ELEKTOR ELECTRONICS

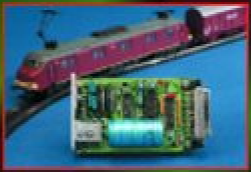
THE ELECTRONICS & COMPUTER MAGAZINE

SEPTEMBER 2000
£2.95

www.elektor-electronics.co.uk

Experiment with USB

PIC Processor Board



Digital Model Train

- PC Serial Peripheral Design Course
- Modern Surround Sound Systems
- GSM Call Indicators
- OSCAR Stand-Alone MP3 Player
- RJ45 Network Cable Tester

P.O.Box 190

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.

Valve Preamplifier

Dear Editor,

I have just finished reading the article on the Valve Preamp, and I feel that I would have to query the validity of the author's statement that using two 22 µF capacitors in parallel for the output capacitor, instead of one single 47 µF, benefits the HF transfer characteristic due to it halving the ESR.

Whilst it is true that the ESR of two capacitors in parallel will be half that of a single capacitor, this only holds true for a single capacitor of the individual value. ESR is directly related to capacitor value, thus, if a single 22 µF capacitor had an ESR of say 2 ohms, then paralleling two of these would result in an ESR of 1 ohm. However, if a single capacitor of theoretical value 44 µF from the **same manufacturer**, and from the **same series** were used, it too would have an ESR of 1 ohm on its own. This is borne out by examining the published data for capacitors of the values cited above. In fact, as the nearest value to two 22 µF's in parallel is 47 µF — i.e. slightly larger than 44 µF - the situation for a single capacitor is actually improved over the dual capacitor scheme in that this larger value consistently yields a lower published ESR value than two 22 µF's from the same series.

I would not have thought that the ESR of the output coupling was an issue anyway. At these sorts of values, the ESR of the caps is almost negligibly small compared to the load impedance of the following stage. Of far more importance, I would have thought, is the relatively high reactance (X_c) of these capacitors at low frequencies, which may well create noticeable effects, and the inherently poor HF performance of standard electrolytics.

If ESR is really an issue here, I would have expected to see a single electrolytic of perhaps 47 µF and of low ESR specification, as in electrolytics designed specifically for high frequency operation in switch mode power supplies, and perhaps for 'belt and braces' HF performance, a shunt capacitor of say 0.1 µF bypassing that single electrolytic.

Geoff R Darby, by email

The designer, Mr. Haas replies:

The sentence about halving the ESR was not well formulated. Halving the ESR may be implemented or not depending on the exact type of electrolytic capacitor available. In so far, Mr. Darby is right. The main reason for using two electrolytic caps of 22 µF each is that the total height of the devices is smaller and that the indicated types could be obtained as planned (by Mr. Haas, Ed.). This allows a more compact case to be employed, because the usual 47 µF/450 V electrolytics are taller. Also, axial electrolytics would have required more PCB space.

These days, high-voltage electrolytic capacitors are plagued by supply delays of up to 50 weeks. For this reason, I only applied components that are either available at reasonably short notice or always in stock (in Mr. Haas' company, Ed.).

As to the parallel connection of 0.1-µF capacitors, most readers have not done their sums and so failed to realize the relative importance of this equivalent capacitor in relation to a 47-µF electrolytic. Assuming a signal frequency of 20 kHz, the 47-µF capacitor represents Xc = 0.18 Ω. The typical ESR will be about 1.2 Ω. By comparison, a 0.1-µF represents Xc = 79.6 Ω at 20 kHz. Question: which, if any, effect does the parallel 0.1-µF capacitor have at 20 kHz?

An article like the one on the Valve Preamplifier always triggers discussion. The audience, however, should be advised that building and testing is more fun and productive than endless theoretical debates. Finally, the test data supplied by the Elektor Electronics sound laboratory prove that the Valve Preamplifier is not a bad performer.

G. Haas, by email

The debate continues...

Dear Editor,

With the publication of the second part this month, I would like to further take issue with the author regarding the "corrections to part 1" panel, where mention is made twice of grid 2 being the suppressor grid. It is not. Rather, it is the screen grid. The suppressor grid is grid 3, closest to the anode, and internally connected to the cathode. Its purpose is to suppress the secondary emission electrons from the anode and prevent them from being attracted back to the screen grid, which results in the well known 'tetrode kink'.

Having done a few quick calculations on the circuit, a concern has come to light regarding the operating conditions of the valve.

By looking at the quoted voltages in the HT section of the circuit, a standing current of around 44 mA give or take can be derived. This results in a voltage drop of some 147 volts across the anode load resistors leaving around 177 volts on the anode. The power dissipation in the resistors will be about 6.9 watts, which goes along with what the author suggests as the "not insignificant power dissipation" in these devices in his text.

According to my data books, the quoted typical screen grid current for a screen grid voltage of 250 volts is 6mA. This would result in a drop across the screen grid resistor, R9, of about 50 volts, resulting in a screen voltage of around 270 volts. Whilst this is slightly higher than the stated 250 volts for 6 mA current, it will not make the current significantly higher, so will not result in the screen grid voltage being much lower. Thus, by calculation, it would appear that the screen grid has a significantly higher voltage on it than the anode. I am not at all sure that this is a valid situation. It has been a long time since I learned my valve theory, but I'm sure I recall that the screen voltage should never be above the anode voltage, otherwise, all manner of unpredictable things can happen, including the screen grid trying to behave as the anode and over dissipating, and negative grid current flow.

Obviously, without having built and measured one of these designs, it is hard to say for sure if my calculations hold up in real life, but some aspects of the design do seem to me to fly in the face of convention, for instance it is unusual not to see the screen grid decoupled to ground with 0.1 µF or so. I assume that you have a finished version which your lab evaluated prior to acceptance for publication. I would be interested to know what your staff members make of my thoughts.

You are right — 'suppressor grid' was simply an incorrect translation, we should have used 'screen grid' — our apologies are due to you and Mr. Haas. As already mentioned, the test data obtained from a prototype (supplied ready-assembled to us by Mr. Haas) gave no reason for concern or, indeed, inspection of the actual valve operation conditions. Other readers are invited to comment.

Working with a Touch Screen

Dear Editor,

I found the article "Working with a Touch Screen" very interesting and, as usual, miles ahead the competition. However to date I have been unable to locate a supplier of touch screens. Can you help?

Kristof Dyrcz, by email

The author when queried informed us that he had obtained his touchscreen from a Dutch company by the name of Naamplaat B.V. On the Internet, they reside at

http://www.naamplaat.nl

After trawling the Internet for the keyword 'touchscreen' we came across another supplier, IQ Automation, at http://www.iq-automation.de

PC Serial Peripheral Design (I)

Part 1: Introduction, course hardware, first program

By B. Kainka

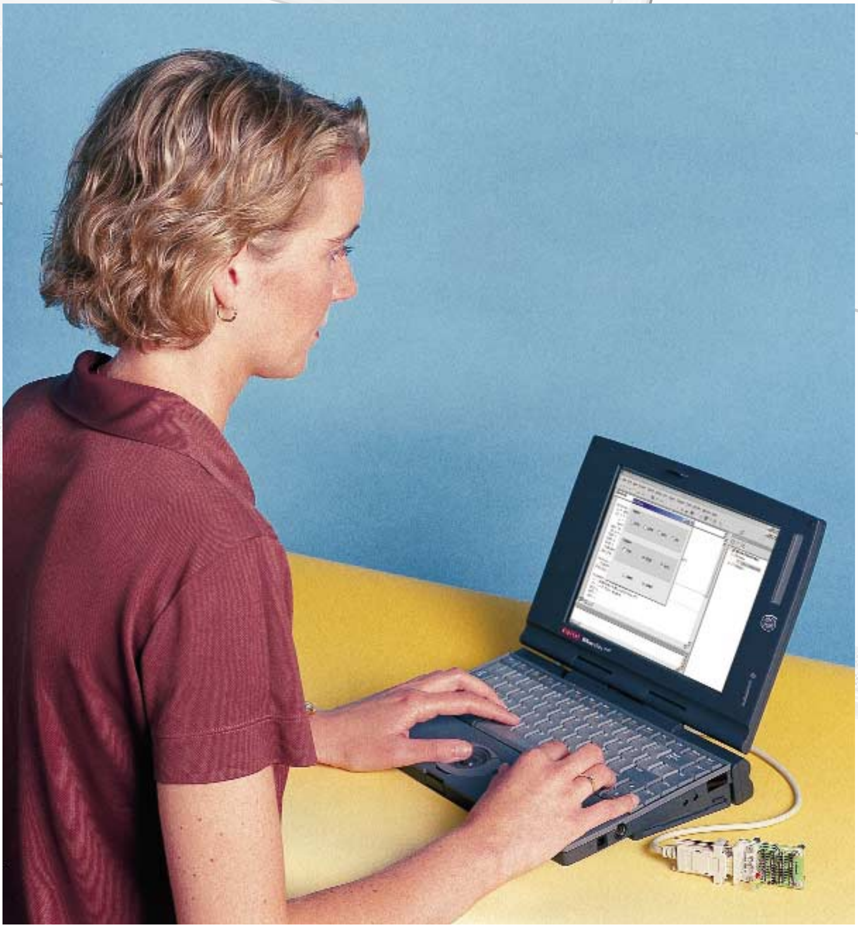
At one time or another many readers may have thought about controlling or monitoring equipment using their PC. Electronics projects using a PC need not be lavish or expensive: often the interfaces provided on the PC can be used directly. In this series of articles we present a range of projects using the serial (RS232) port, controlled using simple programs written in Visual Basic.

Why have we chosen to use the serial port, when it is more complicated and offers fewer connections than the printer port? For a beginners' course using the serial port has several advantages:

- The serial port is well protected against accidental damage. Cables can safely be plugged in while the computer is on.
- There is usually a spare serial port which can be used for experiments.
- The serial port can deliver enough current to power a wide range of projects, and so a separate power supply is not required.

This course is also easy on the pocket, which is important, not least for educational establishments and those on youth employment programmes. Apart from the small printed circuit board, you will need just a few everyday components, such as pushbuttons, NPN transistors, resistors, LEDs, capacitors, diodes and an LDR. All the connections of the serial port are brought out to sockets on the circuit board and there is also a small prototyping area for building experimental circuits.

All the projects are programmed in **Visual BASIC 5**. The programs (available on floppy disk, order code **000074-11**, or for download from *www.elektor-electronics.co.uk*) are clearly commented, so that you can easily



change them to try out your own ideas. In later instalments, we will deal with complex topics such as real-time control and connection to external circuitry. As a bonus, readers will be introduced to applications of various sensor technologies, which arise naturally through the course.

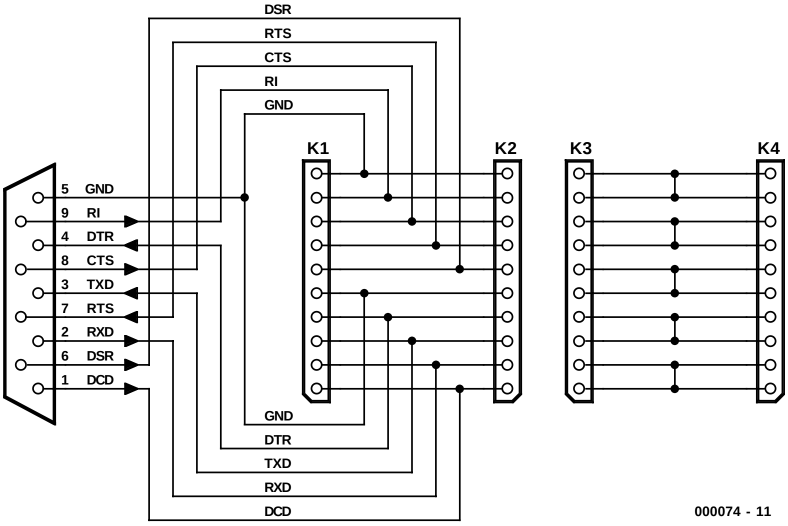


Figure 1. Circuit diagram of the experimental PCB.

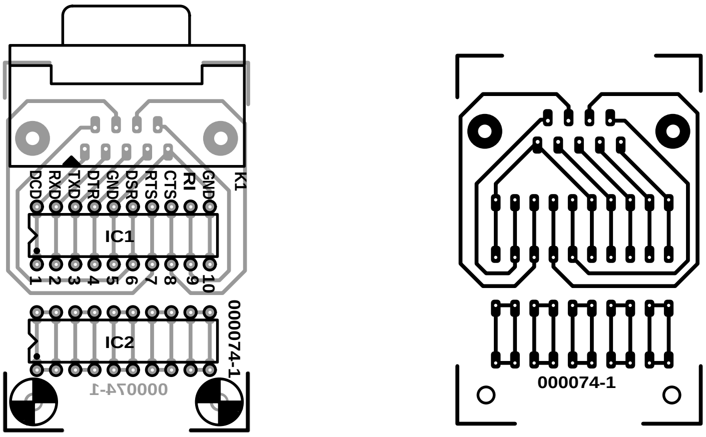


Figure 2. Component layout for the experimental PCB.

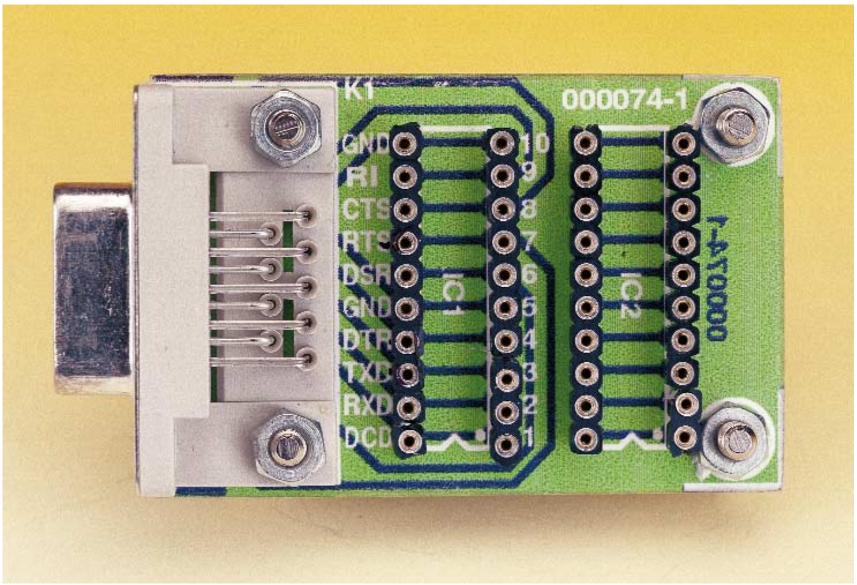


Figure 3. The assembled board.

The printed circuit board

The projects are based on a simple circuit board. As the circuit in **Figure 1** shows, it is a small prototyping board fitted with four SIL socket strips and a 9-way sub-D socket. Two of the four SIL sockets are connected to the sub-D socket, while the other two are simply connected to one another.

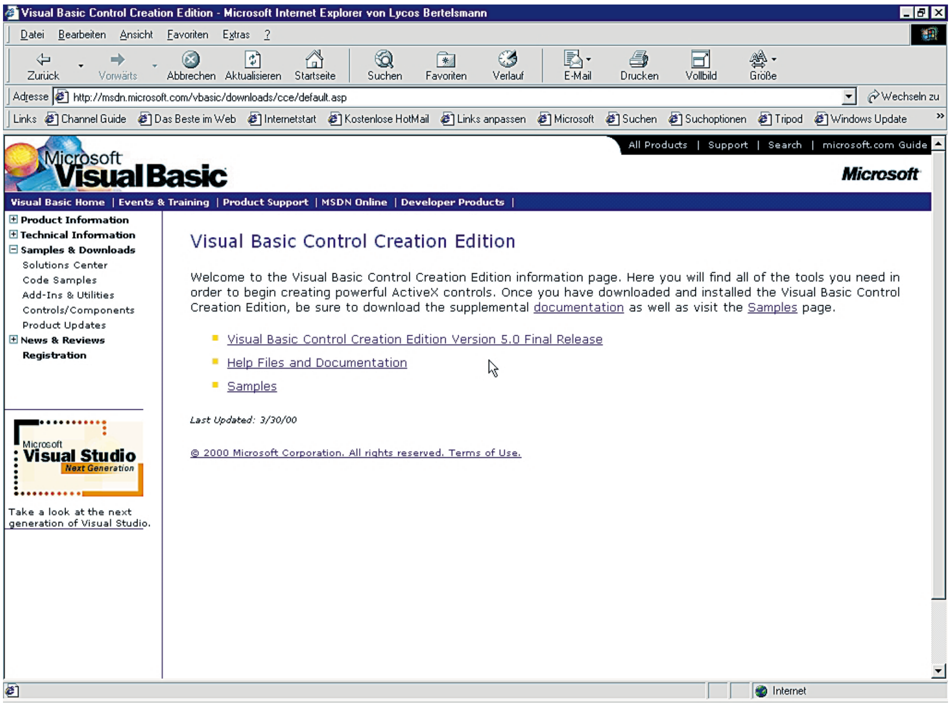
Figure 2 shows the component layout for the circuit board, and the assembled board is shown in **Figure 3**. **Table 1** gives a summary of the pinout of the 25-way and 9-way connectors normally employed for serial ports, showing the names and functions of all the signals that make up the interface. A male connector is invariably provided on the PC side, and so we require a female connector: a 9-way extension cable can be used to connect the PC to the circuit board described here. If your PC is equipped with a 25-way connector, you will need to use a suitable adapter.

Data is normally transferred over the serial interface using the TXD (transmit data) and RXD (receive data) signals. The other signals have auxiliary functions concerned with setting up and controlling data transfer. They are known as ‘handshake signals’, because they are used to acknowledge transfer of data between two pieces of equipment. A particularly useful feature of the handshake signals is that their state can be read or written directly.

The pin labelling on the circuit board follows that of the 9-way sub-D plug. Each pin of this plug is taken to two SIL socket pins, except for GND, which is taken to four. The small prototyping area on the circuit board is divided into five groups of connections with four SIL socket pins in each. Four SIL sockets with turned pins are used in total; two 20-pin DIL sockets — which are easier to obtain — can be used instead.

Visual BASIC

You will need a copy of Visual Basic, version 5 or 6, for this course. All source code is available from the *Elektor Electronics* website in VB5 format. The programs can also be loaded and compiled without modification using VB6. If you do not have a copy of Visual Basic


Visual Basic Control Creation Edition - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Verlauf E-Mail Drucken Vollbild Größe
Adresse http://msdn.microsoft.com/vbasic/downloads/cce/default.asp
Wechseln zu
Links Channel Guide Das Beste im Web Internetstart Kostenlose HotMail Links anpassen Microsoft Suchen Suchoptionen Tripod Windows Update
All Products | Support | Search | microsoft.com Guide
Microsoft Visual Basic
Microsoft
Visual Basic Home | Events & Training | Product Support | MSDN Online | Developer Products |
Product Information
Technical Information
Samples & Downloads
Solutions Center
Code Samples
Add-Ins & Utilities
Controls/Components
Product Updates
News & Reviews
Registration
Microsoft Visual Studio Next Generation
Internet

Visual Basic Control Creation Edition

Welcome to the Visual Basic Control Creation Edition information page. Here you will find all of the tools you need in order to begin creating powerful ActiveX controls. Once you have downloaded and installed the Visual Basic Control Creation Edition, be sure to download the supplemental documentation as well as visit the Samples page.

- Visual Basic Control Creation Edition Version 5.0 Final Release
- Help Files and Documentation
- Samples

Last Updated: 3/30/00

© 2000 Microsoft Corporation. All rights reserved. Terms of Use.

Take a look at the next generation of Visual Studio.

Figure 4. Visual Basic (VB5CCE) is available for download here.

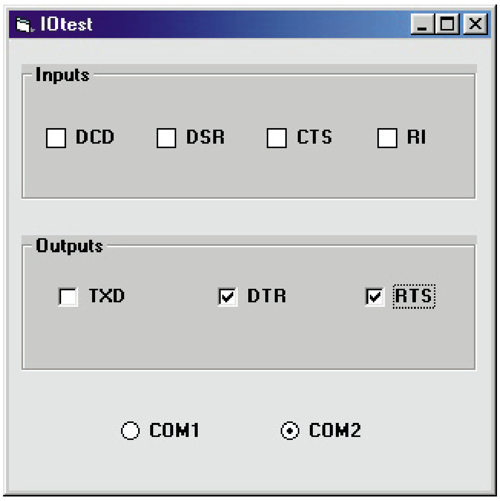
and balk at its price, you can obtain a free version of VB5 from Microsoft: 'Visual Basic Control Creation' VB5CCE is available on the Internet at

http://msdn.microsoft.com/vbasic/downloads/cce/default.asp

Our first program: I/O test

And now, at last, to our first program. It's a simple test program which gives direct access to all serial port connections (except RXD). The three outputs can be toggled with a click of the mouse, while the four inputs are read and displayed. The program is configured to use COM2 by default; however, it will recognise when COM2 is in use and automatically attempt to use COM1 instead. It is assumed that most PCs have a mouse connected to COM1 and that COM2 is unused. Frequently the connectors on PCs are not correctly labelled: in that case this program can help to identify them correctly. If you have two spare COM ports, you can use COM1 or COM2 as you prefer.

What is 'ON', and what is 'OFF'? If you connect a voltmeter between DTR and GND (ground), you will find that 'ON' gives a positive voltage of about 10 V, while 'OFF' gives a voltage of about –10V with respect to GND. Now connect a wire between, for example, the DTR output and the DSR input. As soon as DTR is turned


IOtest

Inputs

☐ DCD ☐ DSR ☐ CTS ☐ RI

Outputs

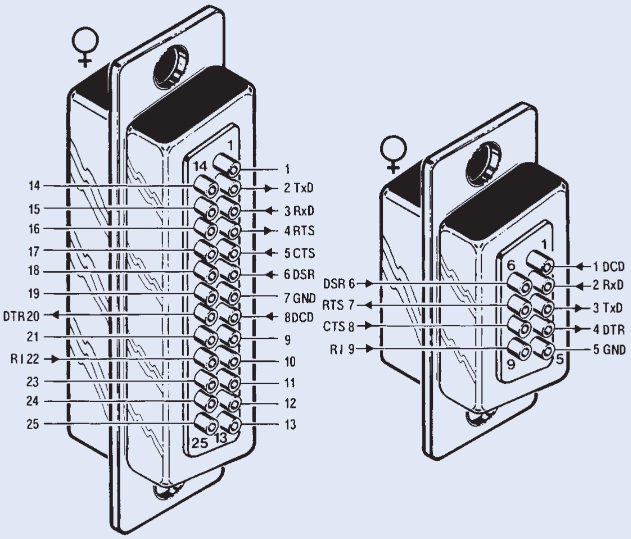
☐ TXD ☑ DTR ☑ RTS

○ COM1 ● COM2

Figure 5. Window showing direct access to serial port signals.

Table 1 Connector pinouts and signal names

Pin 25-way	Pin 9-way	Input/ Output	Symbol	Function
2	3	Out	TxD	Transmit Data
3	2	In	RxD	Receive Data
4	7	Out	RTS	Request To Send
5	8	In	CTS	Clear To Send
6	6	In	DSR	Data Set Ready
7	5		GND	Ground
8	1	In	DCD	Data Carrier Detect
20	4	Out	DTR	Data Terminal Ready
22	9	In	RI	Ring Indicator



000074 - 18

on, the DSR input reports 'ON' too. Our first practical application is to poll a switch (**Figure 6**). In order to read the state of the switch, the DTR signal must be turned on. Now we have a complete PC application: the switch can be placed in some remote location and it state will be relayed to the PC. What that might represent is up to you: you might, perhaps, want to monitor whether that mouse-trap in the cellar has been set off.

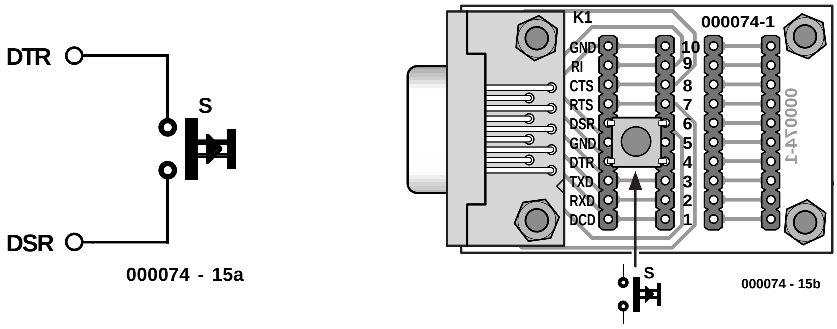


Figure 6. A simple switch is connected for the first experiment.

We round off this first instalment with another neat little application. Here a light-emitting diode is connected to and controlled by the PC (**Figure 7**). Many *Elektor Electronics* readers will protest loudly that one should never connect an LED without a current-limiting resistor. Here, though, it is not necessary, as a resistor is effectively built into the PC.

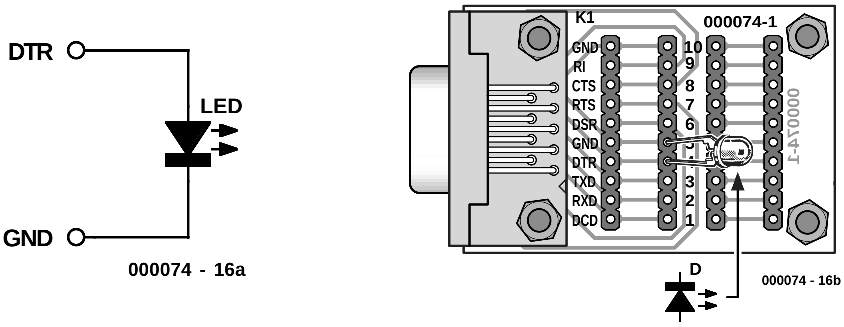


Figure 7. Equally simple: an LED is connected for the second project.

Another protest: in the datasheets, the maximum allowable reverse voltage of an LED is given as 3 V or 5 V. In this circuit, however, reverse voltages of up to 10 V can appear across the LED. Experience has shown that this is not in practice a problem. Even with a reverse voltage of 20 V no significant current flows in most LEDs. But we should provide a 'clean' solution: an additional silicon diode (for example, a 1N4148) will protect the LED from excessive reverse voltages. The diode can be connected either in anti-parallel or in series with the LED (**Figure 8**). As a rule of thumb, for serious projects the reverse voltage should be limited to 5 V; for experimental purposes we can allow ourselves a little latitude.

(000074-1)

On Project Disk #000074-II:

PORTS.BAS	845	03-28-00	5:31p
ANPEL.FRX	1,094	04-10-00	6:01p
BLINK.FRX	1,094	04-10-00	6:43p
COUNTER1.FRX	1,094	05-04-00	6:49p
COUNTER2.FRX	1,094	05-04-00	7:14p
CONTENTS.TXT	0	06-15-00	9:35a
IOTEST.FRX	1,094	05-04-00	12:21p
AMPEL.VBP	497	04-10-00	6:01p
BLINK.VBP	497	04-10-00	6:43p
COUNTER1.VBP	499	05-04-00	6:49p
COUNTER2.VBP	499	05-04-00	7:14p
IOTEST.VBP	497	05-04-00	1:13p
AMPEL.VBW	81	05-05-00	12:57p
BLINK.VBW	80	05-05-00	4:26p
COUNTER2.VBW	80	05-05-00	4:27p
IOTEST.BW	80	06-08-00	1:15p
PORT.DLL	46,080	02-07-99	1:15p
BLINK.FRM	3,488	04-10-00	6:43p
COUNTER1.FRM	3,422	05-04-00	6:49p
COUNTER2.FRM	6,683	05-04-00	7:14p
IOTEST.FRM	5,317	05-04-00	12:21p
ANPEL.FRM	3,669	04-10-00	6:01p
22 file(s)	77,784 bytes		

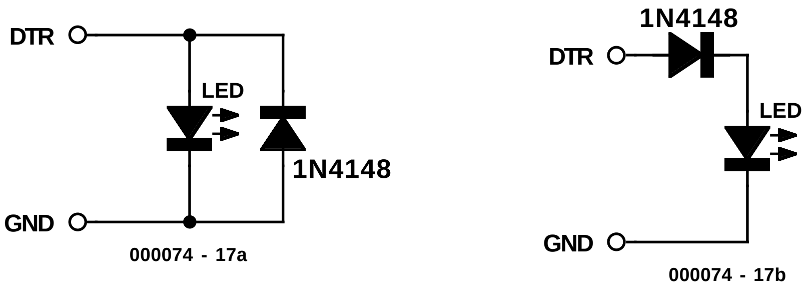


Figure 8. Two ways to protect the LED from excessive reverse voltage.

RJ45 Network Cable Tester

Check out network cables and connectors

Design by A. J. Baur

This simple tester allows faults in ISDN and Ethernet cables using RJ45 connectors to be traced quickly and efficiently.

Many Ethernet and ISDN connection systems employ sockets and plugs called 'RJ45'. They look like a slightly wider version of the RJ11 or 'Western' connectors as found on modern telephone sets. The RJ45 connector has 8 pins, although in some cases only four or six are actually used (designation: RJ45-4/8 and RJ45-6/8 respectively). The connectors are available in screened and non-screened versions. For 10 Mbit/s and 100 Mbit/s (as well as for ISDN), the unscreened variants of the plug/socket combination is sufficient, provided the cables are of the 'twisted pair' type (unshielded twisted pair, UTP). If not, crosstalk is likely to increase the risk of data corruption on long lines. Some popular network cable types and their main characteristics are shown in the **Table**.

The connectors are based on cut/clamp technology to ensure proper contact with the network cable. First, the cable sleeve is removed. Next, the individual wires are inserted, insulation and all, in small holes in the connector. Finally, the wires are pressed firmly onto the connector parts using a special crimp tool. Because this tool is expensive and hard to get, it is not surprising to see hobbyists attempting the use of less sophisticated means to connect cables to RJ45 connectors. We've even heard of people pressing the individual contacts over the cable using a sharp object.

Fitting an RJ45 connector on to a network cable can be a very exasperating exercise, especially in a badly lit place like at the end of a cable conduit. Because it is not possible to actually see if the wires are properly connected in the plug, the only reliable way of testing a cable is to check continuity of all individual wires. Unfortunately, once installed, network cables do not lend themselves very well to testing with an ohmmeter simply because the ends are too far apart. Also, even if you

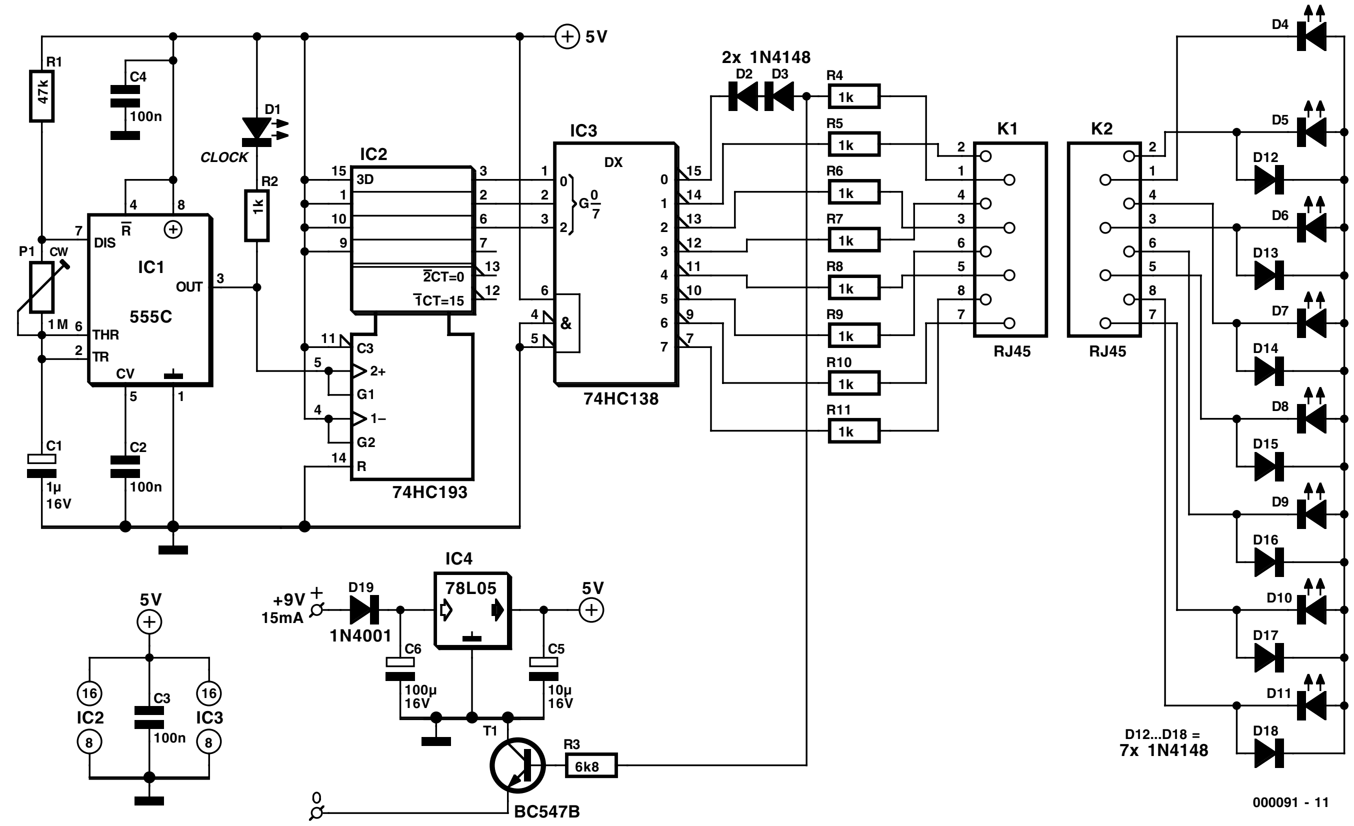


Figure 1. The two-part circuit diagram of the RJ45 Tester.

have both ends of a freshly ‘pressed’ RJ45 cable in front of you on a clean desktop with good lighting conditions, you’ll have a hard time establishing contact between the cable wires and the meter probes.

The solution to all these problems is the RJ45 tester described in this article. The tester is actually a system consisting of a sender and a receiver box, both equipped with an RJ45 socket. All eight wires in the cable are tested in one operation, no need to remove and insert the cable under test.

The tester is realized by the deceptively simple circuit shown in **Figure 1**. The sender box consists of a CMOS timer type 555, IC1, configured as an astable multivibrator whose output frequency is adjustable between 1 Hz and about 40 Hz using preset P1. The output signal on pin 3 is first applied to a counter with BCD outputs (IC2, a 73HC193), which is wired for ‘endless’ counting to produce sequences 0-7 on QA (pin 2) and QC (pin 6) all the time. The code is then applied to a BCD/Decimal converter type 74HC138 (IC3), of which an output is pulled low (active) that matches the current BCD input number.

At the receiver side we find just eight LEDs whose associated series resistors R4-R11 are located on the sender board. Across each LED sits an 1N4148 diode in anti-parallel. The function of these diodes should be clear: if a current is to flow through a selected LED, the return path is not along any of the other LEDs, because they do not conduct. So, because not all wires may be available, each LED is hunted by an anti-parallel diode, except LED D4.

This is by no means an oversight on the part of the designer. Via diodes D2 and D3, a transistor is connected to pin 15 of the 74HC138 decoder. T1 acts as a switch for the power supply, obviating an all-mechanical counterpart. When the tester is not active, junction D3-R3-R4 is at ground potential, and with it, the base of T1. Because D2 and D3 prevent base current from flowing into T1, so that the circuit is held ‘off’. That changes to ‘on’ once the LED box is connected to other end of the cable under test. This automatic power-down system makes the circuit very energy efficient.

Once the sender and receiver are connected up, the voltage on the LED common rail rises to about 4.3 V, of which about 2 V actually reach the base of T1. Incidentally, the base current flowing via D4, R4 and R3 is the cause of LED D4 lighting dimly all the time. On the other hand, D4 when activated also lights less brightly than the other LEDs because of the voltage drop on D2-R3. Fortunately, the difference between D4 being ‘on’ and ‘off’ is clear to see. When the LED is activated, it drops about 1.4 V, which is sure to keep T1 switched on.

Construction and use

Stuffing the printed circuit board shown in **Figure 2** should not cause any difficulty whatsoever. Unfortunately, the PCB shown here is not available ready-made from the Publishers.

If you make sure that all polarised parts are fitted the right way around, and the wire link next to IC4 is not overlooked, the RJ45 Tester should not take you more than an hour to finish, including fitting the sender and receiver boards in suitable plastic cases.

When using the popular 10BASE-T2 cable,

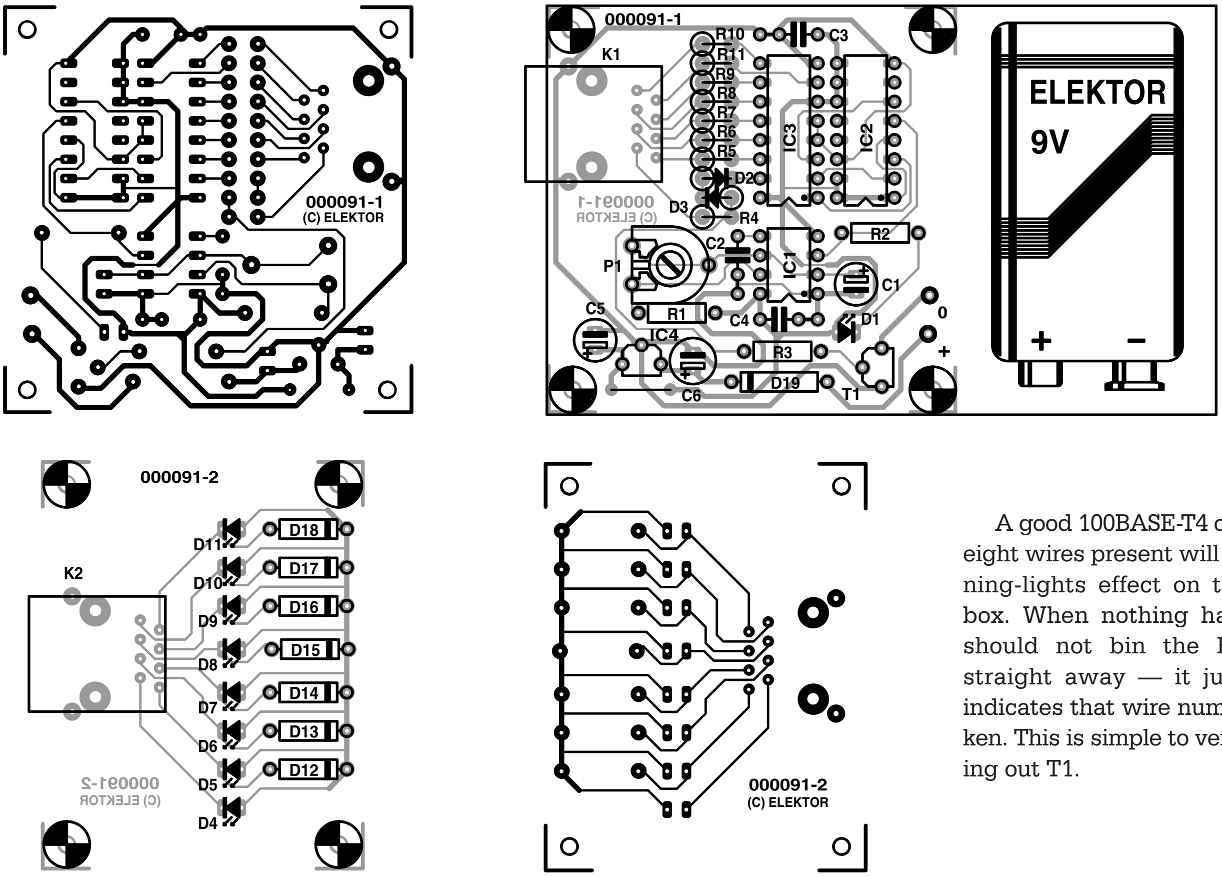


Figure 2. PCB layout designed for the RJ45 Tester (board not available ready-made).

(two twisted-pair ISDN pairs), LEDs D1, 2, 3 and 6 light in sequence. With crossed wires, the same LEDs light but the sequence is reversed (6-3-2-1). If the LED indicators light in a steady order, all cable wires are intact. When several LEDs light at the same time, one or more wires are short-circuited somewhere in the cable or its connectors. Alternatively, if one LED does not light when it should, the relevant wire or connection is broken somewhere.

A good 100BASE-T4 cable with all eight wires present will cause a running-lights effect on the Receiver box. When nothing happens, you should not bin the RJ45 Tester straight away — it just faithfully indicates that wire number 1 is broken. This is simple to verify by shorting out T1.

(000091-1)

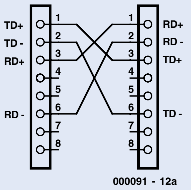
Network Cable Types

Designation	Application	No. of Wires
10BASE-T2	*10 Mbits/s Ethernet:*	*2 pairs Cat. 3 (or better), UTP*
10BASE-T Crossover	*10 Mbits/s Ethernet:*	*2 pairs Cat. 3 (or better), UTP, crossed*
100BASE-T2	*100 Mbits/s Ethernet:*	*2 pairs Cat. 3 (or better), UTP*
100BASE-T Crossover	*100 Mbits/s Ethernet:*	*2 pairs Cat. 3 (or better), UTP, crossed*
100 BASE-T4	*100 Mbits/s Ethernet:*	*4 pairs Cat. 3 (or better), UTP*
100BASE-TX	*100 Mbits/s Ethernet:*	*2 pairs Cat. 5, UTP*

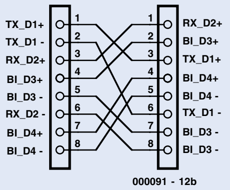
Pin allocation

	10BASE T2	*100BASE T4*
Pin 1	*TD+*	*TX_D1+*
Pin 2	*TD-*	*TX_D1-*
Pin 3	*RD+*	*RX_D2+*
Pin 4	*nc*	*BI_D3+*
Pin 5	*nc*	*BI_D3-*
Pin 6	*RD-*	*RX_D2-*
Pin 7	*nc*	*BI_D4+*
Pin 8	*nc*	*BI_D4-*

10BASE T



100BASE T



COMPONENTS LIST

Resistors:

R1 = 47kΩ
R2,R4-R11 = 1kΩ
R3 = 6kΩ8
P1 = 1k preset H

Capacitors:

C1 = 1µF 16V radial
C2,C3,C4 = 100nF
C5 = 10µF 16V radial
C6 = 100µF 16V radial

Semiconductors:

D1,D4-D11 = LED, red, high efficiency
D2,D3,D12-D18 = 1N4148
D19 = 1N4001
T1 = BC547B
IC1 = 555C
IC2 = 74HC193
IC3 = 74HC138
IC4 = 78L05

Miscellaneous:

K1,K2 = 8-way RJ45 socket, PCB mount (Farnell # 257-102)
2 Solder pins
Plastic case with battery compartment, size approx. 101x60x26mm (Conrad Electronics # 522864)

Modern Surround-Sound Systems

Dolby Digital vs DTS

By Harry Baggen

With the advent of the DVD (Digital Versatile Disc) it has become possible to store more than two good-quality sound channels with onto a user-friendly medium. This in turn has given an impetus to surround-sound reproduction in the average living room. This article takes a look at the facilities of the two currently most important surround-sound systems: Dolby Digital and DTS.

[Photograph reproduced courtesy of B&W loudspeakers]

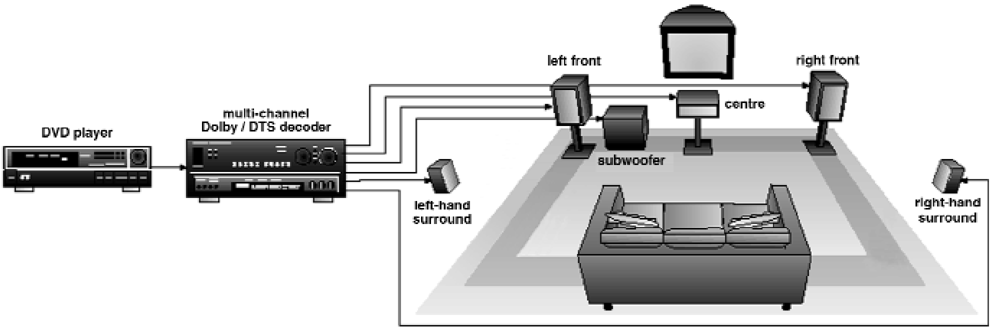


Figure 1. Setup of a domestic surround-sound installation consisting of five loudspeakers and a separate subwoofer.

Since stereo reproduction can provide a realistic spread of sound over an arc of only about 60 degrees in the horizontal plane, full-scale, three-dimensional surround sound became popular in the 1970s in the cinema. This popularity led to its adoption in television broadcasts, video tapes and compact discs (CDs). Microphone techniques are basically the same as those used for stereophonic reproduction by the use of quadraphonic panpots to control music and other sounds.

Although surround sound was adopted by virtually all cinemas, it was not until the 1990s that it was introduced into domestic living rooms. Before that time, quadraphonic sound systems employing four independent channels of sound information for greater realism became available, but they did not enjoy much success owing to their complex design (gramophone records had to be played back with quadra-styli), the lack of source material, and consumer apathy. Good-quality stereophonic systems were expensive enough for most who did not fancy the idea of having to buy two additional loudspeakers, a decoder, and an extra stereo power amplifier.

Nevertheless, surround sound reproduction adds an extra dimension to experiencing sound. This is true not only in the cinema, but also for recorded music where a surround sound system makes you feel as if you are at the actual performance.

Thanks to modern recording techniques, large memories (CD/DVD), and low prices of stereo equipment and loudspeakers, the public at large has become seriously interested in surround sound at home. Moreover, the DVD has proved an ideal source for multi-channel sound reproduction. This is because the DVD has sufficient storage for a number of sound channels and coding systems alongside the video signal.

Currently, there are two systems that have become pre-eminent in surround sound and both originate in the world of cinema. The better known of these is Dolby Digital, although the lesser known, DTS, provides better quality. Most other systems, such as MPEG, have virtually disappeared from the market owing to lack of support.

Hardware

A modern surround-sound system uses five main channels and a separate subwoofer channel: known as 5.1 audio – see **Figure 1**. It consists of two main loudspeakers (left/right front speakers), a centre-channel speaker, two rear speakers (left/right surround speakers), and a (optional) subwoofer.

The left/right front speakers should ideally be placed at either side of the television set.

The centre-channel speaker is perhaps the most important speaker since it not only improves the intelligibility of the dialogue, but also pinpoints the entire acoustic presentation to the sound source (television screen).

The surround speakers provide the spatial acoustics.

The subwoofer is optional, but of interest to those who wish to reproduce subsonic sounds with films.

The set of loudspeakers could be started in many cases with the two stereo speakers of the existing audio system, to which three

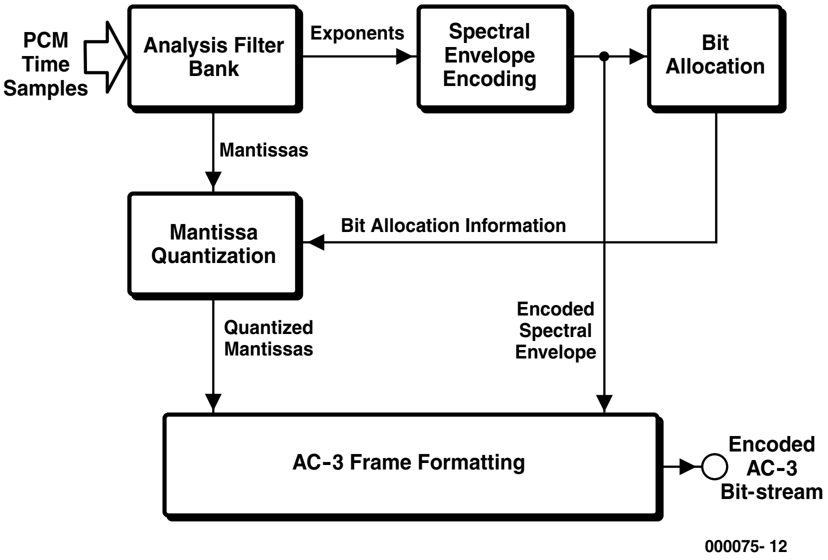


Figure 2. Block diagram of a Dolby Digital (AC-3) encoder.

small speakers (centre and surround), and a subwoofer, if desired, are added. It is equally possible to start with adding just the two surround speakers and leave the centre-channel speaker till later. Modern sound systems are very flexible.

Other equipment for setting up a domestic surround-sound system (home theatre) includes a television set, and a surround-sound amplifier or receiver with integral decoder for at least Dolby Digital. A suitable receiver – with five output stages – is available nowadays from about £200 (in the UK). Another item is, of course, a DVD player (£180–£400) or DVD recorder (currently imported from Japan at about £1500, but should become available in the USA and UK by the end of this year or early next year at a similar price).

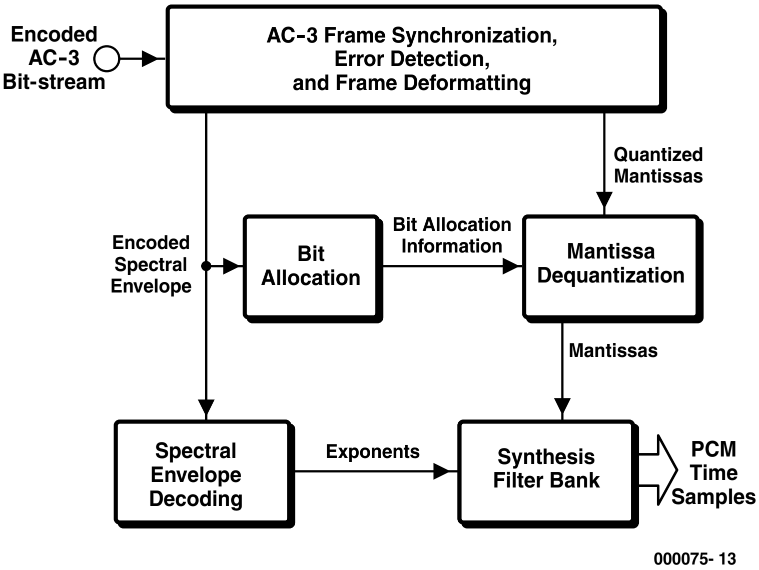


Figure 3. Block diagram of a Dolby Digital (AC-3) decoder which contains all the logic to reconvert encoded data.

Compression

The storing of five or more complete sound channels together with the video data would take too much space, in spite of the large capacity available on modern discs. Therefore, the sound channels are compressed, just like the video data. It must not be thought that compression means a deterioration of the sound quality – modern compression techniques make this virtually indiscernible. Moreover, compression is particularly suitable in a multi-channel system since there is normally more redundancy available (bits that are conveyed, but carry no additional information – they are used to detect or correct errors that may have been introduced into the channel). For instance, non-compressed storing of six channels with 16-bit resolution and 48 kHz sampling rate would require a data stream of 5.184 Mbit/s.

Dolby digital

Dolby Digital uses an algorithm that has been standardized by the United States Advanced Television Systems Committee – ATSC – under the title AC-3. This compression algorithm can compress up to 5.1 audio channels (five full-range channels plus a subwoofer channel) into a data stream of 32–640 kbit/s, depending on the wanted quality.

The block schematic of a typical AC-3 encoder is shown in **Figure 2**. The input signals to it are pulse code modulated (PCM) data. In the first instance a series of PCM signals is converted to the frequency domain by an analysis filter bank. Overlapping blocks of 512 time samples are multiplied in a time window and analysed into a frequency domain.

Because of the overlapping, each PCM input sample is present in two consecutive frequency blocks. Subsequently this representation is halved in the frequency domain so that 256 frequency coefficients remain. The individual coefficients are stored in an exponential notation with a binary exponent and mantissa. The set of exponents is decoded to obtain a rough reproduction of the signal spectrum – the spectral envelope. This envelope is used by a bit allocation to determine how many bits are needed for encoding each separate mantissa. The spectral envelope and the coarsely quantized mantissa for six audioblocks (that is, 1536 audio samples)

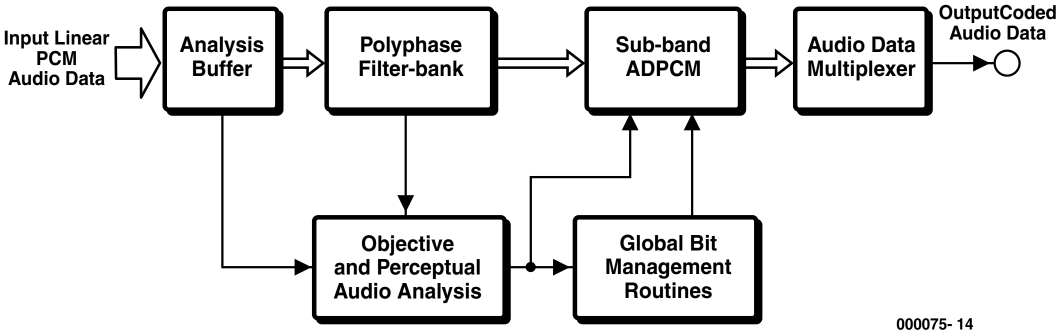


Figure 4. Block diagram of a DTS encoder using the Coherent Acoustics algorithm.

are then stored in a single AC-3 frame. The overall AC-3 bit stream consists of a chain of such frames.

A number of facilities are not included in the block diagram. For instance, there is a frame header containing all necessary data, such as bit rate, sampling rate, number of channels, and others.

Also, there are several ways of encoding. For example, the spectral resolution of the analysis filter bank may be altered dynamically; the spectral envelope may be encoded with a variable time/frequency resolution; and channels may be combined at high frequencies to attain high compression at low bit rates.

Decoding is basically the reverse of encoding. The decoder in **Figure 3** must first be synchronized with the encoded bit stream, and then check for possible errors before it can start to unravel the various kinds of data**,** such as the encoded spectral envelope and the quantized mantissas. The bit allocation routine provides the results necessary to analyse and dequantize the mantissas. The spectral envelope is decoded to determine the exponents. Finally, the exponents and mantissas are reconverted to the time domain.

Figure 3 is not entirely complete: it does not show the blocks for error correction, eventual decoupling of channels, and dynamic adaptation of the filter bank**.**

DTS digital surround

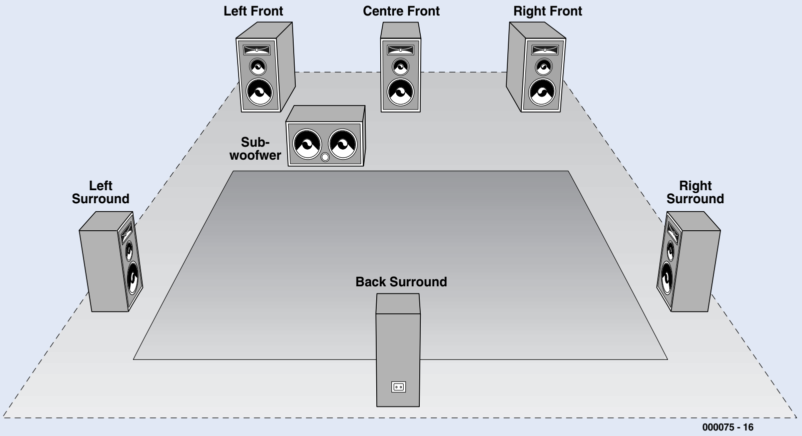
Although Dolby Digital is the most widely used, professionals are agreed that DTS Digital Sound gives better sound quality. The block schematic of the DTS encoder in **Figure 4** is quite different from that in Figure 2, but such diagrams do not show where differences in sound quality may be seated. The only tangible aspect is that DTS uses a larger data stream than Dolby Digital.

DTS uses the Coherent Acoustics compression algorithm. In Figure 4 the input signal is split into a number of sub-bands by a polyphase filter. This provides short-term information about the spectral division of the signals and enables audible redundancies to be eliminated. Each sub-band signal continues to contain linear PCM audio data, but only with limited bandwidth. The bandwidth and number of sub-bands depend on the bandwidth of the input signal, but usually the audio spectrum is split into 32 uniform sub bands.

Within each sub-band differential encoding (sub-band ADPCM) takes place, and at the same time psycho-acoustic and transient analyses are carried out on the signals in the sub-band to detect discernible but unnecessary data. Depending on the bit rate, the results are used to adapt the master routine to the differential encoding. The combination of this differential encoding and the psycho-acoustic adaptation is very efficient.

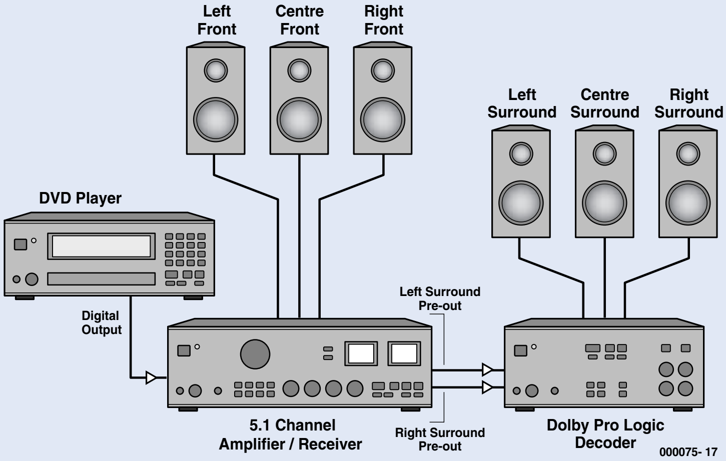
It is noteworthy that with increasing bit rates the psycho-acoustic adaptation is

Dolby Digital Surround EX



Recently a new standard has been introduced that is already available on the more expensive domestic equipment: Dolby EX (also called Dolby 6.1). In this, an additional channel is added to the existing five. This is a third rear channel centred between the existing two: a centre-surround. This pinpoints the location of the various sounds even more precisely. The system is already in use in many cinemas. In practice, it appears that it is fairly simple to add a sixth channel to a domestic system without the need of a new decoder. Needed are a Dolby Pro Logic decoder and an additional output amplifier. The outputs of the two surround-sound channels are applied to the Pro Logic decoder which provides two new rear channels and the additional centre surround channel. Several magazines and surround-sound enthusiasts have tried this setup and all claim that it works very well, indeed.

Competitors are not left behind, however, so DTS have introduced DTS-ES (Extended Surround) that also offers a centre rear channel.



THX and THX Ultra

The THX system from Lucas Films (makers of Star Wars) is not a discrete surround-sound system, but an extension of existing systems combined with a standard of quality that dictates which requirements a surround-sound system and associated software must meet to qualify for displaying the THX symbol. There are two groups: THX Select and THX Ultra. The main difference between these lies in the volume of the listening space: THX Select is for living rooms of 70 m^3 or less, whereas THX Ultra is for rooms of greater volume.

The hardware must meet the stringent requirements proscribed by Lucas Films. For instance, amplifiers are judged on power and distortion, and loudspeakers on their pattern of directivity, frequency response and minimum sound pressure level. The rear speakers should be dipole types.

The THX processor uses a Dolby Pro Logic or Dolby Digital signal and adapts this to the listening environment. A peculiarity here is the decorelation of the rear channels to prevent localization of the sound sources. The processor also provides timbre matching which ensures proper matching of sound between the front and rear channels so that sound shifts between front and rear channels occur more smoothly.

Manufacturers must submit their equipment for test and inspection and only when this is successful is THX approval certification granted. The tests include even a check on the accuracy of the test tones.

reduced gradually so that the signal quality rises with the bit rate.

The bit management routines ensure the allocation and division of the bits over all encoded sub-bands of the audio channels.

The final stage of the encoder consists of a data multiplexer that combines all encoded data of the sub-bands of all channels, any additional data, and synchronization bits into a single data stream.

In the decoder in **Figure 5**, after synchronization the encoded bit stream is unraveled and any errors in the data detected and corrected. After that, the data is demultiplexed to the separate sub-bands of each audio channel.

The differential sub-band signals are inverse-digitized to sub-band PCM signals in the second stage according to the instructions transmitted with each sub-band. The signals so obtained are inverse-filtered to restore the signal in the time domain for the entire frequency spectrum.

An important advantage of the DTS system is the fact that all intelligence is contained in the encoder, which enables a quasi-passive design of the decoder. All decoding information is contained in the encoded bit stream, so that the encoding algorithm can be altered without the decoder having to be adapted.

Various possibilities

Both systems offer the user an number of possibilities to adapt the signals to personal preferences.

It is, for instance, possible to indicate how many and which loudspeakers are connected. This information enables the processor in the decoder to determine which signals must be sent to which channels.

Also, the distances between the loudspeakers and the listener, and the location of the rear speakers, may be input, so that the processor can take account of the delay times.

The frequency division may be adapted to circumstances. For instance, it may be input that the rear speakers are small, so that low frequencies for these speakers are diverted to te front speakers. The same is true for the subwoofer. If this is not used, the information for it is diverted to the other loudspeakers.

More channels?

Modern surround-sound systems are not just technical advances, but they add a sense of realism to the experience of sound in the domestic living room. Unfortunately, as so often when the consumer has accepted a new system, the manufacturers are inclined to go on adding facilities. Currently, there is a tendency to add even more channels, although most consumers are happy enough to be able to add two, or at most three, additional loudspeakers in their living room. Where will it all end? Fortunately, most systems have a downmatrixing mode with which the number of channels can be restricted or reduced to what is wanted

[000075]

More information may be found at the following web sites.
Dolby: *http://www.dolby.com*
DTS: *http://dtstech.com*
THX: *http://www.thx.com*

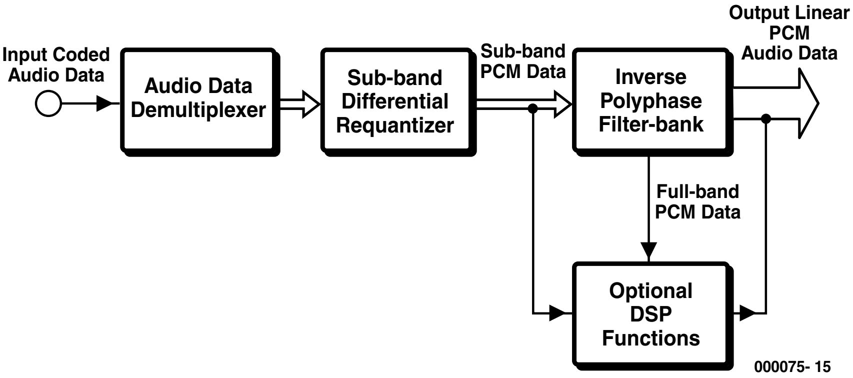


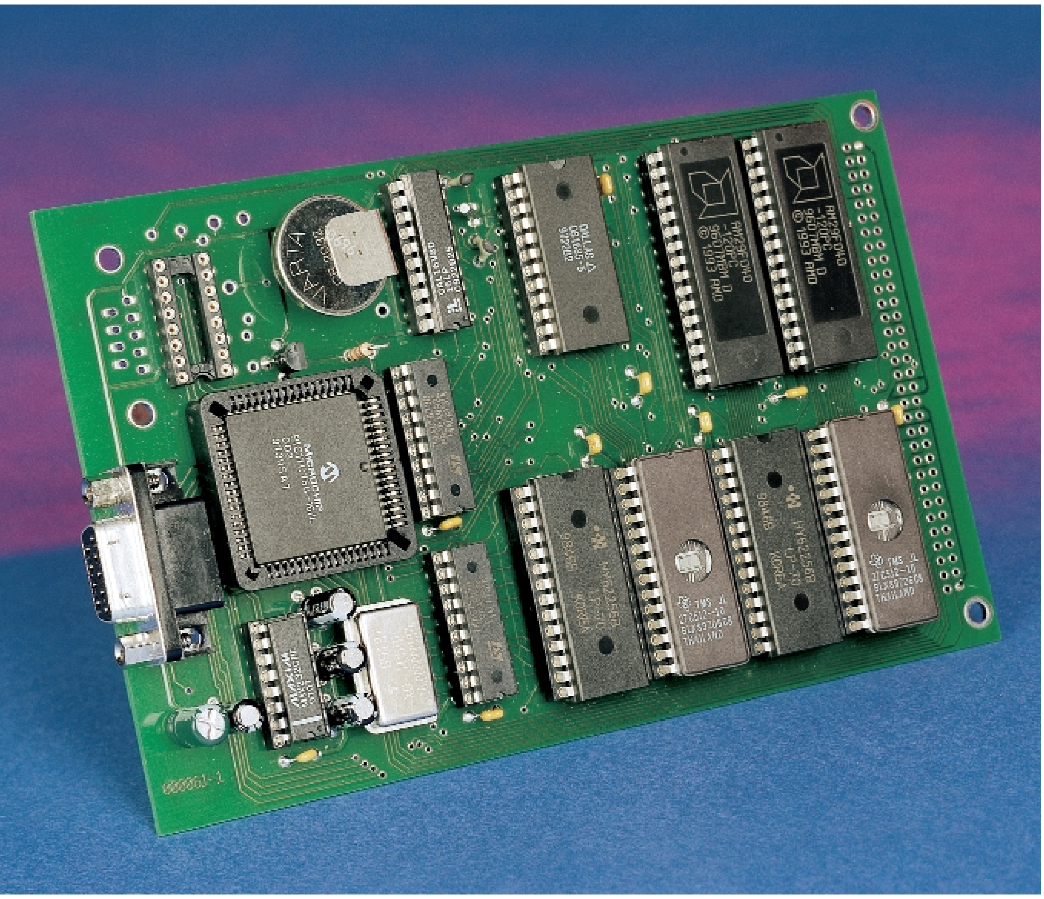
Figure 5. Block diagram of the DTS decoder, which is a quasi-passive type. The information on the manner of decoding is conveyed with the signal data.

PIC17C Processor Board

Single-Board Computer based on the PIC17C756

Design by Prof. F. P. Volpe and B. Wegmann

The PIC17C computer described in this article is a flexible and easily expandable system which is comfortably programmed by a PC rather than a costly, dedicated programmer. At the heart of the system is the extremely fast and powerful PIC17C756 microcontroller from Microchip Technology.



With microcontroller systems featuring 'in-system' programming, the monitor program is usually stored in an external EPROM. This requires that the controller is capable of disabling its complete internal program memory, and fetch all instructions from an external memory. Furthermore, the external memory structure should be highly similar, if not equal, to that of the internal memory (in this case, having a width of 16 bits).

Core and peripherals

At the heart of the PIC17 SBC (single-board computer), whose circuit diagram shown in **Figure 1** is typical of a classic microcontroller system, is a microcontroller type PIC17C756 from Microchip Technology. This controller features 58 single-word instructions which are all executed in one cycle, with the exception of program jumps and table access operations. The PIC17C756 operates from a supply voltage between 3.0 V and 6.0 V, and may be clocked at any frequency between DC and 33 MHz.

Main Features

- PIC17C756 CPU
- EPROM-resident System Monitor
- User programs loaded into Flash memory by way of System Monitor
- SRAM
- Real-time clock with battery backup
- Eurocard with DIN-64AC expansion connector
- 100% compatible with Microchip MPLAB environment
- Flexible and user modifiable address decoding using GAL

The clock is internally divided by four, so that one machine cycle takes a minimum of 121 ns. This specification, by the way, calls for memory devices with access times below 120 ns. The microcontroller core has available an 8?8 hardware multiply unit, 18 interrupt sources and a 16-level hardware stack. The 17C756 has a data memory size of 902?8 bits, and an internal program memory of 16k?16 bits.

Use of the program memory is not obligatory, however, because the controller is capable of working in one of three modes, of which the different memory arrangements are illustrated in **Figure 2**.

Microcontroller mode: the entire program memory is located on the chip.

Expanded Microcontroller mode: the program memory between 0000h and 3FFFFh is located on the chip, while the memory range from 4000h through FFFFh is externally addressed. In this system, PortC and PortD act as multiplexed data and address lines. Demultiplexing takes place using the ALE signal and two D-type latches. Note that the address and data bus are both 16 bits wide.

Microprocessor mode: the complete program memory is external, and a memory range of up to 64 kBytes may be addressed. The on-chip program memory is not used. Access to the external memory device(s) is the same as in extended microcontroller mode, however, the memory is addressed from 0000h onwards. Access to configuration bits, memory test and the boot ROM is not possible in this mode.

On the PIC17C SBC described here, the PIC17C756 operates in Microprocessor mode. When the chip itself is not programmed, it is also in this mode.

Housed in a PLCC-68 case, the PIC17C756 offers a large number of peripheral elements. A total of 50 I/O lines can be addressed individually. Each line can supply or sink 25 mA – lines RA and RB even up to 60 mA (at 12 V), which is useful for direct driving of LEDs. Four pins may be used as capture inputs with a resolution of one machine cycle (121 ns), and three pins as PWM outputs with a resolution of 1-10 bits. The 17C756 features a 16-bit timer/counter with a programmable prescaler, two 8-bit and one 16-bit timers/counters. An independent baudrate generator clocks the communication via the two universal synchronous interfaces (USART/SCI). There is also a synchronous serial port (SSP) supporting various modes (SPI, I^2C and I^2C master). An analogue/digital converter with 12 channels and a resolution of 10 bits allows analogue quantities to be processed directly.

The 17C756 also offers a power-up timer, an oscillator start-up timer and a power-on reset circuit, brown-out reset and software protection. It can be operated in low-power mode as well as with various oscillator options. It should be noted that the watchdog timer (WDT) is on by default. This function can only be switched off with a programmer, and not, unfortunately, by software. When programming, you should always be in control of the WDT, resetting it every now and then to prevent an unwanted reset to be generated. At first blush this may seem tedious, however the conditions as regards WDT maintenance pay dividends for solid programming structures and program stability.

The microcontroller IC has two small ‘satellite’ ICs, an oscillator chip, IC15, and a microcontroller supervisor, IC14, which serves to reset the PIC when the supply voltage drops below a certain level. The push-button connected to the master reset line MCLR allows the SBC to be reset manually. The MPC120/130 microcontroller supervisor chip was discussed in the June 2000 issue of *Elektor Electronics*. The MCP130, by the way, has an internal pull-up resistor, allowing R1 to be omitted.

Memory map

The general structure of the memory is illustrated in **Figure 3**. In microprocessor mode, the entire program memory from 0000h to FFFFh is located ‘off chip’. Figure 3 provides information about the way the external memory is mapped. The monitor EPROM is located in the range 0000h to 1FFFh, occupying 8 Kbytes. Two 27C512s are used (IC9 and IC13) to provide the high and low bytes. The high-byte EPROM is labelled **000061-21b**, the low-byte EPROM, **000061-21a**. Do not mix them up!

In the EPROM, the interrupt vectors are located at addresses 0008h, 0010h, 0018h and 0020h. To be able to use these interrupt vectors in a user program running from Flash memory, these EPROM addresses jump to Flash locations 9F08h, 9F10h, 9F18h and 9F20h.

The user program is held in Flash memory devices type AM29F010 or AM29F040. Here, too, we are using two 8-bit devices in parallel. The user program is given a total size of 32 Kbytes in the address range 2000h to 9FFFh. This size should be adequate for most, if not all applications. If you require more user memory, you will have to reprogram the system GAL. Flash memories type 29F040 have a capacity of 4 Mbit, allowing for sufficient spare capacity. The same goes for the data memory — two SRAMs type 62256 (IC6/IC7) — which has been allocated the (fairly random) size of 12 Kbytes in the range A000h to CFFFh. Although there is spare memory available amounting to 20 Kbytes, you should note that the total address range is limited to 64 Kbytes.

All address decoding is handled by a single component, a GAL type 16V8D. Its content is listed in **Figure 4**. The GAL decodes four signals, Extern1 through Extern4, in the address range D000h through EFFFh. Each of these occupies 2 Kbytes. The Extern1-4 signals are also brought out to pins on the 64-way connector, and may be used to address expan-

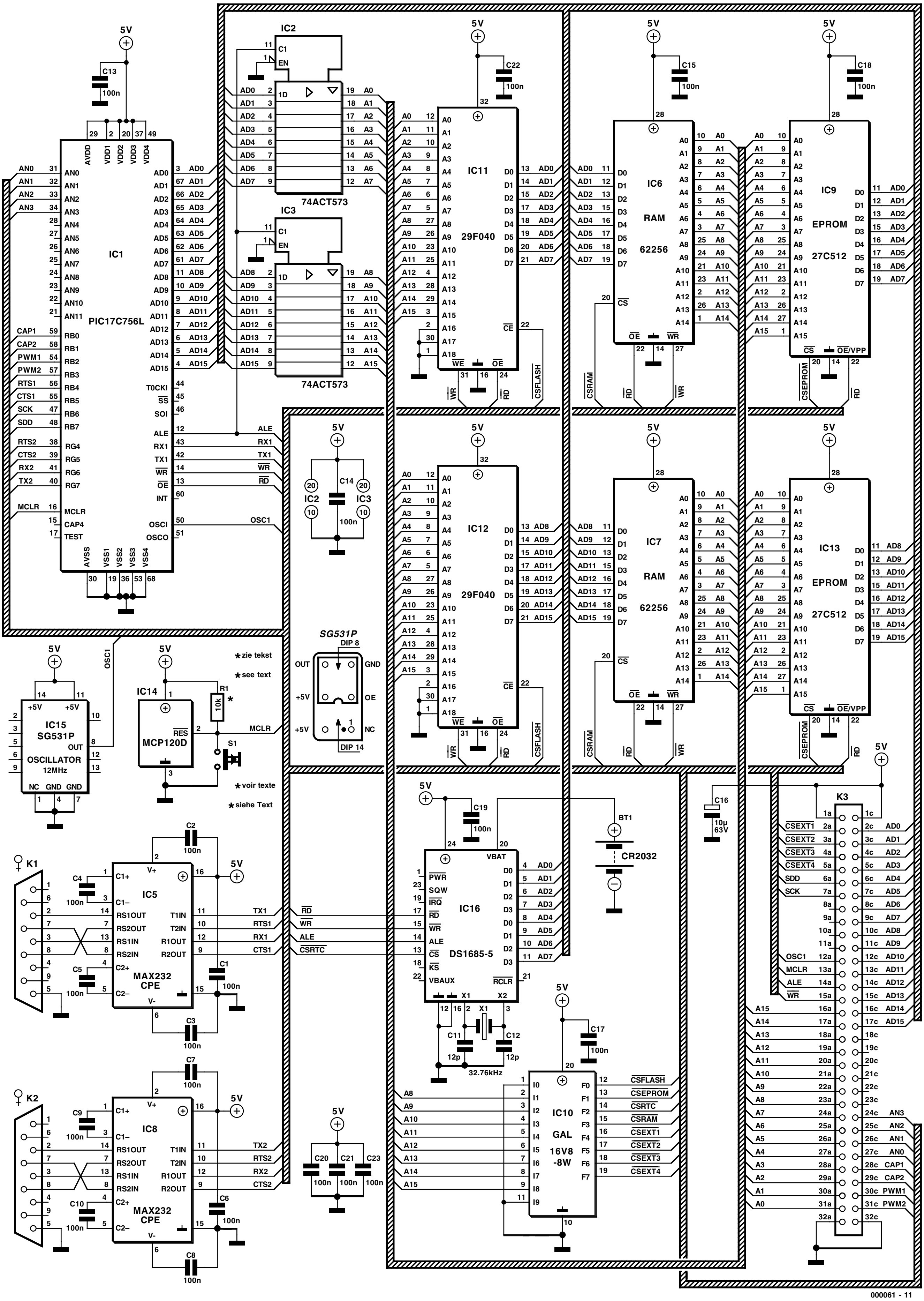


Figure 1. Circuit diagram of the PIC17C Processor Board.

sion cards. The author hopes to be able to describe such cards in future articles in *Elektor Electronics*. Finally, the address range F000h to FFFh is allocated to real-time clock (RTC) IC16, a DS1685-5. As an aside, capacitors C11 and C12 are not strictly required and may be omitted.

Communication

For communication, the PIC17C has two serial RS232 interfaces linked to the two USARTs in the microcontroller. System control and program downloading from the PC are handled by USART2. The PIC17 SBC employs the following communication parameters:

9600 bits/s
8 data bits
no parity
1 stop bit
local echo on
add line feed (LF) when receiving
carriage return (CR)

The first serial interface, USART1, is available for user applications.

Two MAX232 ICs are used to handle the function of level-converting the signals between the microcontroller and the PC serial port (see Inset). Once the program is loaded in Flash memory (via USART2), the SBC is able to work in stand-alone mode. At that point, USART2 is also enabled and available.

Construction

There is little to say about stuffing the board whose artwork is shown in **Figure 5**. We should, however, advise to use a low-power soldering iron with a fine tip. Make sure the soldering is tidy — no splashes or dry joints. All ICs should be fitted in high-quality IC sockets.

The PIC17C SBC board has a standard Eurocard size (10x16 cm), and should be easy to install in a 19-inch rack using a 1-HU fixing plate. If rack mounting is envisaged, the reset switch should be fitted on the front panel and wired to the circuit, rather than mounted on the board.

Development system

An excellent PIC17C development

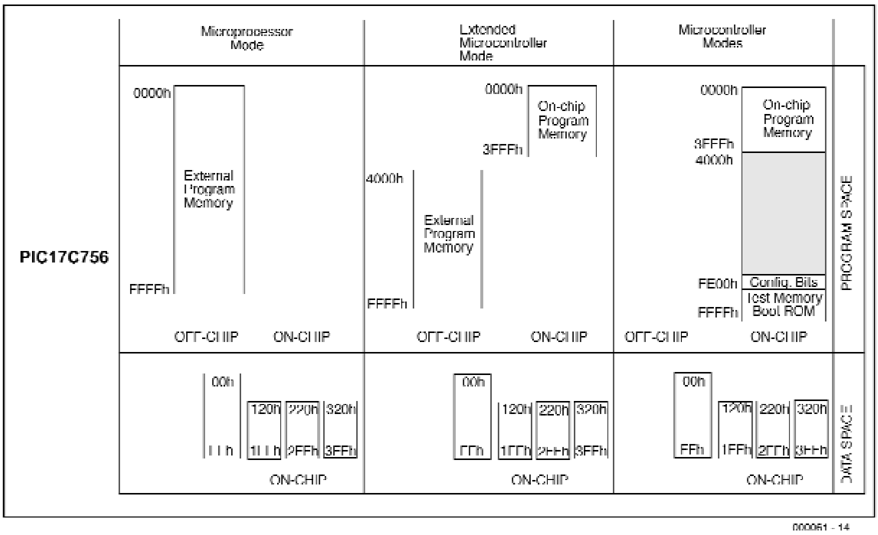


Figure 2. Memory maps for the different modes of operation.

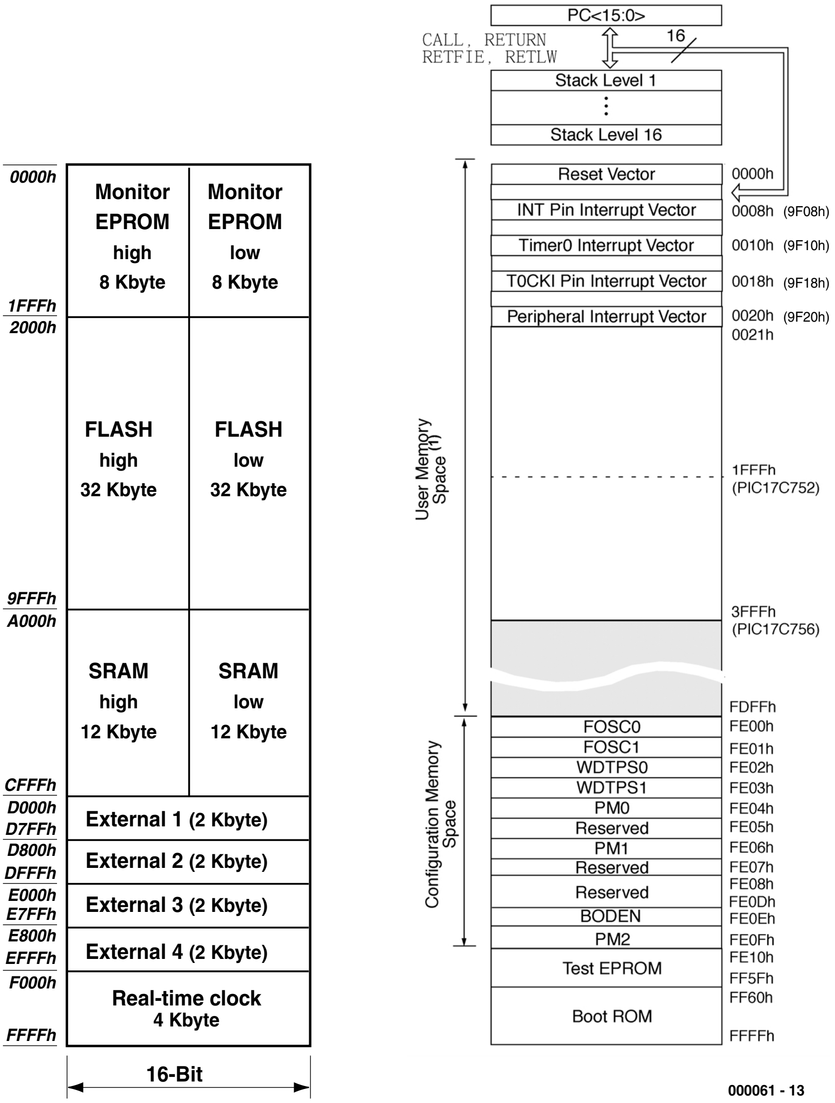


Figure 3. Memory allocation of the PIC17C756. The monitor starts at 1FFFh

```
 1:Name          Mon;
 2:Partno        IC10;
 3:Date          08/11/99;
 4:Rev           01;
 5:Designer      Bernd Wegmann;
 6:Company       FH-Aschaffenburg;
 7:Assembly      None;
 8:Location      None;
 9:Device        g16v8;
10:
11:/*******************************************/
12:/*                                         */
13:/*******************************************/
14:
15:/** Inputs **/
16:
17:Pin [2..9] = [A8..A15];     /* Adressen  */
18:
19:
20:/** Outputs **/
21:
22:Pin 12 = !Flash;  /* EN von Tastenmatrix */
23:Pin 13 = !Eprom;
24:Pin 14 = !Uhr;
25:Pin 15 = !SRam;
26:Pin 16 = !Ext1;
27:Pin 17 = !Ext2;
28:Pin 18 = !Ext3;
29:Pin 19 = !Ext4;
30:
31:/*  equations  */
32:
33:Flash = A14 & !A15 # A13 & !A15 # !A13 & !A14 & A15;
34:
35:Eprom = !A13 & !A14 & !A15;
36:
37:Uhr = A12 & A13 & A14 & A15;
38:
39:SRam = A13 & !A14 & A15 # !A12 & !A13 & A14 & A15;
40:
41:Ext1 = A15 & A14 & !A13 & A12 & !A11;
42:
43:Ext2 = A15 & A14 & !A13 & A12 & A11;
44:
45:Ext3 = A15 & A14 & A13 & !A12 & !A11;
46:
47:Ext4 = A15 & A14 & A13 & !A12 & A11;
48:
49:/* end of equations */
50:
51:
52:
53:
54:
55:

Jedec Fuse Checksum       (3373)
Jedec Transmit Checksum   (cffa)
```

Figure 4. JEDEC listing of GAL contents. This should be easy to modify for 4-Mbit memories.

system can be built if you use Microchip's MPLAB-IDE package, which is available free of charge from this Microchip website page:

http://www.microchip.com/10/Tools/PICmicro/DevEnv/MPLABi/Software/v50000/index.htm

This development environment includes all the necessary tools like an assembler, linker, editor and a simulator.

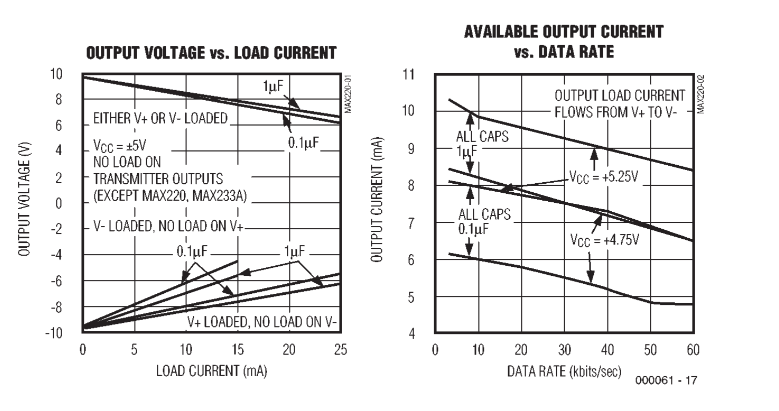
It is also possible to integrate a free C Compiler which, unfortunately, is a test version with a 30-days trial limit. A downloadable handbook plus a tutorial are also waiting for you.

Once you have written and tested a program using MPLAB-IDE, the next thing to do is load it into the SBC in the form of a hex file. That is comfortably handled by the monitor program running on the PIC17 SBC. The assembler will normally produce a HEX file for 16-bit EPROMs. Via the monitor, this format may be transferred to Flash memory. Using Split (S) format, two 8-bit wide files are produced, for the low and high byte.

The HEX file may then be sent to the Flash memory on the SBC from address 2000h onwards by using the *Send Textfile* option of any suitable Terminal program. However, it is

MAX232

The function of the interface level converter type MAX232 is to generate the symmetrical voltages on the RS232 port lines from a single 5-V supply voltage, and the other way around, turn the RS232 levels into signals with TTL swing. The on-chip step-up voltage converter employs two charge pump capacitors connected to the C1+/– and C2+/– pins of the chip. Normally, 1-μF tantalum capacitors are used but such a large value is not absolutely required when the current consumption is not too high. The diagrams show the relation between load current, output voltage and charge pump capacitance for the MAX232A. If you are suspicious about the small caps fitted on the board, feel free to replace them with 1-μF types to ensure that a data rate of 120 Kbits/s can be achieved. The PCB has sufficient space for larger capacitors.
Note that the MAX232E achieves a highest data speed of 200 Kbits/s using the same capacitors.



also possible to write the program to a different memory range, for example, directly into SRAM. The start address is always indicated by means of the ORG directive in the source code. Note, however, that the SBC command *G* only launches the program found at address 2000h, while in Stand-Alone mode it also expects the program to be available from 2000h onwards.

The Stand-Alone mode may be selected without changing a jumper or similar hardware-related activities. Simply reset the controller, and do not send anything by way of

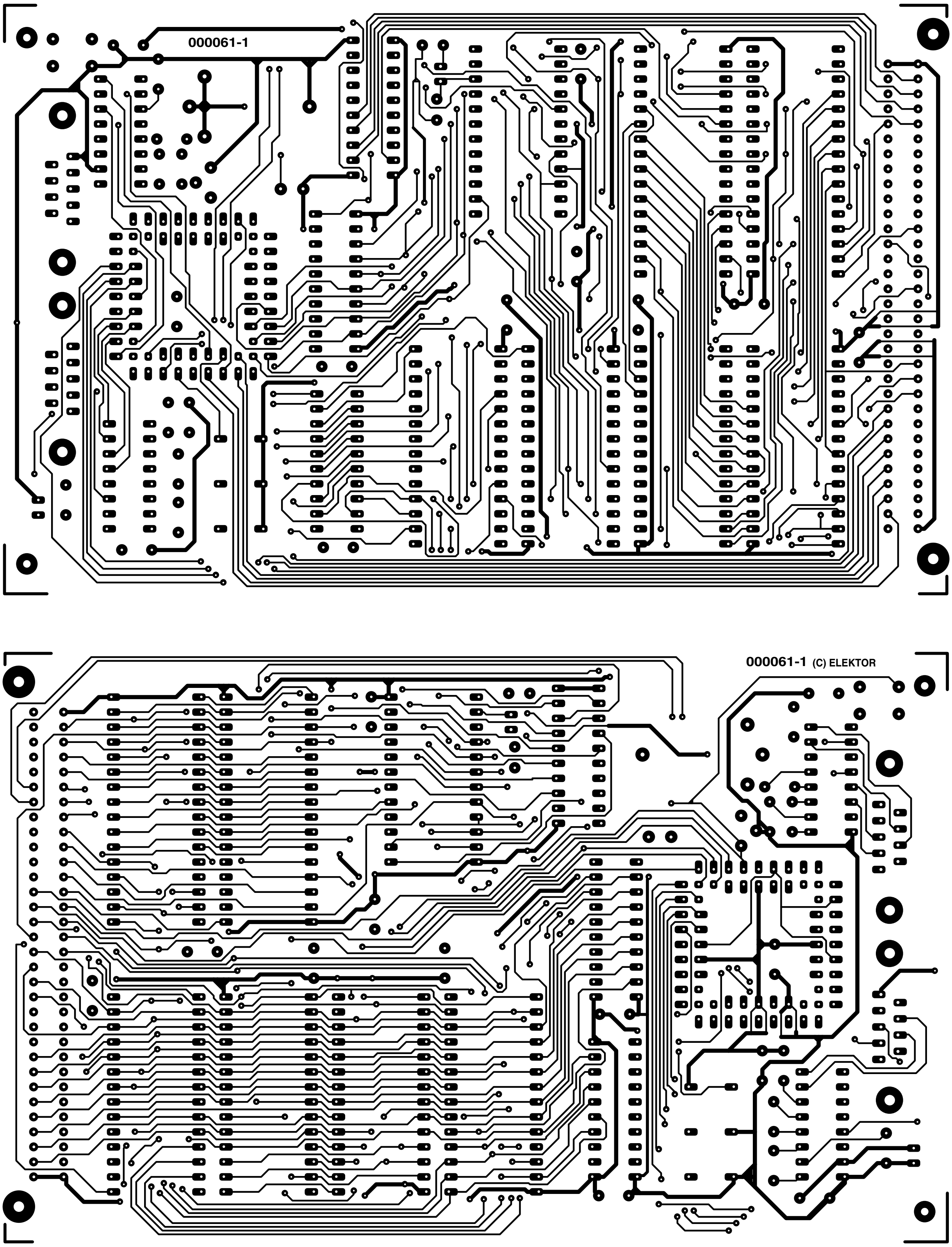
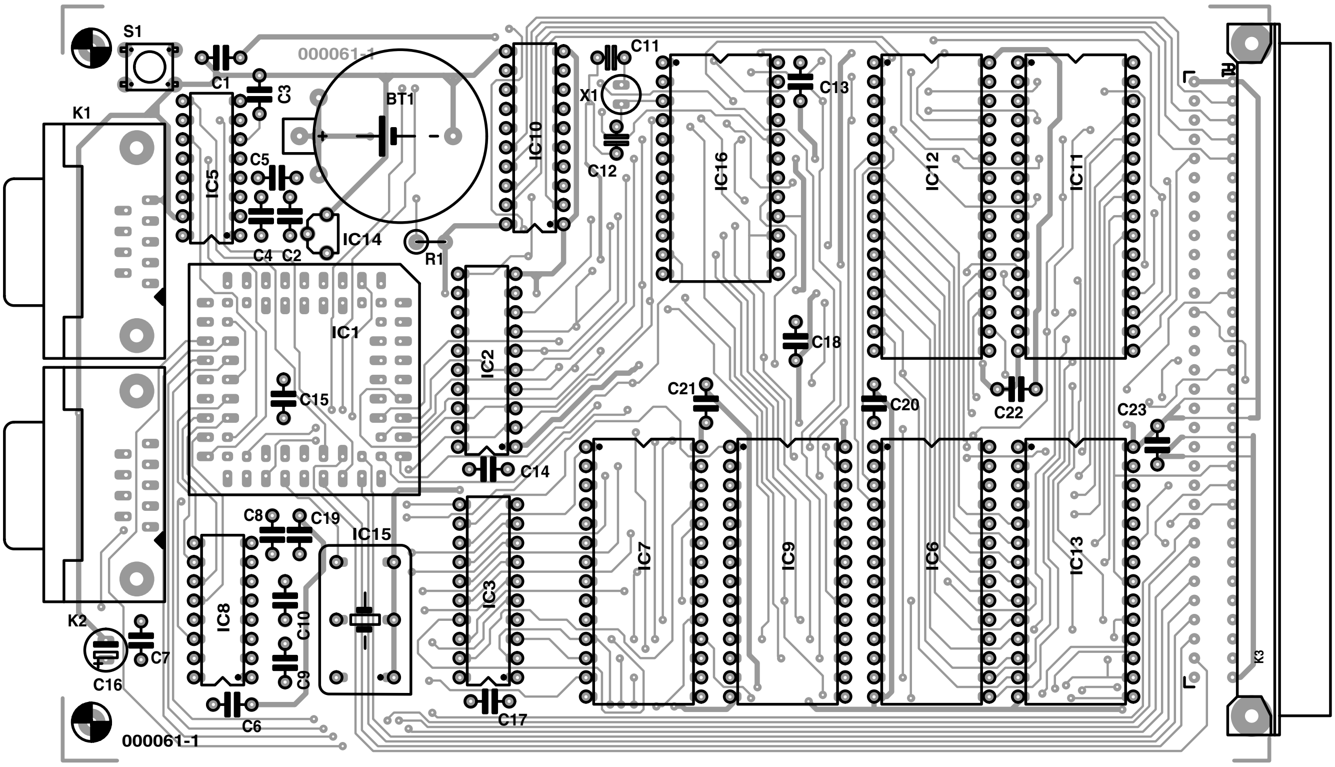


Figure 5. Double-sided Eurocard size PCB designed for the PIC Processor Board.



COMPONENTS LIST

Resistor:
R1 = 10kΩ

Capacitors:
C1-C10,C13,C14,C15,C17,C18,C19 = 100nF
C11,C12 = 12pF
C16 = 10μF 63V radial

Semiconductors:
IC1 = PIC17C756-16L (no suffix, or -A or -B)
IC2,IC3 = 74**A**CT573
IC5,IC8 = MAX232CPE
IC6,IC7 = 62256 (HM6225BLP-12), access time 120ns or less
IC9,IC13 = 27C512 (programmed), order code **000061-21**, access time 120ns or less (two EPROMs, a and b)
IC10 = 16V8-D (programmed), order code **000061-31**
IC11,IC12 = AM29F010 120PC or AM29F040 120PC
IC14 = MCP120D or MCP130 450 DITO
IC15 = 8-pin or 14-pin oscillator block 16 MHz (Seiko Epson 531P)
IC16 = DS1685-5 (Dallas)

Miscellaneous:
K1,K2 = 9-way sub-D socket (female), angled pins, PCB mount
K3 =64-way connector, angled pins, DIN64AC
S1 = Pushbutton, 1 make contact, Bourns type 7906H-001-300
X1 = 32.76kHz quartz crystal
BT1 = CR2032 Lithium cell (if necessary with holder)
68-pin PLCC socket
PCB, order code **000061-1**

USART2 for about 10 seconds. The SBC will then automatically start executing the program stored at 2000h.

The system monitor is pretty Spartan (as it should be), offering the following commands:

D: download
S: set time
T display time
Z: set date, Z DD.MM.YY
A: show date
B: read block, B AAAA AAAA
W: write word, W AAAA DDDD
R: read word, R AAAA
G: run program in Flash
E: erase Flash memory
H: help, commands or syntax

With the PIC17C SBC powered up and connected to the PC, the monitor will report with the *Boot ok* message, which is repeated every 12 seconds.

(000061-1)

PIC17C SBC Product Availability

000061-21	set of two programmed EPROMs (a/b)
000061-31	GAL 16V8D for address decoding
000061-1	Printed circuit board (w/o components)

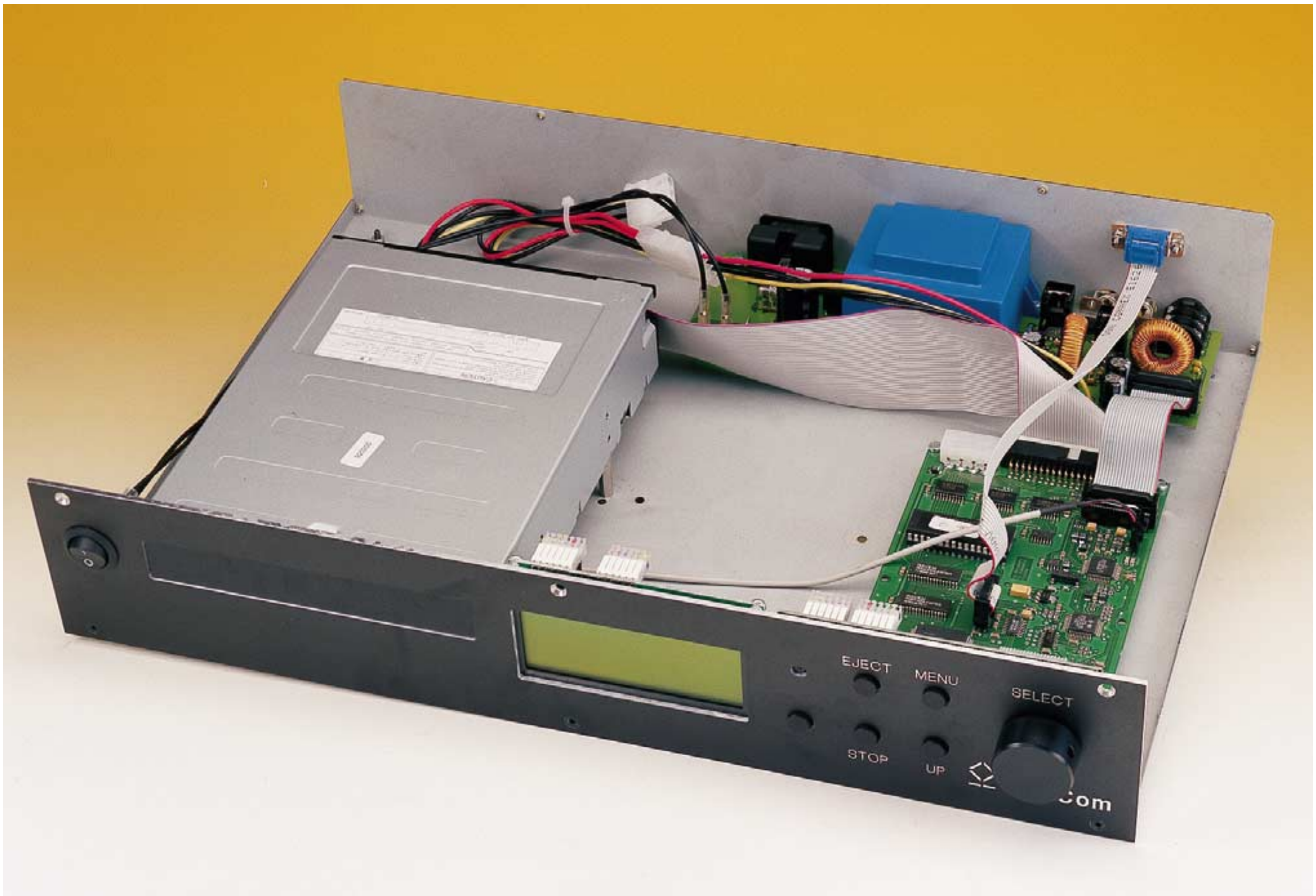
These items are available through the Publishers' Readers Services.
The project software (all source code files) and the PCB design (pdf) may also be downloaded from our website at http://www.elektor-electronics.co.uk

'OSCAR' MP3 Player (1)

A stand-alone MP3 player

Design by Engineering Bureau A. Kurpiers and V. Pantelic

OSCAR is a high-quality MP3 player for upgrading a conventional audio installation. In addition to the CD drive for data storage, you can use an IDE hard-disk drive, which will provide a week's worth of music in hi-fi quality.



It all seemed so nice: the first instalment of the project for a DIY stand-alone MP3 player was supposed to appear in the June issue of *Elektor Electronics*, following an article on the basic principles of MP3 in the May issue, and the series was to culminate in the September issue. But plans are one thing, and reality another. Unforeseeable technical problems cropped up with the MP3 Cube, and in spite of the collective expertise of the *Elektor Electronics* design laboratory, these could not be solved in a reasonable amount of time.

Fortunately, we have found a (better) alternative in the form of an MP3 player that goes by the name of OSCAR (for 'Optical Storage Compressed Audio Replay'), which was developed by staff members at the Technical University of Darmstadt in Germany. You can read a bit about the interesting story of the development of this device in the 'History' inset. OSCAR uses the same chip set as the MP3-Cube, but it is controlled by a much more powerful processor. This is also necessary, since the features of OSCAR surpass those of the MP3-Cube by far.

The most important differences are the possibility of connecting an IDE hard disk drive, direct selection of individual titles using a rotary encoder, and compatibility with MP2 and S/PDIF. The Features sidebar gives a good summary of the most important characteristics of the MP3 player.

MP3 Player Features

- Complete, PC-independent MPEG audio player
- Reads MPEG1/2 Layer 2/3 Audio (MP2, MP3) and CD audio from CD-ROM and hard disk
- Plays all MP2 and MP3 formats at all bit rates between 8 kb/s and 320 kb/s, plus VBR
- Line and headphone outputs
- Supports ISO 9660, including the Joliet extension, as well as FAT16/FAT32 read and write access
- Four-line LCD displays directory and file names and ID3 tags
- Rotary encoder allows quick access to the entire music collection
- High-performance 32-bit RISC processor (Hitachi SH7032)
- Sophisticated operating system with many functions relating to MP3
- Flash BIOS allows software updates
- Serial interface for PC communication and remote control
- Circuit board in Eurocard format; supply voltages 5 V and 12 V

Star performer

OSCAR is a PC-independent playback device for music files in the popular MP3 format, which can come from a purchased or homemade CD-ROM in ISO 9660 file format, or equally well from a Windows file system with FAT16 or FAT32 formatting. In addition, it can play back normal audio CDs according to the S/PDIF standard.

In contrast to a conventional CD player, OSCAR offers a multitude of additional options. For example, the artists and titles of the individual songs are displayed in plain text, and ID3 tags that may be present are also displayed. You can play back individual tracks, subdirectories or an entire CD or hard disk, and you can select sequential, repeat or random playback. Title sequences can be read from m3u playlists (Winamp) or programmed into the device, and they can also be stored on the hard disk if one is present.

The hard disk can either be prerecorded in a PC and then installed in the player, or titles (or even complete directories) can be copied from the CD-ROM drive to the hard disk. The hard disk can be formatted in the player, and directories can be created and deleted. The sleep timer can be used to switch off the device after a certain amount of time. 'Autoplay' starts the playback immediately after the player is switched on.

A combined rotary encoder and pushbutton switch, along with five other pushbuttons, make the equipment easy to operate. Titles can be selected by rotating the wheel, and then started by a simple button push. An uncomplicated menu structure allows all operating modes to be set up and stored. The operating software of OSCAR is being continuously refined and extended, and it can be updated at any time via an update CD or the built-in serial interface.

Thanks to the articles about the theory of MP3 and the various chip sets for decoding MP3 files, we can limit ourselves here to the electronics. The block diagram in **Figure 1** shows that OSCAR is based on the Micronas Intermetall MAS3507D/DAC3550 chip set. The controller addresses an external RAM as well as flash EEPROM, and it controls a series of peripheral functions, such as the operator controls, display and interfaces. The two types of drives, CD-ROM and hard disk, are also supported.

The key figure: a Hitachi RISC controller

In light of the large number of tasks it must perform, the MP3 player is controlled by a very powerful 32-bit RISC controller (SH7032) clocked at 20 MHz (**Figure 2A**). This con-

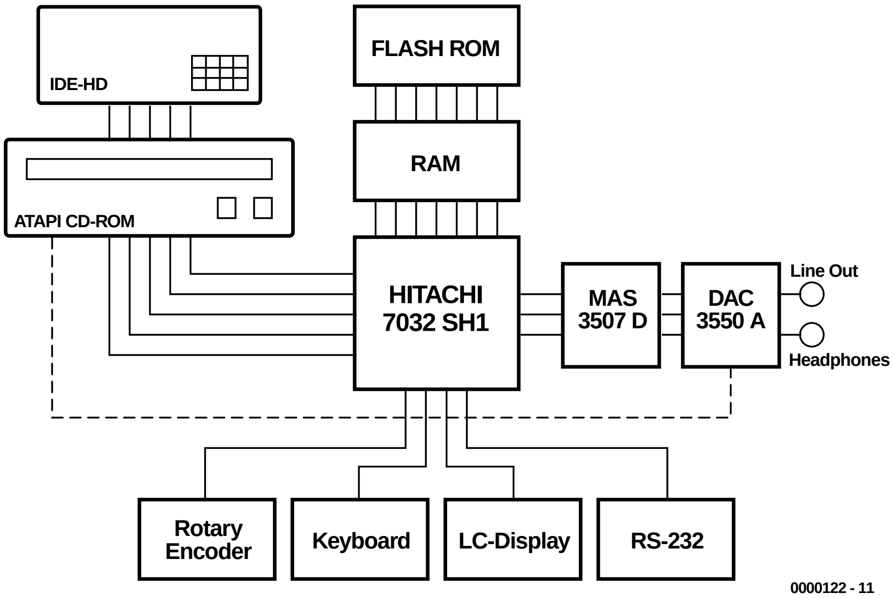
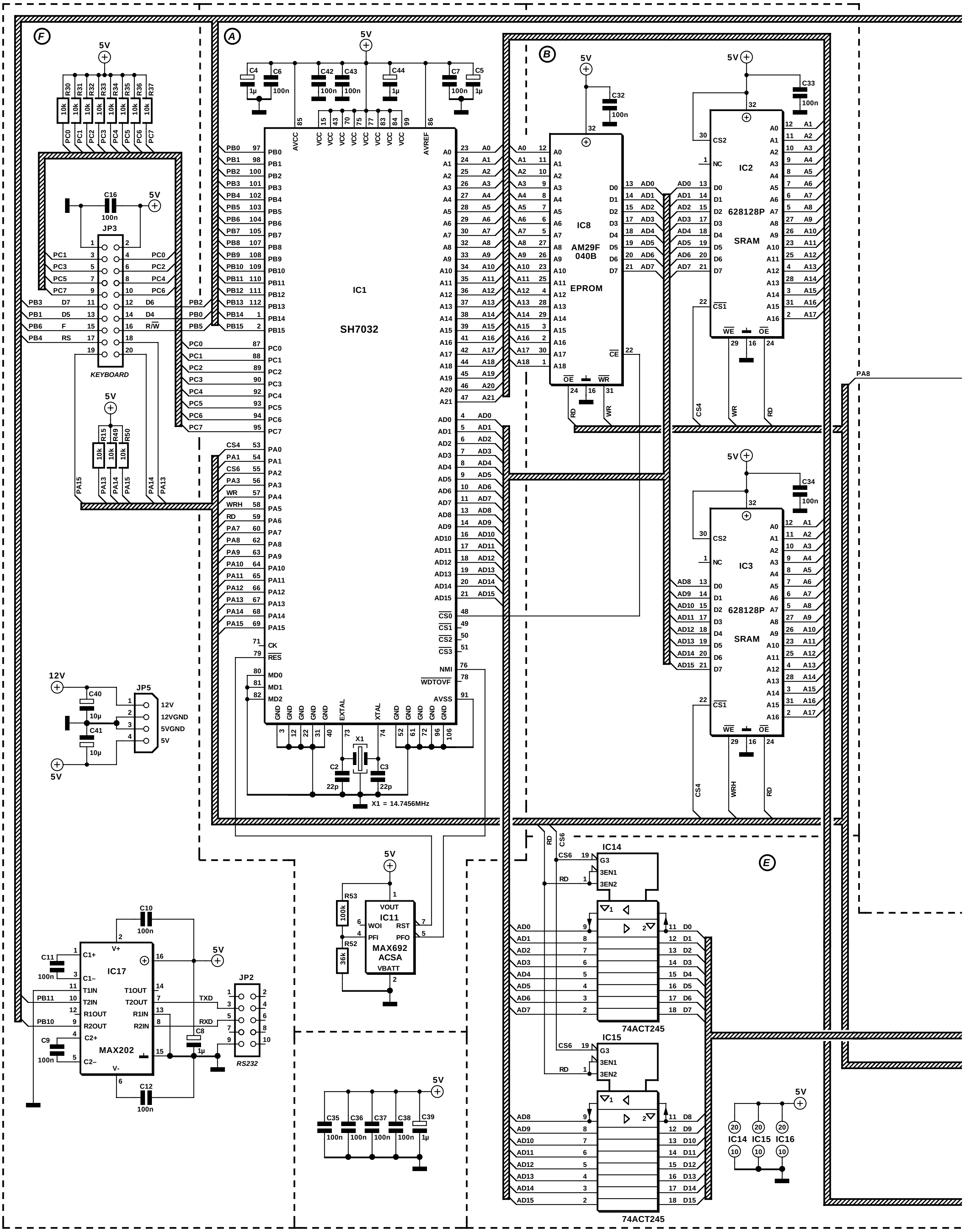


Figure 1. Block diagram of the MP3 player.


F
A
B
E
5V
12V
R30
R31
R32
R33
R34
R35
R36
R37
10k
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
C16
100n
JP3
KEYBOARD
PB3
PB1
PB6
PB4
D7
D5
F
RS
D6
D4
R/W
PB2
PB0
PB5
R15
R49
R50
PA15
PA13
PA14
C4
C6
C42
C43
C44
C7
C5
1µ
IC1
SH7032
AVCC
VCC
AVREF
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
CS4
PA1
CS6
PA3
WR
WRH
RD
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PA0
PA2
PA4
PA5
PA6
CK
RES
MD0
MD1
MD2
GND
EXTAL
XTAL
NMI
WDTOVF
AVSS
CS0
CS1
CS2
CS3
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
AD8
AD9
AD10
AD11
AD12
AD13
AD14
AD15
X1
C2
C3
22p
X1 = 14.7456MHz
C32
IC8
AM29F
040B
EPROM
D0
D1
D2
D3
D4
D5
D6
D7
CE
OE
WR
C33
C34
IC2
IC3
628128P
SRAM
CS2
NC
CS1
WE
OE
PA8
WRH
JP5
C40
C41
10µ
12VGND
5VGND
R53
R52
100k
36k
IC11
MAX692
ACSA
VOUT
WOI
RST
PFI
PFO
VBATT
C10
C11
C12
C9
C8
IC17
MAX202
C1+
C1–
V+
V-
C2+
C2–
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
PB11
PB10
TXD
RXD
JP2
RS232
C35
C36
C37
C38
C39
IC14
IC15
IC16
74ACT245
G3
3EN1
3EN2
D8
D9
D10
D11
D12
D13
D14
D15

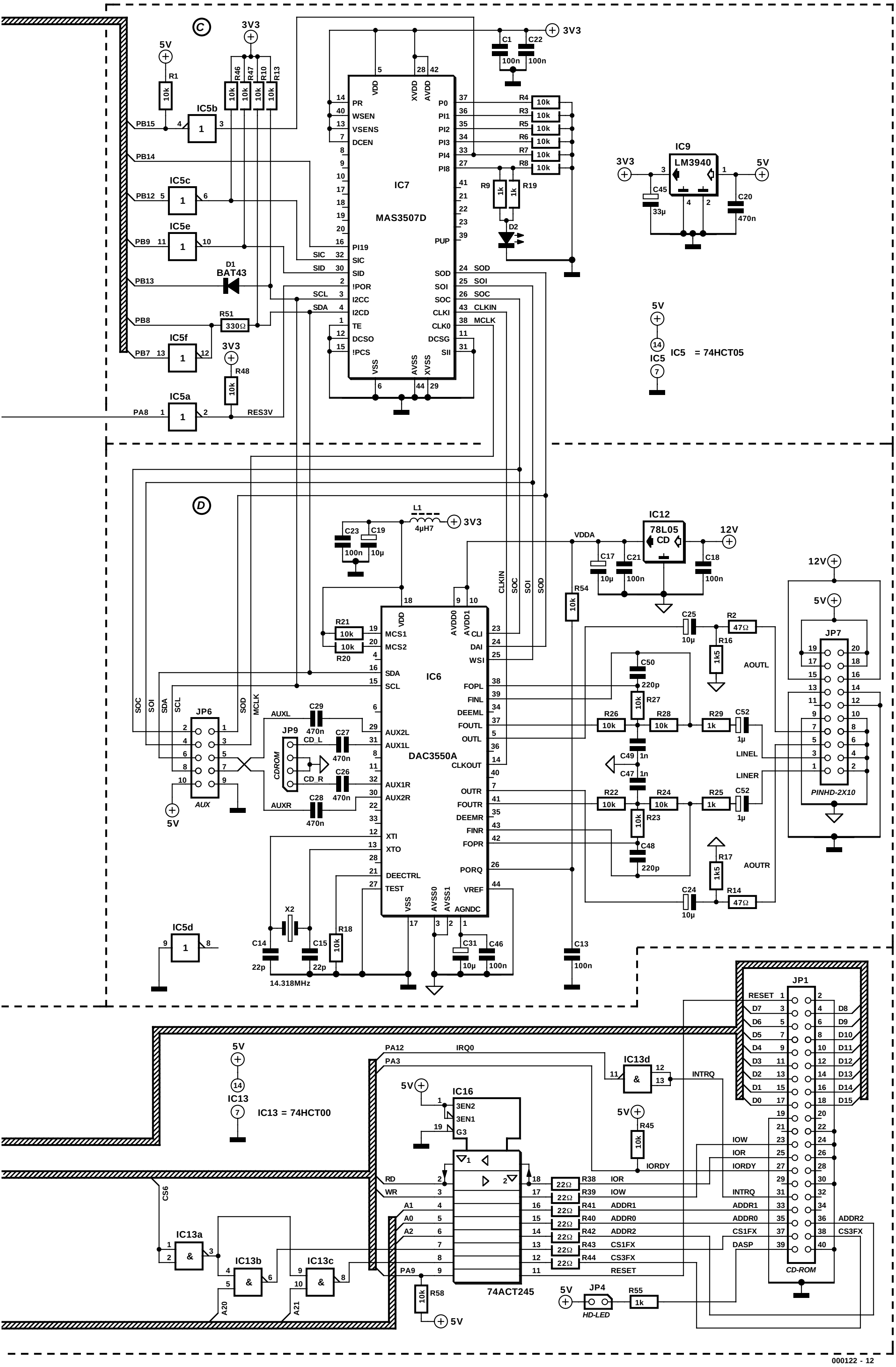


Figure 2

A. The boss: a Hitachi SH7032 32-bit microcontroller

B. Memory expansion: flash EEPROM and SRAM

C. The MP3 decoder, with an I^{2}C-bus link to the controller and I^{2}S-bus to the D/A converter.

D. Thi high-quality digital-to-analogue converter provides both line and headphone outputs.

E. ATAPI interface hardware for the CD-ROM and hard-disk drives.

F. The RS232 interface converter and front panel connector.

Table 1. Port functions

Port	Function	Alternative Function
PA0	Chip select SRAM	CS4
PA1	-	CS5
PA2	Chip select IDEiInterface	CS6
PA3	WAIT to generate wait states	CS7
PA4	Memory write	WR
PA5	Lower byte select for 16bit wide memory	LBS
PA6	Memory read	RD
PA8	Reset MPEG decoder	
PA9	Reset IDE devices	
PA10	spare	
PA11	spare	
PA12	IRQ IDE interface	IRQ0
PA13	IR detection	IRQ1
PA14	spare (wired to JP3)	IRQ2
PA15	spare (wired to JP3)	IRQ3
PB0	LCD D4	
PB1	LCD D5	
PB2	LCD D6	
PB3	LCD D7	
PB4	LCD RS	
PB5	LCD R/$\overline{W}$	
PB6	LCD E	
PB7	I2C data out	
PB8	I2C data in	
PB9	Serial data to MPEG decoder	TxD0
PB10	RxD RS232	RxD1
PB11	TxD RS232	TxD1
PB12	Serial clock to MPEG decoder	SCK0
PB13	I2C clock	
PB14	IRQ demand from MPEG decoder	IRQ6
PB15	IRQ frame from MPRG decoder	IRQ7
PC0-7	Keyboard	ACD inputs

troller, which is a member of the Hitachi SH1 family, is relatively inexpensive in spite of its impressive capabilities. In the first place, it has a very large selection of freely assignable I/O pins and a multitude of integrated peripheral blocks, including serial interfaces (SCI), timers, an interrupt controller (INTC), a DMA controller (DMAC) and an analogue-to-digital converter (ADC). Secondly, it has a large address space (more than adequate for the needs of OSCAR) and a powerful instruction set.

The controller has one 8-bit port and two 16-bit ports (PC0–7, PA0–15 and PB0–5). The wide ports can be used as both inputs and outputs, while the narrow port (PC0–7) can only be used for inputs. The latter port is connected to the A/D converter, so that eight analogue inputs are available. A few leads have special functions, and can for example be used as interrupt sources or for the selection of peripheral subassemblies such as the keyboard, LCD, MP3 decoder, real-time clock and so on. These functions, which are summarised in Table 1, are assigned to the ports when the MP3 player is switched on.

The processor has eight independent memory regions and can address up to 16 MB in each region. In can even support DRAM. The controller itself has 4 kB (7034) or 8 kB (7032) of internal RAM. An additional 256 kB of static RAM is connected to the 16-bit data bus of the controller, and occupies the address region from 0400000_H to $05FFFFF_H$. This consists of two 128 k × 8 SRAMs, type 62812P, with an access time of 15 ns. A 512-kB flash EEPROM (an AMD AM29F040B or Atmel AT29C040) is also connected to this bus, and addressed in the region between 0 and $0FFFFFF_H$ (see **Figure 2B**). The operating system of the player is located in the flash EEPROM, and it is partly copied to the faster SRAM when the player starts up. A small monitor program that is also located in the flash EEPROM allows the main program in the device to be updated. This memory can also be used for non-volatile storage of playlists or short jingles, for example.

The system clock comes from an integrated oscillator and an external 14.7546-MHz crystal. This allows serial data transfer rates of up to 230.4 kb/s.

MP3 decoder

The MP3 decoder is a MAS3507D, whose partial circuit diagram is shown in **Figure 2C**, was originally developed as part of the Worldspace satellite radio project. Consequently, it can synchronise itself to the data rate of an external MP3 bit stream. In

Table 2. IDE register set

/CS1FX	/CS3FX	A2	A1	A0	Read Access	Write Access
0	1	1	1	0	Alternate Status	Device Control
1	0	0	0	0	Data (16bit)	
1	0	0	0	1	Error	Features
1	0	0	1	0	Sector Count	
1	0	0	1	1	Sector Number / LBA Bits 0-7	
1	0	1	0	0	Cylinder Low / LBA Bits 8-15	
1	0	1	0	1	Cylinder High / LBA Bits 16-23	
1	0	1	1	0	Device/Head / LBA Bits 24-72	
1	0	1	1	0	Status	Command

History

The first OSCAR prototype was developed as part of a study project at the Institute for Communications Technology in the Electronics Technology faculty of the Technical University of Darmstadt. As part of the ongoing search for interesting study projects, a pair of students were given the assignment of building an MP3 CD player. A picture of the first OSCSAR (Optical Storage Compressed Audio Replay), which came into existence in the course of six months of labour, can be found at http://www.oscar-mp3.com/pic.html. The first prototype could already display titles in plain text, but it still strongly resembled a conventional CD player. OSCAR was later completely reworked, and the hard-disk function was added. This yielded the new OSCAR, which not only has a completely new user interface that is matched to MP3, but also incorporates many new functions and improvements suggested by the initial OSCAR users. OSCAR is now available as a commercial product, either as a ready-made unit or as a construction kit with pre-finished circuit boards.

OSCAR, the decoder is used in a different mode, which is called MULTIMEDIA. It independently generates the MP3 bit clock, and it demands data from the data source via the DEMAND signal (PI19). Once the buffer in the decoder has been filled, this signal becomes inactive. Once the data have been processed to the point that new data can be accepted, DEMAND goes High, which is signalled to the controller via IRQ6.

The MP3 data (SID) are transferred synchronously from serial interface 0 of the controller to the decoder, with the processor providing the 1-MHz clock signal (SIC). Since this data transfer rate is always greater than the bit rate of the MP3 signal, the data can be transferred in blocks. This offloads the processor and prevents buffer underflow. The MAS3507D generates an IRQ7 for each MPEG frame. This forms an excellent basis for a playing time display. The data transfer takes place via the serial data interface using direct memory access (DMA), independently of current program processes.

The MP3 decoder and the following D/A converter communicate with the processor via an I^{2}C bus. Although this is a bi-directional bus, the microcontroller does not have an open-collector output, so a Schottky diode (D1) must be inserted in the SCL lead. This roughly simulates an open collector.

Two port leads of the microcontroller are provided for the bi-directional I^{2}C-bus data signal. In the direction from the controller to the I^{2}C slave, there is an inverter with an open-collector output (IC5f), while in the opposite direction the signal enters via port pin PB8. As long as IC5f is not active, the other components can readily pull the bus to earth.

The use of the open-collector inverter also directly eliminates another problem. Both the MP3 decoder and the DAC have a 3.3-V supply voltage, in contrast to the TTL-compatible controller. If a 74ACT05 open-collector inverter is placed between the controller and the 3.3-V ICs, this incompatibility is resolved. In the other direction, the 3.3-V lines may be connected directly to the controller. The 3.3-V supply voltage is provided by an LM3940 low-drop linear voltage regulator (IC9), and the internal DC converter of the MAS3507D is not used.

Analogue interface

The DAC3550 D/A converter shown in Figure 2D is specially matched to the MAS3507D, and it provides a common clock signal to the converter. A standard 14.318-MHz crystal (X2) is used for the clock. However, the exact frequency of the crystal is not critical, since the DAC3550 synchronises itself with the data clock provided by the MPEG decoder.

The DAC3550 is a high-quality sigma-delta converter with an integrated anti-aliasing filter, an input multiplexer (which is also connected to the analogue output of the CD-ROM drive), an output amplifier for the line signal and a small power amplifier with volume control, for connecting to a headphone. The connection to the decoder runs via the three I^{2}S lines SOI, SOD and SOC. However, the internal registers of the converter are programmed via the I^{2}C bus.

The converter requires a 3.3-V supply voltage for the digital part (via the noise-reducing low-pass filter formed by L1, C19 and C23) and a 5-V supply voltage for the analogue part. A separate fixed-voltage linear regulator supplies the latter voltage. This, in combination with the separation of the earth tracks for the analogue and digital electronics and the carefully developed circuit board layout, fully exploits the capabilities of the DAC. This ultimately benefits the sound quality.

The anti-aliasing filter consists of a pair of two-pole low-pass filters (R22–R29 and C47–C50). In addition to the analogue CD-ROM input on JP9 (AUX1R and AUX1L), the AUX2L/R inputs are also available on JP6. The can be used to connect a radio, for example. The two bus systems (I^{2}C and I^{2}S) are also brought out on the same connector. This allows a digital S/PDIF output to be implemented, for example.

There is one more detail: the DACX23550A produces a fair bit of noise when the player is switched off. A MAX692 reset pulse generator (IC11) is used to suppress this noise. It generates an NMI pulse for the controller as soon as the supply voltage drops below 4.7 V. The controller then mutes the DAC via the I^{2}C bus.

IDE/ATAPI interface

The CD-ROM drive and the hard-disk drive are controlled via a standard IDE/ATAPI interface. The IDE interface basically consists of a 16-bit data bus and a 5-bit address bus (A0–A2, CS1FX and CS3FX), as well as supplementary control signals. In addition, the IDE device makes a set of registers available, which can be selected via the address lines. These are all 8-bit registers, except for the 16-bit data register. They are described in **Table 2**.

These registers are mapped into the I/O address region of the processor by the IDE controller, in the form of memory locations. This means that they can be written and read by the IDE host. In OSCAR, the IDE registers lie in one of the processor's eight address regions.

The IDE host controls the IDE device according to the ATA protocol. The command parameters are first written to the various

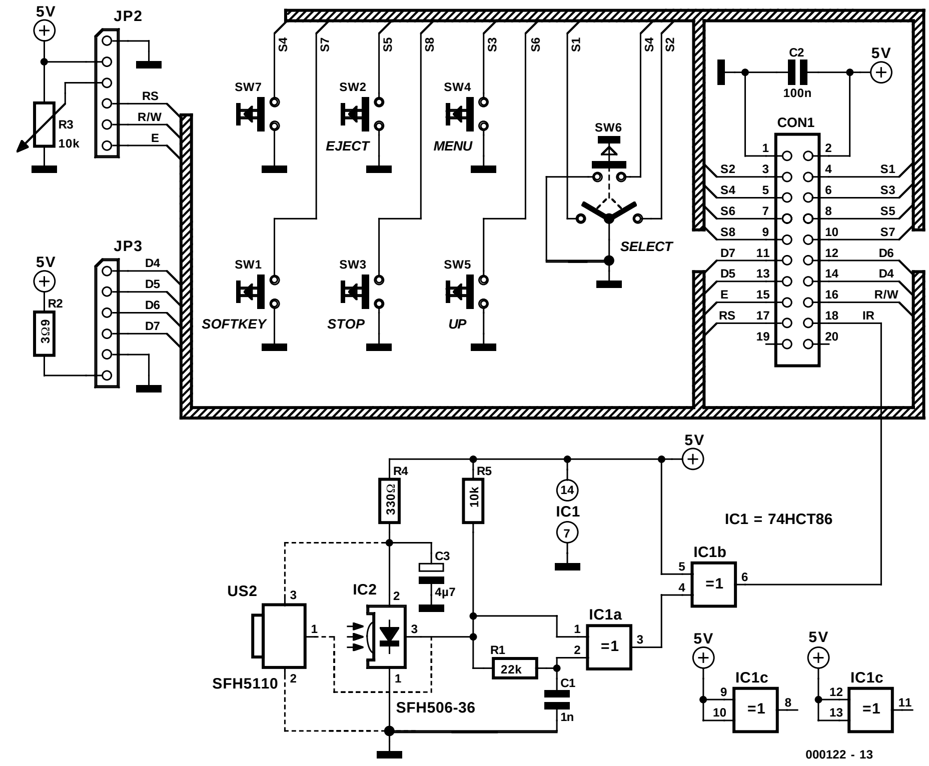


Figure 3. OSCAR is operated using five pushbuttons and a rotary encoder. All user information can be read from the LCD.

registers of the drive, and then the command word is written to the COMMAND register. The drive then executes the command. Data are subsequently transferred by repeatedly reading or writing data from or to the DATA register. The start of a data transfer is either signalled by an interrupt that is triggered by the drive, or determined by the host by repetitive polling of the STATUS register.

The communications with the CD-ROM drive, according to the ATAPI protocol, are based on the ATA protocol as just described. In contrast to the ATA protocol, however, a complete command packet (a block of 12 bytes) is first transferred to the drive via the DATA register. The drive then executes this command, and the actual data transfer takes place as usual via the DATA register.

Figure 2E shows the hardware that is used for the IDE/ATAPI interface in OSCAR. The nine registers of the IDE interface are accessible in address region 6 of the SH7032. They are addressed by address lines A0–A2, A20 and A21. Since the IDE interface allows the use of relatively long flat cables, all signals to the IDE device are passed through bi-directional bus drivers (type 74ACT245). The address and control signals (/RD and /WR) also pass through a 74ACT245 that is permanently wired for output. The $\overline{\text{RESET}}$ signal passes through this driver as well. The active-High interrupt signal of the IDE interface passes through an inverter, since the processor can only handle interrupts with falling edges.

The choice of a suitable hard disk drive or CD-ROM drive should not be especially critical. Any hard-disk drive that supports LBA addressing can be used, which means that all recent models from around 1 GB are suitable. Although FAT16 is also supported, it is recommended to format the disk in the FAT32 file system.

As far as we know, any modern CD-ROM drive that complies with the ATAPI standard can be connected. Only older-model drives (2×–8×) may present problems with 'home-brew' CDs.

Addresses

Designers:

Ingenieurbüro A. Kurpiers und V. Pantelic
Ludwigshöhstr. 77
D-64286 Darmstadt
Germany
Tel.: (+ 49) 61 51 / 16 23 69
Tel.: (+ 49) 61 51 / 16 25 69
FAX: Tel.: (+ 49) 61 51 / 1655 45
E-Mail: oscar@oscar-mp3.com
E-Mail: oscar@oscar-mp3.de

Distributors:

BayCom Hard- and Software
Bert-Brecht-Weg 28
D-30890 Barsinghausen
Germany
Tel.: (+ 49) 51 05 / 58 50 50
FAX.: (+ 49) 51 05 / 58 50 60
E-Mail: oscinfo@oscar-mp3.de
Internet: www.baycom.de

PCC-Computer
Vaalser Str. 148
D-52074 Aachen
Germany
Internet: www.p-c-c.de

Semiconductor Manufacturers:

Micronas Intermetall
Hans-Bunte-Str. 19
D-79108 Freiburg
Germany
Tel: (+49) 7 61 / 5 17 - 0
Fax: (+49) 7 61 / 5 17 - 21 74
E-Mail: info@intermetall.de
Internet: www.intermetall.de
www.intermetall.com
www.itt-sc.de

Hitachi Europe
Electronic Components Group
Dornacher Str. 3
D-85622 Feldkirchen bei München
Germany
Tel.: (+ 49) 89 / 9 91 80 - 0
FAX:(+ 49) 89 / 9 91 80 - 2 65
Internet: www.hitachi-eu.com/hel/ecg/index.htm

Software and information:

ATA/ATAPI-drivers:
www.ata-atapi.com

General information on MP3:
www.come.to/mp3chips

Peripherals

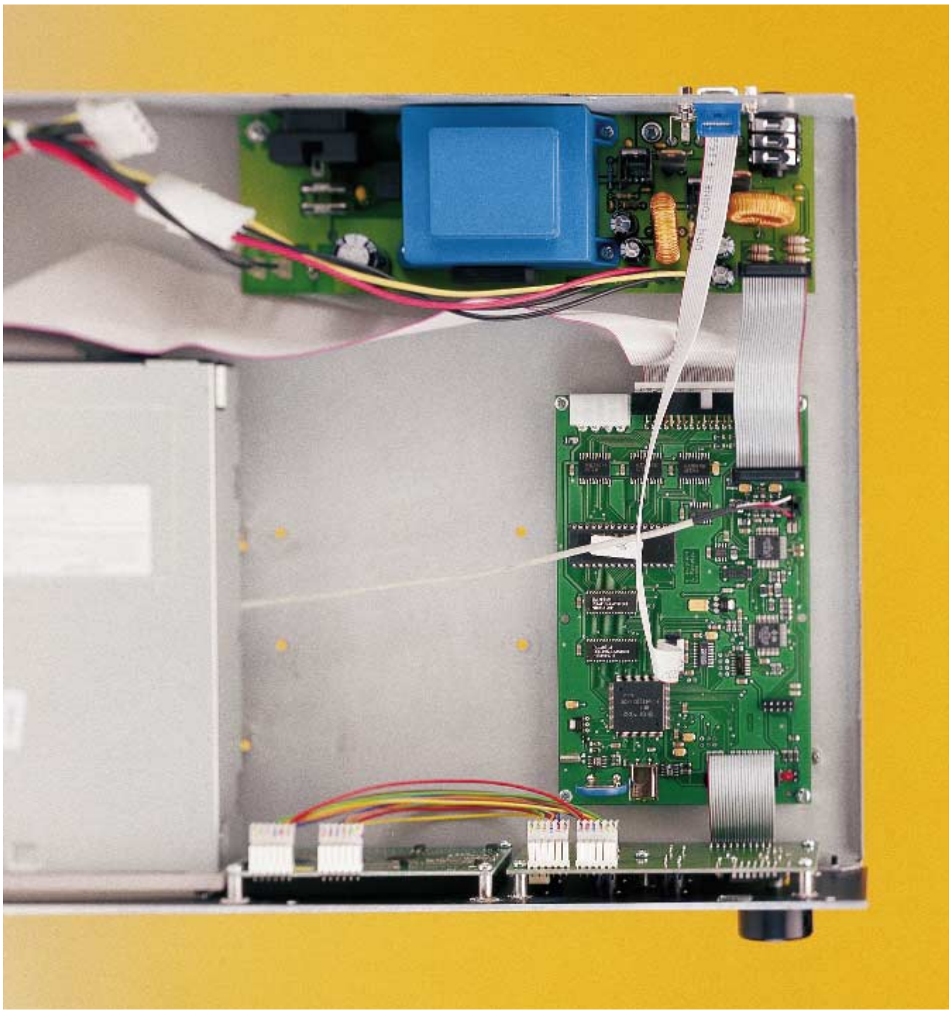
The block diagram shows a series of peripheral subassemblies, whose 'interfaces' can be seen in the full schematic diagram of the main circuit board (**Figure 2**).

Liquid crystal display

A standard, industrial dot-matrix LCD with LED backlighting is used. The display is driven in the 4-bit mode. Naturally, the display is not mounted on the main circuit board, but instead on the front-panel circuit board. It is connected to JP3.

Keyboard

The four pushbuttons EJECT, MENU, STOP and UP, along with the SELECT rotary encoder, are connected to the 8-bit analogue port of the processor. They are periodically polled via a timer interrupt. The encoder works like a pair of switches that are sequentially opened and closed. The controller can determine the direction of rotation from the switching time sequence. Debouncing takes place in software, so the amount of hardware is minimal.



RS232 interface

The controller has two serial interfaces, one of which is compatible with the asynchronous RS232 standard. The baud rate clock is generated by the controller and has a maximum value of 230,400 kb/s. A MAX202, which is a variant of the well-known MAX232A, converts the signals to RS232 levels.

The serial interface serves for future downloads of new software and for issuing status messages during operation, so it is an important aid in program development. With the addition of a modem, music-on-demand can be loaded via the serial interface. In addition, this interface can be used for remote control of the player. A debugger in the PC can communicate with the processor via the RS232 interface (without hardware handshaking).

IR receiver

An infrared receiver (type SFH506-36 or SFH5110) is also mounted on the front panel circuit board. Since both the Philips RC5 and the Sony SIRC codes are based on pulse-width modulation, both the positive and the negative edges must be evaluated. Consequently, the receiver generates a short pulse on the PA13 port lead for each signal edge. This pulse is produced by an RC network (R1 and C1) and two XOR gates (IC1a/b in **Figure 3**). The controller generates an IRQ1 interrupt for each falling edge of this pulse, and can thus evaluate the incoming data.

RTC and Silicon ID

A real-time clock and a unique serial number in the form of a Silicon ID chip from Dallas Semiconductor are both present on the circuit board, but they are presently unused. They are reserved for future applications.

Power supply

In the ready-made version of the player, a switch-mode power supply is used. In principle, however, a linear power supply can be used instead, as long as it can deliver a clean 12 V. For mobile use (with a car battery, for example), it is a good idea to use a DC/DC converter that can convert upwards as well as downwards. The proper contact in this case is the semiconductor manufacturer Linear Technology.

Attractive prospects

A project such as this lends itself to DIY construction only to a limited degree. The tracks on the Eurocard circuit board are very closely spaced, and manually soldering SMD ICs with more than 100 leads is not everyone's cup of tea. Nevertheless, the DIY aura is not completely lost, since OSCAR is available not only as a ready-made unit, but also as a complete kit with pre-finished circuit boards. If you are planning a very special application, such as an automotive MP3 player, you can also obtain the main and keyboard circuit boards together with the LCD.

In the second part of this article, you will learn everything about the construction of the kit, as well as the structure and operation of the software.

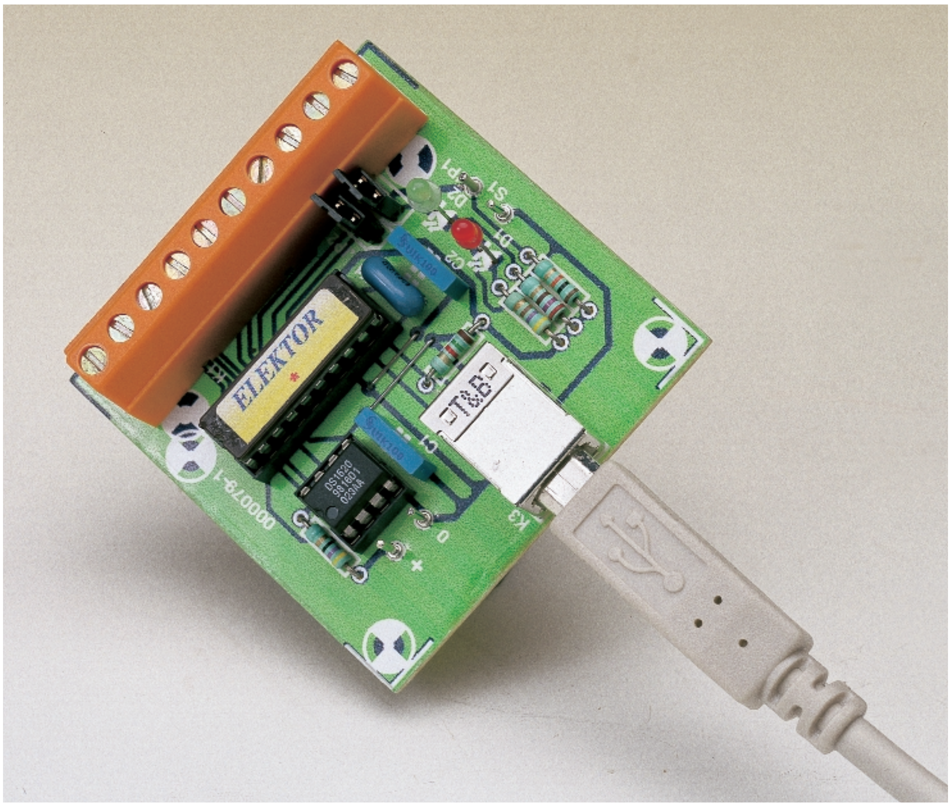
(000122-1)

USB Interface

Experimenting with the Universal Serial Bus

Design by B. Kainka

Everybody is talking about the USB. In the long term, this new serial interface could replace many of the PC interfaces that have been used up to now. This is reason enough for looking at it more closely.



Before getting into practical details, let's first look at the basic features of the Universal Serial Bus. In USB Version 1.0, there are 'lowspeed' devices with a data rate of 1.5 Mb/s, and 'fullspeed' devices with a data rate of 12 Mb/s. Even the lower data transfer rate is several times faster than the possible baud rates of a serial interface.

Up to now, we have often been faced with unpleasant situations in which all of the PC interfaces are already in use. The advantage of the USB here is that you can always make four new ports available by connecting a supplementary bus distributor (hub). Up to 127 devices in total can be connected to the bus.

The USB connection also provides the operating voltage for small devices. Up to 100 mA can be drawn from the interface without any special provisions. With a special demand login, this can be increased to 500 mA. Typical laboratory applications often need significantly less than this. Eliminating the extra power cable for each device helps in the struggle against the general 'cable salad'.

USB devices can always be connected and disconnected while the system is operating. The operating system automatically loads the necessary driver. This extended Plug-and-Play capability considerably simplifies working with multiple devices. A bus number is automatically assigned to each newly connected device (enumeration). The operating system first reads a certain amount of information from the device and then automatically loads a suitable driver.

This means that every USB device is already significantly more intelligent than a device with an RS232 interface. It must make itself known to the operating system by means of a series of description tables (descriptors), in order that the correct driver can be selected. With the very first connection, the operating system requests a diskette with the driver. For all subsequent reconnections, the driver is loaded automatically, so that the user does not have to take any action. If the device is disconnected from the PC, the driver is automatically deleted from memory. Unfortunately, USB can presently be used only with PCs that use the Windows 98 operating system. Although USB can in principle be used under Windows 95 from version SR2 onwards, this is not completely reliable. Consequently, Windows 98 is recommended for use with the USB Interface. Windows NT5 and Windows 2000 support USB, but Windows NT4 does not support USB at all. However, the software for this project has not been tested with Windows 2000.

The circuit

The USB Interface is based on the Cypress CY7C6300 microcontroller. This IC was primarily developed for small USB devices, such as mice and joystick ports, and it contains a complete USB engine for the lowspeed USB. The two data lines (D+ and D–) are connected directly to the microcontroller. **Figure 1** shows the block diagram of the IC. In addition to a RISC processor, it contains an OTP ROM for the operating software (firmware) and two ports with a total of twelve output lines, as well as RAM and a timer.

The USB Interface presented here is based on a Cypress application. Cypress made their USB thermometer available in the CY3640 starter kit. This kit was intended mainly to promote familiarity with the processor and the circuit technology. It is no longer available in the original form, but we have resurrected it in this Elektor project. You can use the USB Interface to get acquainted with the USB, and you can also use it for many different types of serious applications.

The USB Interface includes the following functions:

- temperature measurement
- polling a pushbutton switch
- setting the brightness of a LED

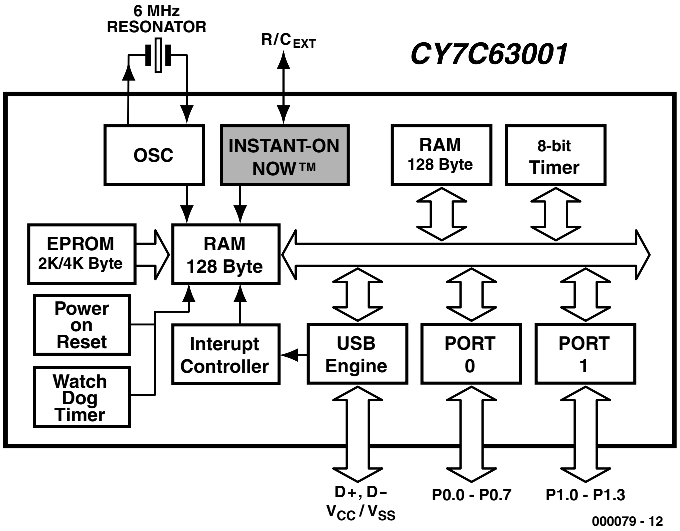


Figure 1. Block diagram of the CY7C63001.

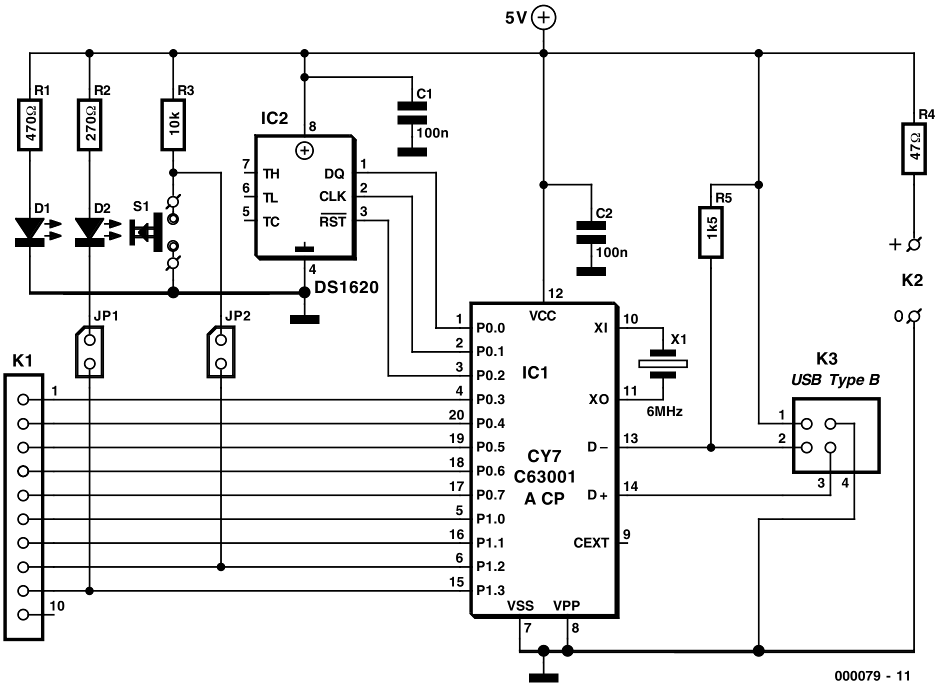


Figure 2. Complete schematic diagram of the USB Interface with temperature sensor.

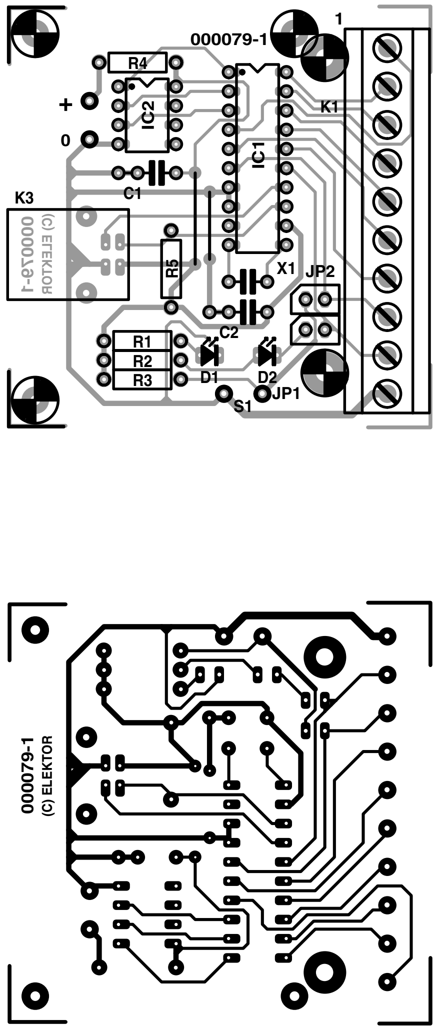


Figure 3. The USB Interface can be build using this printed circuit board.

COMPONENTS LIST

Resistors:
R1 = 47Ω
R2 = 270Ω
R3 = 10kΩ
R4 = 47Ω
R5 = 1kΩ5

Capacitors:
C1,C2 = 100nF

Semiconductors:
D1 = LED, red,
D2 = LED, green
IC1 = CY7C63001ACP (programmed, order code **000079-41**)
IC2 = DS1620

Miscellaneous:
JP1,JP2 = 2-way pinheader with jumper
K1 = 10-way PCB terminal block
K2 = solder pin
K3 = USB socket, PCB mount, type B (Farnell #153-503)
S1 = pushbutton 1 make contact
X1 = 6MHz ceramic resonator, (Murata CSA6.00MG (Farnell # 295-292), or Newport ZTA6.00MT
Enclosure: approx. size 61x22x80 mm (Conrad Electronics #522848)
Project disk, order code **000079-11**
PCB, order code **000079-1**

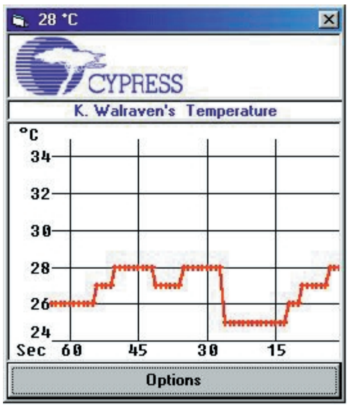
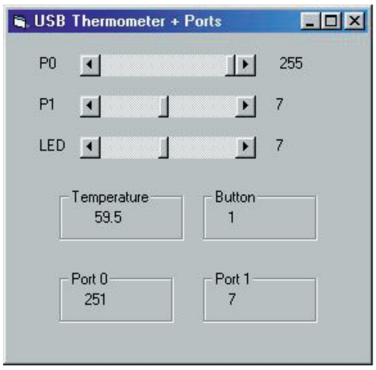


Figure 4. Screen display of the original Cypress demo program.



P0 255

P1 7

LED 7

Temperature 59.5

Button 1

Port 0 251

Port 1 7

Figure 5. The program allows access to all port functions of the microcontroller, and also displays the temperature.

– nine freely-accessible I/O ports

The USB Interface is fully compatible with the original Cypress thermometer application. This means that you can use all sample programs and the thermometer driver. However, in the original application, the port accesses were not fully functional. Consequently, the firmware has been modified, and it also includes additional functions.

Figure 2 shows the circuit diagram. The temperature is sensed by a Dallas DS1620 IC, which occupies three port lines (P0.0–P0.2). The remaining nine lines (P0.3–P0.7 and P1.0–P1.3) are led out to terminal strip K1, for general use. In addition, a green LED (not a low-current type!) can be connected to port connection P1.3 via jumper JP1. This LED indicates successful enumeration, and it can be set to 16 different brightness levels. However, this works only if pushbutton switch S1 is connected to port P1.2 via jumper JP21. The pushbutton toggles an internal program register, which can be polled via the USB.

The two bus lines, D+ and D–, are led to a Type B USB connector. If you cannot come up with such a connector, you may solder the USB cable directly to the circuit board. The USB Interface takes its operating voltage (+5 V and earth) from the bus. The second (red) LED, D1, which should be a low-power type, indicates the presence of the supply voltage. This voltage can also be used for other purposes via the solder posts K2. The 47-kΩ resistor ensures that no more than the allowed 100 mA can be drawn from the USB connection in case of a short circuit.

The microcontroller is clocked by a 6-MHz ceramic resonator with two (and not three!) leads. The necessary capacitors are already integrated into the IC.

Construction and testing

The construction of the small printed circuit board shown in **Figure 3** should not present any problems. Sockets may be used for both ICs. Once everything has been assembled and there are no obvious construction errors, there comes the moment of truth. Connect the USB Interface to a PC using a type A–B USB cable. After a moment, Windows will recognise that new hardware has been connected. A window will appear with a request to insert a diskette with a suitable driver. You can comply by inserting the diskette labelled **000079-11** in drive A. Here the system will find the file

Info on the Internet

Microsoft VB5
www.microsoft.com/msdownload/vbcce.htm

Cypress CY7C63001ACP microcontroller
http://www.cypress.com/cypress/prodgate/usb/cy7c63001.html

Dallas temperature sensor
http://www.dalsemi.com/datasheets/pdfs/1620.pdf

CypressSemiconductorsCYPRESS.INF , which describes the device and the driver. Following this, it loads the actual driver *USBTherm.sys*. As soon as this process is completed, the device is enumerated, which means that it is logged on to the system. Now, not only the red LED is lit (if JP2 is installed) but also the green LED.

The Cypress demo software on the diskette, *Thermometer.exe*, implements a thermometer with a graphic display of temperature versus time. If JP1 is installed, the user can use the keyboard to select Fahrenheit or Celsius degrees for the temperature display. In addition, the brightness of the green LED can be altered. The microcontroller includes a 4-bit D/A converter that can be used to set the level of the current sunk by the Port 1 outputs. However, the software only allows Port P1.3 to be set. **Figure 4** shows how the temperature is displayed on the monitor.

Programming in Visual BASIC

An interface is not especially interesting if you cannot drive it using your own programs. You can access the device via a small Visual BASIC program. We use V5BCCB, which Microsoft allows to be downloaded free of charge and which is fully functional, but which cannot generate executable stand-alone applications.

Listing I. The USBI.BAS module with declarations

```
Attribute VB_Name = “Module1”
Type SECURITY_ATTRIBUTES
nLength As Long
lpSecurityDescriptor As Long
bInheritHandle As Long
End Type

Type OVERLAPPED
 Internal As Long
 InternalHigh As Long
 offset As Long
 OffsetHigh As Long
 hEvent As Long
End Type

Declare Function CreateFile Lib “kernel32” Alias “CreateFileA”
  (ByVal lpFileName As String, ByVal dwDesiredAccess As Long,
   ByVal dwShareMode As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES,
   ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long,
   ByVal hTemplateFile As Long) As Long
Declare Function DeviceIoControl Lib “kernel32” (ByVal hDevice As Long,
   ByVal dwIoControlCode As Long, lpInBuffer As Any, ByVal nInBufferSize
   As Long, lpOutBuffer As Any, ByVal nOutBufferSize As Long,
   lpBytesReturned As Long, lpOverlapped As OVERLAPPED) As Long
Declare Function CloseHandle Lib “kernel32” (ByVal hObject As Long) As Long

Public Security As SECURITY_ATTRIBUTES
Public gOverlapped As OVERLAPPED
Public hgDrvrHnd As Long
Public Const GENERIC_READ = &H80000000
Public Const GENERIC_WRITE = &H40000000
Public Const FILE_SHARE_WRITE = &H2
Public Const FILE_SHARE_READ = &H1
Public Const OPEN_EXISTING = &H3
```

Listing 2. Accessing all ports and functions

```
Dim sFileName As String
Dim htemp As Long
Dim lIn As Long, lInSize As Long, lOut As Long, lOutSize As Long, lSize As Long
Dim lTemp As Long
Sub USB_IO()
  sFileName = “\\.\Thermometer_0”
  hgDrvrHnd = CreateFile(sFileName, GENERIC_WRITE Or
    GENERIC_READ, FILE_SHARE_WRITE Or FILE_SHARE_READ,
    Security, OPEN_EXISTING, 0, 0)
  lTemp = DeviceIoControl(hgDrvrHnd, 4&, lIn, lInSize,
    lOut, lOutSize, lSize, gOverlapped)
  htemp = CloseHandle(hgDrvrHnd)
End Sub

Sub Brightness(Level)
  lIn = Level * 256 + 14
  lInSize = 2
  lOutSize = 1
  USB_IO
End Sub

Function RdPort(Port) As Integer
  lIn = Port * 256 + 20
  lInSize = 2
  lOutSize = 2
  USB_IO
  RdPort = (lOut / 256) And 255
End Function

Sub WrRAM(Adresse, Wert)
  lIn = 65536 * Wert + Adresse * 256 + 23
  lInSize = 3
  lOutSize = 1
  USB_IO
End Sub

Private Sub HScroll1_Change()
  Wert = HScroll1.Value * 8 + 7
  WrRAM 46, Wert
  Label4.Caption = Str$(Wert)
End Sub

Private Sub HScroll2_Change()
  WrRAM 47, HScroll2.Value
  Label5.Caption = HScroll2.Value
End Sub

Private Sub HScroll3_Change()
  Brightness HScroll3.Value
  Label6 = HScroll3.Value
End Sub

Private Sub Timer1_Timer()
  lIn = 11
  lInSize = 1
  lOutSize = 3
  USB_IO
  Temp = ((lOut \ 256) And 255) / 2
  Minus = (lOut \ 65536) And 255
  If Minus > 0 Then Temp = Temp * -1
  Button = (lOut \ 16777216) And 255
  Label7.Caption = Str$(Temp)
  Label8.Caption = Str$(Button)
  Label9.Caption = Str$(RdPort(0))
  Label10.Caption = Str$((RdPort(1)) And 15)
End Sub
```

Functions

0Bh: Read Thermometer
lIn: 0Bh (Länge: 1 Byte)
iOut: Button, Sign, Temp, Status (Length: 4 Bytes)

0Eh: Set LED Brightness
lIn: Brightness (0-15), 0Eh (Length: 2 Bytes)
iOut: Status (Length: 1 Byte)

14h: Read Port
lIn: Port (0,1), 14h (Length: 2 Bytes)
iOut: Wert, Status (Length: 2 Bytes)

15h: Write Port
lIn: Value (0-255), Port (0,1), 15h (Length: 3 Bytes)
iOut: Status (Length: 1 Byte)

16h: Read RAM
lIn: Address (0-255), 16h (Length: 2 Bytes)
iOut: Value, Status (Length: 2 Bytes)

17h: Write RAM
lIn: Value (0-255), Address (0-255), 17h (Length: 3 Bytes)
iOut: Status (Length: 1 Byte)

18h: Read ROM
lIn: Index, Address (0-255), 18h (Length: 3 Bytes)
iOut: Value, Status (Length: 2 Bytes)

On project disk # 000079-II

CypressSemiconductorsCYPRESS.INF	Installation information for the USB driver
Thermometer.exe	Cypress demo software
USB_20e.ASM	Microcontroller assembler file
Usb_20e.hex	Microcontroller HEX file
Usb_20e.lst	Microcontroller source code
USBTherm.sys	USB driver
USBelektor.vbp	Visual BASIC project
USBelektor.frm	Visual BASIC format file
usb1.bas	Visual BASIC module
copyright.txt	Copyright notice
contents.txt	Diskette contents

After installing the program package, open the project in drive A, and then select and open *USBelektor.VSP.* The project encompasses the module *USB1.BAS* (see the text box), with its own declarations, and the format file *USBElektor.FRM*, which is responsible for not only the graphic display on the monitor, but also for the actual application (communications with the Cypress microcontroller and data processing). Start the program by clicking on the Run menu.

For access to the SYS driver, you need the Windows functions CreateFile, CloseHandle and DeviceIoControl. They are declared here in the *USB1.BAS* module.

The format file *USBElektor.FRM* (see **Listing 2**) allows all port functions of the microcontroller to be accessed. In addition, all of the original thermometer functions should be retained. For running the driver, you need certain data that are described below. The monitor display that is generated by this program is shown in **Figure 5**.

When calling the driver with DeviceIoControl, you pass in a driver function number, an input buffer 1In and an output buffer 1Out. The driver function number is always '4', while the actual action is selected by a control byte in 1In. The driver returns two to four bytes, depending on the function. The least-significant byte always contains status information regarding the success of the access. In addition to reading out the thermometer, there are special functions for setting the brightness of the LED and for access to the ports and memory of the microcontroller. All of these are listed in the Functions box.

The Write Port function works only partially with the available driver. We therefore looked for a way to enable all port outputs without modifying the driver, employing only changes in the microcontroller firmware. A solution was found in the use of the WriteRAM command. Two supplementary RAM variables were defined:

Port0 outputs: 2Eh
Port1 outputs: 2Fh

The port data must now be written directly to particular memory addresses in the microcontroller RAM. The microcontroller then updates the port outputs. The firmware is contained on the diskette in the file *USB_20e.asm*.

(000079-1)

The Universal Serial Bus (USB)

Part 1: develop your own applications

By B. Kainka

The USB Interface described elsewhere in this issue demonstrates that the USB may be used for DIY projects related to measurement and control. But how does one develop USB compatible peripherals, which conditions should be satisfied, and where can the necessary information be found?

The USB is a serial bus system designed for connection to and addressing of several peripheral devices. USB is considerably more complex than RS232, but also much faster at up to 1.5 Mbit/s (for low-speed devices) or 12 Mbit/s (for full-speed devices). Furthermore, the designers of the USB have paid a lot of attention to simplicity and ease of use. It is fair to say that the concept of Plug and Play, for the first time, actually applies to the USB, which allows devices to be connected and automatically initialised with the PC switched on or off. With USB, you don't have to grapple with interrupt conflicts, wrong addresses and missing drivers.

Making it 'simpler' for the user means a more complex job for the designer. If you want to design your own USB peripheral device, you should be ready to delve into the USB standard, program your own microcontroller and write your own driver. The inset box provides some suggested starting points.

USB cables and power supply

Two different connector types are around: type A and type B. the system is designed such that these can not be mixed up. As opposed to RS232, the so-called 'crossed' cable does not exist with USB, all cables have simple 1-to-1

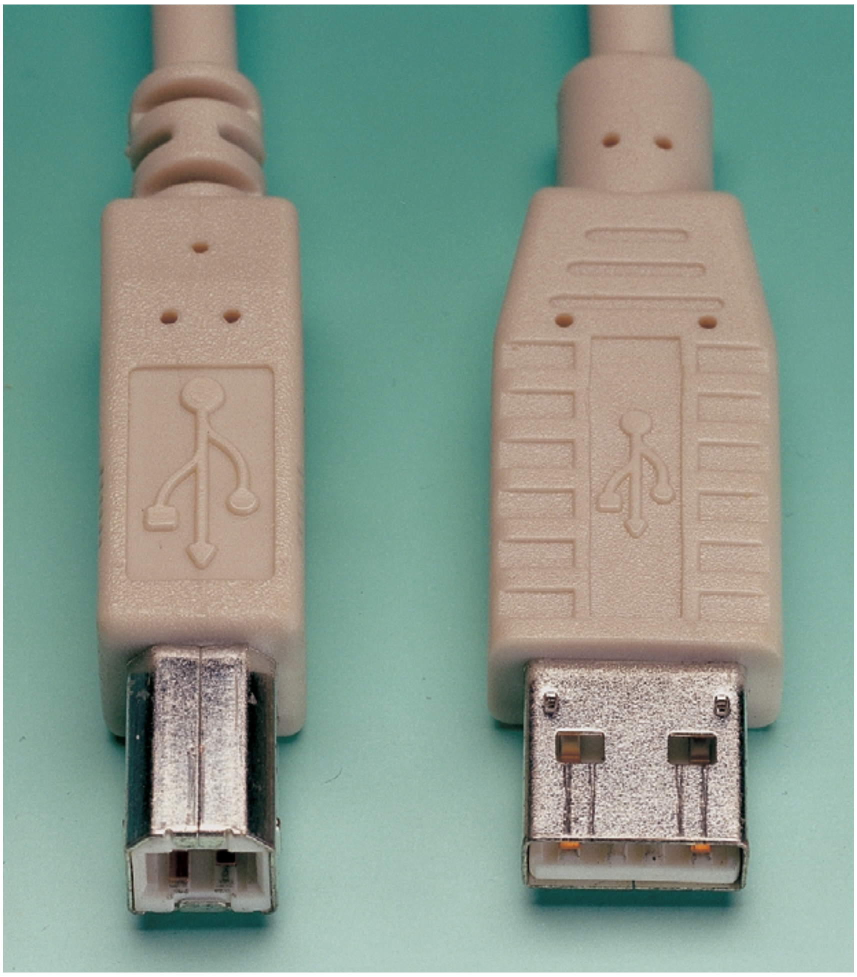


Figure 1. The plugs used.

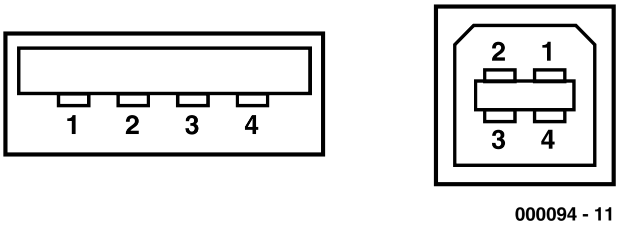


Figure 2. Connections of plugs type A and B.

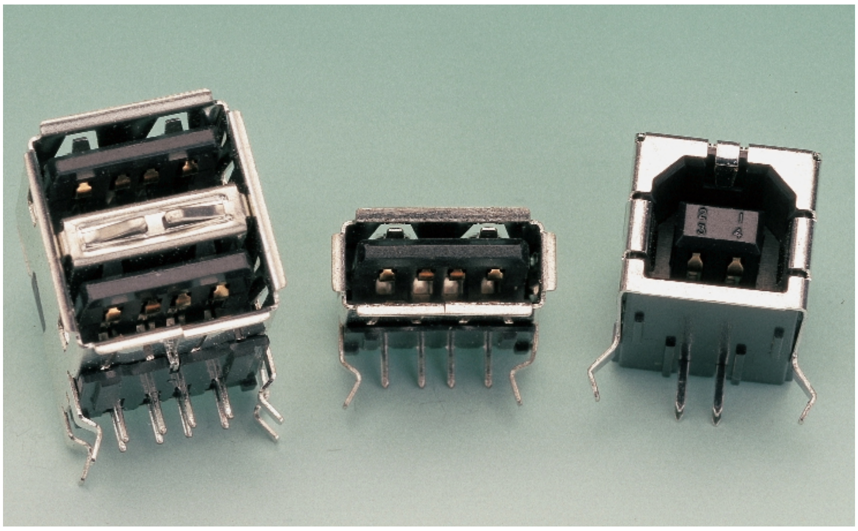


Figure 3. USB socket type B.

correspondence between the pins at either side, and the pin functions are always the same:

1 +5 V
2 Data –
3 Data +
4 Ground

At the rear of a modern PC you will typically find two sockets of the A type. These may be used for direct connection to two devices. Relatively small low-speed peripherals such as a mouse generally employ a thin, fixed cable with a type-A plug. In all other cases, the peripheral will have its own USB socket of the B type. The connection to the PC is then made by means of a type-A cable.

Such cables are only supplied ready-made and completely moulded — spare USB plugs are not available. The length, cable diameter and screening of USB cables are subject to tight specifications. Here, the difference between full-speed and low-speed is an important factor. The system adopted for ready-made cables can be relied on to prevent a low-speed cable being used where a full-speed cable is called for. All cables are full-speed cables unless permanently attached to the USB device (like the previously mentioned mouse). Where a cable is not long enough, a special type A-A extension cable may be used.

While USB plugs are not sold in the shops, the mating socket is available. Though a bit thin on the ground, these (PCB mount) sockets allow you to experiment with the USB. The bus will present a +5 V supply voltage which may be loaded at up to about 100 mA. Many digital circuits and microcontrollers operate from a 5-V supply voltage. By connecting such a circuit to the USB you get the required supply voltage thrown in as a bonus. One ‘must’ however, is that an efficient short-circuit protection be provided, like a regular fuse or a Polyswitch fuse.

The two datalines D+ and D– may only be employed in combination with special USB components, i.e., special microcontrollers, which, as a matter of course, are powered via the USB connector. After a special authorisation request, a peripheral is allowed to consume up to 500 mA.

The supply voltage on the USB is allowed to rise to 5.25 V and drop to 4.2 V. A good voltage regulator will have no problems stepping these levels down to 3.3 V. The entire system — cable and devices — is dimensioned such that the minimum supply voltage can not drop below 4.2 V. Devices drawing more than 100 mA have to report their requirement with the system and are only allowed access to the bus when sufficient current is available.

A clear distinction is made between Bus Powered and Self Powered USB peripherals. In many cases, either of the two modes may be selected. For example, the device may have a supply input socket for connection to an external mains adapter.

According to the USB standard, the current consumption from the bus is automatically limited. Therefore, if more current is drawn than allowed, the supply has to be switched off.

USB classes

Until recently, you would receive a free driver disk with a new mouse. Not so if you buy an USB savvy mouse these days. Having connected the mouse despite doubts and worries about the missing disk, you will be pleasantly surprised to see that the Windows 98 operating system will automatically find and install the driver. In fact, Windows 98 will load the HID (human interface device) driver, which has been lurking in your PC until a USB-compatible device of the ‘defined’ class is connected up. This class is supported by ready-made drivers.

HIDs include mice, keyboards, pointer

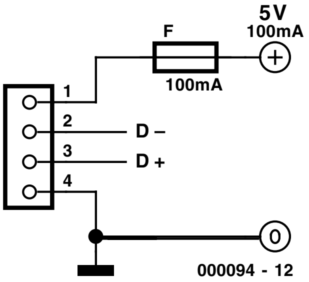


Figure 4. The USB as a stabilised power supply.

devices and joysticks. In addition to HIDs there are also USB classes for soundcards, printers and many other peripheral devices. The upshot is that all 'regular' devices are divided into classes, for which suitable drivers are available. This class allocation automatically creates a certain level of standardisation, after all, the device manufacturers have to make sure their products comply with the relevant class specifications.

Standard USB devices are no longer expensive. Not surprisingly, you may wonder if it is possible to 'misuse' a particular USB peripheral.

For example, a USB compatible soundcard will install itself in such a way that can be used as a regular, internal soundcard. After all, that is the purpose of the whole exercise: all available programs should be able to employ the USB device. The same goes for programs you make yourself. A quick test using PORT.DLL from the 'Compact' software supplied by Elektor for the extremely popular **Universal Interface for Windows** (December 1999) was a direct success. So, your own programs written in Delhi should be able to make use of any USB peripheral connected to a PC. However, our experience in this area is still too limited to be able to make the over-assertive statement that it 'always works'.

An experiment with a USB game controller produced the following results: a USB joystick port may be installed, among others, using DOS emulation. This allows direct access to the virtual port addresses of a game card. In this way, you have added a simple USB interface with four analogue and four digital inputs to your system.

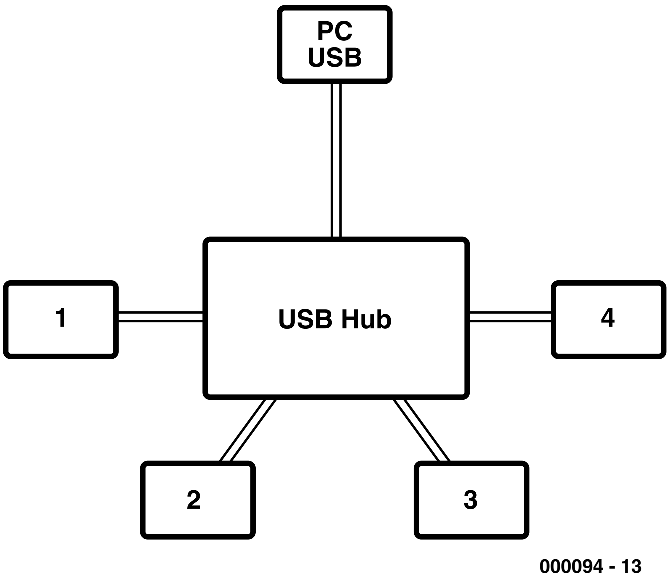


Figure 5. Daisy chain structure of the USB with one hub.

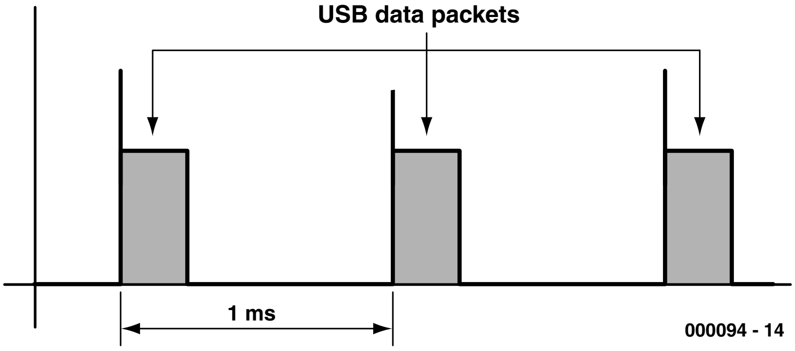


Figure 6. Data packets is 1-ms frames.

Bus topology

The USB is a star-shaped bus with a single master. To be able to connect several USB devices you will require a Hub, which is simply a bus-distributing device with several ports. This device is literally at the hub of the bus system. Usually, a USB hub has one Upstream and four Downstream ports. The PC, however, will already contain a hub to implement two USB ports. This so-called Root Hub is located on the PC motherboard.

A further hub may be connected to the downstream port of a hub. A total of seven hubs may be cascaded in this way, resulting in a maximum of 127 devices. That, however, is a theoretical number because the available bandwidth has to be distributed across all devices connected to the bus.

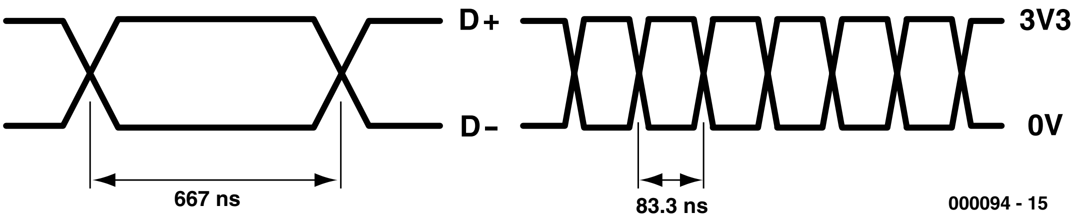


Figure 7. Low-speed and high-speed signals.

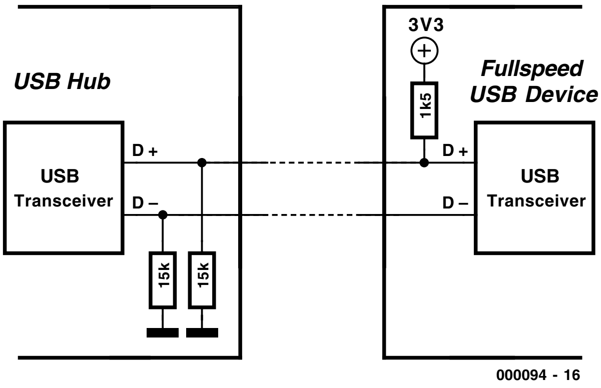


Figure 8. Recognition of a full-speed device.

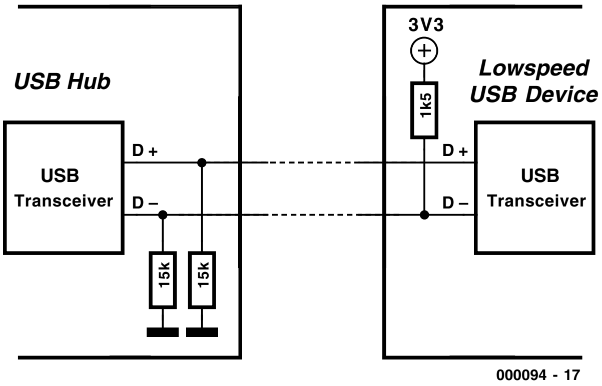


Figure 9. Recognition of a low-speed device

USB signals

The signals on the D+ and D– lines are difference signals with levels of 0 V and +3.3 V. The microcontroller in the USB peripheral will typically operate at a supply voltage of 3.3 V. The USB is a single-master bus — all activities originate from the PC. Data is conveyed in packets of eight to 256 bytes. The PC is authorised to request data from a peripheral. Conversely, a peripheral may transmit data to the PC.

All data traffic takes place in frames of about 1 millisecond. Within a frame, several data packets may be processed for several devices. Low-speed and full-speed packets may occur within the same frame. When several devices are addressed, a bus distributing device like the previously mentioned hub arranges the data distribution.

Low-speed devices operate at a data rate of 1.5 Mbit/s, so the length of a bit is 666.7 ns. On full-speed USB connections, data travels at a rate of 12 Mbit/s, which means that a bit last 83.33 ns. The speed is dictated by the master only. All USB slaves have to synchronise to the datastream. Because a separate clock signal is not distributed, the clock rate has to be recovered from the datastream. The NRZI principle (non return to zero) is used for that purpose. In this system, only logic zeroes cause the voltage level to change — for logic ones, the voltage remains unchanged.

In general, a USB device has several FIFOs to receive data in. To the device address is added an endpoint address which indicates where the data should arrive or where it originates from. A USB mouse, for instance, always has an Endpoint 1 and an Endpoint 0. The latter is used during the initialisation. The internal microcontroller writes the actual data into the Endpoint-1 FIFO at regular intervals, from where they are transmitted to the PC.

The USB software forms so-called *Pipes* to individual Endpoints. A Pipe may be likened to a data channel consisting of a single wire. In actual fact, however, the data intended for a certain Pipe are transmitted in the form of data packets squeezed in a millisecond frame. Next, the hardware employs the Endpoint address to actually distribute the data to real memory devices. To achieve a higher data rate, a device may occupy several Pipes at a time.

Enumeration

A non-used USB connection is inactive, and the hub will not send data frames to it. The two datalines are then Low and represent an internal resistance of about 15 kΩ. Each USB peripheral device has an internal resistor of 1.5 kΩ connecting one of the datalines with +3.3 V. In a full-speed device, the D+ line is pulled up, in a low-speed device, the D– line. This information is used by the hub to first determine the type of peripheral it is talking to, and then set up the required data rate.

With the enumeration of a device, the suitable drivers are automatically loaded. Conversely, the system will also detect a device being disconnected from the bus. In that case, the driver is removed from memory. In this way, it is a simple matter to employ a single USB device with several different computers. All you have to do is disconnect the

Some useful addresses

http://www.usb.org: here, the makers of USB reside, that is, the representatives of several large companies that teamed up to define the USB standard. Their major contribution is the USB Specification, which is available as a downloadable pdf document. Essentially, this document contains everything. However, it is debatable if users should stick to just these sources for the information they require.

The Starter Kit from Cypress (http://www.cypress.com) for their USB microcontroller type CY7C63000 comes with an USB thermometer as an application example. Unfortunately, this relatively cheap kit is no longer available. The Elektor Electronics USB interface published in this issue was developed on the basis of this thermometer. So, if you missed your chance to obtain a Cypress USB kit, an excellent alternative is available.

Craig Peacock (http://www.beyondlogic.com) wrote his own driver for the USB thermometer. Craig supplies extremely useful technical information and documentation on building your own USB driver. His website on the USB is well worth a visit.

The starter kit supplied by Anchor-Chips (http://www.anchorchips.com) (now with Cypress, http://www.cypress.com) for the EZ-USB (AN2131) is based on an 8051-compatible processor with internal RAM and an USB core. Programs written for the 8051 may be loaded into the processor RAM via the USB interface, and started straight away. The USB core is so powerful that very few additional instructions are required to produce useful USB applications.

More information and hyperlinks may be found on the author's homepage http://www.home.t-online.de/home/B.Kainka

device from the first PC and insert the plug into the other one. On the second PC, a new enumeration procedure is started. In this way, a printer is easily used on two PCs.

The USB switch

For the serial (RS232) and parallel (Centronics) port, switches are available that obviate the need to plug and unplug cables when you want to use a different peripheral. Because only very few bus lines are involved, a similar switch for USB is easy to build. This switch should, however, mimic a USB cable being unplugged and plugged in again. Also, to make sure the peripheral is properly reset, sufficient time should be allowed to elapse before it is connected up to the bus. After all, the internal microcontroller has to be restarted to enable a new enumeration process to be initiated. From the above it should be clear why the USB switch has a neutral position in which the printer is not connected to either of the two PCs.

Another small detail should be observed. If you look closely at a USB plug, you will notice that pins 1 and 4 for the positive supply voltage and ground are a little longer than pins 2 and 3 for the datalines. When the cable is plugged in, the supply pins are connected first. Consequently, the USB peripheral is reliably powered before the datalines are connected. This system also reduces the risk of damage caused by static charges building up on peripheral devices, or equalisation current that may flow between devices if there is a problem with the earth connections. In brief, voltage surges caused by connecting devices can not cause damage to the electronics connected to the datalines. By the way, a parallel printer cable has an extended metal cover which touches the computer before the plug contacts are actually inserted.

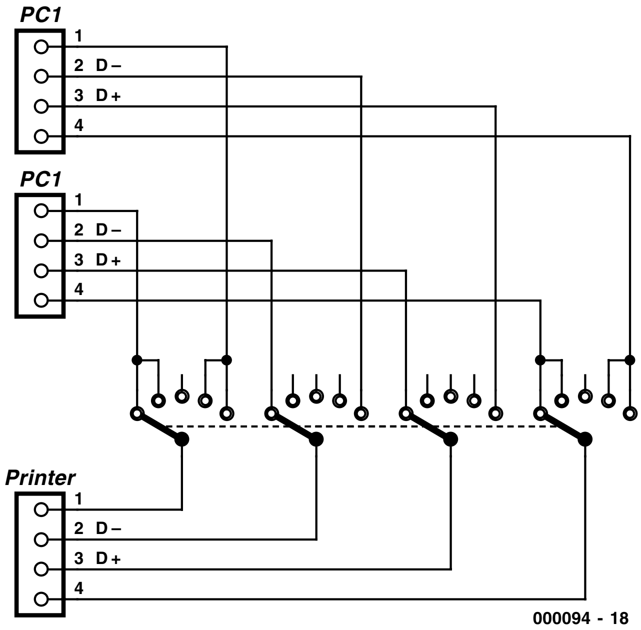


Figure 10. The USB printer switch.

Equalisation currents between USN devices could also be prevented by connecting all ground lines together in the USB switch. This, however, causes a large earth loop via the protective earth lines connected to the PC. Under adverse conditions, earth loops could lead to serious interference and reduced reliability of the entire computer system. Arguably, that should be avoided, hence all four USB lines are switched together in the switch box.

The special behaviour of the longer connecting pins may be mimicked by a switch with five contacts. When switching over to another USB device, the supply voltage is connected first, then the datalines D– and D+. The centre position is 'neutral'. Depending on the USB peripheral involved, a delay of one second may be required.

It should be noted that this is an experimental proposition which could not be extensively tested before publication of this article. In particular, the exact type of printer used will determine if fast or slow switching should be used, or if the delay with the centre position is at all required. Fortunately, it should not be too difficult to 'automate' the switch box using two relays and a small circuit.

(000094-1)

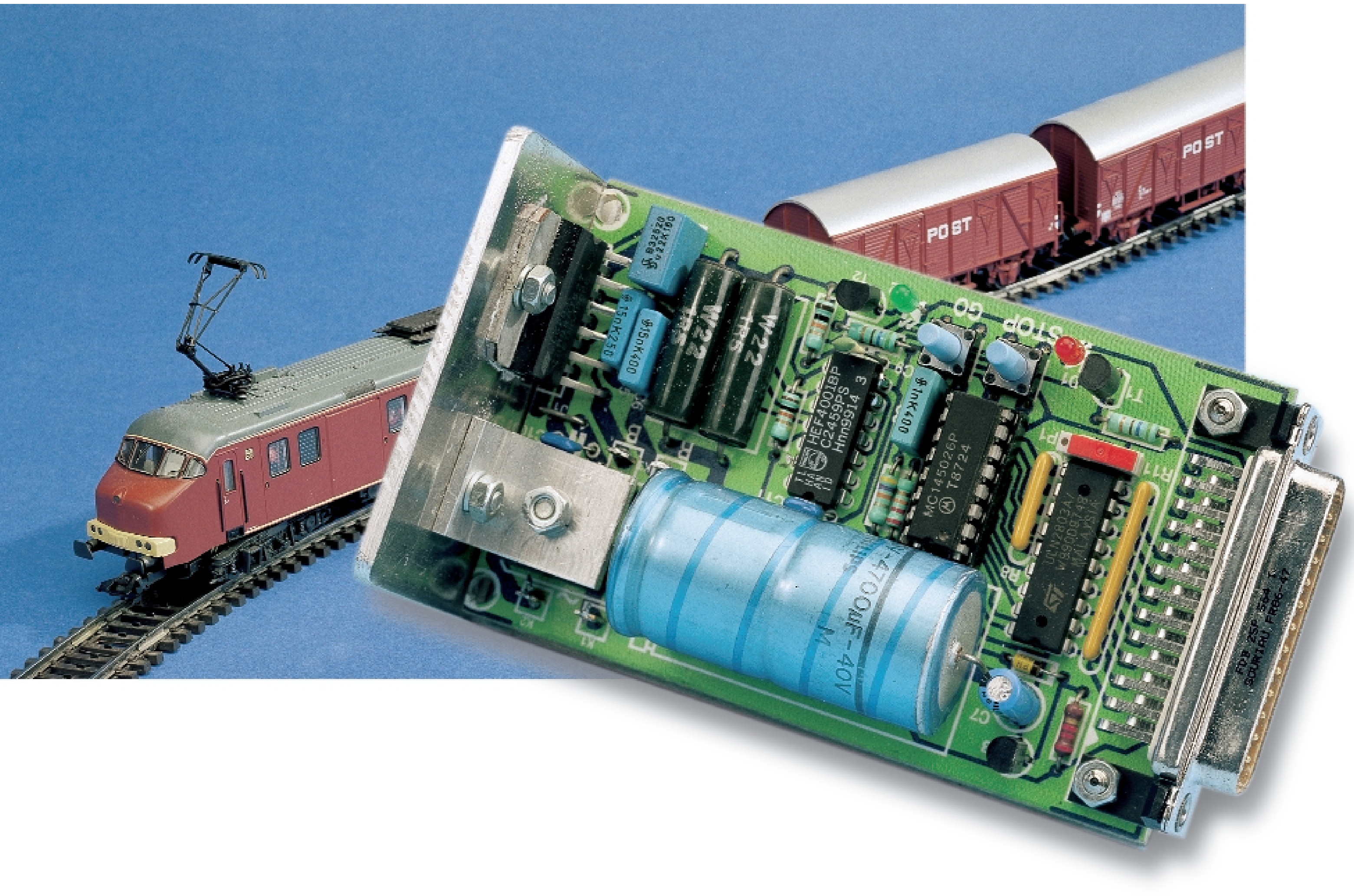
Continued next month

Märklin Digital Model Train Control (1)

A low-budget approach

From an idea by J. Schröder

This circuit provides an excellent opportunity to upgrade your Märklin model train system from ‘traditional AC’ to digital control.



Main Features

- Direct connection to PC parallel port.
- Simple to operate software (Windows 3.1x, 95, 96, NT) for individual control of up to 15 model trains.
- Controls Märklin Digital H0 stock using classic Motorola data format and Delta decoders.
- Integrated compact booster, max. 3.5 A output current, with overload protection.
- Powered by original Märklin transformer or single 15-VAC
- Manual on/off control of extra function.

Beginning model train enthusiasts may have more digital-ready (Märklin) locomotives moving about on the track than they are aware of. Usually, these locomotives are simply powered by the speed regulator on the main transformer. These days, model locomotives with an internal digital decoder are hardly more expensive than traditional types. That is not surprising because electronic circuits are easier and cheaper to produce in large volumes than any of the traditional reversing relays. Märklin always continued to produce decoders capable of working in 'AC' mode as well, allowing them to be used without problems with the famous Märklin transformer. Upgrading to all-digital control is then possible at a later stage. Possible, yes, but admittedly at a price because the cost of the upgrade will easily exceed that of all rolling stock.

Several attempts have been made to lower the threshold. From 1987 roughly to 1991, *Elektor Electronics* published items to create the all-home-made EEDTs (*Elektor Electronics Digital Train System*), a hugely successful series! Some time ago, Märklin introduced their Delta system, which is actually a stripped down version of the original Digital H0 with limited addressing options (4 instead of the usual 80). In fact, the Delta system triggered the author to design the circuit and software described in this article. Many Märklin locomotives come with a Delta decoder fitted as standard, instead of the traditional reversing relay. These locomotives, too, are controlled in old-fashioned AC mode (i.e., by transformer speed regulator), by the vast majority of model train fans. The Digital Control discussed in this article allows anyone and an old PC in the attic, and capable of handling a soldering iron, to get the feel of digital model train control at a very small outlay. So, if you do not yet have a locomotive with a Delta (or regular) decoder, you have a perfect excuse to step inside a model building shop when it's not December…

Recapping

Newcomers to the hobby can, of course, not be expected to know all the ins and outs of digital control for model trains. Hence, a brief recap is given of what we already described in the long series of articles on the EEDTs, plus a short description of the latest developments in the field.

The crux of digital model train control is that several trains on the track can be controlled independently. As opposed to traditionally operated tracks, where turning the speed governor on the transformer puts all rolling stock in motion at the same time, the digital track is marked by each locomotive having its own control element, allowing its speed and direction to be controlled individually. In addition to this function, there is often a plethora of options of the 'bells and whistles variety, including control of turnouts and signals via the track. Unfortunately, however, these extras and their control are outside the scope of this article.

How does it all work, you may wonder. With traditional control systems, a voltage is

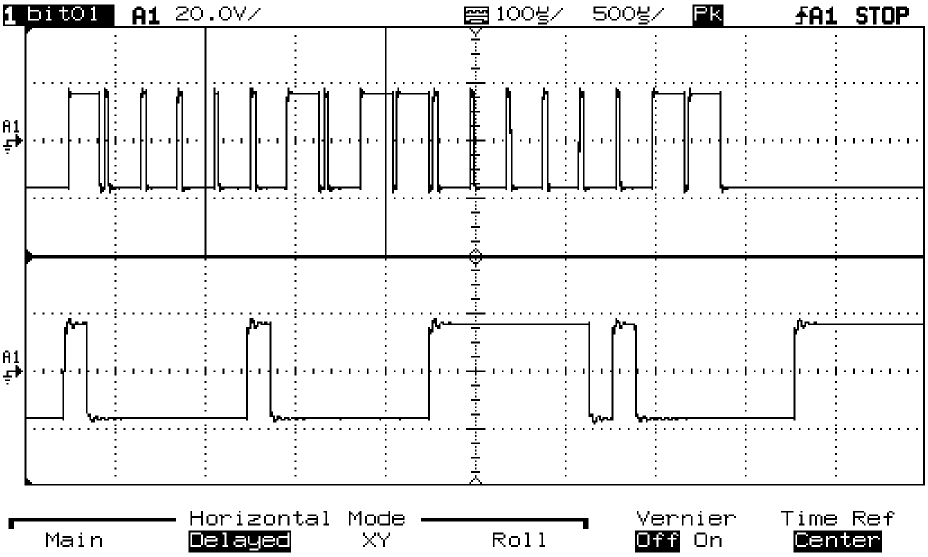


Figure 1. Two trinary signals on the 'scope: The top trace shows loco address 56 (X00X), function bit = 1, speed = 7 (0001).Below, in stretched-display mode, logic 0 (00), logic Open (10) and the start of a logic 1 (1…)

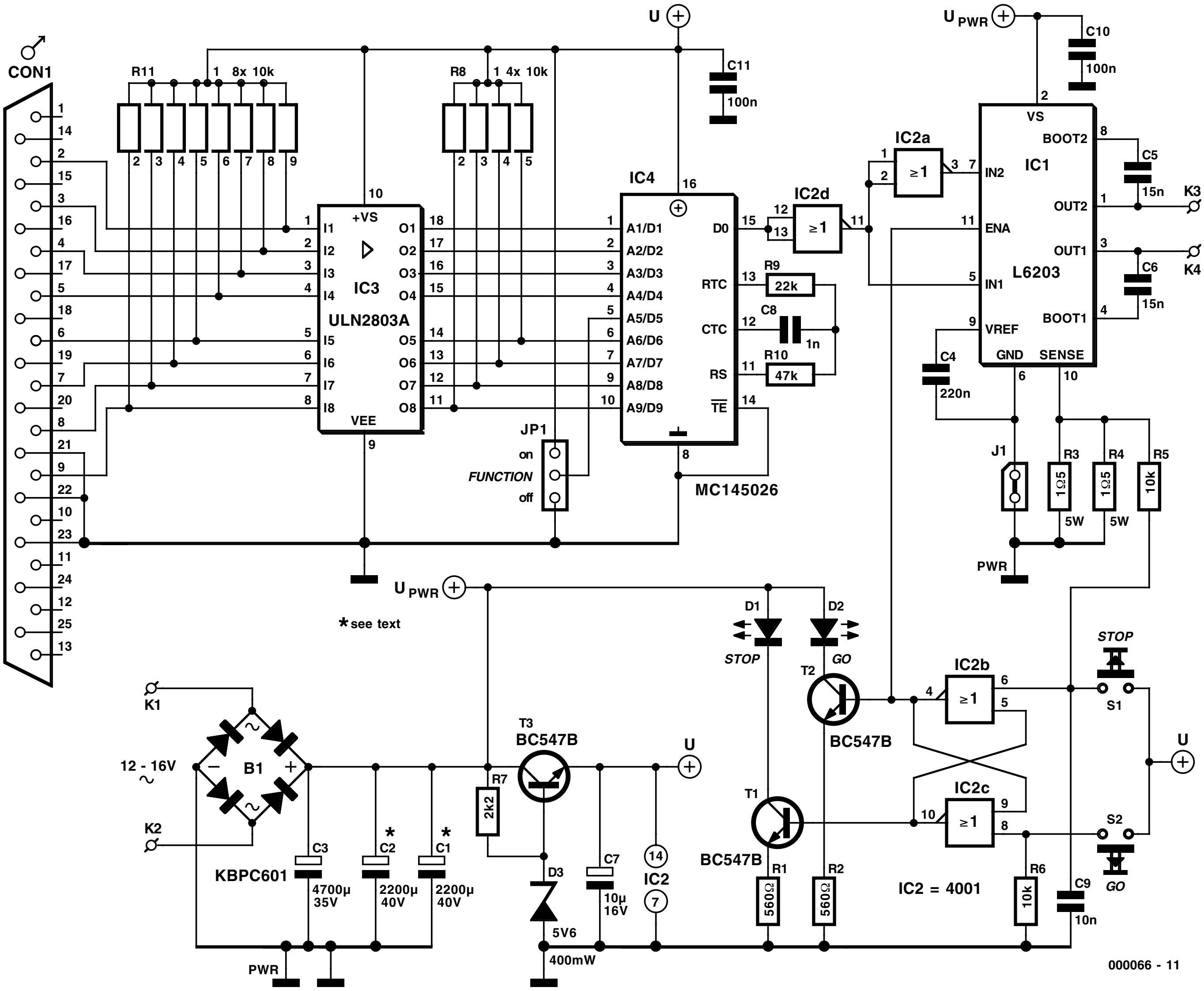


Figure 2. Circuit diagram of the Märklin Model Train Control System. The control elements, a mouse and your PC keyboard, are not shown here.

simply applied to the rails (alternating voltage in Märklin systems, direct voltage in most others), where overvoltage (Märklin) or polarity (other brands) provides information about the direction the locomotive has to travel in. With digitally controlled tracks, the rails carry a signal that alternates between a fixed positive and a fixed negative level. Depending on the model train brand and gauge, this voltage usually lies between ±12 V and ±18 V. The rate at which the voltage swings from + to – represents control information for individual locomotives and, if applicable, other devices like signals.

As with too many other products, the industry did not succeed in agreeing on a common standard in this field. Of the four digital systems originally available (Märklin Digital H0, Lenz, Fleischmann and Selectrics), only the first two actually got a foothold. In addition, Märklin Digital H0 is now flanked by several 'dialects' including the EEDTs data format and the 'New Motorola Dataformat'. Other brands, too, have variants. In this article, we will limit ourselves to Märklin Digital H0, because that is the format recognized by the system described here.

Märklin Digital H0 employs a switching sequence once designed by Motorola for use in remote controls. Information is bundled into packs of 18 pulses. In fact, these pulses are pairs of two pulses each. Three of the four combination possibilities of the pulse pairs are actually employed. 00 equals logic zero, 11 equals logic 1, and 10 is logic open. In the original Motorola data format, the combination 01 is not used — in the New Motorola Dataformat, you guessed it, it is.

A single data burst or packet consists of nine pulse pairs. The first four are used as (locomotive) addresses, supplying 3^4 = 81 addresses of which only 80 are used by Märklin. The remaining five pairs are only decoded in binary fashion: 00 or 11; with bit 5 flagging the on/off state of the extra function, and bits 6-9 containing speed and engine reversing commands.

Some time ago, Märklin introduced the so-called New Motorola' data format, in which all four combination options are allowed (00, 01, 10 and 11). The extra combinations in the function bit and the remaining four bits are used for non-volatile direction information and extra switching functions. The standard Motorola encoders and decoders, however, are unable to process these pulse pair combinations.

Finally, we should mention that a pause with a certain length is inserted between the 18-pulse packets. This is done to synchronize the transmitter and the receiver. The packet has length of about 3.8 ms, while the pause takes about 2 ms. As an extra safety measure, Motorola have built in a protocol that arranges for the receiver to be supplied with the same data packet twice in sequence for it to be recognized as valid. This protocol appears to be surprisingly effective for all rolling stock moving at considerable speed across the track.

The circuit

The circuit diagram shown in **Figure 2** is of an attractive simplicity. The core is formed by IC4, a Motorola encoder chip type MC14026 which looks after all converting into serial digital format of data received from the PC parallel port. In a way, the encoder IC also restricts the operation of the circuit: it is capable of generating the old (traditional) data format only. This allows standard and Delta locomotives to be controlled. Decoders from other brands (for example, Lenz) or decoders having the four extra function outputs utilizing the New Motorola data format can not be used in conjunction with this circuit.

Darlington array IC3 acts as an interface between the parallel port on the PC and the encoder chip. The first four outputs are used to set the locomotive address on the decoder. Because the open-collector outputs are not fitted with pull-up resistors, the status of the address lines is always Low or High-Z. The ability to set a High-Z status is essential because the Delta addresses defined by Märklin all have 'logic open' bits. If the encoder were connected directly to the parallel port, it would not have been possible to activate any Delta decoder at all!

The second nibble of the parallel port is used for setting bits 6-9 on the encoder. These bits contain speed information and the reversing command: the bit combination is 1000.

Bit 5, the function bit, is given a fixed state with the aid of a jumper or switch (JP1) and is therefore on or off for all locomotives to be addressed. That should not be a problem because this bit usually controls the lighting function, which is preferably on by default. Only the oldest EEDTs decoder employs bit 5 for (non-volatile) direction information. Hence that decoder can not be used with this circuit, because changing the state of JP1 would cause all trains to reverse. Later EEDTs loco decoder variants, including the most recent EEDTs Pro, are fully compatible with the present system.

Components R9, R10 and C8 determine the timing of the encoder. Resistor R9 determines the length (duration) of one packet of 18 pulses (3.8 to 4 ms), while R10 takes care of the synchronisation pauses between pulse packets.

Originally, the circuit was designed for direct connection to the EEDTs Booster. However, the Booster, with its 10-amp output current capacity and considerable cost and effort of building may be too much of a good thing, and beyond the reach of beginners. That is why the present circuit comes with its own mini booster, IC1, which is short-circuit as well as overload resistant. To keep the cost of building the project down to the absolute minimum, it is possible to connect the existing Märklin transformer and use it as a power supply. Using B1 and C1/C2 (or C3, see construction details), a single direct voltage is derived from the transformer's secondary voltage. A full bridge output stage has to be chosen to enable a single-rail input voltage to be turned into an output voltage that switches between a positive and a negative value. The L6203 from ST Microelectronics combines the required functions in a single IC. Plus, because the IC is manufactured in DMOS technology, we need not concern ourselves too much with switching speeds or power dissipation. At the maximum output current of 3.5 A chosen for this circuit, the L6203 remains reasonably cool. With insufficient cooling, an internal overheating protection arranges for the IC to be switched off once a certain temperature limit is exceeded. Components C5 and C6 are so-called 'bootstrap capacitors' which serve to ensure a sufficiently high gate voltage on the two power MOSFETs in the upper section of the bridge.

The output current flows to ground via the sense connection and R3-R4. These resistors serve to monitor the maximum output current because the L6203 is not wholly and truly short-circuit proof. The voltage developed across R3 and R4 is fed to the input of NOR gate IC2b via R5-C9, a low-pass filter to suppress inevitable switching pulses. Together with IC2c, the NOR gate forms a bistable with a special 'treat' in that IC2b is (mis-)used as an analogue comparator. Standard CMOS circuits are designed to switch at about half the supply voltage. If the voltage at pin 6 of IC2b reaches 2.5 V (which happens at 2.5 V / 0.75 Ω or 3.5 A), IC2b and IC2c toggle.

The enable input of the L6203 is then pulled low, the bridge is switched off, and the tracks are disconnected from the supply. The green GO LED, D2, also goes out and its red counterpart, D1, marked STOP, lights. When this happens, switch S2 may be used to rerestart the circuit. A stop condition may be forced by operating switch S1.

The two remaining gates in IC2 are used to supply the bridge with the normal and inverted digital signal.

The 5-V logic supply voltage is derived from the unregulated power supply. A zener diode is used in combination with emitter follower T3 acting as a power buffer, because the input voltage conditions are rather uncertain. If, for example, a Märklin transformer is connected and the speed control knob is turned back to the train reversing position, an alternating input voltage of 24-30 V appears at the supply inputs. When rectified, that would produce an input voltage surge that is sure to endanger the life of a 7805 voltage regulator. With D3 and T3 included, the circuit will withstand this abnormal condition. The output bridge can also safely handle this voltage surge of up to 52 volts.

(000066-1)

Next month's second and final instalment will cover the system software and hardware construction.

GSM Call Indicators

'Quiet-Calling' add-ons for mobile phones

By H. Steeman

Those small indicators that either flash or vibrate when there is a mobile phone call are well known by now. But how can they monitor the activity of a GSM phone remotely, and what is inside them? We've looked more closely at some of these gadgets.

These days GSM phones are part of everyday life. Within a period of five years, more mobile phones have been connected in most Western European countries than there are fixed connections. By contrast, it took about a 100 years for fixed phones to reach their current number. The introduction of the mobile phone has affected society greatly. More and more people are becoming irritated that they can be contacted on their mobile. There are numerous occasions when a mobile phone rings loudly when it is least wanted. Sitting peacefully on a terrace seems to have become more of a utopia nowadays.

Passive and yet active

GSM phones are compact transmitters/receivers (transceivers), just like other mobile phones and carphones. The transmitter in the handset uses a frequency of 900 or 1800 MHz and has a maximum power of 1 or 2 watts which allows it to make a connection to the GSM network. This RF power is sufficient to cover more than 20 miles, although the actually available distance will usually be much smaller. In-car mobile phones having an RF booster capable of putting out up to eight watts. Because the distance between the transmitter and receiver is usually much smaller than the maximum theoretical range, the transmission power is dynamically reduced to a level which is appropriate to the circumstances. This causes a reduction in RF power and is less of a drain on the battery.

Extending battery life

The transmitter takes its power from the battery in the phone. Obviously, the battery will be quickly drained at at a continous transmission power of 1 or 2 watts. Under these circumstances, a standby period of several days wouldn't be possible. That's why mobile phones are usually in a standby mode. The transmitter is then switched off, the control functions are idle (the backlights of the keypad and display are switched off), and only a low-power receiver searches for incoming control signals. These control signals use a fixed channel in the RF spectrum of the GSM network; a special time slot has been reserved for them. The control channel is used to send signalling information to and from the mobiles which are within range of the transmitter (BTS or Base Transceiver Station).

When to transmit

The network has to keep track of the location of the mobiles. Each mobile sends a message to the network at regular intervals, usually once every fifteen minutes, sometimes a bit longer. When a mobile is moving, for example during a car or train journey, it has to give a 'location update' each time it moves into a new cell. Again this causes the mobile to send a message to the GSM network. Whilst sending such a message a strong RF field is generated by the transmitter. Motorists will recognise when this happens by the crackling and buzzing sounds coming from the car radio. This signal can also cause interference in other audio equipment, which is in the vicinity of the mobile. The transmitter of a mobile is also active when confirming the reception of an SMS message, or sending one, and, of course, when making a call!

Transmitter LED

Keeping the previous theory in mind, it becomes clear how GSM indicators work. **Figure 1** shows an aerial that is sold for use as an indicator for several Nokia phones. The circuit consists of a small coil (similar to the spring found in ballpoint pens) which is used as the antenna. An LED and a diode are connected to a tap on the coil. During transmissions, a voltage is induced across the coil, which is sufficient to make the LED light up. The extra diode protects the LED against too high a voltage. So whenever the transmitter is on, the LED will light up.

Active detectors

Apart from passive indicators, which take their energy from the mobile, there are also active ones. **Figure 2** shows the circuit of a GSM indicator pen. Standalone vibration alarms also make use of a similar circuit. A small aerial picks up the RF signal and causes an LED to flash, or activates a vibrator (usually a small motor with an unbalanced weight attached to its spindle). A pair of button cells normally provides power to the circuit.

Active or passive, the indicators come on when the mobile sends control signals to the network, when registering with the network, when sending or receiving an SMS message, and during a call. With the transmitter off there will, of course, not be an RF field and the indicators will be idle. Of course, when using a call indicator, you will not forget to switch the GSM to 'quiet' (or 'meeting') mode, will you? Thank you!

(000105-1)

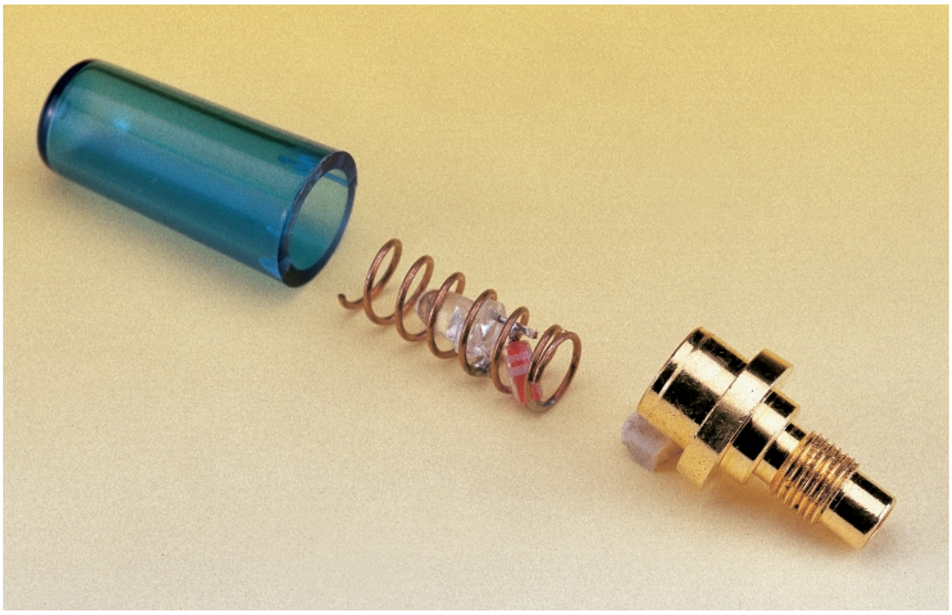


Figure 1. An indicator aerial is available for some Nokia handsets. An LED indicates when the transmitter is active.

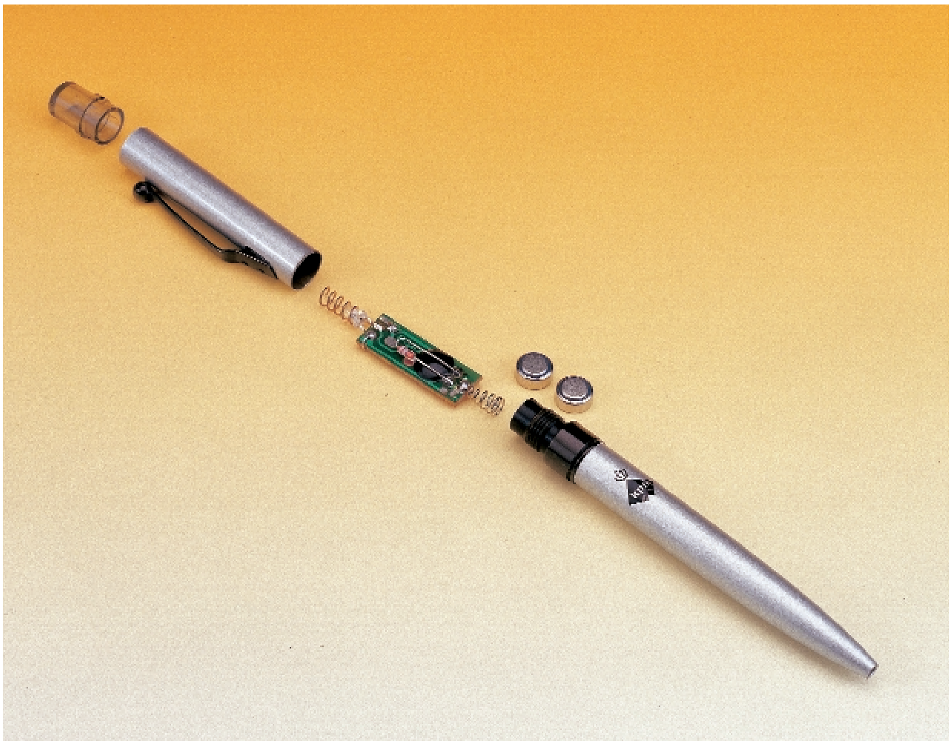


Figure 2. Pens with GSM indicators or standalone vibration alarms use an active detector.

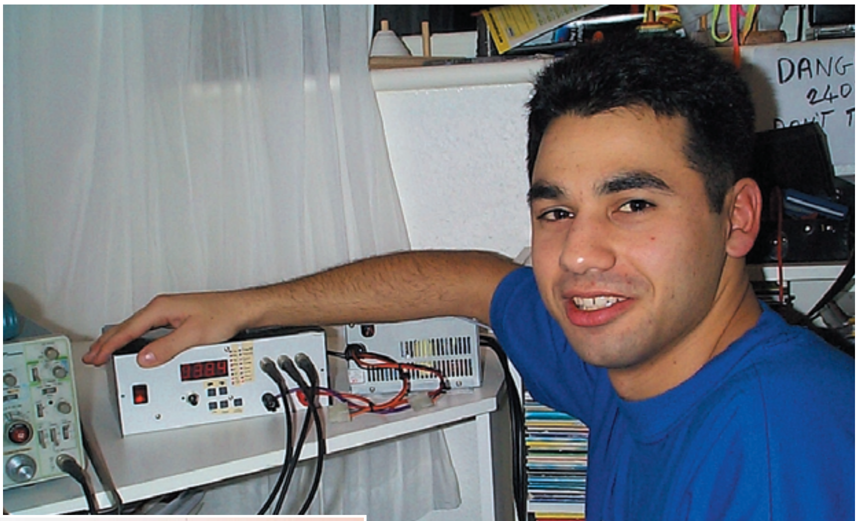
COP8 Design Competition Results

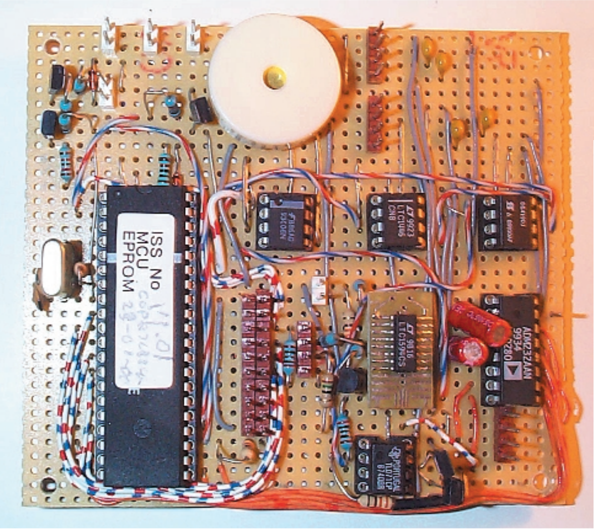
Prize line-up

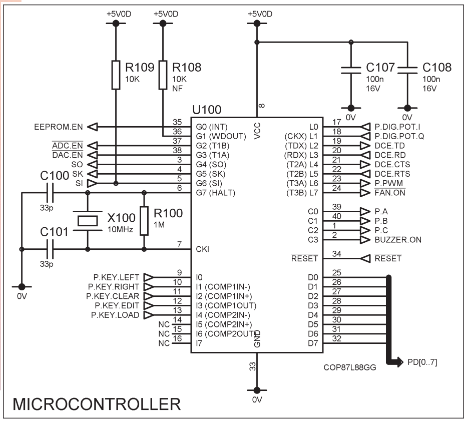
By Guy Raedersdorf

Processing, evaluating and marking the 25 COP8 Design Competition entries received from all four corners of Europe took the Jury members longer than expected. In line with the Competition rules, National Semiconductor put one of their design engineers, Mr. Siegfried Rueth, to our disposal to assist in judging the technical level of the entries.

After a first pre-evaluation round, the Jury allowed 15 or so COP8-based project proposals to pass to the second round for even closer scrutiny. After long hours of marking the projects, debate and further thought the Jury members finally agreed on the laureates. The winner of the **First Prize**, 1,000 US$ in cash offered by National Semiconductor, is **Mario Nicou** from the UK, for his project **Dynamic PSU Load Tester**. As indicated by the project name, Mario set out to design an instrument for testing the dynamic (active load) performance of all kinds of power supplies. The Jury was particularly impressed with the quality of Mario's project, the way it







was presented, and the technical approach to it. The 3,000 lines of COP8 code Mario wrote for the project were recognized as 'exemplary' by NS specialists. The illustrations show a few impressions of Mario's workshop.

The **Second Prize**, 500 US$, was awarded to **Robert Lacoste** from France for his futuristic **SCAN3D** project. It is probably best described as an electronic surveying system employing ultrasonic sound signals to measure the distance to an object, or calculate surface areas. SCAN3D is capable of working in stand-alone mode as well as connected up to a PC. This particular project stood out because of its beautifully designed PCB and PC/user interface.

The winner of the **Third Prize**, 250 US$, is **Vladimir Mitrovic** from Croatia for his Competition entry called **Controller for Graphic LCD Module.** Admittedly the Jury was enticed a bit by the simplicity of the project and the efficient way in which a COP8 is employed in a classic application. In this project, the COP8 controls an LCD module with 32 lines of 122 pixels, with each pixel being individually addressable (whether active or not active).

It did not take the Jury long to decide to award lesser prizes to 'runners-up' in the Contest, as follows:

4. Rob Reilink (Netherlands) for his 'Universal Control Unit'.
Prize: a BE-6-II (Pentium III) motherboard from Abit

5. Arnoud Varoquaux (France) for his 'Laboratory Power Supply'.
Prize; a PW65 (Pentium III) motherboard from DFI

6. Olivier Bureaud (France) for his 'Housekeeper Control'.
Prize: WB6 (Pentium III) motherboard from Abit

7. Paul Gélineau (France) for his 'LCD VU Meter'.
Prize: CA61 (Celeron/Pentium FPGA) motherboard from DFI

8. Michel Vacher (France) for his 'Ni/Mh Battery Charger'
Prize: 4 CD-ROMs of his choice

9. Claude Pinedre for his 'NAVTEX Receiver'
Prize: 3 CD-ROMs if his choice

10. Michael Gaus (Germany) for his 'Message Display'.
Prize: 2 CD-ROMs of his choice

The results of this COP8 Design Competition, realized with hardware and financial support from National Semiconductor Germany, are such that it may be repeated in some form or another in the future. The Jury was surprised to note, on the one hand, the large number of participants from France, and on the other, unexpected low participation levels from Germany, a country reputed for being the European 'citadel' for microcontroller-based electronic projects.

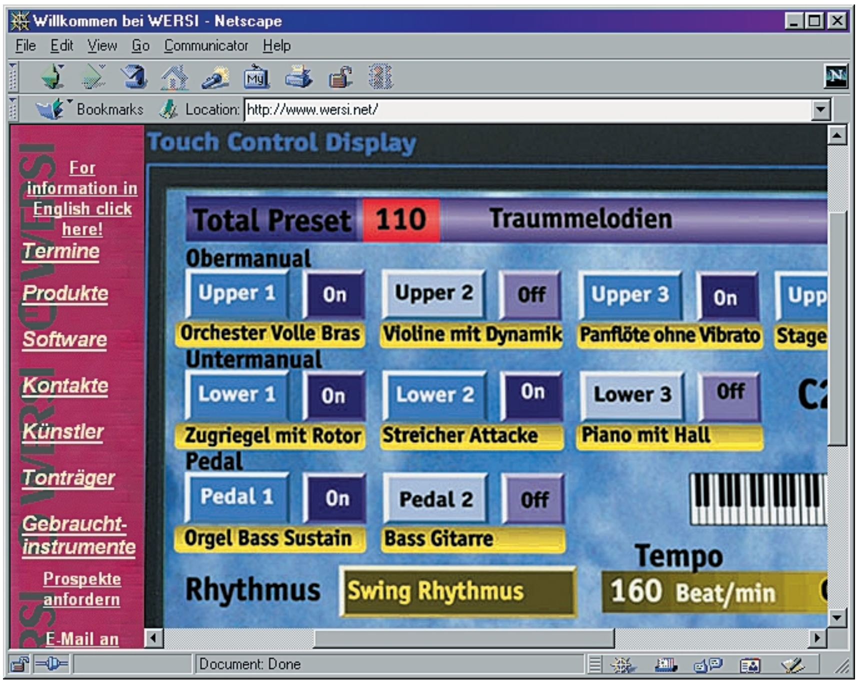
(000073-1)

Web Connected Organs

By Peter Coenders

Today, even electronic organs make use of the Internet. Usually this involves PC downloaded software upgrades. However, Wersi have gone a step further and have provided a new generation of organs with built-in Internet connectivity.

One of the best known suppliers of do-it-yourself electronic organs is the German firm Wersi. Particularly during the '70s and early '80s, Wersi's were very beautiful, and to a certain extend, affordable analogue organs. Models such as Helios and Galaxis have made a name for themselves. Around 1985 the first digital Wersi's became available. As a consequence of continuing miniaturisation of the electronics, the do-it-yourself aspect fell out of favour. In the early '90s Wersi introduced the CD series, with the Spectra and Atlantis as the top of the line models. This series made use of 16 bit sampling which ensured natural sound. One of the disadvantages however, was the limited polyphony. The Spectra with two keyboards and pedalboard provided a maximum of 32 voices (excluding rhythm), while the Atlantis with three keyboards plus pedalboard was capable of a maximum of 56 voices. At the time, this was not too bad, but was still rather limited when compared with analogue organs.



Total Preset 110 Traummelodien

Obermanual

Upper 1 On — Orchester Volle Bras

Upper 2 Off — Violine mit Dynamik

Upper 3 On — Panflöte ohne Vibrato

Upp — Stage

Untermanual

Lower 1 On — Zugriegel mit Rotor

Lower 2 On — Streicher Attacke

Lower 3 Off — Piano mit Hall

C2

Pedal

Pedal 1 On — Orgel Bass Sustain

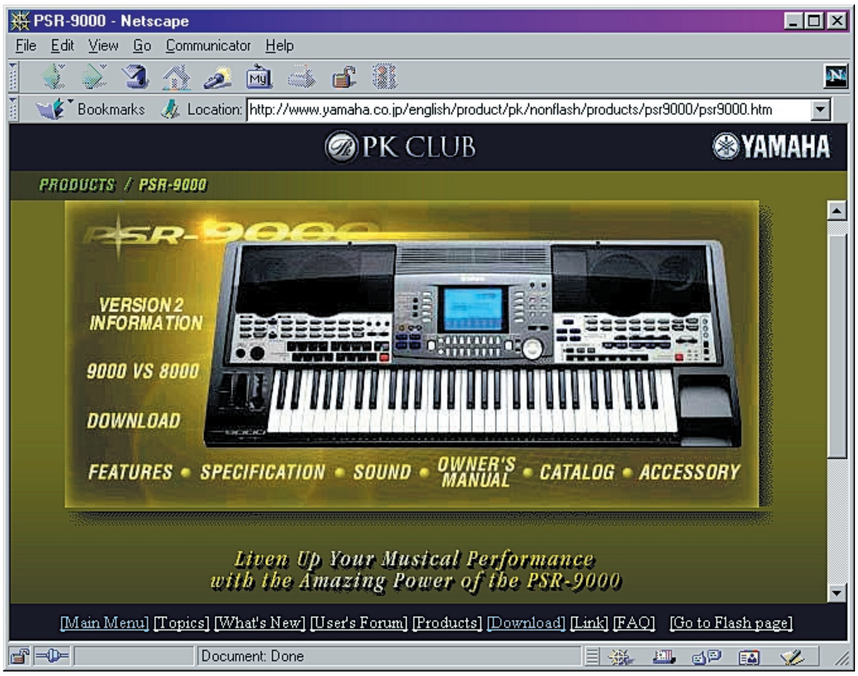
Pedal 2 Off — Bass Gitarre

Rhythmus — Swing Rhythmus

Tempo — 160 Beat/min

Up to last year, Wersi stuck to the CD theme, with beautiful instruments such as PhonX and Pegasus. Relatively unique, even among the most recent instruments (including keyboards and expanders), is the large LCD touch screen that makes for simple and convenient operation. When compared with magnificent instruments such as the analogue Galaxis, the number of buttons, touch switches and drawbars is greatly reduced, without reducing any of the capabilities. Also worth mentioning is that Wersi have addressed the problem of limited polyphony. The most recent versions of the Pegasus are 64 note polyphonic.

Earlier this year, Wersi unveiled the latest series of instruments. Based on modern multimedia computer technology, supplemented with Wersi's own knowledge of musical instruments, a very remarkable series of instruments has been created. It appears that the limitations of the CD series are gone. All organs and even the keyboard are 208 note polyphonic. These instruments also have real-time registration, something that has always been customary on analogue instruments. The new series consists of the models Abacus (keyboard), Vegas (portable organ), Monet, Da Vinci ('living room' organs) and the


PSR-9000 - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://www.yamaha.co.jp/english/product/pk/nonflash/products/psr9000/psr9000.htm
PK CLUB
YAMAHA
PRODUCTS / PSR-9000
PSR-9000
VERSION 2 INFORMATION
9000 VS 8000
DOWNLOAD
FEATURES • SPECIFICATION • SOUND • OWNER'S MANUAL • CATALOG • ACCESSORY
Liven Up Your Musical Performance
with the Amazing Power of the PSR-9000
[Main Menu] [Topics] [What's New] [User's Forum] [Products] [Download] [Link] [FAQ] [Go to Flash page]
Document: Done

top of the line Louvre with three keyboards and 25 note pedalboard. This product series is conspicuous because of the large LCD colour touch screen. The keyboard is remarkable in the sense that it possesses real drawbars, following the example set by Hammond. Finally, Wersi have gone even further and have made it possible to provide each instrument from this series with a web connection. Using this, Wersi take a new step in the direction of sound and rhythm upgrades and possibly a certain amount of technical service. Reputable brands such as Yamaha and Roland have a similar approach, of course. These companies make their upgrades available in a database, accessible with a PC connected to the Internet. The file then needs to be copied from the PC to the instrument using a disk.

Even though these instruments have been displayed to the public on several occasions, they are not yet available. The expectation is that production will start soon.

Those readers who would like more information or a musical foretaste (from Yamaha) should visit the following web sites:

http://www.wersi.net
http://www.yamaha.co.uk/
http://www.yamaha.co.jp/english/product/pk/nonflash/products/psr9000/psr9000.htm
http://www.rolandcorp.com/

(005077)

Valve Preamplifier (2a)

Part 2a: Construction (continued) and test data

By G. Haas

Once you've stuffed all the boards that go into the design, the Valve Preamplifier is ready for assembling and wiring. The results (see pictures and performance data) are pretty convincing.

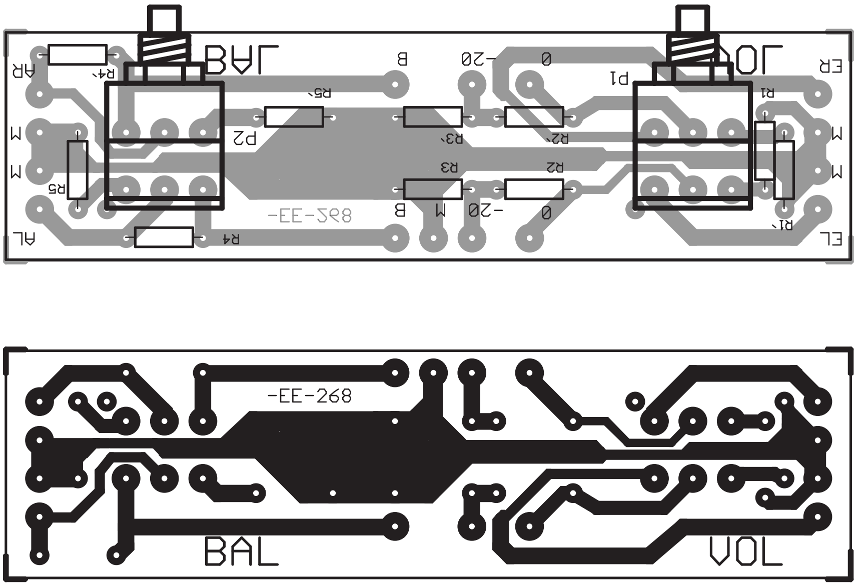
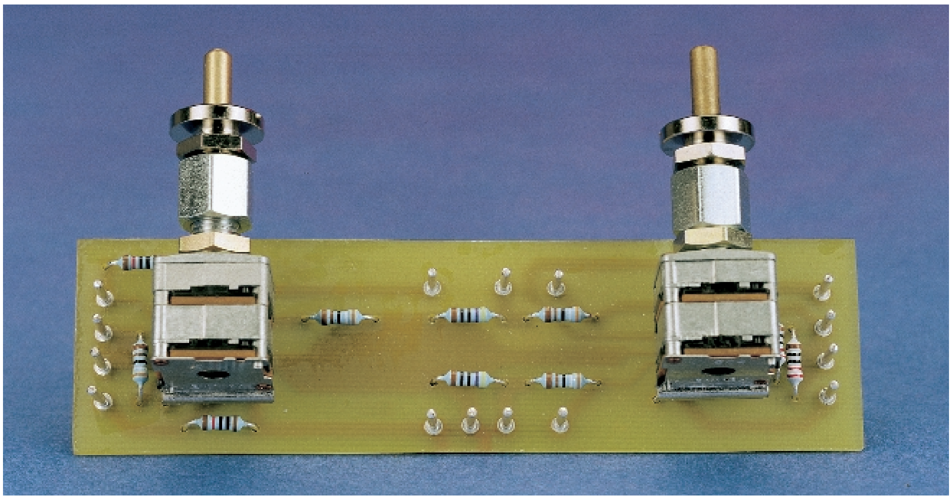


Figure 2. The small printed circuit board for the volume and balance controls.



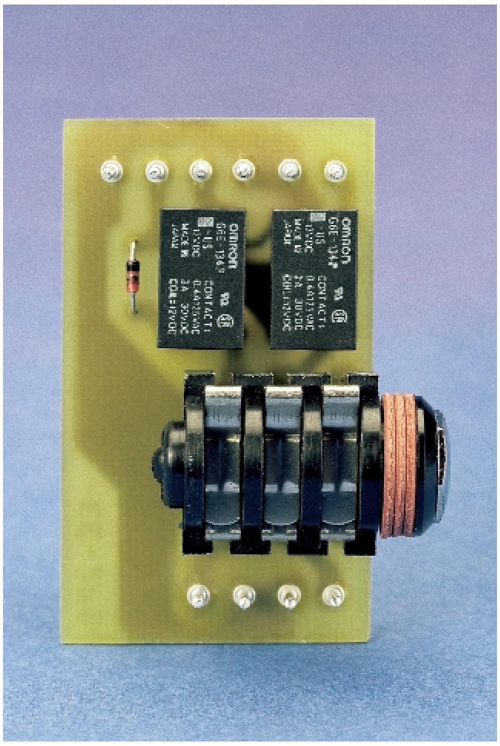
Next comes the volume control board (see **Figure 2**). The photograph shows how it is built. There is nothing difficult about mounting the components, and the board can be solidly attached to the front panel via the potentiometers, using suitable hardware.

The output relays and the headphone socket are mounted on the small circuit board shown in **Figure 5**. The socket is also used to fix the board to the chassis, so no additional fitting hardware is needed. You must pay careful attention to where the signal from the amplifier comes from and where is it goes to afterwards. The headphone socket is protected only if the wiring is done properly. In addition, make sure that

COMPONENTS LIST
Volume Control

Resistors:
R1,R1´ = 100Ω
R2,R2´,R4,R4´ = 3kΩ9
R3,R3´ = 470Ω
R5,R5´ = 10kΩ
P1 = 10kΩ stereo potentiometer, linear, tracking
P2 = 10kΩ stereo potentiometer, linear, tracking
2 holders for potentiometers

Miscellaneous:
S1 = rocker switch, 2 poles, 3 contacts
Solder pins



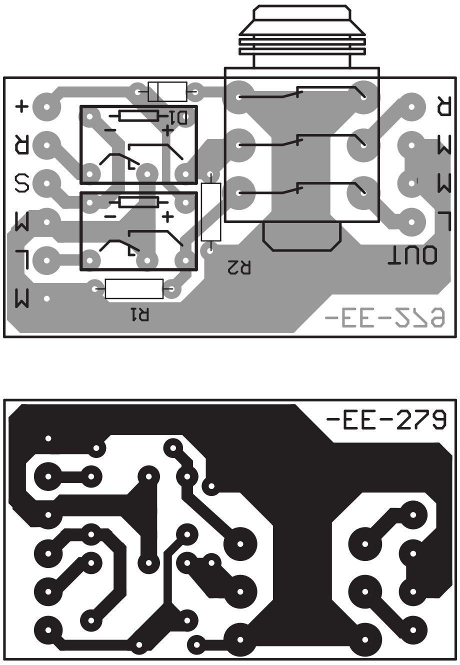


Figure 5. The small printed circuit board for the output relays and headphone socket.

COMPONENTS LIST
Headphones/Line Switch

K1 = stereo headphones socket, PCB mount
Re1,Re2 = relay, 1 changeover contact, 12V coil (Omron G6E)
R1,R2 = 680kΩ

COMPONENTS LIST
High Voltage Supply

Resistors:
R1 = 10kΩ, 2W
R2 = 100Ω, 4.5W
R3,R4 = 1kΩ
R5 = 6Ω8, 2W
R6 = 150kΩ, 2W

Capacitors:
C1 = 100μF 400V, raster 10mm
C2,C3 = 22μF 400V, raster 7.5mm

Semiconductors:
T1 = BUZ92
T2 = BC546
D1,D2,D3 = zener diode 110V, 1.3W
D4 = zener diode 18V, 1.3W
GI1 = B500C1500, rectangular case (500V piv, 1.5A peak)

Miscellaneous:
Heatsink for T1: SK 68/50 (Fischer, Dau Components)
Isolation and mounting material for T1 and heatsink
Solder pins

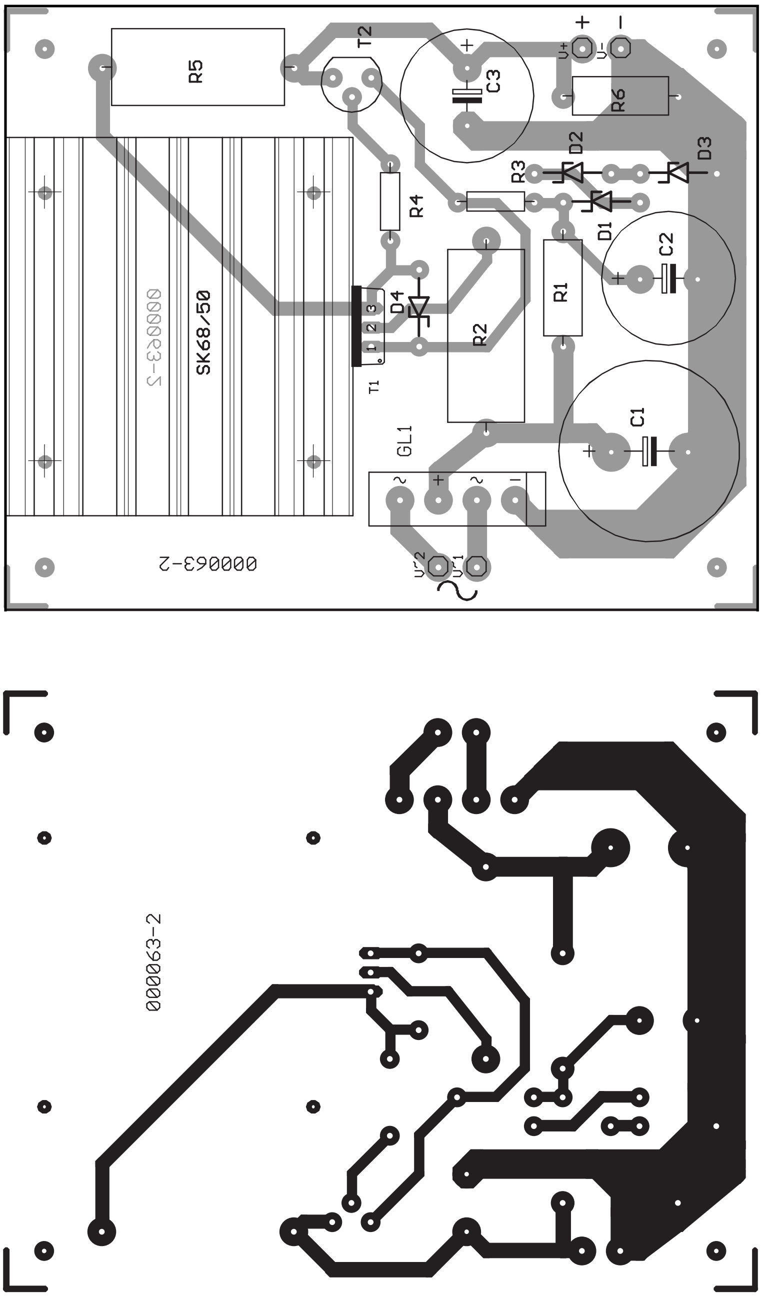


Figure 7. the high-voltage power supply and …

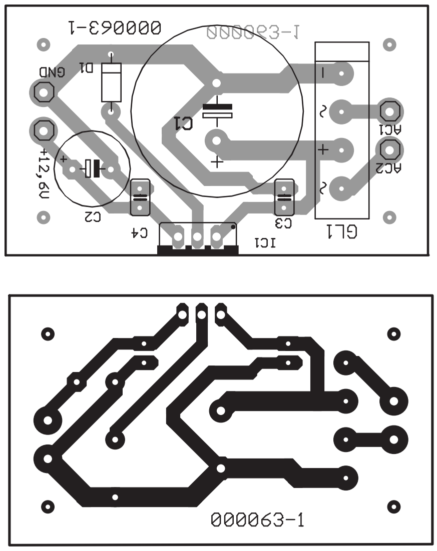


Figure 8. the low-voltage power supply.

COMPONENTS LIST
Low Voltage Supply

Capacitors:
C1 = 2200µF 40V, raster 7.5mm
C2 = 100µF 40V, raster 5mm
C3,C4 = 100nF, ceramic, raster 5mm

Semiconductors:
Gl1 = B80C1500, rectangular case (80V piv, 1.5A peak)
D1 = 1N4148
IC1 = 7812

Miscellaneous:
Isolation material for IC1
Si1,Si2 = fuse, 0.2 A, slow, with PCB mount holder
Solder pins

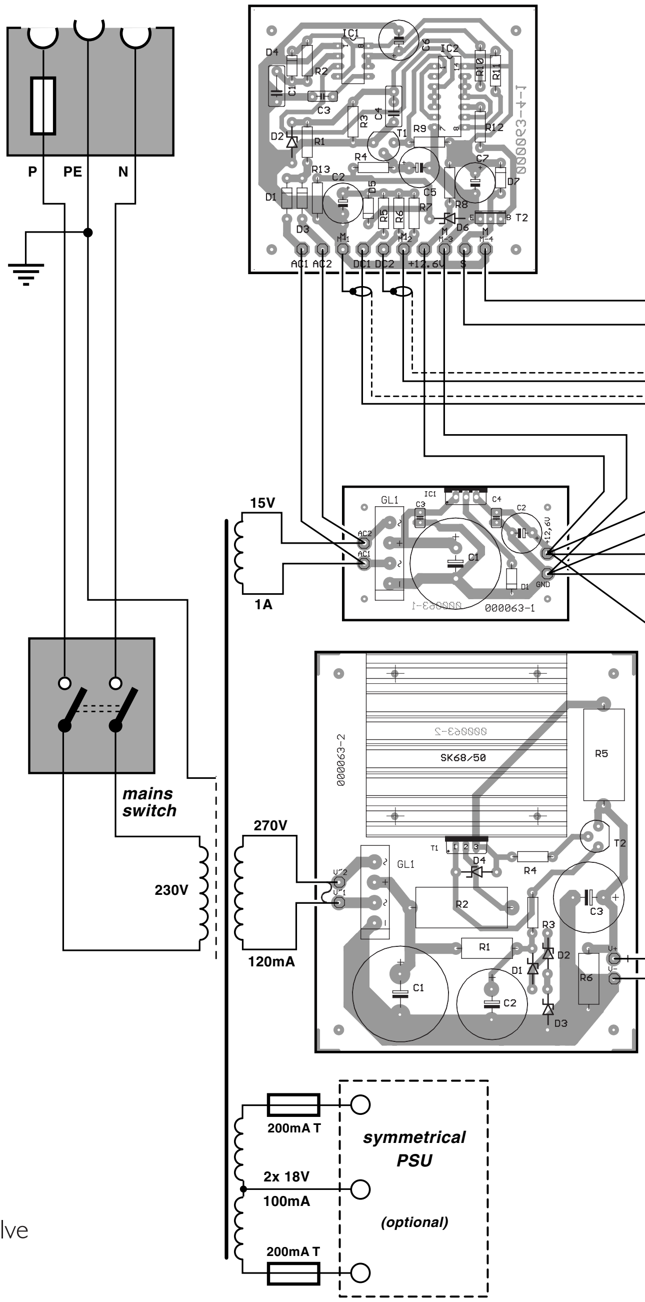


Figure 9. The wiring of the valve preamplifier is complicated.

the relays are correctly polarised, as shown on the component overlay. The selected relays are fully enclosed, which protects the contacts from contamination. In addition, the contacts are gold plated and rated for 250-V operation.

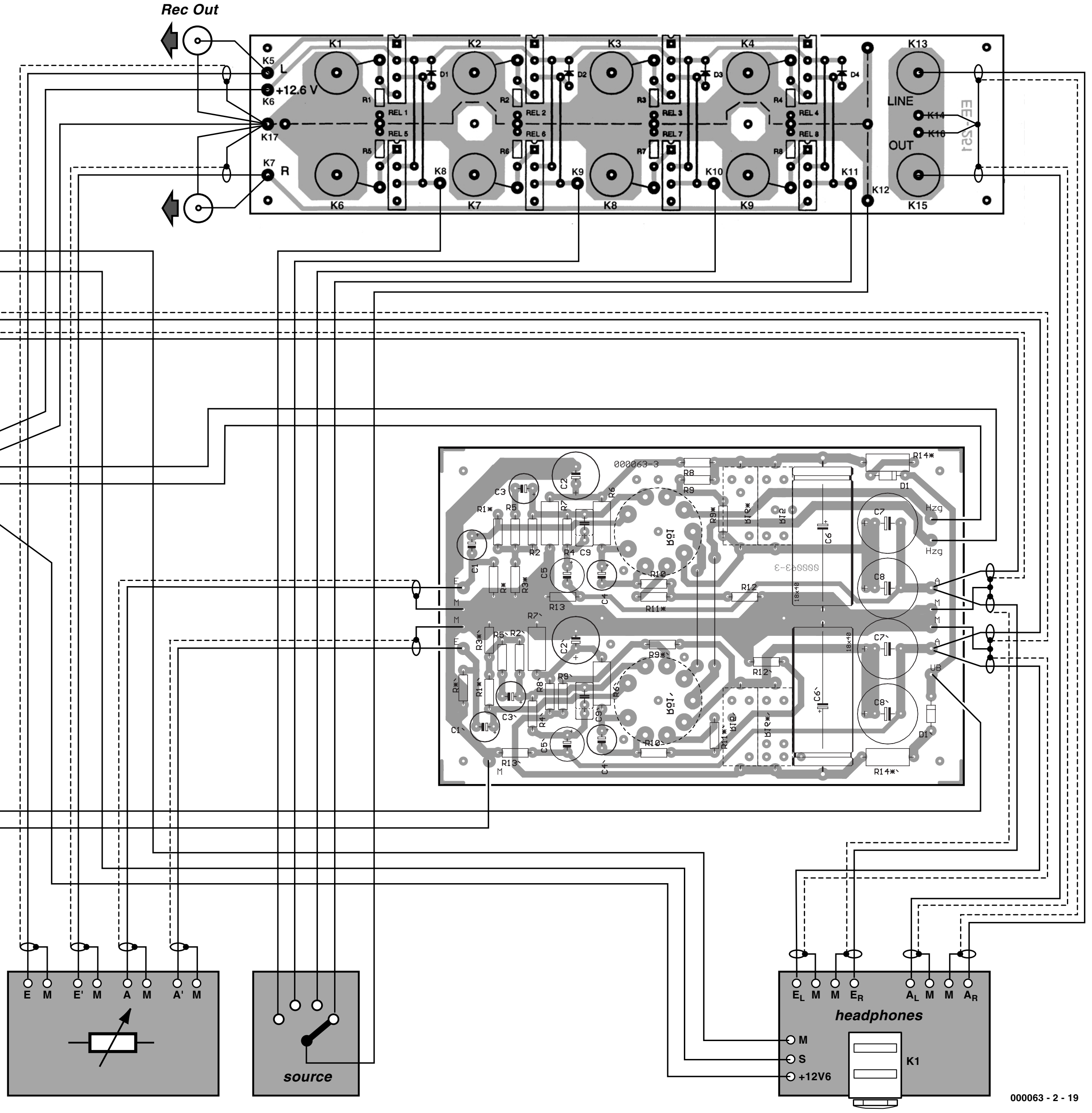
There is no circuit board available for the optional symmetric power supply. However, the rich store of *Elektor Electronics* projects should provide adequate possibilities.

Now that you have finished building all the circuit boards, they can be fitted into the enclosure. If you use the original enclosure, as described above, all of the necessary holes are pre-drilled, so all you have to do is to fit the hardware for the individual parts (including the mains cable input socket, the power switch, the potentiometers and the recorder sockets). If you prepare your own enclosure, take care to arrange the individual items in a favourable manner and ensure that the circuit boards are solidly attached. Don't fit the input module right next to the mains transformer, since otherwise you can be sure of introducing mains hum into the audio signals.

Internal wiring

Installing the wiring that interconnects the various circuit boards is probably the most complicated task of all. Consequently, we have prepared the wiring diagram shown in **Figure 9**. You should photocopy this, and mark each connection on the copied diagram once you have made the connection. This manner of working has proven to be the most foolproof, including in the *Elektor Electronics* labs.

All leads that carry audio signals must be made using good-quality screened cable. Ordinary light-duly multi-strand wire can be used for the remaining leads, which consist of the signal source selector switch wiring, the protection circuit wiring and the power supply wiring. The wire should have a diameter of 1 mm and good insulation. Place the signal wiring and the other wiring so that the two types are separated from



each other as far as possible, and fix the cables to the chassis using clips. Connect the recorder output sockets to the common busses for the other circuit boards using wire bridges.

The moment of truth

When you have finished the wiring and everything has been checked and rechecked, you are ready to try the first functional test. First, break the connection between the mains transformer and the high-voltage circuit board (and insulate it against accidental contact). The filament supply must measure 12.6 V as soon as it is switched on, and it must be short-circuit proof. This voltage may vary by up to ±5 percent. The valve filaments must glow visibly after around one to two minutes. Later on, the filaments will be the 'pilot light' for the preamplifier. However, if you want an additional, more distinct power-on indicator, you can simply connect an LED in series with a resistor and 1N4007 diode to the 15-V winding of the mains transformer. You can also right away check the operation of the relays.

After this initial test, you can activate the high voltage. It should reach its nominal value shortly after being switched on. If nothing smokes, check the expected values noted on the schematic diagram. After this, use a sine-wave generator and oscilloscope to check the audio paths and functions. Once this test has been successfully completed, close up the enclosure, connect the preamplifier to signal sources and a final amplifier, and switch everything on. A valve cathode must warm up for two to five minutes before it can emit enough electrons for the valve to be operational. However, you will have to wait around ten to twenty minutes for the valve to be thoroughly warmed up, before you will hear the 'right' sound. After this, all you have to do is relax and enjoy what you hear!

(000063-2a)

Measured performance

We made our own measurements of the performance of the valve preamplifier in the Elektor laboratories. Naturally, we would like to share the results with you. The ‘raw’ numbers are listed in the table. All measurements were made after a four-hour warm-up interval, with an effective input voltage of 1 V and an effective output voltage of 2 V. The outputs were terminated in 10 kΩ, and free inputs were terminated in 600 Ω. The balance potentiometer was in the middle position.

Signal to noise ratio	A-weighted	100 dBA
	linear, 22 Hz – 22 kHz	87 dB
THD plus noise	1 kHz, BW = 80 kHz	<0.05 %
	1 kHz, BW = 22 Hz – 80 kHz	<0.01 %
	1 kHz, BW = 400 Hz – 22 kHz	<0.05 %
IMD	(50 Hz : 7 kHz = 4:1)	<0.02 %
DIM	(3.15 kHz square wave + 15 kHz sinusoid)	0.003 %
Channel separation	1 kHz	84 dB
	20 kHz	63 dB
Crosstalk	1 kHz	<–115 dB
	20 kHz	<–93 dB
Input impedance	Minimum volume	6.4 kΩ
	Maximum volume	2.1 kΩ
Output impedance		<200 Ω
Amplification factor		2.54
Bandwidth		3.5 Hz - 500 kHz
Balance adjustment range		+3 to - 4.7 dB
Attenuator		18 dB
THD	Output voltage = 50 V_{eff}	0.1 %

It must be noted that the measurements were made without any screening of the valves, so RF disturbances from the measurement environment (Elektor lab with PC-based measurement equipment) may have influenced the measured results. If RF interference sources are present in the vicinity of the preamplifier, screens should be provided for the valves.

The five measured performance curves show the following:

A) Amplitude response

On the strongly enlarged scale up to 200 kHz, (the upper performance limit of our audio generator), a slight rise in the amplitude response can be seen. Within the ‘interesting’ part of the audio frequency band, the curve is dead straight!

B) Channel separation

The curves are self-explanatory; the two channels track each other very nicely. These curves start at 200 Hz, in order to eliminate the effect of power supply ripple on the measurements.

C) Frequency spectrum

The effect of power supply ripple can be seen in the frequency spectrum. The spectrum of this ripple reaches to around 800 Hz. This explains the relatively large difference between the A-weighted and linear measurements (100 dBA and 87 dB, respectively). With the 1-kHz tone (1 V_{eff}), essentially only the second harmonic is visible, reaching up to –90 dB. The influence of interference signals, which in this case come from some old monitors located near the amplifier, can be seen with the unscreened valves as peaks in the high-frequency region of the spectrum (30 kHz and 60 kHz).

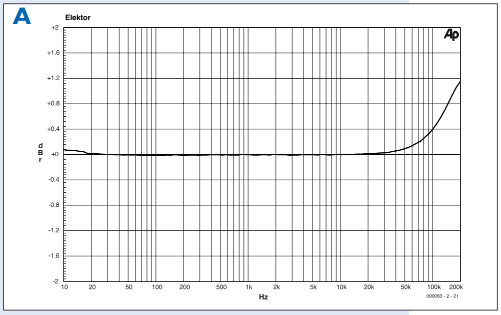
D) THD plus noise

This curve was obtained with a bandwidth of 22 Hz to 80 kHz. The interference signal components come from the effects of power supply ripple, and probably also from induced signals radiated by the transformer.

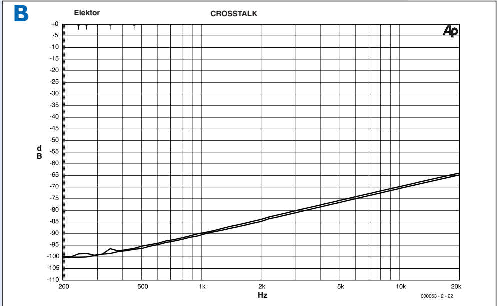
E) Step response

A squarewave signal at the input (10 kHz, 1 V) produces a small overshoot (around 10 percent) at the output.

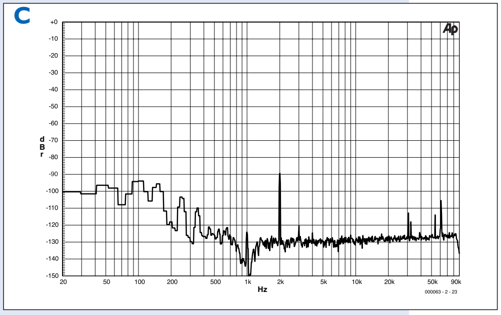
A



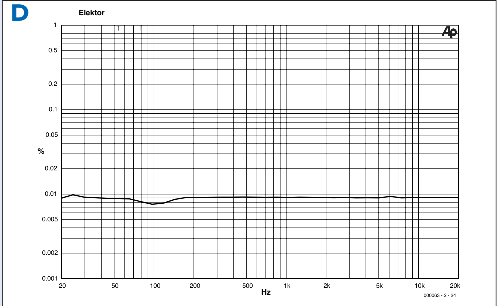
B



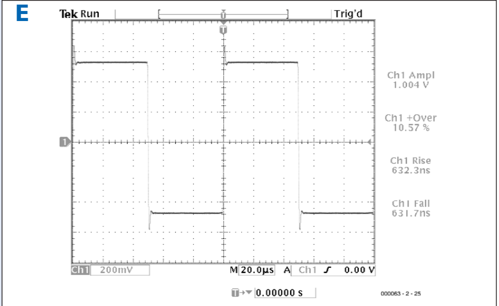
C



D



E



Mnemonic, Operands		Description	Cycles	16-bit Opcode MSb			LSb	Status Affected	Notes
COMF	f,d	Complement f	1	0001	001d	ffff	ffff	Z	
CPFSEQ	f	Compare f with WREG, skip if f = WREG	1 (2)	0011	0001	ffff	ffff	None	6,8
CPFSGT	f	Compare f with WREG, skip if f > WREG	1 (2)	0011	0010	ffff	ffff	None	2,6,8
CPFSLT	f	Compare f with WREG, skip if f < WREG	1 (2)	0011	0000	ffff	ffff	None	2,6,8
DAW	f,s	Decimal Adjust WREG Register	1	0010	111s	ffff	ffff	C	3
DECF	f,d	Decrement f	1	0000	011d	ffff	ffff	OV,C,DC,Z	
DECFSZ	f,d	Decrement f, skip if 0	1 (2)	0001	011d	ffff	ffff	None	6,8
DCFSNZ	f,d	Decrement f, skip if not 0	1 (2)	0010	011d	ffff	ffff	None	6,8
INCF	f,d	Increment f	1	0001	010d	ffff	ffff	OV,C,DC,Z	
INCFSZ	f,d	Increment f, skip if 0	1 (2)	0001	111d	ffff	ffff	None	6,8
INFSNZ	f,d	Increment f, skip if not 0	1 (2)	0010	010d	ffff	ffff	None	6,8
IORWF	f,d	Inclusive OR WREG with f	1	0000	100d	ffff	ffff	Z	
MOVFP	f,p	Move f to p	1	011p	pppp	ffff	ffff	None	
MOVPF	p,f	Move p to f	1	010p	pppp	ffff	ffff	Z	
MOVWF	f	Move WREG to f	1	0000	0001	ffff	ffff	None	
MULWF	f	Multiply WREG with f	1	0011	0100	ffff	ffff	None	
NEGW	f,s	Negate WREG	1	0010	110s	ffff	ffff	OV,C,DC,Z	1,3
NOP	—	No Operation	1	0000	0000	0000	0000	None	
RLCF	f,d	Rotate left f through Carry	1	0001	101d	ffff	ffff	C	
RLNCF	f,d	Rotate left f (no carry)	1	0010	001d	ffff	ffff	None	
RRCF	f,d	Rotate right f through Carry	1	0001	100d	ffff	ffff	C	
RRNCF	f,d	Rotate right f (no carry)	1	0010	000d	ffff	ffff	None	
SETF	f,s	Set f	1	0010	101s	ffff	ffff	None	3
SUBWF	f,d	Subtract WREG from f	1	0000	010d	ffff	ffff	OV,C,DC,Z	1
SUBWFB	f,d	Subtract WREG from f with Borrow	1	0000	001d	ffff	ffff	OV,C,DC,Z	1
SWAPF	f,d	Swap f	1	0001	110d	ffff	ffff	None	
TABLRD	t,i,f	Table Read	2 (3)	1010	10ti	ffff	ffff	None	7
TABLWT	t,i,f	Table Write	2	1010	11ti	ffff	ffff	None	5
TLRD	t,f	Table Latch Read	1	1010	00tx	ffff	ffff	None	
TLWT	t,f	Table Latch Write	1	1010	01tx	ffff	ffff	None	
TSTFSZ	f	Test f, skip if 0	1 (2)	0011	0011	ffff	ffff	None	6,8
XORWF	f,d	Exclusive OR WREG with f	1	0000	110d	ffff	ffff	Z	
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f,b	Bit Clear f	1	1000	1bbb	ffff	ffff	None	
BSF	f,b	Bit Set f	1	1000	0bbb	ffff	ffff	None	
BTFSC	f,b	Bit test, skip if clear	1 (2)	1001	1bbb	ffff	ffff	None	6,8
BTFSS	f,b	Bit test, skip if set	1 (2)	1001	0bbb	ffff	ffff	None	6,8
BTG	f,b	Bit Toggle f	1	0011	1bbb	ffff	ffff	None	
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	ADD literal to WREG	1	1011	0001	kkkk	kkkk	OV,C,DC,Z	

Source:

PIC17C75X Datasheet, Microchip Technology Inc.

Example Application

PIC Single-Board Computer, Elektor Electronics September 2000

Introduction

The PIC17CXXX instruction set consists of 58 instructions. Each instruction is a 16-bit word divided into an OPCODE and one or more operands. The opcode specifies the instruction type, while the operand(s) further specify the operation of the instruction. The PIC17CXXX instruction set can be grouped into three types:

- byte-oriented
- bit-oriented
- literal and control operations.

The field descriptions for the opcodes are shown in the first Table. These descriptions are useful for understanding the opcodes in the Instruction Set table and in each specific instruction description.

byte-oriented instructions,
'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.
The destination designator specifies where the result of the operation is to be placed. If 'd' = '0', the result is placed in the WREG register. If 'd' = '1', the result is placed in the file register specified by the instruction.

bit-oriented instructions,
'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

literal and control operations,
'k' represents an 8- or 13-bit constant or literal value.

The instruction set is highly orthogonal and is grouped into:

- byte-oriented operations
- bit-oriented operations
- literal and control operations

All instructions are executed within one single instruction cycle, unless:

- a conditional test is true
- the program counter is changed as a result of an instruction
- a table read or a table write instruction is executed (in this case, the execution takes two instruction cycles with the second cycle executed as a NOP)

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 25 MHz, the normal instruction execution time is 160 ns. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 320 ns.

PIC17C75x Instruction Set

ELEKTOR ELECTRONICS

Microcontrollers, software

DATASHEET 9/2000

OPCODE FIELD DESCRIPTIONS

Field	Description
F	Register file address (00h to FFh)
P	Peripheral register file address (00h to 1Fh)
I	Table pointer control i = '0' (do not change) i = '1' (increment after instruction execution)
t	Table byte select t = '0' (perform operation on lower byte) t = '1' (perform operation on upper byte literal field, constant data)
WREG	Working register (accumulator)
B	Bit address within an 8-bit file register
K	Literal field, constant data or label
X	Don't care location (= '0' or '1') The assembler will generate code with x = '0'. It is the recommended form of use for compatibility with all Microchip software tools.
D	Destination select 0 = store result in WREG 1 = store result in file register f Default is d = '1'
U	Unused, encoded as '0'
S	Destination select 0 = store result in file register f and in the WREG 1 = store result in file register f Default is s = '1'
label	Label name
C,DC,Z,OV	ALU status bits Carry, Digit Carry, Zero, Overflow
GLINTD	Global Interrupt Disable bit (CPUSTA<4>)
TBLPTR	Table Pointer (16-bit)
TBLAT	Table Latch (16-bit) consists of high byte (TBLATH) and low byte (TBLATL)
TBLATL	Table Latch low byte
TBLATH	Table Latch high byte
TOS	Top of Stack
PC	Program Counter
BSR	Bank Select Register
WDT	Watchdog Timer Counter
$\overline{\text{TO}}$	Time-out bit
$\overline{\text{PD}}$	Power-down bit
dest	Destination either the WREG register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

PIC17CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f,d**	ADD WREG to f	1	0000	111d	ffff	ffff	OV,C,DC,Z	
ADDWFC	**f,d**	ADD WREG and Carry bit to f	1	0001	000d	ffff	ffff	OV,C,DC,Z	
ANDWF	**f,d**	AND WREG with f	1	0000	101d	ffff	ffff	Z	
CLRF	**f,s**	Clear f, or Clear f and Clear WREG	1	0010	100s	ffff	ffff	None	3

Mnemonic, Operands		*Description*	*Cycles*	*16-bit Opcode*				*Status Affected*	*Notes*
				MSb			*LSb*		
ANDLW	**k**	AND literal with WREG	1	1011	0101	kkkk	kkkk	Z	
CALL	**k**	Subroutine Call	2	111k	kkkk	kkkk	kkkk	None	7
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	TO,PD	
GOTO	**k**	Unconditional Branch	2	110k	kkkk	kkkk	kkkk	None	7
IORLW	**k**	Inclusive OR literal with WREG	1	1011	0011	kkkk	kkkk	Z	
LCALL	**k**	Long Call	2	1011	0111	kkkk	kkkk	None	4,7
MOVLB	**k**	Move literal to low nibble in BSR	1	1011	1000	uuuu	kkkk	None	
MOVLR	**k**	Move literal to high nibble in BSR	1	1011	101x	kkkk	uuuu	None	
MOVLW	**k**	Move literal to WREG	1	1011	0000	kkkk	kkkk	None	
MULLW	**k**	Multiply literal with WREG	1	1011	1100	kkkk	kkkk	None	
RETFIE	—	Return from interrupt (and enable interrupts)	2	0000	0000	0000	0101	GLINTD	7
RETLW	**k**	Return literal to WREG	2	1011	0110	kkkk	kkkk	None	7
RETURN	—	Return from subroutine	2	0000	0000	0000	0010	None	7
SLEEP	—	Enter SLEEP Mode	1	0000	0000	0000	0011	TO, PD	
SUBLW	**k**	Subtract WREG from literal	1	1011	0010	kkkk	kkkk	OV,C,DC,Z	
XORLW	**k**	Exclusive OR literal with WREG	1	1011	0100	kkkk	kkkk	Z	

Notes

1: *2's Complement method.*

2: *Unsigned arithmetic.*

3: *If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.*

4: *During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL)*

5: *Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction.*

6: *Two-cycle instruction when condition is true, else single cycle instruction.*

7: *Two-cycle instruction except for TABLRD to PCL (program counter low byte) in which case it takes 3 cycles.*

8: *A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.*