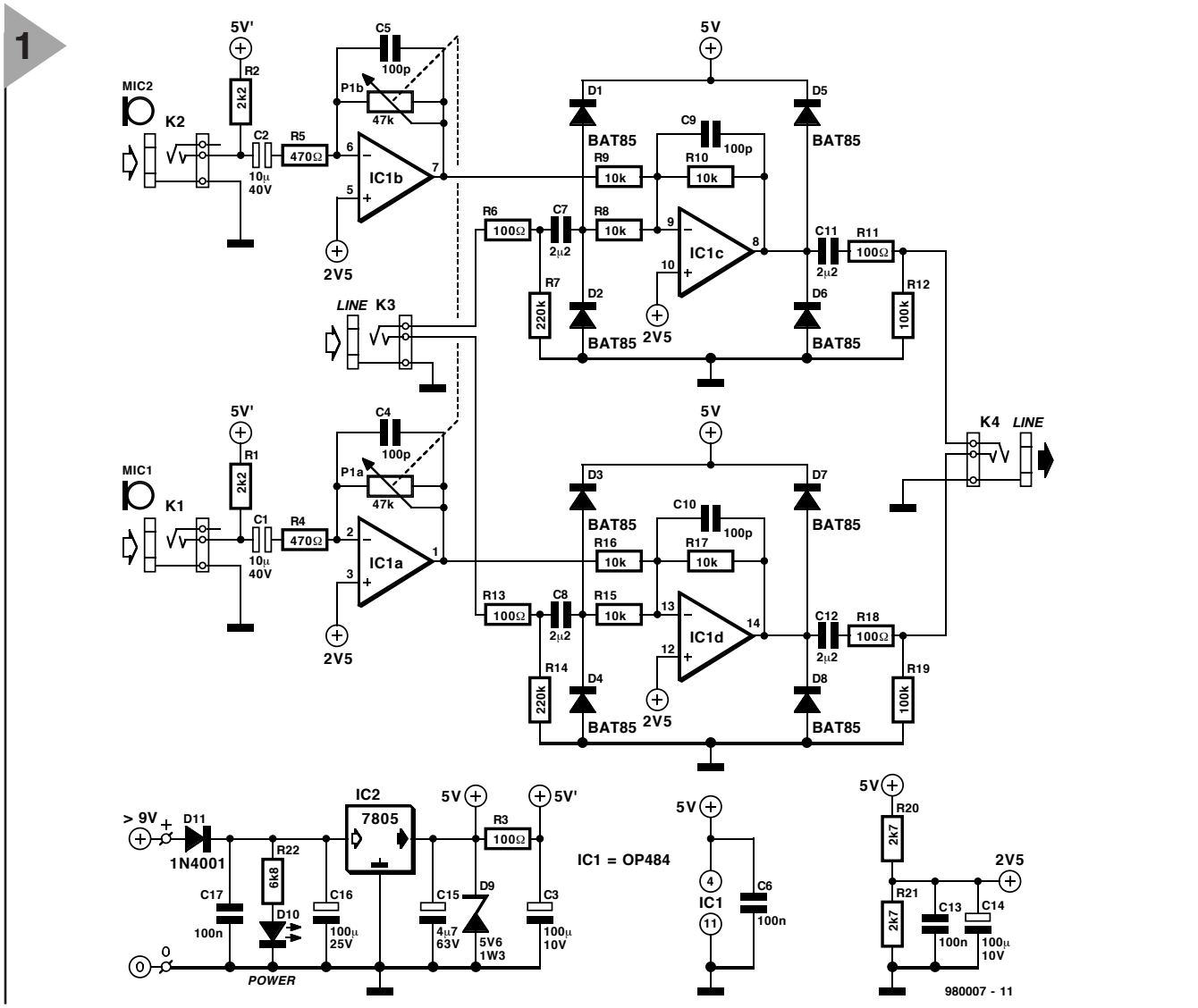


Figure 1. Circuit diagram of the stereo microphone adaptor for PCs.

input of most sound cards used in consumer PCs is monophonic. Luckily, however, many sound cards have a stereophonic line input. which can be converted to a stereo microphone input.

CIRCUIT DESCRIPTION

The conversion of a line input to a microphone input normally entails first of all raising the microphone output signal (a few millivolts to a few hundred millivolts) to line level (standard: 775 mV across 600 Ω designated 0 dBm, but in consumer equipment the r.m.s. level may range from 100 mV to 5 V). It is, however, convenient to retain the original line input function, and this is so in the present circuit.

In the circuit diagram in **Figure 1,** jack socket K_4 serves to connect the adaptor to the PC, socket K_3 is the 'new' line input, and sockets K_1 and K_2 form the stereophonic microphone input. The microphone output signals are amplified by operational amplifiers IC_{1a} and IC_{1b}, while IC_{1c} and IC_{1d} serve as adders and output stages.

Op amps IC_{1a} and IC_{1b} are straightforward inverting amplifiers whose amplification is determined by the ratio P_{1a}:R_1 (P_{1b}:R_2). In the prototype, this is ×23, which is sufficient for the electret microphones used.

Resistors R_1 and R_2 also provide the supply voltage for the FET impedance adaptor in these microphones. (FET = field-effect transistor).

Potentiometer P_1 serves to set the sensitivity of the microphone input or the level of the amplified microphone signal.

The configuration of the adders/output stages is similar to that of the preamplifiers, but their amplification is unity and the output impedance is rather higher.

Resistors R_6-R_7 and R_{13}-R_{14} ensure stable operation with unusual line signals.

Since the supply voltage is only 5 V, the line inputs and outputs are protected against overvoltage by diodes D_1–D_8. Zener diode D_9 makes certain that the supply voltage cannot exceed 5.6 V in any circumstances.

The supply voltage is obtained from a standard 9 V mains adaptor, which need not be regulated nor rated for high currents (the circuit draws only about 10 mA). Regulator IC_2 holds the output voltage steady at 5 V. This low voltage ensures that the sound card cannot be overdriven or damaged by overvoltage.

The amplifier stages are powered by half the supply voltage via potential divider R_{20}-R_{21}, which is decoupled by capacitors C_{13} and C_{14}.

The supply lines to the microphones are decoupled by network R_3-C_3.

Diode D_{11} protects the lines against polarity reversal, while D_{10} is the on/off indicator.

CONSTRUCTION

The adaptor is best built on the printed-circuit board shown in **Fig-**

Parts list
Resistors:
R_1, R_2 = 2.2 kΩ
R_3, R_6, R_{11}, R_{13}, R_{18} = 100 Ω
R_4, R_5 = 470 Ω
R_7, R_{14} = 220 kΩ
R_8, R_9, R_{10}, R_{15}, R_{16}, R_{17} = 10 kΩ
R_{12}, R_{19} = 100 kΩ
R_{20}, R_{21} = 2.7 kΩ
R_{22} = 6.8 kΩ
P_1 = 47 kΩ stereo, logarithmic potentiometer for board mounting

Capacitors:
C_1, C_2 = 10 µF, 40 V, bipolar, radial
C_3, C_{14} = 100 µF. 10 V, radial
C_4, C_5, C_9, C_{10} = 100 pF
C_6, C_{13}, C_{17} = 0.1 µF
C_7, C_8, C_{11}, C_{12} = 2.2 µF, metallized polyester (MKT)
C_{15} = 4.7 µF, 63 V, radial
C_{16} = 100 µF, 25 V, radial

Semiconductors:
D_1–D_8 = BAT85
D_9 = zener diode, 5.6 V, 1.3 W
D_{10} = LED, high-efficiency
D_{11} = 1N4001

Integrated circuits:
IC_1 = OP484FP (Analog Devices) (see text)
IC_2 = 7805

Miscellaneous:
K_1–K_4 = stereo jack socket, 3.5 mm, for board mounting
PCB Order no. 980007-1 (see Readers Services section towards the end of this issue)

2

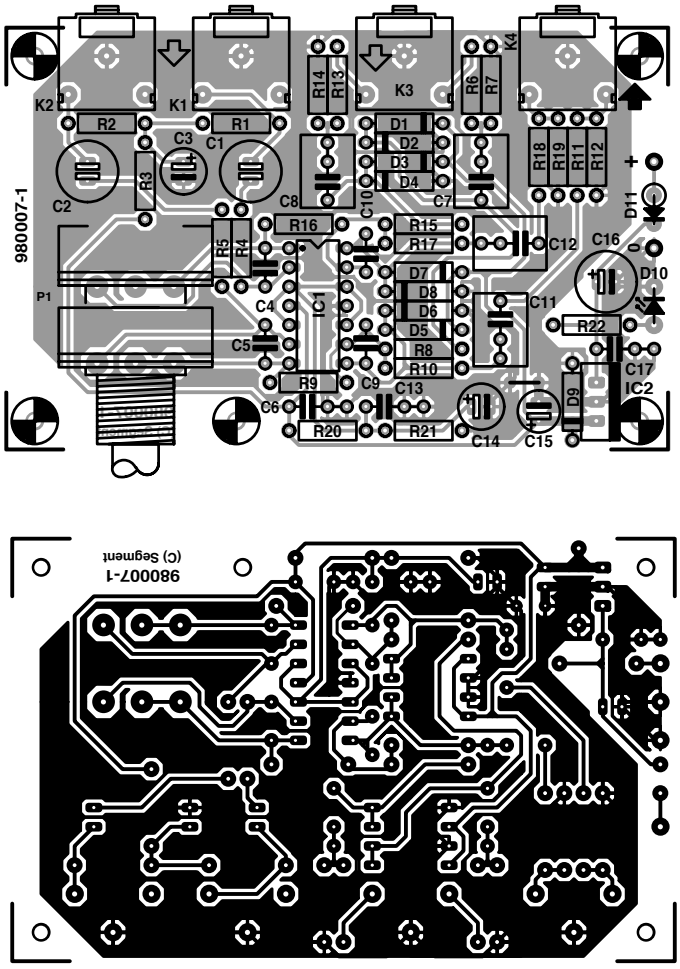
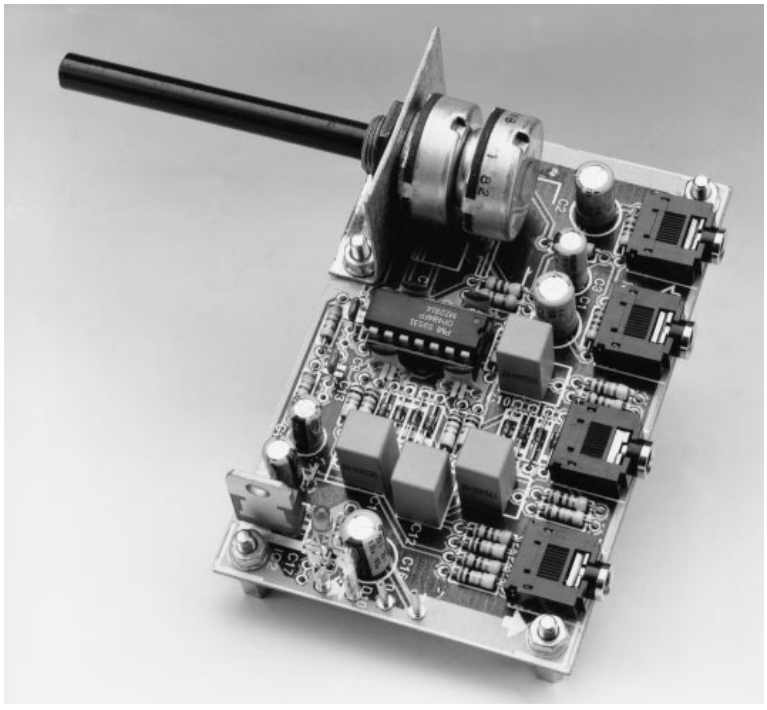


Figure 2. The various jack sockets and potentiometer can be fitted directly to the printed-circuit board.

Figure 3. Photograph of the completed prototype stereo microphone adaptor board.

3



ure 2. All jack sockets are at one side of the board and the volume/sensitivity control at the opposite side.

When the (straightforward) construction has been completed and the correct operation of the adaptor has been verified, the adaptor should be fitted in a suitable enclosure. This is preferably a small metal case to which the earth of the circuit is strapped via one central point (near one of the jack sockets).

OPTIONAL MODIFICATIONS

The amplification of the circuit specified earlier is sufficient for the (electret) microphones used with the prototype. If desired (or required), it may be raised by lowering the value of R_9 and R_{16}, but not below 2 kΩ.

The operational amplifier used in the IC_1 position is a Type OP484 from Analog Devices. This device combines a rail-to-rail input and output with a very low noise factor and a range of supply voltages that extends to well below that of most other types. Nevertheless, other types of op amp, such as the TLC272, may also be used.

[980007]

APPLICATION NOTE

DS5000

soft microcontroller from Dallas

8051-compatible

1

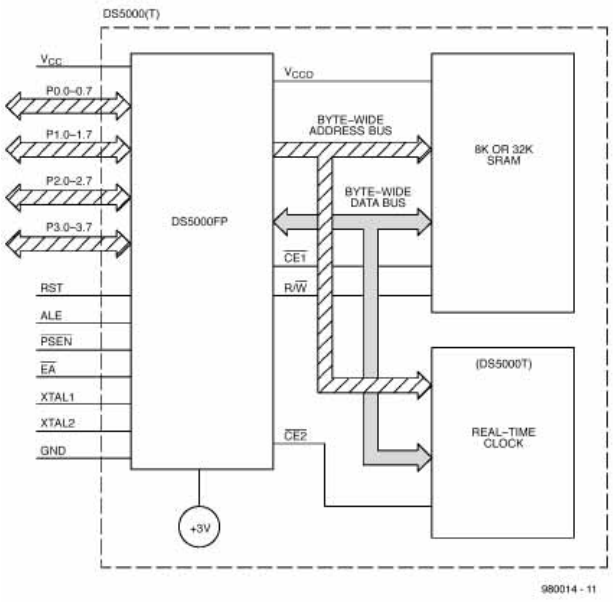


Figure 1. Functional diagram of the DS5000(T). The real-time clock is optional (but standard in the T-version).

The 8-bit 8051-compatible Type DS5000 microcontroller from Dallas Semiconductor has a number of interesting properties, of which the non-volatile section of the memory in the processor and the integral bootstrap loader are particularly noteworthy. Others are:

- crashproof operation;
- executes encrypted software to prevent unauthorized disclosure
- on-chip, full-duplex serial I/O ports
- two on-chip timer/event counters
- 32 parallel I/O lines
- optional permanently powered real-time clock (DS5000T)
- compatible with industry standard 8051 instruction set and pinout

A Dallas Semiconductor Application

2

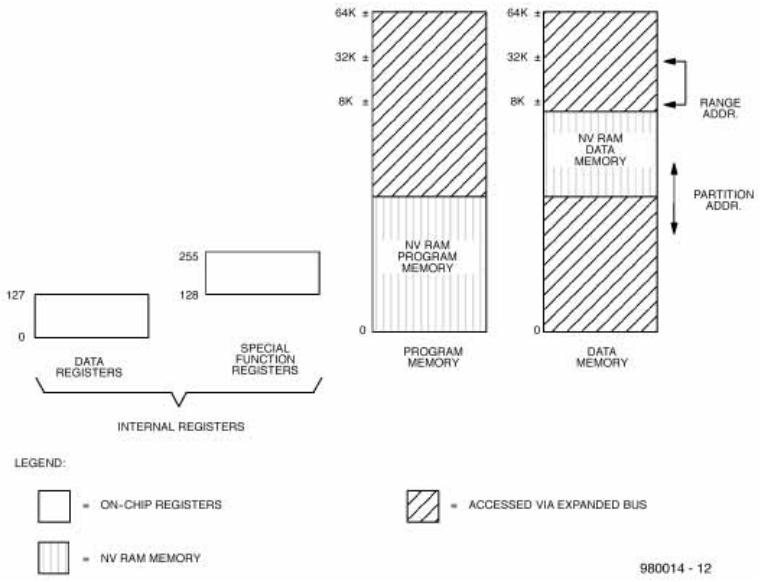


Figure 2. The logical address spaces are largely identical with those of an 8051 processor, except for the non-volatile sections.

The DS5000(T) soft microcontroller is an 8-bit module that is compatible with the 8051 family of microprocessors and which offers 'softness' in all aspects of its application. The softness is accomplished through the comprehensive use of non-volatile technology to preserve all information in the absence of the supply voltage, V_{CC}.

Dependent on which version, the internal program-cum-data memory comprises an 8 kbyte or 32 kbyte non-volatile, static CMOS-SRAM (complementary metal-oxide silicon static random-access memory). The internal data registers and key configuration registers are also non-volatile.

An optional real-time clock (standard in the T-version of the controller) enables constant time monitoring to be implemented. This clock, controlled by an external crystal, keeps time to a hundredth of a second.

FUNCTIONAL DIAGRAM

The internal arrangement of the DS5000 is shown in **Figure 1**. The device consists of three main sections: the processor, DS5000FP, and 8 kbyte or 32 kbyte static random-access memory (SRAM), and an (optional, except T-version) real-time clock. These sections are interlinked by a byte-wide address bus, a byte-wide data bus, two selection lines, $\overline{CE1}$ and $\overline{CE2}$, and a read/$\overline{write}$ line, R/$\overline{W}$.

Communication with the outside world is via four standard interfaces, that is, 8-bit ports P_0–P_3, an RST (reset) line, an ALE (Address Latch Enable) line, a $\overline{PSEN}$ (Program Store Enable) line and an $\overline{EA}$ (External Access) line. The similarity to the 8051 family devices is evident.

INSTRUCTION SET

The DS5000(T) executes an instruction set which is object code compatible with the industry standard 8051 microcontroller. As a result, software development packages which have been written for the 8051 are compatible with the DS5000(T), including cross-assemblers, high-level language compilers, and debugging tools.

MEMORY ORGANIZATION

Figure 2 illustrates the address spaces that are accessed by the DS5000(T). As shown in the figure, separate address spaces exist for program and data memory. Since the basic addressing capability of the machine is 16 bits, a maximum of 64 kbytes of program memory and 64 kbytes of data memory can be accessed by the DS5000(T) CPU. The 8 kbyte or 32 kbyte RAM area inside the DS5000(T) can be used to contain both program and data memory.

The real-time clock (RTC) in the DS5000(T) is reached in the memory map by setting an SFR (Special Function Register) bit. The MCON.2 (memory control) bit (ECE2) is used to select an alternate data memory map. When ECE2=1, all MOVxs (Move instructions) will be routed to this alternate memory map. The real-time clock is a serial device that resides in this area.

If the ECE2 bit is set on a DS5000 without a timekeeper, the MOVs will simply go to a non-existent memory. Software execution would not be effected otherwise.

PROGRAM LOADING

The Program Load Modes allow initialization of the NVRAM (non-volatile RAM) program/data memory. This initialization may be performed in one of two ways.

1. Serial Program Loading which is capable of performing Bootstrap Loading of the DS5000(T). This feature allows the loading of the application program to be delayed until the DS5000(T) is installed in the end system. Dallas Semiconductor strongly recommends the use of serial program loading because of its versatility and ease of use.

2. Parallel Program Loading which performs the initial loading from parallel address/data information presented on the I/O port pins. This mode is timing-set compatible with the 8751H microcontroller programming mode.

The DS5000(T) is placed in its Program Load configuration by simultaneously applying a logic 1 to the RST pin and forcing the $\overline{PSEN}$ line to a logic 0 level. Immediately following this action, the DS5000(T) will look for a parallel Program Load pulse, or a serial ASCII carriage return (0DH) character received at 9600, 2400, 1200 or 300 bps over the serial port.

The hardware configurations used to select these modes of operation are illustrated in **Figure 3**.

Table 1 – Crystal frequency and available speeds

Crystal frequency (MHz)	Speed (bps)					
	300	1200	2400	9600	19200	57600
14,7460		S	S	S	S	
11,0592	S	S	S	S	S	S
9,21600	S	S	S	S		
7,37280	S	S	S	S		
5,5296	S	S	S	S		
1,8432	S	S	S	S		

S = supported

SERIAL BOOTSTRAP LOADER

The Serial Program Load Mode is the easiest, fastest, most reliable, and most complete, method of initially loading application software into the DS5000(T) non-volatile RAM. Communication can be performed over a standard asynchronous serial communications port. A typical application would use a simple RS232 serial interface to program the DS5000(T) as a final production procedure.

The hardware configuration which is required for the Serial Program Load mode is illustrated in **Figure 3**. Port pins 2.7 and 2.6 must be either open or pulled high to avoid placing the DS5000(T) in a parallel load cycle. Although an 11.0592 MHz crystal is shown in Figure 3, a variety of crystal frequencies and loader baud rates are supported, shown in **Table 1**.

The serial loader is designed to operate across a three-wire interface from a standard UART (Universal Asynchronous Receiver/Transmitter). The receive, transmit, and ground, wires are all that are necessary to establish communication with the DS5000(T).

The Serial Bootstrap Loader implements an easy-to-use command line interface which allows an application program in an Intel hex representation to be loaded into, and read back from, the device. Intel hex is the typical format which existing 8051 cross-assemblers output. The serial loader responds to single character commands which are summarized in **table 2**.

PARALLEL PROGRAM LOAD CYCLE

The Parallel Program Load Cycle is used to load a byte of data into a register of memory location within the DS5000(T). The Verify Cycle is used to read this byte back for comparison with the originally loaded value to verify proper loading. The Security Set Cycle may be used to enable the Software Security feature of the DS5000(T).

One may also enter bytes for the MCON register or for the five encryption registers using the program MCON cycle. When this cycle is used, the absolute register address must be presented at Ports 1 and 2 as in the normal program cycle (Port 2 should be 00H). The MCON contents can likewise be verified with the Verify MCON cycle.

When the DS5000(T) first detects a Parallel Program Strobe pulse or Security Set Strobe pulse while in the Program Load Mode following a Power On Reset, the internal hardware of the DS5000(T) is initialized so that an existing 4 kbyte program can be programmed into a DS5000(T) with little or no modification. This initialization automatically sets the Range Address for 8 kbytes and maps the lowest 4 kbyte bank of Embedded RAM as program memory. The next 4 kbytes of Embedded RAM are map-ped as Data Memory.

In order to program more than 4 kbytes of program code, the Program/Verify Expanded cycles can be used. Up to 32 kbytes of program code can be entered and verified. Note that the expanded 32 kbyte Program/Verify cycles take much longer than the normal 4 kbyte Program/Verify cycles.

A typical parallel loading session would follow this procedure. First, set the contents of the MCON register with the correct range and partition only if expanded programming cycles are used. Next, the encryption registers can be loaded to enable encryption of the program/data memory (not required). Then, program the DS5000(T) with either normal or expanded program cycles and check the memory contents with Verify cycles. The last operation would be to turn on the security lock feature by either a Security Set cycle or by explicitly writing to the MCON register and setting MCON.1 to a 1.

[980014]

3

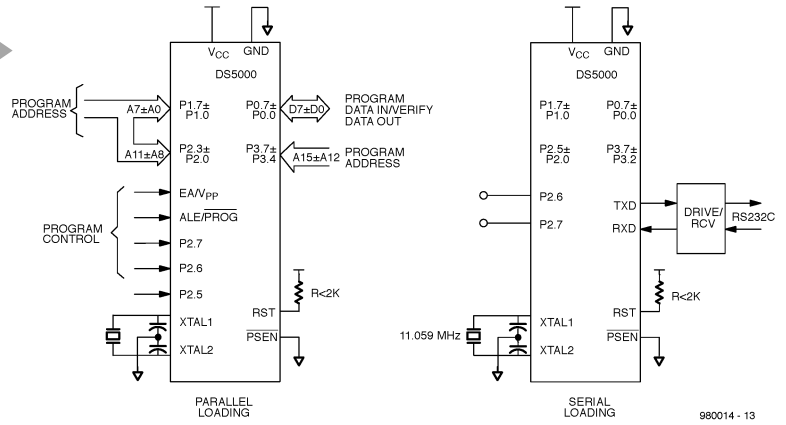
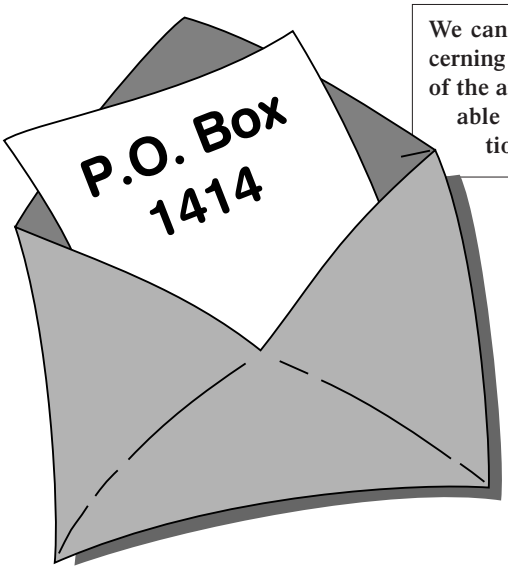


Figure 3. The controller may be programmed in two different ways. Parallel loading is common, but serial loading enables the DS5000(T) to be loaded with software during an application.

Table 2.

Command	Function
C	*Return CRC-16 checksum of embedded RAM*
D	*Dump Intel hex file*
F	*Fill embedded RAM block with constant*
K	*Load 40-bit encryption key*
L	*Load Intel hex file*
R	*Read MCON register*
T	*Trace (echo) incoming Intel hex data*
U	*Clear security lock*
V	*Verify embedded RAM with incoming Intel hex*
W	*Write MCON register*
Z	*Set security lock*
P	*Put a value to a port*
G	*Get a value from a port*



We can only answer questions or remarks of general interest to our readers, concerning projects not older than two years and published in *Elektor Electronics*. In view of the amount of post received, it is not possible to answer all letters, and we are unable to respond to individual wishes and requests for modifications to, or additional information about, *Elektor Electronics* projects.

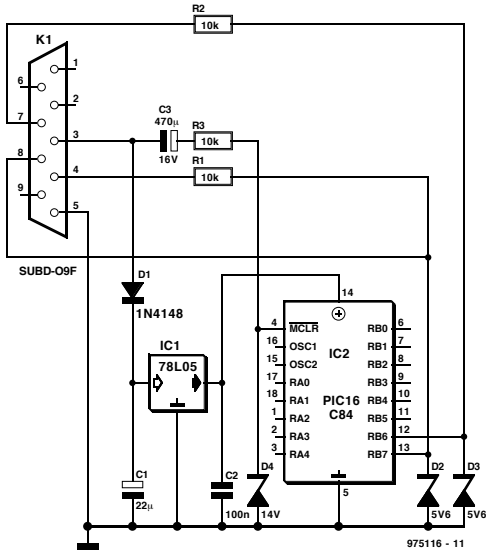
Mini PIC Programmer

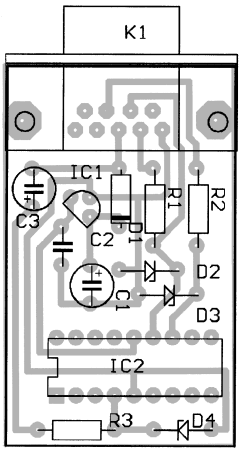
Dear Editor, in the June 1997 Supplement you covered a mini programmer for the PIC16C84. Although I have managed to build this little circuit without problems, I am unable to actually program a PIC using the PIP02 program as suggested in the article. It is impossible to program either the data EEPROM, program memory or the on-chip fuses. I have used several PIC chips, to no avail. I am therefore asking you to indicate as soon as possible (by fax) if special measures are required. Note that the serial port supply of various computers I have used is at ±12 V, and that the serial port is not used out of spec. Moreover, the mini programmer works perfectly for EEPROMs type 24C02 (also using the PIP02 utilities).

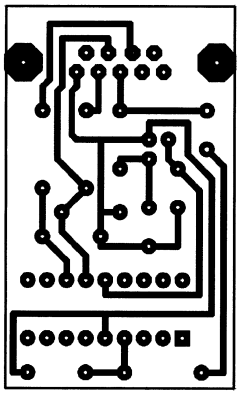
B. Alcoverro (France)

The above is an example of several letters, faxes and e-mails we received on the subject of the mini PIC programmer. It seems that some PCs, portables and others, do not supply a sufficiently high voltage on the serial port lines. While a number of readers have only reported problems, others, like Mr. Weber from Germany, have come up with solutions. Apparently, many PCs supply only 8 to 9 volts and not 12 volts as specified; some PCs even less (no hope of ever being able to use these with the programmer). Mr. Weber's solution is simple: it is sufficient to modify the value of just one component, and add another. The additional component is a 470-µF, 16V electrolytic capacitor, C3, which is connected between resistor R3 (positive terminal) and pin 3 of connector K1 (TxD) (negative terminal). Before programming, TxD will be at about – 8 V; C3 is then charged via ground, D4 and R3. When the programming sequence begins, TxD swings to about +8 V, to which the voltage across C3 is added. The result is a voltage of about 16 V across R3. If zener diode D4 is replaced by a 13 V or 14 V type, the charge built up in C3 (470 µF) should be sufficient to cover the time needed to program a PIC16C84. The author, Mr. Gueulle, advises that he has no objection against this modification, because it will only be required in a relatively small number of cases.

The modified programmer circuit as suggested by Mr. Weber, and his PCB design, are shown here.







CPU Overclocking

Dear Editor, the table on page 9 of the October 1997 Supplement on PC Upgrading indicates three different clock speeds for the 6x86M-PR166 processor. I believe the processor types should have read PR-133, PR-166 and PR-200. Or am I wrong?

By the way, I have achieved good results using the tips you provided on CPU overclocking: I now run a K5 PR-166 CPU (which has 66 MHz external clock printed on it) at a system clock speed of 75 MHz. During booting the BIOS now reports a K5 PR-200.

Regarding CPU coolers may I suggest that there are also 38-mm high coolers available (e.g. ICK type Pen 38), which cost about the same as 7-10 mm high types, yet provide nearly double the cooling capacity. The only disadvantage is that you need to purchase the fan separately (which is not required in all cases) and secure it by means of screws. The higher heatsink guarantees a lower air flow resistance, which may be fully exploited by using a 50-mm fan. The resulting cooling is so good on the over-clocked K5 that the temperature rise is just 5 degrees. It should be noted, though, that my PC has an additional fan to cool the main board.

G. Friday (by e-mail)

Regarding your question, it is answered by the section on the Cyrix/IBM CPUs which you may find on page 11 in the same Supplement. There you will be able to read that the 6x86MX-PR233 CPU gets its name from the fact that it offers the same performance as a Pentium II CPU running at 233 MHz. So, the first entry in the table (6x86MX-PR166) is correct (internal clock: 150 MHz), while the next two should be corrected to read -PR200 with the 166 MHz type, and -PR233 with the 188 MHz type.

We welcome your tips regarding CPU cooling as they may be of interest to many other readers. A pity, though, that 50-mm (square) fans are rare birds, and

apparently only available as surplus items.

Video Copy Processor ready-built?

Dear Editor, I wish to inquire about an article you published in your November 1997 issue. The article in question is about the 'Video Copy Processor' designed by W. Foede. The article explains how 'Macrovision' works and details how to build a device which 'stabilizes/eliminates' the signal. Although I have a knowledge of electronics, all be it somewhat limited, I still do not have the necessary abilities to construct this project. What I wish to inquire about is, is there a service you provide which allows the public to have the products/devices you publish, be built? On page 32, under the heading 'Construction', the articles states: "For your reference, and for those of you who insist on making the board themselves, the artwork of the singled-side printed circuit board is shown in Figure 4". By this, I am led to believe that there is service which provides my request, perhaps it is an extension on your Reader Services. If my request cannot be made, is it possible for you to put me in contact with someone who can, e.g., the writer of the article?

Christopher Burke (by e-mail)

Hi Christopher, welcome to the wonderful world of Elektor Electronics magazine. If you want to build the projects published in our magazine, there are basically three options:

***1.** Purchase the PCB and any software item(s) (like a preprogrammed PIC or EPLD) from our Readers Services, and then shop around for the other components.*

***2.** Make the PCB yourself (pretty difficult, hence our note in the article), purchase the software item(s) (like the preprogrammed EPLD) from our Readers Services, and then shop around for the other components.*

***3.** Buy a complete kit (this usually includes the PCB and software items, if applicable) from a kit supplier. Our advertisers C-I Electronics and Stippler supply kits as well as individual hard-to-get components for many of our published projects, including the Video Copy Processor. C-I Electronics may be reached by e-mail on dil@euronet.nl, or see their adverts in this and last month's issue for price and ordering information. Mind you, you still have to assemble the kit yourself, we are not aware of any kit build-up service in the UK. Perhaps other readers can help?*

Hybrid Power Amplifier – Copyright Violation

Dear Editor, I wish to raise attention to an article presented in your September 1997 issue, in which an audio stage is in direct violation of a copyrighted design by Audio Enhancement Design, now Chris Found Designs, and the subsequent manufacture of this circuit by a then UK company.
The output stage of the Hybrid Power Amplifier design by E. Wincek is a copy of a design manufactured by Beard Audio Ltd., and is subsequently still in manufacture by an Italian franchise. This company is UKD, Preganziol, Italy, but all correspondence for this design should be made to Chris Found Designs, 9, 437 Fulham Road, Chelsea, London SW10 9TY. Phone 0171-460-2143, fax 0171-924-5501.
Furthermore, the PCB was originally designed by Gray PCB Design in St. Ives, Cambridgeshire.

Christopher Found,
Chris Found Designs,
Chelsea, London.

We have investigated this matter using documents you kindly supplied (circuit diagram and an original PCB layout plot), and found that the design as published is only marginally different from yours. We apologize for not identifying Audio Enhancement Design (now Chris Found Designs) as the rightful copyright owner. Despite the copyrights, the project may be built for personal purposes only, i.e., advice on any commercial use of the design should be sought from Chris Found Designs.

Corrections & Updates

4-way serial port switch

October 1997 - 970057
The correct value for capacitors C4 and C5 is 10µF, 25V, not 4.7µF, 25V as stated in the Components List on page 31.

PIC-controlled home alarm system

April 1997 - 970022
In some cases, D7 starts to flash rapidly although the alarm should go off. Normally, rapid flashing of D7 indicates an alternating-voltage fault. The problem may be solved by increasing the value of C2 to 33µF.

Function Generator

June 1995 - 950068
If the decimal point and the colon (:) are not displayed correctly, IC2 (in the frequency meter section) has to be replaced by a type 74HC7266.

ADC for Centronics port

July/August 1997 - 974088
Lines 130 and 140 in the program printed with this design generate a clock pulse too early. Once CS has dropped low, D7 is available on D-out (IC2). Lines 130 and 140 however cause D7 to be overwritten by D6, thus killing the MSB. This happens again in line 150 as a result of the assignment i=1.
The recommended remedy is to (1) delete lines 130 and 140 from the program, and (2) change line 150 to read: FOR i = 0 to 7.

Switchboard

Switchboard allows PRIVATE READERS of *Elektor Electronics* one FREE advertisement of up to 106 characters, including spaces, commas, numerals, etc., per month. The advertisement MUST relate to electronics, and it MUST INCLUDE a private telephone number or name and address; post office boxes are NOT acceptable.
Elektor Electronics (Publishing) can not accept responsibility for any correspondence or transaction as a result of a free advertisement or of any inaccuracy in the text of such an advertisement.
Advertisements will be placed in the order in which they are received. Elektor Electronics (Publishing) reserve the right to refuse advertisements without giving reasons or without returning them.

SWAP Tektronix/Telequipment D65 dual-beam scope in g.w.o. for B♭ clarinet, must also be in g.w.o. Phone Graham on (01142) 483587.

WANTED National cathode-ray oscilloscope model VP-513A manuals or diagram. Willing to pay. John S. Syros, 26 Kyprou Str., Athens 141-22, Greece.

FOR SALE Kits for 1.2 GHz Multifunction Frequency Meter (EE Dec 1992). Unmatched project. Kit includes EPROM, PCB, LCD, front panel, drilled case, all parts and description. Few available, ultra-low price £75 each. Anita, Sibberkerkstr 100, NL-6301-AW Valkenburg, Netherlands. Email techtext@worldonline.nl.

apparently only available as surplus items.

Video Copy Processor ready-built?

Dear Editor, I wish to inquire about an article you published in your November 1997 issue. The article in question is about the 'Video Copy Processor' designed by W. Foede. The article explains how 'Macrovision' works and details how to build a device which 'stabilizes/eliminates' the signal. Although I have a knowledge of electronics, all be it somewhat limited, I still do not have the necessary abilities to construct this project. What I wish to inquire about is, is there a service you provide which allows the public to have the products/devices you publish, be built? On page 32, under the heading 'Construction', the articles states: "For your reference, and for those of you who insist on making the board themselves, the artwork of the singled-side printed circuit board is shown in Figure 4". By this, I am led to believe that there is service which provides my request, perhaps it is an extension on your Reader Services. If my request cannot be made, is it possible for you to put me in contact with someone who can, e.g., the writer of the article?

Christopher Burke (by e-mail)

Hi Christopher, welcome to the wonderful world of Elektor Electronics magazine. If you want to build the projects published in our magazine, there are basically three options:

***1.** Purchase the PCB and any software item(s) (like a preprogrammed PIC or EPLD) from our Readers Services, and then shop around for the other components.*

***2.** Make the PCB yourself (pretty difficult, hence our note in the article), purchase the software item(s) (like the preprogrammed EPLD) from our Readers Services, and then shop around for the other components.*

***3.** Buy a complete kit (this usually includes the PCB and software items, if applicable) from a kit supplier. Our advertisers C-I Electronics and Stippler supply kits as well as individual hard-to-get components for many of our published projects, including the Video Copy Processor. C-I Electronics may be reached by e-mail on dil@euronet.nl, or see their adverts in this and last month's issue for price and ordering information. Mind you, you still have to assemble the kit yourself, we are not aware of any kit build-up service in the UK. Perhaps other readers can help?*

Hybrid Power Amplifier – Copyright Violation

Dear Editor, I wish to raise attention to an article presented in your September 1997 issue, in which an audio stage is in direct violation of a copyrighted design by Audio Enhancement Design, now Chris Found Designs, and the subsequent manufacture of this circuit by a then UK company.
The output stage of the Hybrid Power Amplifier design by E. Wincek is a copy of a design manufactured by Beard Audio Ltd., and is subsequently still in manufacture by an Italian franchise. This company is UKD, Preganziol, Italy, but all correspondence for this design should be made to Chris Found Designs, 9, 437 Fulham Road, Chelsea, London SW10 9TY. Phone 0171-460-2143, fax 0171-924-5501.
Furthermore, the PCB was originally designed by Gray PCB Design in St. Ives, Cambridgeshire.

Christopher Found,
Chris Found Designs,
Chelsea, London.

We have investigated this matter using documents you kindly supplied (circuit diagram and an original PCB layout plot), and found that the design as published is only marginally different from yours. We apologize for not identifying Audio Enhancement Design (now Chris Found Designs) as the rightful copyright owner. Despite the copyrights, the project may be built for personal purposes only, i.e., advice on any commercial use of the design should be sought from Chris Found Designs.

Corrections & Updates

4-way serial port switch

October 1997 - 970057
The correct value for capacitors C4 and C5 is 10µF, 25V, not 4.7µF, 25V as stated in the Components List on page 31.

PIC-controlled home alarm system

April 1997 - 970022
In some cases, D7 starts to flash rapidly although the alarm should go off. Normally, rapid flashing of D7 indicates an alternating-voltage fault. The problem may be solved by increasing the value of C2 to 33µF.

Function Generator

June 1995 - 950068
If the decimal point and the colon (:) are not displayed correctly, IC2 (in the frequency meter section) has to be replaced by a type 74HC7266.

ADC for Centronics port

July/August 1997 - 974088
Lines 130 and 140 in the program printed with this design generate a clock pulse too early. Once CS has dropped low, D7 is available on D-out (IC2). Lines 130 and 140 however cause D7 to be overwritten by D6, thus killing the MSB. This happens again in line 150 as a result of the assignment i=1.
The recommended remedy is to (1) delete lines 130 and 140 from the program, and (2) change line 150 to read: FOR i = 0 to 7.

Switchboard

Switchboard allows PRIVATE READERS of *Elektor Electronics* one FREE advertisement of up to 106 characters, including spaces, commas, numerals, etc., per month. The advertisement MUST relate to electronics, and it MUST INCLUDE a private telephone number or name and address; post office boxes are NOT acceptable.
Elektor Electronics (Publishing) can not accept responsibility for any correspondence or transaction as a result of a free advertisement or of any inaccuracy in the text of such an advertisement.
Advertisements will be placed in the order in which they are received. Elektor Electronics (Publishing) reserve the right to refuse advertisements without giving reasons or without returning them.

SWAP Tektronix/Telequipment D65 dual-beam scope in g.w.o. for B♭ clarinet, must also be in g.w.o. Phone Graham on (01142) 483587.

WANTED National cathode-ray oscilloscope model VP-513A manuals or diagram. Willing to pay. John S. Syros, 26 Kyprou Str., Athens 141-22, Greece.

FOR SALE Kits for 1.2 GHz Multifunction Frequency Meter (EE Dec 1992). Unmatched project. Kit includes EPROM, PCB, LCD, front panel, drilled case, all parts and description. Few available, ultra-low price £75 each. Anita, Sibberkerkstr 100, NL-6301-AW Valkenburg, Netherlands. Email techtext@worldonline.nl.

Uninterruptable Power supply (UPS) for cordless telephones

keep phoning during power cuts

Although the electricity supply will rarely fail in most of the UK, unexpected power cuts lasting several hours are not uncommon in many other countries where this magazine is published. Most inexpensive short-range cordless telephones of the 'base and handset' type can no longer be used to make calls when the mains supply fails. This article describes a simple, low-cost battery backup unit which enables you to keep telephoning during a power cut.

Design by Pradeep G.

As opposed to a normal telephone set, a cordless phone is useless when there is no mains voltage. In most cases, the telephone network will keep functioning despite a power cut in your home, in your street, or even in your district. This is because the affected telephone exchange(s) automatically switch to an emergency generator for their power supply, which enables them to keep providing the ring voltages and line currents which are normally needed to place, uphold and receive telephone calls.

Inexpensive cordless phones for use in and around the home usually consist of a handset, a base unit and a mains adapter. The mains adapter is plugged into a mains socket, and powers the base unit. It also charges the battery in the handset via a pair of electrical contacts.

A telephone call is a two-way simultaneous communication, which is also called *full duplex*. Let's examine what happens in a typical cordless phone. The speech signal you receive from the other party is picked from the telephone line and conveyed to the handset using a low-power transmitter. The handset transmits your own speech signal to the base unit, where it is picked up by a receiver. In true duplex fashion, the base unit and the handset will typically use different frequencies.

Everything will work just fine as long as the base unit is powered. If the mains power disappears, however, two of the four communication elements in the system, the receiver and the transmitter in the base unit, are unable to work because there is no supply voltage. Of course, there is a battery in the handset, but the system as a whole is still useless under these circumstances,

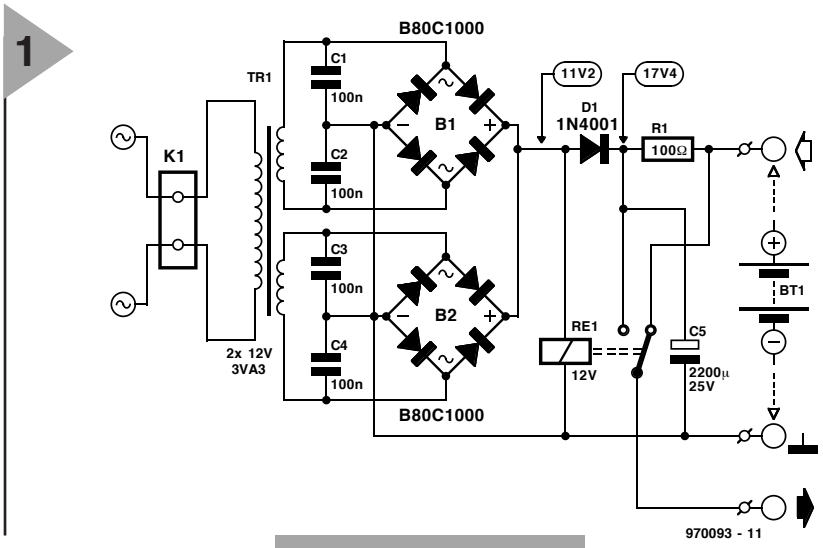


Figure 1. Circuit diagram of the UPS, a mains supply annex battery charger. The value of resistor R1 depends on the capacity of the NiCd cells you wish to use.

rare as they may be. Depending on the electronics inside the base unit, the unit may still produce its ringing sound when a call is received during a mains outage. If you pick up the handset, however, you will only hear a lot of noise because the transmitter in the base unit is unable to go 'on the air'. The situation is even worse if you are, say, in the garden with your handset on standby, because you will never even know that the power has failed on the base unit, and calls will not get through to you. More importantly, you will not be able to make a call in an emergency situation.

The only solution to overcome these problems is to equip the base unit with an *uninterruptable power supply* (UPS), which 'takes over' when the mains supply fails. Since a battery will be used as the emergency power source, the power cut should not last for more than a couple of hours (depending on the battery capacity, see further on). Fortunately, very long power cuts will be few and far between, and the UPS as described here will give satisfactory results in all but extreme cases.

HOW IT'S DONE

The basic idea behind the present circuit is to replace the mains adapter that came with the cordless phone set with a combination of a mains supply, a battery charger and a 12-V rechargeable battery pack consisting of ten series-connected 1.2-V NiCd (nickel-cadmium) cells. The aim of the project was to make an inexpensive but reliable UPS which can be built by beginners. However, to prevent disappointment, do check beforehand if you can use this project: you will not be able to use it if the base unit has an internal power supply or if the original mains adapter supplies an unusually low or high voltage, say, outside the range from 10 to 18 V a.c. or d.c. There is no problem if the adapter supplies, say, 12 V a.c. because the direct voltage supplied by the UPS (i.e., the mains supply or the battery) will not harm the internal rectifier.

The circuit diagram of the UPS is shown in **Figure 1**. It has one input, for the mains voltage, and two outputs, one for the rechargeable battery, and one for the base unit. The two outputs share the common ground line.

First, let's examine what happens in the fortunate case of the mains voltage being present. Each of the two secondary windings of mains transformer Tr1 then supplies approximately 12 V a.c. Each voltage is individually rectified by a bridge rectifier type B80C1000. This type number, by the way, indicates a peak reverse voltage (PIV) of 80 V and a maximum continuous output current of 1,000 mA (1 A). Anti-rattle capacitors C1 through C4 help to reduce mains-borne interference. The large smoothing (reservoir) capacitor you would normally expect across the output of a bridge rectifier is located behind a diode, D1. The 'raw' direct voltage produced by the rectifiers is sufficient to energise the coil of relay Re1. As a result, the relay contact is pulled in, and the smoothed, direct voltage across C5 is applied to the base unit. Note the voltage readings we have indicated across the relay and the smoothing capacitor, C5. The former is 'only' 11.2 V because it indicates an effective value. C5, on the other hand, carries the peak value of

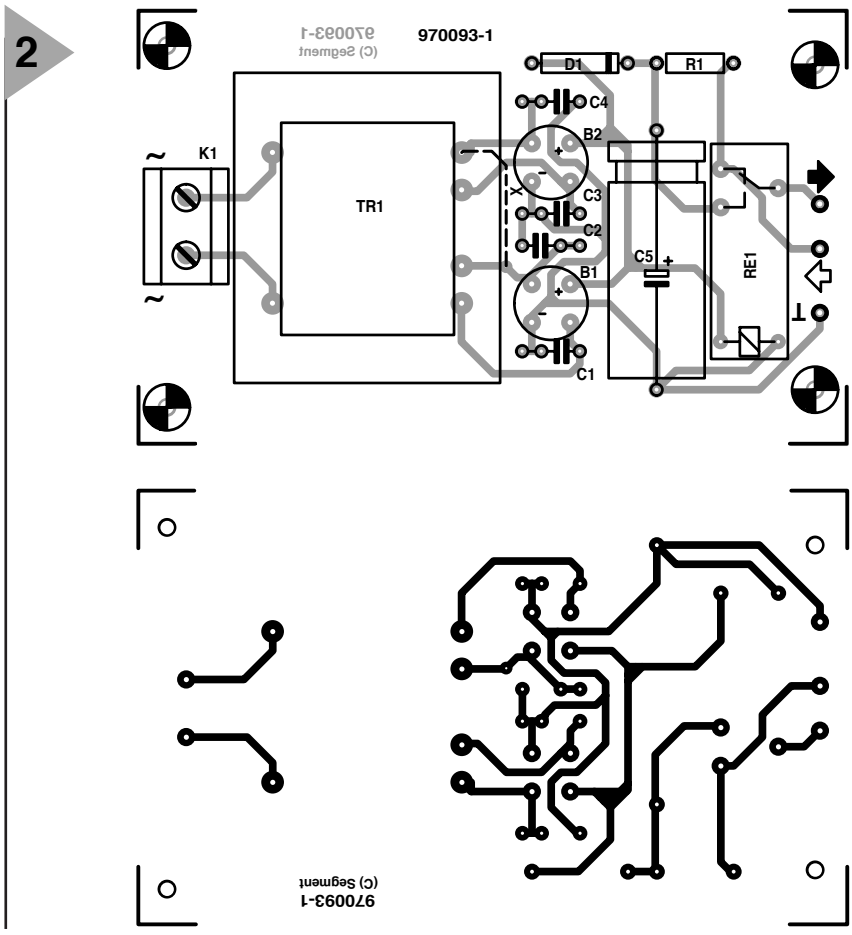


Figure 2. Copper track layout and component mounting plan of the PCB designed for the UPS (board not available ready-made).

COMPONENTS LIST

Resistors:
R1 = 100Ω (see text)

Capacitors:
C1-C4 = 100nF
C5 = 2200μF 25V

Semiconductors:
B1,B2 = B80C1000
D1 = 1N4001

Miscellaneous:
K1 = PCB terminal block, 2-way, pitch 7.5mm
TR1 = 12V 3VA3 transformer, e.g., Monacor 3112, or 2x12V 3VA3, e.g. Monacor 3212
Bt1 = external rechargeable battery pack, NiCd, 12V
Re1 = 12V relay E-card (Siemens) V23127-B2-A101
Printed circuit board not available ready-made

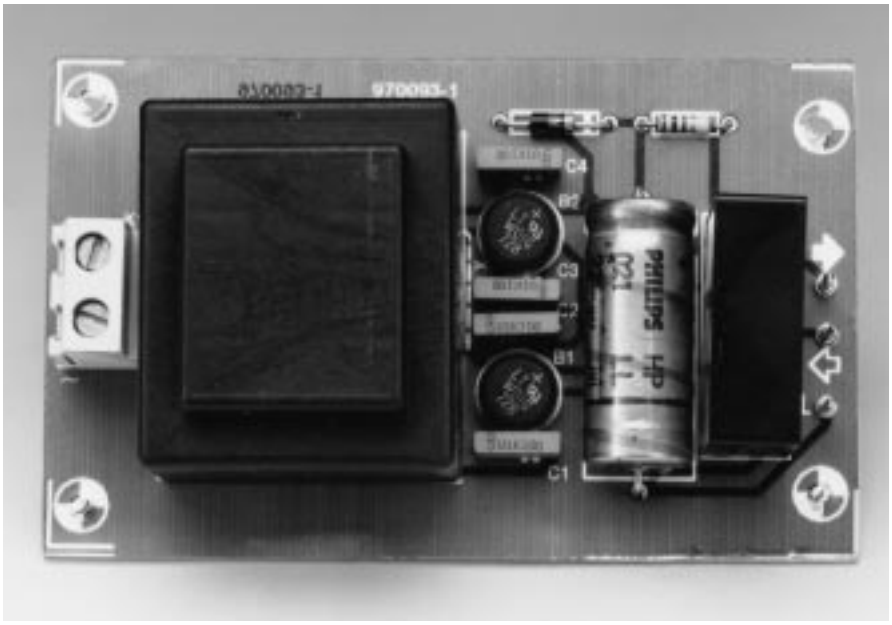


Figure 3. Finished board, just before it is mounted into its plastic case.

the same voltage, i.e., about 17.4 V (under no-load conditions).

The voltage across C5 is also used to charge the battery pack by way of a series resistor, R1. More about this further on.

When the mains voltage disappears, the relay will almost instantly lose its coil voltage because D1 blocks the higher voltage which still exists at its cathode side. Consequently, the relay contact falls back, and connects the + terminal of the battery to the + input of the base unit. D1 also prevents the battery from taking over the relay coil supply. The ground line, as you will recall, is not switched because it is shared between the battery and the mains supply/charger. The charge built up in C5 helps to ensure that the base unit is never actually without power, even if there is a short mains interruption.

R1 AND THE BATTERY PACK

This circuit is designed to charge a pack of 10 NiCd cells, **not** a (sealed) lead-acid or gel battery. In most cases, NiCd cells withstand (nearly) continuous charging fairly well provided the charging current is smaller than about $1/40^{th}$ part of the *nominal capacity*. This capacity you will typically find printed on the battery, and is usually expressed in mAh (milli-ampère-hour). So, for 800-mAh batteries, you would choose a charging current of 800/40 or about 20 mA. This is not critical, though, and in fact any value between 15 mA and 30 mA will be just fine. In practice, you have to establish the charging current, and adapt the value of R1 until the desired current is achieved. The charging current is readily calculated by measuring the voltage drop across R1 and dividing it by the value of R1. The higher the battery capacity, the lower the value of R1, and the longer the power cut the UPS is capable of covering.

CONSTRUCTION

The PCB design for the UPS is shown in **Figure 2**. Regrettably, this board is not available ready-made through our Readers Services, so you have to produce it yourself, or have it produced. All parts are mounted on to the board as indicated by the component overlay. Be sure to fit the bridge rectifiers in accordance with the ~, – and + symbols that you may see printed on top of the devices — check against the PCB overlay! To assist in their cooling, these rectifiers must be mounted at some distance above the board surface. The same applies to resistor R1, which may run a bit hot to the touch.

Read the inset on the transformer and the bridge rectifiers if you want to use a transformer with a single secondary winding.

The finished circuit board should be fitted in an all-plastic case with due attention paid to electrical safety, in particular, the connection between the board and the mains. Be sure to use properly rated cable and a rubber grommet plus strain relief. If you feel uneasy about mains cable connections, ask a more experienced friend for assistance.

FINAL HINTS

Make sure you know the polarity of the low-voltage d.c. input socket on the base unit, because you may have to make a new cable to connect the UPS. Also measure the no-load output voltage of the mains adapter. In most cases, the base unit will have an internal 8-volt voltage regulator, which will happily accept any unregulated input voltage between 12 and 18 volts.

Once you have the UPS up and running, your cordless telephone *always* works. (970093-1)

Two secondaries, two bridges

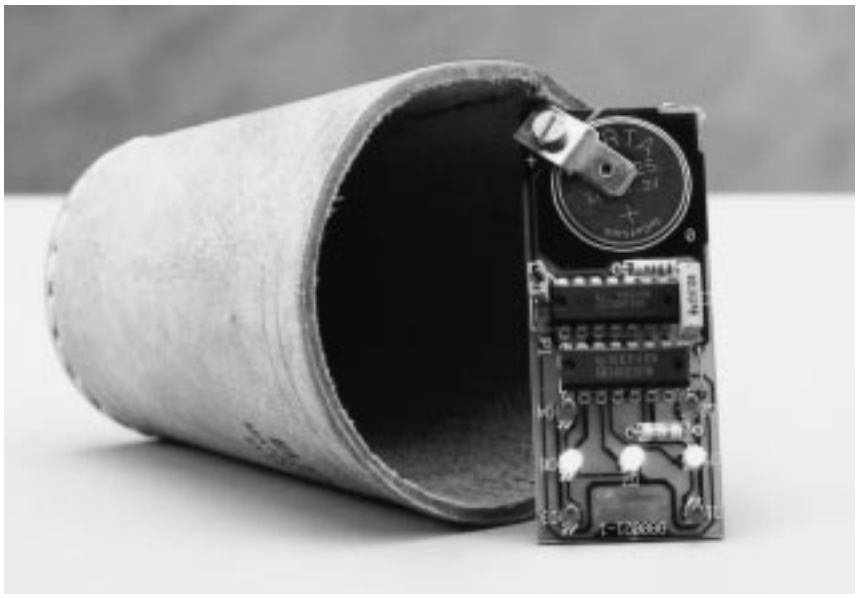
Some of you may wonder why the two secondary windings of the mains transformer are connected to separate bridge rectifiers rather than in parallel. The reason is that many of the currently available low-power PCB-mount transformers have two secondary windings which supply slightly different voltages. As opposed to modern toroid transformers, these windings are not matched using bifilar winding techniques. The resulting difference between the exact secondary voltages may cause appreciable losses (dissipation) if the windings are connected in parallel, because the winding supplying the higher voltage will attempt to compensate the difference.

The solution with two bridge rectifiers as adopted here prevents unnecessary self-heating of the transformer. Because of the blocking action of the diodes in the bridge rectifiers, the highest rectified voltage automatically appears at the output, and it is not possible for one winding to load the other.

No problem if you happen to have a transformer available with a single 12-V secondary, which (1) fits on the board and (2) is capable of supplying the necessary current: in that case, omit B1, C1 and C2, and fit a wire link, X, as indicated by the dashed line on the component overlay.

electronic die

powered by 3 V lithium battery



This article describes an electronic version of an age-old tool for games of chance: the die. It is, however, not cubical, but the number of spots on the six faces of the traditional cube are represented by LEDs on a small printed-circuit board that light when a contact is touched by hand. Powered by a small 3 V lithium manganese button cell, the compact device may also be used as a trinket for which a clip can be soldered on to the board.

Design by P. Verhoosel

The die is built on a printed-circuit board measuring only 68 × 28 mm (2.68" × 1.10") and consists of just two integrated circuits, seven light-emitting diodes and some passive components. Its spots are formed by lighting diodes which are actuated randomly when two tracks on the printed-circuit board are touched by finger. The die is powered by a 3 V lithium manganese battery for which provision is made on the board.

CIRCUIT DESCRIPTION

The size of the circuit is very small: without special tools it would be hardly possible to build a smaller one. The circuit, whose diagram is shown in **Figure 1**, consists of an oscillator, a binary scaler and a number of LEDs.

The binary (scale-of-two) counter is formed by IC_1, which has been configured to cause the traditional die patterns to be displayed by the LEDs. Output QA (pin 3) controls diode D_7, QB (pin 2), diodes D_5 and D_6, and QC (pin 6), diodes D_1–D_4. Which of the three outputs is high at any given moment is determined entirely randomly by oscillator IC_{2d}.

The scaler is started by touching (that is, short-circuiting) the circuit tracks between diodes D_1 and D_4. When that happens, clock oscillator IC_{2d} is linked via the shorted tracks and buffer IC_{2e} to the clock (CLK) input of the binary scaler (pin 14). All the die patterns are then displayed in quick succession, quick enough to make cheating impossible. When the finger is removed from the tracks, the last displayed pattern is retained.

FUNCTIONAL DESCRIPTION

Three of the outputs of the binary scaler are used to give eight (2^3) possible output states. This results in a die with values from 0 to 7, which is, of course, not in accord with a traditional, cubical die. To rectify this, some of the additional features of the IC are used.

When the highest scaler position (15) is reached, output MX/MN (maximum/minimum) (pin 12) goes high. This level is inverted by IC_{2c} and applied to input LOAD (pin 11).

When LOAD goes low, the levels at inputs A–D are read as new counter states: in this case, 9. In other words, the scaler starts at 9 every time and continues to 15, whereupon it rapidly returns to 9. The period it is at 15 is so short that it cannot be discerned: only states 9, 10, 11, 12, 13, and 14 can be noticed

In the case of the lowest three Q-outputs, this means that the LEDs are driven with states 1–6, that is,

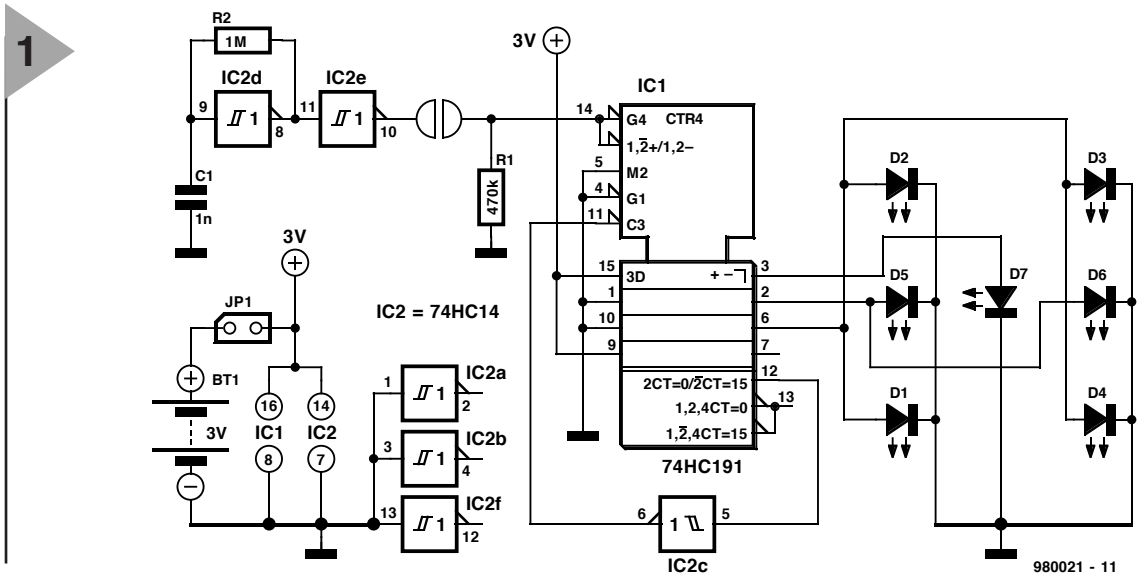


Figure 1. The circuit of the electronic die is about as small as practically possible with standard tools. Note that the unused gates of IC_2 are strapped to earth.

always eight lower than the scaler states just mentioned.

CONSTRUCTION

The die is easily built on the printed-circuit board shown in **Figure 2**. The ICs and passive components are fitted at the centre of the board, the seven LEDs in the familiar die-spot pattern, while the remaining space is intended to house the (round) lithium battery.

It is essential that a lithium manganese battery as specified in the parts list is used. A hole is provided in the board to house the battery. The rim of this is surrounded by copper. It is recommended that diagonally across this hole a strip of tin plate is soldered, which then serves as the 'bottom of the battery compartment' and at the same time as the negative terminal of the battery.

The positive terminal of the battery is formed by another narrow strip of tin plate secured to the right-hand corner of the board with a screw and nut. This ensures that the battery has no tendency to wander.

To ensure good contact between the finger and the two tracks, and the battery and the + ve and –ve battery terminals, carefully remove any solder flux from these points.

FINALLY

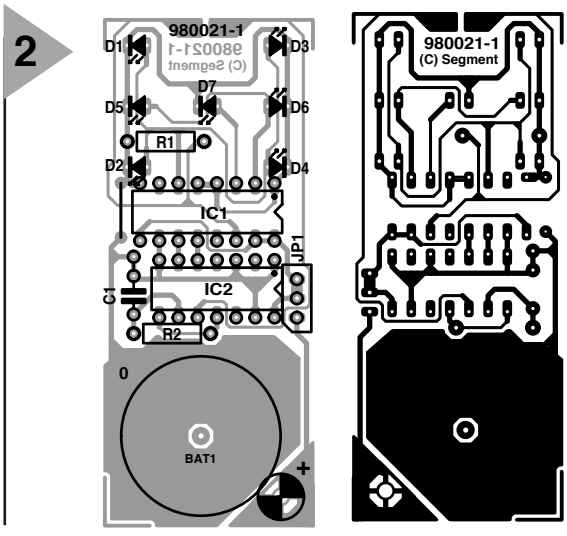
The die is switched on and off with jumper JP_1, which may, of course, be replaced by a small on/off switch.

As mentioned earlier, the 'touch' contacts are formed by the two copper tracks between diodes D_1 and D_4.

To ensure as low a current drain as possible, the LEDs should be low-current types, while IC_1 and IC_2 should be HC versions.

If the device is to be used as an adornment, it is more interesting to let the LEDs flash continuously. In this case, the clock frequency should be lowered appreciably by giving capacitor C_1 a value of 0.022 μF. Such a trinket may be made into a pin-on type by soldering (or gluing) a small crocodile clip to the bottom strip of the 'battery compartment'. [980021]

Figure 2. The tiny printed-circuit board for the electronic die is designed to house the lithium battery.



Parts list

Resistors:
R_1 = 470 kΩ
R_2 = 1 MΩ

Capacitors:
C_1 = 0.001 μF – see text

Semiconductors:
D_1–D_7 = LED, low current

Integrated circuits:
IC_1 = 74HC191
IC_2 = 74HC14

Miscellaneous:
JP_1 = two-way jumper or on/off switch (see text)
Bt_1 = 3 v lithium manganese battery, Type CR2016, CR2025 or CR2032
PCB Order No. 980021-1

Smartcard reader/writer

part 2

In last month's instalment we discussed the hardware, which consists of two sections: the card reader module and the computer interface. In this second and final instalment we will concentrate on the software utilities which are part and parcel of the project.

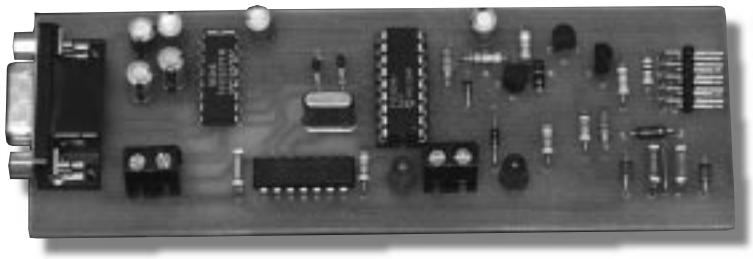


Photo du prototype du coupleur de l'auteur

Let's start by mentioning that all programs discussed in this and last month's instalment are available as executable files on a diskette which you can obtain through our Readers Services as order code **976014-1**. Alternatively, this disk may come as part of a kit supplied by one of our advertisers. Below are short descriptions of the main functions of the various software utilities on the diskette.

TELL ME...

To push on with your investigations after the initial (simple) decoding of the ATR string, you will soon find that additional software utilities are needed to be able to strike up a real 'conversation' with the Smartcard inserted into the reader module.

The utility **ELEKTINV** will be used with Smartcards of the inverse-ISO type, while **ELEKTDIR** is for cards having the straight-ISO format. Before using these utilities, the communication format (protocol) used by the card is of course established using the **ATREAD** and **ATRDEC** programs discussed last month.

The byte clusters transmitted by the Smartcard are enclosed in frames made from stars. These frames make the data chunks easy to spot later (for example, on screenshots).

It is possible at any time to send bytes to the Smartcard, simply by typing them on the keyboard. You have to stick to the hexadecimal format, though. Transmission to the card takes place instantly, byte-by-byte, and there is no need to press a key to confirm transmission. In principle, the reader module acts as the master in the dialogue with the PC: it has to send instructions to the card, which may respond or not depending on the security measures it has been programmed to enforce (in this context, 'no answer' should be interpreted as a sign of protest).

An instruction may be inbound or outbound. An inbound instruction serves to transmit a block of data to the card, while an outbound instruction requests data.

When the protocol **T= 0** applies, all instructions have to employ a group of five bytes which is generally referred to as the *header*. The basic structure of the header is shown in **Figure 5**.

The first byte in the header, **CLA**, identifies the *ISO class* of the Smartcard. The value of this code is between 00_H and FE_H, and particular to the card.

For example, CLA may read BC_H if the Smartcard is a Bull CP8 type (a French bank card) or a TB100 type from Philips. Two other examples the author came across are $A0_H$ for a GSM telephone card, and 53_H for a Sky-TV card. The default value, 00_H, is also frequently encountered.

The next byte in the header, **INS**, identifies the *instruction* or *opcode* (operational code) sent to the card. The following opcodes are 'evergreens', and recognised by nearly all Smartcards:

$B0_H$: read bytes
$D0_H$: write bytes
20_H: present code

Design by Patrick Gueulle

Although these codes already enable interesting manipulations here and there, it will always be advantageous to obtain a complete overview (dictionary) of the card type you want to tackle.

For example, here are a couple of commands which are normally supported by GSM Smartcards:

24_H:	change CHV
26_H:	disable CHV
28_H:	enable CHV
$C0_H$:	get response
32_H:	increase
04_H:	invalidate
$B0_H$:	read binary
$B2_H$:	read record
44_H:	rehabilitate
88_H:	run GSM algorithm
$A2_H$:	seek
$A4_H$:	select
FA_H:	sleep
$F2_H$:	status
$2C_H$:	unblock CHV
$D6_H$:	update binary
CD_H:	update record
20_H:	verify CHV

If you don't have information available on opcodes for your particular type of Smartcard, then there's no objection against experimenting with opcodes 'borrowed' from other cards, and simply make a note of the ones that seem to be accepted.

The bytes **A1 and A2** which follow the INS code constitute what is referred to as the *command reference*. This may be the physical address of a memory address range to be read or written, or the number of a code, a key, etc.

The fifth byte, **L**, specifies the length of the data block associated with the instruction: data which are due for transmission to the Smartcard if the instruction is inbound, or data awaited by the Smartcard if the instruction is outbound.

A length of 00_H is commonly associated with instructions which are neither inbound nor outbound: card invalidation, rehabilitation, sleep, code verification, etc.

Once it has recognised the header of a valid instruction, the Smartcard replies by transmitting a *procedure byte*. This immediately precedes the block of data returned by the card if the instruction is outbound. Alternatively, if the instruction is inbound, it is a 'request to send' aimed at the card reader.

In the latter case, it is self-evident that the card reader has to transmit *exactly* the number of bytes declared in the 'L' byte.

Most of the time, the procedure byte will be identical to the opcode (INS). It may, however, assume a value of INS+ 1 to indicate that the Smartcard requires a programming voltage (V_{pp}) to be applied to enable write instructions or code outputting.

Two bytes, **SW1 and SW2** (in rare cases, only one of these), indicate that the Smartcard has finished processing the instruction. The card transmits SW1 and SW2 either at the end of an outbound data block, or after the reception of the last byte in inbound data.

If everything is all right so far, the 'report' will read 90 00. Any other value indicates an incident, and you may have to examine SW1 and/or SW2 to find out what has happened.

A report starting with $6E_H$ or $6D_H$ indicates that the relevant instruction is not supported by the card. $6E_H$ means that the CLA byte does not tally with the class of the card, while $6D_H$ indicates that the INS does not match any of the known opcodes.

All other reports starting with '6' indicate that the Smartcard does support the instruction but that the instruction is badly formulated: for example, $6B_H$ for 'incorrect reference' (you may be attempting to read an address which does not exist), or 67_H for 'incorrect length' (you may be presenting a code which is either too long or too short).

5

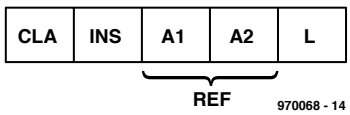


Figure 5. Structure of the header which is present in all instructions complying with the T= 0 format.

The value $6F_H$ is reserved to signal the occurrence of an incident whose cause and nature are undetermined.

The meanings of reports starting with a '9', as well as those of SW2 bytes, is not subject to regulations, and will typically depend on the application.

Meanwhile, general consensus has established the following meanings:

9010_H:	incorrect code applied (first attempt);
9020_H:	incorrect code applied (second attempt);
9040_H:	incorrect code applied (third attempt);
9080_H:	incorrect code applied (card blocked);
9001_H:	failure to write to EPROM;
9202_H:	failure to write to EEPROM;
9802_H:	invalid secret code;
9804_H:	bad code applied;
9806_H:	code cancelled;
9810_H:	security condition not satisfied;
9820_H:	inactive secret code.

Many operations will, no doubt, require some code or other to be presented which endows the Smartcard with a certain degree of security. If you want to know how to get on with these cards, you should know that the procedures differ from one card family to another.

In case of the Bull CP8 Smartcard, for example, a four-number 'card bearer' code has to be transcoded into a four-byte word, according to an algorithm which is not in the least secret. So, the PIN (personal identification number) '4950' has to be presented as $12543FFF_H$ (assuming a reference of 0000_H). Once presented, this code has to be ratified by an instruction which reads BC 40 00 00 00.

As far as the 'COS' Smartcards from Gemplus are concerned, the four-byte codes have to be extended to eight positions before they are applied. This operation consists of affixing an MS (most-significant) nibble (group of four bits) with a random value in front of each nibble. In this way, the code 45343F45H may equally well be applied as 04040304030F0405H or, more conveniently, as 54455354434F4445H (this spells TESTCODE in ASCII).

SOME HELPFUL UTILITIES

Although the four software utilities already mentioned allow you to tackle all topical operations to the most frequently used Smartcards, a couple of small programs have been added to the toolkit for special applications and advanced users.

RDINV (Read INVerted) and **RDIR** (Read DIRect) enable you to read, on inverted-ISO cards and straight-ISO cards respectively, a specified number of bytes (up to 250), starting from a specified address (which has to be in an area which can be read freely).

In addition to their hexadecimal display function, these programs also save the results of the read operation in a file called CARD.CAR, the contents of which may provide a useful starting point for all kinds of informed processing, without having to re-read the card all the time.

The utilities **INVCLASS** and **DIRCLASS** enable you to run a systematic ISO-class exploration on a totally unknown Smartcard by trying out all possible combinations within a totally harmless reading sequence.

Finally, we should temper your hopes of using this little tool to discover a confidential code on the Smartcard, as any properly protected card will permanently block itself after three unsuccessful attempts.

(970068-2)