NOVEMBER 2003
£3.45

ELEKTOR ELECTRONICS

THE ELECTRONICS & COMPUTER MAGAZINE

www.elektor-electronics.co.uk

Rev Counter for R/C Models

Running Text Display

Stepper Motors Uncovered

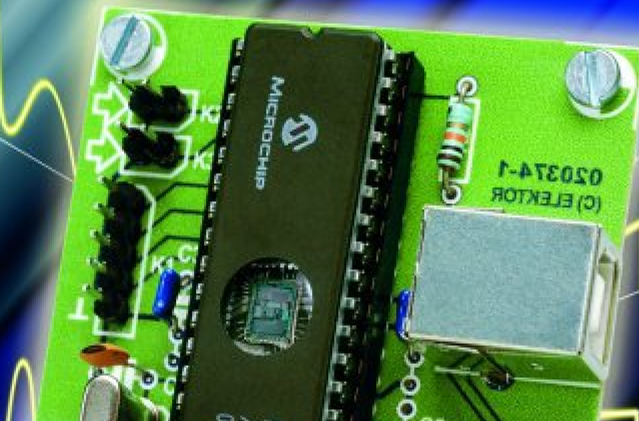
FET Amp with Valve Sound

Electric Charge Meter

Directional Microphone

USB Analogue Converter

measure and control via USB



11
9 770268 451104

USB Analogue Converter

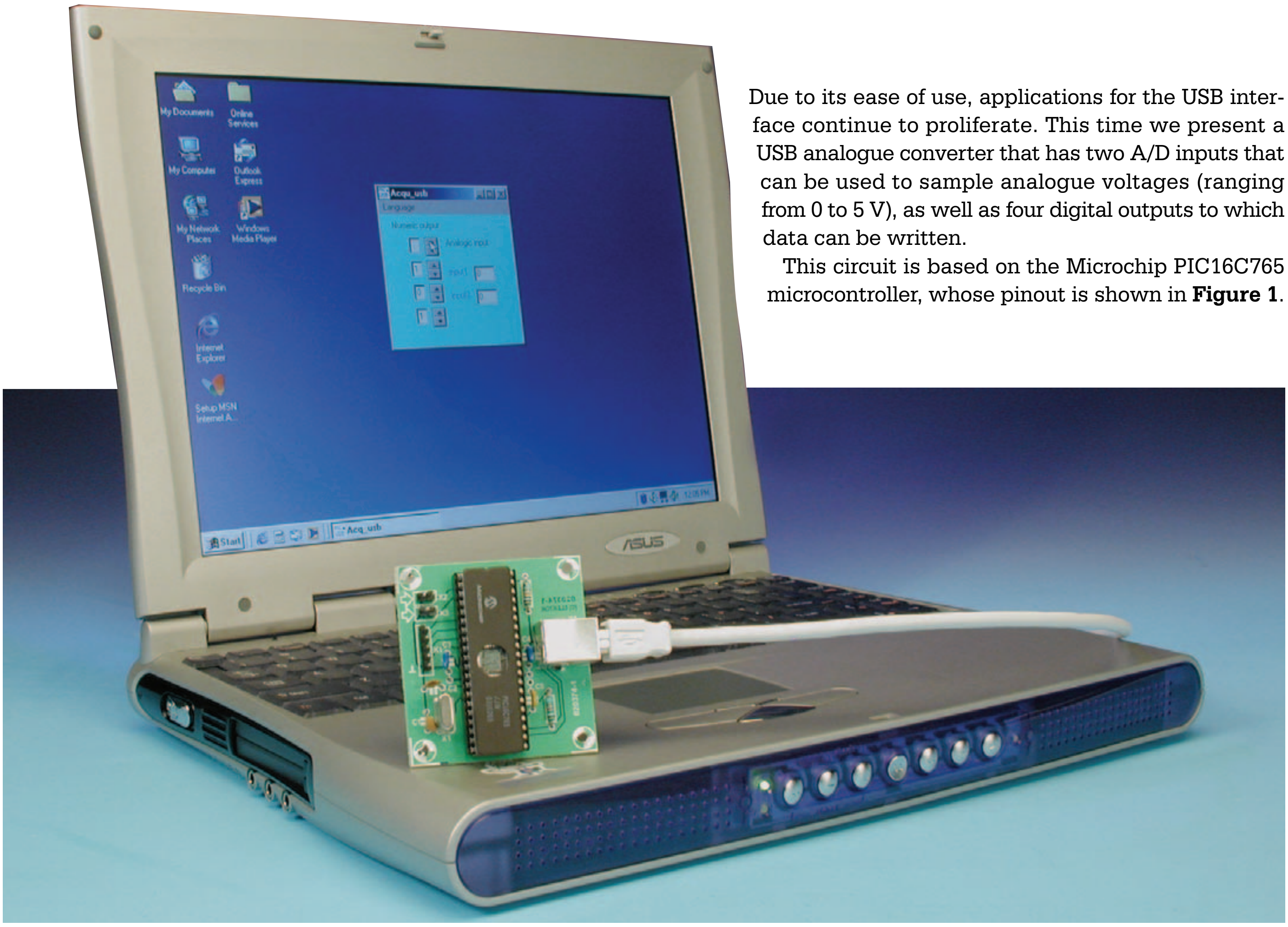
With two analogue inputs and four digital outputs

Design by E. Brador

This circuit is based on the PIC16C765, a recent addition to the large Microchip PIC family. It can be used to measure two analogue voltages (0–5 V) via two analogue inputs and send the results to a PC via a USB connection. There are also four digital outputs whose levels can be set using the PC.

Due to its ease of use, applications for the USB interface continue to proliferate. This time we present a USB analogue converter that has two A/D inputs that can be used to sample analogue voltages (ranging from 0 to 5 V), as well as four digital outputs to which data can be written.

This circuit is based on the Microchip PIC16C765 microcontroller, whose pinout is shown in **Figure 1**.



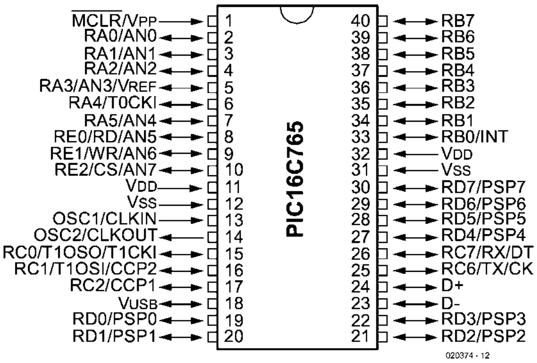


Figure 1. PIC16C765 pinout (source: Microchip).

The PIC16C765

Together with its little brother, the PIC16C745 (28 pins), the PIC16C765 was the first PIC processor to have an on-board USB interface. This RISC processor (with only 35 instructions) has 8 Kwords of 14-bit program memory and 256 bytes of data memory. It also features eight analogue inputs, a variety of digital inputs and outputs, a UART and a USB controller, among other things.

Microchip provides assembler files for this microcontroller in order to simplify the essential communications between the USB PIC and the PC. For more information, you can visit the manufacturer's website at http://www.microchip.com/

Any electronic technician or hobbyist who wants to keep abreast of new developments can hardly avoid the USB port. Here we use USB Type 1 (USB 1.1), which is more than fast enough for sampling signals whose levels change relatively slowly, such as temperature, pressure, etc.

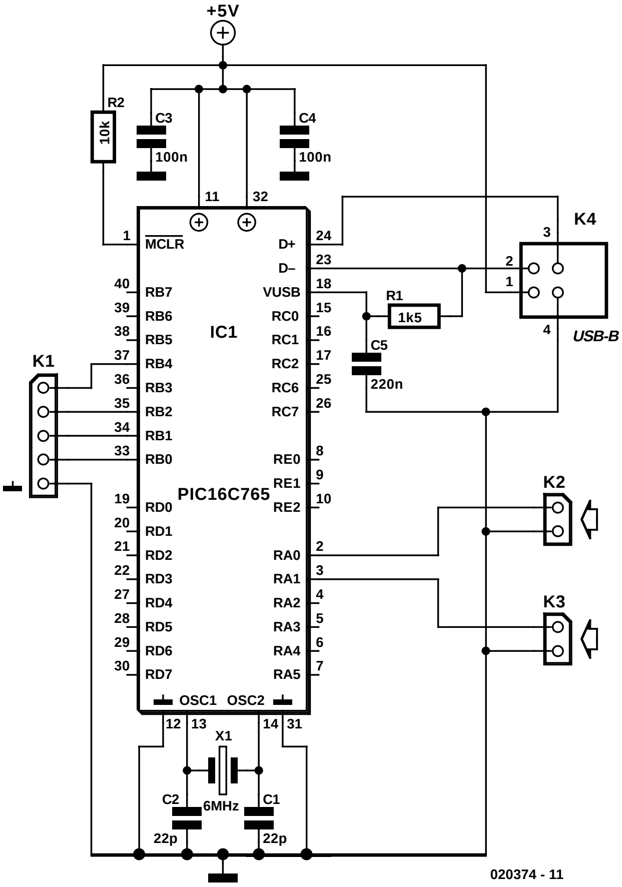


Figure 2. Except for IC1, there's not a lot of electronics in the USB analog converter.

The electronics

A glance at the schematic diagram in **Figure 2** clearly shows that our circuit is almost unbelievably simple. The microcontroller is surrounded by only a handful of components, consisting primarily of the quartz crystal and the usual capacitors connected to inputs OSC1 and OSC2, which collectively ensure that the program runs at 24 MHz inside the IC, despite the fact that a 6-MHz crystal is used. This is because an integrated PLL causes the basic frequency generated by the crystal to be multiplied by a factor of four.

Pushbutton switch S1 can be used to reset the processor in the event of problem. The presence of a device on the USB port is detected via resistor R1 (1k5). The rest of the circuit consists of various connectors for the inputs and outputs.

One of the advantages of USB can already be seen here: the circuit draws its power directly from the USB port.

The circuit board

Although a printed circuit board has been designed for this circuit (**Figure 3**), it is naturally not much larger than the IC. Considering the small number of components (a microcontroller, a USB chassis-mount connector, a crystal, two resistors and five capacitors), instructions for assembling the circuit board are actually unnecessary, except that we recommend using a good-quality socket for the microcontroller.

If you cannot program the PIC16C765 used in this circuit, a pre-programmed version can be ordered form Readers Services under order number **020374-41**.

Installation

After the USB cable (which also provides the power) has been connected, the circuit will be recognised as a Human Interface Device (HID). However, this only works properly under Windows 98 SE, Windows ME, Windows XP or Windows 2000. After the device has been recognised, the proper driver will be installed.

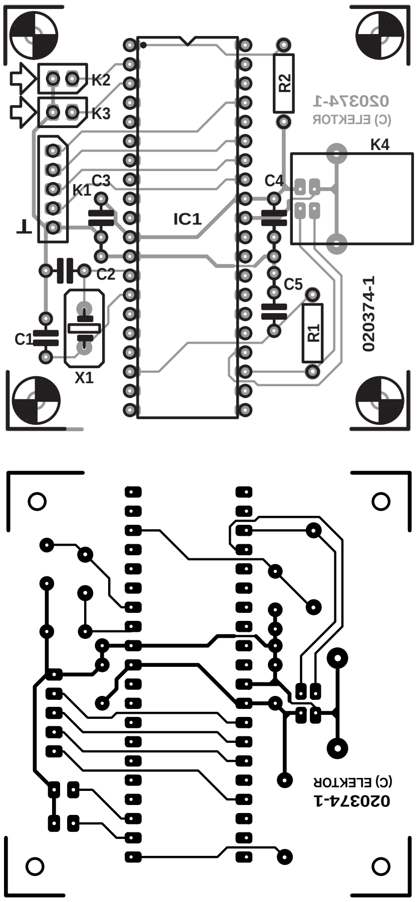
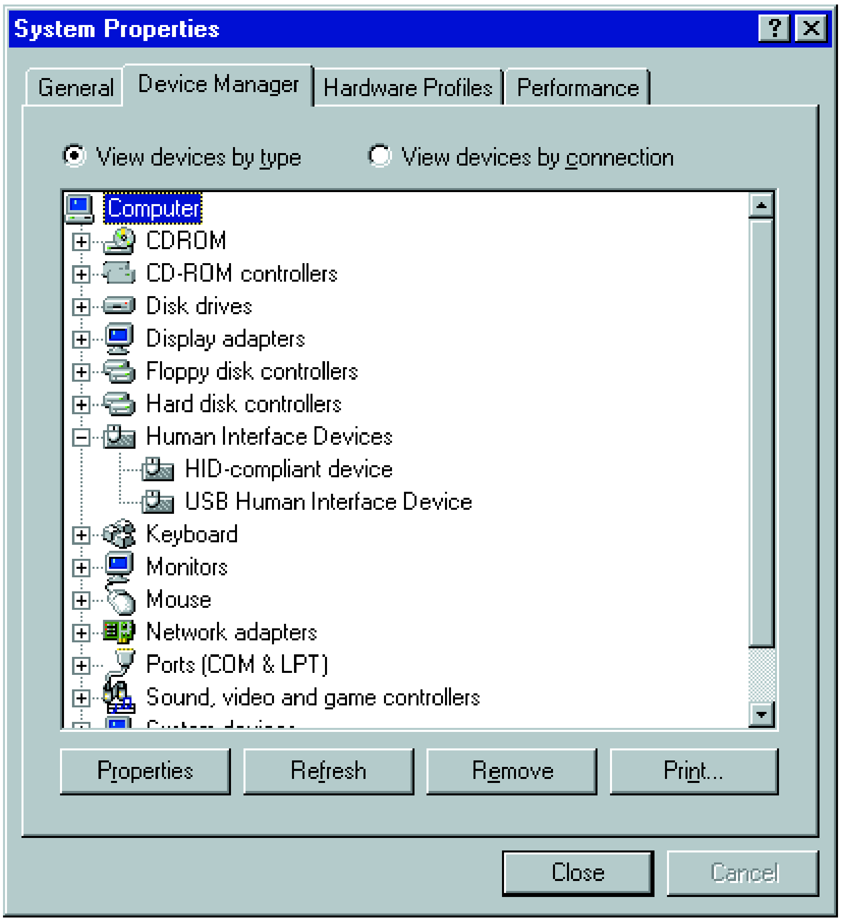


Figure 3. Printed circuit board and component layouts for the USB converter circuit.



● View devices by type ○ View devices by connection

```
Computer
+ CDROM
+ CD-ROM controllers
+ Disk drives
+ Display adapters
+ Floppy disk controllers
+ Hard disk controllers
- Human Interface Devices
    HID-compliant device
    USB Human Interface Device
+ Keyboard
+ Monitors
+ Mouse
+ Network adapters
+ Ports (COM & LPT)
+ Sound, video and game controllers
```

Figure 4. Windows recognises the interface as an HID (Human Interface Device). This screendump is taken from Windows XP.

COMPONENTS LIST

Resistors:
R1 = 1kΩ5
R2 = 10kΩ

Capacitors:
C1,C2 = 22pF
C3,C4 = 100nF
C5 = 220nF

Semiconductors:
IC1 = PIC16C765, programmed, order code **020374-41**

Miscellaneous:
X1 = 6MHz quartz crystal
K1 = 4-way PCB terminal block
K2,K3 = 2-way PCB terminal block
K4 = USB socket, PCB mount, type B
PCB, order code **020374-1**
Disk, contains hex-code and Windows-software, order code **020374-11** or Free Download

The microcontroller identifies itself to the PC by specifying its vendor ID (VID) and product ID (PID). As this is a hobbyist circuit, we have retained the default settings from Microchip. If you consider that up to 256 different devices can be connected to a single USB port, you can understand the need for unambiguous identification.

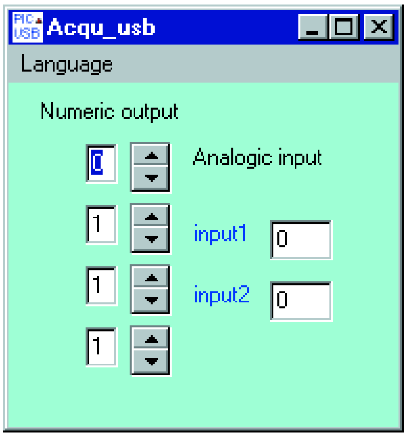
After being installed, the circuit should be visible as a USB device in the Device Manager window (see the screen dump in **Figure 4**).

The assembler program

At the Microchip website, you can find a variety of programs that illustrate how to program this microcontroller for USB communications, which means for data traffic via the USB port.

If you want to write programs for this microcontroller, we advise you to first become quite familiar with the USB protocol.

Some of the important portions of the program are printed in **Listings 1 & 2**, so you can see what is involved



Numeric output

0	Analogic input	
1	input1	0
1	input2	0
1		

Figure 5. The program window shows the values measured on the two analogue inputs and the levels of the four digital outputs.

in making an analogue measurement and sending the measured values.

Assembling the program is more complex than with a PIC16F84, since several files from Microchip are necessary for this purpose. The program can be compiled using MPLAB, the well-known PIC development environment that Microchip makes available free of charge.

The microcontroller program is available in the form of a hex file (Elk.hex) and is available on diskette from Readers Services (order number **020374-11**) or from the *Elektor Electronics* website.

Operation

The PC software for the PIC is written in Delphi, and some of its subroutines are suitable for use with USB.

It is necessary to install the file port.dll in the same directory as the main program.

The compiled PC program, which is called Acq_usb, has been tested under Windows XP, Windows ME and Windows 98 SE, and it works perfectly under these three operating systems.

Using this program, voltages between 0 and 5 V can be cyclically sampled on two channels (see the screen dump in **Figure 5**). The program automatically starts sampling.

To test the circuit, you can use a power supply or battery with a potentiometer to allow the voltage to be adjusted over the range of 0–5 V.

The arrow buttons can be used to write '1' of '0' values to the four outputs by clicking on the up arrow (for '1') or the down arrow (for '0').

Although this is a 'bare bones' circuit, it can be used for a wide variety of measurements by connecting various types of converters or transducers to the inputs (via an interface if necessary). In this way, you can use it to measure temperatures or pressures.

(020374-1)

Listing 1. Analogue measurement subroutine (one channel).

```
acqudat
    banksel PIE1           ;memory bank selection
    bcf PIE1,ADIE          ;don't use interrupt for conversion

    banksel ADCON0
    bcf ADCON0,CHS2        ;input selection
    bcf ADCON0,CHS1        ;microprocessor's
    bcf ADCON0,CHS0        ;RA0


    bsf ADCON0,ADON        ;set internal converter


    pagesel Pdelay
    call Pdelay            ;call delay for conversion

    banksel ADCON0
    bsf     ADCON0,GO      ;start new conversion

loop
    btfsc   ADCON0,GO_DONE ;loop

    goto loop              ;conversion not yet finished

    banksel PIR1
    BCF PIR1,ADIF          ;reset end of conversion flag

    banksel ADRES          ;fetch conversion result
    movf ADRES,w
    banksel buffer
    movwf buffer           ;and move to buffer for transmission to PC
```

Listing 2. Subroutine for transmitting measured values to the PC.

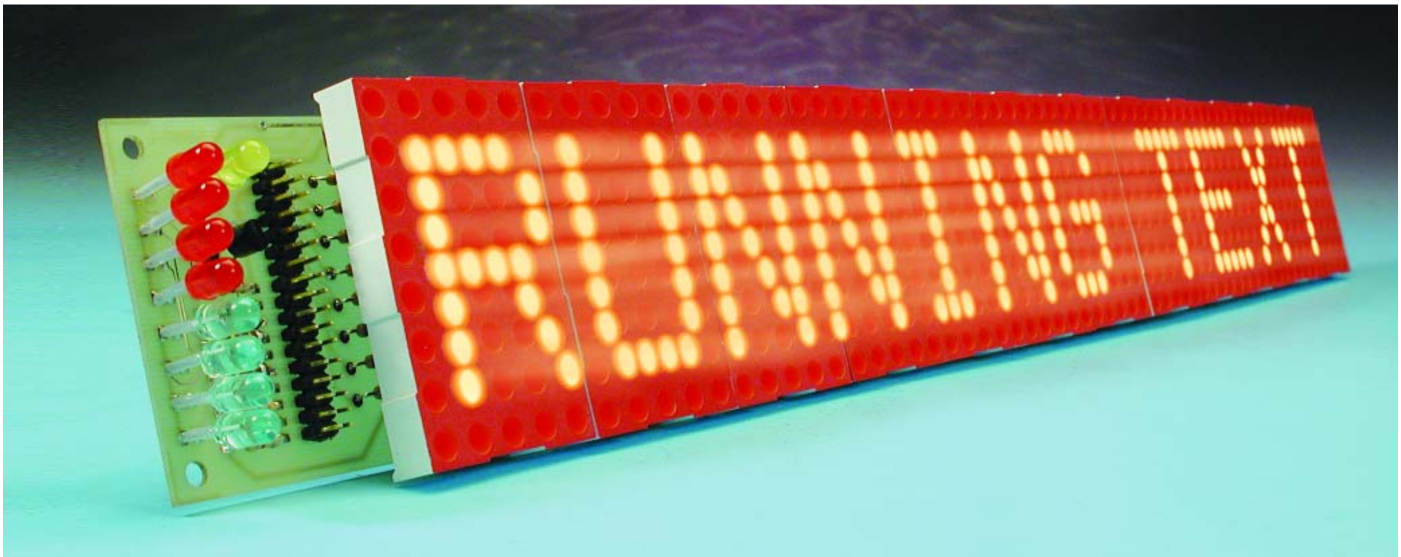
```
SendData              ;subroutine for transmission of values to PC
  pagesel  PutEP1
  bankisel buffer     ;place buffer address into FSR
  movlw    buffer
  movwf    FSR
  movlw    0x4      ;send 4 bytes
  call     PutEP1
  btfss   STATUS,C ;stay in this loop until data transmission complete
  goto sendData

  banksel buffersFilled
  clrf     buffersFilled
  bankisel buffer
  clrf     buffer
  clrf     buffer+1
  clrf     buffer+2
  clrf     buffer+3
  clrf     buffer+4

  pagesel acqudat
  goto acqudat
```

Running Text Display

A 10-character display for the Elektor Flash Micro board

Design by W. Wätzig

This peripheral circuit for the 89S8252 Flash Board provides a large-format, 10-character running-text display using dot-matrix modules.



The subject of this article may sound familiar. Haven't we already devoted enough attention to running-text displays? The answer is, yes and no! In the 'Modular Dot Matrix Display' (*Elektor Electronics*, June 2001), the text to be displayed is passed from a PC to an 89S8252 microcontroller using an RS232 interface. In that design, the task of the microcontroller is to convert the characters into bit patterns for the matrix modules and manage the multiplexed drive signals for the display modules.

At first glance, the basic idea of the circuit shown in **Figure 1** appears to be the same. However, in this case the only 'intelligence' is located in a separate microcontroller system, in the form of the AT89S8252 Flash Board. The actual running-text display, which can display ten characters on its twelve display modules, is fully passive. However, this design has a few unusual features.

The running-text functions can be controlled using a PC keyboard connected directly to the circuit, as well as via the serial interface of the Flash Board. In addition, a DS1302 real-time clock is included in the circuit, to allow the date and time to be output alternately.

Inputs

A serial PS/2 interface is used for entering characters and control codes. The PC keyboard is connected to K1, and the scan codes are stored in the 40105 first-in, first-out (FIFO) memory. The Flash Board reads the keyboard data from the FIFO memory at its own speed and converts the scan codes into ASCII characters. The character coding can be selected to match the German (DE) or English (EN) keyboard layout. The keyboard selection is indicated by the most significant bit of the status display (textsel); this LED is on if the EN layout is selected.

Alternatively, the circuit can be controlled via the serial interface of the microcontroller, using a maximum data transfer rate of 1200 baud. The baud rate can be set to 150, 300, 600 or 1200 using jumpers JP2 and JP3. Jumper JP1 selects either keyboard or serial interface control; the selection takes effect after the microcontroller is reset.

The texts to be displayed are stored in the microcontroller EEPROM, so they are protected in case of loss of power. The 2048 bytes of EEPROM memory are divided into eight 240-byte text blocks. Each block can hold twelve lines of text, with each line having up to twenty characters.

To show where the text is being stored during entry, the block number of the currently selected text

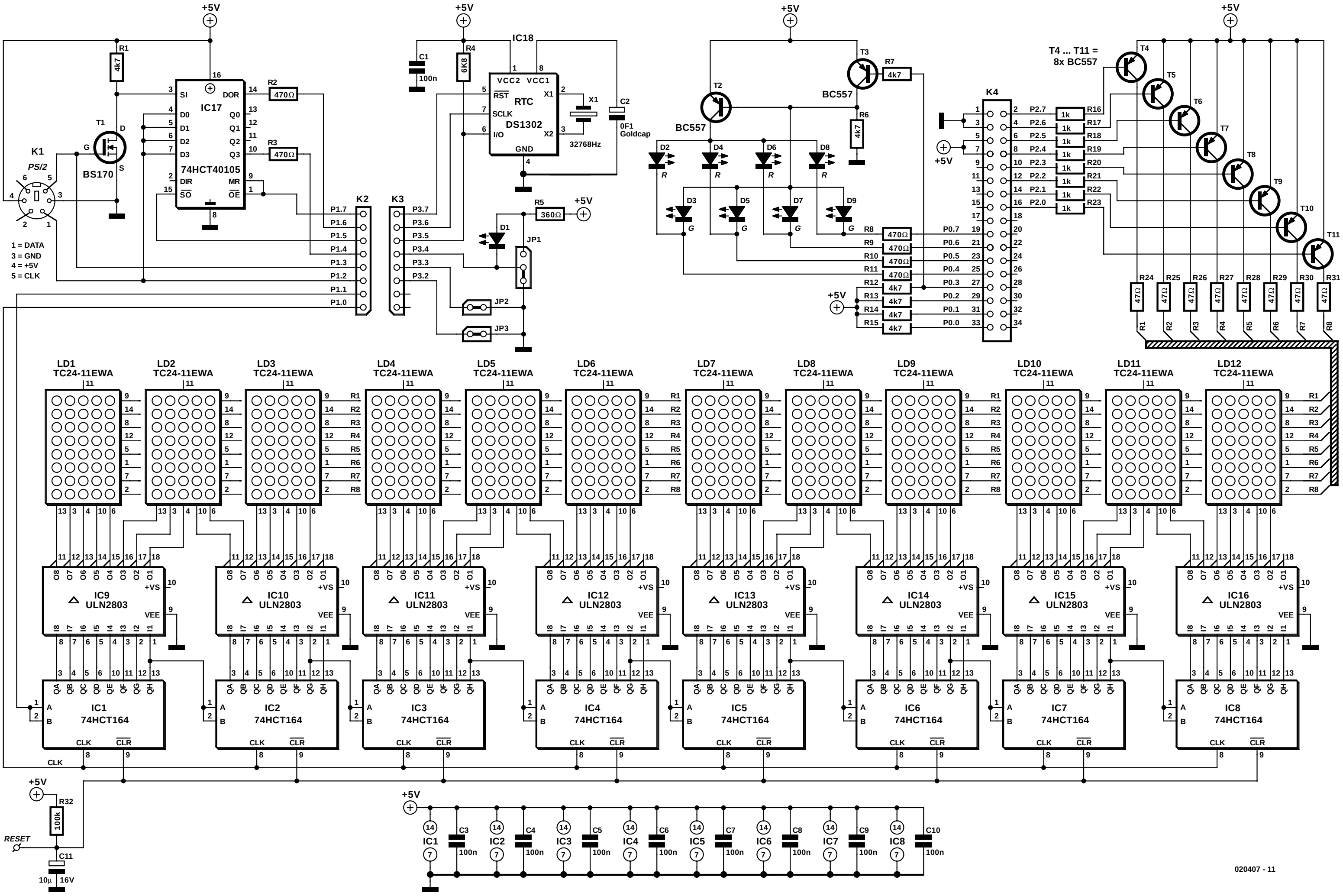


Figure 1. The running-text display has twelve display modules and a 60-stage shift register.

block is shown in binary form in the upper nibble of the eight-bit status display, and the current line number is shown in the lower nibble.

Outputs

The display consists of twelve dot-matrix display modules, each of which has a 5 × 8 LED matrix. This provides a total of 60 columns, each containing eight LEDs. Ten characters can be displayed at the same time, each in a 5 × 8 matrix, with one column of space between each pair of characters. With such a large number of LEDs, multiplex operation is the only reasonable choice.

A display buffer for a total of 20 characters is provided using 120 bytes of the internal RAM. The LED columns are multiplexed at a rate of 3600 Hz, which is driven by an interrupt. The refresh rate for the entire display is thus 60 Hz.

The column data is clocked out from the microcontroller board via P1.1 into a shift register formed by cascaded 74HCT164 ICs. The clock signal appears on P1.0 and is applied to all of the shift register ICs simultaneously. Once the column data has been clocked into the shift register, the row to be displayed is enabled using one of the driver transistors T4–T11. These driver transistors occupy all of port P2 of the microcontroller.

Text management

Control characters are used for text management. Function keys F1–F11 and other special characters, such as CR, Pause, Enter, Delete and so on, are used for this purpose.

When the serial interface is used, the control function codes are entered using the surrogate representation **#x**. The # character indicates a control function, and the following character specifies the specific function. For example, function key F1 is replaced by the character sequence **#1**. The control characters that can be entered are listed in the 'Control character input' table, along with their keyboard designations.

The circuit

The circuit consists of the microcontroller on the Flash Board, which provides the control functions, and the peripheral circuitry on the display board, which is connected to the Flash Board using three flat cables. The display board consists of four functional blocks:

- The 74HCT105 FIFO (IC17), for entering the scan codes from the keyboard. This memory decouples the keyboard data clock from the scan code read routine, in order to prevent any bits from being lost when the read routine is interrupted by the display routine.

- The DS1302 real-time clock (IC18) with the Goldcap storage capacitor (C2). When the running-text display is used for the first time, the actual date and time must be set in the real-time clock IC. We recommend doing this ‘top down’, starting with the year and finishing by setting the seconds. This is because the seconds register of the real-time clock is preset in the factory to ‘80’, which corresponds to a wait mode. The Goldcap capacitor provides backup power for the real-time clock, so it will continue running when the power is switched off.

- Two 4-bit LED status displays (D2–D9) for the text block number and line number.

- The dot-matrix display, consisting of the display modules (LD1–LD12) and 74HCT174 shift registers (IC1–IC8), the ULN2803 column drivers (IC9–IC16) and the BC557 row driver transistors (T4–T11).

The software

The microcontroller program can be divided into three parts, consisting of the initialisation routine, the display loop and the character input loop.

Initialisation

After the microcontroller is switched on, the initialisation routine is triggered by the reset interrupt and executes once. It first sets the baud rate for the serial interface (the microcontroller UART) according to the value selected by the levels on pins P3.2 and P3.3. It also sets the Keyboard/Serial flag (jumper JP1), creates the display buffer (120 bytes in RAM, starting at address 080h), and initialises the display counter.

Next, the keyboard is switched to Scan Code 3, in which mode only one character is sent for each keypress. This considerably simplifies further processing of the scan codes.

The real-time clock is also initialised, allowing the Goldcap storage capacitor to be recharged from the supply voltage. The initialisation phase ends with writing an initial text to the display buffer and enabling the interrupt, thus allowing Timer0 to run.

The display loop

The display loop is triggered by a Timer0 interrupt every 277 μs, which corresponds to

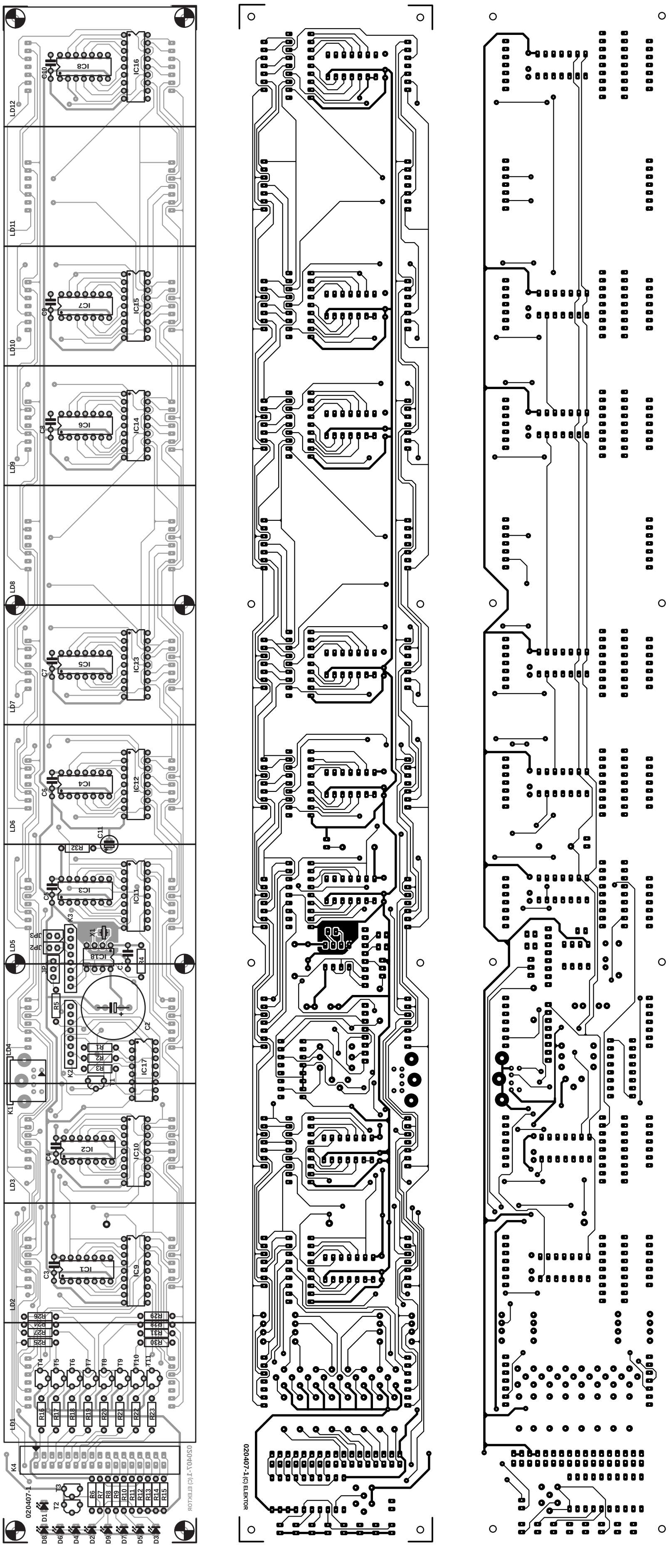


Figure 2. The printed circuit board layout (50% of actual size).

a rate of 3600 Hz. The entire dot-matrix display is thus refreshed 60 times per second. The display loop fetches the display bytes sequentially from the RAM and presents them to the shift register, generates clock pulses for the shift register, generates the data patterns for the LED displays, and outputs the control signals for switching the text, scrolling the text, and setting the display to inverted or blinking mode.

COMPONENTS LIST

Resistors:

R1,R6,R7,R12-R15 = 4kΩ7
R2,R3,R8-R11 = 470Ω
R4 = 6kΩ8
R5 = 360Ω
R16-R23 = 1kΩ
R24-R31 = 47Ω
R32 = 100kΩ

Capacitors:

C1,C3-C10 = 100nF
C2 = 0.1F 5.5V Goldcap (Panasonic NF)
C11 = 10µF 63V radial

Semiconductors:

D1 = LED, yellow, low current, 5mm
D2,D4,D6,D8 = LED, red, low current, 5mm
D3,D5,D7,D9 = LED, green, low current, 5mm
IC1-IC8 = 74HCT164
IC9-IC16 = ULN2803A
IC17 = 74HCT40105
IC18 = DS1302 (Dallas)
T1 = BS170
T2-T11 = BC557B

Miscellaneous:

JP1 = 3-way SIL pinheader with jumper
JP2,JP3 = 2-way SIL pinheader with jumper
K1 = 6-way mini-DIN socket, PCB mount (PS/2)
K2,K3 = 8-way SIL pinheader
K4 = 34-way SIL pinheader
LD1-LD12 = 5x8 dot matrix display with common cathode, size 60.8×38 mm, Kingbright type TC24-11EWA
AT89S8252-24PC for 89S8252 Flash Micro board (010208), programmed
Disk, contains controller program (source and hex), order code **020407-11** or Free Download
PCB, order code **020407-1** (see Readers Services page)

The character input loop

This loop waits for the next character from the keyboard or the serial interface. Each scan code from the keyboard is read via the FIFO and converted into a control character or an ASCII character using a code table (tastasc). Alternatively, ASCII characters are read from the serial interface. In this case, control characters are generated using a surrogate representation.

When a control character (such as Return, Shift, or Backspace) is encountered, the associated routine is called using a dispatcher. Each ASCII character is stored in the next free location in the EEPROM, and its character image (five bytes) is read from a table (charimg) and written to the display buffer in RAM.

Printed circuit board

It's not easy to design and produce a printed circuit board big enough to hold twelve large display modules and the control electronics (how many people have an etching tray that can hold a 50-cm PCB?). For this reason, we have designed a circuit board layout, which is shown at half its actual size in **Figure 2**. This double-sided, through-hole-plated circuit board is not available from Readers Services, but it can be obtained from The PCBShop via the link on the *Elektor Electronics* home page.

Fortunately, stuffing the board is quite easy. The SIL connectors, jumper pins and mini-DIN socket are fitted on the bottom of the board, while the remainder of the components are fitted on the top. Make sure that none of the components extends more than 8 mm above the surface of the board, since otherwise the display modules installed above these components will not fit into their sockets. The Goldcap capacitor specified in the components list is exactly 5.5 mm high. If you wish to use sockets for the ICs, only very low-profile types are suitable.

(020407-1)

Control character input

Function	Key	Serial Input	Function
Text selection	Pause Fx	#P #x	Selection of text block, Fx = F1-F8
	F1-F12	#1 to #9 #A #B #C	election of display lines #1 to #12 in current text block
Edit/Enter Texts	Cr	#+	Go to next line
	BackSpace/Delete	#–	Move back one character
	Shift left/right		Lower/Upper case letters
	Alt		Switch to special characters @ { [ ] } ~ | \
	Enter (num. keypad)	#Z	Clears current text line
	Insert	#*	Starts entry of a long text of up to 240 characters in current display line, extending over several lines. During text entry the display is switched to inverse characters. Text entry is ended by a pressing Insert again.
Display manipulation	Scroll Lock	#R	Scrolls running text
	Home	#0	Start of line and reset scrolling
	Print Screen Fx1 Fx2	#D #x1 #x2	Automatic text display #Fx(1) to #Fx(2)
	Num Lock	#N	Invert/blink display
Real-time Clock control	Esc Esc	#E #E	Display date and time
	Esc Fx yy	#E #x yy	F1: seconds (yy = 00-59)
			F2: minutes (yy = 00-59)
			F3: hours (yy = 00-23)
			F4: day of month (yy = 01-31)
			F5: month (yy = 01...12)
			F6: day of week (yy = 01-07)
			F7: year (yy = 00-99)
Diacriticals etc.	Direct entry on German keyboard	:s = ß :: = : :a = ä :A = Ä :o = ö :O = Ö :u = ü :U = Ü	
Keyboard encoding	Page Up		English keyboard
	Page Down		German keyboard

Electric Charge Meter

In search of the Coulomb

Design by B. Kainka

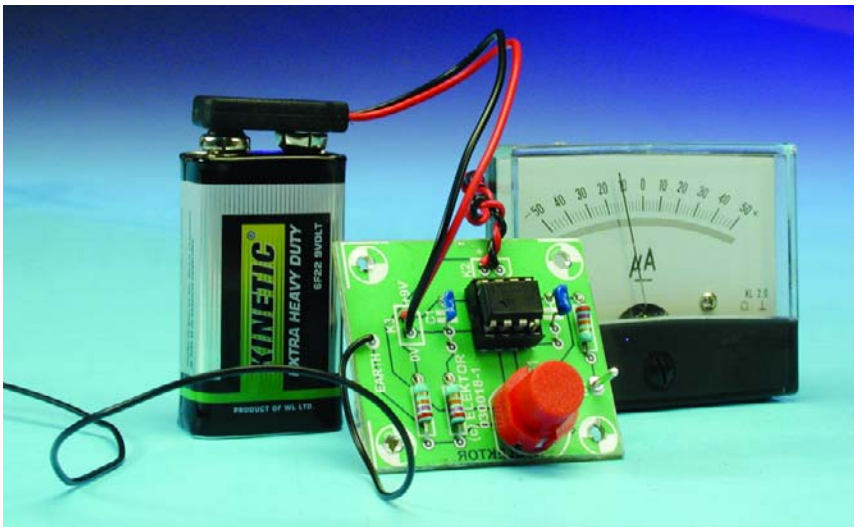
Just knowing that electrostatic charges can damage and destroy expensive components is fine, but being able to measure them is even better because it allows you to put a finger on potential problems (pun intended).

Static charges are feared in all electronics laboratories and workshops — and rightly so because a vinyl carpet has the ability to charge up persons to voltages in excess of 20 kV. Careless touching of sensitive ICs can cause their instant demise. However, problematic as they are, static charges also have very interesting aspects. For example, it has been observed that two persons walking the same carpet with different shoes can accumulate opposite static charges, that is, release voltages with different polarities (positive and negative). Many more questions remain to be answered by physics. For example, it is true that women generally accumulate a negative static charge whereas men carry a positive one, or is this just a rumour spread by 'bachelors for life'? Obviously, the question may only be answered by statistically founded research to which *Elektor* readers are kindly invited to contribute.

In experiments with electrostatic charges, the decisive quantity is not voltage but electric charge, expressed in coulombs. A high voltage is not necessarily dangerous when present on a relatively small object. The decisive relation is expressed by the equation

charge = capacitance × voltage

A human body represents a capacitance of just a few picofarads. When charged to a couple of kilovolts (as may happen in everyday situations) the corresponding charge is of the order of a few hundred nanocoulombs (nC).



Circuit operation

Basically, the electric charge meter consists of a measurement capacitor coupled to a high-impedance measurement amplifier. A charged body should briefly touch the measurement input. From elementary physics, you should remember that this causes the charge to be distributed equally between charged body and capacitor, so the latter is charged to an extent that an be measured with the aid of a charge meter and a regular voltmeter. In practice, assuming that a measurement capacitor of 100 nF is used, a voltage increase of 1 V corresponds to a measured charge of 100 nC. The circuit supply voltage of 9 V affords a measurement range of about –300 nC to +300 nC.

Because an extremely high impedance input is required (in order to prevent charge loss) an opamp with a FET (field effect transistor) input is applied in our circuit. Such is the input impedance of the opamp used here that the measured voltage shows very little or no decrease even after several minutes. The measurement amplifier proper

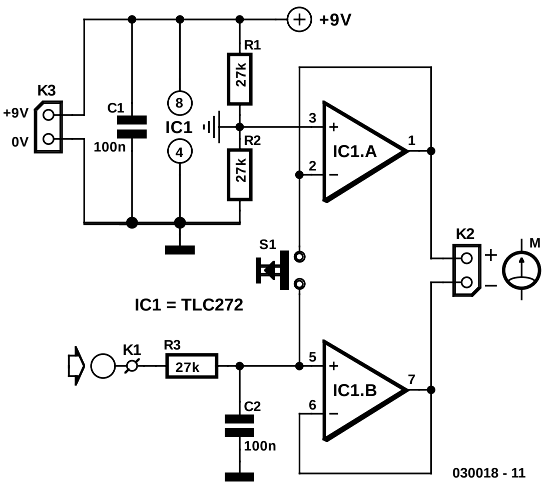


Figure 1. The circuit of the Electric Charge Meter is simplicity itself.

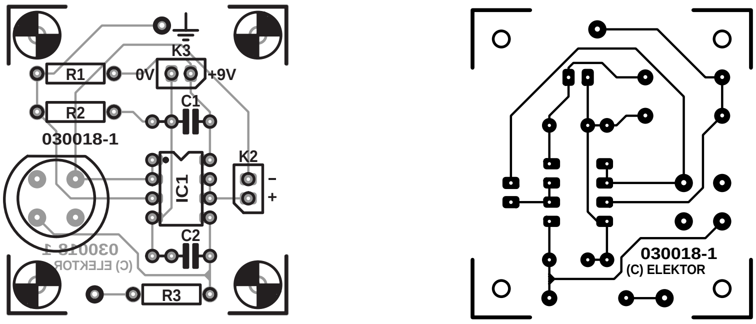


Figure 2. PCB layout with component mounting plan.

is configured as a unity-gain (×1) buffer with full feedback. The second opamp, IC1, supplies a low-impedance bias voltage of half the supply rail. This arrangement allows a simple voltage source to be used to measure both polarities, provided the bias voltage is fully earthed, for example, by a connection to a copper water pipe.

Before the actual measurement can begin, you have to press the Zero pushbutton. Contrary to what you might expect, this causes the measurement capacitor to be charged to half the supply voltage rather than being discharged to 0 V. However, because the instrument's negative terminal is connected to the 4.5-V (mid-supply) level, it will duly indicate 'zero' charge.

Practical use

Next we can bring a charged body into contact with the measurement input. Depending on its charge polarity, the body can add to the charge already present on the capacitor, or discharge it to some extent. The resulting voltage increase or decrease that can be observed on the meter indicates the charge level as well its polarity. The duration of the contact is immaterial in this respect — what's imperative is to keep the body itself properly isolated when the measurement takes place.

From your physics classes you may (fondly) remember that experiments with electrostatic voltages and discharges are subject to haphazard results. As a matter of course, the actual amount of charge carried on a body is heavily dependent on all sorts of leakage, isolation defects, not forgetting the relatively humidity of the (classroom) air. In general, electrostatic discharges are easier to observe under relatively dry air conditions, to which we should add that free ions are capable of reversing the charge polarity of free air. This ability can be proved by using the Electric Charge Meter to verify the operation of air ionizers.

(030018-1)

COMPONENTS LIST

Resistors:
R1,R2,R3 = 27kΩ

Capacitors:
C1,C2 = 100nF

Semiconductors:
IC1 = TLC272

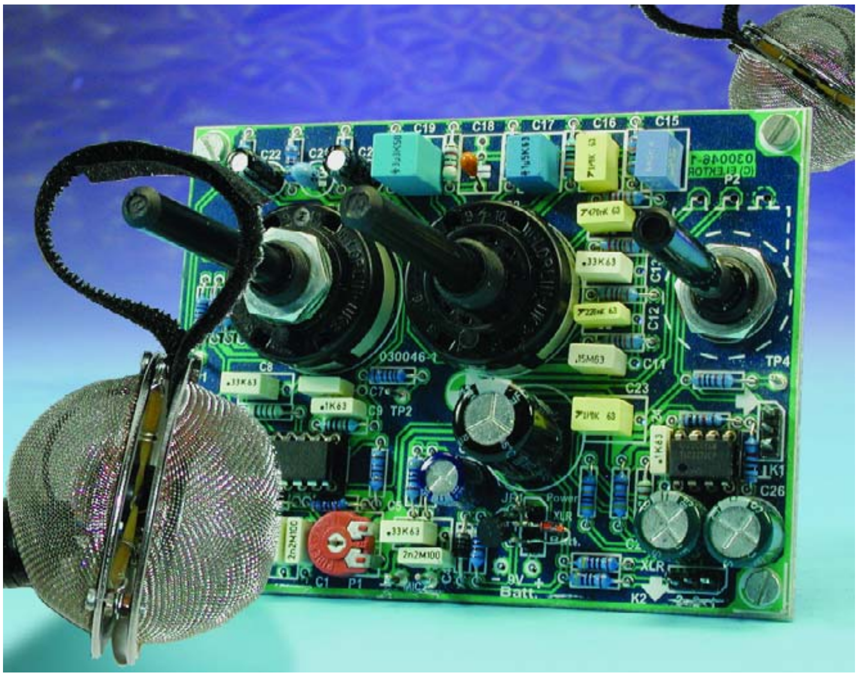
Miscellaneous:
K1 = PCB solder pin
K2 = connection for moving-coil or multimeter
K3 = connection for 9-V battery (with clip-on connector)
S1 = pushbutton, 1 make contact
PCB, order code 030018-1 (available from The PCBShop)

Directional Microphone

with switchable characteristic

Design by K. Rohwer and G. Schmidt

The directional microphone published in the July-August 2002 issue of *Elektor Electronics* does its job, but does not have the right directional characteristic. The authors have developed a new circuit which satisfies all the requirements in practice as well as in theory.



The microphone published in the July-August 2002 issue of *Elektor Electronics* was designed to reduce the risk of feedback when used with a jazz harmonica. However, it does not exhibit the ideal cardioid directional characteristic. It can be shown mathematically that as the characteristic is adjusted from omnidirectional at one extreme to figure-of-eight at the other, it is never strictly cardioid or supercardioid: instead a null is gradually introduced, like a belt being tightened around the omnidirectional response until it becomes a figure-of-eight.

The frequency response also leaves something to be desired, since it has a distinct high-pass characteristic (see inset). That has little effect in the original application, since the so-called proximity effect comes into play, as always with devices sensitive to pressure gradient. This means that if we have a sound source in close proximity to the microphone, the low frequencies are boosted, which tends to cancel out the high-pass characteristic.

These effects have been taken into account in this new circuit and specifically compensated for. The new circuit has the following features:

- Switchable directional characteristic: omnidirectional, wide cardioid, cardioid, supercardioid, hypercardioid and figure-of-eight.
- Frequency response (for the far field) compensated to below 25 Hz (–3 dB)
- Close-talking compensation (so-called roll-off or low-cut filter) with switchable cutoff frequency
- Choice of jack socket output using 9 V battery supply or XLR output using 12 V to 48 V phantom power

Block diagram

The block diagram of the new pre-amplifier if **Figure 1** shows the two condenser microphone capsules Mic1 and Mic2 which are mounted back-to-back in the same way as described in the July/August 2002 article. The signal from Mic1 is first

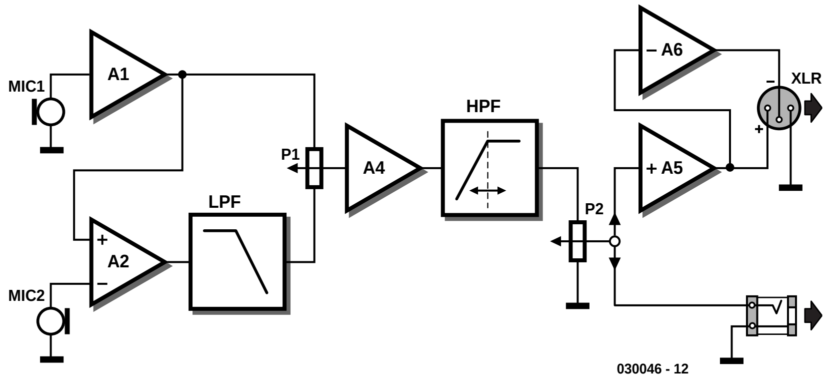


Figure 1. Block diagram of the circuit.

buffered by IC1.A and then taken, along with the signal from Mic2, to differential amplifier IC1.B. This forms a pressure gradient transducer from two pressure transducers. Next comes a low-pass filter (LPF) in the form of opamp IC1.C, which compensates for the high-pass characteristic that results from the arrangement of the microphones. The correct response for this filter depends on the physical distance between the two microphone diaphragms. Using the voltage divider P1 that follows, we can continuously adjust the output between (buffered) output signal of Mic1 (omnidirectional characteristic) and the output of the low-pass filter (figure-of-eight characteristic). This time the weighted sum does indeed produce the various cardioid characteristics, since the low-pass filter also introduces the correct phase shifts.

Next there is a further buffer amplifier (IC1.D), providing a low-impedance input to the high-pass filter (HPF) used for close-talking compensation. The frequency setting of this filter is switchable to allow for talkers at various distances from the microphone.

Finally we come to volume control potentiometer P2. At this point we can take off the output signal to a jack socket. It is, however, also possible to produce a 'studio style' balanced output signal on an XLR connector. The balanced output is generated by buffer IC2.A and inverting amplifier IC2.B.

Circuit diagram

It is easy to identify the various parts of the block diagram in the detailed circuit diagram shown in **Figure 2**. Note that the inputs to IC1.B have simply been exchanged in order to compensate for the inversion undergone by the signal in the low-pass filter IC1.C.

The TS925 device used is a quad low-noise rail-to-rail operational amplifier, with a relatively high output drive capability. Also, the TS925 includes a half-supply buffer (output on

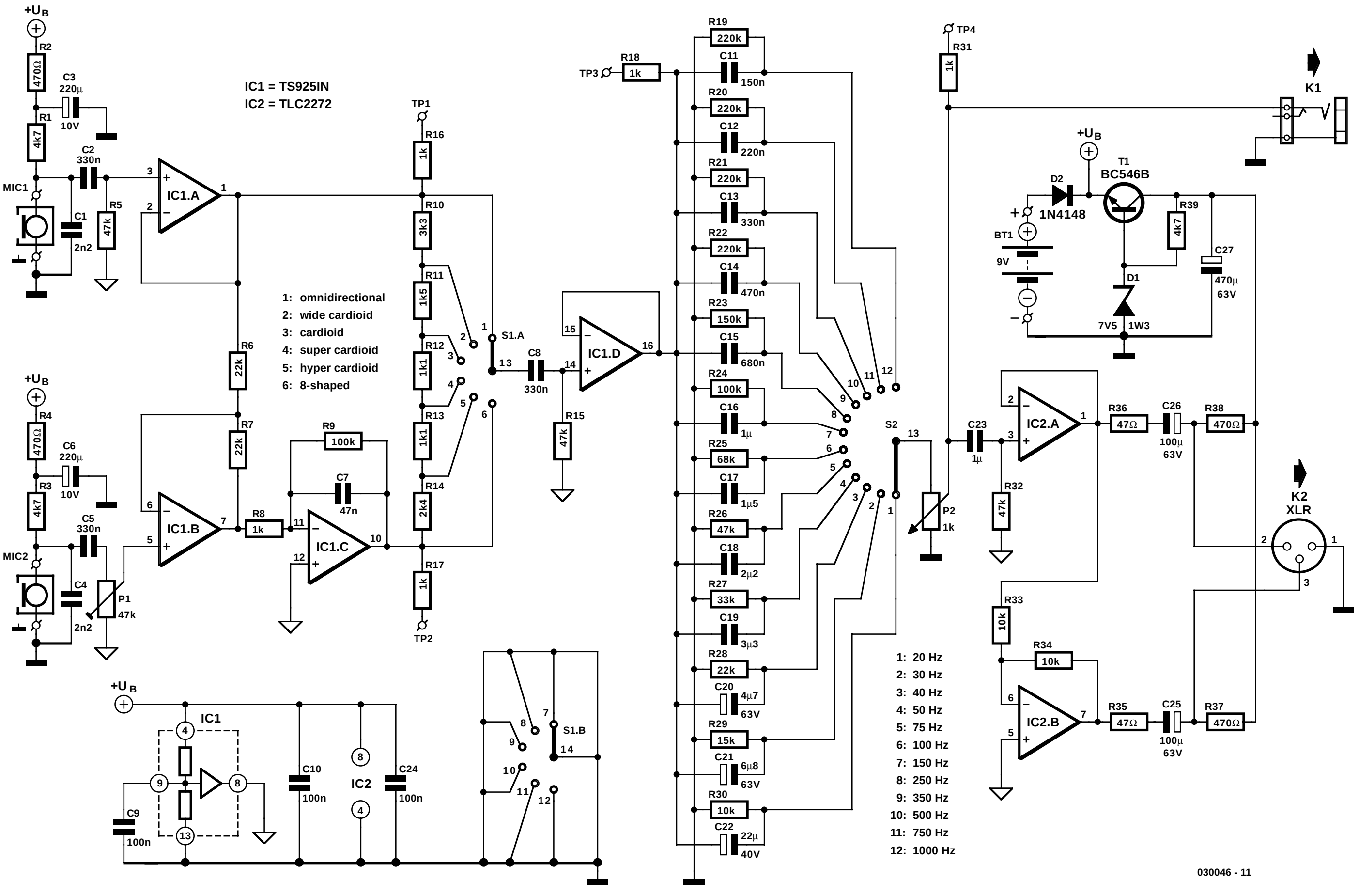


Figure 2. Detailed circuit diagram of the directional microphone.

COMPONENTS LIST

Resistors:
R1,R3,R39 = 4kΩ7
R2,R4,R37,R38 = 470Ω
R5,R15,R26,R32 = 47kΩ
R6,R7,R28 = 22kΩ
R8,R16,R17,R18,R31 = 1kΩ
R9,R24 = 100kΩ
R10 = 3kΩ3
R11 = 1kΩ5
R12,R13 = 1kΩ1
R14 = 2kΩ4
R19-R22 = 220kΩ
R23 = 150kΩ
R25 = 68kΩ
R27 = 33kΩ
R29 = 15kΩ
R30,R33,R34 = 10kΩ
R35,R36 = 47Ω
P1 = 47kΩ preset
P2 = 1kΩ potentiometer, logarithmic

Capacitors:
C1,C4 = 2nF2
C2,C5,C8,C13 = 330nF
C3,C6 = 220µF 10V radial
C7 = 47nF
C9,C10,C24 = 100nF
C11 = 150nF
C12 = 220nF
C14 = 470nF
C15 = 680nF
C16,C23 = 1µF 63V MKT, lead pitch 5mm or 7.5mm
C17 = 1µF5 63V MKT, lead pitch 5mm or 7.5mm
C18 = 2µF2 63V MKT, lead pitch 5mm or 7.5mm
C19 = 3µF3 63V MKT, lead pitch 5mm or 7.5mm
C20 = 4µF7 63V radial
C21 = 6µF8 63V radial
C22 = 22µF 40V radial
C25,C26 = 100µF 63V radial
C27 = 470µF 63V radial

Semiconductors:
D1 = zener diode 7.5V 1.3W
D2 = 1N4148
T1 = BC546B
IC1 = TS925IN (ST), e.g., Farnell
IC2 = TLC2272 (Texas Instruments)

Miscellaneous:
K1 = 3-way SIL pinheader + 6.3mm jack socket, chassis mount with separate ground contacts
K2 = 3-way SIL pinheader + XLR plug (male) chassis mount
S1 = 2-pole 6-way rotary switch, PCB mount
S2 = 1-pole 12-way rotary switch, PCB mount
BT1 = 9-V battery 6F22 with clip-on leads
MIC1,MIC2 = electret microphone element (e.g., Monacor MCE2000)
PCB, order code **030046-1**, (available from The PCBShop

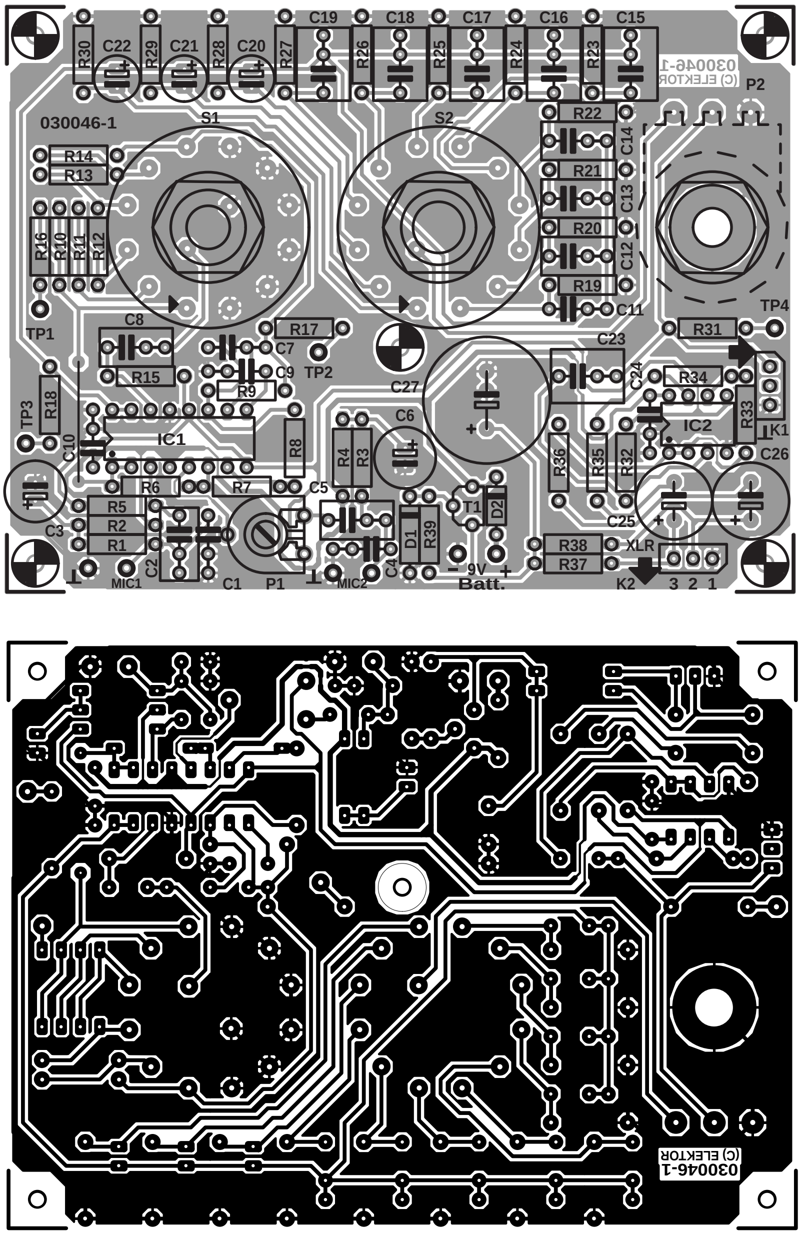


Figure 3. The printed circuit board features four test points.

pin 8), which can be used as a virtual ground. The selection of R8 depends on the distance *d* between the diaphragms of the two microphone capsules. Here we have assumed that this is 15 mm, which using the formula

R8 = *d* / (c·C3)

and taking *c*, the speed of sound, to be 344 m/s, gives us a value of 956 Ω. The value is not particularly critical, and we can round it off to 1 kΩ. P1 of the block diagram is implemented as a switched resistor ladder, which makes it easier to set a given directional characteristic.

The most complicated part appears to be the adjustable high-pass filter, but in fact it simply consists of twelve high-pass combinations (C11 to C22 in conjunction with P2) in parallel, each with a different cutoff frequency. S2 switches between the different frequencies. Resistors R19 to R30 do not play a significant part in determining the high-pass characteristic: they are there simply to charge the capacitors to half the supply voltage so that there are no transients on the output when the filter is switched. The value of volume control potentiometer P2 is rather low at 1 kΩ, so that the signal for the jack output can be taken directly from the wiper.

Both the microphone capsules and the output buffer IC2.A are connected via RC-combinations (C2/R5, C5/P1 and C23/R32). This attenuates the output below 16 Hz at 12 dB per octave. This ensures that IC1.C and IC1.D will not be overdriven.

Two opamps are connected in parallel with the jack output in order to produce a balanced (as well as buffered) output. IC2.A acts solely as a buffer, whereas IC2.B inverts the signal. The balanced signal is taken to the XLR connector via C25 and C26.

Phantom power

The power supply for the electronics and for the microphones can be obtained either from a 9 V PP3 battery or from a mixing desk that supplies phantom power. D2 prevents the power sources from interfering with one another. R37 to R39, C27, T1 and D1 extract the phantom power: the pure phantom supply voltage appears at the junction of R37 and R38, since the AC signal has been cancelled out. Then, after smoothing (C27) T1 and D1 form a simple, but entirely adequate, voltage regulator producing about 6.8 V. Because the ICs offer rail-to-rail operation, the circuit has plenty of output drive.

The circuit can be used with phantom power at the usual voltages of 12 V, 24 V or 48 V.

Construction and adjustment

Apart from the microphones, there is little to say about the construction of this circuit. The printed circuit board shown in **Figure 3** should ensure a stress-free soldering experience. Be careful not to omit the wire link near R18. There are four test points on the circuit board, which should be fitted with pins if measurements are to be carried out on the circuit.

The only adjustment to make is to compensate for the difference between the two microphone capsules using P1. This requires an oscilloscope or an audio-frequency millivoltmeter to be connected to the output of the circuit. Set the directional characteristic to figure-of-eight and the high-pass filter to its lowest frequency. A low-frequency (but at least 16 Hz) sinewave audio signal should be generated near to the microphones, with the axis of the microphone pair perpendicular to the direction of the sound source. A 50 Hz signal is suitable since it is easy to generate and feed to a stereo amplifier. The loudspeaker will not produce a perfectly undistorted sound, and so it is necessary to concentrate only on the fundamental component of the signal on the oscilloscope display. The output amplitude should be adjusted to a minimum using P1. It can be helpful to try adjusting the orientation of the microphone in order to find the minimum.

It is best to use a low frequency, because at higher frequencies sound reflections can easily mask the true direction of the sound source — at least, assuming that you do not have access to an anechoic chamber and that you do not want to do these experiments in the open air.

If you do not have the appropriate test equipment, then you can put your faith in the microphone manufacturers, assume that the capsules are identical, and set P1 to its centre position.

The microphone in use

The authors mounted the two microphones exactly as before (Figure 2 of the original article). However, rather than using the construction shown in Figure 3 of the article, we did not take the screened cable all the way to the capsules. Instead we used short flexible lengths of braid to connect to a two-core screened cable. The rubber tube holding the two capsules is suspended inside a spherical stainless steel mesh tea infuser (**Figure 4**) using elastic bands. The elastic bands, which are fixed to the edge of the infuser using three M2 screws, serve to reduce handling noise. The microphone cable is fed in obliquely via a strain relief, which is also fixed to a hole on the infuser (**Figure 5**). Electrically connecting the screen of the cable to infuser gives the required shielding from noise.

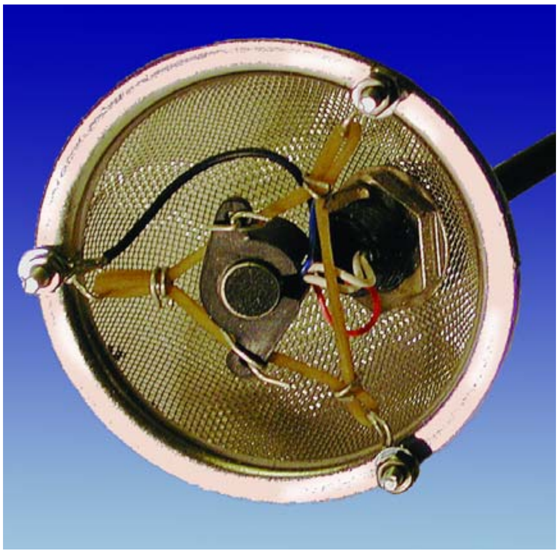


Figure 4. The microphone capsules are suspended using lengths of elastic.

Finally the two halves of the infuser are screwed together and a pop filter fitted over the whole assembly. For the prototype we used two large black foam pads from headphones, pulled over the halves of the infuser (**Figure 6**), so that the whole thing rather resembles a burnt doughnut. A hook-and-loop strap is fitted between the two halves to make the assembly easier to hold.

The electronics is contained in a separate enclosure, which can for example be clipped to the belt. Connection is via a two-core screened cable fitted with a 5-pin DIN plug, with pairs of contacts wired in parallel to

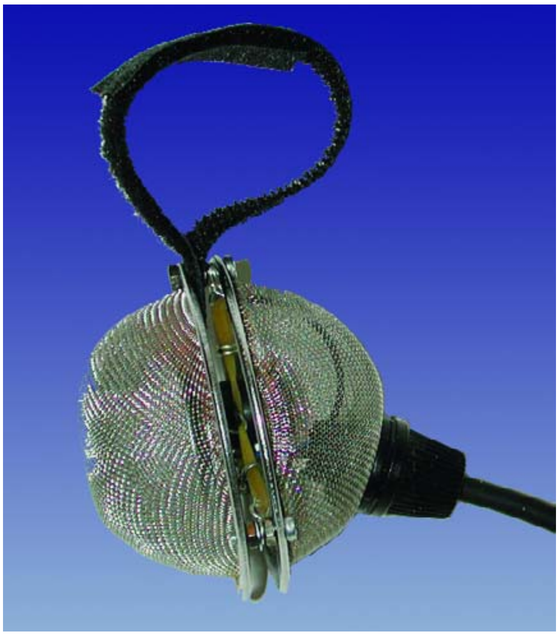


Figure 5. The prototype before the pop filter is fitted.

Figure 6. The completed microphone unit.

improve contact reliability. A chassis-mounted XLR plug is used for the output with phantom power supply, and a 6.3 mm jack socket, with an isolated switching contact, is used when under battery power. This saves the bother of an on/off switch, as well as that of remembering to use it.

In practical use on stage it will be necessary to experiment in each case to find which of the available directional characteristics makes for the least feedback. This will depend on the positions of the loudspeakers and monitors as well as on any sound-reflecting surfaces nearby. The cutoff frequency of the high-pass filter should be set appropriately. The proximity effect, for which this compensation is designed, is not always undesirable; sometimes its particular tonal characteristics are just what is wanted.

(030046-1)

References:

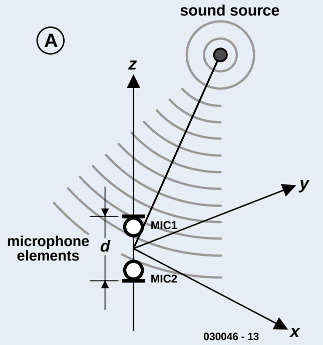
TS925 datasheet, ST Microelectronics.

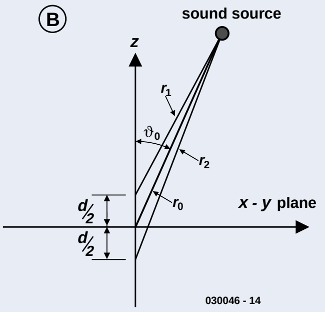
'Directional Microphone'
by Klaus Rohwer, Elektor Electronics
July-August 2002, page 84.

'Handbook of Mathematics'
by Bronstein and Semendyayev,
Springer-Verlag, Berlin, 2002

Calculating the directional characteristic and frequency response

We will use complex numbers in the following calculations, since they offer many advantages in this case. The calculations could also be carried out using real quantities, but would be rather more convoluted. As usual in engineering, we use *j*, rather than *i*, for the imaginary axis, since *i* can be confused with a current.





We choose a coordinate system where the diaphragms of the microphone capsules lie a distance *d* apart on the z-axis, symmetrically disposed with respect to (i.e. distance *d*/2 above and below) the x-y plane: see **Figure A** and **Figure B**. The sound source is at a distance r_0 from the origin of the coordinate system. Let the distances between the sound source and the diaphragms be r_1 and r_2.

These distances can be calculated using the cosine rule:

$$r_1^2 = r_0^2 + \left(\frac{d}{2}\right)^2 - r_0 \cdot d \cdot \cos\vartheta_0$$

$$r_2^2 = r_0^2 + \left(\frac{d}{2}\right)^2 - r_0 \cdot d \cdot \cos\vartheta_0$$

Here ϑ_0 is the angle between the z-axis and the line joining the origin and the sound source. The pressure *p* due to a sound wave, which spreads out in a spherically symmetric fashion, can be written as follows in terms of time *t* and distance *r*:

$$p = \frac{p_0}{r} \cdot e^{j(\omega t - kr)}$$

Here $\omega = 2 \cdot \pi \cdot f$ is the angular frequency of the sound and $k = \omega/c = 2 \cdot \pi \cdot \lambda$ is the number of cycles of the sound, with *c* being the speed of sound (344 m/s) and λ being the wavelength ($\lambda = c/f$). p_0 is a parameter to represent the intensity of the sound wave: it is not necessary to know this value for the calculation here. Equation (3) shows that the sound pressure is inversely proportional to the distance from the sound source. We now suppose that each microphone capsule produces a voltage proportional to the sound pressure at its diaphragm. We can now write these voltages as follows.

$$U_1 = \frac{U_0}{r_1} \cdot e^{j(\omega t - kr_1)} \qquad U_2 = \frac{U_0}{r_2} \cdot e^{j(\omega t - kr_2)}$$

Here U_0 is another undetermined parameter, which is proportional to p_0. It will drop out of the calculation later. It is convenient to separate out the explicit time dependency in these equations.and introduce a complex amplitude $\hat{A}$. Now we have

$$U_1 = \frac{U_0}{r_1} \cdot e^{-jkr_1} \cdot e^{j\omega t} \qquad U_2 = \frac{U_0}{r_2} \cdot e^{-jkr_2} \cdot e^{j\omega t}$$

Later we will need to resolve (for example) $\hat{U}_1$, and hence the complex amplitude, into its real and imaginary parts. We therefore

$$\hat{U}_1 := \frac{U_0}{r_1} \cdot e^{-jkr_1} \qquad \Rightarrow U_1 = \hat{U}_1 \cdot e^{j\omega t}$$

$$\hat{U}_2 := \frac{U_0}{r_2} \cdot e^{-jkr2} \qquad \Rightarrow U_2 = \hat{U}_2 \cdot e^{j\omega t}$$

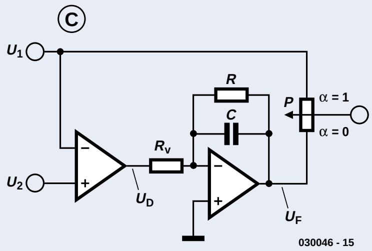
need to rewrite the complex exponential into its real and imaginary parts:

$$\hat{U}_1 = \frac{U_0}{r_1}\left(\cos kr_1 - j \cdot \sin kr_1\right) \qquad \Rightarrow \mathrm{Re}\left(\hat{U}_1\right) = \frac{U_0}{r_1}\cos kr_1, \quad \mathrm{Im}\left(\hat{U}_1\right) = \frac{U_0}{r_1}\sin kr_1$$

The amplitude is not the real part, but rather the modulus, of the complex amplitude (which we will need later):

$$\left|\hat{U}\right| = \sqrt{\mathrm{Re}\left(\hat{U}\right)^2 + \mathrm{Im}\left(\hat{U}\right)^2}$$

We now look at the theoretical circuit in **Figure C**. The difference voltage U_D is given by:



$$\hat{U}_D = \hat{U}_2 - \hat{U}_1 = U_0\left(\frac{1}{r_2}e^{-jkr_2} - \frac{1}{r_1}e^{-jkr_1}\right)$$

The equation for the output voltage of the low-pass filter, U_F, is derived by applying Kirchoff's law at the inverting input to the operational amplifier:

$$U_F = -U_D \frac{R}{R_V} \cdot \frac{1}{1 + j\omega RC}$$

At low frequencies the gain of this stage tends to the constant value

$$\nu := \frac{R}{R_V}$$

Write δ for the ratio between the frequency of interest and the cutoff frequency of the low pass-filter.

$$\delta := \omega RC$$

$$\Rightarrow U_F = -U_D \cdot \nu \cdot \frac{1}{1 + j\delta} = -U_D \cdot \nu \cdot \frac{1 - j\delta}{1 + \delta^2}$$

$$\Rightarrow \hat{U}_F = -U_0 \cdot \nu \cdot \frac{1 - j\delta}{1 + \delta^2} \cdot \left(\frac{1}{r_2}e^{-jkr_2} - \frac{1}{r_1}e^{-jkr_1}\right)$$

$$\Rightarrow \hat{U}_F = -U_0 \cdot \nu \cdot \frac{1 - j\delta}{1 + \delta^2} \cdot \left[\left(\frac{1}{r_2}\cos kr_2 - \frac{1}{r_1}\cos kr_1\right) + j\left(\frac{1}{r_1}\sin kr_1 - \frac{1}{r_2}\sin kr_2\right)\right]$$

$$\Rightarrow \mathrm{Re}\left(\hat{U}_F\right) = U_0 \cdot \frac{\nu}{1 + \delta^2} \cdot \left[\frac{\cos kr_1}{r_1} - \frac{\cos kr_2}{r_2} + \delta\left(\frac{\sin kr_2}{r_2} - \frac{\sin kr_1}{r1}\right)\right]$$

$$\Rightarrow \mathrm{Im}\left(\hat{U}_F\right) = U_0 \cdot \frac{\nu}{1 + \delta^2} \cdot \left[\delta\left(\frac{\cos kr_2}{r_2} - \frac{\cos kr_1}{r1}\right) + \frac{\sin kr_2}{r_2} - \frac{\sin kr_1}{r_1}\right]$$

[Potentiometer *P* lets us mix U_1 and U_F in variable proportions. Let α be the proportion of U_1. Then we have:

$$\hat{U}_A = \alpha\hat{U}_1 + \left(1 - \alpha\right)\hat{U}_F$$

From this point the calculation is best continued numerically. This can be done using an ordinary spreadsheet program such as Excel. First create a column for the parameter to be varied with space for several values. In the case of the directional characteristic this is ϑ_0, and in the case of frequency response this is *f*. Then create more columns for the intermediate quantities which are calculated using the above formulae. Generally spreadsheet programs cannot work with complex numbers, and so separate columns will be required for the real and imaginary parts of each complex quantity. It is a good idea to reserve a corner of the spreadsheet for the constant values such as r_0, α, *R*, R_V, and *C* (all in SI units: *m*, Ω, and *F*).

When calculating the directional characteristic, it is of course necessary to keep the frequency *f* constant (for example at *f = 1000 Hz*), and likewise to keep the angle of the incoming sound ϑ_0 constant (for example at *0*°) when evaluating the frequency response. And what about the unknown U_0? It can simply be set to *1*. In the penultimate column we can calculate the behaviour of $\hat{U}_A$ using equations (14) and (23); in the last column we can convert these values into decibels using the formula:

$$u = 20 \cdot \log\left(\frac{\left|\hat{U}_A\right|}{U_0} r_0\right)$$

Here we finally see the justification for setting U_0 to *1*, simplifying equation (24). *u* is now a suitable quantity for observing the dependence on the variable parameter. Spreadsheet programs usually offer a charting format suitable for the directional characteristic, such as the polar plotting function in Excel. Including the factor of r_0 allows the easy comparison of results with different distances between sound source and microphone. If the factor were not included, the value of *u* would be highly dependent on this distance.

It is of course interesting to vary the mixing proportion α. The following values give the directional characteristics shown:

α	Directional characteristic
0	Figure-of-eight
0.25	Hypercardioid
0.37	Supercardioid
0.5	Cardioid
0.66	Wide cardioid
1	Omnidirectional

The characteristic is best approximated when

$$R_V = \frac{d}{cC}$$

The optimal value of R_V depends on the distance *d* between the two diaphragms. *R*, on the other hand, affects the cutoff frequency of the low-pass filter and hence the usable bandwidth of the microphone, as well as the gain (see equation 17). The value of *R* is therefore always a compromise between wide bandwidth and the risk of noise due to the high gain.

An interesting aspect is the effect of varying r_0. We can clearly observe the proximity effect, which, with α = *0* and with a small value of r_0, gives a boost to the lower frequencies.

By setting *C* = *0* and *R* = R_V, we can simulate the circuit of the July-August 2002 issue without a low-pass filter. Here we can see the strong dependence of output signal level on frequency, even at greater values of r_0.

An excessive amount of trust should not be put in the results of these calculations. We have not taken into account that in reality the diaphragms are not points, but have a definite extent. This has a particular impact at high frequencies. Also, the microphone capsules are not ideally positioned in free space, but are generally surrounded by a grille and a pop filter. This affects both the frequency response and the directional characteristic. These calculations, then, are not a substitute for a real test in an anechoic chamber, but in the absence of such facilities, can at least provide some useful guidance.

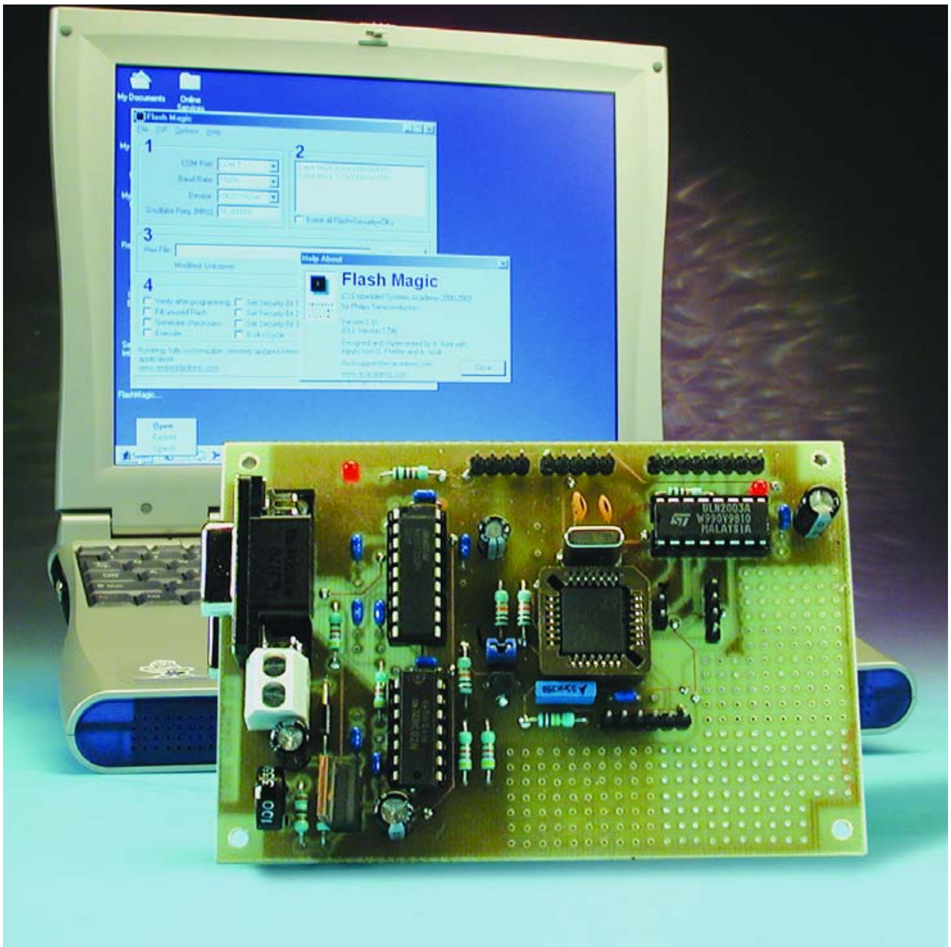
The P89LPC900 (2)

Part 2 (final): practical application

M. Stottelaar

Eurodis Texim Electronics

In this second instalment we will look at the actual application of this microcontroller in greater detail. Notably the ‘in system programming’ functionality is described in detail. In addition, a practical design is presented that not only functions as a programmer for the controller, but as a prototyping board as well. As an example project, guided by an application note from Philips, a very good ADC can be built without requiring any additional peripherals.



Addition to Part 1 (October 2003)

Hitex are the sole distributor of the Keil/Philips LPC kits the UK. The kits are available from www.hitex.co.uk/shop. We thank Mr. Chris Hills of Hitex UK for bringing this to our attention.

Flash memory

Many of you will recognise the ceramic microcontroller with a glass window. These are the UV erasable versions that were (and sometimes still are) used during the debugging stage of the application software. An expensive solution indeed, but the alternative, the burning of one-time programmable (OTP) devices was often more costly and far from flexible. The arrival of multi-time programmable flash (MTP) has made the UV-erasable type mostly obsolete. With that, the relatively long UV erasure time is now also a thing of the past.

Despite this large step forward, there is another highly desirable feature: it would be very nice if the microcontroller could be programmed and erased while it

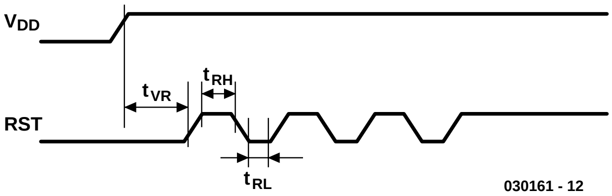
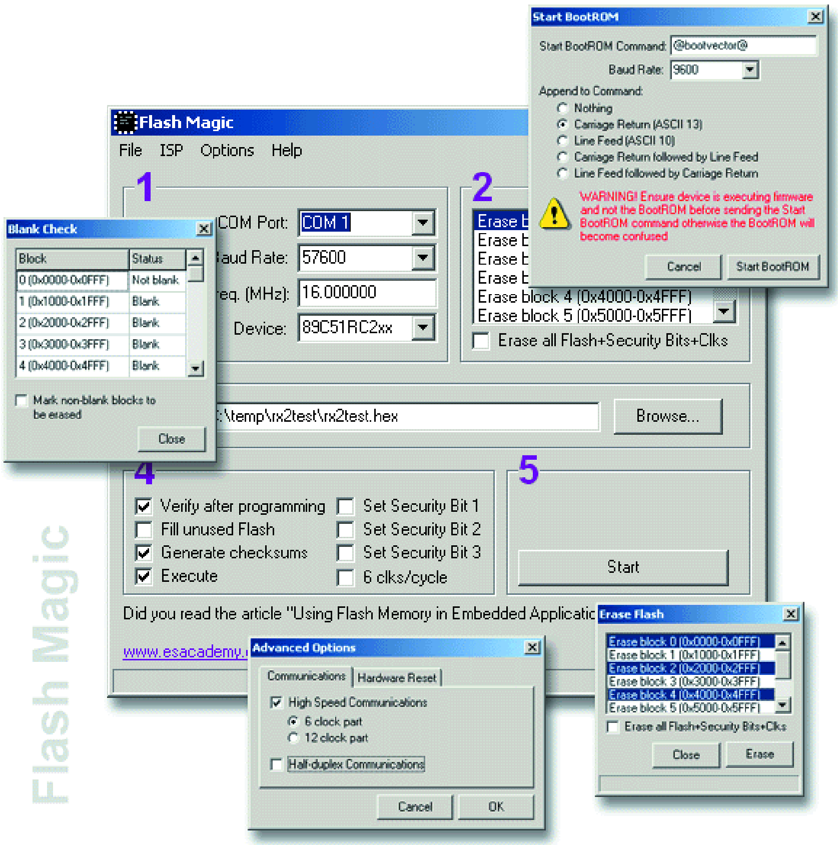


Figure 1. Three specific pulses at the Reset pin put the chip in boot-mode.

remains in the application circuit. In particular with SMD packages this very desirable of course. And if it were also possible to replace the code ‘in the field’, then all of our wishes would have come true.

Two pieces of code

All the recent versions of microcontrollers from Philips Semiconductors have the option of being (re)programmed while fitted in the circuit, and this includes the P89LPC900 series. Because nearly all versions have at least one UART, this is an eminently suitable method to provide the microcontroller with code. The only other things that are necessary are a PC running a small application and a level shifter to convert the RS232 signal levels to TTL logic levels. All the programming functionality is controlled by two pieces of code that reside in the microcontroller. The *low-level* part takes care of the actual reading and writing of the memory, while the *high-level* part (boot-loader) provides for the communications via the UART. The low level functions are available to a programmer and can be used to write your own boot loader, which would be able, for example, to reprogram the microcontroller via a CAN-network.
The P89LPC900 is incidentally quite happy with only one power supply voltage during programming, so there is no need for an additional programming voltage. The next few paragraphs provide more detail about the inner workings of the various elements.



COM Port: COM 1
Baud Rate: 57600
Freq. (MHz): 16.000000
Device: 89C51RC2xx

Erase block 4 (0x4000-0x4FFF)
Erase block 5 (0x5000-0x5FFF)
□ Erase all Flash+Security Bits+Clks

:\temp\rx2test\rx2test.hex

☑ Verify after programming
□ Fill unused Flash
☑ Generate checksums
☑ Execute
□ Set Security Bit 1
□ Set Security Bit 2
□ Set Security Bit 3
□ 6 clks/cycle

Did you read the article "Using Flash Memory in Embedded Applicatic

www.esacademy.c

Block	Status
0 (0x0000-0x0FFF)	Not blank
1 (0x1000-0x1FFF)	Blank
2 (0x2000-0x2FFF)	Blank
3 (0x3000-0x3FFF)	Blank
4 (0x4000-0x4FFF)	Blank

□ Mark non-blank blocks to be erased

Start BootROM Command: @bootvector@
Baud Rate: 9600
Append to Command:
○ Nothing
● Carriage Return (ASCII 13)
○ Line Feed (ASCII 10)
○ Carriage Return followed by Line Feed
○ Line Feed followed by Carriage Return

WARNING! Ensure device is executing firmware and not the BootROM before sending the Start BootROM command otherwise the BootROM will become confused

☑ High Speed Communications
● 6 clock part
○ 12 clock part
□ Half-duplex Communications

Erase block 0 (0x0000-0x0FFF)
Erase block 1 (0x1000-0x1FFF)
Erase block 2 (0x2000-0x2FFF)
Erase block 3 (0x3000-0x3FFF)
Erase block 4 (0x4000-0x4FFF)
Erase block 5 (0x5000-0x5FFF)
□ Erase all Flash+Security Bits+Clks

Figure 2. Menu structure of Flash Magic.

In System Programming (ISP)

When it leaves the factory, the P89LPC900 has been pre-programmed with an enabled boot loader. When power is applied the microcontroller is ready to receive the application code. After the code has been sent, the microcontroller is switched to *user-mode*, so that the application code will be executed following the next reset.

There is, of course, a method to return the microcontroller to boot mode again. The P89LPC900 actually has multiple methods to do this, two of which will be described here. A third method is outside our scope; it is also possible to put the processor in boot mode using a parallel programmer.

The first method of putting the chip in boot mode is to apply three specific pulses to the reset pin of the processor. In practice, the serial port of a PC can be used to generate these pulses. In addition to the usual RxD and TxD signals a third line is required to generate these pulses (refer to **Figure 1**).

The second, possibly even simpler, method uses the *Break Detect* function of the microcontroller. If this function has been enabled, the TxD line from the PC can put the microcontroller in boot mode. This means that the *Break Command* has to be sent to the microcontroller. This is certainly a very convenient method during the debugging stage of the software development cycle, particularly if the UART is accessible to the world via a sub-D connector. However this method also has its risks. The wave shape for the Break signal is rather simple and there is a significant risk that the controller will inadvertently enter boot mode. It is therefore recommended to disable this feature when the project has been completed. Software examples for enabling the break detect feature can be found in the application notes from Philips.

In part 1 we mentioned that Flash Magic (refer to **Figure 2**) is the name of the PC software used to program the microcontroller. This software is maintained by ESACADEMY (www.esacademy.com) and is periodically updated with a new list of microcontrollers to be programmed. The design is very simple and the environment has the look and feel of a typical programmer (note that Flash Magic is also suitable for other derivatives from the 80C51 family from Philips).

The circuit described below is suitable for use with Flash Magic and includes all the fea-

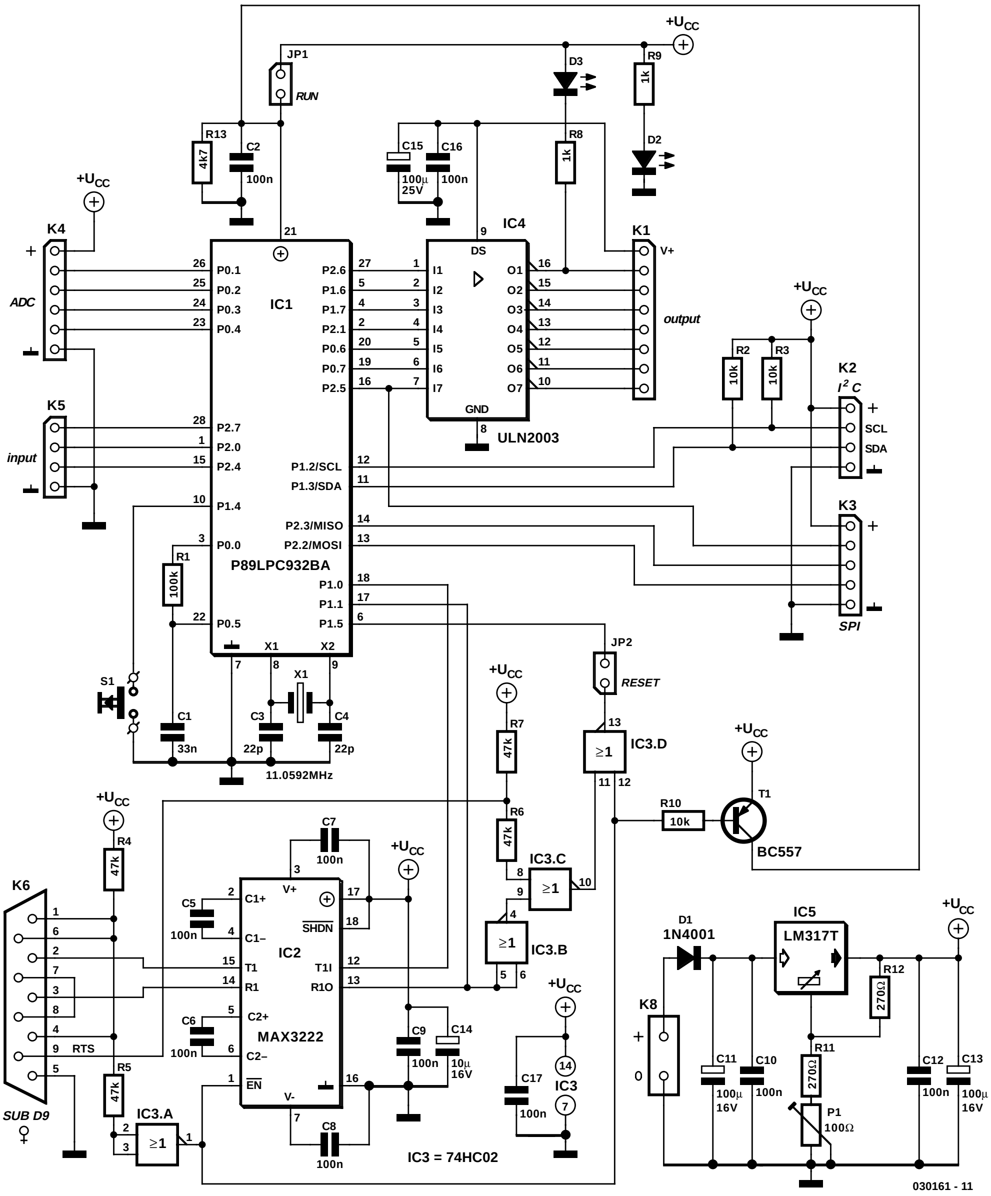


Figure 3. The schematic for the programmer and prototyping board.

tures for putting the microcontroller in bootmode. Flash Magic can also be operated from the command line. This makes enables you to use this tool as part of an automated production process.

It is also possible, of course, to write your own programming environment to suit your own unique requirements. For this purpose, the Intel Hex strings are supplemented with additional instructions. It is beyond the scope of this article to describe all these instructions in detail, but the P89LPC932 user manual provides all the necessary information.

In Application Programming (IAP)

In some high-end applications, there may be a need to replace the code in the microcontroller without replacing the IC itself. Of course, the ISP method is a possible solution in these situations. But what about an environment where there is no technical know-how or nobody is actually present?

In all these situations, the IAP-method can be used to replace the code using a method chosen by the application programmer. The communications method can be selected to suit the environment (telephone line, Internet, RF, etc.). The principle is very simple. The programmer defines the communication method and makes use of the so-called *IAP calls* in the microcontroller to carry out the actual erasing and writing operations. Obviously, part of the code needs to remain active during this process in order to maintain the communications channel. However, using segmentation, all the code can be replaced in two or more steps (in many cases the system software can be left unaltered because it is generally subjected to fewer changes than the actual application code).

A practical description of the IAP operation may be found on the *Elektor Electronics* website under Free Download file number **030161-2** (November 2003) on www.elektor-electronics.co.uk.

The hardware

That was enough theory for now. The circuit shown in **Figure 3** makes it possible to get started with the P89LPC900 family in a practical way. The design supports the previously mentioned ISP and IAP facilities, so that the testing of your own programs can be realised very easily. A C compiler, for example, can be used for the development of the software. If you don't have a compiler, one can easily be downloaded from the Internet (for example Tasking or Keil). In general these are 'full featured' demo versions, but limited with respect to the maximum generated code size. However, a 2k limited version is often adequate to write a realistic application. Also, the 'Code Architect' from Esacademy can be very helpful when first treading in 80C51 territory. Using this tool, you can interactively write code for the P89LPC932 that has been used in this design. This member of the family is part of a whole range of controllers with 8 to 28 pins and also available in PLCC packages.

To maximise the functionality of the board, all the I/O from the microcontroller has been brought to the outside and is available for use. The combination of the serial interface and analogue and/or digital I/O can easily be put to practical use to collect measurements with a PC or control some outputs.

In order to program the microcontroller, the simplest connection between the PC and P89LPC932 could consist of a single RS232 level shifter that we have used on many previous occasions in *Elektor Electronics* designs. The RXD and TXD lines from the PC and P89LPC932 are simply connected via the level shifter. In this design we have chosen for a slightly more elaborate setup. Not only are the RXD and TXD lines connected, but some additional signals as well, to provide a safe method of switching the programmer on and off, and for generating the clock pulses at the reset pin (for reboot).

COMPONENTS LIST

Resistors:
R1 = 100kΩ
R2,R3,R10 = 10kΩ
R4...R7 = 47kΩ
R8,R9 = 1kΩ
R11 = 390Ω
R12 = 270Ω
R13 = 4kΩ7
P1 = 100Ω preset V

Capacitors:
C1 = 33nF
C2,C5-C10,C12,C16,C17 = 100nF
C3,C4 = 22pF
C11,C13,C15 = 100μF 25V radial
C14 = 10μF 16V radial

Semiconductors:
D1 = 1N4001
D2,D3 = low-current LED
T1 = BC557
IC1 = P89LPC932BA
IC2 = MAX3222CPN
IC3 = 74HC02
IC4 = ULN2003
IC5 = LM317T

Miscellaneous:
JP1,JP2 = jumper
K1 = 8-way SIL pinheader
K2,K5 = 4-way SIL pinheader
K3 = 5-way SIL pinheader
K4 = 6-way SIL pinheader
K6 = 9-way sub-D socket (female), PCB mount
K8 = 2-way PCB terminal block, lead pitch 5mm
S1 = pushbutton, one make contact
X1 = 11.0592MHz quartz crystal
PCB, available from The PCBShop

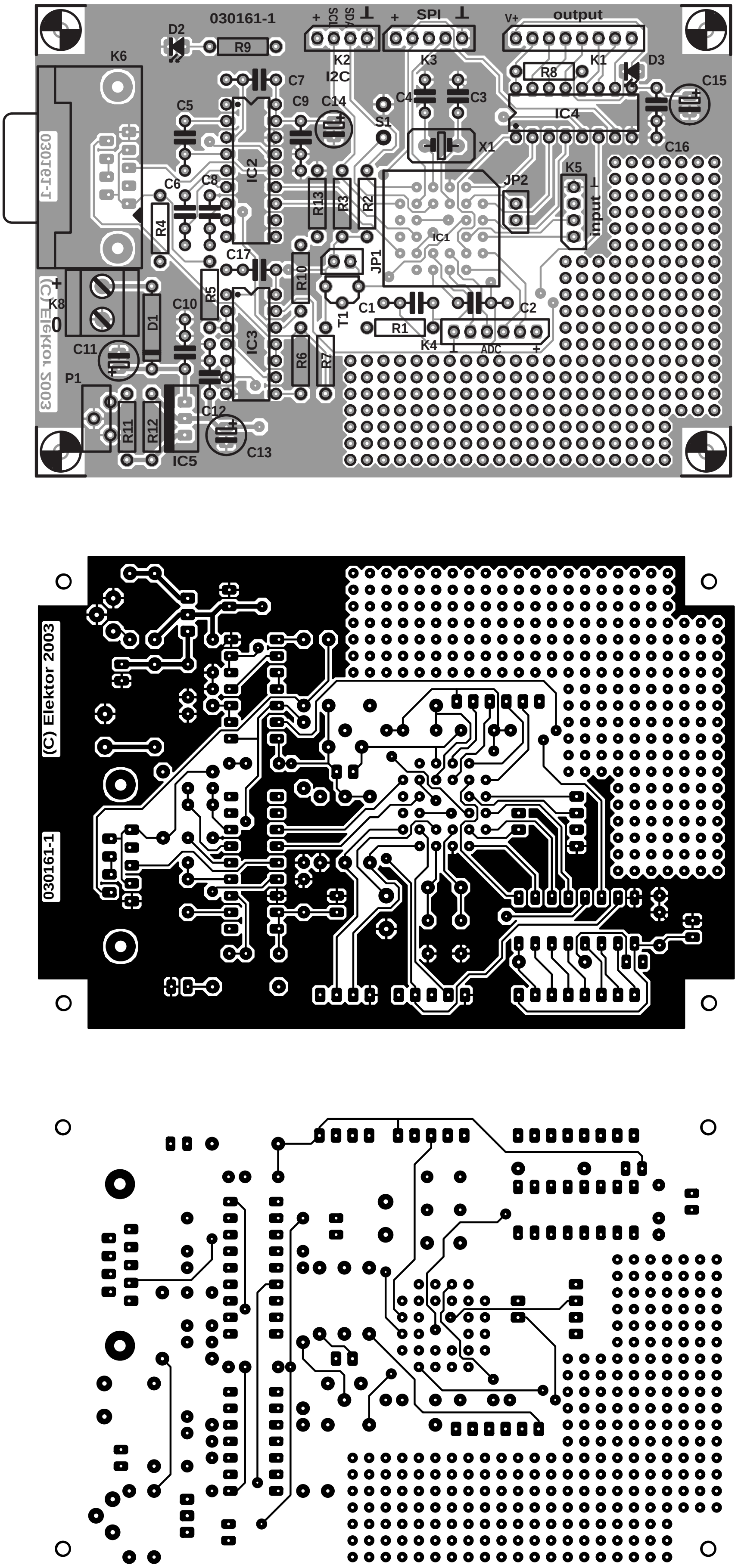


Figure 4. There is plenty of room on the PCB for your own applications.

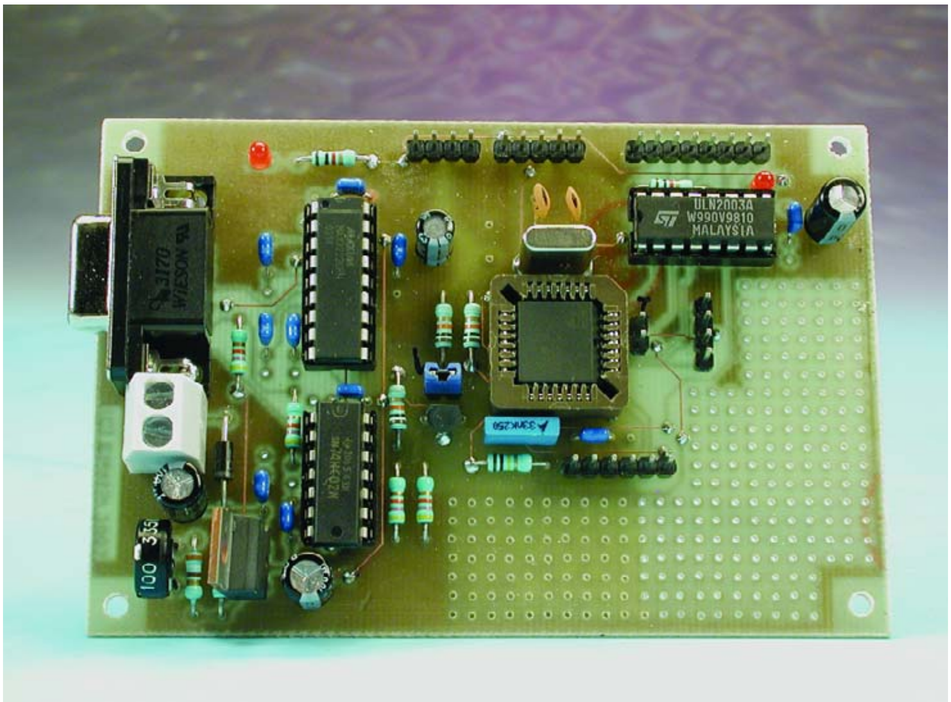


Figure 5. The construction of the prototyping board is a relatively simple job.

IC2 converts the signal levels and is provided with an enable input, which causes the chip to go completely off-line when the DTR signal is no longer active. The DTR line is also used, via IC3 and T1, to switch the power supply voltage to the P89LPC932 on and off during the programming session. The three pulses that are required to put the microcontroller in boot mode are generated using the RTS line from the PC, combined with the TXD signal.

A few more schematic details

R1 and C1 form the circuit that's used for the sigma-delta converter to be described later on. P0.1 through P0.4 are used as analogue inputs for this purpose, so that a four-channel ADC is created. Fitting C3, C4 and X1 is not strictly necessary, because the processor has a sufficiently accurate RC oscillator built in. These connections can be used as digital I/O. The signals for both I^{2}C and SPI are also brought out to pins so that many popular chips can be connected directly.

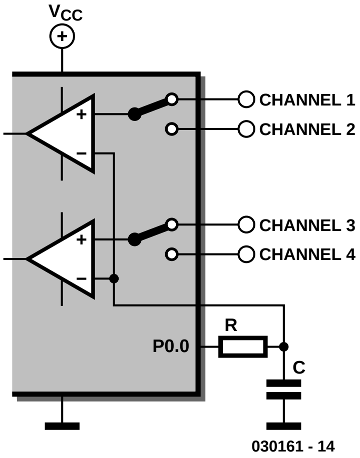


Figure 6. The switchable comparator inputs provide the option of four ADC channels.

The four compare outputs (pins 2 through 5 of K1) and another three ports are buffered and made available via IC4, allowing relays or similar to be controlled directly. Port P2.6 (pin 2 on K1) is fitted with an LED (D3) that can be useful during software development. The remaining ports, including P2.7, P2.0 and P2.4 are available as general-purpose I/O (on K5).

Note that P1.5 can never be configured as an output and in this application is only used as Reset.

A pushbutton (S1) is connected to port 1.4. This could be used, for example, to test the software example of boot-mode via IAP. Even though the processor is a 3-V type, all the inputs and outputs are tolerant of 5-V signals. It may be necessary in some cases to convert the signal levels if they do not conform to the input requirements of some optional external chips.

The power supply of the circuit is a very simple design. We chose an LM317 (IC5) with a trimpot, because it is more readily available than a fixed 3.3V-regulator. Before placing the processor in its socket check that the power supply voltage is around 3.3 V, adjustable with P1.

Construction

Figure 4 shows the (double sided) PCB designed for the prototyping board. The layout is quite spacious and conveniently arranged, so that even less experienced hobbyists will have no problem with the construction, provided you stick to the component overlay and parts list. Microcontroller IC1 is fitted in a special PLCC socket. The various connecters have been placed near the edge of the PCB as much as is possible.

A section beside the controller has been deliberately designed as prototyping board. This provides an easily accessible space for your own applications. In this way you can carry out a lot of different experiments without immediately having to make another PCB.

A standard 9-VDC mains adapter can be used to power the entire circuit. There are no specific requirements for the adapter, since IC5 provides voltage regulation and the current consumption of around 20 mA is not very large. Diode D1 prevents an incorrectly connected adapter causing damage to the circuit.

The programming

After everything has been put together and carefully checked, the PCB can be connected to the power supply and serial port. Use the following jumper settings:

	During Flash Magic	During Run-mode
JP2	Fitted	Removed
JP1	Removed	Fitted
N.B.	Reset and power controlled with Flash Magic	User-reset and uncontrolled power

Although Flash Magic is a program that's self-explanatory to a large extent, it is worthwhile to take a quick look at the settings related to *Start Bootrom* in the ISP menu. A number of things depend on whether *Break Detect* or *Pulses on Reset Pin* is used. The latter option will always work, provided the reset jumper is fitted and in the menu 'Options/Advanced Options/Hardware Config' the tickbox 'Use DTR and RTS to enter ISP-mode' has been ticked. The other method will only work when the (running) code has provision for this option, as described earlier.

Ex factory, the boot-loader is already enabled so that loading the first program will succeed even without explicitly enabling the boat loader. Consult the accompanying PDF-file for more instructions with regard to Flash Magic.

Attention: up to version C of the P89LPC932 it was not possible to activate the boot loader using the reset pin. With these devices you will always have to make use of either the *Break Detect* or *IAP* method. So be careful and convince yourself that your code is correct. The only method of escape once you're stuck is by using a parallel programmer to clear the chip after a mistake. Fortunately only a small number of these devices have been manufactured.

ADC

As already mentioned, the above design has a feature that allows an ADC to be realised even though the P89LPC932 does not have an *on-board* ADC. The principle of operation is based on the *Sigma-Delta* method. The two analogue comparators that are standard peripherals on this controller are employed for this purpose. Because each comparator has a selectable input (refer to **Figure 6**), it becomes possible to make a four-channel ADC. Application Note AN10187Ω1, available from the Philips website provides all the necessary information. But, briefly, the circuit operation is essentially as follows: C1 on the reference input is charged via R1 until the level is equal to the input of the comparator. From then on, the charge/discharge cycle is controlled by the result from the comparator. The number of charge cycles is proportional to the comparator input voltage. By counting the number of charge cycles during a predetermined number of cycles (charge and discharge) the level of te input voltage can be determined. For a more detailed description and background theory we refer you to the aforementioned application note.

In conclusion

It would go too far to describe all the details of the P89LPC932. Fortunately, there is plenty of code available for the 80C51 (often well documented in application notes). In addition, new application notes frequently appear on the Philips website, explaining the details of specific peripherals.

(030161-2)

CORRECTIONS & UPDATES

Atmel AVR Micro Programmer

September 2001, p. 52, 010005-1

The author has come up with another new release of the software, which may be downloaded free of charge from our website. The old version contained a bug causing incorrect programming of the last few bytes in the 4K device (4051).

Minimalist Induction-Balance Metal Detector

October 2003, p. 60, 020290-1

Unfortunately a few mistakes have been made in the numbering of three ICs on PCB 020290-1. This affects the component overlay of the board (Figure 2).

Position IC1 on the printed circuit board should be IC4 (LM317).
Position IC2 on the printed circuit board should be IC1 (7555).
Position IC4 on the printed circuit board should be IC2 (7555).

DDS RF Signal Generator

October 2003, p. 14, 020299-1

The values of R7 and R8 should be interchanged in the schematic and the parts list.
The correct descriptions are: R7 = 86Ω6; R8 = 43Ω2.

Stepper Motors Uncovered (1)

Part 1: stepper motor fundamentals

By Timothy G Constandinou

Stepper motors, and in particular the way they must be driven, continue to riddle many of our readers. This two-part article is a solid base for your understanding of stepper motor operation as well as practical use. Next month, the article culminates in the construction of a high-power 4-channel motor driver with PC control.

Often a mechanical operation is required in an application to accurately position a load or device. While a conventional DC or AC motor can be used, it is difficult to determine the exact position of the load, motor speed, or how much total motion has been produced, unless external positioning sensors, encoders, servo loops, and controlling devices (brakes or clutches) are used.

The preferred alternative is to use a stepper motor. This is an electromechanical device, which converts pulses into discrete mechanical movements.

Typical stepper motor applications…

include computer peripherals (scanners, disk drives, printers, etc), cameras, telescopes and satellite dish positioning systems, medical applications, robotics, barcode scanners and numerically controlled machine tools.

Types of stepper motors

At first blush stepper motors seem to differ only in respect of size, weight and the number of wires or connecting pins. A suggested classification is given below.

Variable reluctance (Figure 1)
This type of motor does not use a permanent magnet. As a result, the rotor can move without constraint or 'detent' torque. This type of construction is the least common and is generally used in applications that do not require a high degree of torque, such as the positioning of a micro slide.

Permanent magnet (Figure 2)
Also referred to as a 'canstack' or 'tincan' motor, this device has a permanent magnet rotor. It is a relatively low speed, low torque device with large step angles of either 45 or 90 degrees. The simple construction enables these motors to be produced at low cost, making them the ideal choice for low power applications,

Features of Stepper Motors

- Rotation angle is proportional to number of input pulses.
- Rotational speed is proportional to the frequency of input pulses.
- Open loop system with no position feedback required.
- Excellent response to acceleration, deceleration and step commands.
- Non-cumulative positioning error (±5% of step angle).
- Excellent low speed and high torque characteristics without need for gear reduction.
- Holding torque when energized.
- Inherent detent torque.
- Bi-directional operation.
- Can be stalled without motor damage.
- No brushes for longer trouble free life.
- Precision ball bearings (depending on brand/type).

Drawbacks of Stepper Motors

- Resonances can occur if not properly controlled.
- Not easy to operate at extremely high speeds.
- If overtorqued, all knowledge of position is lost and system must be reinitialized.
- Produces much less torque, for a given size than the equivalent DC/AC motor.

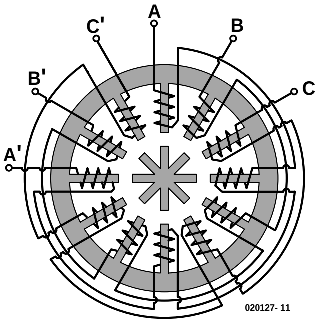


Figure 1. Variable-reluctance stepper motor.

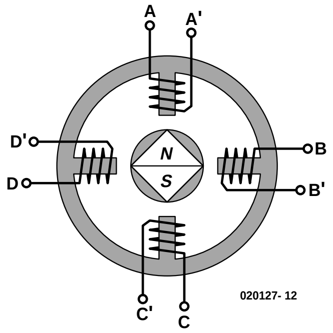


Figure 2. Permanent-magnet stepper motor.

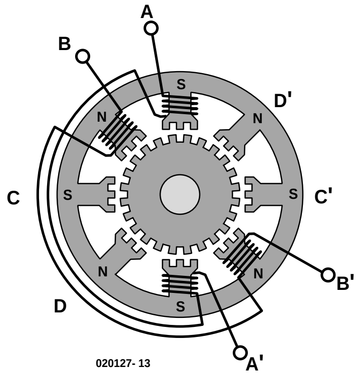


Figure 3. The hybrid stepper motor is a clever combination of the variable-reluctance and permanent-magnet types.

such as in inkjet printers.

Hybrid (Figure 3)

This combines the best characteristics of the variable reluctance and permanent magnet motors. They are constructed with multi-toothed stator poles and a permanent magnet rotor. Hybrid motors are available with step resolutions of 0.9°, 1.8° or 3.6°, the standard being 1.8°. Because they exhibit high static and dynamic torque and run at very high step rates, hybrid motors are the preferred type for industrial applications.

Size

Generally stepper motors are classified according to their frame size (body diameter). For example, a size 23 stepper motor has a body size of approximately 2.3 inches. The most common frame sizes are 11, 17, 23, 34 and 42.

Power

Power levels for stepper motors range typically from a few hundred milliwatts (for smaller motors) up to several watts (for larger motors). The maximum power dissipation of a stepper motor is determined by the thermal limits of the windings in the motor. This is seldom clearly stated in the motor manufacturer's data. To determine this we must apply the relationship $P = V I$. For example, a size 23 step motor may be rated at 6 V and 1 A per phase. Therefore with two phases energized the motor has a rated dissipation of 12 watts. It is normal practice to rate a stepper motor at the power dissipation level where the motor case rises to 65 °C above the ambient temperature. Therefore, if the motor can be mounted to a heatsink it is often possible to increase the allowable power dissipation.

More aspects of stepper motors

What else do we need to know about the stepper motor before we're able to make it turn and do something useful?

The Rotating Magnetic Field

When a phase winding of a stepper motor is energized with current, a magnetic flux is developed in the stator. The rotor then aligns itself so that the flux opposition is minimized. As a result, in requiring the motor to rotate, the windings must be energized in a sequence to produce a rotating magnetic flux.

Torque Generation

The torque produced by a stepper motor depends on several factors, including step rate, drive current in the windings and drive design or type.

In a stepper motor a torque is developed when the magnetic fluxes of the rotor and stator are displaced from each other. The stator is made up of a high permeability magnetic material, which results in the magnetic flux being confined to the stator structure. This serves to concentrate the flux at the stator poles. The torque produced by the motor is proportional to the intensity of the magnetic flux generated when the winding is energized.

Phases

The number of different paths for the magnetic flux to flow in is referred to as the number of phases of the motor.

- Generally, stepper motors have two phases even though motors exist with three or five phases.

– A bipolar motor has one winding per phase.
– A unipolar motor has a winding with a centre-tap, or two separate windings per phase.

Winding Connections

Stepper motors are produced in a number of different lead configurations. The most popular are:

Stepper Leads	Connection Schemes
4 Lead	Bipolar
5 Lead	Unipolar
6 Lead	Unipolar Bipolar (series connected)
8 Lead	Unipolar Bipolar (series connected) Bipolar (parallel connected)

The associated wiring diagrams are given in **Figure 4**.

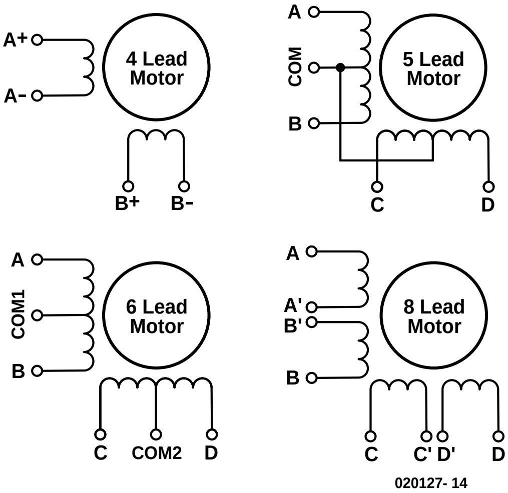


Figure 4. Basic connections of 4-, 5-, 6- and 8-lead motors.

Poles

A pole can be defined as one of the regions in a magnetized body where the magnetic flux density is concentrated. Usually both the rotor and stator of a stepper motor have an equal number of poles, even though this is not always the case.

Stepping Angle

The full-step angle of a stepper motor is determined by the following relationship:

Step angle = $360 / (n_{Ph} \times ph) = 360 / n$.

where
n_{Ph} = number of equivalent poles per phase = number of rotor poles;
ph = number of phases;
n = total number of poles (all phases).

This relationship is only valid if the number of stator and rotor poles are equal.

Stepping modes

The stepper motor can be driven in a number of different sequences. The most common of these are:

Wave Drive

In this mode only one phase is energized at any given time. For unipolar motors this means only 25% of the available windings are utilized, or 50% utilization for bipolar motors.

Order	Phase A	Phase B	Phase $\overline{A}$	Phase $\overline{B}$
1	■			
2		■		
3			■	
4				■

Full Step Drive

In this mode two phases are energized at any given time. For unipolar motors this means 50% of the available windings are utilized, or 100% utilization for bipolar motors.

Order	Phase A	Phase B	Phase $\overline{A}$	Phase $\overline{B}$
1	■	■		
2		■	■	
3			■	■
4	■			■

Half Step Drive

In this mode the sequences of the wave and full-step drives are interleaved to enable the rotor to be aligned in half steps. For unipolar motors this means 37.5% of the available windings are utilized (on average) or 75% utilization for bipolar motors.

Order	Phase A	Phase B	Phase $\overline{A}$	Phase $\overline{B}$
1	■	■		
2		■		
3		■	■	
4			■	
5			■	■
6				■
7	■			■
8	■			

Microstepping

The currents in the windings are continuously varied to be able to break up one full step into many smaller discrete steps. This excitation sequence will not be covered in this article.

Technical data and terminology

Holding Torque

The maximum steady torque that can be applied to the shaft of an energized motor without causing continuous rotation.

Detent Torque

The maximum torque that can be applied to the shaft of a non-energized motor without causing continuous rotation.

Speed/Torque Curve

The speed/torque characteristics of a stepper motor are a function of the drive circuit, excitation method and load inertia.

Maximum Starting Frequency

The maximum pulse rate (frequency) at which an unloaded stepper motor can start and run without missing steps, or stop, without taking more steps than pulses.

Pull-out Torque

The maximum torque that can be applied to the shaft of a step motor (running at constant speed) and not cause it to lose step.

Pull-in Torque

The maximum torque at which a step motor can start, stop and reverse the direction of rotation with-

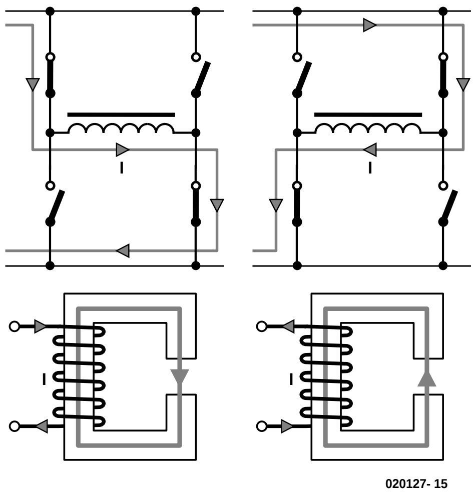


Figure 5. Bipolar drive scheme.

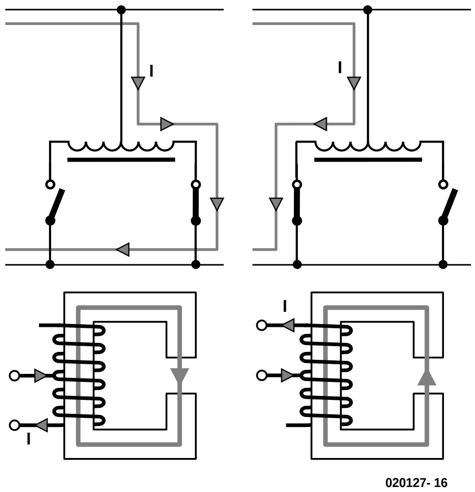


Figure 6. Unipolar drive scheme.

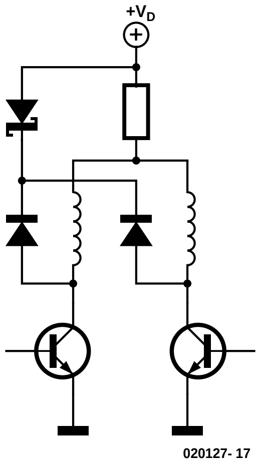


Figure 7. Series resistors used to limit the motor coil current.

out losing step.

Accuracy

This is defined as the difference between the theoretical and actual rotor position expressed as a percentage of the step angle. Typically this is approximately ±5% and this positioning error is non-cumulative.

Hysteresis Error

This is the maximum accumulated error from theoretical position for both forward and backward direction of rotation.

Resonance

A stepper motor responds to a series of input pulses, each pulse causing the rotor to advance one step. In this time the motor's rotor must accelerate and then decelerate to a stop. This causes ringing, overshoot and vibration. There are some speeds at which the motor will not run. This is called its resonant frequency. The objective is to design the system so that no resonant frequencies appear in the operating speed range. This problem can be eliminated by means of using mechanical dampers or external electronics.

Stepper motor driving

In next month's instalment, we'll discuss a practical design for a high-power 3-channel stepper motor driver. To be able to understand the operation of this circuit, some more theory is required, together with an explanation of what the various terms mean. We'll take it step by step.

The stepper motor driver circuit has two major tasks:

1. To change the direction of magnetic flux in the phase by controlling the current direction and/or path.
2. To control and limit the current magnitude through the windings, with as short current rise and fall times as possible to achieve the highest performance drive.

Flux Direction Control

To continually step the motor requires a change of the flux direction, independently in each phase. This change is done by changing the current direction, and may be done in two different ways, using a bipolar or unipolar drive.

Bipolar Drives

This refers to the principle where the flux direction in one phase is reversed by shifting the voltage polarity across the winding causing the current direction to reverse. As illustrated in **Figure 5**, this requires four switches per phase.

Unipolar Drives

This term refers to the principle where the flux direction in one phase is reversed by shifting the current from one half of the winding to the other half. This requires two switches per phase, see **Figure 6**.

Current Control

In order to limit the power dissipation in the winding resistance and control the torque, the current must be controlled. There are two main principles to limit the current: current-limited drive and chopper-drive. Both can be realized either as a bipolar or unipolar driver.

Current limiting

This method (**Figure 7**) basically utilizes a constant voltage supply with a series ('dropper') resistor used to limit the current. The disadvantage of this method is the power dissipation in the series resistor.

Bi-level L/R Drive

This method, illustrated in **Figure 8**, utilizes a secondary supply, connected to the winding during the initial current build-up period. After a preset period, as the current has reached its nominal value, this secondary supply is disconnected. The disadvantage of this method is the need of a second power supply.

Chopper Control

The chopper driver provides the optimal solution both to current control and fast current build-up. The basic idea (**Figure 9**) is to use a supply voltage, which is several times higher than the nominal voltage of the motor. The current rise rate, which initially is *V/L*, is allowed to increase substantially. By controlling the duty cycle of the chopper, an average voltage and an average current equal to the nominal motor voltage and current are created. Constant current regulation is achieved by utilizing a current sense resistor in series with the motor winding to provide feedback to limit the nominal current.

(020127-1)

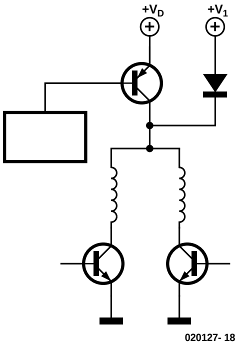


Figure 8. Bi-level current limiting.

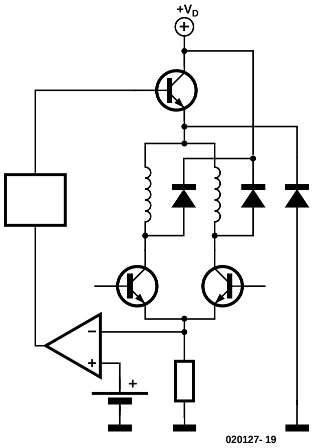


Figure 9. Principle of chopper control.

Next month

we'll discuss a high-power 3-channel stepper motor driver with PC control, complete with associated control software. The driver is PIC controlled and communicates with your PC through a serial link.

Driver Selection Chart

Drive Method	Component Cost / Count	Electrical Noise	Torque / Speed / Resolution	Power Efficiency	Design Notes	Application
1. Unipolar Constant Voltage	Low, 4 power transistors per motor.	Low	Holding torque depends on supply voltage and motor temperature. Large torque ripple when used in half-step mode.	Lowest output power. Low efficiency. Maximum power dissipation at standstill.	Driver transistors must be rated twice the supply. Power supply must be designed for given motor winding.	Low speed and low power. Normally used only with small motors.
2. Unipolar L/nR Constant Voltage	Low, 4 power transistors per motor.	Low	Holding torque depends on supply voltage and motor temperature. Large torque ripple when used in half-step mode.	Very low efficiency. Maximum power dissipation at standstill increased by L/nR ratio.	Problems with heat dissipation from the series resistors.	Low/medium speed and low power.
3. Unipolar Timed bi-level	Medium, 6 power transistors per motor. Requires CPU or timing circuit.	Medium	Holding torque depends on supply voltage and motor temperature.	Medium efficiency.	If bi-level voltage ratio is high, controlling of the holding torque & step accuracy is made difficult.	Low/medium speed and low/medium power.
4. Unipolar Constant Current	Medium / high, 6 power transistors per motor.	Medium / High	Optimum torque at high speeds. 70% optimum torque at low speeds. No torque ripple when used in half-step mode.	Medium / high efficiency	Driver transistors must be rated twice the supply. Winding leakage inductance has to be considered when designing snubber circuit.	High speed and medium power.
5. Bipolar Constant Current	High, 8 power transistors per motor. Integrated drivers available for low-medium power applications.	High	Optimum torque at both low and high speeds.	High efficiency. Maximum O/P power. Power loss in current sense resistors.	Problems with noise and interference may occur.	High speed and high power.
6. Bipolar Constant Current Micro-stepping	Very High, 8 power transistors per motor. Current control electronics more costly. Integrated drivers available for low/medium power motors.	High	Optimum torque at both low and high speeds. Resonant free operation at low speeds. Increased resolution.	High efficiency. Maximum output power. Power loss in current sense resistors.	Problems with noise and interference may occur.	High speed and high power. Used if a higher resolution or resonant free operation is required.

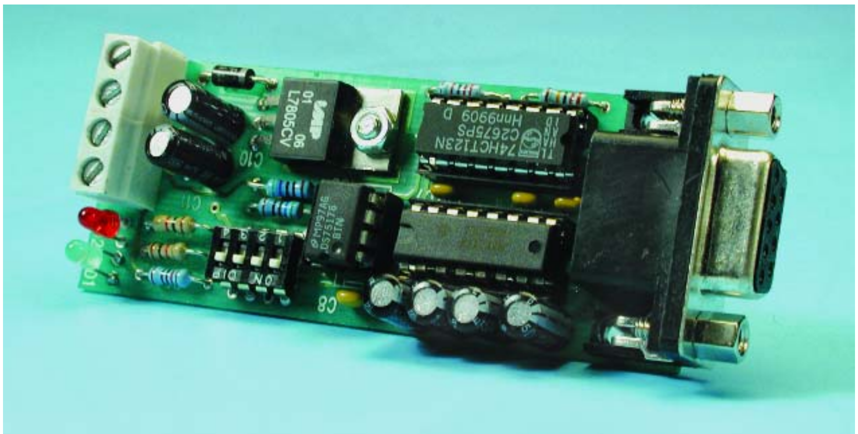
Precision Measurement Central (4)

Part 4: RS485, or networking the MSC1210 micro

Design by J. Wickenhäuser www.wickenhaeuser.com

The RS485 port on the MSC1210 board provides great opportunities for communication with other systems. For example, it allows a number of MSC1210 boards to be connected up into a network and talk to one another over distances of several kilometres.

This fourth instalment in the series on the MSC1210-based Precision Measurement Central (kicked off in the July/August 2003 issue) is entirely devoted to the RS485 interface. This month we will cover the theory and hardware. The December 2003 issue, finally, will discuss the practical use of the RS485 interface in a small project that goes to show how up to 255 MSC1210 boards can communicate with a single PC through an RS485 link. The converter required for the job is discussed in this article instalment.



RS485

Although the RS485 interface has been around virtually since the earliest days of the PC era, it is rarely seen in real life, that is, on the PC's connector panel. That's why we take this opportunity to present a suitable converter for RS232 to RS485, which has been tailored for use in combination with the MSC1210 board. First, however, we need to wade though some theory.

Bus topology

Normally an RS485 network consists of a long cable (bus) of which the ends are terminated with resistors. Up to 32 devices may be connected to the bus, or even 256 if a suitable driver chip is used. Ideally, all devices are connected to form a long thread. In practice, however, individual bus participants may also be connected via branches of several metres (**Figure 1**). In 'minimum' cases only the two wires designated 'A' and 'B' are required, although it is recommended to implement potential cancellation using a third wire and an optional resistor. Another popular variant has four wires and allows the devices to be powered over the bus.

The digital information conveyed over the RS485 bus comprises the voltage difference between line 'A' and 'B'. If the difference is positive, the bus is said to convey a logic 1. In the other case, a logic 0 is conveyed. The use of differential voltages makes the bus rather immune to noise, provided a number of conditions are satisfied. **Figure 2** shows a bus with two devices on it, where the signal is to be carried from IN to OUT over a long cable. An oscilloscope is connected to the bus at the indicated position. Any cable will have a more or less distinctive impedance — values of 60 to 200 Ω being typical. The signal on the cable will remain undistorted only if the cable is terminated with a resistor whose value is approximately equal to the cable impedance. Wrong termination values cause signal distortion that can lead to data corruption and increased susceptibility of the bus to noise.

Fortunately, correct cable termination is really critical only in those cases where signals with a high data rate (> 57,600 bits/s) are conveyed over cable lengths exceeding

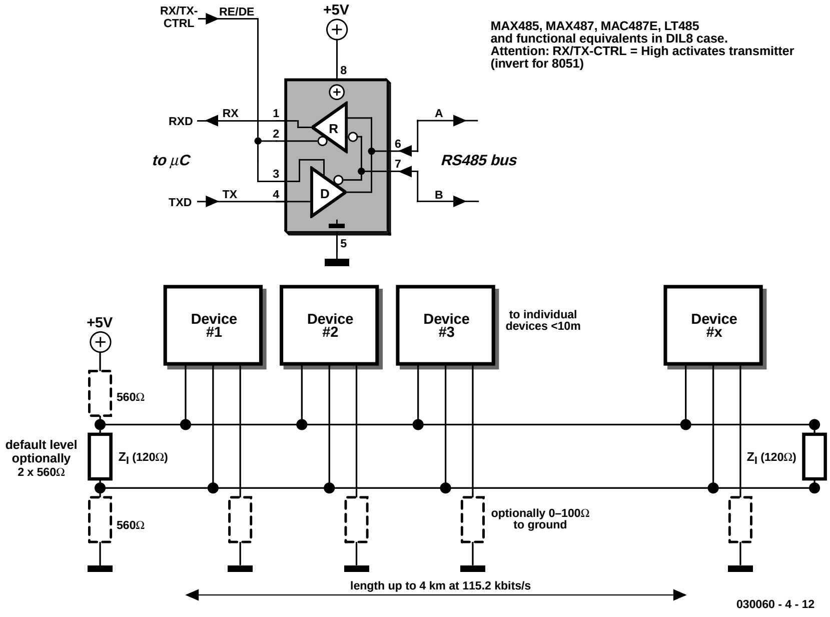

RX/TX-CTRL
RE/DE
+5V
8
MAX485, MAX487, MAC487E, LT485
and functional equivalents in DIL8 case.
Attention: RX/TX-CTRL = High activates transmitter
(invert for 8051)
RXD
RX
1
2
R
A
6
7
to µC
RS485 bus
3
4
D
TX
TXD
B
5
+5V
Device #1
Device #2
Device #3
to individual devices <10m
Device #x
560Ω
default level optionally 2 x 560Ω
Zl (120Ω)
Zl (120Ω)
560Ω
optionally 0–100Ω to ground
length up to 4 km at 115.2 kbits/s
030060 - 4 - 12

Figure 1. RS485 topology.

500 m, when the propagation times start to approach the bit rates (the signal travels 100-300 m in 1 µs). With cable lengths under 50 m and bit rates under 57,600, a cheap cable with no specified impedance but terminated with 120 Ω at both ends should be adequate.

Overvoltage Protection 'lite'

An RS485 driver must be capable of withstanding line voltages between –7 V and +12 V at its input(s). Fair enough, but longish cables in particular are prone to pick up (by induction) much higher peak voltages caused by electrostatic fields, discharges, EMI transients and so on. Consequently it makes sense to protect all RS485 against voltage surges. With suitable protection, short and not too powerful transients can not harm the drivers ICs (**Figure 3**).

Data traffic

The drawing in Figure 1 includes all optional resistors. These resistors serve to pull the bus to a fixed logic level when there is no traffic (that is, all devices are receiving and none is putting data on the bus). Unfortunately, this variant is rather wasteful in terms of energy usage and at the same time likely to cause problems with long cables.

A better alternative is shown in **Figure 4**. In this configuration, the initial level of the bus is not known. If a device wants to put a data block on the bus, it has to activate its transmitter section first. However, as a result of improper termination as illustrated in Figure 2, it may happen that activation of the transmitter alone is sufficient for receiving devices on the bus to recognise data, when in fact no valid data has been sent as yet (see block marked ???).

This leads to the absolute requirement for a pause to be inserted immediately after activation of each transmitter section. This 'dead' period needs to be much longer than prescribed for the transmission of a single byte (t_byte). The time needed to convey a byte over the bus is roughly ten times the reciprocal of the baud rate, or about 1 ms per byte when a rate of 9,600 bits/s is used.

The data packet proper always starts with a special marker byte (START). Note, however, that START may never occur within the actual data (more about this in the next instalment).

Because only one device can transmit at a time ('half duplex') a strict protocol is needed describing who's allowed to transmit what and when. If the protocol is not observed, data contention, collision and corruption is imminent. The practical application dis-

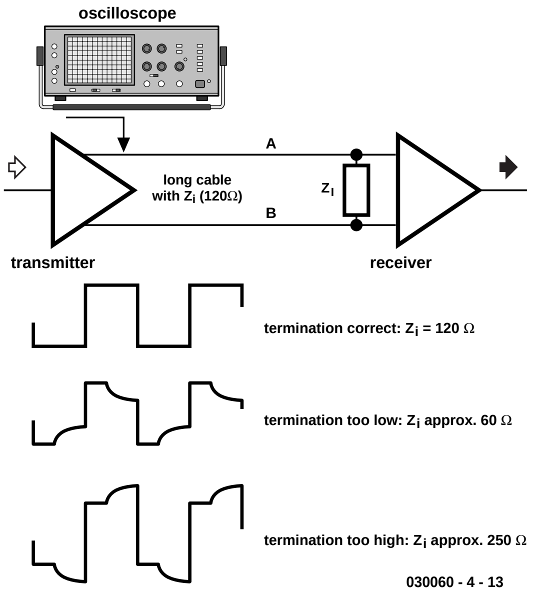

oscilloscope
A
long cable with Zl (120Ω)
Zl
B
transmitter
receiver
termination correct: Zl = 120 Ω
termination too low: Zl approx. 60 Ω
termination too high: Zl approx. 250 Ω
030060 - 4 - 13

Figure 2. Why terminate?

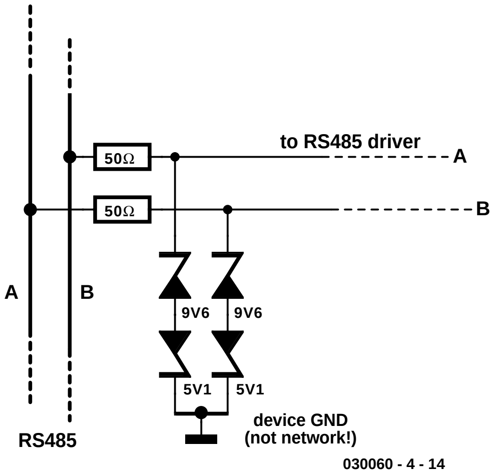


Figure 3. Surge protection by means of two zener diodes.

cussed in next month's instalment employs a simple but effective protocol.

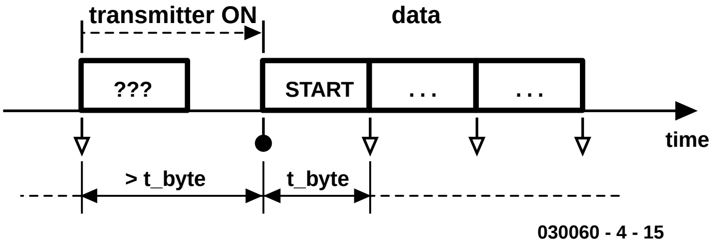


Figure 4. Transmit timing.

A PC on the RS485 bus

If the two 560-Ω resistors shown in Figure 1 are used, the risk of 'undesired databytes' is greatly reduced. Problems may only occur with long cables (>50 m) and higher baud rates (>57,600 bits/s), particularly when the transmitting junction and the resistors are relatively far apart. Unfortunately, a PC can not exercise exact control over the pulse timing on its serial interface and so activate the transmitting device using an appropriate delay. After all, most PCs employ a UART (RS232 driver) with an internal transmitter FIFO. This prevents the PC from 'knowing' whether or not a databyte was actually sent or not. Consequently it is the task of the PC converter to ensure that the RS485 bus is at a valid logic level when there is no traffic.

The converter

The converter, of which the circuit diagram is given in **Figure** 5, operates largely as a 'dumb' extension which has the benefit of it supporting different protocols and, of course, bit rates. The design consists of an RS232 and an RS485 converter separated by good old TTL levels.

Bits arriving at the RS485 bus are copied directly to the PC irrespective of the bit rate. Once the PC sends a character, the converter immediately switches on its transmitter via a retriggerable monostable (IC1.A). The transmitter remains active for the time its takes to convey the byte. In the case of our circuit, two times may be selected: 20 ms for 1200 bits/s (S1 opened) or 2 ms for 9600 bits/s and higher (S1 closed).

In addition, switch S1.4 may be used to include the terminator resistor, and/or S1.2/3 (operate at the same time) for the 560-Ω resistors to set the default level on the bus. These provisions on the converter allow several PCs or a combination of PCs and MSC1210 boards to be included in a network.

The construction of the converter on the printed circuit board shown in **Figure 6** should be an uncomplicated affair as there are no pitfalls with regard to soldering etc. The introductory photograph shows the author's prototype. The converter operates from an external supply voltage of

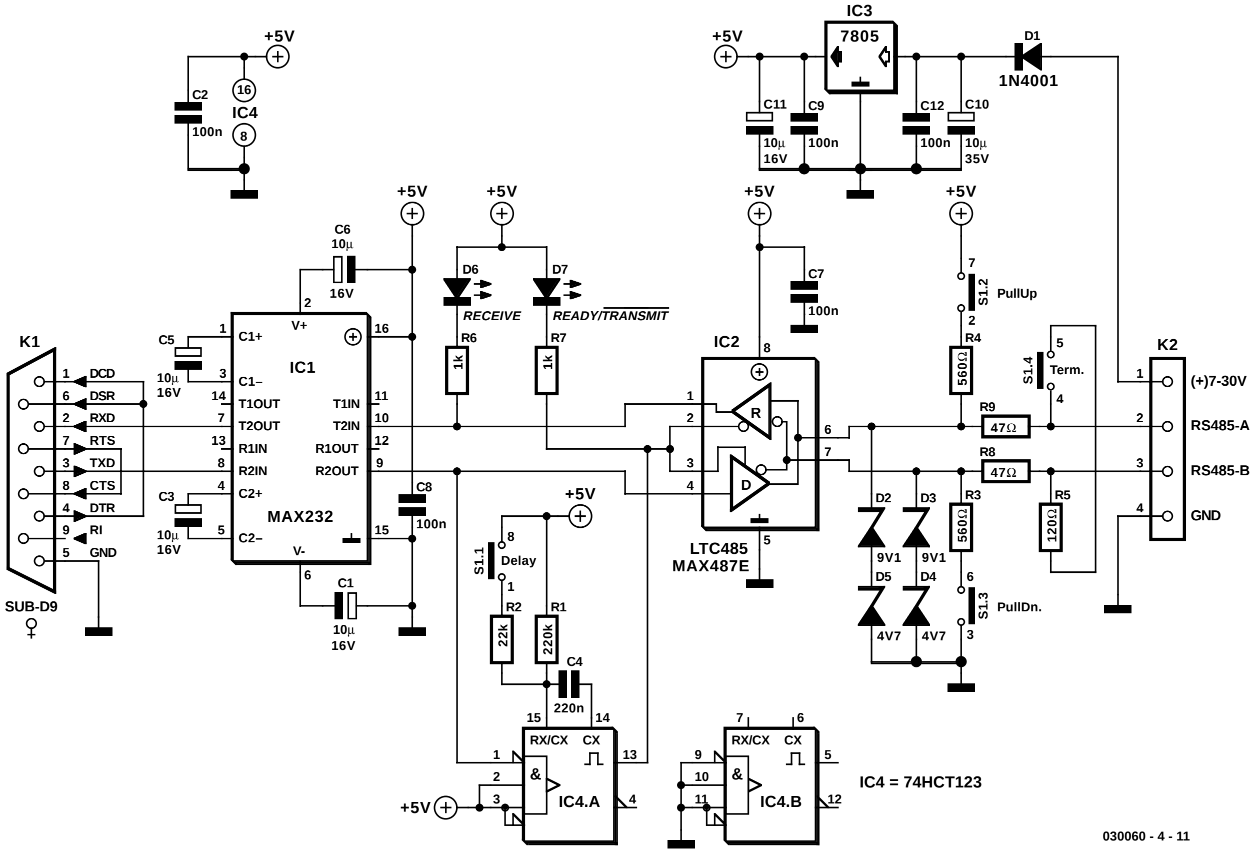


Figure 5. Circuit diagram of the RS232/RS485 converter.

COMPONENTS LIST

Resistors:
R1 = 220kΩ
R2 = 22kΩ
R3,R4 = 560Ω
R5 = 120Ω
R6,R7 = 1kΩ
R8,R9 = 47Ω

Capacitors:
C1,C3,C5,C6,C11 = 10µF 16V radial
C2,C7,C8,C9,C12 = 100nF
C4 = 220nF (10% tolerance)
C10 = 10µF 35V radial

Semiconductors:
D1 = 1N4001 (DO41 case)
D2,D3 = 9V1 zener diode
D4,D5 = 4V7 zener diode
D6 = LED, low current, green, 3mm
D7 = LED, low current, red, 3mm
IC1 = MAX232 (DIL case)
IC2 = LTC485 or MAX487E (ESD/EMI-optimised) (DIL case)
IC3 = 7805 (TO220 case)
IC4 = 74HCT123 (DIL case) (Philips preferred)

Miscellaneous:
K1 = 9-way sub-D socket, PCB mount, angled pins
K2 = 4-way PCB terminal block, 5mm lead pitch
S1 = 4-way DIL switch

between 7 V and 30 V and draws a modest 20 mA or so. In case the supply voltage is not obtained from the bus, the RS485 ground is connected to the circuit ground using a 100-Ω resistor for potential equalisation.

(030060-4)

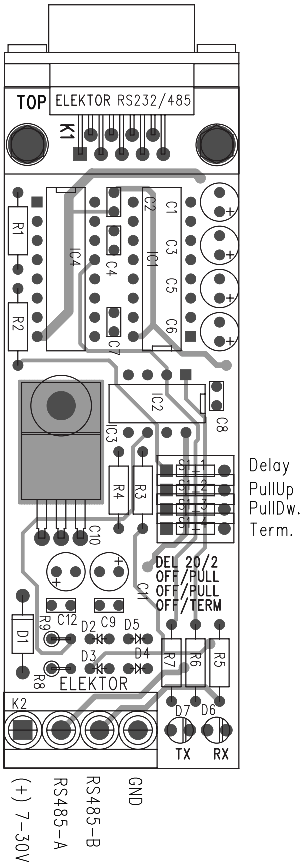


Figure 6. PCB design for the RS232/RS485 converter.

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 and email received, it is not possible to answer all correspondence, and we are unable to respond to individual wishes and requests for modifications to, or additional information about, *Elektor Electronics* projects.

Valve Preamplifier (1)

Dear Sir — with reference to the above design I would like to comment as follows: An excellent design by B. Stuurman, but as “an old timer” who learnt amateur electronics with valve amplifiers and +300V stabilisers in the 1960's, the one golden rule has been broken; 6-volt heaters must be applied at all times when HT (here +260 volts) is applied. The valves can be damaged by the strong electrostatic field between anode and cold cathode, resulting in what is I believe is called “cathode stripping”.

The Designer has given the User a choice of which half of the circuit to use; either the EF80 or the ECC82. With all due respect to the Designer, one suggestion would be to power both tubes from the 6.3 volt supply at all times, using the LM317 (or a higher rated variable linear power supply IC). Or, a Switch Mode design based on the Luxeon LED Driver, Figure 7, page 42, in either Voltage (+6.3 volts) or Current (300 mA) mode. (In 1978, to power the heaters of a valve-TV using parallel heater tubes that had been changed from the series heater types, a switch mode supply was used based on a 741 IC and 2N3055's).

Ian Newcombe (by email)

Many thanks for your critical notes on our publication, which have been copied to Mr. Bob Stuurman. When an answer is available we will revert to the subject on this page.

PICProg 2003

Dear Sir — with respect to the above project in the September issue, I would like to know where the software for IC6 is available from. Your article makes the statement that the project software comes in two parts: the executable for IC6 and the windows software to control the programmer using a PC. I have bought the windows software 010202-11 but can find no trace of the IC6 executable.

I know it is mentioned that a ready programmed IC6 can be purchased but I am taking this to be an option not a necessity. Could you please advise.

Berni C. Kendrick (by email)

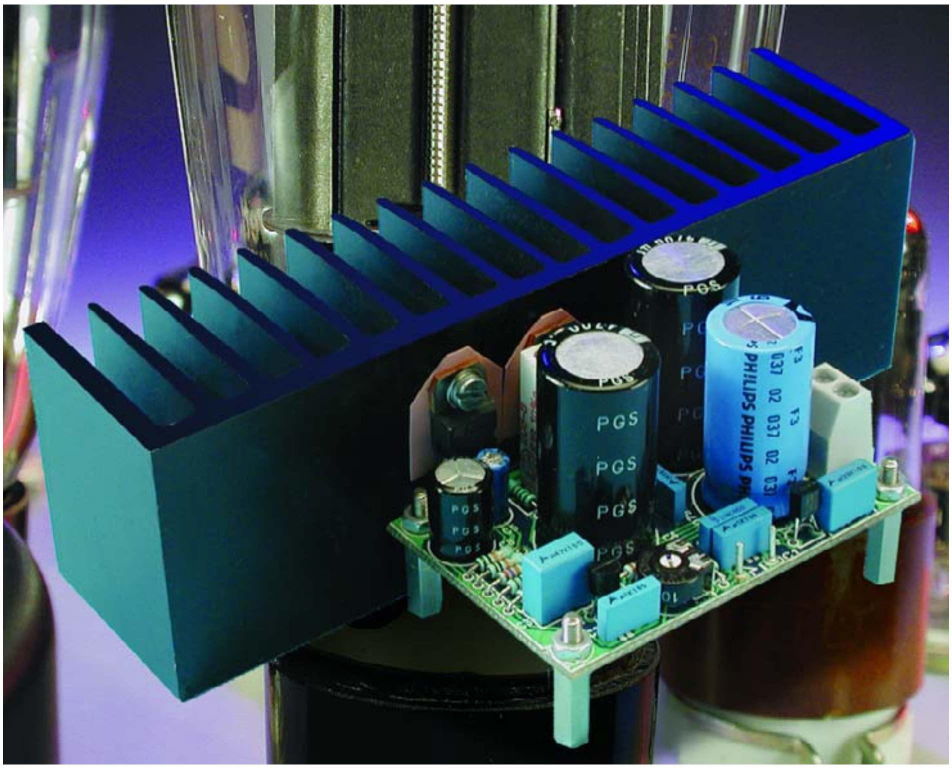
The executable for IC6 is not available as a free download or on diskette, and a statement to that effect is not made in the article. The relevant microcontroller can only be purchased ready-programmed from us, in compliance with a contract signed by the author/designer who receives royalties. The disk contents should equal that of the free download, and comprises the Windows software and runtime files only. For your information, disks are supplied at a nominal charge for the convenience of readers without internet access. The HEX file readers can download is a firmware upgrade only and not the full content of the program memory. Only upgrades are offered free of charge. By the way, ready-programmed PICs supplied through our Readers Services already contain the latest firmware version — the upgrade is only available to check if this option of the programmer software is operational.

FET Amp with Valve Sound

Warm sound from FETs?

Design by B. Kainka

A class-A FET amplifier can be made to sound as good as a valve amp when you follow the right design principles. This amp may not have a particularly high output power but using just standard components it produces a 'warm sound' reminiscent of a valve amplifier.



This purpose of this design is not to produce an amplifier to compete with modern high quality sound systems in terms of output power and low distortion figures but instead to be a bit more experimental and use modern FETs in a valve circuit configuration to try and recreate the warm 'valve sound' produced by valve amplifiers and much appreciated by many audiophiles.

To start with, it would seem a good idea to compare a typical single-ended EL84 valve output stage with a modern semiconductor push-pull output amp:

- The output stage of a valve amplifier operates in pure 'class A' configuration. Even with the volume control turned all the way down (quiescent) there is still appreciable dissipation in the output stage so the valves will always run hot. The dissipation in an EL84 could be 12 W so the valves will be running quite hot.

  By comparison, semiconductor amplifiers are typically designed with a 'class AB' output stage. Under quiescent conditions very little power is dissipated in the output transistors. At higher signal levels the two output transistors operate in push-pull, sharing the load but this also gives rise to a certain amount of crossover distortion.

- The valve operates as a current source and its output impedance is relatively high so that it offers little electrical damping to the movement of the speaker cone. The motion of the speaker cone is also affected by many

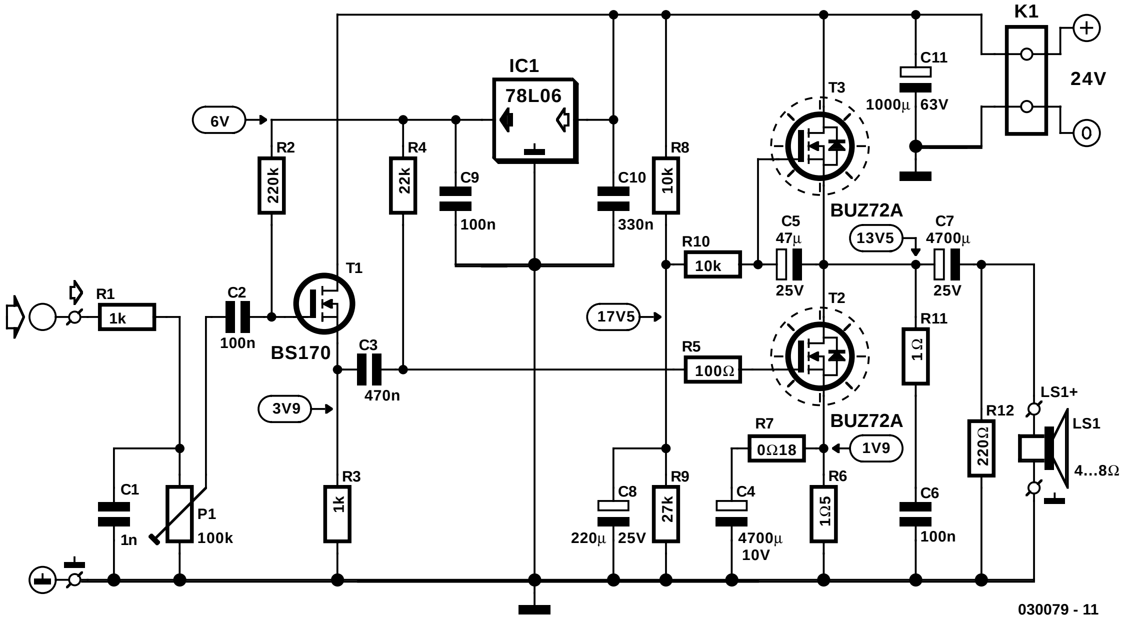


Figure 1. The FET 'valve' amp uses few components.

Technical Data

(supply voltage 24 V_{DC}, $I_{T2/T3} \approx$ 1.28 A)

Input sensitivity (P = 4 W/ 8 Ω)	0.47 V
Amplification factor (at 10 mV_{IN})	13
P_{max} (8 Ω, 10 % THD+N, 1 kHz)	4 W ('sine')
P_{max} (8 Ω, <20 % THD+N, 1 kHz)	5.5 W ('music')
THD+N (1 W, 8 Ω/1 kHz)	4.2%
THD+N (0.1 W, 8 Ω/1 kHz)	1.3%
S/N (at 1 W/8 Ω)	89 dB(A)
S/N (at 1 W/8 Ω, B = 22 kHz)	85 dB
Bandwidth (at 0.1 W)	83 Hz to 155 kHz
Damping factor	0.21 (!)
Output impedance	38 Ω (approx.)
Efficiency (4 W/8 Ω)	13%

other factors including the resonant frequency of the loudspeaker enclosure.

By contrast, a semiconductor amplifier has a low impedance output stage operating as a voltage source. This configuration leads to a much 'stiffer' control of the speaker with effective damping and gives a more controlled, flatter frequency response characteristic to the sound system. At the same this also means that the individual sound character of the speaker system is suppressed.

– The characteristics of a valve are not linear but slightly curved. At low volume the signal levels are small so that the valve operates over a near-linear region of its characteristics this will give little distortion of the output signal. When volume is increased the signal swing is larger and the non-linear nature of the valve becomes more apparent, introducing a soft limiting to the output signal. This type of limiting is quite pleasant and is achieved without using any overall negative feedback.

A typical AB class semiconductor amplifier uses a great deal of negative feedback to produce an amplifier with very low levels of signal distortion and low output impedance. When the volume is turned up in this type of amplifier the output signal increases proportionately until the signal peaks approach the levels of the amplifier supply voltage. Any further increase in signal input will cause 'overdrive' when the driver transistors switch off at the signal peaks. This condition occurs suddenly and the resultant signal clipping gives rise to a very unpleasant harsh sound. For this type of amplifier it is therefore important to make sure that it isn't overdriven, by contrast a valve amp is much more forgiving and responds to overdriving in a more 'listener friendly' manner. For this reason valves are often used in guitar amps where they are deliberately overdriven to produce different sound effects.

The Amplifier circuit

The amplifier operates in class A configuration. The design does not use any overall negative feedback to compensate for the non-linear characteristic of the output FET. The output impedance for this amp is quite high in order to reduce loudspeaker damping and allow the individual acoustic properties of the speaker system to be heard.

At the input, R1/C1 act as a low-pass filter to remove unwanted high frequency signals. The first stage of the circuit shown in **Figure 1** consists of BS107 (T1) this is used an input buffer giving a high input impedance and low output impedance with a voltage gain of about 1. The low output impedance helps to overcome the effects of the gate capacitance of T2. T3 operates as a constant current source; it passes dc current while the ac content of the output signal from the drain of T2 is applied across the loudspeaker.

The amplifier requires a 24 V input voltage. No set-up is necessary with this circuit because the operating point is automatically adjusted. A 6 V voltage regulator is used to derive the gate voltage for both T1 and T2. Current through the FET is defined by the value of source resistance. The quiescent current in T2 is self-regulated to 1.28 A but the relatively wide spread of individual FET characteristics can influence this figure.

The ouput driver

As part of the background to this project a earlier prototype amplifier was built and is shown in **Figure 2** This amplifier uses just one single VMOS FET per channel. The output transformer acts as a choke. The dc resistance of the secondary winding is less than 1 Ω so that only a small proportion of the drain current flows through the loudspeaker. A preset is used to adjust the amplifier operating point. No driver stage is used so the relatively high input capacitance of the power FET means that it is only possible to drive the amplifier successfully at high frequencies by using a low impedance signal e.g. from the headphone output of a CD payer.

The sound produced by this simple circuit is almost as good as the amplifier described here. The design however needed to be improved in three main areas:

– The FET Amp should be able to connect to normal impedance loudspeakers without the need for an output transformer. An electrolytic output capacitor would ensure ac-coupling to the loudspeaker.

– The amplifiers operating point should not require any set-up adjustment and must be self-stabilising. A stabilised gate voltage should be used together with some localised feedback from a resistor in the FET source lead.

– The input to the amplifier should be compatible with a standard high impedance line-in signal (1.0 V_{pp}). (The input transistor T1 is configured as a source-follower to act as a buffer).

With these requirements in mind the amplifier circuit shown in **Figure 1** was produced.

The diagram shows just one channel and the most obvious difference from the earlier design is the extra power FET (T3) in the output stage. This FET is used as a high-impedance current source and biases the drain of the output FET at the mid point of the supply voltage. Potential divider network R8/R9 provides the bias voltage to the gate of T3 with an additional 4 V necessary to take into account the gate-source voltage of the FET. Capacitor C8 reduces any ac ripple on this reference voltage. C5 maintains a constant gate voltage to T3 during operation. This gives the current source a high output impedance which together with the high impedance output of the signal amplifying FET gives the complete amplifier its high impedance output characteristic. During tests the dynamic impedance of the amplifier was measured as 38 Ω.

The output signal to a low impedance loudspeaker (4 to 8 Ω) is not in the form of a signal voltage but rather a signal current, this is unusual for a semiconductor amplifier but gives rise to the characteristic 'valve sound' produced by the amplifier.

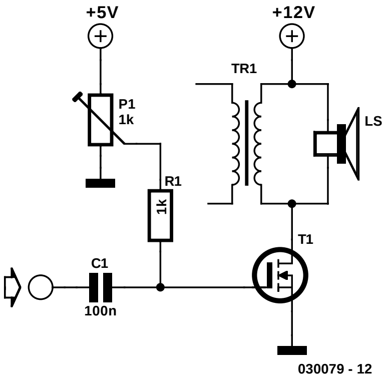


Figure 2. An earlier prototype amplifier.

The use of an FET as a current source in the output stage instead of the more usual impedance matching transformer means that we do not need to compensate for the frequency response of the transformer. A suitable transformer or output choke is a specialist item and not that easy to find these days either. A disadvantage of this approach however is the increased power consumed by the circuit.

The signal amplifier FET (T2) dissipates around 12 V × 1.28 A ≈ 15 W in its quiescent state and the constant current source uses about the same amount so altogether this gives more than 30 W of energy converted into heat. With this in mind its important to ensure that this energy can be safely dissipated so we recommend that the heat sink should have a minimum thermal resistance of 1 K/W or lower.

The FET amp can also be run with a higher supply voltage up to around 35 V giving a larger drain current. In this case the source resistor will need to be correspondingly reduced in value. It would be worth considering this modification if you like the sound produced by this amp and would like to experiment a little.

The voltage produced across the source lead resistor generates a localised negative feedback and reduces the slope and amplification factor of the amplifier without reducing its output impedance. C4 shunts this feedback signal to ground and controls the amplifiers frequency response. The amplification factor at high frequencies is set by resistor R7 (0.18 Ω). To extend the amplifier response down to 30 Hz the value of capacitor C4 would theoretically need to be increased to 30,000 μF! This modification

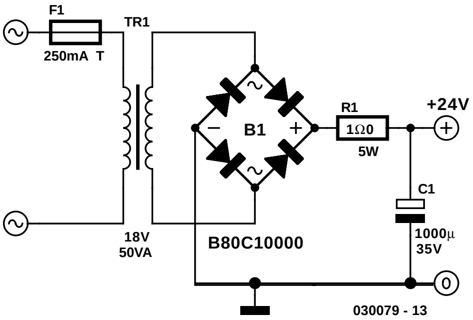


Figure 3. The basic power supply with a series resistor.

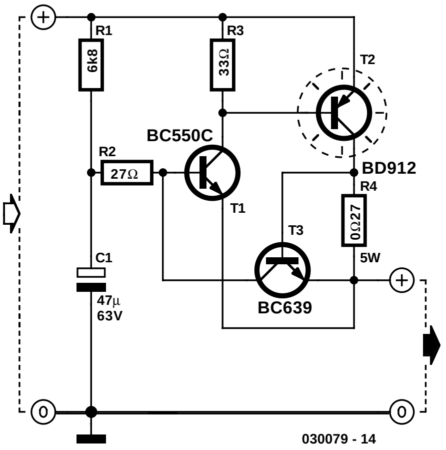


Figure 4. An add-on circuit to reduce hum.

would not only be expensive but also potentially damaging because it is much easier to overload the amplifier at low (bass) frequencies.

The power supply

The power supply shown in Figure 3 can be constructed quite easily. A small amount of mains ripple on the output voltage will be damped by the constant current output of the amplifier. A resistor in series with the reservoir capacitor reduces the peak charging current and helps to attenuate the higher frequency content of the ripple voltage.

The amplifier has a relatively poor hum rejection figure of 20 dB. An additional add-on circuit is shown which reduces the supply ripple so that the output hum is no longer audible.

The add-on circuit is shown in **Figure 4** and consists of a voltage regulator circuit built from discrete components. The reference voltage is derived from an averaged value of the supply voltage. The voltage drop over the regulator is independent of the value of the rectified transformer voltage. T1 and T2 are configured to form a Darlington transistor with T3/R4 providing short-circuit current sensing in the collector lead. With T2 mounted on a sufficiently big heat sink the regulator is short-circuit proof.

The regulator voltage drop is about 1.9 V so in normal operation the power dissipation should be less than 2.5 W. The current limit is set to around 1.9 A, a short circuit at the output will produce approximately 45 W power dissipation by the circuit.

For testing we used a 50 VA toroidal mains transformer with two 18 V secondary outputs together with a 10,000 μF (35 V) reservoir capacitor. With no additional filtering 800 mV$_{pp}$ ripple voltage was measured on the output. In order to reduce this ripple a higher voltage is necessary so R1 (6.8 kΩ) produces an additional 1 V voltage drop across the regulator.

It is also necessary to take into account the thresholds of transistors T1 and T2. In practice it is possible that the H$_{FE}$ of T1 could cause the voltage to be too high or too low so that R1 will need to be changed to compensate (this is the only disadvantage of this simple solution).

Transistor T3 limits the output current by controlling the base voltage of T1 so that the output current reaches a maximum value defined by the voltage drop across R4. A BD912 is used for T2, this transistor can handle 15 A so a higher maximum output current can be safely selected

COMPONENTS LIST

Resistors:
R1,R3 = 1kΩ
R2 = 220kΩ
R4 = 22kΩ
R5 = 100Ω
R6 = 1Ω5 5W
R7 = 0Ω18 5W
R8,R10 = 10kΩ
R9 = 27kΩ
R11 = 1Ω
R12 = 220Ω
P1 = 100kΩ preset

Capacitors:
C1 = 1nF
C2,C6,C9 = 100nF
C3 = 470nF
C4 = 4700μF 10V (radial, max. diam. 18.5mm, lead pitch 5mm or 7.5mm)
C5 = 47μF 25V radial
C7 = 4700μF 25V (radial, max. diam. 18.5mm, lead pitch 5mm or 7.5mm)
C8 = 220μF 25V (radial, lead pitch 2.5mm or 5mm)
C10 = 330nF
C11 = 1000μF 63V (radial, max. diam. 18.5mm, lead pitch 5mm or 7.5mm)

Semiconductors:
T1 = BS170
T2,T3 = BUZ72A
IC1 = 78L06

Miscellaneous:
K1, LS1 = 2-way PCB terminal block, lead pitch 5mm
Heatsink, R$_{th}$ < 1 K/W

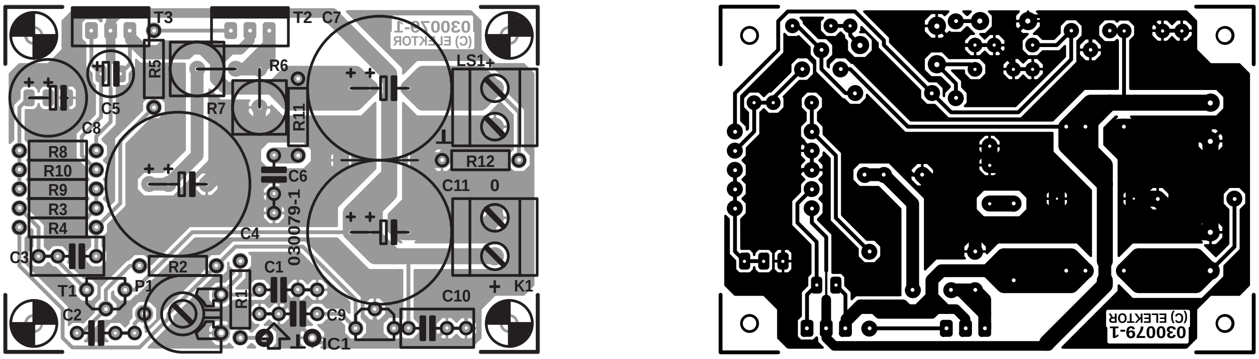


Figure 5. The power amp PCB.

if necessary by reducing the value of R4. The maximum current for T3 (a BC639) is 1 A (1.5 A peak). Under no-load conditions (with the voltage at approximately 27 V) or sudden output short-circuit, resistor R2 limits the current through T3 to 1 A as mentioned earlier. Capacitor C1 is discharged and the power dissipation in R2 remains small.

Construction

The hum suppressor is not included on the layout of the amplifier PCB (**Figure 5**), there is only room on this board for the amplifier. The large value electrolytics are mounted vertically on the board, so ensure that their case size and lead spacing will fit. This also applies to the power resistors.

Before any components are fitted solder the wire link between C7 and C11. Next fit all the low-profile components like the standard resistors, small capacitors, preset resistor, solder pins and connection blocks. Next fit the large upright components like the electrolytic capacitors and power resistors. Finally the two power FETs can be soldered into place but not before the heatsink has been fitted to the PCB with a suitable bracket. This ensures that no mechanical strain will be put on the FET leads and joints. Make sure that the FETs are fitted to the heat sink with electrical insulators and heat conducting paste.

Once the components have been fitted and all the soldered joints carefully inspected the amplifier can be tested. The voltage levels shown on the circuit diagram should help during testing but power FETs are produced with fairly wide tolerances on their characteristics so the voltage levels shown should be considered as approximate values only.

In use

As can be seen from the technical data the output power of this class-A amplifier is not particularly high. Theoretically with this design it should be possible to produce an output power about 25% of the quiescent power dissipated in the amplifier, but in practice it's a little bit less. This amp is therefore probably not the best choice for the sound system if you are planning a big party but it does produce a sweet sound with the volume control set around its mid-range. The soft limiting characteristic ensures that even with the volume cranked up to maximum the sound is not at all unpleasant. At low volume settings it is distortion-free with a good transparent tone. Altogether the sound produced has a full and soft character, truly reminiscent of a valve amplifier.

The sound quality of any amplifier is of course subjective and depends on many factors, not the least being the quality of the speakers connected to the output. In many cases we found that during testing, the same speakers produced a more pleasant sound from this amp than when they were connected to a conventional hi-fi amplifier.

(030079-1)

Data Buses

In electronic systems

By H. Nordmann

In this article we take a look at the pros and cons of different types of data bus structure in electrical design and compare some of the more popular varieties of serial bus available today.

Mention the word bus and chances are that most people would conjure up a picture of the big red double-decker variety used for moving people from one point to another. In this article we concern ourselves with the electronic variety: the data bus. There is however a link between the two, in both cases 'bus' is just the name given to the vehicle or carrier used for transporting things, in one case it is people and in the other it is bits of information.

Choosing the right type of data bus for a particular application is often a crucial design decision and it's important to study the specifications of bus structures available today to determine which is most suitable. If the finished equipment interfaces to third-party products make sure that the chosen bus is well supported and allows maximum flexibility.

The main questions to ask when deciding on the type of bus system to implement in a design are:

- How fast does the data need to be transferred?
- How many devices need to connect to the bus (now and in the future)?
- Is the cost of the bus design important? (Will the finished design be used in high-volume, low-cost equipment?).

Data buses can be broadly split into serial or parallel configuration using either synchronous (the data is sent along with some sort of clock signal to transfer the data) or asynchronous (where data is transferred without a clock).

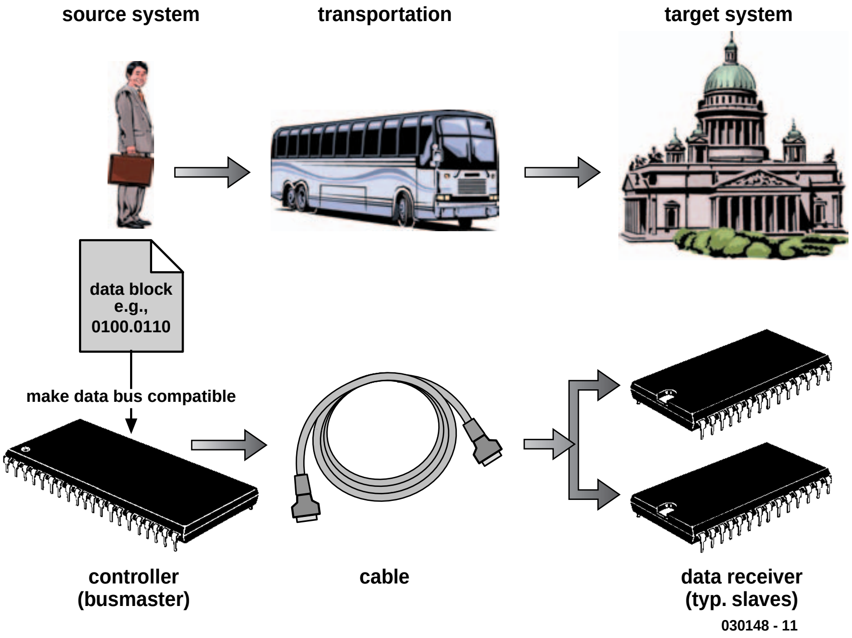


Figure 1. Every bus is a carrier.

The parallel Bus

A parallel data bus will typically have a minimum of eight bi-directional parallel data paths to send information between devices on the bus. Microcontrollers and DSPs with 32 bit data buses are currently being produced for applications requiring high-speed data manipulation. In short, parallel buses are chosen when you cannot compromise on data throughput: they will typically be used within equipment, over short distances to connect processor busses with their memory chips and other fast memory hardware.

The serial Bus

The serial bus structure uses just one single data path to send or receive information. A typical serial bus would use a minimum of three wires to transfer data. A byte of information is split into its individual bits, passed over the serial data line one after another and then reassembled into a byte at the receiver.

Even from this simple outline you have probably already worked out the strengths and weaknesses of

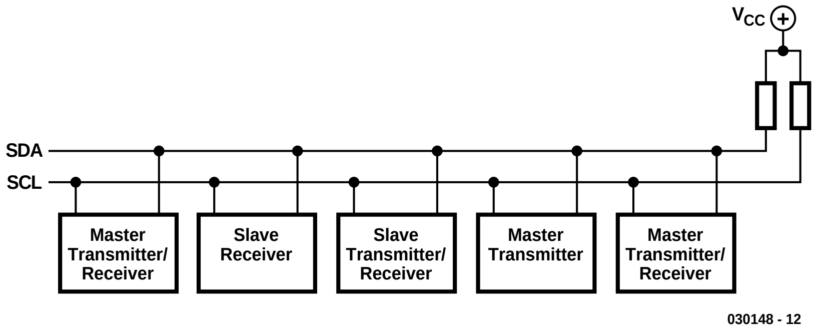

VCC
SDA
SCL
Master Transmitter/ Receiver
Slave Receiver
Slave Transmitter/ Receiver
Master Transmitter
Master Transmitter/ Receiver
030148 - 12

Figure 2. I²C bus topology.

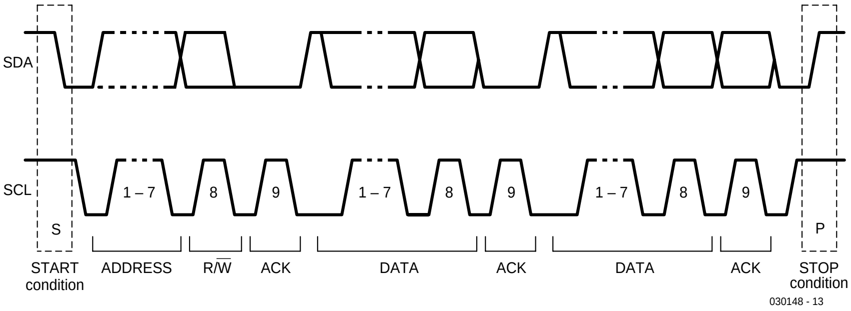


Figure 3. Data transfer on the I²C-Bus.

these two main contenders for bus layouts. The serial bus squeezes all the information down a single wire so data transfer will be relatively slow compared to the parallel bus that offers many more simultaneous data paths. The parallel bus requires more hardware in the form of connectors and cabling and this can give rise to problems for densely packed PCB layouts.

In addition to data signals, the parallel bus also requires some form of latch signal or chip select (sometimes called STROBE or ENABLE) to synchronise the data transfer. Asynchronous parallel buses are another variant and use a handshake system to transfer data: the transmitting device places data on the bus and then issues a strobe to the receiving device indicating that data is available to be read. The receiver reads the data and sends an acknowledge signal back to the sender to indicate that the data has been read. This type of data transfer is termed 'interlocked communication' and ensures that devices on the bus with different response times can communicate without problems.

Microwire Bus

Despite their inherently lower performance compared to parallel buses, synchronous serial busses are the most prolific, the majority of external interfaces to PCs are serial and there are a host of synchronous bus standards to choose from to suit all situations.

The Microwire bus is one example of this type of bus; it was developed by National Semiconductor and is implemented on the COP8 family of processors. This bus is used predominantly as an interface to serial EEPROMs such as the 93C46 (a 64×16 bit EEPROM). Microwire belongs to the '4-wire' family of busses because each slave connects to the bus via SO (serial data out from the master), SI (serial data into the master) and SK (the signal clock), together with a chip select giving four signal wires to each slave device. A disadvantage of this bus implementation is that each slave device on the bus requires its own chip select signal so that with ten slave devices on the bus, the master device needs to generate ten chip select signals. The Microwire bus has not been too widely adopted for this reason and is almost exclusively used in a 'single-master single-slave' system. Typical applications for this bus are memory modules (EEPROMs) in phones to store telephone numbers or tuner memory in TV sets or satellite receivers to store channel frequency data and also in other consumer equipment where information is stored and does not need to be changed very often.

The I²C Bus

The I²C bus was designed by Philips around twenty years ago and has since been widely adopted throughout the electronics world. The I²C (Inter Integrated Circuit) bus consists of just two (bi-directional) wires SDA (Serial Data) and SCL (Serial Clock). Originally the bus specification defined a seven bit device address, allowing up to 128 devices to be addressed ($2^7 = 128$) and the maximum data rate was 100 kbits/s but since a revision in 1992 the device address has gone up to 10 bits and the maximum data rate has been increased to 400 kbits/s (fast-mode), backward compatibility to the earlier specification is retained (more recently the data rate has undergone a further speed increase). **Figure 2** shows the general structure of an I²C bus layout with all devices connecting to both SDA and SCL.

Unlike the Microwire bus the I²C bus does not use a separate wire to address slave devices but instead defines device addresses in the communication protocol so that each device has a unique address. Devices connected to the bus can be either master or slave and I²C is a 'multi-master' bus that allows bus control by more than one device.

The I²C message format is shown in **Figure 3,** each message begins with a start sequence and is terminated by a stop sequence. The start condition is indicated uniquely by a falling edge on SDA while SCL is high and the stop condition is indicated by a rising edge on SDA while SCL is high, at all other times SDA only changes when SCL is low. A pull-up resistor on SDA ensures that the bus goes to a logic '1' in between messages. The clock pulses on SCL are generated by the master device sending the message and are inhibited between messages.

The message begins with a 7-bit slave address followed by a single R/$\overline{W}$ (read/$\overline{write}$) bit. If this bit is a '0' the master will be transmitting data while a '1' indicates that the master will be expecting to receive data from the slave. The PCF8574 port expander chip has an I²C interface and can be addressed to either read or write information so that if 'read' is selected it will send back the logical state of its input port while 'write' will output data from the master to its output port.

An acknowledge (ACK) signal is used to ensure that the addressed slave device is working correctly: the master device continues to produce the clock signal during this one bit period but releases the data line so that it rises to a logic '1' it is now the slave devices job to pull the data line low just for this one bit period to produce the acknowledge signal. An ACK signal is also required after each 8 bits of data.

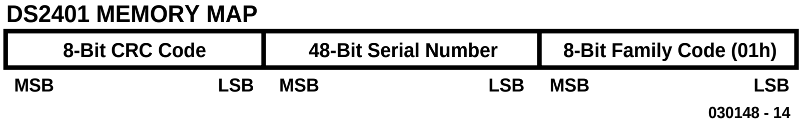
The specification has been revised to expand the numbers of device addresses, increase data rate and also to operate at lower voltage levels so that this bus remains a popular choice for communications between different micro modules in the consumer and entertainment fields. This standard is well supported and there is a wide selection of bus devices for connecting sensors and actuators.

1 Wire Bus

Lastly we will take a look at the '1-wire bus' (also known as MicroLAN) from Dallas Semiconductor. This bus design uses just a single data wire to transfer data and also power to the one wire chip. This method of power feed is known as a **parasitic power** connection. To complete the circuit to the bus it is of course necessary to provide an earth return path so two connections are actually required. This bus needs very little interface hardware and uses a sophisticated software protocol to ensure data security.

The sending device sends data by pulling the data line low. If the line is low for less than 15 μs the receiving device interprets this pulse as a logic '1', pulses that are at least 60 μs are interpreted as a logic '0'.

Figure 5. A ring iButton



DS2401 MEMORY MAP

8-Bit CRC Code		48-Bit Serial Number		8-Bit Family Code (01h)	
MSB	LSB	MSB	LSB	MSB	LSB

Figure 4. Format of the 64 Bit memory.

The tolerances on these timing pulses are deliberately wide to allow for worst-case timing inaccuracies in the slaves. The 1-wire bus is classed as a PWM (Pulse Width Modulated) coded bus system.

The advantages of the 1-wire bus are clear: very little hardware interfacing is required and the development and production costs are low.

As an example of a 1-wire device, the DS2401 is a 'silicon serial number' or ID chip and is packaged as an SMD or as a three pin TO92 outline (like a small-signal transistor). This device just contains a unique serial number programmed during manufacture. It can be used as an identification device for a PCB or piece of equipment to enable the bus master to identify devices connected to the bus, its serial number can be read out but it is not possible to write to the chip. The format of the stored serial number is shown in **Figure 4**. The least significant byte is an 8-bit family code identifying the type of device. The next 48 bits is a unique (factory programmed) serial number while the most significant byte contains a CRC (Cyclic Redundancy check), used to detect any data corruption of the preceding seven bytes. The large 48-bit serial number ($2^{48} = 2.8 \times 10^{14}$) ensures that each device produced can have a unique address.

One-way communication is not the only thing that the 1-wire bus has to offer, as well as the ID chips there is also a wide and growing range of different 1-wire devices ranging from simple EEPROMs to easily programmable real-time clocks with event counters and multi-channel Analogue to Digital converters. The unique 48-bit serial number and 8-bit family code associated with every individual device produced ensures complete component traceability.

The designers of the bus protocol have built-in some interesting and original features. The bus master can issue a Search ROM command to allow it to identify the 64 bit codes of all the attached devices on the bus or a Skip ROM which saves time in a single drop bus system by allowing the bus master access to the control functions without the need to provide the 64 bit ROM code. These features and many others make the 1-wire bus a very powerful solution for many applications.

As electronic systems get larger and more complex so the numbers of sensors and actuators connected to the bus increases which leads to greater bus loading and a greater volume of data to be processed. Bus couplers (the DS2409 together with the DS2406) can be used in this case to split large bus systems into many smaller branches to reduce the bus loading.

A good application example for the 1-wire bus would be in the area of personnel entry or 'clocking in' systems for industry. Each employee would carry their uniquely coded chip that would communicate with a central processor and allow them access to permitted areas.

A 1-wire device called an iButton has been developed for just these types of applications. The device is a 16 mm diameter stainless steel capsule containing a 1-wire device that can be mounted in a key fob (**Figure 5**). The capsule casing offers good electrical shielding from sources of interference and is robust enough to withstand pressure equivalent to being stepped on… just try that with a chipcard! (On second thoughts maybe that's not such a good idea).

(030148-1)

Web Addresses:

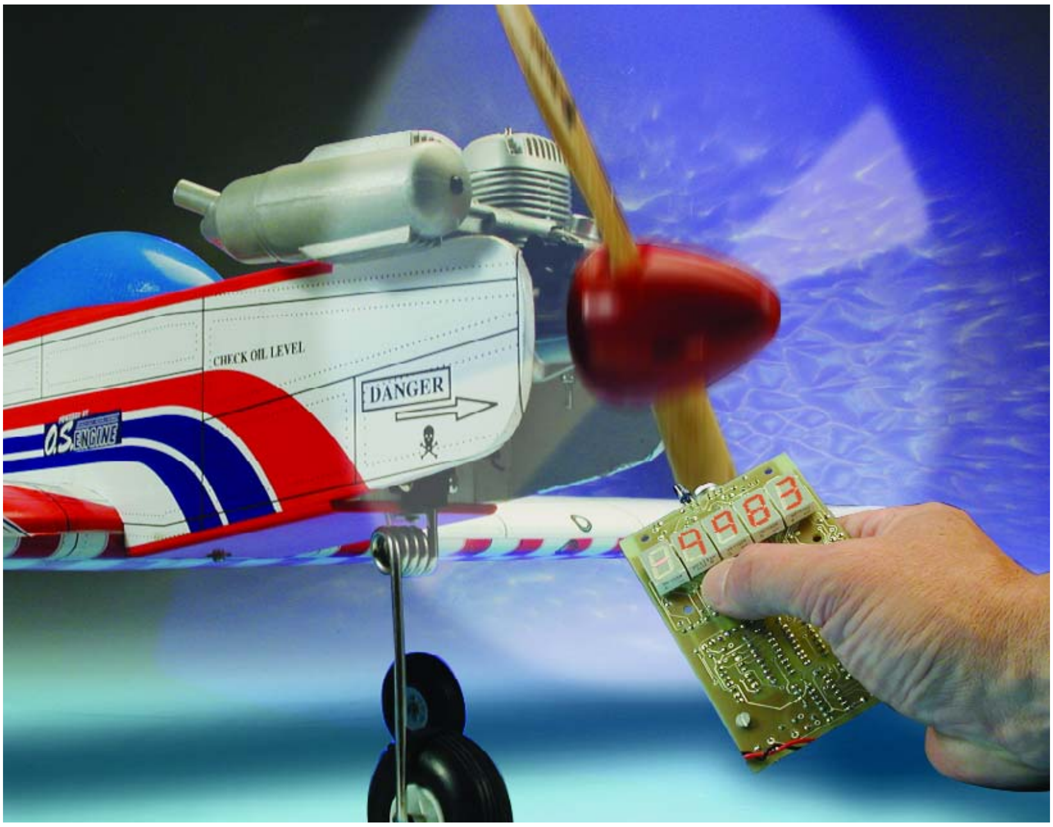
www.dalsemi.com
www.maxim-ic.com
www.ibutton.com

Rev Counter for R/C Models

Using an optical sensor

Design by P. Goossens

The engines used in modelling have to be carefully tuned if they are to run at their best. An important measurement is the engine speed at full power. This circuit can be used for such measurements, as long as a propeller is driven or a marker line is visible.



If you want to obtain the best performance from a model engine, it is essential that the engine speed can be measured accurately. With the help of a good rev counter the carburettor can be adjusted correctly at full power and engines can be compared with each other. This can become quite complicated, since the type of propeller, fuel, glow plug, etc, all affect the maximum engine speed.

The use of the rev counter described here isn't limited to measuring the engine speed of model airplanes. In principle, you can draw a white line on any drive shaft and then measure the rotational speed using this circuit. If the drive shaft is shiny, it is better to draw a matt-black line. The practicalities of taking measurements will be covered in detail later.

Sensor

In normal petrol engines the engine speed is easily arrived at by measuring the frequency of the pulses to the spark plugs. Unfortunately, the engines in the modelling world use glow plugs instead of spark plugs, so this method cannot be used. We there-

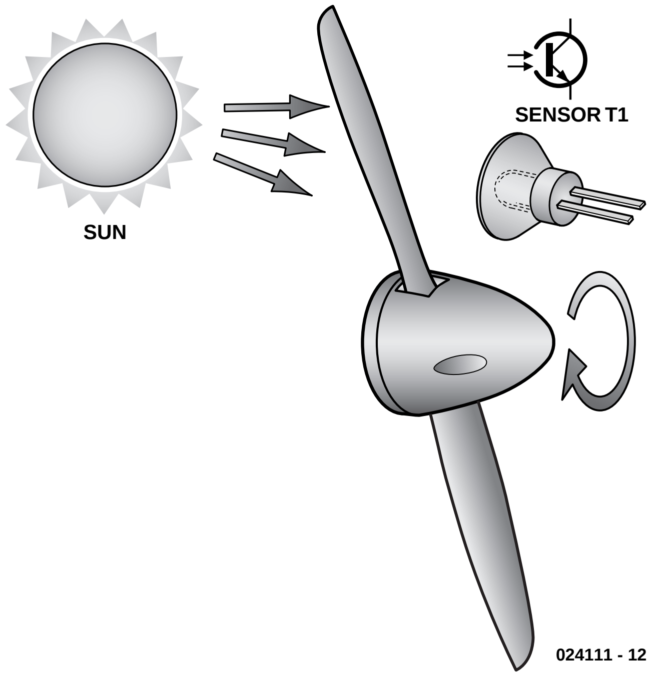


Figure 1. The light/dark transitions of the sunlight are picked up by the sensor.

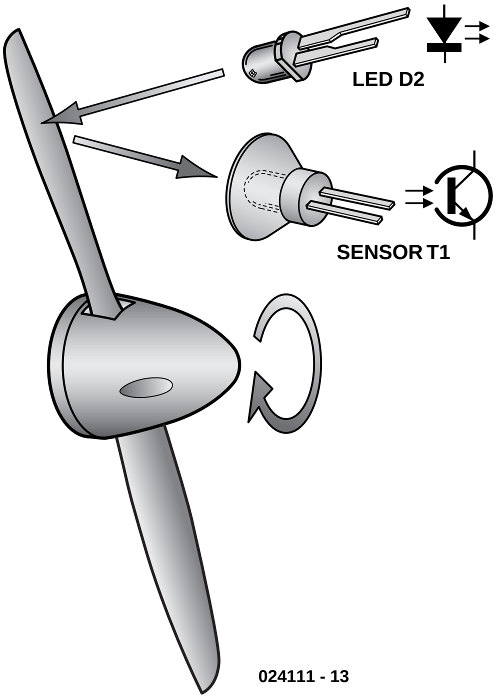


Figure 2. The light from an LED is reflected from the propeller blades and then detected.

fore have to look for a different solution. It would be ideal if no special parts had to be mounted inside the airplane, since the last thing you want is to increase the weight of a model airplane.

Most model airplanes still use a propeller that is driven directly by the crankshaft of the motor. The rotational speed of the propeller is then equal to that of the engine. If we now hold an optical sensor in front of the propeller such that the shadow of the blades falls across the sensor, we can measure the frequency at which the blades rotate (see **Figure 1**).

This method does require sufficient background light, because the difference between light and shadow has to be big enough for it to be detected by the sensor. When there is too little background light, it's very easy to give Nature a helping hand. Just to the side of the sensor of the rev counter we add an IR LED, which doesn't illuminate the sensor directly, but which emits infra red light in parallel towards the propeller. The blades of the propeller will reflect this light back periodically to the sensor, which can be measured (**Figure 2**).

Measurement

The frequency of the signal from the sensor is directly proportional to the engine speed. With a two-bladed propeller the light intensity will change twice per revolution, with a three-bladed one three times, and so on. From the frequency of these interruptions it is possible to calculate the number of revolutions per second. However, a rev counter normally shows the number of revolutions per *minute* and not per second. That means that we still have to multiply the shaft frequency by 60 to obtain a value for RPM (revolutions per minute).

Fixed measuring time

Right then, to start with we have to measure a frequency. There are two standard methods used for this. The first is to count the number of pulses that occur during a fixed period of time. The measured frequency is then obtained using the formula:

$$f = \frac{\textit{measured number of pulses}}{\textit{time}},$$

where the time (in seconds) is constant. If we take a fixed period of 1 second, the accuracy of the measurement will be ±1 Hz. In our case, this frequency has to be multiplied by 60 and divided by the number of propeller blades.

For a two-bladed propeller the accuracy would become:
±1 Hz x 60 / 2 = ±30 RPM
If possible, we would like to have a resolution of one RPM. (Note that this is not usually provided, since most rev counters show a value x 100, so the resolution is then 100 RPM). Simply increasing the measurement time by 30 will give the required resolution. Unfortunately, this also means that the measurement is only updated twice per minute. This is unacceptably slow, so this method cannot be used.

Fixed number of pulses

Another method is to measure the time taken for a certain number of pulses to occur. This method is therefore the exact opposite of the previous one. The frequency is now calculated as follows:

$$f = \frac{\textit{pulses}}{\textit{measured time}},$$

This measurement is most accurate at lower frequencies and becomes progressively less accurate at higher frequencies. The resolution

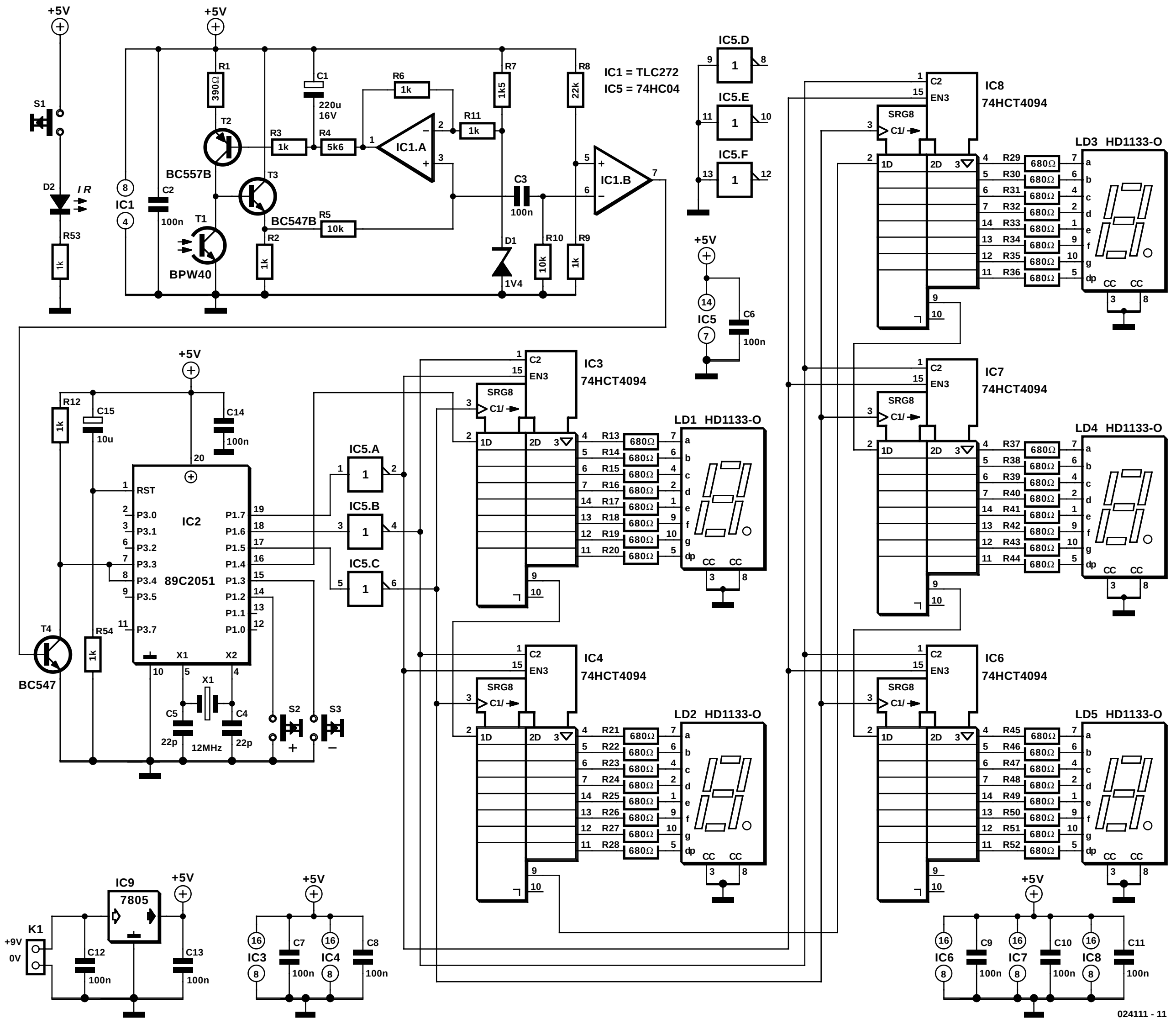


Figure 3. The circuit consists of three sections: the sensor, the controller and the display.

here depends on the number of pulses counted (constant), the accuracy of the time measurement, the frequency to be measured and the number of blades on the propeller. To give an acceptable accuracy over the whole range, a large number of pulses are required for this method, which results in a long measuring time at low revs; this isn't ideal either.

A different way

We can conclude that neither of the previously mentioned measurement techniques is suitable for our purpose, so some other method has to be found. In the end we used a technique that is really a combination of the two, but without their disadvantages.

The measurement starts when the first pulse is detected by the sensor. During the next second the number of pulses is counted. After this second has passed, we wait until final pulse is detected. The frequency can then be calculated using the formula:

$$f = \frac{\textit{measured number of pulses}}{\textit{measured time}},$$

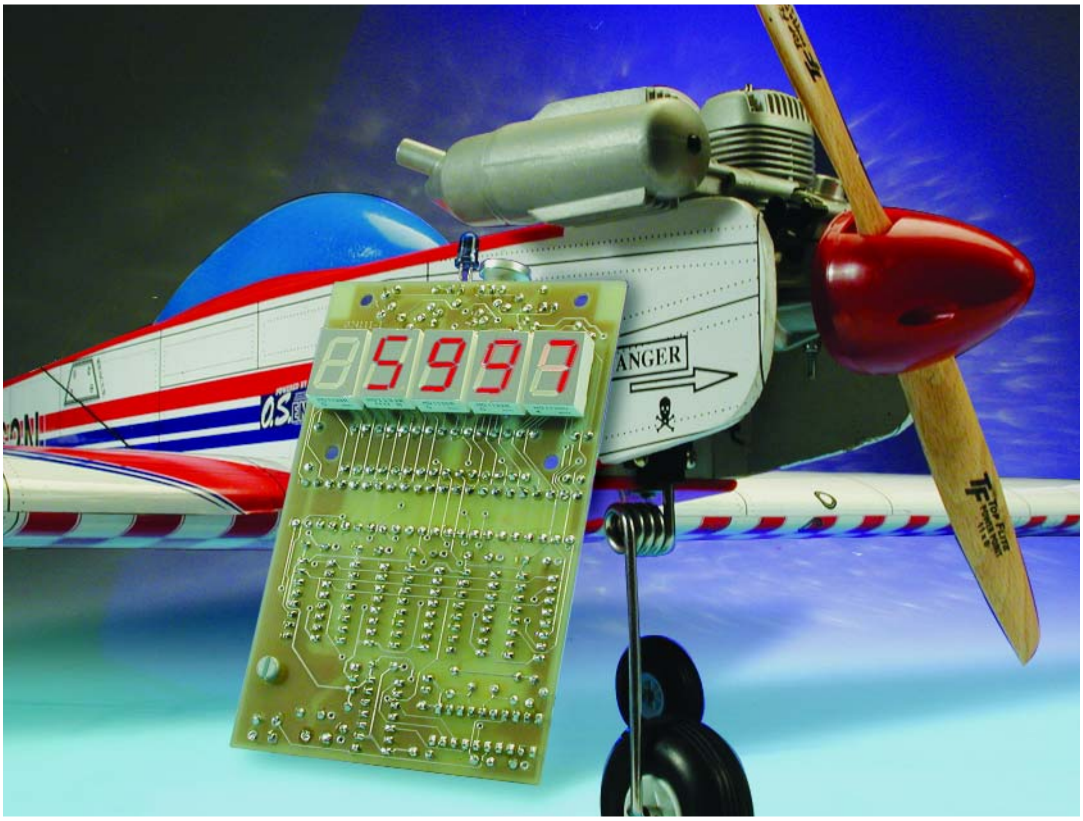
From this it can be seen that *both* variables have been measured, combining the advantages of both methods described earlier.

The circuit

The circuit (**Figure 3**) can be divided into three sections: the sensor circuit, the controller and the display. The sensor circuit (built round IC1) uses a BPW40 as the light-sensitive device. This is in fact just a transistor without a base connection, where the die has been left exposed to the light. The more light that reaches the die, the more the transistor conducts.

IC1a tries to keep the voltage on pin 3 at about 1.4 V. Should the voltage rise above this, the output of the opamp will also rise, as will the voltage at the base of T2 after a short delay. This results in a reduced current flow through T1, which in turn reduces the voltage on pin3 (via buffer T3).

R4, C1 and R3 form a low-pass filter, slowing down the regulation. This causes the average voltage on pin 3 to be 1.4 V, but fast pulses (such as the shadow of a propeller blade) are not smoothed out. IC1b functions as a comparator, which converts the pulses into a logic level for the processor.



IC2 is an old favourite of ours, the 89C2051. This controller barely requires an introduction, since it has often been used in Elektor Electronics circuits. In this circuit a simple program takes care of the pulse measurements, driving the display and reacting to pushbuttons S2 and S3. Since this controller has an internal flash memory for the program, there is no need for an external EPROM, address decoder and associated logic gates. This keeps the controller section very compact. IC2 is available fully programmed, but you can also program it yourself using the software that is available separately on floppy disk.

IC5 is used as a buffer for signals from the controller. These signals are used to drive the serial shift registers (IC3, IC4 and IC6 to IC8). The outputs of these shift registers drive the displays directly (LD1 to LD5).

For the displays we recommend the HD1133 O-K. These are bright enough at a relatively low current. Other common cathode types with the same pin-out can also be used of course, but you may have to change the values of resistors R13 to R52 to increase the brightness of the displays. S1, D2 and R53 provide an artificial light for cases when there is not enough background light, as we explained earlier.

The power supply for the circuit is very simple. A 9 V battery powers the circuit, where a 7805 (IC9) is used to provide a stable 5 V to the controller.

Construction

A double-sided PCB has been designed for the circuit, and is shown in **Figure 4**. Populating this board isn't difficult, as long as you go about it the right way. First the component side has to be populated (so without the displays), with the exception of D1 and R4. You should use two 4-pin SIL sockets for IC1 instead of a standard 8-pin IC socket, otherwise some of the pins of the displays will no longer be accessible for soldering! T1 and D2 have to be mounted at right angles to the board so that they can stick through the front of the case. Cut off the ends of the leads of C2, R6, R11 and the SIL sockets for IC1 as close as possible to the board, as they will otherwise get in the way when the displays are mounted.

For D1 you have to keep in mind that a 1V4 zener diode is similar to two ordinary diodes connected in series. The means that the connections are exactly the opposite of what you would expect; the band on the diode in this case corresponds to the *anode*). Photo transistor T1 (BPW40) looks similar to an LED; the longer leg is the collector.

The displays are mounted next. If the board has been populated according to the instructions, none of the display pins should be obstructed and they are easily soldered to the board. And finally it's the turn of D1 and R4. Bend their leads the right distance apart and cut them such that they lie neatly on the board. If they are too long they will come into contact with the displays and it won't be possible to solder them neatly onto the board. D1 and R4 have to be soldered onto the component side of the board, since the solder side is obstructed by the displays. This doesn't matter, since the PCB has plated-through holes.

You are of course free to choose your own enclosure. However, the PCB has been designed to fit into an EG1530B made by Bopla, although there are likely to be many other suitable enclosures.

Operation and test

When the PCB has been fully populated and checked we can start testing its operation.

First the rev counter has to be switched on, after which all displays should show a '–' sign.

If we then hold the sensor near a burning incandescent lamp, the display should show a value in the region of 3000 RPM after about a second. This is not an error, but a consequence of the fact that an incandescent lamp flickers with a frequency of 100 Hz when it is connected to the 50 Hz mains supply. The sensor therefore 'sees' a frequency of 100 Hz. The rev counter is initialised for use with a two-bladed propeller when it is switched on. Using these figures in the formula:

RPM = frequency x 60 / number of blades,

this gives:

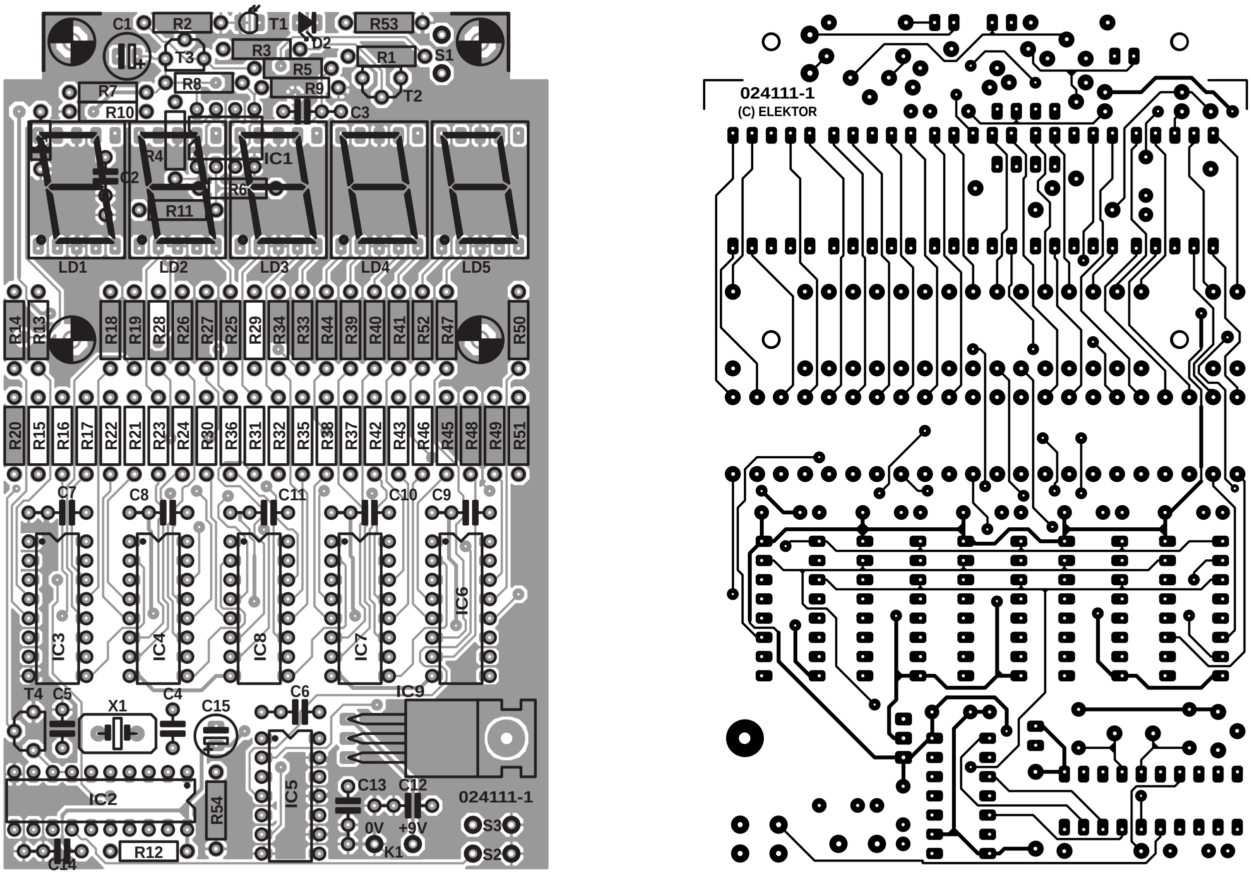


Figure 4. A well-arranged double-sided PCB has been designed for the rev counter.

RPM = 100 Hz x 60 / 2 = 3000 RPM

Pressing the button with the '+' causes the number of propeller blades to increase by one. The button with the '–' obviously does the exact opposite. The controller only looks at these buttons once per second, so it may not react immediately when you press a button. If you hold a button down longer, the setting will change by one every second. Every time that the controller changes the number of blades, this will be shown on the display as —X—, where 'X' is the current number of propeller blades.

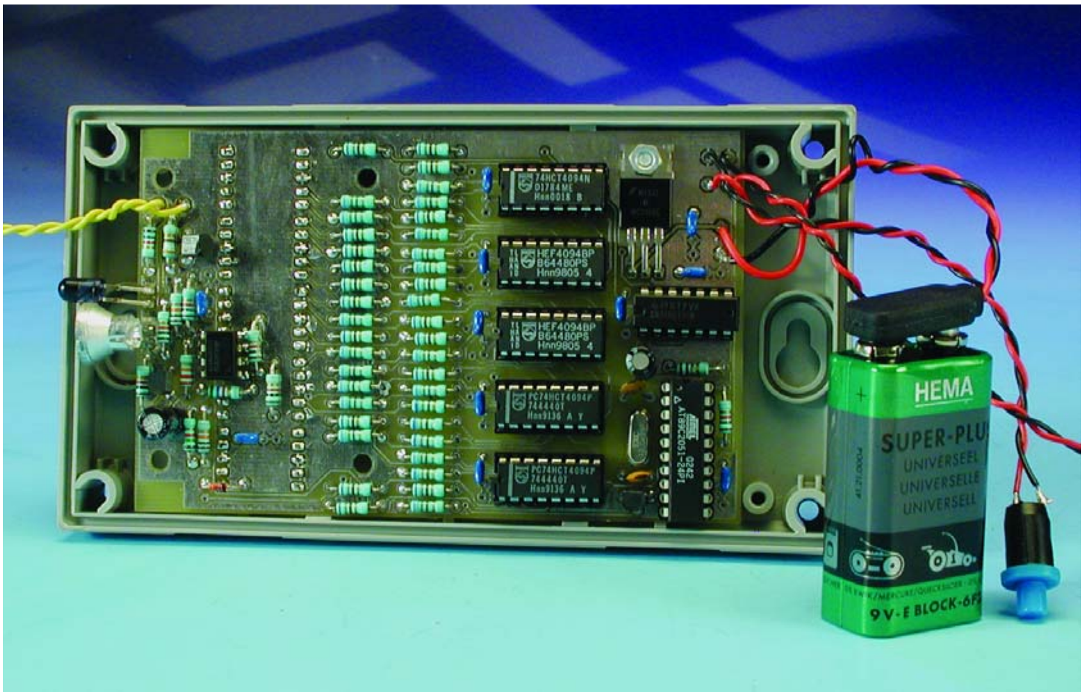
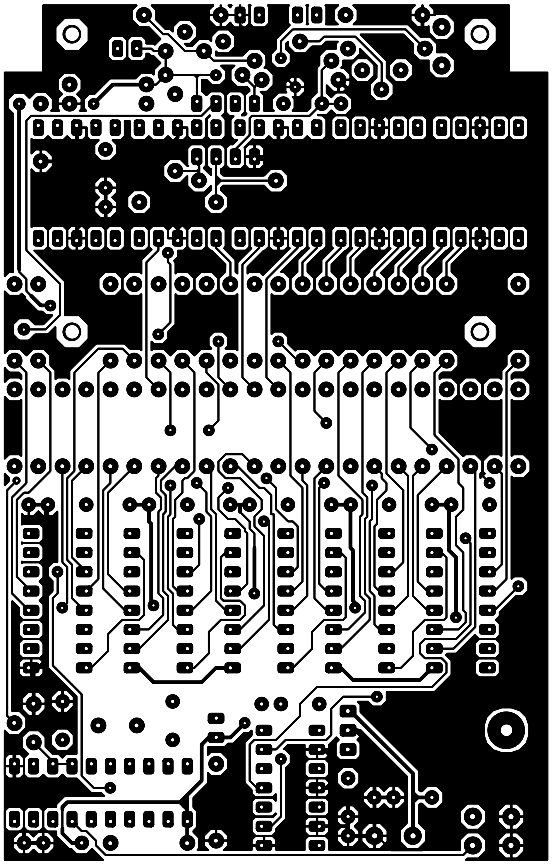


Figure 5.The component side of a fully assembled board.

If it has passed all tests, it's become time to try the rev counter on a real model airplane. First, set the rev counter to the right number of blades for the propeller. Start the engine of the airplane as usual (hopefully in a safe manner!) and let somebody else hold it for you. Hold the sensor in front of the blades such that the blades throw a shadow on the senor at every revolution. **Take great**



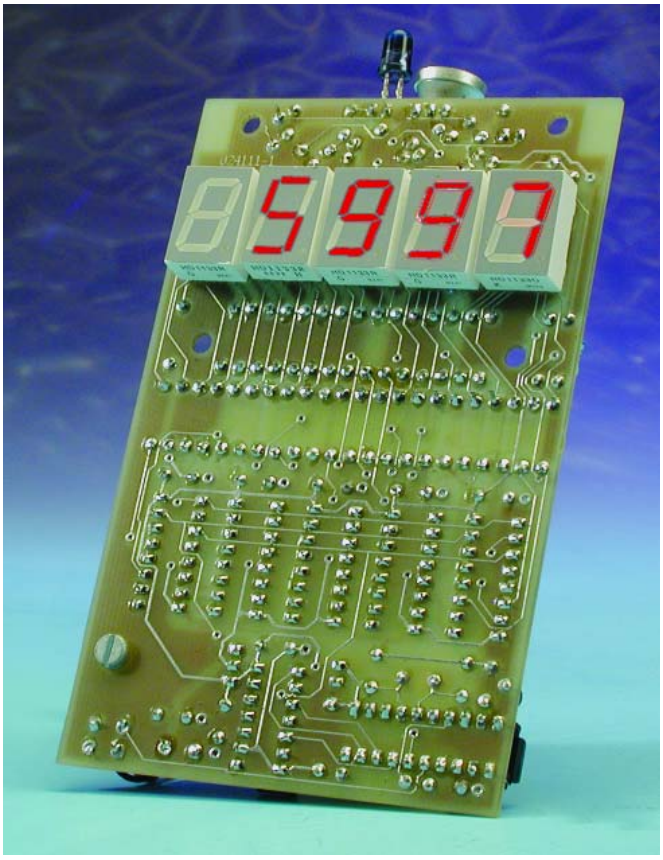


Figure 6. The five displays are mounted on the solder side of the board.

COMPONENTS LIST

Resistors:
R1 = 390Ω
R2,R3,R6,R9,R11,R12,R53,R54 = 1kΩ
R4 = 5kΩ6
R5,R10 = 10kΩ
R7 = 1kΩ5
R8 = 22kΩ
R13-R52 = 680Ω

Capacitors:
C1 = 220µF 16V radial
C2,C3,C6-C14 = 100nF
C4,C5 = 22pF
C15 = 10µF 16V radial

Semiconductors:
D1 = zener diode 1.4V 500mW
D2 = IR LED (e.g., LD271)
IC1 = TLC272CP
IC2 = 89C2051-12PC, programmed, order code **024111-41** (see Readers Services page)
IC3,IC4,IC6,IC7,IC8 = 74HCT4094
IC5 = 74HC04
IC9 = 7805
LD1-LD5 = HD1133 O-K
T1 = BPW40
T2 = BC557
T3,T4 = BC547

Miscellaneous:
K1 = 9-V battery clip
S1,S2,S3 = pushbutton, 1 make contact
X1 = 12MHz quartz crystal
Enclosure: Bopla EG1530B
PCB, order code **024111-1** (see Readers Services page)
Disk, contains source- and hex code, order code **024111-11** (see Readers Services page)

care that the circuit does not come into contact with the rotating propeller. This could be very dangerous, since the power present in even a small propeller shouldn't be underestimated! A measurement will now be taken at about one-second intervals and the display will show the exact engine speed. If this is not the case, you can turn on the IR LED by pressing push-button S1. This emits infrared light, which is invisible, but which the sensor certainly reacts to. In this way, a measurement can still be taken in weak light!

Other applications

The use of the rev counter is not restricted to measuring propeller speeds. In principle it can be used to measure anything that produces a fast-changing light intensity, as long as there is a big enough difference between the light and dark periods. The display will still show a figure in RPM, which has to be converted into a frequency. To simplify this, the circuit should be set to use six blades, when the display will show the frequency multiplied by 10 (for the previously mentioned incandescent light with a frequency of 100 Hz this will be shown as 1000 on the display).

In this way it would be possible to measure the frame-rate of your computer monitor. The best contrast between light and 'shadow' is obtained when the monitor shows a white screen. Should the rev counter give an obviously incorrect value, you can try increasing the brightness of the monitor.

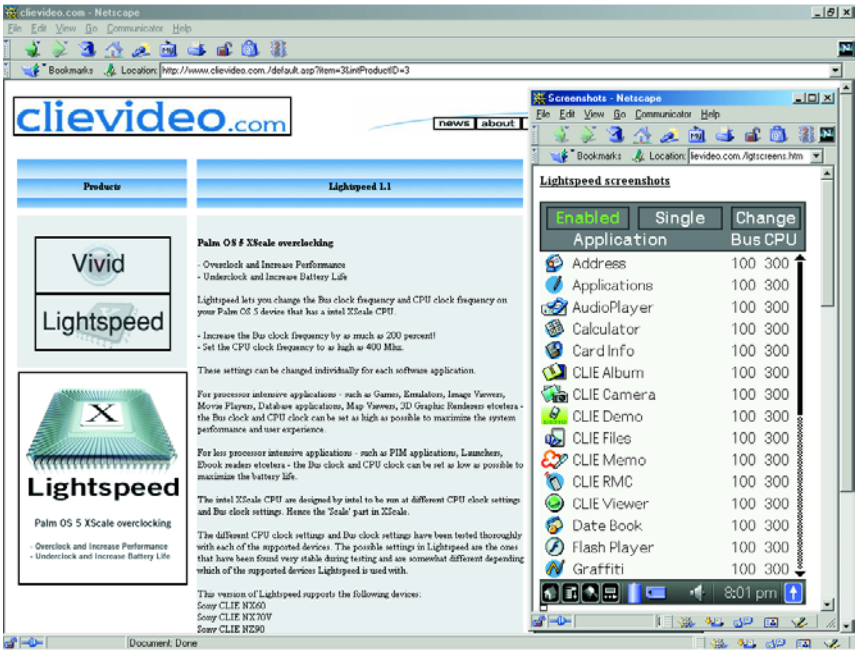
(024111-1)

PDA & PPC Overclocking

Hints & kinks for higher speed

By Harry Baggen

Over the past few years the Personal Digital Assistant or PDA has seen an evolution from a simple electronic diary to a miniature computer that can do almost anything its larger brother the desktop PC is known to be capable of, from keeping an appointment book right up to playing movies. Not surprisingly, fervent PDA users hunger after speed & more speed, so CPU overclocking is topical once again.


clievideo.com - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://www.clievideo.com./default.asp?item=3&intProductID=3
clievideo.com
news
about
Products
Vivid
Lightspeed
Lightspeed
Palm OS 5 XScale overclocking
- Overclock and Increase Performance
- Underclock and Increase Battery Life
Document: Done
Screenshots - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: levideo.com./lgtscreens.htm
Enabled
Single
Change
8:01 pm

Lightspeed 1.1

Palm OS 5 XScale overclocking

- Overclock and Increase Performance
- Underclock and Increase Battery Life

Lightspeed lets you change the Bus clock frequency and CPU clock frequency on your Palm OS 5 device that has a intel XScale CPU.

- Increase the Bus clock frequency by as much as 200 percent!
- Set the CPU clock frequency to as high as 400 Mhz.

These settings can be changed individually for each software application.

For processor intensive applications - such as Games, Emulators, Image Viewers, Movie Players, Database applications, Map Viewers, 3D Graphic Renderers etcetera - the Bus clock and CPU clock can be set as high as possible to maximize the system performance and user experience.

For less processor intensive applications - such as PIM applications, Launchers, Ebook readers etcetera - the Bus clock and CPU clock can be set as low as possible to maximize the battery life.

The intel XScale CPU are designed by intel to be run at different CPU clock settings and Bus clock settings. Hence the 'Scale' part in XScale.

The different CPU clock settings and Bus clock settings have been tested thoroughly with each of the supported devices. The possible settings in Lightspeed are the ones that have been found very stable during testing and are somewhat different depending which of the supported devices Lightspeed is used with.

This version of Lightspeed supports the following devices:
Sony CLIE NX60
Sony CLIE NX70V
Sony CLIE NZ90

Lightspeed screenshots

Application	Bus	CPU
Address	100	300
Applications	100	300
AudioPlayer	100	300
Calculator	100	300
Card Info	100	300
CLIE Album	100	300
CLIE Camera	100	300
CLIE Demo	100	300
CLIE Files	100	300
CLIE Memo	100	300
CLIE RMC	100	300
CLIE Viewer	100	300
Date Book	100	300
Flash Player	100	300
Graffiti	100	300

or quite some time the PDA has been in the exclusive realms vaguely defined as 'business', where it was really no more than a show-off electronic diary. However, today's Palm and Pocket computers offer so much power and performance in terms of multimedia that it is seems wasteful to limit their use to scheduling the odd appointment. Most PDAs of the latest crop offer a clearly legible colour screen, a large amount of memory (if necessary, using expansion cards) and quite impressive processing speeds. A galaxy of software is available at reasonable prices, and some of it is even free! For example, it has become possible to use your portable electronic diary to play games, play back MP3 files (using the headphone connector) or even watch a complete movie.

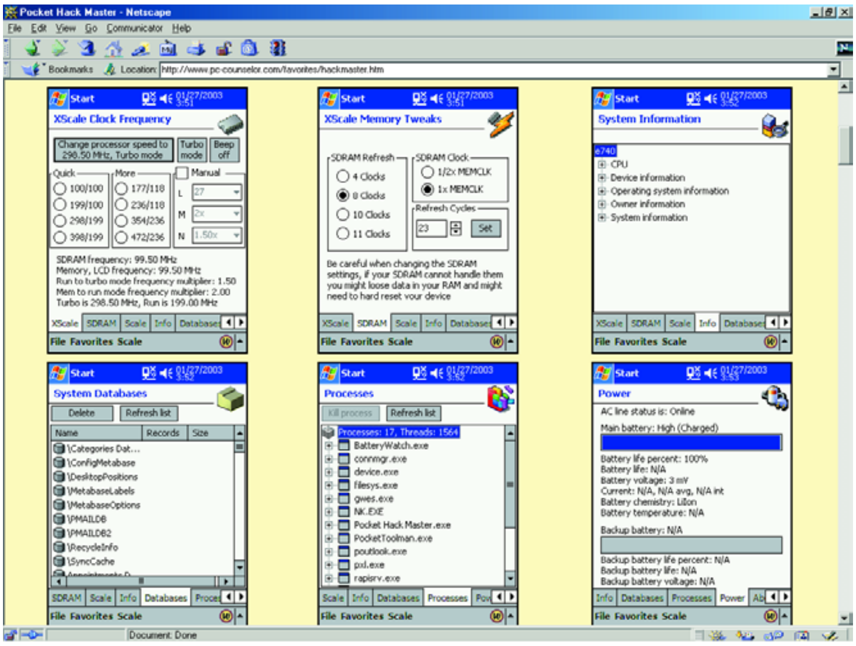
The last application being quite demanding in terms of processor speed and data throughput, several users have started to explore the possibilities of 'souping up' their little machines. The effects, desires and trends observed in this quest are really the same as with regular PCs: the faster the better. Even a quick search across the Internet is sufficient to confirm that a lot of software is available that allows a PDA to be souped up in a simple manner.

For Palm computers we should mention **Afterburner** [1] for use on Palm PDAs with a Dragonball processor, like the Palm III and V. This little program gives a considerable increase in the clock frequency depending on the PDA type. Unfortunately, some users have reported display interference, so some caution is advised in this area.

A nicely styled (graphically, that is) program is called **FastCPU** [2], which is suitable for nearly all Dragonball processors running Palm OS releases 3.5, 4.0 and 4.1.

Just like the latest Pocket PCs, the newer Palm OS computers (including several Sony Cliё models like the NX60 and 70) boast an Xscale processor that can be set to a higher speed using a program called **Lightspeed** [3].

For the Pocket PCs, similar programs may be found on the net. **Pocket Hack Master** is an excep-



Pocket Hack Master - Netscape

Location: http://www.pc-counselor.com/favorites/hackmaster.htm

XScale Clock Frequency

Change processor speed to 298.50 MHz, Turbo mode

Quick: ○ 100/100 ○ 199/100 ○ 298/199 ○ 398/199

More: ○ 177/118 ○ 236/118 ○ 354/236 ○ 472/236

□ Manual: L 27, M 2x, N 1.50x

SDRAM frequency: 99.50 MHz
Memory, LCD frequency: 99.50 MHz
Run to turbo mode frequency multiplier: 1.50
Mem to run mode frequency multiplier: 2.00
Turbo is 298.50 MHz, Run is 199.00 MHz

XScale Memory Tweaks

SDRAM Refresh: ○ 4 Clocks ● 8 Clocks ○ 10 Clocks ○ 11 Clocks

SDRAM Clock: ○ 1/2x MEMCLK ● 1x MEMCLK

Refresh Cycles: 23

Be careful when changing the SDRAM settings, if your SDRAM cannot handle them you might loose data in your RAM and might need to hard reset your device

System Information

e740
CPU
Device information
Operating system information
Owner information
System information

System Databases

Name	Records	Size
\Categories Dat...		
\ConfigMetabase		
\DesktopPositions		
\MetabaseLabels		
\MetabaseOptions		
\PMAILDB		
\PMAILDB2		
\RecycleInfo		
\SyncCache		

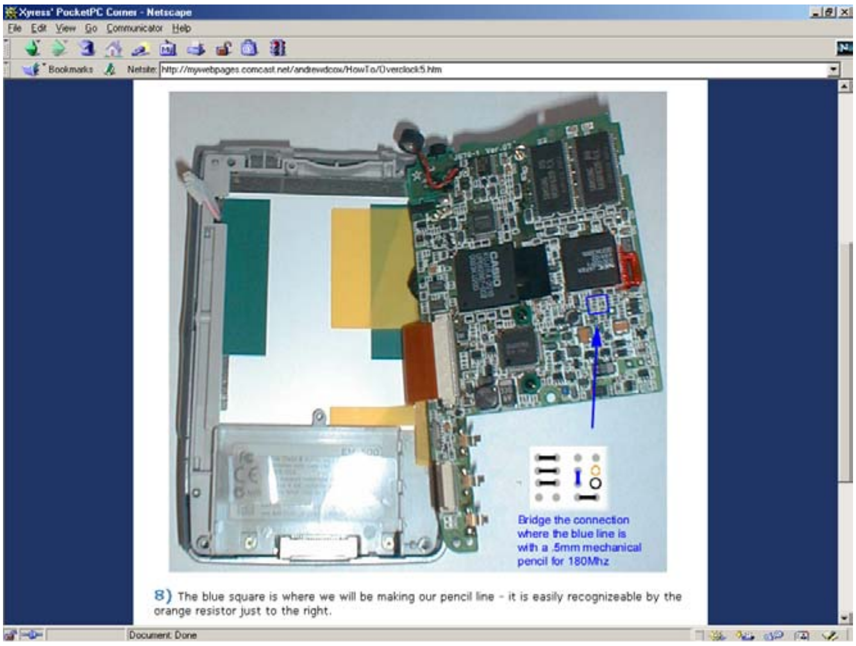
Processes

Processes: 17, Threads: 1564
BatteryWatch.exe
connmgr.exe
device.exe
filesys.exe
gwes.exe
NK.EXE
Pocket Hack Master.exe
PocketToolman.exe
poutlook.exe
pxl.exe
rapisrv.exe

Power

AC line status is: Online
Main battery: High (Charged)
Battery life percent: 100%
Battery life: N/A
Battery voltage: 3 mV
Current: N/A, N/A avg, N/A int
Battery chemistry: LiIon
Battery temperature: N/A
Backup battery: N/A
Backup battery life percent: N/A
Backup battery life: N/A
Backup battery voltage: N/A

Document: Done



Xyress' PocketPC Corner - Netscape

Netsite: http://mywebpages.comcast.net/andrewdcox/HowTo/Overclock5.htm

Bridge the connection where the blue line is with a .5mm mechanical pencil for 180Mhz

8) The blue square is where we will be making our pencil line - it is easily recognizeable by the orange resistor just to the right.

Document: Done

tionally extensive program for overclocking of PPCs with an XScale processor. Test utilities are available to see at which clock speed combinations the PPC continues to operate as you expect it to Not only the CPU clock frequency may be changed — you can also have a go at increasing the refresh speed and the SDRAM clock. The program is capable of measuring the CPU load and, if necessary, switch over to a higher clock speed. That is possible because XScale processors allow their clock frequency to be changed on the fly. The program comprises a number of information windows. The **PC Counselor** website [4] contains an extensive and illustrated description of this program.

A simpler program with fewer bells and whistles called **JS Overclock** [5], comes free of charge. It is suitable for most PDAs with an XScale processor and allows the clock to be adjusted within a range of 162 MHz to 236 MHz.

Clear Speed [6] is an overclocking utility specially written for the HP iPaq 1910/1915. According to several Internet sources this type of PPC is a particularly suitable candidate for overclocking. Reportedly, users are able to run their 1910 or 1915 at 300 MHz instead of the standard 200 MHz using a program like Clear Speed. A free alternative to it, **Speed Stepper** [7], comes with a utility called Turbo Tray that makes controlled (i.e. stepwise) increase of the clock frequency an uncomplicated affair.

Just as with the ubiquitous desktop PC, Pocket PCs are subject to a lot of experimentation in the hardware department with the clear aim of making the thing run faster. At the **Xyress' Pocket PC Corner** website [8] you'll find a description how to modify a Casio EM500 for higher clock speed.

With any attempt at overclocking a CPU or indeed any other logic circuit there is a real risk of system 'hang-ups' owing to too high clock frequencies. When that happens, the only thing you can do is give a 'hard reset' which unfortunately means you lose all current data and programs. It is, therefore, advisable to back up all your PDA data before you start using overclocking programs.

Incidentally, most of the programs mentioned in this article also allow the PDA to run slower, which may be useful if you want to reduce power consumption with a view to stretching battery life.

Most of the software mentioned here is available from several sources including **Handango** [9] at a price of about 7 to 15 pounds. You can also obtain trial and demo versions of a number of programs, which give you an opportunity to see if the relevant program actually runs on your Palm or Pocket PC.

(035067-1)

Internet addresses

[1] Afterburner: www.palmblvd.com/software/pc/Afterburner-2000-10-05-palm-pc.html
[2] FastCPU: www.megasoft2000.com/palm_division/fastcpu.htm
[3] Lightspeed: www.clievideo.com./default.asp?item=3&intProductID=3
[4] PC Counselor (Hack Master): www.pc-counselor.com/favorites/hackmaster.htm
[5] JS Overclock: www.jimmysoftware.com/Software/Overclock/
[6] Clear Speed: http://revolution.cx/speed/speed.htm
[7] Speed stepper: www.elevatedstudios.com/pocket/
[8] Overclock your EM-500: http://mywebpages.comcast.net/andrewdcox/HowTo/Overclock.htm
[9] Handango: www.handango.com