ELEKTOR ELECTRONICS

THE ELECTRONICS & COMPUTER MAGAZINE

OCTOBER 2001
£3.20

www.elektor-electronics.co.uk

Infrared Code Analyser

Wobbler Mini Robot

MINIATURE PCM MODEL CONTROL

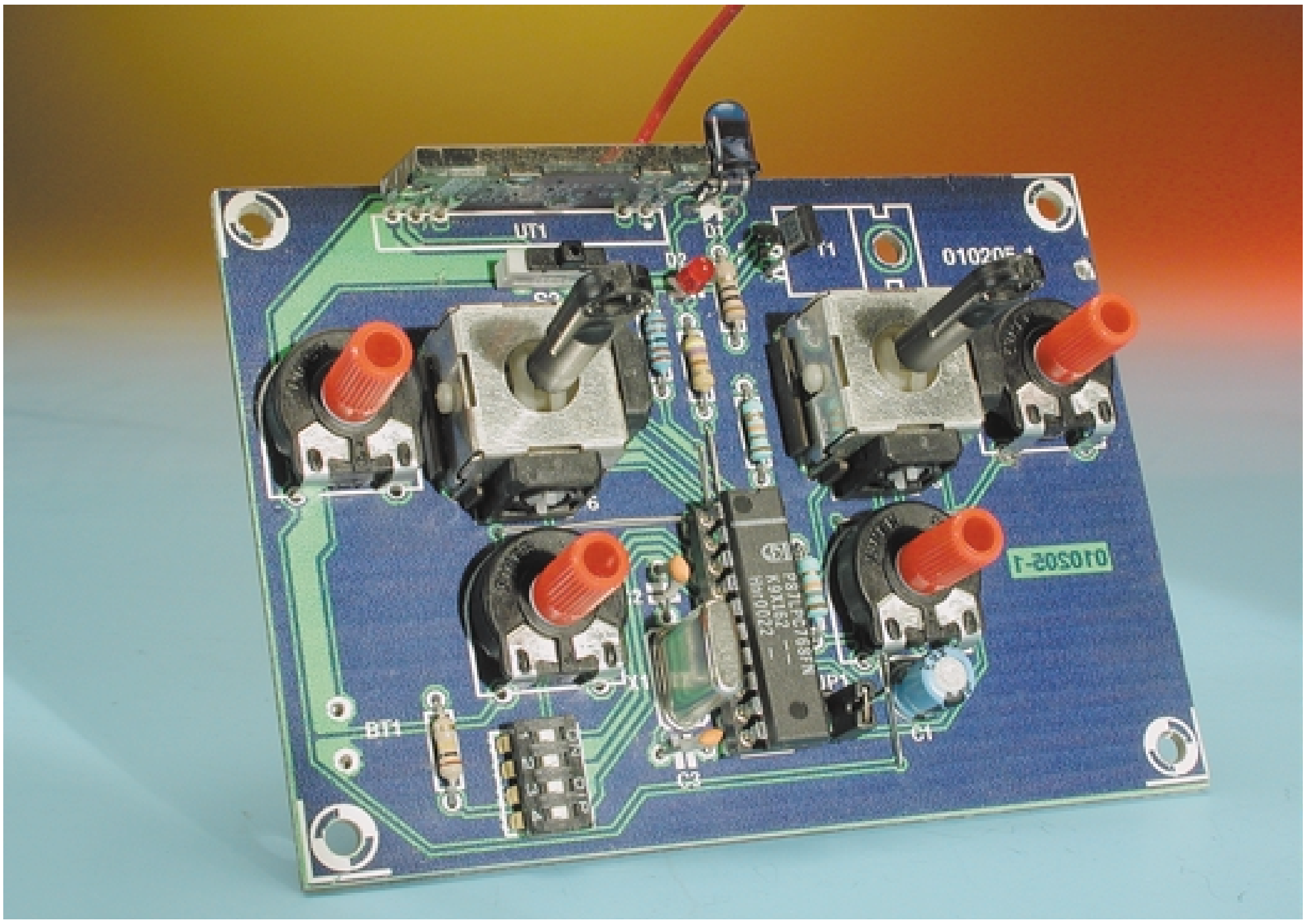
- Personal Mini Webserver (2)
- Intelligent Sensor / Actuator Control
- RS232 In-Line Data Spy
- Tube Clipper
- RS485 Meets CAN

10
9 770268 451098

Miniature PCM Remote Control

part 1: transmit at 433 MHz and 950 nm!

Here's a design that should gladden the hearts of many model builders. A really small proportional remote control unit built using standard - low-cost components. It's ideal for loads of applications and so flexible that simply by changing on-board jumpers you are free to choose the transmission medium: radio, Infrared or two-wire!



Remote control main features:

- – Lightweight
- – Simple construction with no set-up or alignment
- – 6 channels → 4 proportional and 2 digital
- – Switched outputs for 5 A load
- – Four standard servo connectors
- – Trim function on 4 channels
- – Small, light and compact
- – PCM transmission method
- – Selectable Infrared or radio operation
- – Current saving transmission method
- – Type-approved and licence-exempt 433MHz Tx and Rx modules
- – Voltage supervision in both the transmitter and receiver
- – Built-in speed regulator in receiver
- – Built-in soft start switch in receiver
- – Speed regulator and softstart controller suitable for loads <15A
- – Battery Low indicator
- – Servo reverse (change the servo travel direction) for all 4 servos
- – Operating voltage: transmitter 3.3 V to 4.8 V (3x AA Alkaline cells or 4x Nicad cells)
- – Receiver 5 V from on-board BEC voltage regulator

It used to be that constructional projects using RF transmitters and receivers would only be attempted by the more courageous electronics enthusiasts. Thankfully since the advent of complete integrated RF modules we can forget the tedious set-up and adjustment process and just treat the RF stage as another plug-in building block. The design described here is very flexible and can use these RF modules to implement a radio remote control. Microcontrollers are also employed in the transmitter and receiver to replace the discrete shift registers, clock generators and timing circuitry that you normally find in remote control designs. The result is a compact, reliable and low cost remote control unit. The low radiated power emitted by these licence-exempt radio modules limits the range of the units so this design is only really suitable for controlling indoor models.

A speed controller is also built into the receiver allowing direct connection to the models drive motor. Using the infrared control option a low-cost 4-channel proportional control system can be built that simply cannot be matched for price and performance by anything available commercially. The two-wire control option is useful if the device to be controlled is in a fixed location, for example the receiver could drive two model servos controlling the pan and tilt of a remote security camera.

This remote control design is a good demonstration of how microcontrollers can be used to implement features that not so long ago would require dedicated circuitry. The design need not stop here. If, for example you decide that a menu driven transmitter together with a keypad and LCD would better fulfil your needs then it is entirely possible to implement this. Many features of the microcontroller together with unused ROM are just waiting for an inspired programmer to make good use of them.

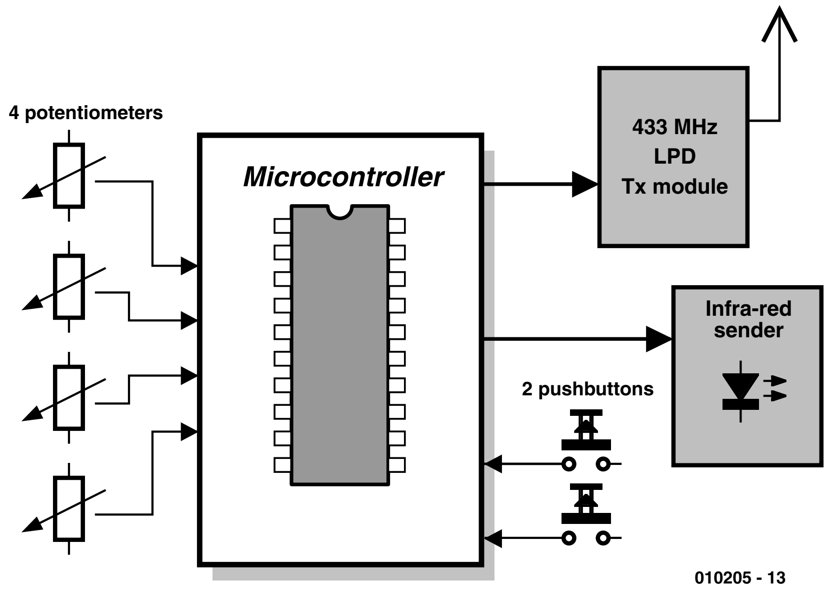


Figure 1. Transmitter block diagram.

The transmitter

The block diagram shown in **Figure 1** indicates that a microcontroller does most of the donkey-work. It digitises the analogue resistance values of the four input potentiometers, checks for push button presses, and generates the transmitted signal protocol. The signal can be transmitted at RF or by infrared.

Even if you are only intending to use the transmitter at RF it is worth fitting the infrared components to make the finished unit more flexible. The additional cost incurred by fitting the IR diode and its drive transistor is relatively low.

During initialisation the transmitters microcontroller will read the state of the input at port pin P0.7. If it is high (jumper JP2 not fitted) then RF control will occur. If it is low (jumper JP2 fitted) then signals will be sent using infrared. When the controller sends infrared signals the driver stage uses more peak current than with the RF option. The controller formats the output signal differently by modulating the Pulse Code Modulation (PCM) data at 36 kHz and reducing the data rate in order to prolong battery life. The same modulation is used by remote controllers for TVs etc and gives good transmission security.

The components used for the transmitter circuit shown in **Figure 2** are low-cost and widely available from numerous suppliers but its worth taking a closer look at some of the more unusual items:

The Joystick assembly

The remote control system should be small, economical, simple to build and suitable for

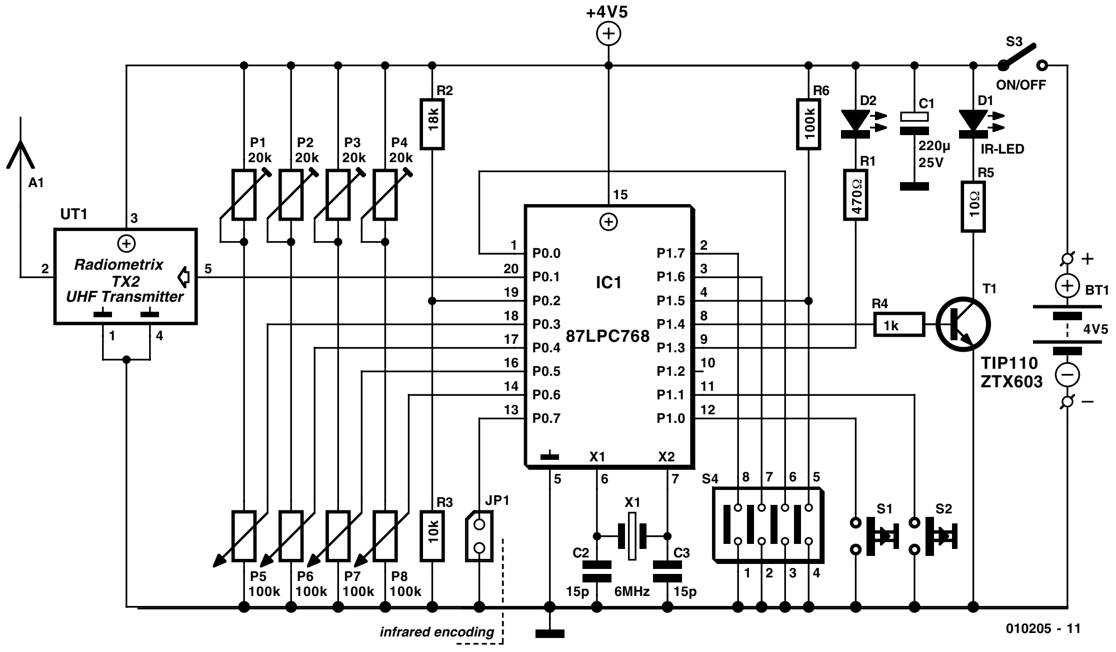


Figure 2. Transmitter circuit diagram.

use with standard model servos from Graupner, Multiplex, Futaba and Ripmax etc. One of the first hurdles in building this proportional remote control system was to track down an economical joystick assembly to use for control input. The US Company CTS produces a suitable assembly. Its miniature joystick is self-centring, has low mechanical 'play' and can optionally have a switch fitted (**Figure 3**).

The most important technical features of this joystick are:

- 1 Million operating cycles
- Sturdy metal housing
- 2 Potentiometers
- Available potentiometer resistor values between 10 kΩ and 150 kΩ
- Selectable potentiometer resistor tolerance from 10% to 30%.
- Available with in-built pushbutton switches
- Switch contact resistance less than 0.1 Ω
- Switch rating 50 mA @ 12 V
- Push button life: 100,000 operations

Unfortunately the joystick control paddle passes through a circular opening in its housing. This has the effect of limiting control when the joystick is moved to the extremes of both the X and Y axis simultaneously. The joystick housing will therefore require a little modification and most competent modellers will have no problem in 'squaring out' the opening to allow maximum deflection of the paddle at all positions. The paddle itself can also be modified to improve the feel of the joystick. On the prototype we used some drilled-out plastic guides from ball-joints to extend the sticks.

Standard linear or turn potentiometers can be substituted for the joystick assembly if the combined X-Y control is not necessary in your model. In series with the potentiometers are also trim pots that allow the servo position to be finely adjusted or trimmed out. This feature is useful for example when setting up a model car to ensure that 'hands off' it steers straight ahead.

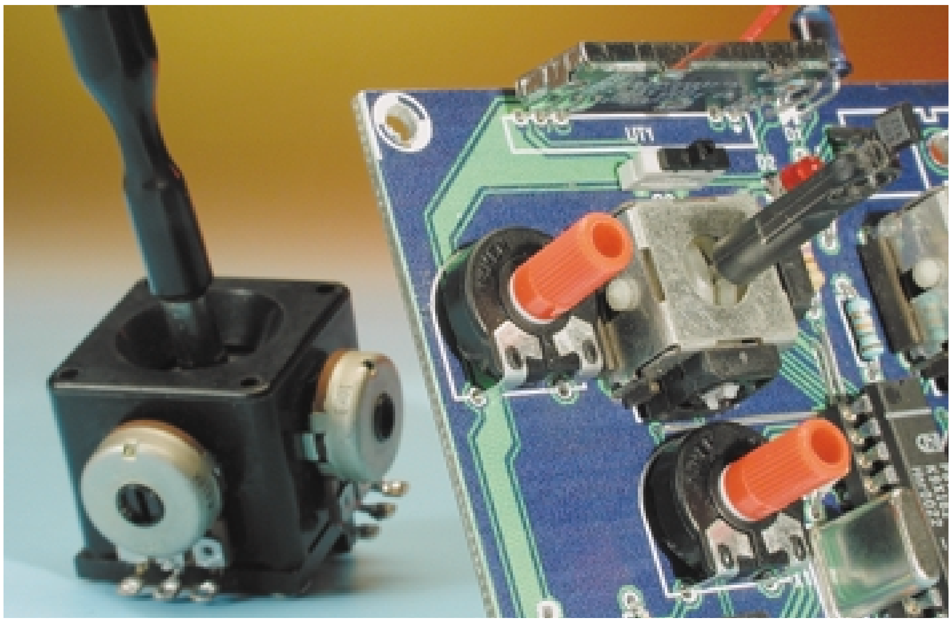


Figure 3. Miniature joystick assembly from CTS.

Microcontroller

The central control element in the transmitter is the 87LPC768 microcontroller from Philips. This device inputs analogue values from the joy-

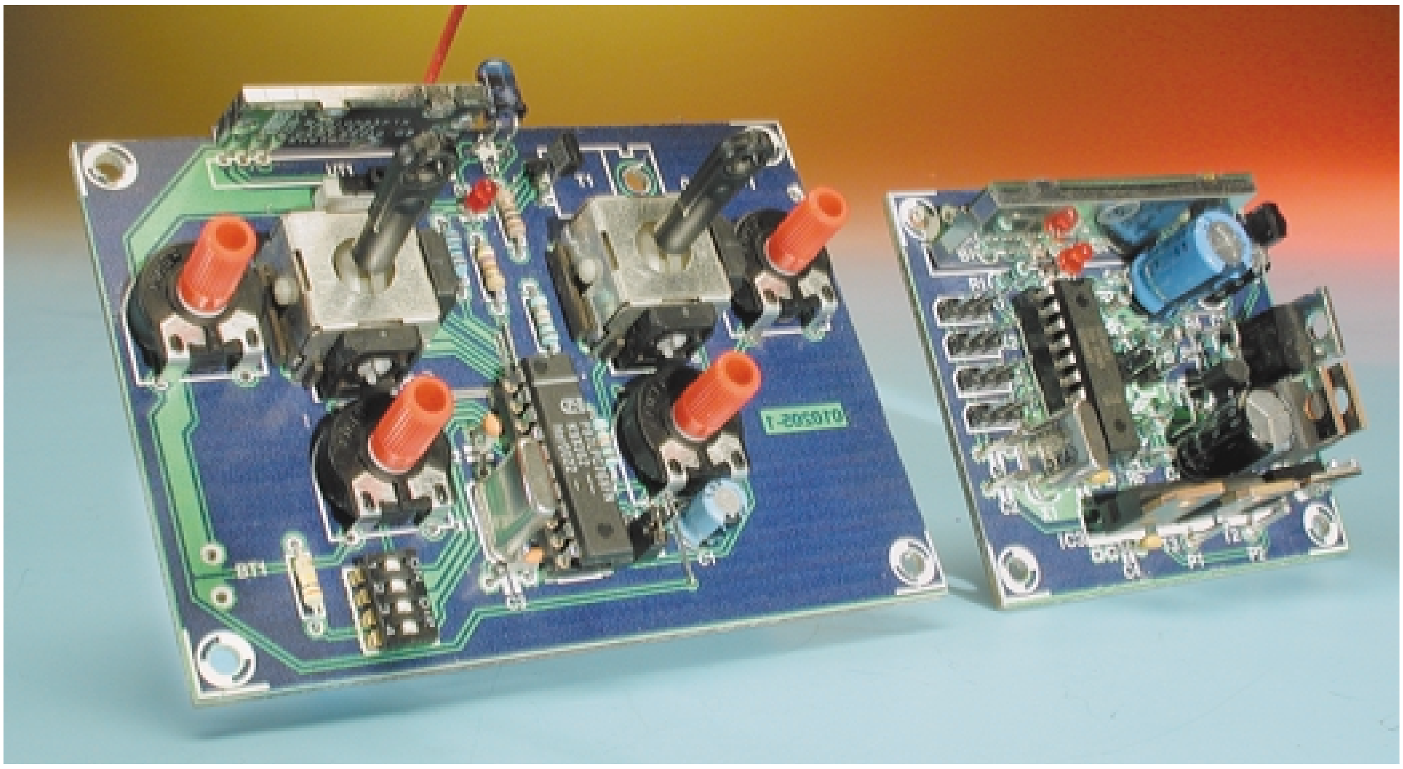


Figure 4. The 433 MHz transmitter and receiver modules.

sticks and pushbuttons and converts them into digital values using its analogue to digital (A/D) converters. A PCM signal is then produced and output to the IR diode or RF module.

The microcontroller is based on the well-established Intel 80C51 architecture so for programming there are many low-cost development tools including shareware available.

The 87LPC768 is described as a '*Low power, low price, low pin count microcontroller*'. A 6 MHz clock is necessary to cope with the processor intensive software routines and this produces a cycle time of 1 µs. The main controller features can be summarised as:

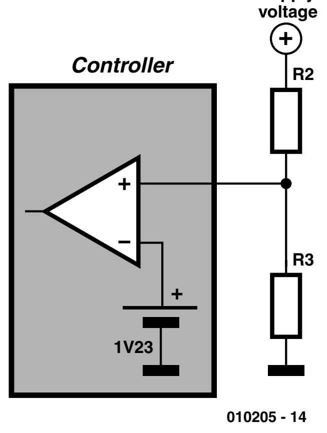


Figure 5. Internal comparator of the controller.

– 4 Kbytes ROM
– 128 Bytes RAM
– 32 Bytes Customer Code EEPROM
– Operating voltage 2.7-6.0 V
– Two 16 Bit Timer/Counters
– 4 channel Pulse Width Modulator (PWM) using 10 Bits
– 4 channel A/D converter, 8 Bit resolution, conversion time 9.3 µs @ 20 MHz clock frequency
– integrated reset
– Selectable internal RC oscillator.
– 20 mA driver current from output port
– 18 I/O-Pins maximum, when internal Reset and RC Oscillator is selected
– 2 analogue comparators
– I^{2}C interface
– Full duplex UART
– Serial In-circuit Programming

The RF Module

In recent times we have seen a big increase in the number of different applications using the 433 MHz band to send data. As well as supplying complete transmitter/receiver solutions some manufacturers produce integrated RF modules that can be used in many applications. The big advantage of these modules is that they do not require any set-up or tuning, they are supplied ready to go. The modules used in this circuit are produced by the company Radiometrix (**Figure 4**, together with the receiver module). The main features of the transmitter module are summarised in the following table:

Transmitter TX2

– Frequency 433 MHz, +9 dBm
– Modulation: FM
– Data rate 40 kbps max.
– Operating voltage 4.0 V to 6.0 Vdc
– Current consumption 6 mA

The transmitter RF module has five pins enabling it to be soldered or plugged directly to the PCB. The module will begin transmitting as soon as power is applied provided that the modulation input signal is high.

The Infrared transmitter

The infrared diode driver stage consists of Darlington transistor T1. The transistor is ideal as an LED power driver with a current gain of 2000 and a maximum collector current of 1 A. Two other alternative transistors are the BCX38 and TIPP110 but these have different pin-outs. Resistor R5 limits the diode peak current to approximately 240 mA. This figure was chosen as a trade off between battery life and transmitter range. R1 can be reduced to increase range as long as the peak forward current rating of the diode is not exceeded.

The IR LED type TSUS5201 emits IR light at 950 nm with a beam intensity of 230 mW/sr at 1.5 A with a half angle of ±15°.

In principle any IR LED can be used but if you do intend to use a substitute choose one with a beam intensity >200 mW/sr at 1.5 A and with a wavelength of 950 nm. The IR LED operating current is given by the formula:

$$I_{LED} = (V_{BAT} - V_{CESAT} - V_F) / R1$$

where:
V_{BAT} = operating voltage 4.5 V
$V_{CESAT} \approx 0.7$ V
V_F = forward voltage drop of an infrared diode ≈ 1.6 V

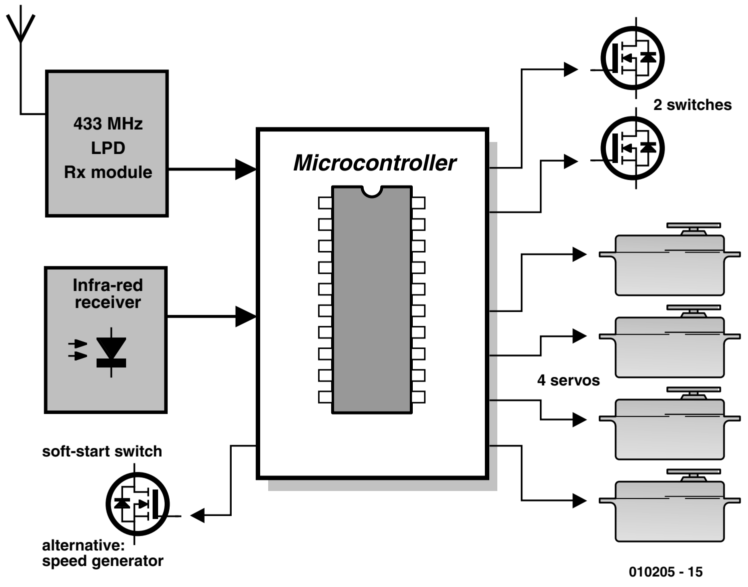


Figure 6. Receiver block diagram.

A crucial consideration affecting the reliability of the IR signals is the peak current that can be supplied by the battery. It is recommended to use either alkaline or Nicad cells to supply the high impulse current required by the IR driver stage. Cheap zinc-carbon cells may not be able to supply the necessary current.

Servo reverse
A 4 way DIP switch is conveniently included on the transmitter enabling the direction of travel of any of the

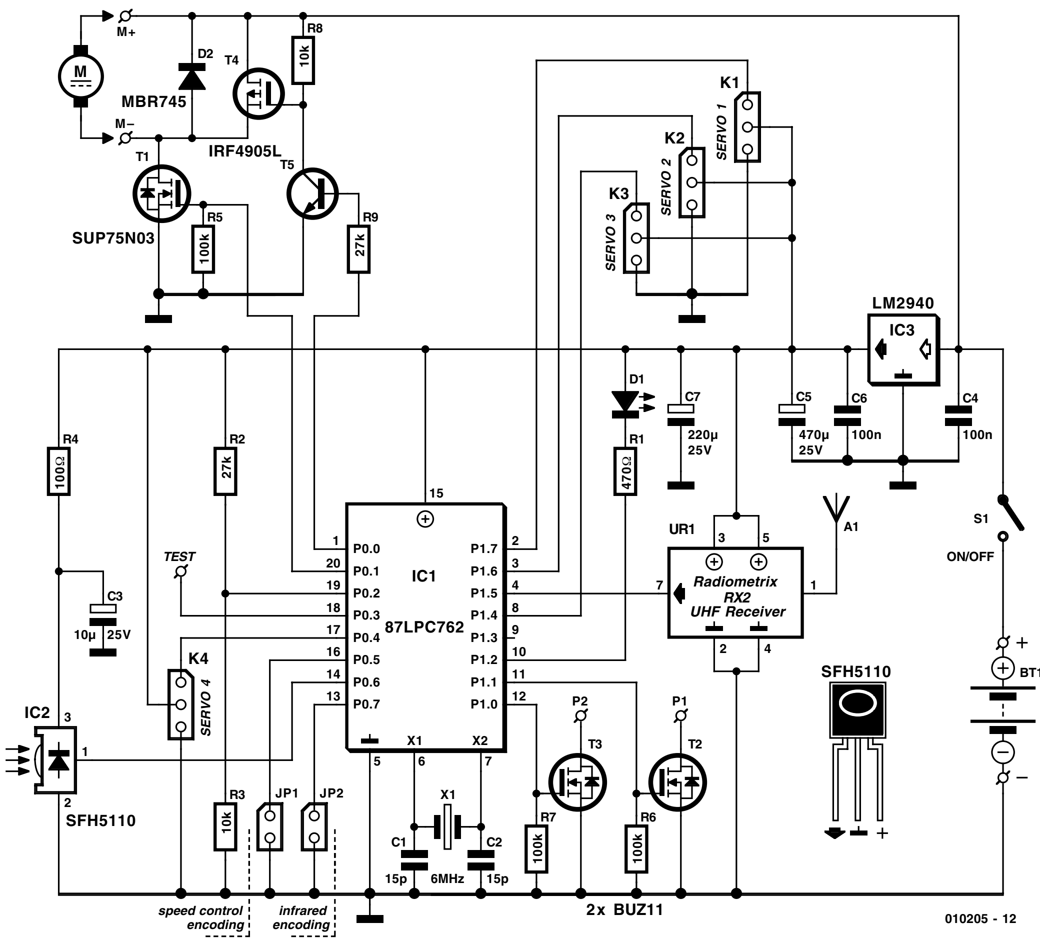


Figure 7. Receiver circuit diagram with two communication methods.

servos to be reversed.

Voltage monitoring
The minimum operating voltage of the microcontroller A/D converter is specified as 3.0 V. An internal comparator in the microcontroller is used to measure the supply voltage and detect when it gets close to this level. LED D1 is lit continually acting as a software watchdog but starts to blink when the supply voltage falls below 3.3 V indicating a low battery condition.

The microcontroller has an in-built comparator with a reference voltage level of 1.23 V (**Figure 5**). The values of external resistor R2 and R3 are calculated using the following formula:

Internal Reference = 1.23 V
R3 is chosen as 10 kΩ
R2 = 10 kΩ × [(threshold voltage/1.23 V) – 1]
R2 = 10 kΩ × [(3.3 V/1.23 V) –1)
R2 = 16.829 kΩ (nearest standard value: 18 kΩ)

The Receiver

The receiver block diagram shown in **Figure 6** is divided into several functional blocks like the transmitter.

The infrared receiver
Thanks to the growing popularity of infrared controlled equipment there are many integrated IR receiver chips on the market from different manufacturers all with broadly the same characteristics. The function of the IR receiver is to filter out any optical or electrical interference, demodulate the IR signal and amplify it. The receiver/demodulator IC's generally have three pins, two for connection to the power supply and one for the data output. The output pin can be connected directly to the input pin of a microcontroller. The IR receiver specified for this design is one of a family of devices that can operate in the frequency range from 30 kHz to 56 kHz. The main features of this device are:

- Integrated receiver diode and amplifier
- Electrically shielded
- Internal filter for PCM frequency.
- TTL and CMOS compatible.
- Active-low output.
- Low current consumption
- Interference suppression from continuous light sources (incandescent lamps or sunlight), or light sources pulsed at 36 kHz or other frequencies (fluorescent lamps).

The RF receiver for 433 MHz
The receiver module type RX2 from Radiometrix is the partner to the transmitter module type TX2. The RX2 consumes approximately 13 mA and is a very small, lightweight unit. A significant advantage for modellers is that at 433 MHz the antenna length needs to be only 15.5 cm long. The main features of the receiver module are:

- Receiver frequency 433.92 MHz
- Receiver type: double superheterodyne
- Sensitivity –107 dBm
- Operating voltage 3 V to 6 V
- Current consumption 14 mA
- Digital data output

As soon as power is applied to the RF receiver module it will be operational but without an in-range functioning transmitter module the receiver output will just be meaningless data.

The Microcontroller
Figure 7 shows the circuit diagram of the receiver. A central microcontroller is again used to do all the tricky stuff. It inputs digital control information from the transmitter signal and supplies control signals to four servos and two switched outputs.

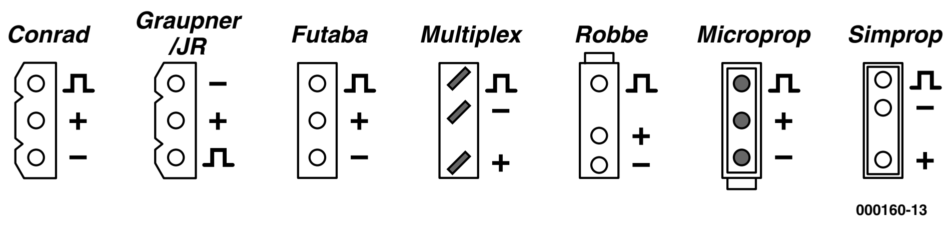
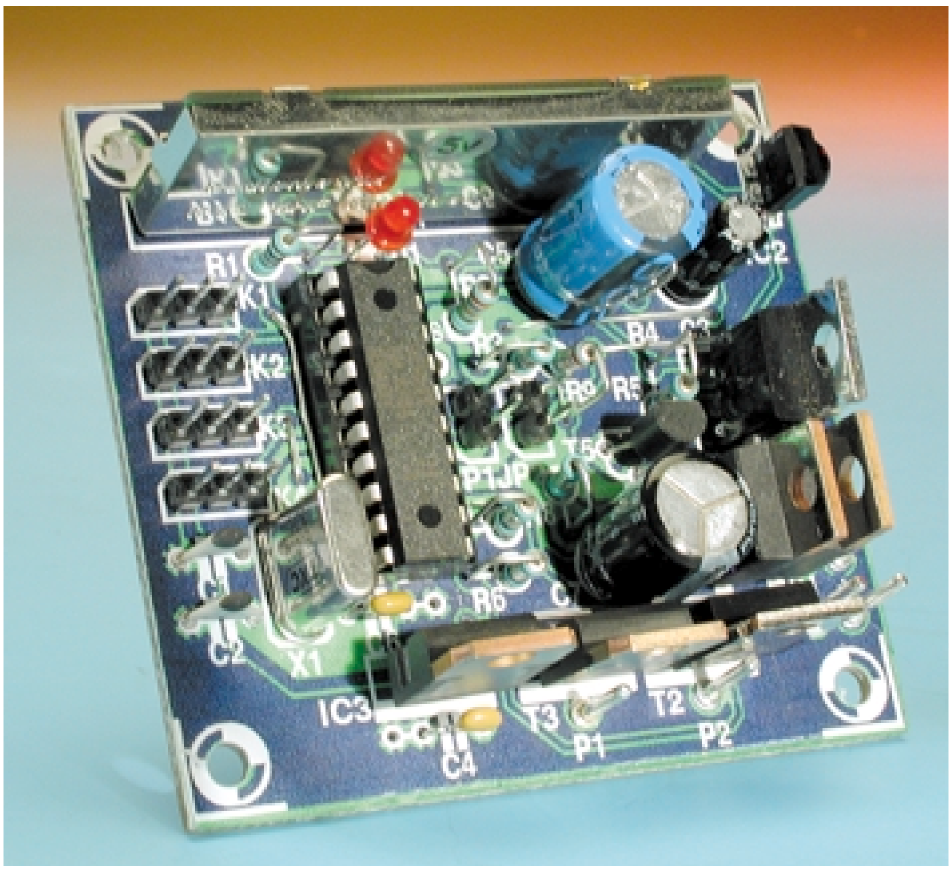


Figure 8. Pin-outs of different servo manufacturers.



The 87LPC762 controller used in the receiver is a slightly reduced version of the 87LPC768 used in the transmitter. It does not have the four channel pulse width modulators or the A/D converters and the size of the program ROM is only 2 kBytes. Apart from this the internal circuit is the same. A 6 MHz clock frequency is used once again.

Servo control

Model servos are pulse width controlled. A pulse width of 1.5 ms will cause the servo actuating arm to travel to its centre point and varying the pulse width between about 0.8 ms and 2.3 ms will move the servo arm from one end of its travel to the other. The actual maximum and minimum values of the pulse width vary slightly from one manufacturer to another but the difference is not critical. The pulse is repeated approximately every 20 ms.

The pulse width, pw, is calculated in the receiver microcontroller using the formula:

$$pw\ [\mu s] = \text{A/D value} \times 5 + 900$$

Where the A/D value is an eight bit value sent in the message from the transmitter (see the second part of this article in the coming month).

The output pulses are generated using a built-in timer in the microcontroller. The four output pulses are produced one after another in order to reduce the peak current that would otherwise occur if the pulses were sent simultaneously to all four servos. The microcontroller port pins connected to the servos are configured as push-pull outputs.

In the past, servos from different manufacturers had incompatible connectors. More recently equipment bought in the UK has standardised connectors conforming to the Futaba, Graupner/JR and Ripmax layout. The receiver PCB layout takes this into account so that the servo connector pins are suitable for this pin layout. If you are using older servos **Figure 8** gives information on the pin-outs.

Soft start/Speed regulator

The operating lifetime of motors and gearboxes can be extended by gently bringing them up to speed rather than switching them from zero to full power. Two methods of controlling the model motor are implemented in this design. Firstly if the ‘soft start’ option is selected (jumper at port pin P0.5 not fitted) the left joystick switch is connected to the transmitter input at port pin P1.0 and corresponds to port pin P0.1 at the receiver. This output is connected to MOSFET T1. When the switch at the joystick is activated the motor speed will be ramped up to a maximum in about one second. This is accomplished by the controller switching the MOSFET with a PWM signal where the on/off ratio of the waveform gets longer until the MOSFET is switched fully on.

If the ‘speed regulator’ option is selected (jumper at port pin P0.5 fitted) then the up-down axis of the left joystick will become the motor speed controller. Pulling the stick downward from its neutral position will cause the motor speed to increase.

The gate control voltage is 5 V so a logic level MOSFET is used. To ensure maximum power in the drive motor the MOSFET R_{DSON} should be <0.001 Ω. The SUP75N03 meets this criterion and can switch 15 A continuous current.

The gate inputs of all the MOSFETS are tied to ground by 100 kΩ resistors to ensure that the transistors do not become conducting during microcontroller initialisation when the port output pins are tri-stated. Schottky-diode D2 protects the MOSFET from large voltage spikes produced by the motor.

FET T4 is used to stop the motor suddenly when it is switched off. This brake function is necessary for power assisted model gliders to make the special propeller blades fold back at the end of the climb thereby reducing drag.

Switched outputs

The output port pins P1.0 and P1.1 have an on/off toggle action when push buttons S1 and S2 are pressed at the transmitter. Each press of a push button will cause the corresponding MOSFET T3 or T2 to change state. The BUZ11 MOSFETs specified here have an R_{DSON} of 0.04 Ω and will have no problems handling 5 A continuous current. If the MOSFETs are used to switch inductive loads like relays then it is important to add protection diodes across the load to prevent destruction of the MOSFETs. These two MOSFETs can simply be omitted if you do not want this function and need to save weight.

BEC

The receiver includes a BEC or Battery Elimination Circuit. This allows the receiver and servos to be powered from the same battery pack that provides power to the electric motor. The voltage of this battery pack will be typically 7.2 V or greater. A low-drop regulator type LM2940 (IC3) is used to supply the 5 V necessary for the receiver and servos. In some applications the BEC is not required in which case IC3 can be omitted and a +5 V supply is connected to point P1 on the PCB.

The power dissipated in IC3 can be calculated from:

$$P = (V_{batt} - V_{BEC}) \times I_{BEC}$$

V_{batt} will be the voltage of the battery pack while V_{BEC} is 5 V and I_{BEC} is the current to the receiver and servos. If this power dissipation is greater than that recommended for IC3 then it is necessary to fit a heat sink to IC3. A heat sink is certainly recommended if you are using high performance servos with this receiver. Lastly, for the BEC to function correctly the battery voltage must be greater than 6 V.

The main features of the LM2940 are:

– 1 V maximum drop across the regulator
– 1 A maximum output current @ 25 °C
– Reverse voltage protection
– 26 V maximum input voltage

Voltage monitor

Just like the transmitter, the receiver also monitors its supply voltage. When it dips to below 4.5 V LED D1 will begin flashing. In normal operation the LED will be lit permanently to indicate correct operation of the software. The potential divider chain formed by R2 and R3 is calculated identically as it was in the transmitter but this time with a threshold voltage of 4.5 V. When a low voltage condition occurs it is stored in volatile memory. This ensures that even if the battery recovers slightly after working at full load it still gives a correct indication that the battery requires recharging.

(010205-1)

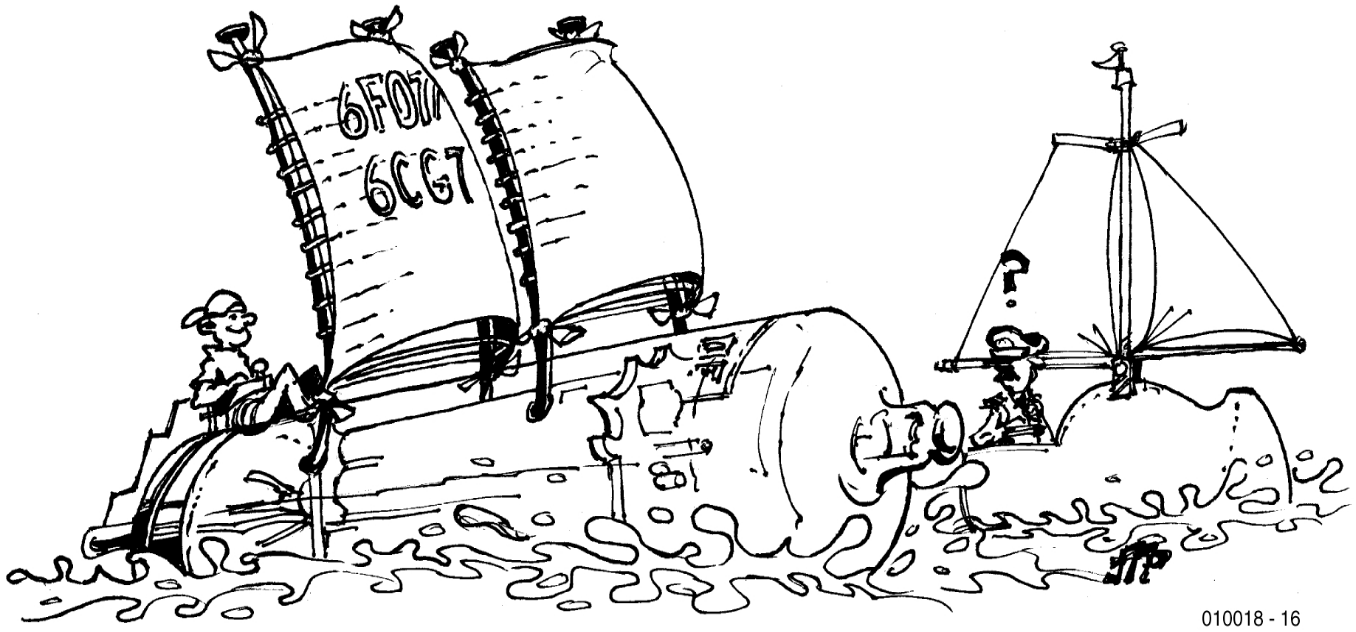
In the second part of this article we will look at the assembly of the units, and take a closer look at the transmitter and receiver software.

Tube Clipper

audio processor — with valves!

By Dr. H. Friedli

The Tube Clipper is an effects unit which can be inserted in the audio path of a stereo system. It provides that 'valve sound', but without the use of high voltages.



010018 - 16

There are many audiophiles who like 'valve sound' but who are put off buying a valve amplifier by the prohibitive cost and who are put off building their own by the dangerous voltages involved.

The Tube Clipper described here is a kind of effect/preamplifier which gives the valve sound but which uses low-voltage valves and which needs no voltage higher than ±12 V. It is an ideal introduction to valve technology.

The Tube Clipper has the following distinctive features:

- Limit setting for both stereo channels via a single potentiometer;
- No coupling capacitors are required on the output;
- Symmetry is individually adjustable for each valve;
- No voltages higher than ±12 V are required;
- Gives that authentic 'valve sound';
- Visible indication of when the input signal is being clipped.

Principles

It is of course an idea abhorrent to any audio freak that an audio signal might not be copied faithfully from sound source to loudspeaker. So what is a sound processor like this doing in a hi-fi system? Why is valve technology so popular at the moment? The reason can be found in the studio: CDs are in general not particularly carefully recorded and mixed, and voltage peaks in a recording are routinely clipped at the expense of signal-to-noise ratio. You will probably remember the discussion (initiated in *Elektor Electronics*) of overdriven CDs, the reasons they are produced and their undesirable consequences. We will not, however, continue that theme here.

The idea behind the Tube Clipper is to limit the audio signal at a given preset maximum level. Valves do this in a rather gentle way: they do not limit excessive voltages as ‘hard’ as transistors do when driven into saturation, but rather as the voltage rises towards the limit value, the valve begins to clip gradually. Hence we obtain the desired smooth curve.

When a signal clipped in this way is analysed, we find that only harmonics at integer multiples of the fundamental frequency have been introduced. To a musician it is now obvious: when the harmonics of a sound are mixed in this way, no dissonance can arise. In contrast the non-integer multiples that are introduced by transistor clipping have a very adverse effect on the sound. The Tube Clipper does modify the sound of a CD, but in such a way that no sounds disturbing to the ear are introduced. Indeed, the added integer harmonics fill out the sound considerably to make it very similar to the original. Ask a guitarist why he swears by his valve amplifier! Overdriving helps considerably to improve the sound of the electric guitar.

The circuit described here can be inserted directly in a high-impedance audio chain. The Tube Clipper then comes into its own when the original sound already exhibits some undesired clipping.

The circuit stage by stage

The circuit in **Figure 1** shows the components that make up the Tube Clipper. At the heart of each of the two stereo channels is an ECC83 double triode and a quad operational amplifier type TL074. A further TL074 is used to control the grid voltage and the LED indicators. The audio signal goes first to impedance converter IC1.A. The buffered output signal is taken to the grid of the first triode B1.A. The triode is connected in cathode-follower configuration with the anode being connected directly to the +12 V supply without a series resistor. The voltage on the 33 kΩ cathode resistor is in phase with the input voltage.

The part of the circuit containing the potentiometer and IC3.D forms a buffered voltage source that offsets the cathode DC

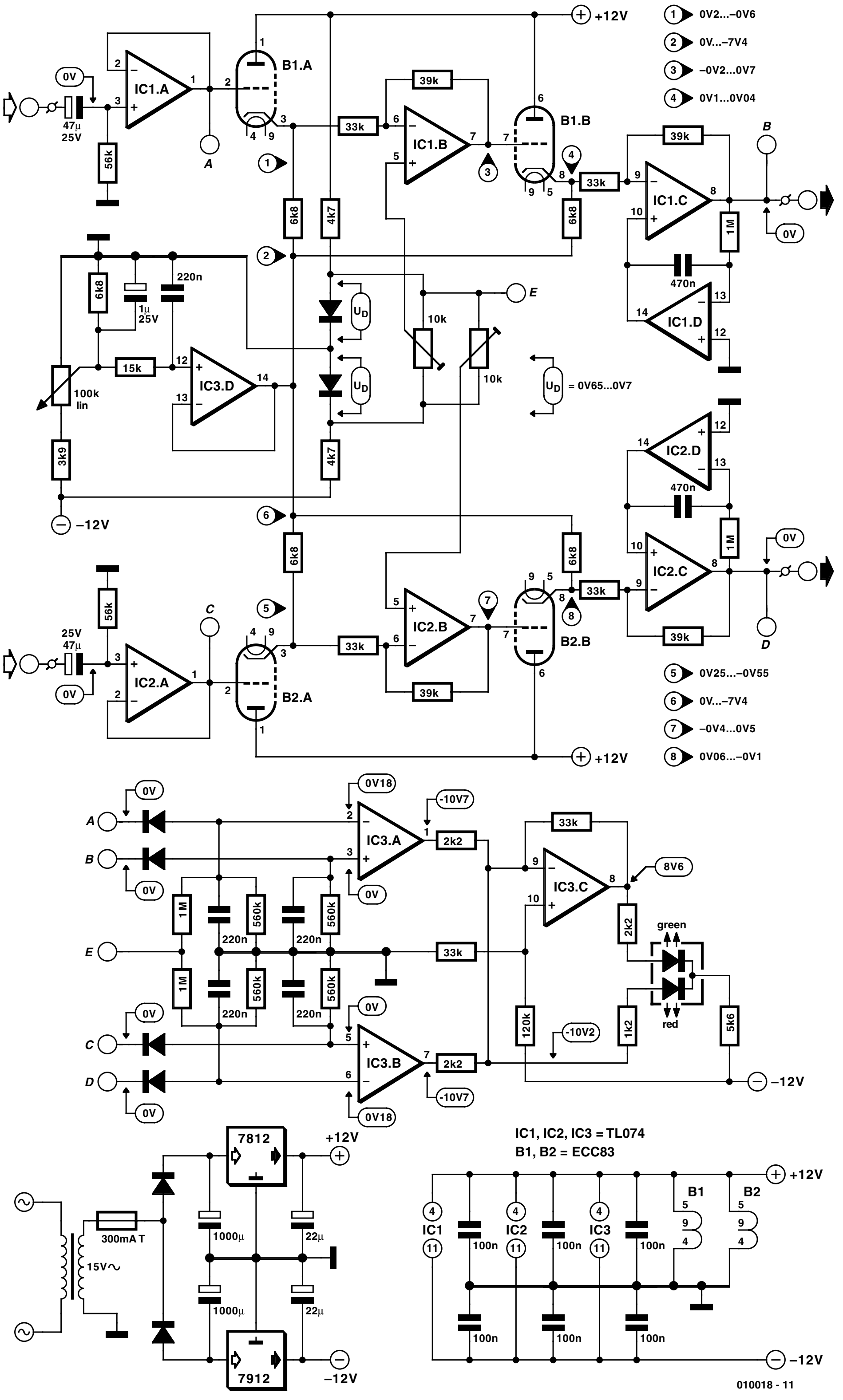


Figure 1. Circuit of the Tube Clipper with low-voltage valves.

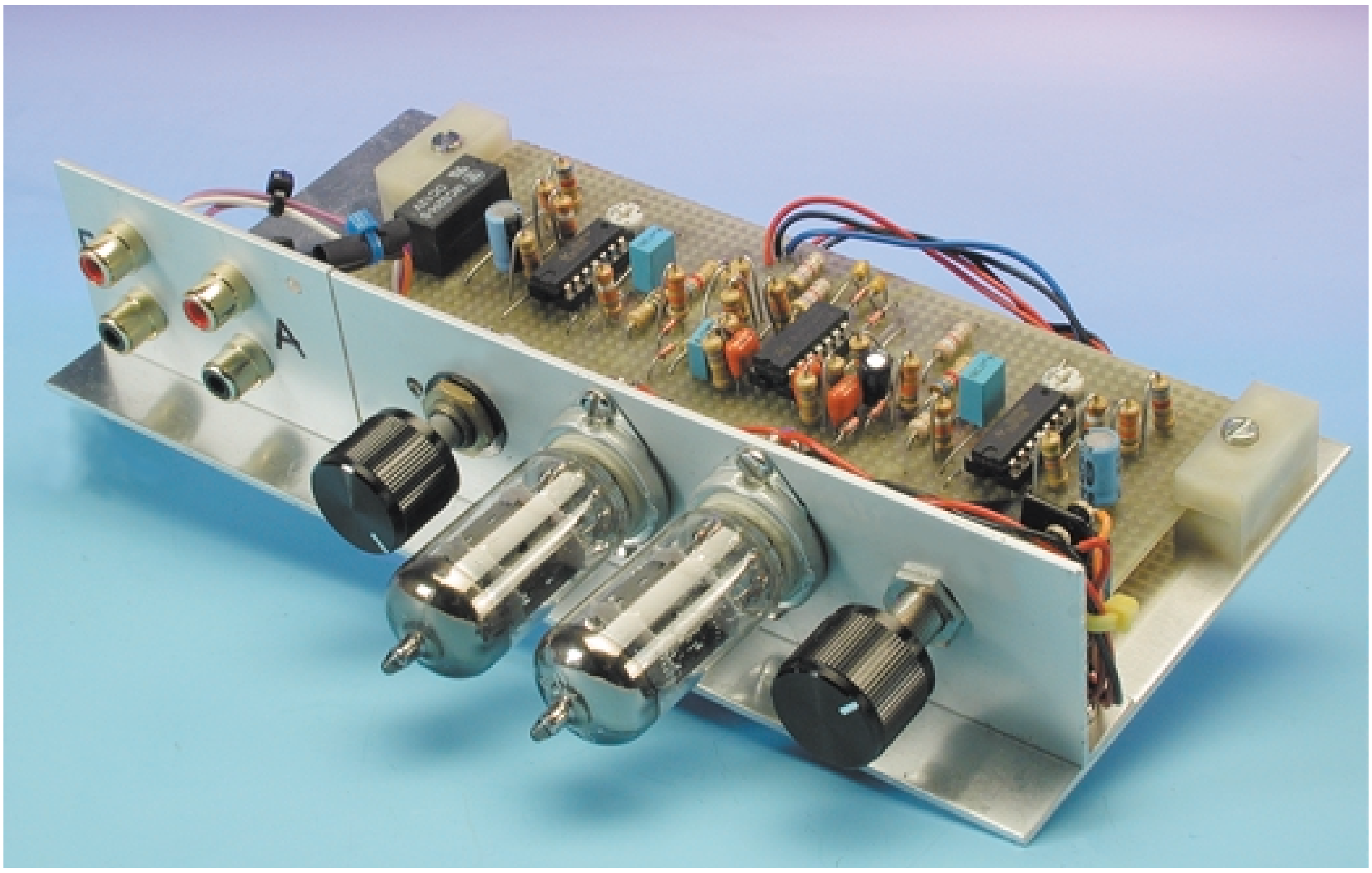
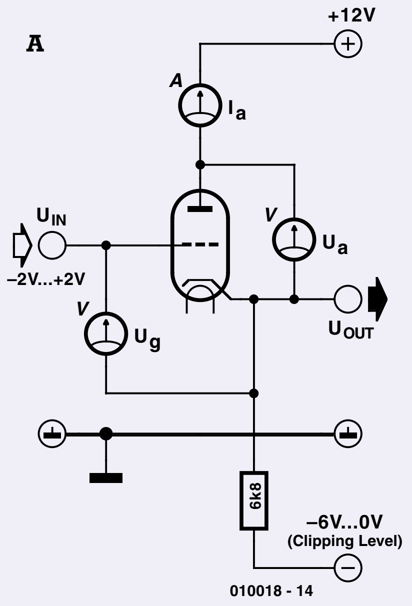


Figure 2. Example construction

voltage negatively with respect to the grid so that the triode can be overdriven to a greater or lesser degree. The first triode, then, clips only on positive-going peaks in the signal. IC1.B simply inverts the signal and the second triode clips the negative peaks of the input signal in the same way. The ratio between the two resistors, 39 kΩ and 33 kΩ,

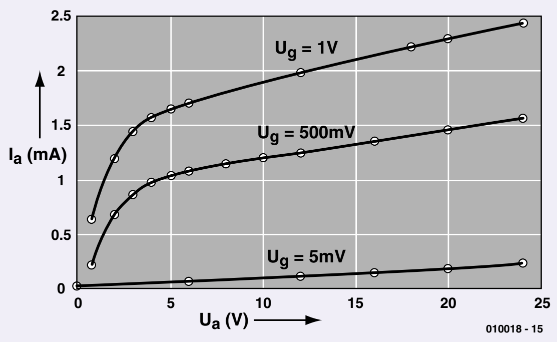
Valve characteristics

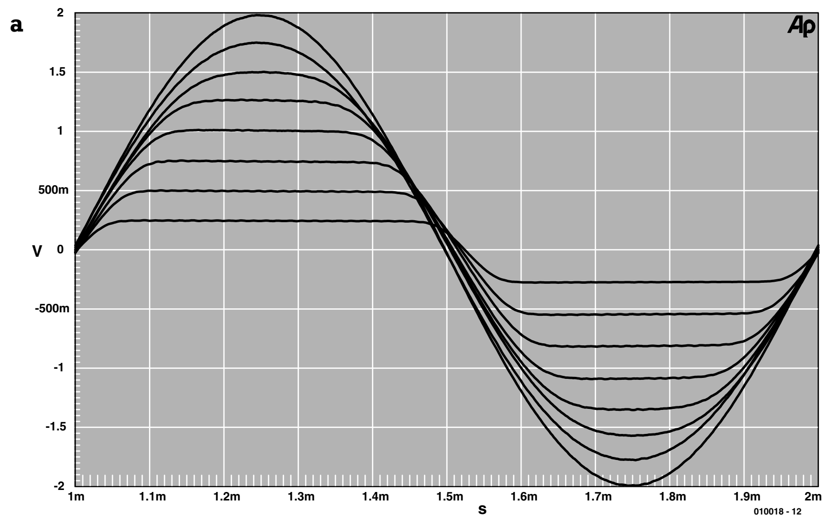
A



It is possible to get to grips with the triode valve using only simple equipment. Take three multimeters and two voltage sources and connect the triode as shown in Figure A. Measure the anode current as a function of grid voltage at constant anode voltage, and obtain curves as shown in Figure B (clipping level at 0 V). The three sets of values measured at different grid voltages exhibit the typical characteristic curve of the valve: as the anode voltage is gradually raised the anode current rises, linearly at first, and then the curve bends —gradually, in contrast to the transistor — into a flatter region as the anode current reaches its maximum level. If all the grid voltages corresponding to the audio signal are in the linear range, the signal will not be distorted; if the grid voltages go higher, then the anode current will saturate. In the circuit presented here this anode current also flows through a resistor, resulting in a voltage proportional to the current.

B





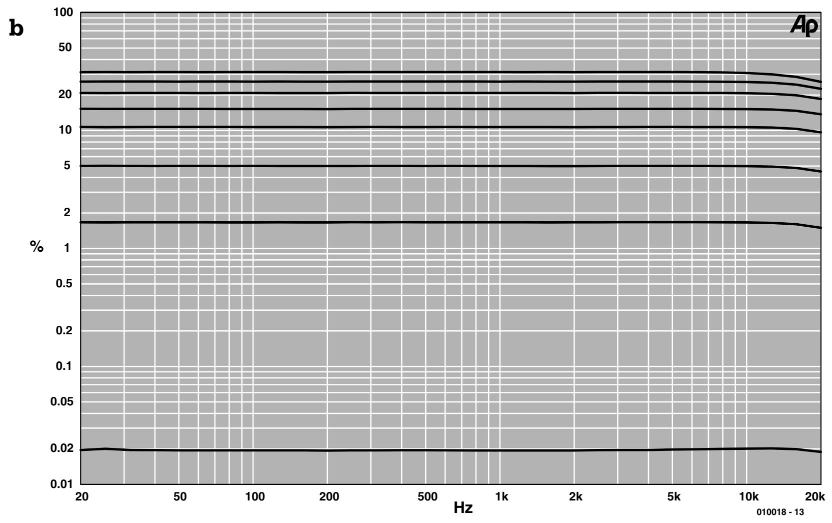


Figure 3. Signal shape at various clipping levels (a), with corresponding distortion curves (b).

results in a small gain that compensates for losses elsewhere in the system. The output signal (if not clipped) will then be at about the same level as the input signal.

IC1.C produces a low-impedance, and slightly amplified output signal for the following stage (which might be a power amplifier). Together with IC1.D it ensures that the output signal has no DC component. IC1.D is connected as an integrator. A DC offset at the output will cause a large voltage to build up on the output of the integrator, with opposite sign. This voltage is fed to the non-inverting input of IC1.C and acts so as to oppose the DC offset. In this way the average output voltage is kept at zero, and an electrolytic capacitor at the output can be avoided. If the source signal is guaranteed free of DC offset, the Tube Clipper can be built entirely without coupling capacitors in the signal path. The second channel is built identically to the first using B2.A and IC2.

Op-amp IC3.D provides a low-impedance voltage source for the cathodes that is adjustable via a potentiometer. Using the two trimmers the output signal can be made symmetrical by compensating for component tolerances. This also allows the ECC81 to be used without difficulty. Input and output signals can be found at points A-D. These voltages are used to drive an indicator, for example using the circuit shown using op-amps IC3.A, B, C and the dual LED.

The LED lights green when the stereo signal is not being clipped; yellow, when one channel is being clipped; and red, when both ECC83s are clipping. The input and output signals are rectified by diodes and smoothed by capacitors. Peak voltages are stored briefly and then decay gradually away. IC3.A/B are comparators whose outputs are positive or negative according to whether the corresponding input voltage is higher than the output voltage or not (i.e., whether there is clipping or not). The comparator outputs are summed by resistors and drive the red part of the LED. The junction of the resistors is also connected to an inverter (IC3.C), which in turn drives the green LED. We therefore get the following behaviour: if there is no clipping, the green led lights. If just one channel is clipping, then the red LED lights also, resulting in a yellowy-orange mixture; if both channels are clipping, the green LED goes out and the dual LED appears red. A small positive voltage on point E ensures that the green LED lights at very low audio input levels, which gives a pleasant effect.

Other variations on the indicator theme can be considered: these are left to the whim of the constructor.

The Tube Clipper circuit can be powered from a simple regulated ±12 V power supply. The valve heaters can also be driven from +12 V DC, since the ECC83 (as well as the ECC81 or ECC88) do not require an AC supply. It is, however, important to ensure that pin 5 is connected to the positive side of the supply.

We have not produced a printed circuit board for the Tube Clipper, since the valves do not have any dangerously high voltages present on them and there are so few other components. **Figure 2** shows an example of how the circuit might be constructed.

After the circuit has been built and the soldered joints checked, the Tube Clipper can be switched on and the various voltages and tolerances shown in the circuit diagram can be checked. Then apply a sinewave signal to the input and set the symmetry of each channel separately using the trimmers. The unit is now ready for use.

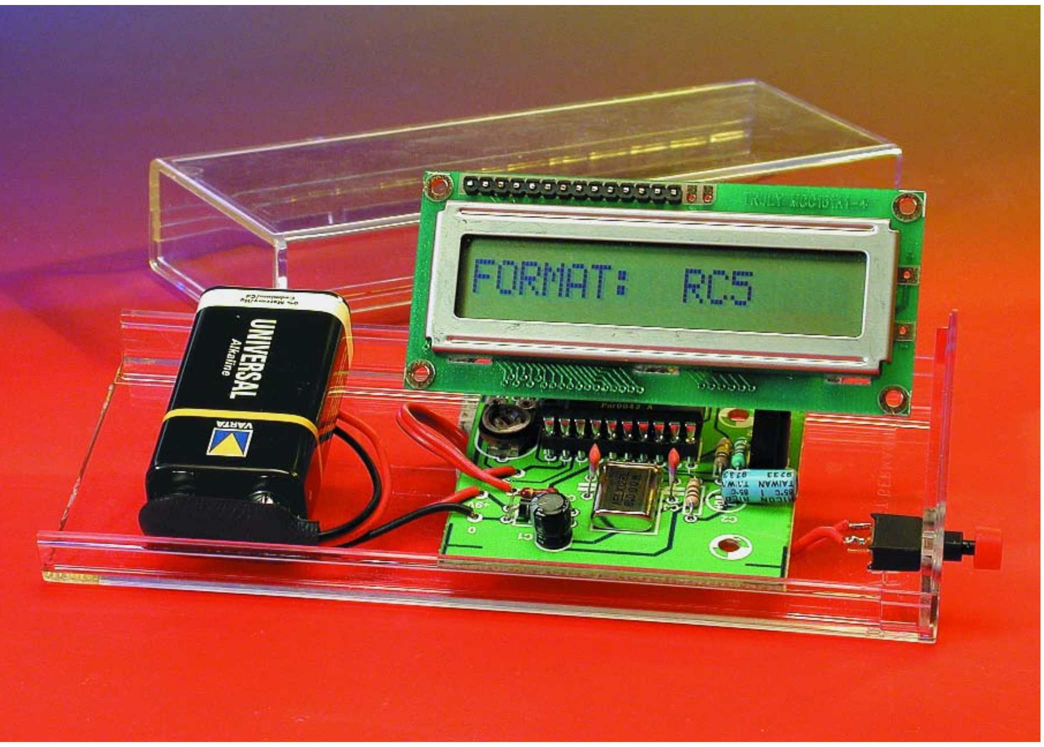
The effect of the Tube Clipper is shown graphically in **Figure 3a**. The effect of the 'level' potentiometer is shown in eight steps from minimum (unclipped curve) to maximum. Finally, **Figure 3b** shows the corresponding distortion curves. When the signal is not clipped, the distortion is at its lowest, around 0.02 %, while, as might be expected, at the highest clipping level the distortion is at its most severe.

(010018-1)

IR Code Analyser

identify remote control codes!

Point any IR remote controller at this nifty device and its in-built microcontroller will quickly analyse the signal and reveal which of eight standard control protocols the controller is using.



Microcontroller

- 4 KByte ROM
- 128 Byte RAM
- 32 Byte EEPROM for user code
- 2.7 to 6 V Supply voltage
- Two 16 Bit Timer/Counter
- Integrated reset
- Internal RC Oscillator (selectable)
- 20 mA drive current from all the port pins
- Maximum 18 I/O-Pins, when the internal reset and RC oscillator is used.
- 2 analogue comparators
- I^{2}C interface
- Full duplex UART
- Serial In-circuit Programmable

It's a rare piece of entertainment equipment that does not come with its own remote controller. After a few years it's easy to accumulate a drawer-full of (functional) controllers belonging to equipment long since consigned to showrooms at your local household amenity tip.

Equipment manufacturers tend to use different controller protocols so it is rare to find by chance a controller from one manufacturer able to control equipment from another. All of the protocols used should however comply with one of the eight industry standards.

Many electronics enthusiasts can probably think of lots of applications for a discarded remote controller. For example using the well-documented RC5 or RECS80 protocol from Philips a simple receiver/decoder can be built using the SAA3009 or SAA3049 IC from Philips if only you knew details of the controller coding. Similarly if you are looking for a second controller for a piece of equipment its not practical to lug a portable oscilloscope to the local car boot sale to analyse each controller on offer.

This device offers the ideal solution. The chief design criteria for the IR analyser was to produce a neat, portable unit that would quickly enable a check to be made of the output signal of an unknown remote controller. A microcontroller decodes the IR message and displays the important information on an LCD. In use the unknown IR controller should be placed close to the analyser's receiver IC before a command key is pressed. The microcontroller will decode the message and display information indicating the

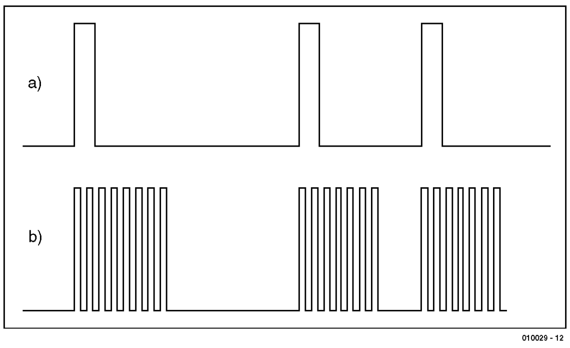


Figure 1. Flash mode signal (a) modulated mode signal at 36 kHz (b).

protocol used and the command sent. A good source of detailed information about the different IR protocol formats can be found in the March and April 2001 edition of *Elektor Electronics*.

This analyser is only suitable for decoding modulated signals in the range of 30 - 40 kHz. Fortunately all the common control protocols use this frequency range.

Some (older) controllers use flash mode where the message information is conveyed by switching on the sender diode for a short period rather than modulating it at 36 kHz (see **Figure 1**). The analyser's IR receiver chip will only respond to modulated signals so that flash mode messages will be ignored.

Operation

Once the analyser is switched on the message 'CODE ANALYSER VER 1' will be displayed and then 'WAITING...' indicating that the analyser is ready and waiting to receive an IR controller message. As soon as a message is received the analyser will display the type of coding sent by the controller and pressing pushbutton S1 will display further information about the received message.

1. Key press: Address
This hexadecimal value indicates the type of equipment that the controller was originally designed to control. As an example, if the controller uses RC5 protocol format and the address 0 is displayed this indicates that controller will control a TV. Some manufacturers do not use the address field in the message so in this case two lines (—) will be shown on the display.

2. Key press: Command
This hexadecimal value indicates the type of command that was sent in the message. A 10_{hex} (16_{dec}) for example in RC5 coding indicates that the command will increase volume of the controlled equipment.

3. Key press: Complete code
The contents of the entire message are displayed in hexadecimal. The hex value assumes that the data is sent in the order Bit 7 to Bit 0. Some manufacturers (e.g. Sony) however reverse the bit order. In this case the values are corrected before display.

4. Key press: Type of Code
This displays the protocol type of the last received message e.g. RC5, SIRCS or RECS80.

When the display shows UNKNOWN this indicates that the received message does not conform to any of the eight standard protocols. It could also indicate however that the signal was too weak or too distorted. The IR receiver IC2 is optimised for reception of a 36 kHz modulated signal so its sensitivity to some signal protocols at the extremes of the 30-40 kHz band will be greatly reduced (see **Figure 2**). Always place the output diode of the handheld controller as close as possible to IC2 of the analyser to ensure a good signal. LED D2 will blink to indicate signal reception irrespective of the type of protocol so it gives a good indication that the controller is sending out a signal. If the diode does not blink try a new set of batteries in the controller. Items at car boot sales rarely have batteries fitted so it's a good idea to carry a few spares with you.

Some of the more modern remote controllers will send out two messages each time a key is pressed. The first message is in one format while the second message contains the same information in a different format. The analyser will only display the last message format sent.

Hardware

There are no surprises in the circuit diagram of the IC code analyser shown in **Figure 3**.

The infra red signal is detected by IC2. Infineon have ceased production of their receiver IC but any one of the compatible devices specified in the parts list would be suitable. This device contains a highly sensitive infra red receiver, amplifier, filter and demodulator tuned to an IR carrier frequency of 36 kHz. A data sheet for this device can be down loaded from:
(www.infineon.com/cmc_upload/0/000/008/562/sfh5110.pdf).
The inverted demodulated received signal is connected directly to an input port of the microcontroller IC1. IC2 is relatively sensitive to supply rail disturbances so R2 and C2 are used to form a low pass filter, decoupling any interference.

The core of the microcontroller is based on the Intel 51 architecture that lends itself to simple low-cost program development by using any of the available Shareware development tools. The 87LPC764 is produced by Philips and is described as a *Low power, low price, low pin count microcontroller.* Decoding

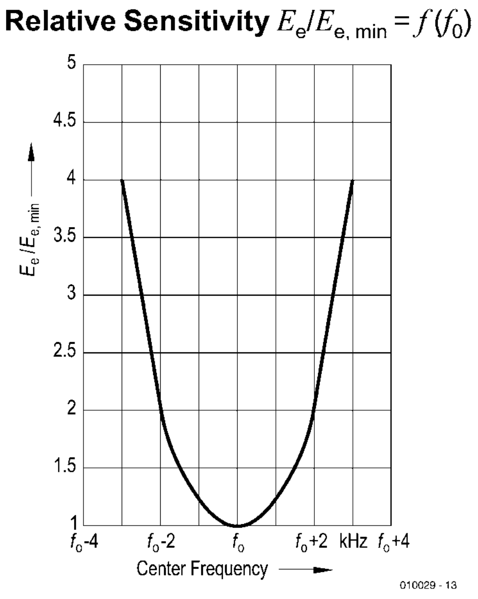


Figure 2. The sensitivity of the IR receiver falls off sharply either side of its centre frequency.

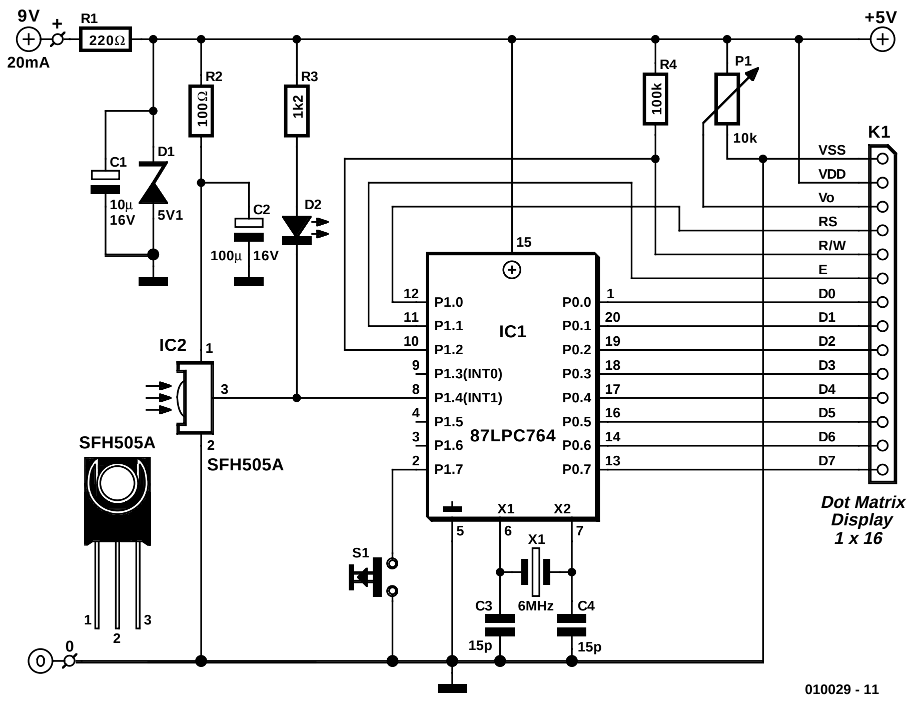
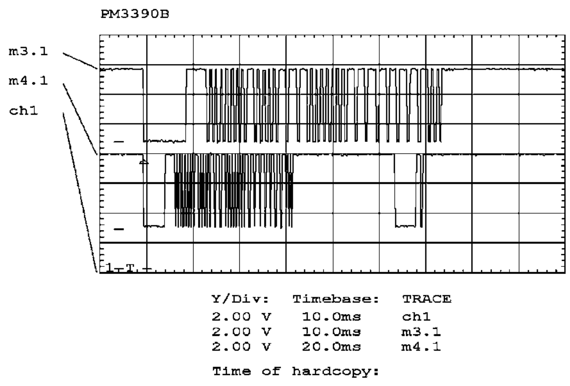


Figure 3. Only two IC's are used in the analyser circuit diagram.

the IR message is relatively processor intensive so a clock rate of 6 MHz is used for the microcontroller giving a machine cycle of 1 µs. The main features of this processor are summarised under the heading 'microcontroller'.



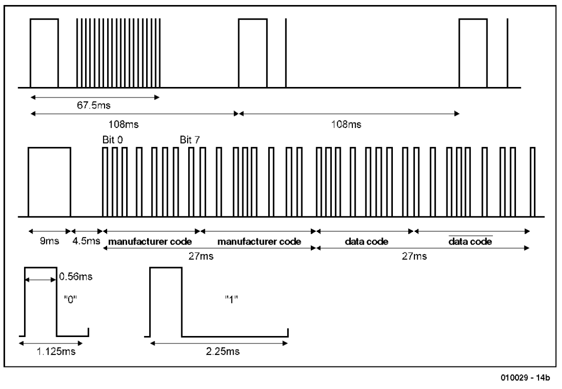


Figure 4. An NEC coded signal (used by Harman/Kardon or Yamaha).

Power for the analyser is provided by a 9 V battery and zener diode D1 regulates this on-board to 5 V. Resistor R1 limits the supply current and overall consumption is approximately 20 mA.

Display

The display is a standard 1x16 LCD (indicating 1 row of 16 characters). This display is a standard item from many manufacturers. It has a built-in controller using a standard command set. All of the compatible displays mentioned in the parts list have identical pin-outs, RAM addresses and multiplex rate. The information generated by this analyser could be displayed on a four line LCD but to reduce costs it was decided to employ a single line display and use a pushbutton to toggle through the information. The display control software is simplified by using eight controller port pins. P1 is a variable resistor, allowing adjustment of the display contrast.

Software

All the relevant data for the following protocols are integrated into the software so that they can be matched to the incoming signal:

JAPANESE NEC RC5 RECS80
SIRCS DENON DAEWOO
MOTOROLA

The software starts its analysis of the incoming signal by measuring the first low-phase. Almost all of the protocol formats have different length start bits so this simplifies the identification process. All of the incoming signal time measurements are made using the microcontrollers timer, this gives a maximum resolution of 1 ms with the 6 MHz clock frequency. After the start bit that usually consists of a low followed by a high of predefined length the microcontroller decides how the incoming data will be stored in the internal registers. The message length will depend on the type of protocol and can be from 11 bits for RCS80 protocol up to 48 bits for Japanese coding. The timing tolerance of the message length is taken into account and the length of each bit is measured and compared to its limit values.

The Software, including source

code is available free of charge from the *Elektor Electronics* website or alternatively can be ordered (at nominal cost) on diskette **010029-11**. If you are planing to modify or customise the software it should be noted that the pulse width of the received IR signals varies quite substantially with received signal strength so the decoding software should be correspondingly tolerant. The following table gives a practical example of pulse tolerances for the signal shown in **Figure** 4 (NEC code from Harman/Kardon or Yamaha):

1. Low phase:	8700 - 9200 ms
1. High phase:	4352 - 4607 ms
Second low phase:	400 - 767ms
Gap between low phases:	
	0-Bit: 400 - 767 ms
	1-Bit: >767 ms

The relatively large tolerances often correspond to the high byte value of the 16 bit timing counter so that the hex calculations can be simplified by ignoring the least significant eight bits of the timing counter.

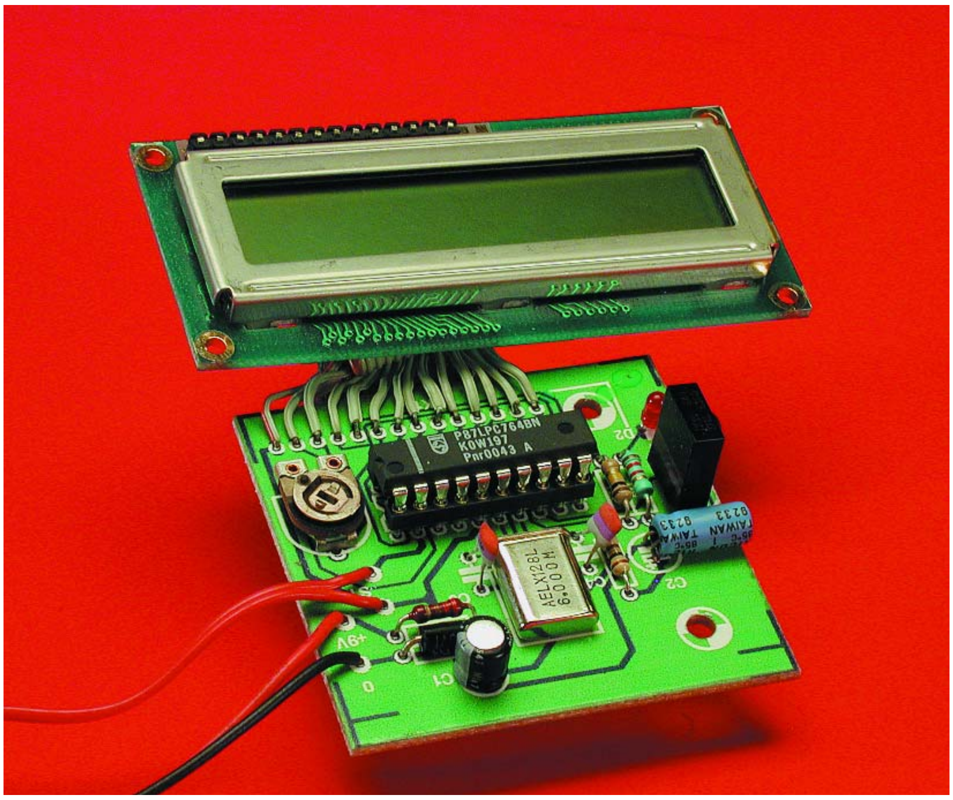
COMPONENTS LIST

Resistors:
R1,R2 = 220Ω
R3 = 1kΩ2
R4 = 100kΩ
P1 = 10kΩ preset H

Capacitors:
C1 = 10μF 16V radial
C2 = 100μF 16V radial
C3,C4 = 15pF

Semiconductors:
D1 = zener diode 5V1, 1 W
D2 = LED, high-efficiency, red
IC1 = 87LPC764 (programmed, order code **010029-41**)
IC2 = SFH505A (TSOP1736, SFH5110-36, PIC26043SM, IS1U60, TFMS5360)

Miscellaneous:
K1 = 14-way flatcable
S1 = push-button, 1 make contact
X1 = 6MHz quartz crystal
Dot matrix display, 1 x 16 characters (MCC161A1-4, MCC161A2-3(Truly), LM161556 (Sharp)
9V battery with clip
Enclosure (Heddic 222)
On/off switch



Construction

The PCB layout is shown in **Figure 5.** The board shown is unfortunately not available ready-made through our Readers Services, so you have to make it yourself. Microcontroller IC1 is fitted to the PCB using a 20 pin DIP socket. The IR receiver (IC2) shown is the Infineon type but other compatible types given in the parts list can be fitted. The compatible types have similar performance but not identical pin-outs so extra mounting pads on the PCB enable these other types to be accommodated (just double-check the pin-outs).

The LCD is connected to the PCB via a short length of ribbon cable. The displays given in the parts list have a 1:1 pin to pin compatibility with the PCB layout. Other types of displays can be used but you will need to study the corresponding data sheet to ensure correct signal connections. Once the circuit has been given a functional test it can be mounted together with the 9 V battery in a suitable case. If a transparent case is used it will only be necessary to drill two holes in the case, one for the on/off switch and one for the mode switch. The display will be visible through the case.

(010029-1)

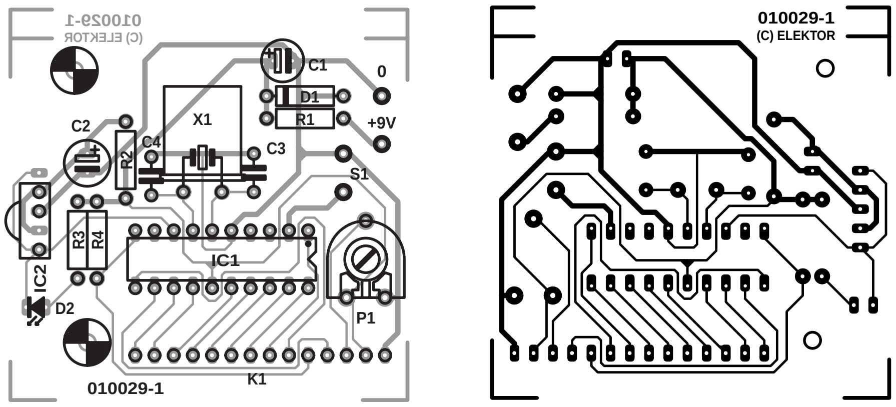


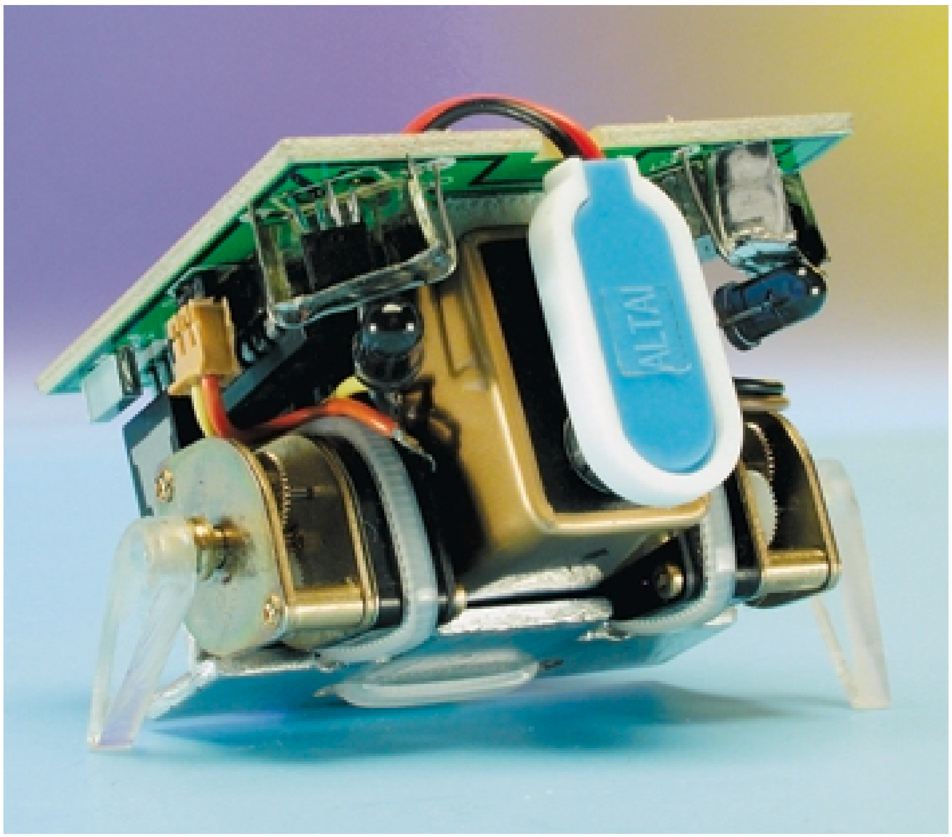
Figure 5. The crystal needs to be mounted flat on the PCB (board not available ready-made).

Wobbler

a robot that doesn’t drive or walk, but ‘wobbles’!

Design by A. Vreugdenhil

Just its unusual method of motion alone makes the mini-robot an eye-catching appearance. It is an extremely simple design, because it consists of hardly anything else besides a processor, two sensors and two little motors. Nevertheless it is possesses a great deal of dexterity when it comes to avoiding obstacles. The moment it ‘spots’ one, it will make an evasive manoeuvre and is off, on its two funny rotating legs.



The most remarkable aspect of this little robot is undoubtedly its unique method of movement. It does not actually have any wheels, but has two cams that function as a kind of paddle, driven by a couple of miniature motors. The dimensions of the cams are such that they protrude a little below the underside so that the creature pushes itself off and ‘wobbles’ forwards every time a cam touches the ground.

Strictly speaking, the unique under-carriage was the motivation towards the ‘birth’ of Wobbler. The idea surfaced at one moment and from there it just became a challenge to find out whether this method of movement would work in practice or not. The other parts were quickly added afterwards: two IR sensors for vision (IC3, IC4), two driver stages for the motors (IC2) and for the ‘brains’ a simple processor (Atmel 89C2051) with a small piece of software.

The finished shape measures about 5 ? 5 ? 3.5 cm and the schematic of **Figure 1** shows how little is involved from the electronic perspective. An additional advantage of this simple design is that Wobbler is quite affordable; apart from the printed circuit board the builder is unlikely to spend more than about thirty pounds for all the parts.

Behaviour

The intentions for Wobbler were to make it behave in such a manner that it is able to freely move about its living environment and be able to avoid obstacles. Its living environment consists of a marked out terrain of arbitrary size, surrounded by an edge of about 5 cm high.

Our mini-robot does not react to light or dark; its aspirations amount to not much more than to casually stroll ('wobble') over its terrain, all the while keeping an eye on any fixed or moving obstacles. Whenever an object appears within its approximately 10 cm sized field of view, it will take action to avoid it.

Two infrared detection sensors type IS471F (IC3, IC4) function as 'eyes'. This is not an ordinary light sensitive detector, because this clever little IC contains a modulator, demodulator, oscillator and a voltage regulator. The sensor transmits an infrared pulse train via an external infrared LED (D1, D2) and subsequently looks if a nearby object has reflected this pulse train. When this is the case, the output of the sensor will go 'low'.

The significant advantage of this LED/sensor combination is that at reacts only to the transmitted pulse train and is therefore not affected by ambient light. Problems may arise however, when two Wobblers face each other unexpectedly. But even this could be solved with a small change to the software, if necessary.

Because Wobble has to be able to move forwards and backwards, it is necessary for the motors (M1, M2) to rotate in both the clockwise and anticlockwise direction. This requires a reversing circuit for the DC motors in the form of an H-bridge. This could be built from discrete parts, but nowadays ready-made building blocks are also available. The L293D IC used here contains two H-bridges complete with built-in freewheeling diodes; take note of the suffix 'D', because the regular L293 does not contain these diodes. M1 and M2 are two miniature DC motors with built-in reduction gears. The operating voltage is 3-6 V and the speed is 22-44 rpm. Further details can be found in the parts list. Other motors with comparable specifications may also be used.

For the central processing unit an 80C2051 from Atmel (IC1) was selected. The flash-EPROM contained within provides ample room for a simple program. The processor has the additional advantages that it is readily available and inexpensive to program. And also, a free BASIC-compiler is available that can generate code up to 1 kbyte in size. This compiler goes by the name 'BASCOM-LT' and the programmer is the 'BlowIT'. You can obtain more information on the web site www.mcselec.com from the company MCS Electronics in Zaandam, the Netherlands, who developed the compiler.

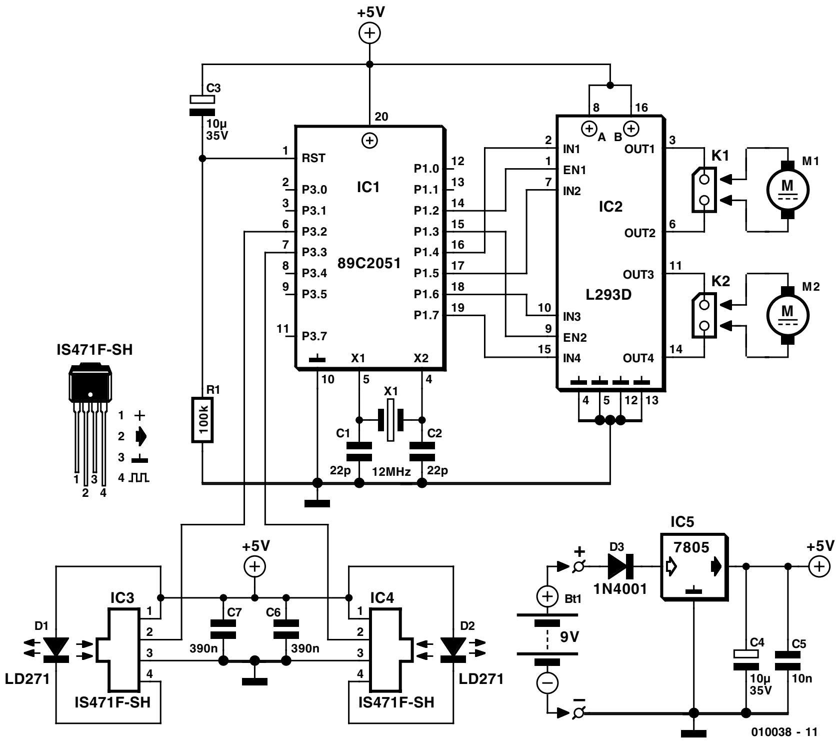


Figure 1. The electronics consists of little more than a processor, a motor drive and two sensors.

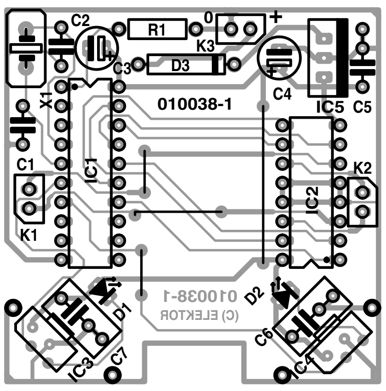
Construction

The necessary electronics for the robot are so modest that the printed circuit board (**Figure 2**) can be made pleasingly compact. As can be seen, room has been made available for a 9-V-battery in the centre of the PCB. Small SIL connectors are placed on the PCB to make the connections to the battery and the two motors. In order to increase the directivity of the sensors, the prototype was fitted with makeshift blinkers, consisting of a few pieces of tin soldered together (if the sensor continuously sees the stray light from the LED it will give a false alarm). **Figure 3** shows the completely built up PCB.

The introductory photograph of the article illustrates that the prototype was constructed in just about the simplest way imaginable. The mini-motors are attached with cable ties and the chassis is a short length of T-profile aluminium that was made to fit with a little filing and drilling. The battery also acts as the supporting element for the circuit board; they are joined with two small pieces of hook and loop fastener (Velcro).

The 'legs' or 'paddles' or 'cams' or whatever best describes them are made from perspex (a.k.a. Plexiglas). They may be fastened to the motor shaft with a small drop of glue, if a suitable hole is drilled in the underside first. The length of the paddles has to be determined experimentally and is quite critical. If they are too long, then Wobble will tilt too far forward when moving; it this point it will sense the ground as an obstacle and 'back paddle' continuously. This is boring and definitely not the intention. If the legs are too short, then it will react as it should, but will have insufficient ability to push off and as a result move very slowly. The length of the paddles in our prototype measured 18 mm from the centre of the shaft to the end.

All that said, the mechanical construction is so simple that an evening of tinkering should suffice to get the robot to wobble.



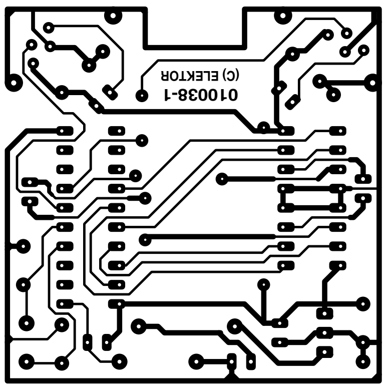


Figure 2. Printed circuit board and component overlay for Wobbler's electronics.

Although it is governed by only a few simple rules, the program is nonetheless put together in such a way that it appears that Wobbler is able to cleverly deal with every situation in its environment.

The program can easily be changed of you would like to modify its behaviour. In addition to the programmed controller a diskette with the program and source code is available from *Elektor Electronics* Readers Services (refer to parts list). The software is not available as a free download from our website because of contractual agreements with the author.

Because the hardware is extremely simple, relatively small changes to the software will obtain good results. To make Wobbler appear a little more 'intelligent', it would be possible to have it make a small turn after some random time. Enthusiastic tweakers will undoubtedly think of many more variations after experimenting a little.

(010038-1)

COMPONENTS LIST

Resistors:
R1 = 100kΩ

Capacitors:
C1,C2 = 22pF
C3,C4 = 10µF 35V
C5 = 10nF
C6,C7 = 390nF

Semiconductors:
D1,D2 = LD271 or general-purpose IRED
D3 = 1N4001
IC1 = 89C2051 (programmed, order code **010038-41**)
IC2 = L293D
IC3,IC4 = IS471F-SH
IC5 = 7805

Miscellaneous:
X1 = 12MHz quartz crystal
Bt1 = 9V battery (PP3)
M1,M2 = mini motor with reduction gear, 3-6 V, 22-44 rpm (Conrad Electronics order code 242543-50)
K1,K2 = 2-way SIL header with plug
PCB, order code **010038-1** (see Readers Services page or Elektor website)
Disk, project software and source-code files, order code **010038-11**

Software

The control program written for Wobble is stored in IC1 and is only 312 bytes in size. It is both simple and logical by design.

– After switching on, the robot is instructed to move forwards for about 5 seconds. If one of the motors initially turns backwards then the corresponding connector has to be reversed.

– Subsequently, the sensors are checked. If one of the sensors observes an obstacle, then Wobbler will first go backwards for 5 seconds, followed by an evasive turn for 3 seconds.

– Following this, it moves forward once more and checks the sensors all over again. While moving backwards the sensors are ignored, because it does not make much sense to do otherwise.

– When **both** sensors simultaneously detect an obstacle, if for example, it faces a wall after an evasive manoeuvre, then it will move backwards for 3 seconds and turn about its axis for 4 seconds.

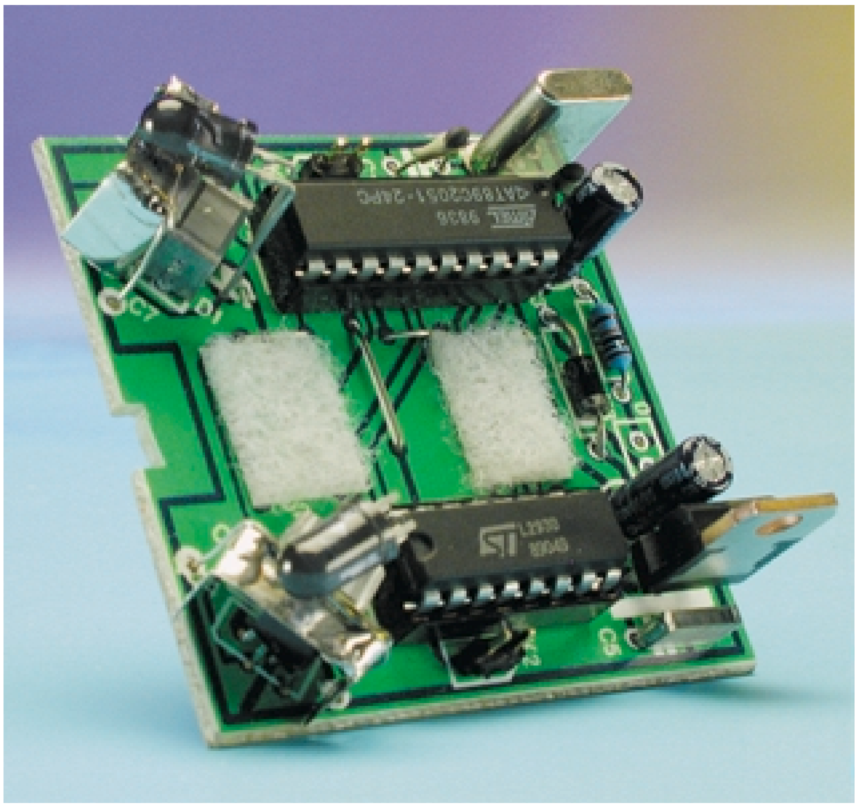


Figure 3. The construction of the PCB is not a major undertaking.

CAD in Europe (2)

a review of schematic drawing and PCB layout programs

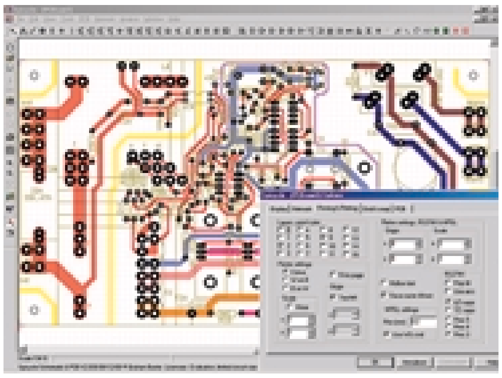
By Guy Raedersdorf

In the first instalment of this article (published in the June 2001 issue) we discussed three electronics CAD software packages supplied to us as complete versions. This month we conclude our survey with programs that came either as a demo, evaluation or time-limited version.

Correction to Part 1 (June 2001)
On page 56, the Internet address of Quick-route should be changed to www.dotqr.com.

Spicycle™

Source: Those Engineers Ltd.
Version tested: Evaluation SpiceAge V5.174
Spicycle V2.030
Medium: CD-ROM
Language: English
Installation: okay
De-installation: perfect
Documentation: small manual, 60 pages
www.spiceage.com



To be able to work with Spicycle™ you first need to install SpiceAge.

Spicycle™ is is an electronics design tool comprising circuit diagram drawing (schematic capture), PCB design and simulation (the latter is outside the scope of this article).

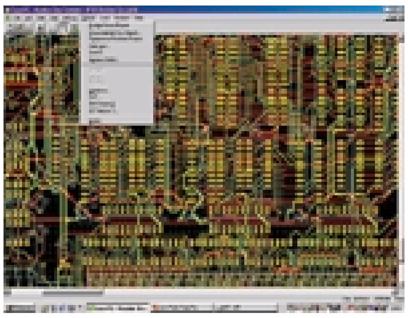
Some notable features of Spicycle™:
- no dongle used
- scalable hard copy output for laser printers and high-resolution inkjet printers
- multiple documents for quasi-simultaneous work on several schematics and PCBs.
- max. schematic size 82 × 82 cm (32 × 32 in.)
- easily modifiable circuit symbols
- triple design rule checking (DRC), independent for PCB areas.

Getting started with Spicycle™ is rather less evident than with other programs discussed in this article. Quite some time is needed to get acquainted with the program and be able to exploit all the possibilities offered by the various utilities it consists of.

In conclusion, Spicycle™, as opposed to certain other programs discussed here, is a universal program suite, whose feature *par excellence* seems to be the simulation tool, to which we should hasten to add that schematic capture and PCB design components are extremely attractive as well.

Easy-PC for Windows

Source: Number One Systems
Version tested: V 4.0 Demo
Medium: CD-ROM
Installation: no problems
De-installation:
Documentation: 2 pages plus leaflet; 225-page manual supplied with full version
www.numberone.com



Although only a demo version was supplied to us, Easy-PC for Windows fits the bill for the framework set up for this article because it is ‘no more’ than a combination of a schematic capture and a PCB design tool, the latter assisted by a component library editor).

Easy-PC for Windows comes in four flavours, from a version limited to 500 pins (*sic*), through 1,000 and 2,000 pins, right up to a version with an unlimited number of pins (your designs, please?). An autorouter function is available in two versions that come as program extensions; they are called Easy-Router II 500 and Easy-Router II Unlimited. The numbers refer to the number of component pins (or leads), and do not include vias, if used.

Easy-PC for Windows as we received it contains a perfect drawing tool for circuit diagrams — see the illustrating screendump. The function allows you to work quasi-simultaneously on several documents. The maximum size of the circuit is 1 × 1 m (39 × 39 in.).

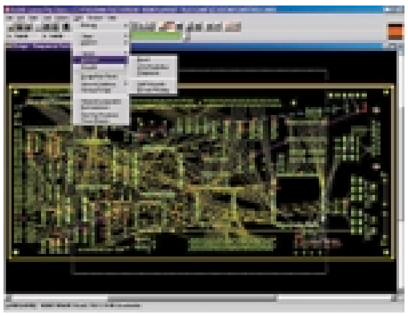
The schematic capture tool and the PCB design tool are linked seamlessly — combining the two does not even require any netlisting. It is, however, possible to import netlists from OrCAD EDIF and EWB/MultiSIM.

As is evident from the screendump produced by the PCB design tool, the screen layout is beyond reproach.

Easy-PC for Windows is undemanding as far as available PC hardware is concerned, the program running happily, for example, on an 800 x 600 SVGA screen.

OrCAD Capture & OrCAD Layout

Source: OrCAD Inc.
Versions tested: OrCAD Capture V 9.10.93 CIS; OrCAD Layout: Plus V 9.10 Demo
Medium: CD-ROM
Installation: easy
De-installation: deletion of a number of .dll files to be confirmed manually.
www.orcad.com



OrCAD Capture may be considered as a kind of archetype of schematic capture programs, its first version having been released nearly a quarter of a century ago. The *Elektor Electronics* in-house laboratory staff have been using this program for nearly two decades, despite many offers for competing products.

The file 'capture-layout.pdf' contains 63 pages that are well worth printing to get a good idea of to go about working with the schematic drawing program. Although the screendumps in the manual are slightly blurred (deliberate to keep the pdf file reasonably small?), you should find the program features and processes explained in sufficient detail to get going. The technical documentation presented in colour wit the CD-ROM is beyond reproach. Capture CIS is a functionally extended version of Capture.

The version of Capture as found on the CD-ROM is two years old already but still up to present-day demands.

As you can see on the screendump, OrCAD Capture CIS (Component Information System) is a very 'clear' program, perfectly structured, easily mastered and comfortable in everyday use. The moment a component is placed on to the worksheet and connected up to the rest of the circuit, conflicting and otherwise doubtful points are immediately identified and visualised. The various options available to the user have become increasingly sophisticated.

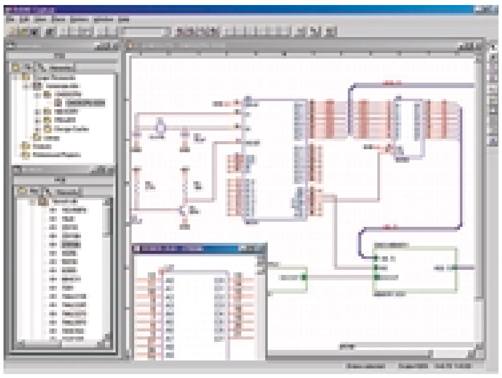
OrCAD Capture CIS comes with an Internet Component Assistant^{TM} that gives access to databanks on the Internet.

The PCB design program, OrCAD Layout, is available in three variants. The simplest is OrCAD Layout Engineers Edition, the most powerful, OrCAD Layout Plus, with OrCAD Layout positioned in the middle.

The appearance of the task list buttons is vaguely reminiscent of the old DOS days, mainly because of the reduced size of the texts. In general, however, the button functions are clearly legible, and cosmetics should be less of an issue than functionality.

The design examples in the 'Samples' folder are well presented and should set a standard for all PCB design programs.

The autorouter function supplied as part of the demo version of OrCAD is impressive. You can watch the program 'knitting' the tracks thanks to over 10 Mbytes of RAM it has in use, auto-routing the 747 interconnections on the board of which the screendump gives an impression. All this does take some

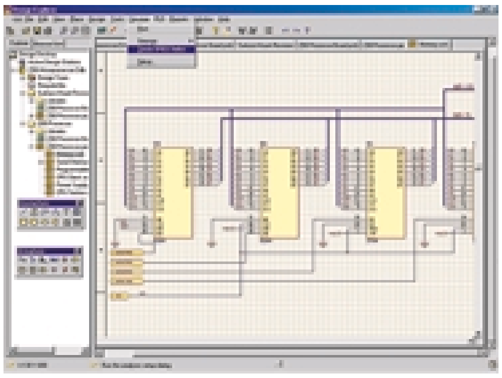


time, if indeed it is possible to run a completely automatic autorouting operation on such a complex circuit. You can watch the active window jump from the centre to positions along the edge of the screen.

However, OrCAD is not just a schematic capture program combined with a PCB designer — simulation is also catered for by a family member called Pspice.

Protel 99 SE

Source: Protel
Version tested: Trial, 30 days
Medium: CD-ROM
Installation: perfect
De-installation: perfect
www.protel.com



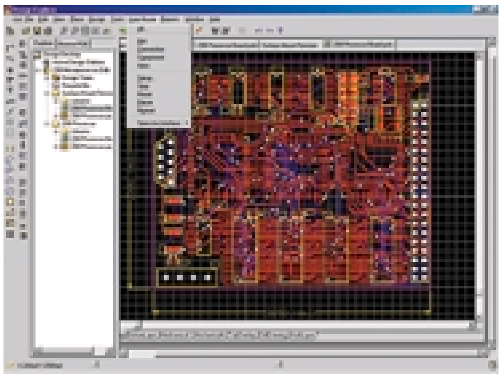
Together with OrCAD, Protel is a major player in the highest league. A number or you may remember the programs Advanced Schematic, Advanced PCB and Advanced Route, which were among the first to run under Windows and offering a level of graphics sophistication second to none at that time. Their 32-bit versions are now found back in the Protel 99 SE shell. Protel 99 SE is a complete PCB design suite.

The evaluation version of 99 SE is supplied with a colour 32-page technical manual which we found very instructive and extremely well laid out, covering everything from drawing the circuit diagram to producing PCB artwork.

The CD-ROM itself comes with a 40-page

booklet containing a step-by-step description of the way the programs are best used.

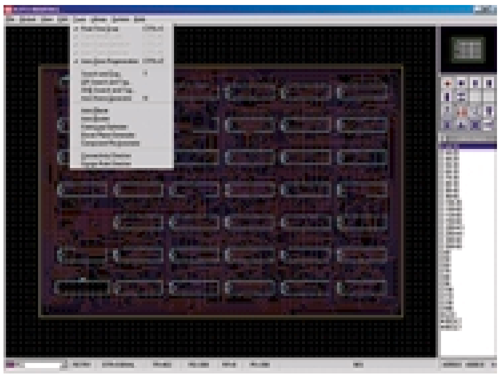
Every licensed copy of Protel 99 SE comes with a free copy of CAMtastic!® 2000 Designer edition, a program for PCB artwork editing.



Without doubt, Protel is one of the best combinations of a schematic capture, a PCB design tool and an autorouter. However, its price tag will keep it securely in the professional domain. With OrCAD, Protel is a kind of Rolls Royce in electronic CAD land.

Proteus VSM

Source: LabCenter
Version tested:
Medium: CD-ROM
Installation: okay
De-installation: perfect
www.labcenter.co.uk

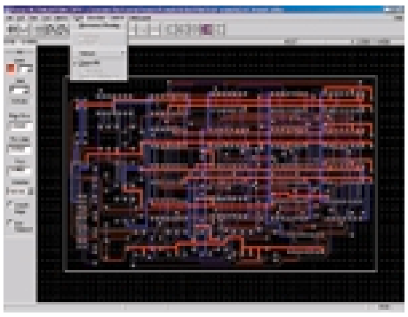


Proteus consists of two software components:
ISIS 5.2
For schematic drawing and
ARES 5.2 Professional
for printed circuit board design.
The demonstration version of ARES presents a number of slick examples of autoplacement and autorouting. An empty workspace on the screen is filled with components and their interconnections. By a single click on the Auto Router function, you can see the tracks appear. An impressive 'spectacle' to watch is the Rip-up & Retry function 'knitting' on the screen in an attempt to accommodate all tracks on the PCB shown in the screendump. Manual routing on a PCB of such complexity is out of the question, even for a highly skilled PCB designer.

ISIS 5.2, the circuit drawing companion, has a lively appearance. This program suite also comprises a simulation utility.

Ranger XL

Source: Seetrax
Version tested: Ranger XL 1.57
Medium: Internet
Installation: perfect
De-installation: perfect
Note: requires a license via e-mail
www.seetrax.com



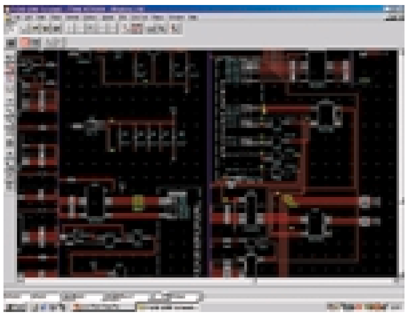
Ranger XL is a suite of programs that enables a PCB design to be drawn accurately, starting from a circuit diagram (preferred) or a netlist combined with a parts list, imported as files or entered manually.

Having opened a project file, you should not be surprised to see that nothing happens on the screen. To make the circuit of your choice appear on the screen, you need to click on the button identified by a couple of logic gates. You will search in vain for handles to move the circuit.

Even if the program has a relatively simple structure, it is capable of rapidly producing a circuit diagram. In fact, it is possible to create PCB artwork starting from a circuit diagram.

p-cad® 2000

Source: Protel
Version tested: p-cad 2000 Trial Version
Medium: CD-ROM
Installation: perfect
De-installation: perfect



www.protel.com

p-cad 2000 is the new product line from Protel. In fact, we're talking about a renamed version (15.02) of ACCEL EDA. The basic version comprises all necessary modules: circuit diagram drawing, PCB design and autorouting plus project maintenance.

There exists a Lite version of p-cad 2000, called 6L/400C (6 layers, 400 components).

The evaluation version of p-cad 2000 comes with a 'main features' brochure containing impressive illustrations.

The p-cad 2000 CD-ROM is supplied with a cute 48-page colour manual which is extremely useful when learning to use the program, which is actually a collection of many software utilities. The new version has been enhanced with updated functions, meaning that it has become more intelligent at the level of the auxiliary functions. In particular, the enhancements make the autorouter easier to get to grips with.

To give you an idea of what may be expected from today's CAD programs, here are some of the main features of p-cad:
- up to 30 layers
- up to 4,000 components
- up to 5,000 pins per component
- up to 10,000 netlist lines
- up to 16,000 connections

p-cad 2000 employs a sophisticated autorouting technique based on case shapes. In this way, the program is capable of optimising component placement on the board area available for autorouting.

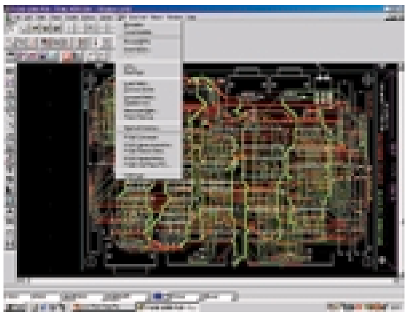
Each licensed copy of the full p-cad 2000 package includes a copy of CAMtastic® 2000 Designer Edition which was already mentioned above.

p-cad 2000 is a high-end product

aimed at users in the electronics industry.

CircuitMaker 2000

Source: Protel
Version tested: Student V6.2c
Medium: Internet
Installation: no problems
De-installation:
www.circuitmaker.com



CircuitMaker is a nicely made schematic drawing program complete with an attractive circuit simulator.

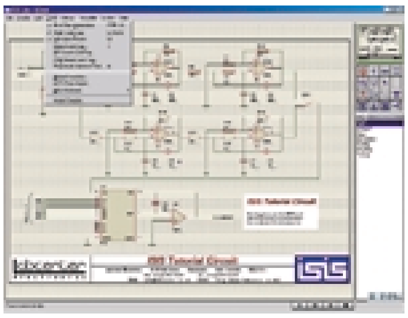
You can watch an alarm circuit in operation and a few other small designs with a clearly educational intention.

The PCB artwork program is actually 'Traxmaker'.

The Internet version of this program, cm20000trialversion, requires a password for installation. For the purpose of this article, however, we had to stick to the version accessible to us: Student V6.2c (without Traxmaker).

CAD PACK

Source: Matrix Multimedia
Version tested: ARES IV Lite 4.73 Level 0, ISIS Lite 4.73 Level 0
Medium: CD-ROM
Installation: okay
De-installation: all files cleared
www.matrixmultimedia.co.uk
We received a Multimedia CD-ROM containing, among others, a Lite version of ARES PCB design software (v4.73, Level 0) and ISIS schematic capture (also v4.73, Level 0). ISIS was already discussed above as part of the Proteus VSM package.



The program versions are well-tried if not a bit rusty considering that Proteus is currently at version 5.2 None the less, the CAD PACK multimedia CD-ROM is of considerable interest because it presents a great introduction to an impressive number of programs. The CD-ROM makes reference to PROSPICE Lite simulation software which we were able to find (after much searching) disguised as an audio device control box in the right-hand bottom corner of the screen.

On the CD-ROM we found, among others, a CadPack directory, and a second directory named demoCD containing no fewer than seven subdirectories called demo, downloads, dworks, etc. which are well worth looking into.

The problem with this type of 'covers-all' CD-ROMs is that the content is quickly outdated. On the positive side, they do allow you see and try out a lot.

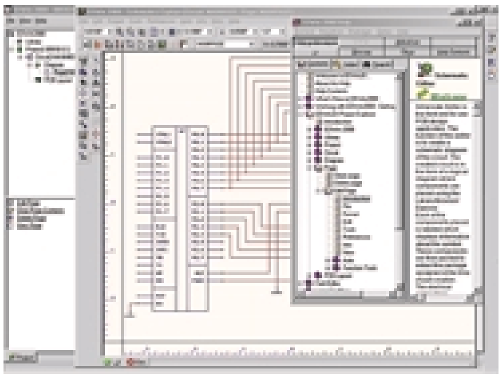
VUTRAX

Source: Computamation Systems Ltd.
Version tested: VUTRAX 12.2c
Medium: Internet
Installation: unclear what's happening but okay in the end
De-installation: partial; one subdirectory to clear manually
www.vutrax.co.uk

Vutrax is a program with an installation best described as surprising. Once installed, the use of the program is highly intuitive. The evaluation version allows you to design circuits with up to 275 pins. Vutrax offers a simulation function, either Pspice Analogue or SpiceAge, and sports a 3D viewing function. The full version of Vutrax allows circuits and PCB artwork of great complexity to be designed.

EDWin

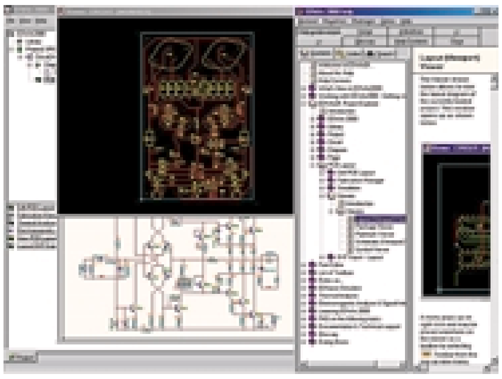
Source: Mercure-Telecom
Version tested: EDWIN 2000 Ver 1.0



Medium: CD-ROM
Source: Norlinvest Ltd.
Version tested: V 1.71 Demo
Format: CD-ROM
Installation: okay, takes a few minutes
De-installation: okay; one subdirectory with a couple of files in it to be cleared manually.
www.switfteurotech.co.uk

During the installation, a number of older files have to be skipped.

The CD-ROM with the EDWIN 2000 version requires a user license number to run. This number may be found on the CD-ROM itself. All you have to do is replace the A:\ by a D:\ (or any other letter identifying the CD-ROM drive in your PC). Installation takes a while but that is not surprising considering that EDWin 2000 also contains a Spice simulator, EDSpice. Nice presentation, never a dull moment.



Edwin 2000 is a complete and complex program, resulting in a rather steep learning curve. Fortunately, a context-sensitive help menu is available. As shown in the two screenshots, the help menu is accessible at each stage of drawing a circuit diagram or PCB artwork.

(000148-2)

Note: prices of CAD programs and CAD program suites discussed in this article may be obtained from the manufacturers or suppliers.

Personal Mini Webserver (2)

Part 2: the first steps

There are so many different configurations of the SC12 that we only have space here to cover the basic installation and operation of the Personal Mini Webserver.

During programming and remote operation of the Personal Mini Webserver, it is always helpful to have the Internet site http://www.bcl-online.de/documentation/api/index.htm on hand (**Figure 1**). This contains a complete, up-to-date index of the software documentation for the SC12 chip. For more in-depth information on any of the procedures described below, this documentation can always be consulted.

Before the Personal Mini Webserver can be programmed for its final application, a few questions need to be answered:

❮ Is the circuit completely functional?

❮ How will the basic settings be made?

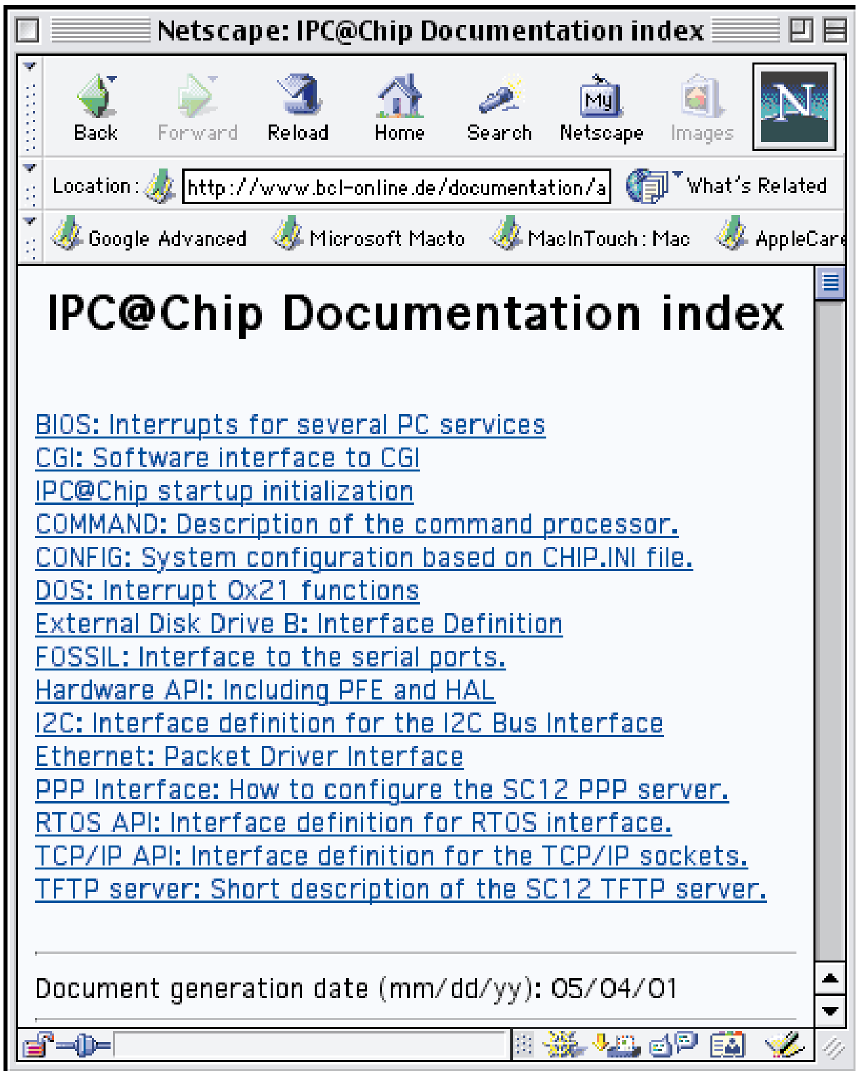
❮ How will files be transferred to the Personal Mini Webserver?

❮ How will the Personal Mini Webserver be provided with a network address?

Functional test

Once assembly of the Personal Mini Webserver hardware has been successfully completed, a full functional test should be carried out. First connect the Personal Mini Webserver to a PC via connector K4 and a **null modem** cable, and start a terminal program. This could be HyperTerminal, provided with Windows, although in principle any terminal software will work fine, as long as the communications parameters are set as follows:

19,200 bits/s
8 data bits
No parity
1 stop bit



IPC@Chip Documentation index

BIOS: Interrupts for several PC services
CGI: Software interface to CGI
IPC@Chip startup initialization
COMMAND: Description of the command processor.
CONFIG: System configuration based on CHIP.INI file.
DOS: Interrupt 0x21 functions
External Disk Drive B: Interface Definition
FOSSIL: Interface to the serial ports.
Hardware API: Including PFE and HAL
I2C: Interface definition for the I2C Bus Interface
Ethernet: Packet Driver Interface
PPP Interface: How to configure the SC12 PPP server.
RTOS API: Interface definition for RTOS interface.
TCP/IP API: Interface definition for the TCP/IP sockets.
TFTP server: Short description of the SC12 TFTP server.

Document generation date (mm/dd/yy): 05/04/01

Figure 1. Index of all documentation of the Application Program Interfaces of the IPC@Chip.

Table 1.

```
@CHIP-RTOS
Copyright (c) Beck 1999
SC12
Serialnumber=012BE
Bootstrap version 02.06
Bios version V1.00
  LARGE_PPP
Build: May  4 2001
Starting TCPIP
Init network
Open ethernet packet
  interface
PPP client
Config server
FTP server
Web server
Telnet server
Ext. disk interface.
No handshake
```

As usual, when the IPC@CHIP is switched on, the LED should flash three times: one long flash and then two short. The first messages from the webserver should appear on the monitor (Table 1).

Press the ENTER key, and in the terminal window you should see the familiar line:

```
A:\
```

This looks just like a DOS prompt; and indeed it is. In fact this is a DOS clone called RTOS. Our readers of more advanced years, who grew up with DOS, will have no difficulty in getting to grips with RTOS commands such as **copy**, **md**, **cd**, **rd**, **del**, **format** and so on. The command **help** causes RTOS to produce a list of all the available commands, which are described in full, along with all parameters, in '*COMMAND: Description of the command processor*' at www.bcl-online.de/documentation/ api/command.htm.

Simple I/O test

Next we check that the I/O ports of the SC12 are functioning correctly. The signals on connector K6 correspond to I/O address 0x100, while those on K7 correspond to address 0x101. On both connectors, the outputs are on the even-numbered pins, and the inputs are on the odd-numbered pins. The outputs can readily be tested by connecting each pin in turn to +5 V via an LED (and 1 kΩ series current-limiting resistor).

In order to activate the ALE and CS signals of the SC12, the following two commands can be entered at the prompt:

```
PCS 1
ALE 1
```

The outputs can now be switched on and off using the **OB** (OutByte) command. This command requires addresses and data in hexadecimal format. So, for example, in order to write the decimal value 170 (or AAh) to connector K6, the following command is required:

```
A:\OB 100 AA
```

which elicits the confirmation:

```
OB address=0x0100 data=0xAA
```

The **IB** (InByte) command behaves similarly. The command

```
A:\IB 100
```

produces the result

```
IB 0x0100=0x00,
```

if all inputs are low.

Initialisation

Signals such as ALE and PCS, which are required for LCDs, digital I/Os or other expansion ports, can be activated immediately on system start-up, either in an *autoexec.bat* file, or in the file *chip.ini*.

In the first case, we simply require the two lines

Example CHIP.INI file

```
[STDIO]
STDIN = EXT COM
STDOUT = EXT COM

[IP]
ADDRESS=192.168.100.101
NETMASK=255.255.255.0
GATEWAY=192.168.100.254

[DEVICE]
NAME=WEBIO

[TELNET]
ENABLE=0

[UDPCFG]
LEVEL=0x00

[WEB]
ENABLE=1
MAINPAGE=MAIN.HTM
DRIVE=0
ROOTDIR=\WEB\

[FTP]
ENABLE=1
USER0=FTP
PASSWORD0=FTP

[TELNET]
ENABLE=1
USER0=TEL
PASSWORD0=TEL
```

Figure 2. A chip initialisation file such as this contains the basic settings of the Personal Mini Webserver.

```
PCS 1
ALE 1
```

in autoexec.bat. If autoexec.bat is not present, for example after a BIOS update, a suitable file can be created using a text editor and sent to the SC12 using the XModem protocol with the terminal program.

Chip.ini can be compared with the config.sys file in a DOS system. This text file contains all the settings which the SC12 requires at start-up. At system start, *autoexec.bat* is the file processed first. The most significant difference between the SC12's operating system and 'normal' DOS is that DOS is at root a single-tasking operating system, while the SC12 always operates by default in multitasking mode. For that reason the prompt always returns very quickly after a program is started, even if the program has not exited. If this behaviour is not desired, the following line can be added to the start of the autoexec.bat file:

```
BATCHMODE 1
```

After this command all programs are executed sequentially. Conversely, the command

```
BATCHMODE 0
```

allows for a semi-parallel execution of programs.

Batch mode can also be configured in chip.ini, with the two lines:

```
[BATCH]
BATCHMODE = 1
```

Figure 2 shows a more complete example of a chip initialisation file.

Local data transfer over the COM port

The SC12 chip boasts a flash disk which is known to DOS as drive A:. There are two methods available for transferring data between the Personal Mini Webserver and the PC, either over the network connection or via the COM interface.
The XTRANS command is used to transfer files over the serial COM interface. The communication uses the well-known XModem protocol, which is available as standard on most terminal programs. For example, to transfer the file *test.txt* from the PC to the Personal Mini Webserver, the command

```
XTRANS COM R TEST.TXT
```

is given at the RTOS prompt. The character R (receive) after the port identifier COM (K4), indicates the direction of data transfer. Next (and without too much delay) the file must be sent from the terminal program using the XModem protocol.

For transfer in the opposite direction, from Personal Mini Webserver to PC, the command

```
XTRANS COM S TEST.TXT
```

is used. The second serial interface, K3, has identifier EXT. Normally (except when a BIOS update is being carried out) this is where the modem is connected in a PPP server configuration. The configuration and use of this interface will be described in more detail later.

Network settings

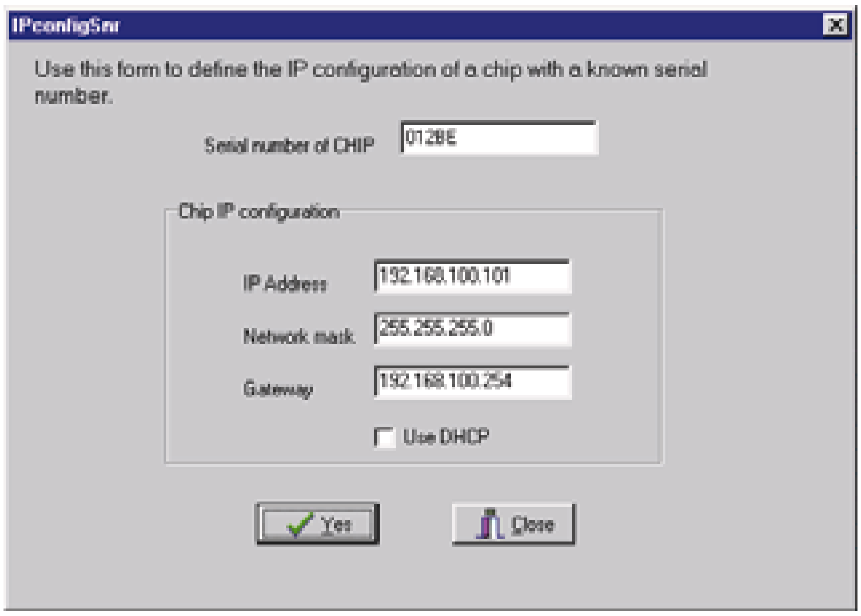
The most important aspect of the SC12 is the network connection. For our first experiments it is easiest to use Ethernet, since it is fast and easy to use. To construct the network, all that is required is a crossover twisted-pair cable with RJ45 connectors (or two cables and a hub) and an Ethernet card in the PC. The Ethernet driver in the SC12 is only compatible with the 'slow' 10Base-T Ethernet standard, and so in principle a 10Base-T card will be adequate. If, however, you are buying a new Ethernet card, you might as well purchase a faster 100Base-T version, since they only cost about thirty pounds.

We will consider here an existing network with the following IP settings:

```
IP Adress:
192.168.100.100
Subnet Mask:
255.255.255.0
```

Table 2.

`A:\DHCP 0`	Disable automatic network configuration
`A:\IP 192.168.100.101`	Set IP address...,
`A:\Netmask 255.255.255.0`	... net mask...
`A:\Gateway 192.168.100.254`	... and gateway address
`A:\IPETH`	Activate new settings


IPconfigSnr

Use this form to define the IP configuration of a chip with a known serial number.

Serial number of CHIP: 012BE

Chip IP configuration

IP Address: 192.168.100.101

Network mask: 255.255.255.0

Gateway: 192.168.100.254

Use DHCP: [ ]

Figure 3. Configuration of the IP address using Chiptool.

`Gateway:`
`192.168.100.254`

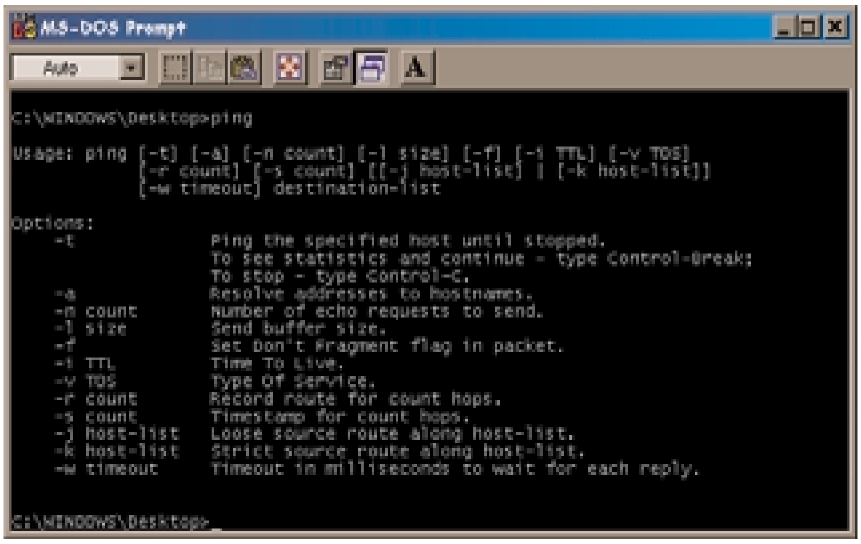
The TCP/IP configuration of the Ethernet card can be found in the System Settings under Network. The next address after the quoted 'IP Address' (i.e. 192.168.100.101) cannot already be assigned to a device on the network.

Before the Personal Mini Webserver can communicate over the Ethernet, a number of addresses must be set. These are its own network address, the address of the network to which the Personal Mini Webserver is connected, and the address of the gateway (if present). The gateway allows devices on different networks to communicate with one another.

First we enter the settings manually at the RTOS prompt, as shown in Table 2.

To confirm the settings, reply to the message

```
Reconfigure IP ethernet -
ok
```



```
C:\WINDOWS\Desktop>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] destination-list

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet.
    -i TTL         Time To Live.
    -v TOS         Type Of Service.
    -r count       Record route for count hops.
    -s count       Timestamp for count hops.
    -j host-list   Loose source route along host-list.
    -k host-list   Strict source route along host-list.
    -w timeout     Timeout in milliseconds to wait for each reply.

C:\WINDOWS\Desktop>_
```

Figure 4. Ping on the PC.

with the command

```
A:\IPCFG
```

and obtain the values:

```
IP=192.168.100.101
Netmask=255.255.255.0
Gateway=192.168.100.254
DHCP=0
MAC= 00 30 56 F0 12 BE
```

The last line gives the hardware address of the internal Ethernet controller and cannot be changed. These settings can also be applied using Chiptool (**Figure 3**) or, if preferred, as commands on `autoexec.bat` or entries in `chip.ini`.

Once the Personal Mini Webserver is connected to a network, the connection can be tested using PING, which is part of RTOS. It is very easy to use:

```
A:\PING 192.168.100.100
```

This causes PING to send data from the SC12 to the specified network address (in this case to the PC). The same can be done from the PC end, using the PING command, which, although a DOS program, is provided as part of Windows (**Figure 4**). Here, of course, the network address of the Personal Mini Webserver, and not of the PC, must be specified. If this all goes well, this indicates that communication has been successfully established.

The settings can, as we said above, also be entered in `chip.ini`, for example as follows:

```
[IP]
ADDRESS=192.168.100.170
NETMASK=255.255.255.0
GATEWAY=192.168.100.254
DHCP=0
```

FTP server

Several protocols are built into the SC12 to enable easy handling of network traffic.

Files can be transferred not only using the XTRANS command, but also, as might be expected for a webserver, over the TCP/IP network using the File Transfer Protocol (FTP). All that is required is an FTP client, such as is available in any Windows system and which can be started using the FTP command in a DOS window. This command-line oriented program is, however, not exactly user friendly. If you are planning to use FTP extensively, a graphics-oriented program such as the universally-used WS-FTP pro-

```
   char tmpbuffer[180];

   /***********************************/
   //building the page
   /***********************************/

   //insert the head of the page
   strcpy(HtmlPage,PageHead);


   //append the given request arguments

   sprintf(tmpbuffer,"<BR>Method     : %02d<BR>",CgiRequest->fHttpRequest);
   strcat(HtmlPage,tmpbuffer);

   sprintf(tmpbuffer,"<BR>Url        :    %s<BR>",CgiRequest->fPathPtr);
   strcat(HtmlPage,tmpbuffer);

   sprintf(tmpbuffer,"<BR>Host       :   %s<BR>",CgiRequest->fHostPtr);
   strcat(HtmlPage,tmpbuffer);


   sprintf(tmpbuffer,"<BR>Referer    :
%s<BR>",CgiRequest->fRefererPtr);
   strcat(HtmlPage,tmpbuffer);

   sprintf(tmpbuffer,"<BR>User Agent :
%s<BR>",CgiRequest->fAgentPtr);
   strcat(HtmlPage,tmpbuffer);

   sprintf(tmpbuffer,"<BR>Content    :
%s<BR>",CgiRequest->fLanguagePtr);
   strcat(HtmlPage,tmpbuffer);

   sprintf(tmpbuffer,"<BR>User       :
%s<BR>",CgiRequest->fUserNamePtr);
   strcat(HtmlPage,tmpbuffer);

   sprintf(tmpbuffer,"<BR>Password   :
%s<BR>",CgiRequest->fPasswordPtr);
   strcat(HtmlPage,tmpbuffer);



   if(strlen(CgiRequest->fArgumentBufferPtr))
   {
   sprintf(tmpbuffer,"<BR>Arguments:
%s<BR>",CgiRequest->fArgumentBufferPtr);
   }
   else
   {
   strcpy(tmpbuffer,"<BR>No arguments<BR>");
   }
   strcat(HtmlPage,tmpbuffer);


   //append the predefined tail
   strcat(HtmlPage,PageEnd);
```

```
    /***********************************/
    //give it to the webserver
    /***********************************/

    CgiRequest->fHttpResponse         = CgiHttpOk;       //HTTP 200 OK
    CgiRequest->fDataType             = CGIDataTypeHtml; //text/html


    CgiRequest->fResponseBufferPtr    = HtmlPage;
    CgiRequest->fResponseBufferLength = strlen(HtmlPage);


}
/***************************************************************************
*/
```

Figure 5. Example software fragment.

gram can be used. This program is available from www.ipswitch.com/Products/index.html.
A user name and password are required to connect to an FTP server. These settings can only be placed in `chip.ini`. Two different identifiers and passwords are allowed. The default settings are:

USER: ANONYMOUS
NO PASSWORD

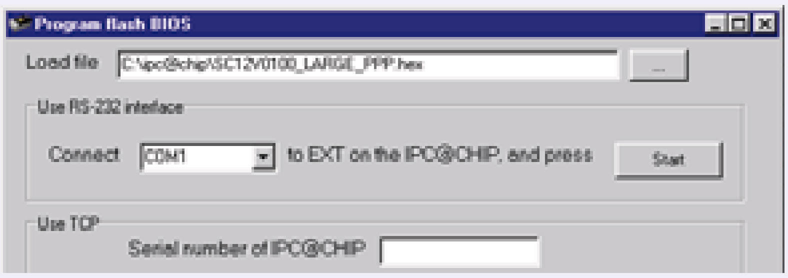
and

USER: FTP
PASSWORD: FTP

An FTP program also allows you, for example, to modify `chip.ini` or `autoexec.bat` remotely, or do upload new programs or Internet pages.

BIOS upgrade

The most important information in the start-up message is the BIOS version. Our software requires beta version V1.00 or later. The most up-to-date version is available for download from Beck's website, which can most conveniently be transferred to the SC12 using the Chiptool program, also downloadable from their site. The transfer can be done either locally via the PC's COM interface, or remotely over TCP, as can be seen from the Chiptool window. When upgrading the BIOS, note that the serial interface used is the SC12's EXT connection, not the COM connection!



Load file C:\ipc@chip\SC12V0100_LARGE_PPP.hex
Use RS-232 interface
Connect COM1 to EXT on the IPC@CHIP, and press
Use TCP
Serial number of IPC@CHIP

Telnet server

In order to operate a device remotely it is not sufficient merely to be able to transfer files. It is also important to be able to run programs or restart devices. TCP/IP offers the Telnet protocol for these applications. The protocol allows you to operate a computer over a network as if you had direct access to its monitor and keyboard.

A Telnet program can be used to establish a connection with the Personal Mini Webserver. On the screen, the appearance is similar to that of a terminal program communicating over the serial port. The standard settings for Telnet, which can be modified in `chip.ini`, are as follows:

USER: TEL
PASSWORD: TEL

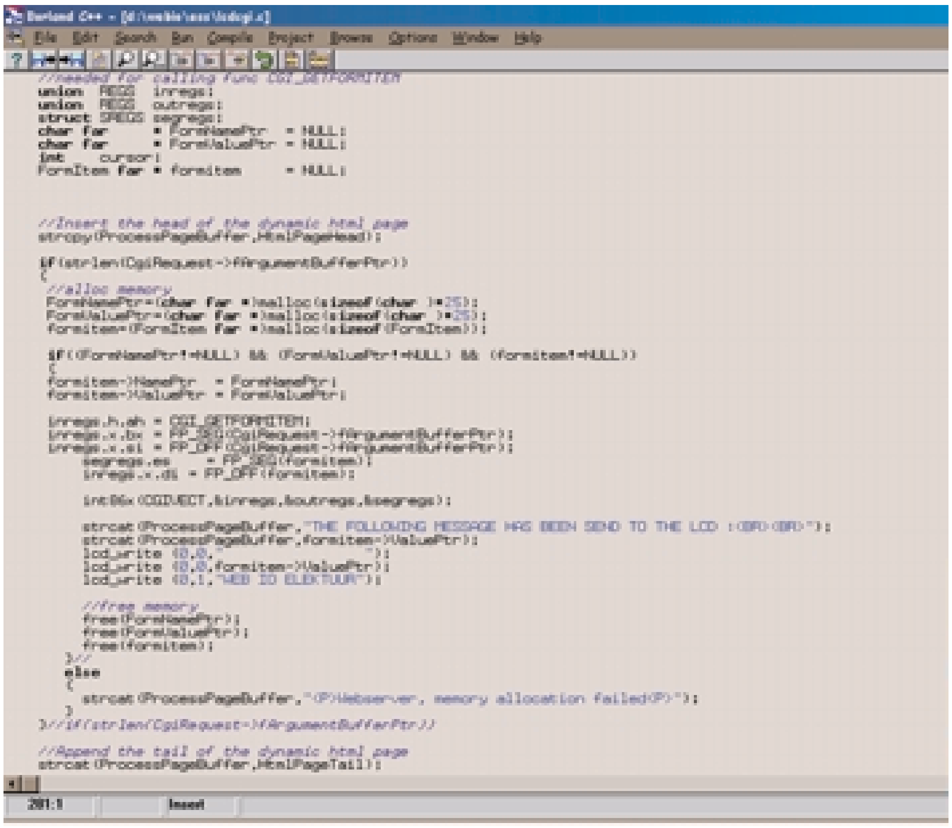
Using the command

A:\REBOOT

the Personal Mini Webserver can be rebooted remotely.

Http server with CGI

The best known application of TCP/IP is without doubt the Hypertext Transfer Protocol (HTTP). This protocol is used by most browsers (Netscape Navigator, Internet Explorer, etc.) for displaying web pages. The SC12's BIOS includes its own webserver. This means that you can store HTML pages in the Personal Mini Webserver and request them


Borland C++ - [d:\mobile\sss\lcdcgi.c]
File Edit Search Run Compile Project Browse Options Window Help
201:1
Insert

```
//needed for calling func CGI_GETFORMITEM
union  REGS  inregs;
union  REGS  outregs;
struct SREGS segregs;
char far      * FormNamePtr  = NULL;
char far      * FormValuePtr = NULL;
int    cursor;
FormItem far * formitem      = NULL;


//Insert the head of the dynamic html page
strcpy(ProcessPageBuffer,HtmlPageHead);

if(strlen(CgiRequest->ArgumentBufferPtr))
{
 //alloc memory
 FormNamePtr=(char far *)malloc(sizeof(char )*25);
 FormValuePtr=(char far *)malloc(sizeof(char )*25);
 formitem=(FormItem far *)malloc(sizeof(FormItem));

 if((FormNamePtr!=NULL) && (FormValuePtr!=NULL) && (formitem!=NULL))
 {
 formitem->NamePtr  = FormNamePtr;
 formitem->ValuePtr = FormValuePtr;

 inregs.h.ah = CGI_GETFORMITEM;
 inregs.x.bx = FP_SEG(CgiRequest->ArgumentBufferPtr);
 inregs.x.si = FP_OFF(CgiRequest->ArgumentBufferPtr);
      segregs.es     = FP_SEG(formitem);
      inregs.x.di = FP_OFF(formitem);

      int86x(CGIVECT,&inregs,&outregs,&segregs);

      strcat(ProcessPageBuffer,"THE FOLLOWING MESSAGE HAS BEEN SEND TO THE LCD :<BR><BR>");
      strcat(ProcessPageBuffer,formitem->ValuePtr);
      lcd_write (0,0,"                ");
      lcd_write (0,0,formitem->ValuePtr);
      lcd_write (0,1,"WEB IO ELEKTUUR");

      //free memory
      free(FormNamePtr);
      free(FormValuePtr);
      free(formitem);
    }//
    else
    {
      strcat(ProcessPageBuffer,"<P>Webserver, memory allocation failed<P>");
    }
}//if(strlen(CgiRequest->ArgumentBufferPtr))

//Append the tail of the dynamic html page
strcat(ProcessPageBuffer,HtmlPageTail);
```

Figure 6. Programming a LCD.

using a browser. A page can contain images, Java applets, sounds and so on.

Once the Personal Mini Webserver is connected to the network, the IP address must simply be entered into the browser to connect to the WEB@CHIP. If no particular page is requested by the browser, the Personal Mini Webserver starts with the main page (usually `index.htm` or `main.htm`). In order for it to know which of the stored pages this is, the following should appear in `chip.ini`:

```
[WEB]
ENABLE=1
MAINPAGE=INDEX.HTM
DRIVE=0
ROOTDIR=\WEB\
```

These pages should be stored in a separate directory in the general-purpose part of the flash disk. The file names must follow the 8.3 naming convention.

Publishing your own web pages is a diverting pastime, but our application aims to do more: we want to be able to obtain measurements over the network in real time, or control devices remotely. This is also made possible by the Hypertext Transfer Protocol (http).

There are two ways to produce a web page. The best known is to create an HTML file and send it to a browser. In most cases this is the ideal solution. However, if, for example, only a temperature reading needs to be updated, this procedure is awkward and inelegant. The processor is continuously occupied with producing and sending out new HTML pages which no-one requires.

The best way to send real-time data over the network is to use a program that employs the Common Gateway Interface (CGI). Such a program can produce an HTML page at the moment it is requested and send it to the HTTP server. The program also might, for example, cause the digital inputs to be read and build an up-to-date result into an HTML page.

CGI programs are also able to obtain information from the browser. Information entry is via HTML forms, which are used everywhere on the Internet. Information entered into the form is sent when the user clicks on the appropriate button.

Since this method of sending out HTML pages is not common and not trusted by many users, we have prepared several ready-made and thoroughly commented examples written using Borland C to show how the digital inputs and outputs are used, and how to control the LCD. The programs are available free of charge from the *Elektor Electronics* website or (at nominal cost) on a disk with order code **010036-11**. If you are reasonably familiar with C, you will have no difficulty in adapting these examples for your own purposes.

Programming

It is not necessary to become familiar with a new language and a new environment in order to program the Personal Mini Webserver. In principle any (old) DOS compiler will do to generate code for the 80196. The most widely used programming environments, Borland C and Turbo Pascal, have been used with success in the *Elektor Electronics* laboratories. A host of DOS compilers are also available freely on the Internet.

It is outside the scope of this article to go into the details of programming in C. Nevertheless, we can show a fragment of source code in **Figure 5**. This fragment would be called whenever a browser such as Internet Explorer, Netscape Navigator or Lynx requests a page. The routine constructs an HTML page which will be delivered by the HTTP server.

Test programs which exercise the individual functions of the Personal Mini Webserver are available from the *Elektor Electronics* website as well as on the above mentioned project diskette. The source code is of course fully commented so that you will quickly get used to how the Personal Mini Webserver works. As mentioned above, Beck's company Internet site contains complete documentation of all functions of the SC12 and its APIs, as well as an interesting FAQ list and a lively newsgroup. Particularly interesting are the links to the colleges at Winsen and Mosbach in Germany, where experimental systems have been developed using the SC12 and where comprehensive introductory courses are run.

(010036-2)

RS485 Meets CAN

two systems on one bus

By K. J. Thiesler

The new TTP/A (Time Triggered Protocol type A) field bus technology embraces both RS485 and CAN and is now being introduced into automotive and process control applications.

The TTP/A protocol allows a CAN bus device and an RS485 transceiver to work together in the same system. Originally, the CAN bus was designed for use in automotive applications, while the RS485 bus was designed for process control applications. In this article we will show how to operate the two systems on a single bus at data rates up to 1 Mbit/s.

CAN and RS485 use a practically identical data transfer medium: both systems send data differentially over twisted-pair cable. The signal levels, however, are different, and so communication errors will occur of the two systems are connected directly.

RS485

The bus topology is strictly linear, and a termination resistor of around 120 Ω is required at either end. An RS485 driver chip contains two push-pull output stages that drive the low and high signals onto the bus with low output impedance. When transmitting, the TxD output stage is enabled via its enable input; in the quiescent condition the output stage is switched to a high-impedance mode to allow other devices on the bus to transmit. **Figure 1** shows the internals of an RS485 transceiver.

A digital signal at the input is converted into a differential signal on the bus. A high level (logic 1) corresponds to a positive differential signal ($U_A > U_B$), while a low level (logic 0) corresponds to a negative differential signal ($U_A < U_B$). In RS485, the two states use the same voltage levels. This symmetrical feature enables the transmission of clean pulse edges with bus frequencies up to 12 Mbit/s. A disadvantage is that an enable input is required to turn the output driver on and off.

Figure 2 gives a comparison of the absolute bus voltage levels. The values shown are only indicative: they will vary depending on how the components on the bus are connected. In contrast, the relative values shown in **Table 1** are the maximum and minimum values specified for the transmission protocol to allow all devices on the bus to communicate with one another. The maximum length of the bus depends on the transfer rate. The absolute maximum is 1200 m at a data rate of 93.7 kbit/s with at most 32 devices on the bus. With the aid of repeaters (signal amplifiers) four groups of 32 devices can be connected together. Examples of RS485 interface devices are the SN75176 from Texas Instruments and the MAX485 from Maxim.

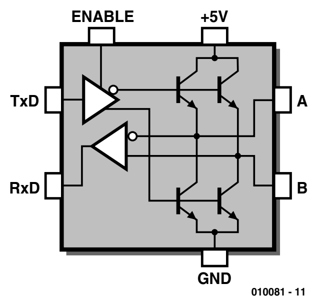


Figure 1. Internals of an RS485 transceiver.

CAN

The CAN bus was developed by Bosch and Intel in 1981 for networking electronic modules in cars. Since then it has found use outside its original domain and established itself in consumer devices and in industrial control. The communications protocol defines the CAN signal using a 'dominant' and a 'recessive' voltage level. A low level (logic 0) corresponds to a positive differential signal ($U_{CAN\,H} > U_{CAN\,L}$) and uses the dominant bus voltage level. A high level (logic 1) corresponds to a small differential signal ($U_{CAN\,H} \approx U_{CAN\,L}$), using the recessive bus voltage level. **Figure 3** shows the internals of a CAN interface chip.

The dominant 'L' level can override the recessive 'H' level without collision or interference. Hence no enable signal is required. The rather higher output impedance, however, limits the maximum communication rate to 1 Mbit/s.

Physically, the CAN bus looks exactly like the RS485 bus. It consists of a twisted-pair cable that must be terminated at both ends with 120 Ω resistors to obtain a typ-

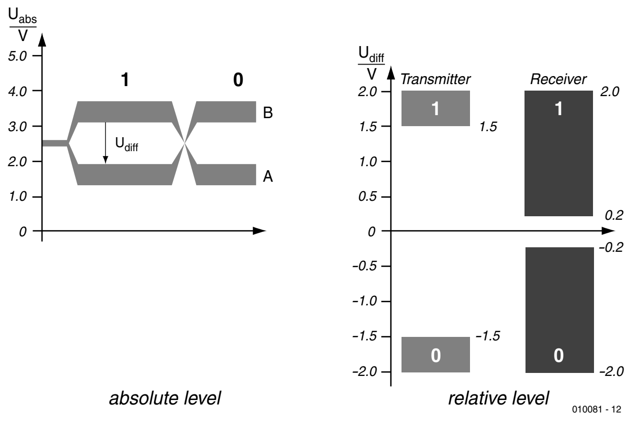


Figure 2. RS485 bus levels.
For operation, only differential values are relevant, and these are the same for all devices. In contrast the absolute values play only a secondary role. Their values can vary within broad limits depending on the device and load and even between devices of the same type.

Relative RS485 bus levels

U_{DIFF}	'H' level	'L' level
Receiver	0.2 to 2.0 V	–0.2 to –2.0 V
Transmitter	1.5 to 2.0 V	–1.5 to –2.0 V

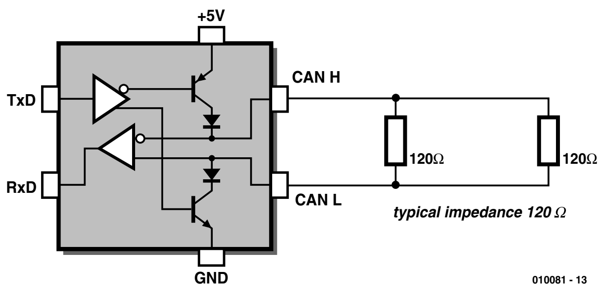


Figure 3. Internals of a CAN transceiver.

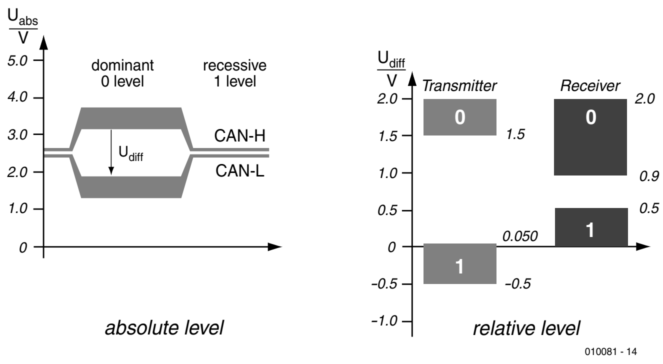


Figure 4. Relative and absolute bus levels.

Relative CAN bus levels

U_{DIFF}	recessive 'H' level	dominant 'L' level
Receiver	0 to 0.50 V	0.90 to 2.00 V
Transmitter	0.05 to –0.50 V	1.50 to 2.00 V

ical impedance of 120 Ω.

The absolute and relative signal levels are shown in **Figure 4** and **Table 2**. Here also it is only the differential values that are important; the absolute voltages are only indicative and can vary depending on the devices connected to the bus and its topology.

The physical length of the bus depends on the special arbitration method of the CAN bus, and the signal propagation delays limit the possible length. At a communication rate of 1 Mbit/s the maximum allowable length is 30 m, while at 62.5 kbit/s it is 1000 m.

Well-known CAN bus transceiver devices include the SN75LBC031 from Texas Instruments and the PCA82C250 from Philips.

Interconnection

CAN and RS485 are rather similar. Differential signalling is used in both systems. The signal levels, both absolute and differential, partially overlap. The data transmission medium is identical, with allowable impedance between 100 Ω and 120 Ω. The RS485 driver consists of two push-pull output stages. In the CAN transmitter two open-collector transistors drive the signal onto the bus.

Because of the symmetry of the outputs, an RS485 network can operate at a significantly higher data rate than a CAN network. In summary, the two standards only differ significantly in the bus signal levels. A CAN receiver can understand the levels of the RS485 protocol without modification; however, in the opposite direction errors will occur in the transmission. An RS485 receiver will recognise correctly only the 'L' level of the CAN transmitter. The CAN 'H' level, by contrast, is undefined in the RS485 system.

In order to get the RS485 receiver to understand the CAN 'H' level, it must be offset negatively by at least 0.25 V. This can be achieved with a 470 Ω pull-down resistor. In order to make the signal levels once again symmetrical about approximately 2.5 V, the CAN 'L' level must be raised by the same amount, by fitting the CAN 'L' wire with a 470 Ω pull-up resistor.

A difficulty is caused by the enable input, which causes the RS485 interface to be disabled during reception by switching off the push-pull output stages. By changing the input circuit to the RS485 interface device we can put the output into a tristate condition, effectively converting it into an open-collector output stage. The enable input serves as the data input, and the TxD input is held low. For a logic 1 the output turns on and drives the logic 0 signal from the TxD input onto the bus. For a logic 0, the output stage turns off an the two outputs go to a high impedance state. Compared to the original communications protocol, the outputs are now inverted. If we exchange the A and B outputs, the inversion is undone. **Figure 5** shows the full circuit of how the buses are linked.

In Summary

With two added resistors and the modified input circuit to the RS485 driver IC the problem is solved. We no longer have dominant and recessive signals. This unified field bus system might be called 'biased CAN'. Baud rates of 1 Mbit/s present no problems. Despite the increased load due to the coupling resistors the signal edges do not suffer.

Without the external pull-up and pull-down resistors the network has an impedance of 60 Ω, and with the resistors an impedance of 48 Ω. The effect of the two 470 Ω resistors can be clearly seen on the voltage difference between the two lines, which is around –0.3 V. Otherwise the two lines would be affected by both the recessive CAN 'H' level and the (disabled) RS485 high level.

(010081-1)

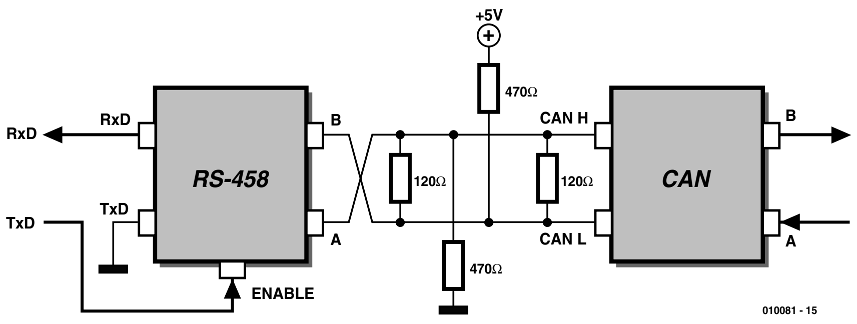


Figure 5. 'Biased CAN' field bus with tristate control of RS485 output stage.

ISAC (1)

part 1: introducing the Intelligent Sensor/Actuator Controller

Design by Prof. Dr. B. vom Berg, P. Groppe and M. Müller-Aulmann

MicroConverters are 8-bit microcontrollers specially designed for sensor/actuator applications. These new components from Analog Devices combine a powerful 8051-family microcontroller core, including flash program and data memory, with a multi-channel 12-bit analogue interface which includes an A/D as well as a D/A converter.

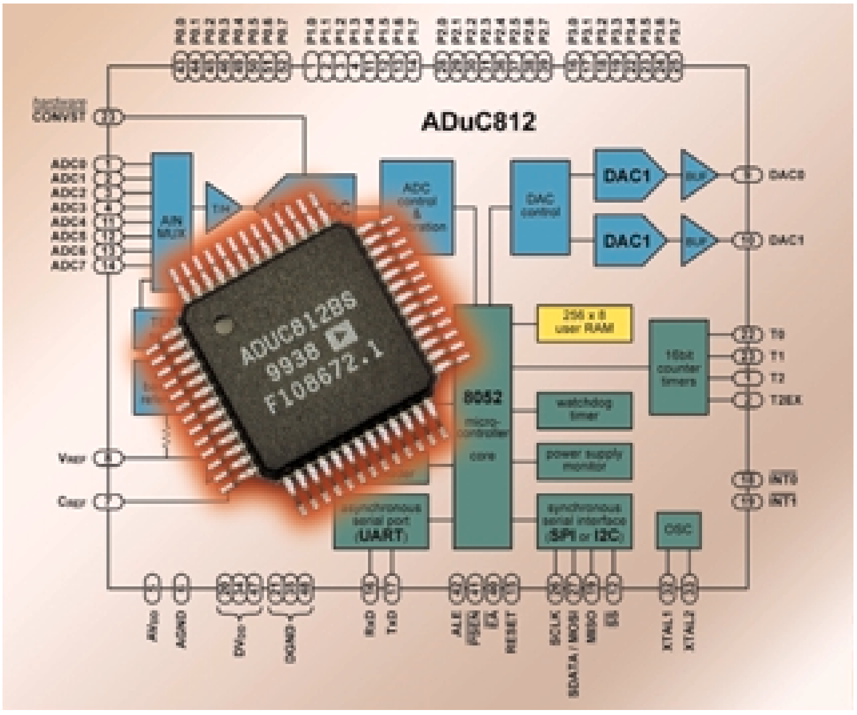
In this first article in the ISAC series we describe in detail the main component, the ADuC812 MicroConverter, along with its on-chip peripherals. In the second part we will develop a small experimental circuit which will let us take the first easy steps — and the later, more complicated steps — in the world of the MicroConverter.

Analog Devices produces, free of charge, a complete integrated development environment (IDE) for the device, which allows easy programming in Assembler or in C. We will look at this in the third instalment.

This series is self-contained: it includes several applications, such as push-button connection, a radio time-code receiver, and an opto-electronic shaft encoder interface. Later, in separate articles, we plan to describe further interesting projects, such as an LCD interface and driver and a CAN-bus connection, so that we can keep you up-to-date with details of newly-developed members of the MicroConverter family.

The name ISAC hints at microcontroller-operated sensor/actuator applications, and indeed the MicroConverter is ideally suited for control projects.

With the concept of the MicroConverter, Analog Devices has fulfilled the requirement of today's development and application engineers for the so-called 'system on a chip' (SoC). The well-known analogue IC technology of this manufacturer, in particular with regard to high-end A/D and D/A converters,



has been combined with mature CMOS flash memory technology and a microcontroller core accepted as the industry standard. The features of the new MicroConverter family are listed in **Table 1**.

As well as fulfilling this requirement for SoCs, the MicroConverters, in combination with one or more sensor/actuator systems, allow so-called 'intelligent sensor/actuators' to be constructed.

The first member of the MicroConverter family, the ADuC812, appeared in mid-1999. **Table 2** shows the technical characteristics of this device, and of the subsequently-developed members of the family.

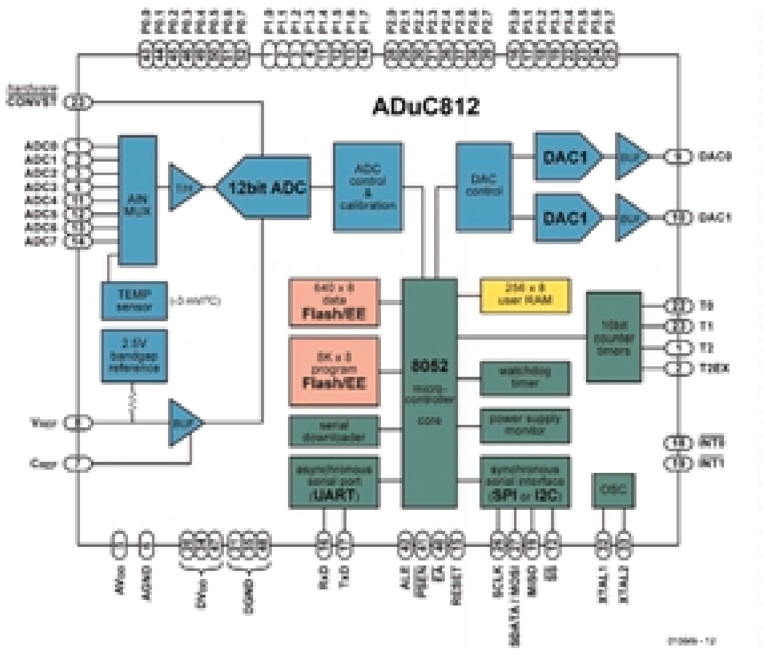


Figure 1. Block diagram of the ADuC812.

The ADuC812

We begin with the first member of the family, which is shown in the photograph at the start of this article. The innards can be seen in the block diagram (**Figure 1**). Let us now take a look at the individual functional units:

The microcontroller core is a standard 8051 core, which in the ADuC812 has all the well-known and well-loved features as well as a few extras, as shown in **Table 3**. Thanks to this core, all the programming languages and programming tools from the 8051 world can be used with the MicroConverter. The chief programming languages available are Assembler51 (ASM51), Basic52, Pascal51 and C51. One example of an easy-to-use integrated development environment is μVision from Keil. A complete collection of tools, the 'QuickStart Development Tools', is available from Analog Devices: these will be described in the third article in this series. Extensive information about programming 8051s in the Pascal51 programming language can for example be found in [1].

A special feature not found in the 8051 is the on-chip 8 kBytes of flash EEPROM program memory. This can be programmed quickly and easily from the development PC via the on-chip UART with a user program. The required 'Serial Downloader' and the high-voltage supply needed for programming the flash memory is already built into the MicroConverter chip, so no further external components are required for downloading.

On the PC side an easy-to-use Windows program allows trouble-free transfer of Intel hex object files to the MicroConverter: this will be described in the next article in this series. A detailed map of the program memory of the ADuC812 is shown in **Figure 2**.

Because the ADuC812 already has 8 kBytes of flash EEPROM program memory integrated onto the chip, in many applications no external program memory (EPROM, for example) is required. For larger-scale applications the program memory is expandable up to a total of 64 kBytes: the ADuC812 automatically switches between the internal and external program memory areas.

Soon, another milestone in 8-bit microcontroller technology will appear on your workbench: the ADuC812/824B2 (see **Table 2**) is the first 8051-series microconverter in the world to offer the full 64 kBytes of program memory in the form of a directly user-programmable flash EEPROM integrated onto the chip. External program memory expansion is then not required, unless memory-mapping is required using banks.

The final key question which remains is: what about the (still) external data memory area of the 8051 (**Figure 3**)?

It is notable that the ADuC812 does not have an on-chip RAM area that can be used as an external data memory. The user is forced, as with the 8051, to use the 256 byte internal RAM for storage and processing of data. If more data memory is required, external expansion is possible up to 16 Mbytes(!) (a 'normal' 8051 can only access 64 kbytes

Table 1

MicroConverter features

– Industry standard microcontroller core:

- 8 bit:	8051 core

– Precision analogue I/O interface:

- A/D converter:	- 12 bit or higher resolution
	- 5/8 channels
- D/A converter:	- 12 bit resolution
	- 1/2 channels

- 2.5 V on-chip reference voltage source, optional external voltage reference source easily connected

– Flash EEPROM memory:

- Program memory:	8/64 Kbytes
- Data memory:	640 byte / 4 Kbytes

- On-chip programming voltage supply
- Serial program downloader

– Interesting extra on-chip circuits (in addition to microcontroller peripherals):
- Temperature sensor
- Watchdog timer
- Supply voltage monitoring
- SPI/I^{2}C bus interface
- CAN bus interface

and many others

– Summary:

First complete precision data acquisition system on one chip

Drastically reduced circuit board area and lower power consumption than previously-available multiple chip solutions directly). The ADuC812/824B2 derivative provides 4 kbytes of on-chip RAM which can be used for external data storage.

A further special feature of the ADuC812 is an extra 640 bytes of on-chip flash EEPROM area, which the user can employ for any purpose, such as non-volatile storage of conversion values or identification codes. Access to this memory is via a special register (the Special Function Register, or SFR), and so this flash memory area lies outside the normal ADuC812 address space.

Table 4 gives an overview of the peripheral functional blocks integrated onto the chip. The particularly outstanding features of the MicroConverter are the highly accurate interfaces to the analogue world, the A/D and D/A converters. At present, the D/A converters offer a resolution of 12 bits, while the A/D converters offer 12-, 16- or 24-bit resolution, depending on the ADuC device in question.

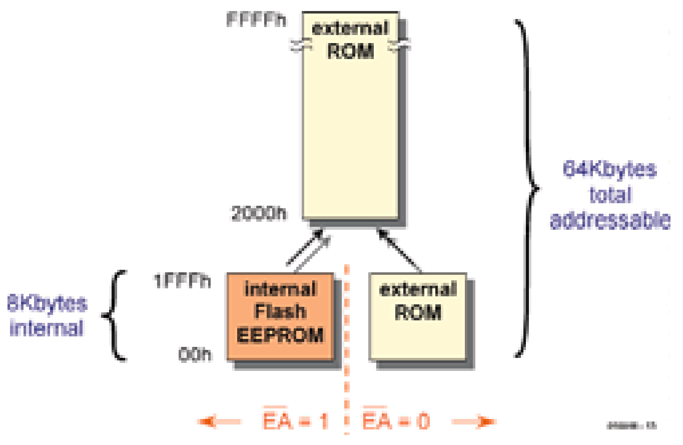


Figure 2. Program memory map.

The A/D converter

On the ADuC812 the A/D converter has a resolution of 12 bits and has eight input channels. Its most important characteristics are listed in **Table 5**. As well as the high resolution (for a microcontroller on-chip converter) of 12 bits, the short conversion time of 5 μs is particularly noteworthy. If the converter is driven continuously at this high speed, however, the microcontroller will have difficulty in storing the raw data which are arriving every 5 μs. The traditional method of operating the converter requires the following steps, as illustrated in **Figure 4**:

Table 2

Technical details of the MicroConverter series (ADuC812xx ≡ Analog Devices MicroConverter, 8 bit data bus, 12 bit A/D converter resolution, xx indicating derivative components)

	AduC812	AduC812B2	AduC812S	AduC816	AduC824	AduC824B2
ADC	12 bit, 8 channels	12 bit, 8 channels	12 bit, 6 channels	16 bit, 2 channels	2 channels, 16 bit and 24 bit	2 channels, 16 bit and 24 bit
DAC	12 bit, 2 channels	12 bit, 2 channels	12 bit, 2 channels	12 bit, 1 channel	12 bit, 1 channel	12 bit, 1 channel
Core	8052 compatible	8052 compatible	8052 compatible	8052 compatible	8052 compatible	8052 compatible
Internal RAM memory	256 bytes	256 bytes	256 bytes	256 bytes	256 bytes	256 bytes
Internal FLASH memory	8 kbytes program, 640 bytes data	***64 kbytes program,*** 4 kbytes data	8 kbytes program, 640 bytes data	8 kbytes program, 640 bytes data	8 kbytes program, 640 bytes data	***64 kbytes program,*** 4 kbytes data
External on-chip RAM memory	———	2 kbytes	———	———	———	4 kbytes
External memory	64 kbytes program, 16 Mbytes data	64 kbytes program, 16 Mbytes data	16 Mbytes data	64 kbytes program, 16 Mbytes data	64 kbytes program, 16 Mbytes data	64 kbytes program, 16 Mbytes data
PIO	32	32	16	26	26	26
Serial	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI
External crystal frequency	400 kHz - 16 MHz	32.768 kHz, on-chip-PLL 0.131 - 16.77 MHz	32.768 kHz, on-chip-PLL 0.131 - 16.77 MHz	32.768 kHz, on-chip-PLL	32.768 kHz, on-chip-PLL	32.768 kHz, on-chip-PLL
Package	52 MQFP	52 MQFP	28 TSSOP	52 MQFP	52 MQFP	52 MQFP
Availability	1998 / 1999	End 2001	End 2001	December 2000	October 2000	End 2001

Notes

In versions which use a 32.768 kHz quartz crystal and on-chip PLL, a real-time clock (RTC) can be implemented in software by using an additional TIC.

In versions with on-chip PLL the CPU operating frequency can be set to one of eight different values. This makes it possible to reduce current consumption dynamically.

Versions with on-chip CAN controller are planned.

Versions in smaller packages are also planned (Chip-Size Package = CSP).

Table 3

Basic 8051 functionality, with extensions:

- 8 bit microcontroller core with 128 bytes of user RAM (256 bytes on ADuC812)
- 4 kBytes of on-chip ROM (8 kBytes of flash EEPROM program memory on ADuC812)
- up to 64 kBytes of external program memory can be connected
- up to 64 kBytes of external data memory can be connected (up to 16 Mbytes on ADuC812)
- 32 digital I/O port pins (four eight-bit-wide ports P0-P3)
- two 16-bit timer/counters (three 16-bit timer/counters on ADuC812)
- 5 interrupt sources assignable to two interrupt priority levels (9 sources and two levels on ADuC812)
- full-duplex serial interface (UART)
- on-chip oscillator
- clock frequency 400 kHz to 16 MHz (12 MHz nominal)
- two power-save modes: idle and power-down

Futher special features of ADuC812:
- 640 bytes of on-chip user flash EEPROM, usable as data memory
- P1 port pins can be used only as digital inputs or as inputs for the on-chip A/D converter
- P3 port pins have high current capability, sinking up to 8 mA (1.6 mA on standard 8051)
- operating voltage: 3 V or 5 V
- current consumption (at 12 MHz clock):

	5 V	**3 V**
Normal	26 mA	12 mA
Idle	15 mA	6 mA
Power down	50 μA	50 μA

1. Start the A/D converter.
2. Wait for the converter's ready signal (by polling or by interrupt).
3. Read data.
4. Address external data memory.
5. Write data into external memory.

It is worth noting that steps 3 and 5 each require two access cycles, since the 8051 is an 8-bit machine and the converter result is 12 bits wide. This is rather too much for the 8051 core, which, at the maximum speed of the A/D converter, is faced with a new conversion result every 5 μs. The bottleneck is the necessity to transfer the A/D converter data via the interal accumulator of the 8051. In order to solve this problem, Analog Devices have equipped the A/D converter with a DMA (Direct Memory Access) controller (**Figure 5**).

With the aid of the DMA controller the A/D converter is now able to bypass the 8051 microconverter core and store its data directly in the external data memory. In this way the maximum continuous speed of 200 Ksps (200,000 samples/s) can be achieved.

While DMA is going on, the 8051 core can get on with other operations as long as it does not need to access the external memory. If it attempts such an access, the 8051 will simply be held off by the DMA controller for the duration of the DMA transfer so that no contention arises.

The DMA transfer completes when the A/D converter has carried out the specified number of measurements and written the results into the external memory. The DMA controller then terminates its activity and informs the 8051 core, by means of an interrupt, that the desired values are available in the external memory. The 8051 core now has full control once more over the external memory, and so can read out the stored values and process them.

At lower sample rates, for example one value every 10 ms (100 samples/s), the DMA controller need not be used, and the 8051 can cope with the job on its own.

The digital temperature sensor

In addition to its eight externally visible input channels, the A/D converter has a 'hidden' ninth channel, which is connected to an analogue on-chip temperature sensor. This delivers a highly linear voltage signal proportional to the instantaneous chip temperature (!), which can be converted by the A/D converter and hence provide the user with information as to the chip temperature, and, with suitable processing, the ambient temperature. There will be more details on these applications in the next instalment.

The voltage reference

The ADuC812 includes an internal band-gap voltage source (+2.5 V), which can be used as a reference voltage for the A/D and D/A converters. An external reference voltage source in the range +2.3 V to +5.0 V can also be connected, the internal reference voltage source then being automatically shut down.

The D/A converter

The two D/A converter channels of the ADuC812 have the following characteristics:

- 2 separate D/A channels, guaranteed monotonic
- 8-bit or 12-bit resolution selectable
- 8 μs settling time
- buffered output voltage (output load ≥ 10 kΩ)
- output voltage range:
  0 to internal VREF (+2.5 V), or
  0 to AV_{DD} (+5 V)
- rail-to-rail operation (ΔU ≤ 100 mV)
- DNL: ± ½ LSB

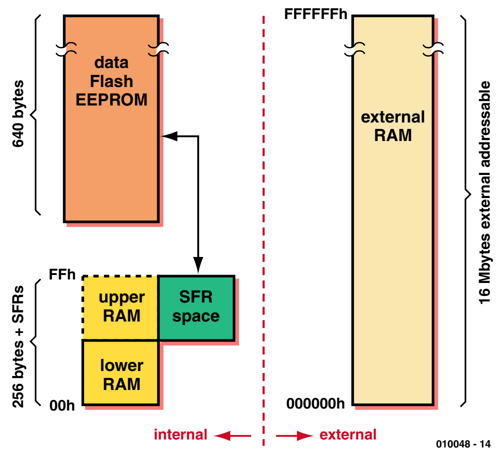


Figure 3. Data memory map.

Table 4

Extra on-chip functional units integrated onto ADuC812:

– 8-channel A/D converter with 12 bit resolution
– digital temperature sensor
– 2-channel D/A converter with 12 bit resolution
– 2.5 V voltage reference source
– SPI/I^{2}C bus synchronous serial communications interface
– watchdog timer
– power supply monitor

Table 5

The ADuC812 A/D converter:

– 8-channel, self-calibrating A/D converter with 12 bit resolution
– successive approximation conversion
– conversion time: 5 μs => sample rate of 200 Ksps (200,000 samples per second)
– reference voltage V_{REF}: internal +2.5 V source or external source +2.3 to +5.0 V.
– input voltage range: 0 V to V_{REF}
– integrated sample-and-hold unit
– integrated DMA controller enabling transfer of data to memory at maximum sample rate
– INL: ± ½ LSB
– SNR: 70 dB

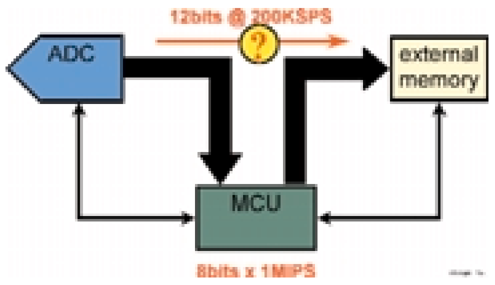


Figure 4. Driving an A/D converter using the 'traditional' method.

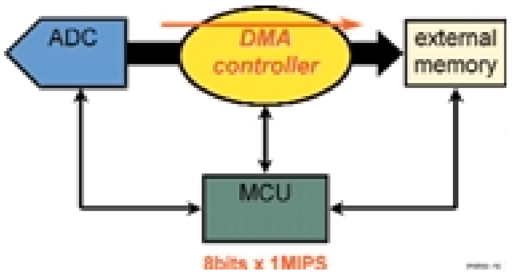


Figure 5. Driving an A/D converter using DMA.

The D/A converter will also be dealt with in more detail when we look at its applications.

The synchronous serial interface

The synchronous serial communications interface can be configured as follows:
1. SPI (Serial Peripheral Interface):
Standard 3-wire serial communications interface, full-duplex, master/slave operation, four selectable data transfer rates.
2. I^{2}C bus (inter-IC bus interface):
Standard 2-wire data communications interface, master/slave mode, 7-bit addressing mode.

The watchdog timer

This on-chip peripheral monitors the running of the MicroConverter's program. In the case of an error, a reset is forced, so that the ADuC812's program can at least start afresh from a known initialised state. The monitoring period is selectable in eight steps in the range 16 to 2048 ms.

The power supply monitor

The two supply voltages AV_{DD} and DV_{DD} are monitored by this functional unit. If the voltages fall below one of five selectable monitor thresholds (for example total power loss or a brown-out) then this monitoring unit immediately provides an interrupt to the microcontroller core before the voltage drops below a critical lower threshold. Thus the program could enter an interrupt service routine to store important data in a non-volatile storage area such as EEPROM or battery-backed RAM before the system fails completely.

It is then possible, when the supply returns, to analyse these data and thereby restore the system to its previous state or determine the causes of the supply loss or brown-out.

The next step

After this overview of the hardware structure of the ADuC812, we will present in the next part of this series a complete experimental system with which you will be able to carry out your first experiments with the ADuC812. If you cannot wait until the next issue for more, there is plenty of information to be found on the Internet: Analog Devices have created a special website for MicroConverters, where you can download data sheets, application notes, and (most importantly) the free development software package, including assembler and C compiler [2]. The site is worth a visit is any case, since there is much more detail on the ADuC812 in the datasheet, and the software will be used in the third part of this series.

(010048-1)

Literature and websites:

[1] B. vom Berg, P. Groppe: Das 8051er Lehrbuch, Elektor-Verlag, Aachen, second edition (available in german only)
[2] MicroConverter Internet homepage: www.analog.com/microconverter
[3] Analog Devices on the Internet: www.analog.com
[4] Keil μVision IDE: www.keil.com

Neural Networks in Control

neurocomputers are here

By Owen Bishop

Most of the actions of most animals are controlled by a network of nerve cells, or *neurons*. In this article Owen Bishop looks at the possibilities of implementing the neurocomputer.

Neurons are usually organised into well-defined structures, such as the brain, the spinal cord and the peripheral nerves. Each neuron is a centre of information transfer (Note: transfer, not storage). It can receive information as electrical signals from a number of other neurons in the network, process this information, and pass it on to other neurons. In the human brain, for instance, a neuron connects to as many as 10,000 other neurons. The extent to which information is transferred from neuron to neuron depends on the number of physical connections existing between the neurons and the intensity (or strength) of each connection. The number of possible connections is far greater than the number of neurons present. We estimate that a human brain may contain as many as 10^{16} connections, an enormous resource for memory and intelligent action.

Learning

A young animal begins life with a given set of network connections built in. It soon adds to these connections by experience as it learns useful responses to a range of outside stimuli. Learning involves not only making totally new connections, but also by strengthening some existing connections and weakening others. Learning is thus an essential feature of a neural network. The animal learns how to act in a given set of circumstances. It also learns patterns, as when we learn to recognise friend or foe, or to identify the smell of a well-barbecued steak. Parts of the network are sites of higher thought processes.

Given that the neural network of an animal is capable of complex recognition and control actions, researchers have tried to build the artificial equivalent, an **artificial neural network** or **ANN**. An ANN consists of neuron units mimicking the action of animal neurons. They have input, processing and output stages. The earliest electronic ANNs were based on valve circuitry, but little progress was made until the advent of VLSI. Now, with the enormous computing power of even the smallest of bench-top computers, we model ANNs mathematically. This can be done in a system based on conventional microprocessors, as illustrated by the software described later. There are also custom-designed processors used to build true **neurocomputers**. These are designed to excel in the particular kinds of processing that are needed to model ANNs, which is done most efficiently by using parallel processing.

It is important to realise that, although ANNs are built from neurons (electronic or

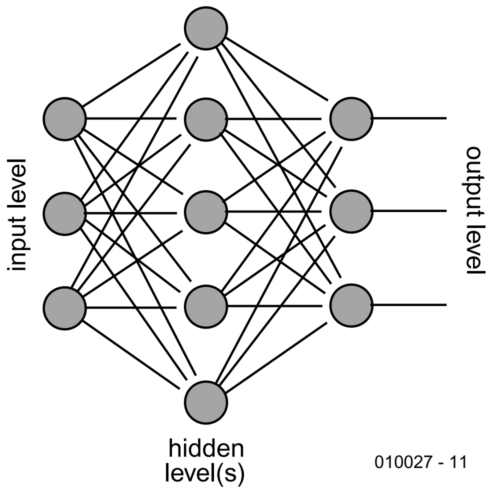


Figure 1. An artificial neural net consists of at least three levels of connected neurons.

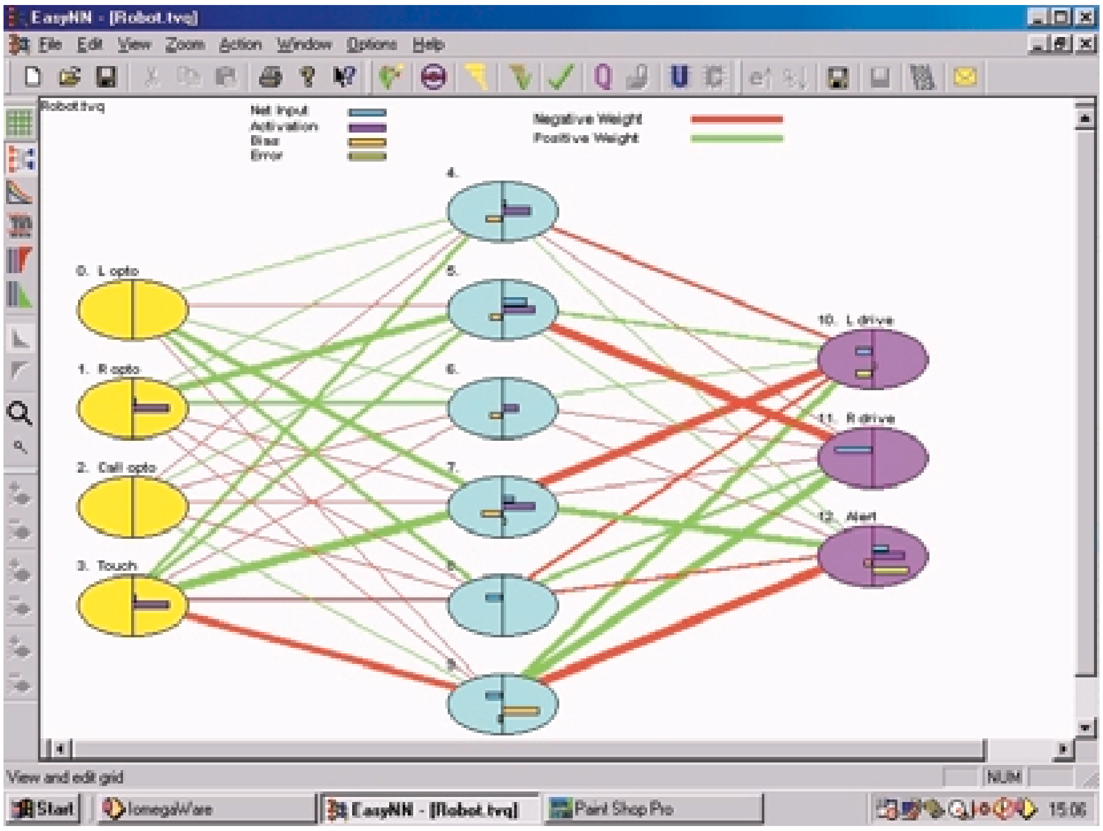


Figure 2. This ANN models the control system of a platform robot.

mathematical), the action of which superficially resembles that of animal neurons, the analogy goes no further. We know a lot about the structure and activities of animal neurons but very little about how they interact. A neurocomputer operates in a way that is probably very different from the operation of a brain. Nevertheless, it can perform similar functions.

Architecture

The structure of a typical ANN (electronic or mathematical) is illustrated in **Figure 1**. It consists of at least three layers (or levels) of neurons, and there may be any number of neurons in each level. The input level consists of neurons that receive input from sensors. In a control system, these might include thermal, optical, proximity or pressure sensors. In a visual pattern recognition system, the inputs might be from specified areas in the field of a video camera. The **output level** consists of the neurons that drive actuators, such as motors, solenoids and lamps. On the other hand, maybe they generate bits in a digital code that is then interpreted by a computer. Between the input and output levels are one or more **hidden levels**, which are not directly connected to the outside world.

In the network of Figure 1, signals pass in one direction from the input level, through the hidden levels, to the output level. This is known as **feedforward architecture**, and is the one most commonly used. Certain applications may have other architectures involving feedback, connections between neurons on the same level, and other configurations.

Modelling an ANN

A number of computer programs are available for modelling ANNs, including a PC program known as **Easy NN**, by Stephen Wolstenholme. To study an ANN in action, we set up a control network for a hypothetical platform robot. The robot circulates the workstations in a factory, to collect completed workpieces and to deliver spare parts. Its route is determined by a white line painted on the floor. The line is a complete loop with branches to each of the workstations. The robot straddles the line and has two optical sensors at the front, which scan the floor to the left and right of the line. When the robot is moving along the line, neither of the sensors detects the line and both its drive motors are running. However, if it deviates from the line, or where the line bends, one sensor or the other is stimulated and the appropriate drive motor is switched off to cause the robot to return to the line. When there is a branch line, the extra width of the tracks stimulates both sensors at the same time. Then the robot may ignore the main track and branch off to the left toward the workstation, but it only does this if its ‘call’ optical sensor is receiving a beam of light directed at the robot from the workstation.

The robot also has a touch sensor at the front, to detect a barrier when it reaches the workstation. Then it sounds an alert and stops while items are unloaded or loaded, until the barrier is moved aside. It then continues along the branch track until it rejoins the main track. The touch sensor also detects obstacles it may encounter. Then it stops and sounds the alert until the obstacle is removed.

This is an oversimplified control system that would probably not work well in practice, but it serves to illustrate the nature of an ANN.

Another point is that this network has only binary inputs and outputs. However, mathematical models of ANNs are perfectly capable of working with analogue values.

Training

The first stage in setting up the network using **Easy NN** is to specify that there are to be four input neurons, three output neurons, and a single level of hidden neurons. The software was left to decide how many hidden neurons there should be. Next, the network was taught how to behave by keying in a table, showing a selection of possible input values and what output values are expected in each case. Since this is a binary example, we keyed in all sixteen possible combinations of the four inputs at levels 0 (false) and 1 (true). We also keyed in the expected output responses for each combination of inputs. We simply decided what the robot is to do and keyed in '0's and '1's accordingly. This is known as training by **supervision** and is the equivalent of programming an expert system. Because the system is working on binary values, the table is the equivalent of a truth table.

When the computer is told to generate the network, it first decides how many neurons are needed in the hidden layer. Then it connects all the neurons (as in Figure 1) and gives each connection random but relatively low weighting. It then puts in the combinations of inputs, computes the corresponding outputs and compares them with the expected outputs. This process is repeated and, because each neuron has a memory associated with it, the weightings can be adjusted to produce an output that more closely matches the expected output. The processing continues until stable values are obtained. The network is then displayed as a diagram (**Figure 2**). From this, we can see that the computer has elected to have six hidden neurons. The inter-neuron connections are shown as lines of varying width, depending on the weightings. Some of these are displayed in green to indicate positive (excitatory) action. The others are in red to indicate negative (inhibitory) action. The signals arriving at each neuron are summed by the neuron in proportion to the weightings. In most systems a number of weak signals arriving at a neuron are ignored. Only when the total of signals exceed a given threshold, does the neuron generate an output signal.

Figure 2 represents one particular combination of inputs and the corresponding outputs. Parameters at each neuron are shown by bar charts of which the second bar down represents the activation state. To investigate the behaviour of the network (and the robot) we call up a 'Query' panel and key in the level of each sensor signals (0 or 1). The resulting output is automatically calculated and a version of Fig. 2 with the new inputs and outputs is displayed. In Fig. 2, for example, the activation states have been set so that the right opto-sensor and the touch sensor are both '1'. The left opto-sensor and call sensors are both '0'. The corresponding output is '0' for both drives and '1' for the alert. The robot has met an obstacle on its path, has stopped and is sounding the alert. If we alter the inputs to make all opto-sensors '1' and the touch sensor ''0', the output instantly changes to left drive '1' (ON), right drive '0' (OFF) and alert '0' (OFF). The robot is turning right at a junction. Trying various combinations of inputs always gives the result expected according to the training that was given.

This simple controller could have been built from three or four hardwire logic gates, or programmed as a few lines of logic commands in a BASIC program. On the other hand, we could have programmed the 'truth table' as a lookup table and searched it to find the correct outputs. Instead, we made use of algorithms that describe the action of a neuron and what happens when several such neurons are connected in a network. The only causal connection between input and output is through the ANN. Incidentally, the network does only what we have trained it to do. If we train it to respond differently, it will learn the new pattern of behaviour.

Analogue input

One of the examples available at the **Easy NN** site shows how well an ANN can cope with analogue input. The network is trained to identify a specimen of one of three species of *Iris*, given only the length and width of the petals and sepals. The network has four input neurons, one for each of the four input quantities. It has three output neurons, one for each of the species, *I.setosa, I.versicolor* and *I.virginica*. The computer generates seven neurons in the hidden layer.

With analogue input, the network can not be trained by tabulating all possible combinations of inputs. Instead, it is given a sample of analogue data to work on. The four measurements taken from each of a sample of flowers are keyed in to the data table. 'True' is entered in one of the three output columns to indicate to which species each set of data refers. 'False' is entered in the other two columns. The table holds data from 50 flowers of each species, which is considered sufficient for training.

Although the correspondence between dimensions and species is by no means easy to distinguish by inspection, the network quickly learns to identify the species of other specimens of *Iris* of these three species from these four basic measurements.

A similar but much larger network is able to distinguish between sixteen species of cichlid fish, using 10 input neurons receiving analogue data on body features.

Applications

Given a certain pattern of input, an ANN learns to produce an appropriate pattern of output. This makes ANNs. We have shown examples in the fields of control and taxonomy. ANNs are often used for scanning masses of data looking for patterns. It is reported that data on credit card fraud revealed that fraud is especially common among young women buying shoes. Visual recognition is important in security systems and robotics. ANNs are finding many applications in these fields. The flexibility of ANNs and their ability to respond to changes in input make them useful in tuning the parameters of fuzzy logic systems. In the future, we may find that many devices in the home are controlled by a combination of fuzzy logic and an artificial neural network.

Software

At the time of writing, full details of **Easy NN** are published at http://www.tropheus.demon.co.uk/.

(010027-1)

RS232 In-Line Data Spy

instant analysis of serial links

Design by M. Müller

With the help of a minimal quantity of hardware and a simple piece of software it is possible to tap into the data stream between two serial interfaces and observe it on the computer monitor.

The name 'Data Spy' immediately conjures up images of clandestine practices, in general however, the circuit described here will not be used for such purposes. There are many legitimate reasons for analysing the data stream between a PC and a peripheral, or a second PC. It could be part of a faultfinding procedure, for example, or as a con-

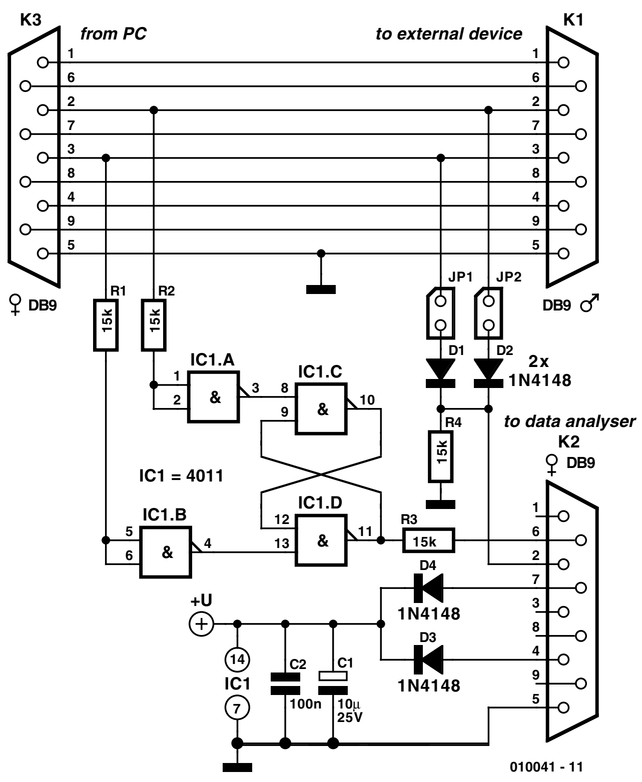


Figure 1. The hardware for the Data Spy is really simple. The supply voltage is 'stolen' from the PC.

tinuous data monitor, or as an educational exercise to unravel the details of certain protocols.

The implementation of the circuit described here is a kind of in-line plug that is simply inserted in series with the connection between the RS232 interfaces of the PC and the peripheral under consideration. A third sub-D connector functions as the connection for the additional PC, which is used to examine the data stream.

The Data Spy is very easy to use. The software is compatible with Windows9x and WindowsNT4.0, the data can be displayed in ASCII, decimal or hexadecimal format. With two circuits in series, it becomes possible to observe both data streams when they occur in both directions simultaneously. The only requirement is that the two data streams must have identical baud rates. The circuit does not require its own power supply, because its supply voltage is obtained from the analysing PC.

Hardware

A brief glance at **Figure 1** makes it clear that there is very little to the necessary hardware. In fact, it consists of no more than a flip-flop (bistable) (IC1) and some ten passive components.

The path K1/K3 will be in series with the serial connection to be examined. As can be seen, all data and control lines are simply

COMPONENTS LIST

Resistors:
R1-R4 = 15kΩ

Capacitors:
C1 = 10µF 25V radial
C2 = 100nF

Semiconductors:
D1-D4 = 1N4148
IC1 = 4011

Miscellaneous:
K1 = 9-way sub-D plug (male), PCB mount
K2,K3 = 9-way sub-D socket (female), PCB mount
JP1, JP2 = jumper
PCB, order code **010041-1** (see Readers Services page or Elektor website)
Project software: code **010041-11**, free download from Elektor website.

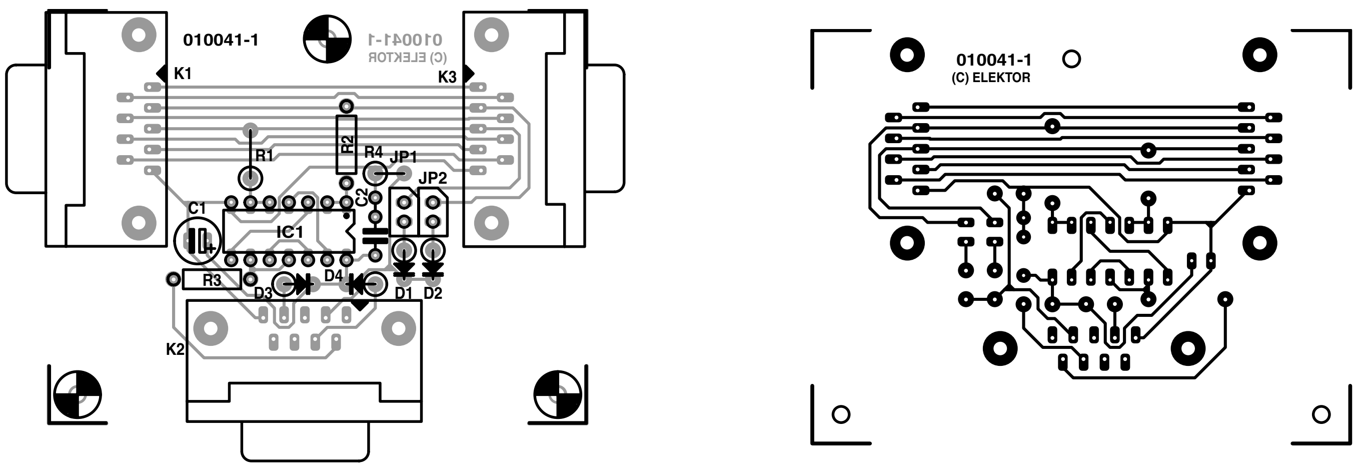


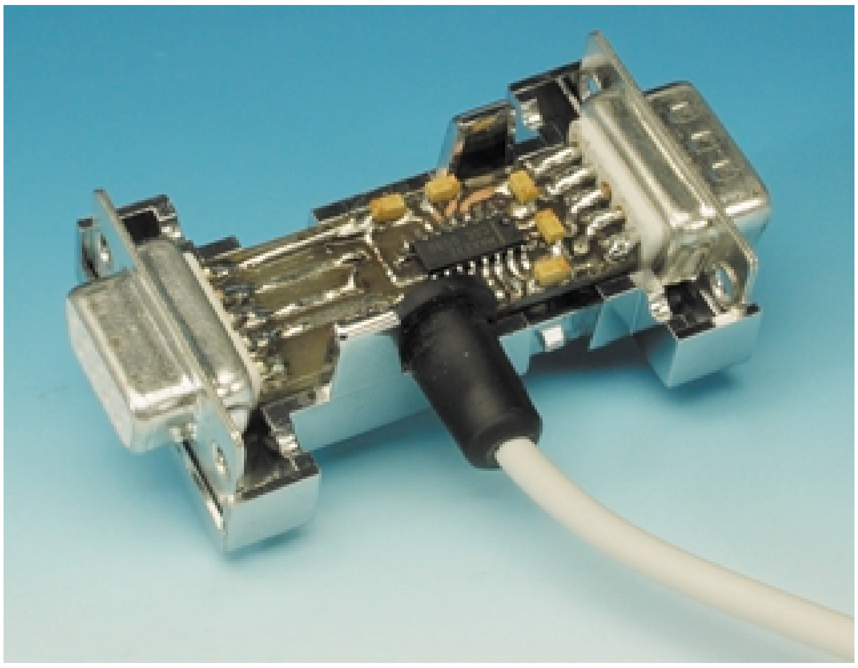
Figure 2. The printed circuit board for the 'in-line plug' is available from Elektor Electronics Readers' Services.

connected straight through. For our snooping activities, only the data lines from either side are required. When data is received from one of the directions, the flip-flop is either set or reset and the corresponding data direction is indicated with the DSR control line. With the aid of jumpers JP1 and JP2 a selection can be made between the data from either source; if JP1 is closed, data from the PC is looped through to K2, and when JP2 is closed, the data is from the external device or the 'SPS' (Serial Peripheral System). The signal, via D1/D2 and R4, is applied to pin 2 (RXD) of K2, to which the monitoring PC is connected.

Resistors R1 and R2 ensure that IC1 is not subjected to voltage spikes or negative potentials. The resistance value selected, in conjunction with the internal capacitance of IC1, has the result that the maximum baud rate is limited to 38,400 baud (kbits/s).

The power supply for IC1 is obtained from the PC, via D3 (DTR), D4 (RTS) and C1. This is not a problem because of the extremely low power consumption of IC1. This way, an additional power supply with all its wiring is avoided. Capacitor C2 provides additional RF decoupling.

Figure 2 shows the printed circuit board for the project. Since there are so few components, the construction of the circuit should take an hour at most. If you follow the component overlay closely and stick to the parts list than it is very unlikely that something will go wrong with the practical realisation of the hardware.



Software

The program, written and compiled in Borland Delphi 5, has the name 'BinTerm' (from binary terminal) and can be downloaded from the *Elektor Electronics* web site. The item number is **010041-11**. The program runs under Windows9x and WindowsNT4.0 and consists of only one file (`BinTerm.exe`). When exiting the program, all set-ups are stored in an INI-file.

The program performs three tasks:

Main task: handles all visible elements.

Serial port read task: storing all incoming characters in a circular buffer (94000 symbols).

Display task: reading the data from the circular buffer and displaying it on the screen.

The three tasks work independently of each other. Unfortunately, processing the data synchronously is not possible with these 32-bit operating systems. To compensate for this, the serial interface is read by its own task at higher priority. The displaying of data is then processed using a second task. The data, therefore, is stored correctly together with the corresponding direction and is displayed on the screen a bit later. If the data arrives too fast, or if the turnaround time between transmission and reception is too short, then the PC is unable to properly determine the direction.

Usage

After starting `BinTerm.exe` you are faced with a blank screen onto which the incoming data will be displayed, as well as a few buttons. If you are using two Data Spies in series in order to read the simultaneous data in both directions, then the screen is divided into two windows, as illustrated in **Figure 3**.

The operation of the buttons is as follows:

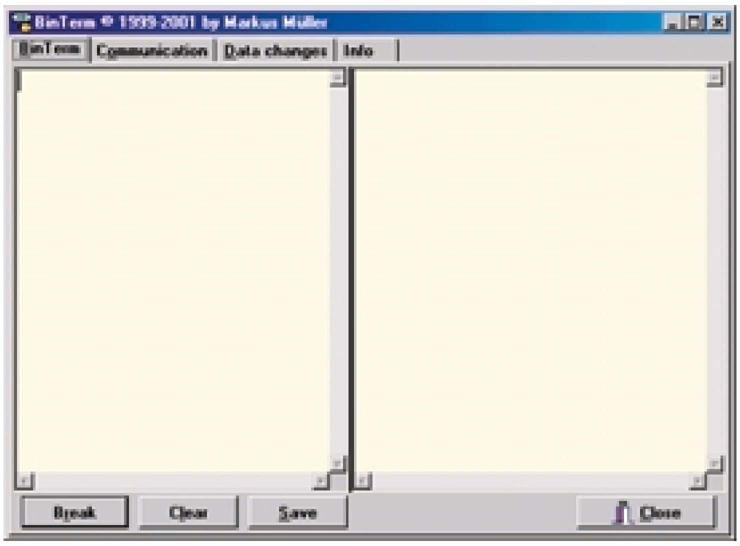
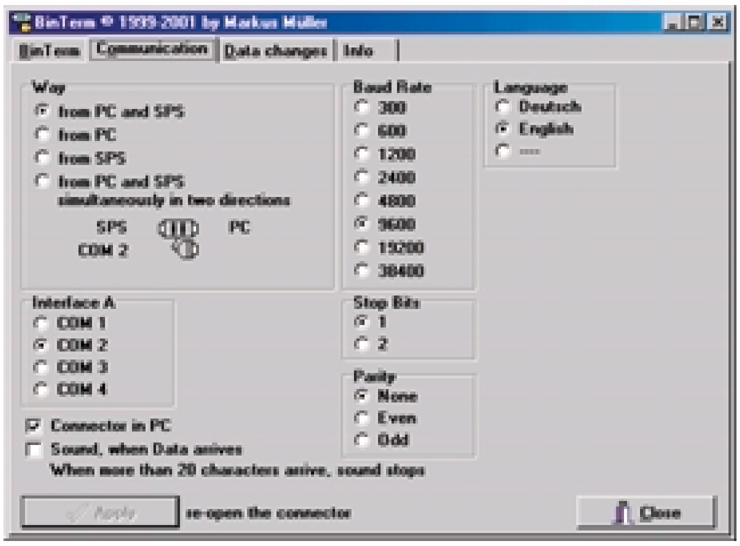


Figure 3. Initial window after starting BinTerm, in this case divided into two for the display of two-way traffic.


BinTerm © 1999-2001 by Markus Müller

BinTerm | Communication | Data changes | Info

Way
- [✓ from PC and SPS]
- from PC
- from SPS
- from PC and SPS simultaneously in two directions

SPS COM 2 PC

Baud Rate
- 300
- 600
- 1200
- 2400
- 4800
- [✓ 9600]
- 19200
- 38400

Language
- Deutsch
- [✓ English]
- ----

Interface A
- COM 1
- [✓ COM 2]
- COM 3
- COM 4

Stop Bits
- [✓ 1]
- 2

Parity
- [✓ None]
- Even
- Odd

[✓ Connector in PC]

Sound, when Data arrives

When more than 20 characters arrive, sound stops

Apply | re-open the connector | Close

Figure 4. Under 'communication' all the settings necessary for data communications can be found.

break: pause the display of data;

clear: erase the contents of the window;

save: store the displayed contents in a text file;

close: quit the program.

Individual characters and lines can be selected and copied with CTRL-c and pasted into a text editor with CTRL-v. After clicking '**break**' the legend on the button will change to '**continue**'. Clicking the button again will resume the display of the data on the screen.

To facilitate the storage of the data in a text file, BinTerm writes as the first line 'left box', followed by the text displayed in the left box. This is followed by the line 'right box' after which the content of the right box is stored.

Communications

After clicking the tab 'communication' at the top of the screen, the setup window appears (**Figure 4**), where most options will require little comment. A few remarks follow nonetheless.

The '**apply**' button causes the serial interface to be reopened with the new settings; the '**close**' button ends the program.

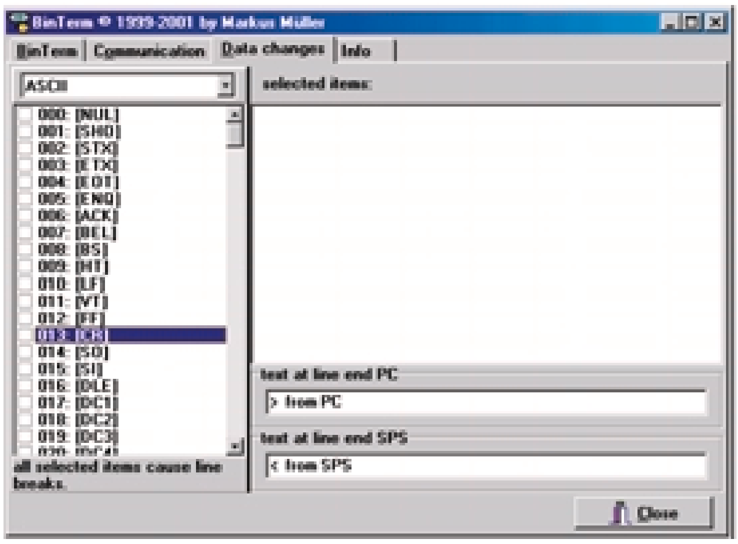
Of course, the interface parameters (baud rate, stop bits and parity) need to be set correctly. The language can be selected under '**language/sprache**'. When selecting the method of data collection (box '**way**'), the correct position of the jumpers is automatically indicated. Collecting data simultaneously in both directions can, as previously mentioned, only be achieved with two Data Spies connected in series. In principle, the data rate has to be the same in both directions, and in practice this will be nearly always the case. There is a little trick however, that allows the Data Spy to operate during the rare occurrence of two different baud rates. Naturally, you need two circuits connected in series and start BinTerm twice. In the first copy, select 'from PC' and 'COM1' and in the second version, select 'from SPS' and 'COM2'.

If the data stream contains relatively few characters then the arrival of each of these may be indicated with an audible signal ('**sound**'). If however, the data rate is greater than 20 characters per second then this feature will be automatically disabled. The higher the data rate, the faster the PC needs to be in order to determine the data direction correctly.

Data changes

After clicking on the tab '**data changes**', another setup window will be displayed (**Figure 5**). Here you can choose how the data is displayed: ASCII, decimal or hexadecimal. It is also possible to add an interpretation for each item. The appropriate characters will then be displayed in the right-hand window. A double-click will remove the selected items from the list.

When the displayed data from the PC wraps to the next line, the text 'text at line-end PC' is shown, when the same happens with SPS data the text 'text at line-end SPS' appears.


BinTerm © 1999-2001 by Markus Müller

BinTerm | Communication | Data changes | Info

ASCII

selected items:

```
000: [NUL]
001: [SHO]
002: [STX]
003: [ETX]
004: [EOT]
005: [ENQ]
006: [ACK]
007: [BEL]
008: [BS]
009: [HT]
010: [LF]
011: [VT]
012: [FF]
013: [CR]
014: [SO]
015: [SI]
016: [DLE]
017: [DC1]
018: [DC2]
019: [DC3]
```

all selected items cause line breaks.

text at line end PC: > from PC

text at line end SPS: < from SPS

Close

Figure 5. Under 'data changes' selection can be made from three data formats and individual items can also be selected.

Final Remarks

After clicking on the tab '**info**' the name, email- and Internet address of the designer of Data Spy and BinTerm software is displayed. Any remarks or requests for further information should be directed there.

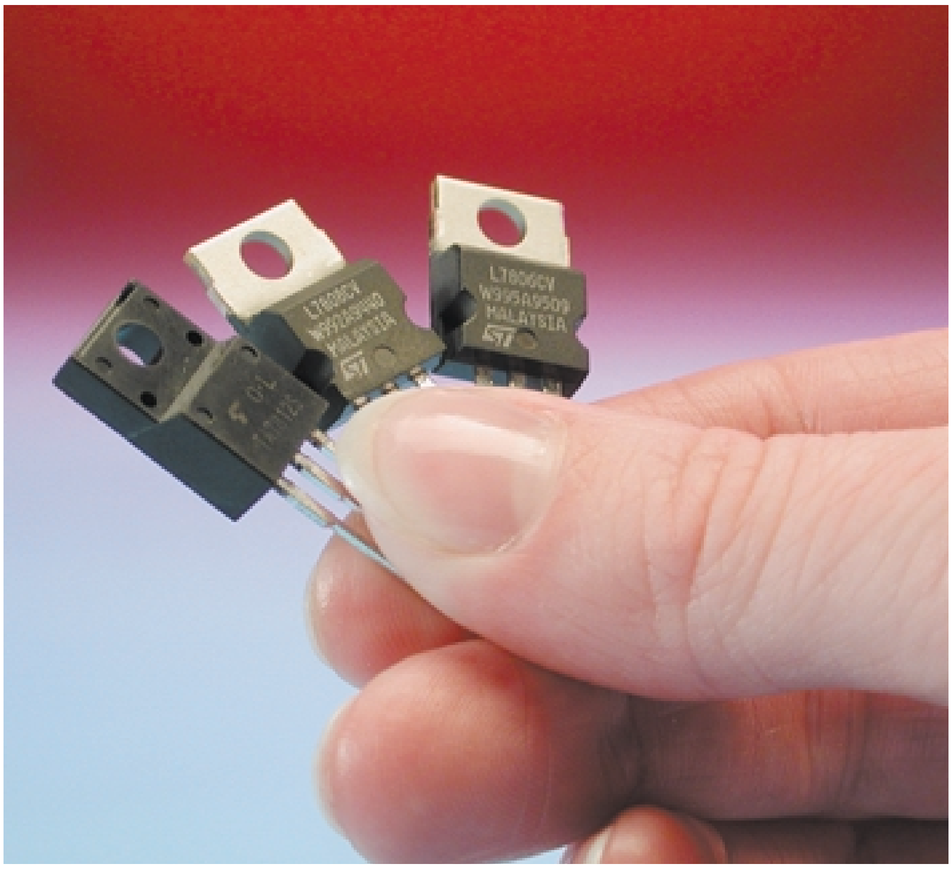
(010041-1)

Voltage Regulators

in practice…

From an idea by A. Voigt

Three-legged voltage regulators are known as handy and straightforward components. There are, nevertheless, still a number of interesting facts to report. As always, practice is never quite the same as theory.



Since their introduction, the three-legged voltage regulators have quickly displaced their discretely built counterparts. And with good reason, why be difficult when one component will suffice? They are available for practically any desired output voltage and the performance of the famous 78xx-series is usually more than adequate for most applications. In addition, hardly anything can go wrong, because it comes with a robust thermal and overload protection circuit built in.

The only demand that the three-legged creature makes, is that the unregulated DC input voltage is at least 3 V higher than the output voltage. Otherwise the regulator circuit cannot do its job properly.

These 78xx-regulators are handy building blocks, because they need little real estate on the PCB and require almost no external components. The schematic of a stabilised power supply employing one of these regulators usually looks something like **Figure 1**. The transformer output voltage is rectified with a diode bridge and smoothed with filter capacitor C1. C2 and C3 improve the stability of the regulator as well as its transient response, while C4 acts as the local buffer (reservoir) for the attached load.

The annoying thing about the standard schematic is that it does not make clear what the specific functions of each of the components are. Strictly speaking, it would be better to redraw the schematic of **Figure 1** like the one sketched in **Figure 2**. Granted, it doesn't look quite so neat and tidy, but makes the purpose of the various components immediately clear. Filter capacitor C1 should be placed as close as possible to the bridge rectifier, C2 and C3 should connect directly to the input and output of the regulator and C4 should be very close to the load. At least as important is the requirement that all the 0-V traces are connected at only one common point. The 'cold' side of the output electrolytic is the most appropriate place for this.

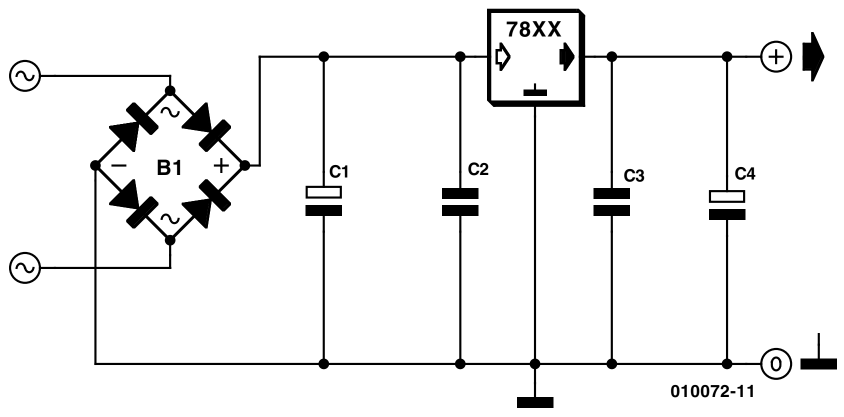


Figure 1. Typical circuit with a voltage regulator from the 78xx-series.

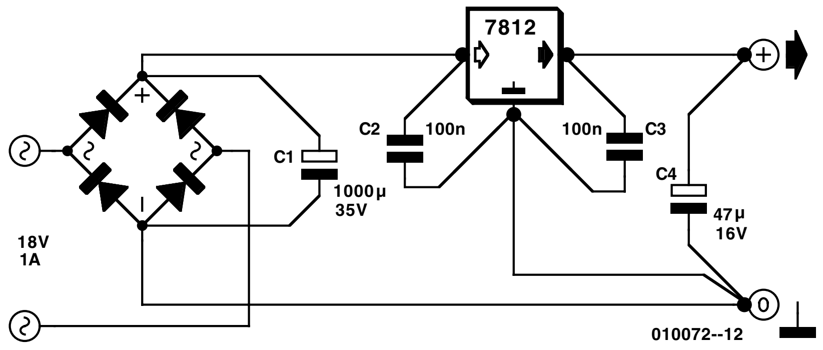


Figure 2. Functionally it would be better to draw the schematic this way.

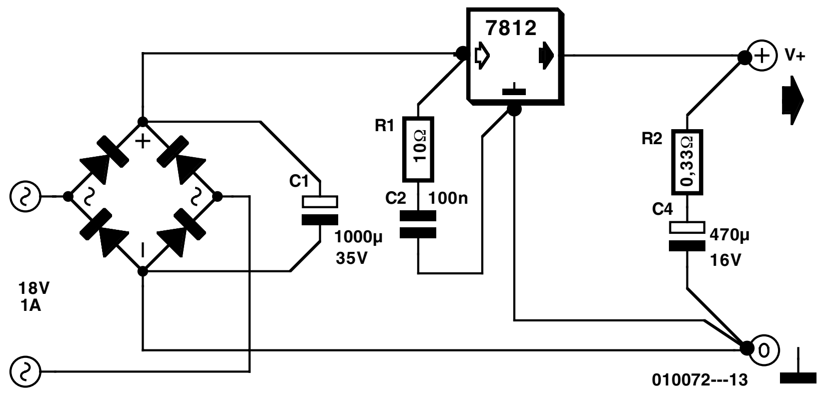


Figure 3. Adding two resistors can improve the transient response under certain circumstances.

The stability, ripple rejection and transient behaviour of the regulator is much better with the design of **Figure 2** than with some haphazard placement of components. This is particularly the case when, due to the circumstances of the PCB, the distance between the bridge rectifier and the load is relatively large.

Some remarks about the value of the capacitors. In practice, a value of 100 nF for both C2 and C3 appears to work well. The value of electrolytic C4 is not really critical and usually varies between 10 µF and 47 µF, depending on the output current. The following rule of thumb can be applied to filter capacitor C1: the value in µF should be at least equal, but preferably a factor of 2 greater, than the current in mA. Thus when taking the 1000 mA in the example of Figure 2, we arrive at a value of 1000 µF or 2200 µF.

Two additional resistors

Although we were unable to repeat some of the things in the Elektor Electronics lab with 100% accuracy, the author claims that the performance of the standard 78xx-circuit can be significantly improved with some small modifications. This manifests itself in a better transient response when the load varies rapidly.

The change does not amount to anything more than the addition of two series resistors for C2 and C4. Capacitor C3 can be omitted without loss of quality and C4 is increased in value. **Figure 3** shows the amended schematic.

It is difficult to come up with a rigorous scientific explanation for the improved properties. A possible theory is that the capacitors and the PCB traces may, under certain circumstances, form LC-loops that can lead to oscillation and cause the voltage regulator to react more slowly. A small series resistor can have a beneficial damping affect. The brand of regulator may also play a role.

Even though the beneficial effects of the series resistors may not be realised under all circumstances, it is certainly worth the effort to experiment with this. The cost is practically insignificant and if this simple method does indeed improve the dynamic behaviour of the regulator, then this is a welcome benefit.

The values for R1 and R2 are difficult to pin down. In the author's prototype, built around a 7812, the values shown in **Figure 3** performed the best. By experimenting it appears that the following principle holds true for R2: as the value of electrolytic C4 is increased, the resistor can be made smaller. The value of R1 is less critical. The advice: just try it in practice.

(010072-1)

Dead Pixels

how to identify and (sometimes) correct them

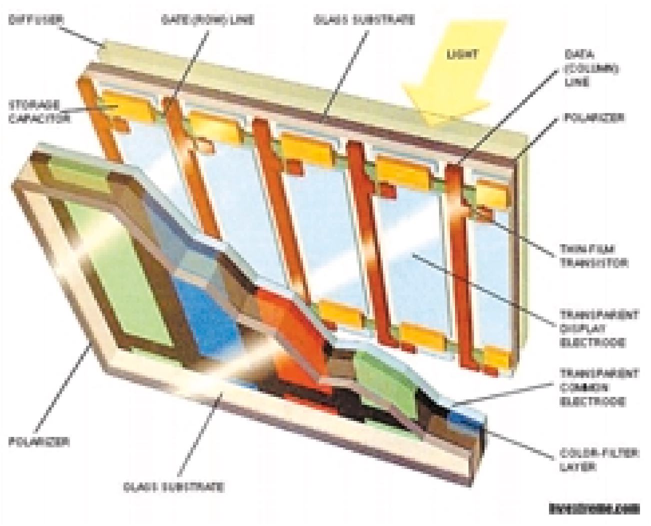
By H. Baggen

TFT monitors and digital cameras are becoming steadily more popular and more affordable. Unfortunately, users of such devices are frequently confronted with pixels that do not function correctly. How can you identify such pixels, and what can you do to correct them?

Dead pixels, stuck pixels, hot pixels: all of these terms refer to pixels in displays or CCDs that don't work properly. Users of TFT screens and Digicams complain increasingly about this problem. However, this is certainly not the result of poor product quality or inadequate quality control by the manufacturer. We are simply being confronted with a consequence of the fact that we demand increasingly higher resolution from our screen or camera. In the case of a 15-inch TFT monitor with a resolution of 1024 × 768 pixels, there are no less than 2,359,296 transistors on the glass plate, with their associated colour filters (red, green and blue). If something is wrong with one of these transistors, its associated switching electrode or a lead, there is a chance that this pixel will always remain fully dark or fully light. As a user, you will see this on the screen as a spot that always has the some colour: white, black, one of the primary colours or a combination of the primary colours. The manufacturer or importer will generally be inclined to replace a TFT screen with such 'dead' pixels only if a fair number of them are defective (e.g. seven).

If you want to know how a TFT screen works, you might want to look at Hardware Extreme [1] or Tom's Hardware Guide [2]. You can also find a good description of the dead-pixel problem on the American site of NEC/Mitsubishi [3].

In the case of a TFT screen, you can avoid buying an unsatisfactory unit by testing the screen before making your purchase, since there is no remedy for dead pixels. For this, you can use a standard monitor test program that can display uniform red, green and blue screens. However, there is also a small program that displays the three primary colours just for this purpose. This can be downloaded from the Notebook Buyer's HomePage [4] (among others).



In the case of digital cameras, a similar problem occurs with defective pixels, but the cause is different. However, the result is comparable to a TFT display. At fixed locations in the photo, for example, points may appear that always have the same colour (or are white). These are referred to as 'hot pixels' in the case of cameras, since the defective pixels almost always appear to be brighter than their surroundings.

You can find a good description of this phenomenon on the (German-language) site of FotoKing [5].

The manufacturer of a digital camera is usually able to eliminate defective pixels using software, so that they are no longer visible. However, you have to send the camera to the importer to have this done. A handy and free expedient for eliminating dead pixels is available from Mediachance [6]. 'HotPixels Eliminator' is a program that automatically 'repairs' defective pixels in all subsequent photos once the coordinates of these pixels have been entered.

(015094-1)

Internet Addresses:

[1] Hardware Extreme:
www.hwextreme.com/reviews/monitor/lcd/tech/print.shtml

[2] Tom's Hardware Guide:
www4.tomshardware.com/display/99q2/990624/tft1-05.html

[3] NEC/Mitsubishi:
http://support.necmitsubishi.com/nec/common/library/lcd_dead_pixel.htm

[4] Notebook Buyer's HomePage:
www.edgeworld.com/notebook/old/dead.htm

[5] FotoKing:
www.fotoking.de/body_hot_pixel_stuck_pixel.html

[6] Mediachance:
www.mediachance.com/digicam/hotpixels.htm

GPS Receivers with NMEA Output

principles and applications

By G. May DL3ABQ

GPS modules with NMEA output are available for less than 80 pounds, and compact hand-held GPS units with the interface are now hardly any more expensive. This article shows what can now be done with these units.

The Global Positioning System (GPS) allows the user to determine very accurately his position in space. Since the artificial error introduced for civilian users has been switched off, the accuracy is now around 20 to 50 m, and in practice an error of rather less than 10 m can be achieved. Using Differential GPS (DGPS) the accuracy can be improved still further with the aid of correction information.

GPS receivers have been getting cheaper and cheaper. Basic units are available for just over one hundred pounds. More user-friendly devices with more sophisticated mapping software, however, still command a rather higher price. If you are not interested in the ultimate in miniaturisation, there is an economical alternative: you can buy a 'pure' receiver without display, but equipped with a data output which you can connect to a port of your PC or hand-held computer. Such modules are available for under a hundred pounds, and suitable GPS mapping software for the PC is also not expensive.

The GPS Receiver

As can be seen from **Figure 1**, the GPS receiver circuit divides into two or three main parts: the antenna, which in some systems is separate from the rest of the device and which normally contains a preamplifier, the receiver proper, and a control circuit. The control circuit usually takes the form of a microprocessor with associated display and keyboard. Optionally a DGPS receiver can be incorporated, which receives a broadcast DGPS correction data signal.

A more detailed block diagram appears in **Figure 2**, taken from the data sheet of the Jupiter LP receiver module from Conexant (formerly Rockwell). The actual GPS receiver is generally built around a chipset consisting of two manufacturer-specific components. The HF frontend (or 'RF monopac') receives and demodulates the GPS signal. The rest of the job —

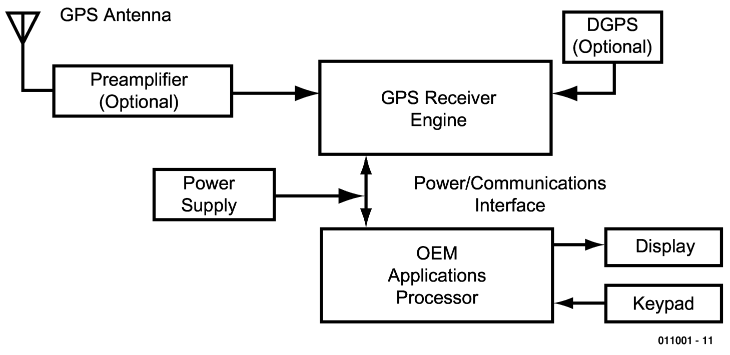


Figure 1. Block diagram of a complete GPS receiver.

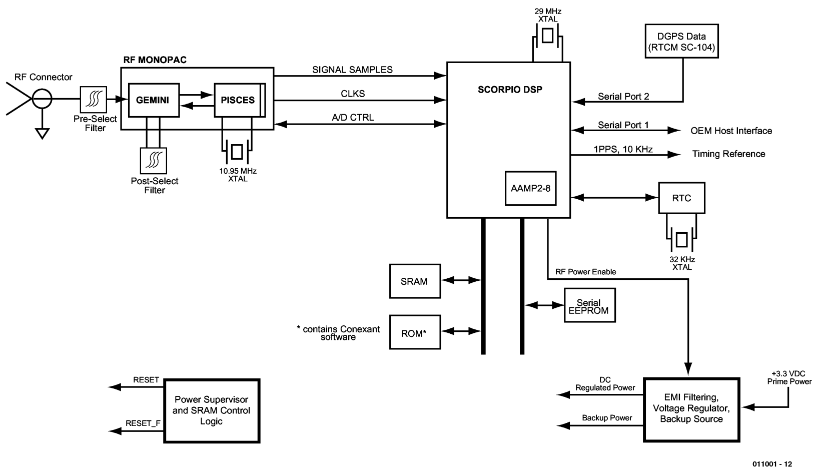


Figure 2. Block diagram of a GPS receiver module.

Figure 3. High-specification GPS receiver.

the processing and decoding of the GPS signals and the control of the frontend — is performed by a dedicated digital signal processor (DSP).

The hardware and the software form a significant part of the cost of a complete GPS receiver. That goes especially for large colour TFT displays, as well as mapping software and databases. **Figure 3** shows an example of a high-specification GPS receiver. We can do the same job by buying a simple receiver and using it in conjunction with a PC. The simple receiver can be a GPS module like the one shown in **Figure 4** or a so-called 'GPS mouse' (**Figure 5**), with integral antenna. Only a little dearer is a basic hand-held GPS receiver, such as the tiny Garmin device shown in **Figure 6**, which offers only a simple user interface–but also a data interface.

To use this interface it is if course essential that the PC software and the receiver communicate using a standardised protocol.

NMEA

NMEA is a communications protocol that was developed for data interchange between electronic sensors in marine applications.

Since GPS quickly found application in this area, NMEA has also established itself as a protocol standard for communications between GPS receivers and terminal equipment. Despite this there are still a few receivers around that use their own protocol. Since practically all PC applications use NMEA, these receivers should be avoided.

By 'NMEA' here we mean 'NMEA 0183', of which versions 1.5 and 2.0 are widely used. Generally version 2.0 is encountered, and newer receivers tend not to support the older versions (NMEA 0183 version 1.5, NMEA 0182 and NMEA 0180).

The NMEA output of the receiver is connected to the serial port of the PC. The RS232 parameters for the protocol are 4800 baud, 8 data bits, one stop bit and no parity. Observe that some receivers use TTL or CMOS levels at their interfaces; such units require the use of a level-shifter circuit. A suitable level shifter is the well-known standard MAX232 circuit shown in **Figure 7**.

If, however, the unit's manual claims that the device can be connected directly to a PC, no such complications are necessary.

NMEA Sentences

NMEA data is divided into records each at most 80 characters long, called sentences. The NMEA sentences can easily be displayed using a terminal program such as Hyperterminal for Windows or the Norton Commander terminal. The data can be read directly, since the sentences consist of printable ASCII characters. There are, however, rather more user-friendly alternatives, which we will describe in more detail below.

The sentences start with a '$', followed by a system ID, which in the case of a GPS receiver is 'GP'. The individual data fields follow.

In **Table 1** an RMC sentence is shown as an example, as generated by all GPS receivers (RMC = Recommended minimum specific GPS/Transit data). A full overview of the sentence format can be found at the NMEA FAQ [1].

PC Software

The best GPS receiver in the world is little use if there is no software to display its data satisfactorily. Fortunately there is a huge choice of free programs available for this purpose, of which a few are described briefly below:

GPSS (see **Figure 8**)
A highly recommended freeware program including map display. Unfortunately the user interface takes a little getting used to. The download address is www.gpss.co.uk/

NMEA Monitor
A freeware program to display NMEA data. Downloadable from
www.navtec.de/english/nmea_mon.htm

WinGPS
A very powerful commercial program with map display; a shareware version is available for download from www.stentec.com/

This is of course only a small selection; in particular there is no shortage of NMEA monitor programs. For portable use there is also software available for Palm and Psion machines etc.

Generally the programs that are capable of displaying maps already have some built in; more can be obtained, for example, from the well-known 'TOP 50' CDs or other collections of map data. In general the maps have to be calibrated using a couple of known fixed points on the map.

APRS

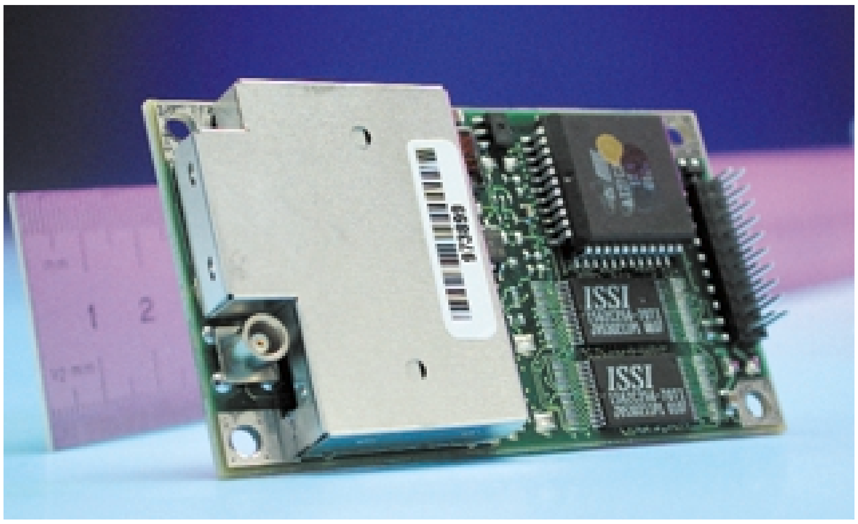


Figure 4. Example GPS module.

Table 1

Example sentence:

RMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68

225446	time of fix: 22:54:46 UTC (Universal Time Coordinated)
A	GPS receiver status (A: normal; V: warning)
4916.45,N	latitude: 49° 16.45' north
12311.12,W	longitude: 123° 11.12' west
000.5	speed
054.7	course
191194	Fix-Date: 19 November 1994
020.3,E	magnetic variation: 20.3° east
*68	checksum

Figure 5. A 'GPS mouse' includes a compact GPS receiver with integral antenna and can, according to the model, be connected directly to a PC's serial port or USB port. The photograph shows a NaviMouse (sold by Unitronic) for connection to a Palm PDA.

'APRS' (Automatic Positioning Reporting System) is a new amateur radio system wherein a GPS receiver with NMEA output is connected to amateur radio equipment. The various stations can then exchange position information. More information is available from:

www.dididahdahdidit.com/

Internet Links

There is a huge range of resources on GPS and NMEA worldwide. To conclude this introductory article we present a small selection:

GPS Primer
www.aero.org/publications/GPSPRIMER/index.html

National Marine Electronics Organisation www.nmea.org

Trimble
www.tri-m.com/products/trimble_navigation_ltd.products.html

CommLinx, an Australian supplier of GPS modules
http://commlinx.com.au/gps_oem.htm

Joe Mehaffey and Jack Yeazel's GPS Information Website
http://joe.mehaffey.com/

Peter Bennett's GPS and NMEA Site
http://zazu.optiva.ee/pub/nmea/

The Global Positioning System (GPS) Resource Library www.gpsy.com/gpsinfo/

GPS newsgroups
alt.satellite.gps
www.yahoogroups.com
(by subscription only; search for 'GPS')

(011001-1)

Figure 6. Even low-cost hand-held GPS units are available with NMEA output.

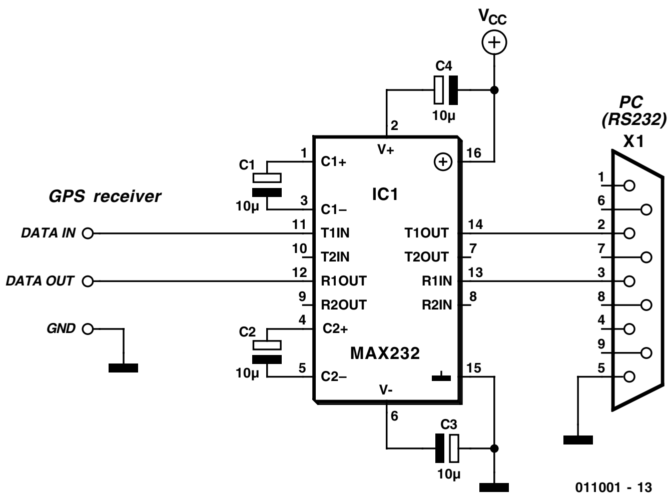


Figure 7. Some receivers require a level shifter to connect to an RS232 port. This standard MAX232 circuit fits the bill.

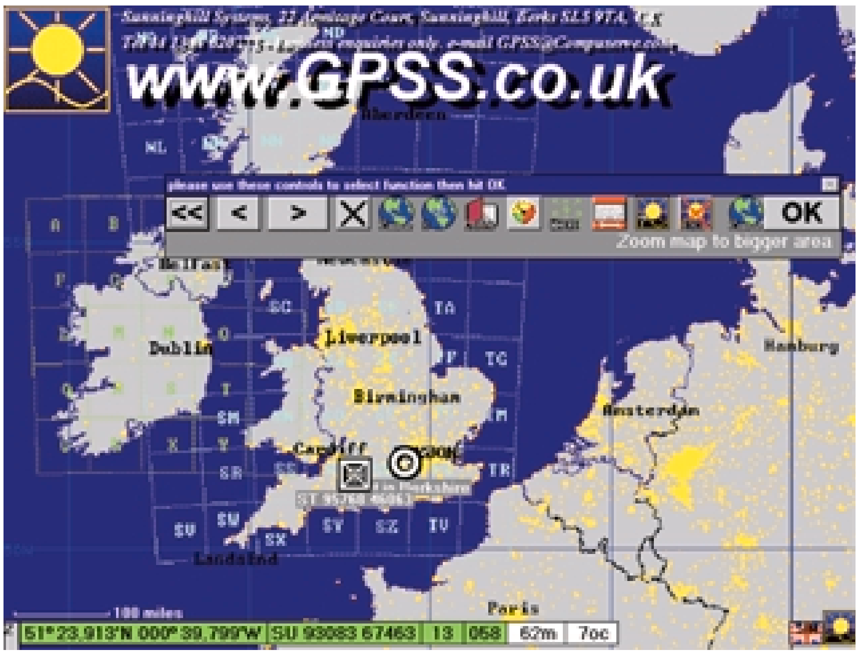


Figure 8. GPSS is a highly recommended freeware program that includes map display.

References:

[1] Bennett, Peter (2000): The NMEA FAQ Version 6.3 (http://vancouver-webpages.com/peter/nmeafaq.txt)
[2] GPS receivers (Figures 3 and 6): www.garmin.de
[3] GPS receivers (Figures 4 and 5): www.unitronic.de
[4] Block diagrams (Figures 1 and 2): www.conexant.com