ELEKTOR ELECTRONICS

FEBRUARY 2004
£3.70

THE ELECTRONICS & COMPUTER MAGAZINE

www.elektor-electronics.co.uk

HANDS-ON CPLDs

complex logic for your own designs

iButton Access Control System

Simple 12-to-230 V Power Inverter

Electronic Valentine Heart

Digital Alarm Clock

DIY RF Inductors

Touch-Controlled Switch

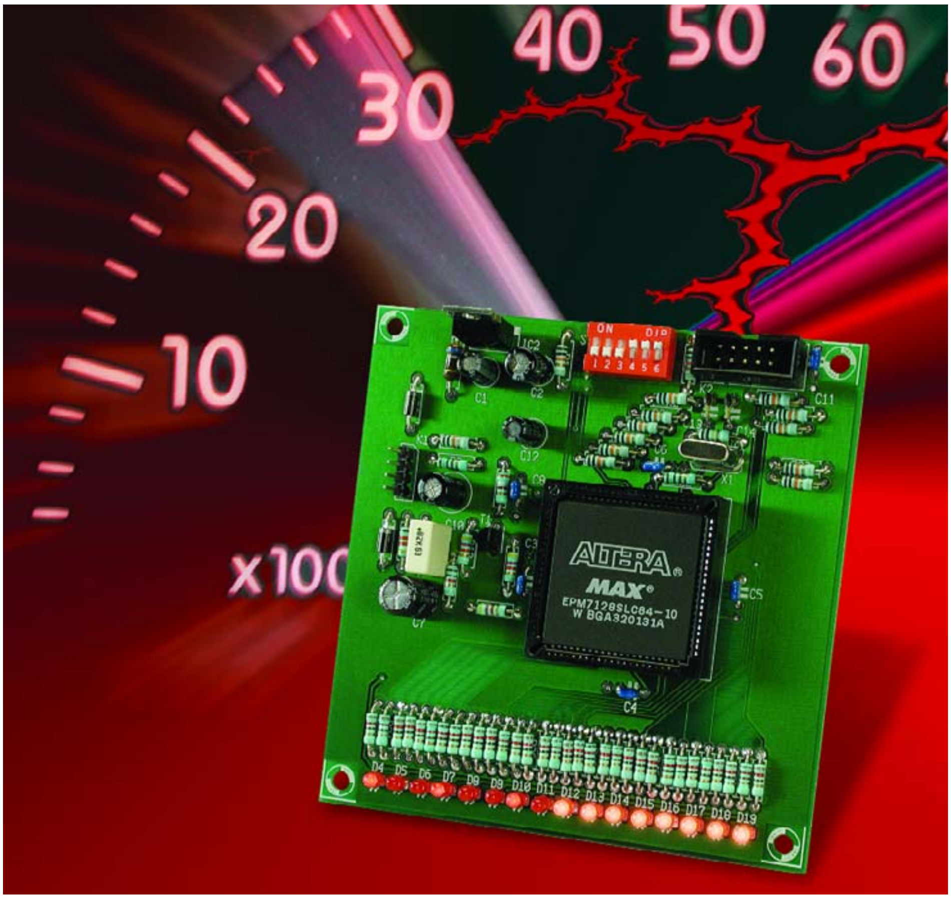
Hands-on CPLDs

Part 1: Experimental rev counter

By A. Rosenkränzer

In the past Complex Programmable Logic Devices (CPLDs) have been something of a closed book to most semi-professional electronics enthusiasts but this hands-on series should go some way to help dispel that myth.

Times change. It wasn't long ago that programmable logic devices (PLDs) were thought to be the preserve of the specialist designer who with the help of expensive development environments and unwieldy software emulators took months to cram large digital designs onto a single PLD. Nowadays the whole process has been simplified to such an extent that the only expensive bit of kit you are likely to need is a PC. An interface adapter is also necessary to connect to the in-circuit programming socket of the CPLD circuit board but its cost is relatively low. When you add to this the fact that the price of the chips themselves have tumbled greatly and the development software is now freely available, these devices start to look attractive even to the hobbyist.



Programmable Logic – An overview

Lift the lid on any piece of modern-day electronic equipment and you will be lucky to find any discrete components from say the 74' family of logic gates used in the design. Occasionally you may see a clock circuit or bus driver but these are essentially used for driving or buffering and do not have a logic function. High volume consumer products such as computer boards have all the logic functions implemented in ASICs (Application Specific Integrated Circuits). The per-unit cost of an ASIC is quite low but there is also the expensive (one off) mask cost to consider. The equipment manufacturer must be sure of the potential market before a commitment to use ASICs in a design is made.

In the equipment development

phase and also for low-volume production ASICs are not cost effective; here the favoured technology is the Programmable Logic Device or PLD. These devices started to appear back in the 80's and since then many more companies have got in on the act to offer bigger and better devices in all sorts of package outline and more recently with lower operating voltages.

The first examples had relatively few registers (flipflops), required a lot of external buffers/drivers and consumed quite a lot of power. On top of this there was also special equipment necessary to program the device. Most of these chips were one-time programmable so that if a mistake was made in programming or some function required modification then the programmed devices could not be reused. On cost alone (development and simulation software together with the programmer device) PLDs were never considered suitable for the hobbyist.

As we said at the beginning times change, PLDs now offer far more gates per package and have become reusable. The real breakthrough came with the 'in system programmable' devices. As the name suggests these chips are fitted permanently into the finished equipment and can be reprogrammed via a 10-pin JTAG interface connector fitted somewhere on-board. The interface allows several devices to be daisy-chained and programmed from just the one connector. The devices do not require a socket so this has led to an increase in the device pin density right up to the BGA package outline. A few manufacturers still offer devices in the more user-friendly PLCC outline which is more suited to a typical hobbyist project.

The necessary development software from practically every chip manufacturer is now freely available for download from the Internet. A simple piece of hardware like the ByteBlaster from Altera is all that you need to connect a PC to the JTAG connector on the PCB. Regular readers will remember the ByteBlaster compatible 'Parallel JTAG Interface' design featured in the September 2002 edition of *Elektor Electronics.*

CPLDs (Complex PLDs) evolved from PLDs and the next step along the way is FPGAs (Field Programmable Gate Arrays). The main difference besides the increased complexity is that CPLDs (mostly using EEPROM technology) do not lose there programming data when the device is powered down.

In contrast FPGAs must be reprogrammed every time they are switched on. Reprogramming is performed automatically from a 'boot device' (either a serial EEPROM or flash memory) fitted to the PCB. The reason for this is that it is not yet possible to fabricate high-speed EEPROM technology at the necessary cell density required for an FPGA.

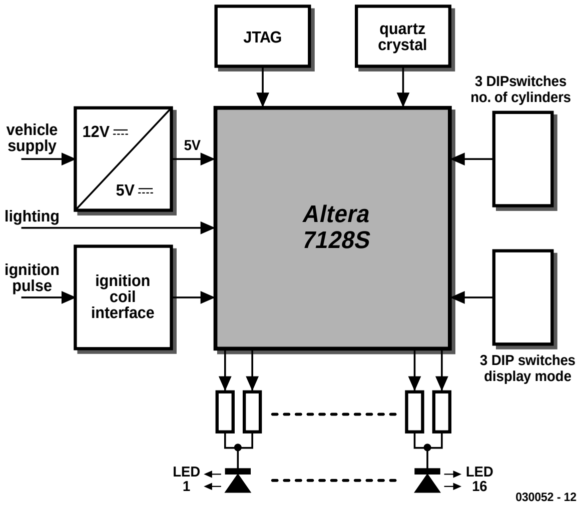


Figure 1. Block diagram of the digital rev counter.

The clock speeds of FPGAs can be several hundred megahertz and some manufacturers are offering SERDES technology that pushes the data rate up to 3.7 Gbps (10 Gbps will no doubt be available by the time you read this) these devices include complete processor cores (ARM, NIOS). Dropping the supply voltage down from 5 V to around 1.5 V has also led to reduced power dissipation in the device.

Around 90% of all FPGA's sold are supplied by three manufacturers: Xilinx, Altera and Lattice. FPGAs have an enormous capacity of gates and can be used in thousands of applications. They also do not have the expensive one-off mask costs of an ASIC.

Hands-on

This article and its follow-up does not concentrate too heavily on a theoretical textbook approach to the use of CPLDs in digital designs instead we use a practical hands-on method. Here we build a digital rev counter (tachometer) suitable for use in most cars or motorbikes.

The rev counter design is quite straightforward and should make a good introduction to the use of CPLD technology. This article takes you through the complete process of construction and programming and will give a good foundation and insights into the advantages of this type of technology.

The project is described using a top-down approach; first we will take a look at the block diagram and point out the features of the CPLD at the hardware level. Next we will consider the peripheral circuitry and finally (in the second part) the internal structure of the CPLD and the programming process is described.

Figure 1 shows the block diagram of the digital tachometer circuit. The central component is the CPLD 7128S from Altera, this chip does all the work and is surrounded by some peripheral circuitry.

An interface is required to convert the ignition pulse from the low-tension connection on the ignition coil to TTL levels and remove any noise spikes that may be present on this signal. The 'lighting' input signal controls the light intensity of the LED column (the 12 V to 5 V interface is not shown on this diagram but can be seen in detail in **Figure 2**). Under daylight viewing the intensity can be at maximum but at night it needs to be dimmed to avoid dazzling.

The chip is programmed over the JTAG interface using a ByteBlaster interface adapter from Altera (or a ByteBlaster compatible interface) which connects to a PC parallel port.

A quartz crystal together with a few discrete components generates the chip clock. This clock is sufficiently accurate to ensure that the circuit will not require calibrating. The setting of three DIP switches allows the tachometer to be used on vehicles with different cylinder counts and ignition systems. Without reprogramming the circuit can be configured for one, two, three, four, six or eight cylinder engines using either single or double spark ignition. Three more DIP switches are used to change the display mode. This allows the range of displayed revs to be changed to suit the performance of the vehicle engine:

1000 to 6000 rpm with a resolution of 3 LEDs per 1000 rpm.

750 to 4500 rpm with a resolution of 4 LEDs per 1000 rpm (Diesel mode).

2000 to 10000 rpm with a resolution of 2 LEDs per 1000 rpm (motorcycle engines).

4000 to 6000 rpm high resolution mode with 8 LEDs per 1000 rpm.

5000 to 7000 rpm (the so called 'Schumacher mode').

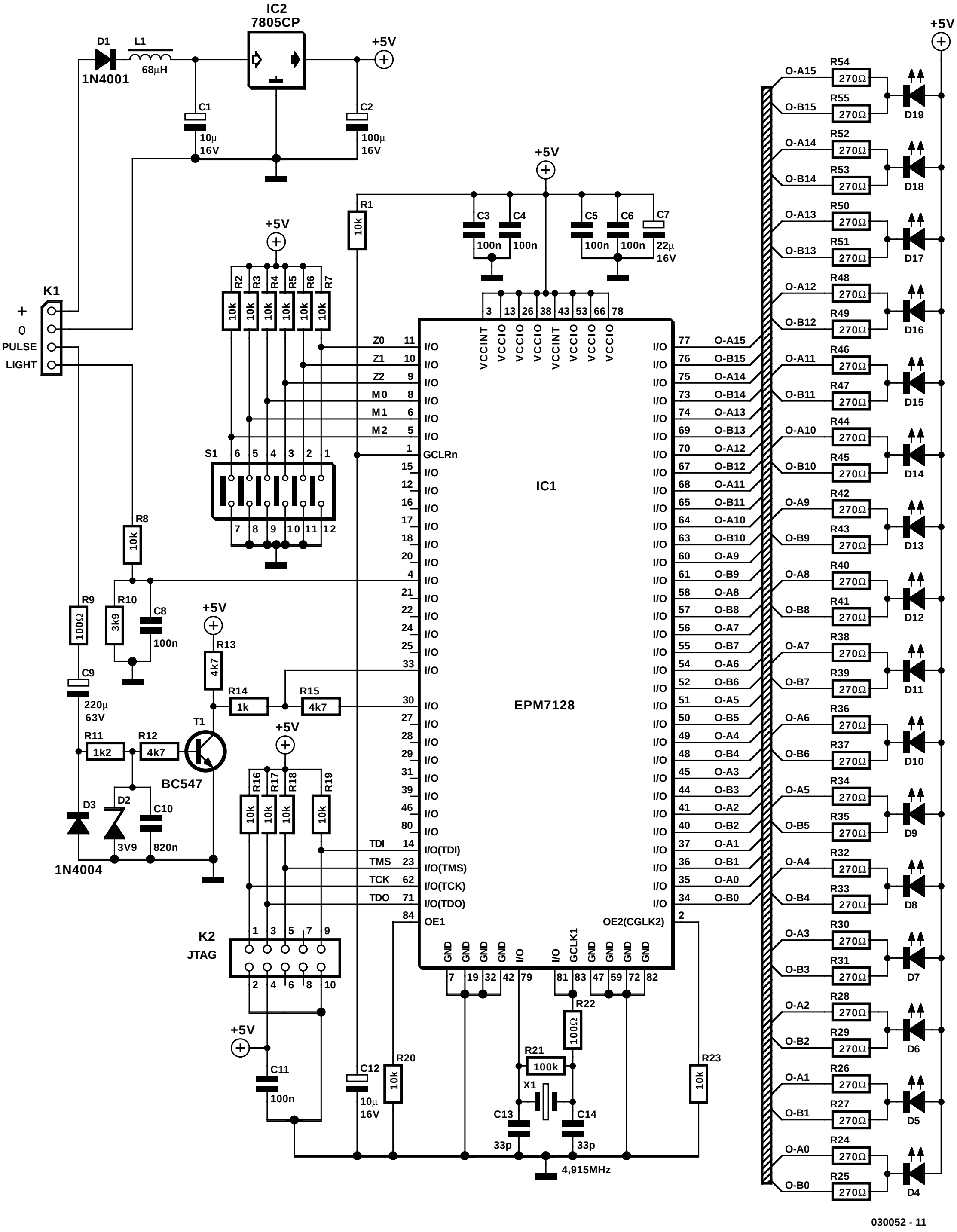


Figure 2. The complete circuit.

Almost any type of display representation including non-linear can be programmed into the CPLD and selected by three DIP switches. To make programming easier there is an Excel spreadsheet available that calculates the frequency scaling necessary in the program given the oscillator frequency and the required display resolution.

The tachometer display consists of a single line of 16 LEDs. Each LED is driven by two outputs from the IC and through two series resistors. The reason for this is that the data sheet indicates that the maximum output current is limited to 12 mA; most standard LEDs require around twice this current if they are to be visible under daylight conditions. Low power LEDs could be used to solve the problem but here the use of two outputs to drive one LED gives us a simple method to indicate not just the rpm but also each 1000-rpm graduation that glow differently.

The peripheral circuitry

Figure 2 shows the detail circuit diagram of the tachometer. IC1 is the CPLD type EPM7128S from Altera packaged in a 84-pin PLCC outline. The chip has a total of eight ground pins that must all be connected to GND.

There are also two types of VCC connection on the chip. Two of these connections (both called VCCINT) are connected to the core of the chip while the other six (VCCIO) supply power to the I/O buffers. This supply can be connected to either 3.3 V or as in our case 5 V. For other applications it would be useful to be able to use 3.3 V if it was found necessary to reduce the EMI produced when the buffers switch or it would greatly simplify interfacing to 3.3 V logic.

For supply decoupling it is good practice to use at least one 100 nF capacitor for each VCC connection sited as close as possible to the pin and one or two tantalum capacitors (10 to 22 μF) to reduce any supply rail noise.

Four connections are used for the JTAG programming interface (TDI, TMS, TDO and TCK). TCK is the clock input used during programming. TDI is not as you may have thought some kind of diesel engine but in this case the data input signal. This signal together with the data output (TDO) allows several CPLDs to be daisy chained by connecting TDO to TDI of the next chip. The TDO of the last chip is then connected to the JTAG interface. TMS is used to configure the JTAG interface for reset, data, command, boundary scan and so on. All four pins are fitted with a 10 kΩ pull-up resistor.

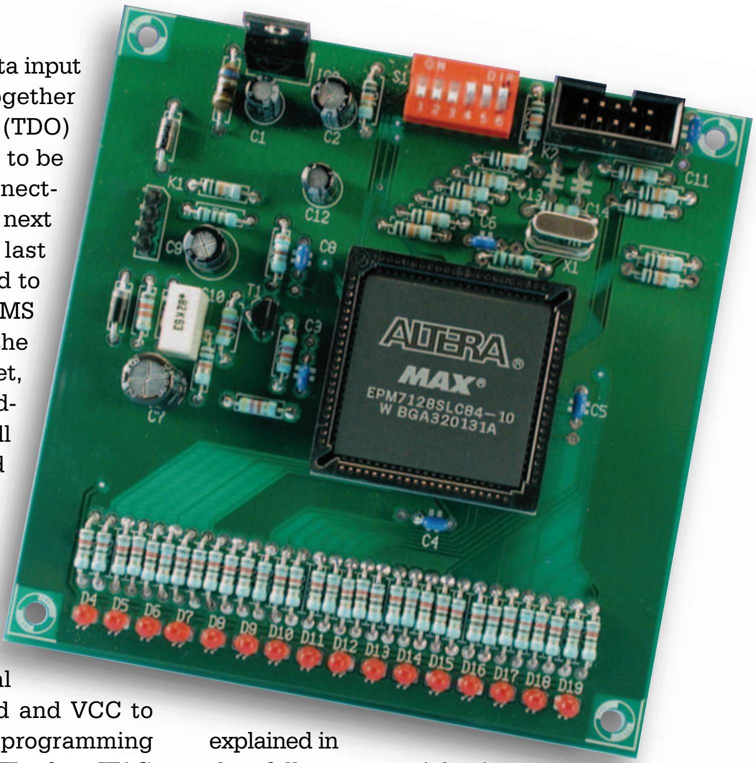
The connector for the JTAG-Interface (K2) is a 10-way header plug. Along with the four signal wires are also ground and VCC to supply power to the programming adaptor (ByteBlaster). The four JTAG connections can also be used as general-purpose I/O pins but this is only advisable if there are no other I/O pins available.

R1 and C12 provide a reset pulse to the active low Global CLeaR input (GCLRn) during power-up. This reset clears all the chips internal registers.

GCLK1 is the first of the two global clock inputs. It is important to ensure that the first system clock for the CPLD is connected to this input and not to some general purpose I/O pin which is not internally wired to the cells and registers. Two I/O pins close to the GCLK1 input are used as a buffer to interface to the crystal. The output from the buffer is connected directly to GCLK1 input. Resistor R22 reduces the output loading while the 100 KΩ resistor R21 in parallel to the crystal provides a positive feedback path for the buffer.

Pin OE1 is a general Output Enable, and must be defined in the programming data. OE2 (GCLK2) is another special function pin and can be used as either a second clock input (for designs with two clocks) or as a second output enable input. In the tachometer design described here both of these functions are not required.

All other components and external signals are connected to general-purpose I/O pins. The criteria used to allocate these pins will be explained in the follow-up article in *Elektor Electronics.*



The remainder

Power for the circuit is supplied by the 12 V vehicle battery via connector K1. Diode D1 prevents damage to the circuit if the supply polarity gets mixed up. L1 filters noise spikes on the supply to the 5 V regulator IC2 while C1 and C2 are also used to provide noise decoupling. The complete circuit takes about 0.5 A worst case so power dissipation in the voltage regulator is given by:

$$P_V = 0.5\ \text{A} \cdot (14.4\ \text{V} - 5\ \text{V}) \approx 5\ \text{W}$$

It is important to ensure that the heatsink for IC2 can dissipate this amount of energy.

The six way DIP switch S1 is used to select the vehicles engine cylinder count and the display mode. When a switch position is open circuited a 10 kΩ resistor pulls the line up to 5 V. R8 and R10 reduce the voltage on the 'light on' signal to TTL levels while C8 reduces any noise spikes on this input.

The signal from the ignition coil is conditioned by the circuitry around transistor T1, it produces a filtered TTL compatible output signal for IC1 from the 'raw' ignition signal. C9 AC couples the signal. The negative transition of the pulse is clamped to -0.7 V by diode D3 while R9 protects D3 from excess power dissipation. R11 and zener diode D2 limit the positive level of the pulse to 4 V. C10 filters out high frequency noise while R12, T1 and R13 form a transistor switch which produces a clean TTL compatible output pulse to the IC from the noisy ignition signal.

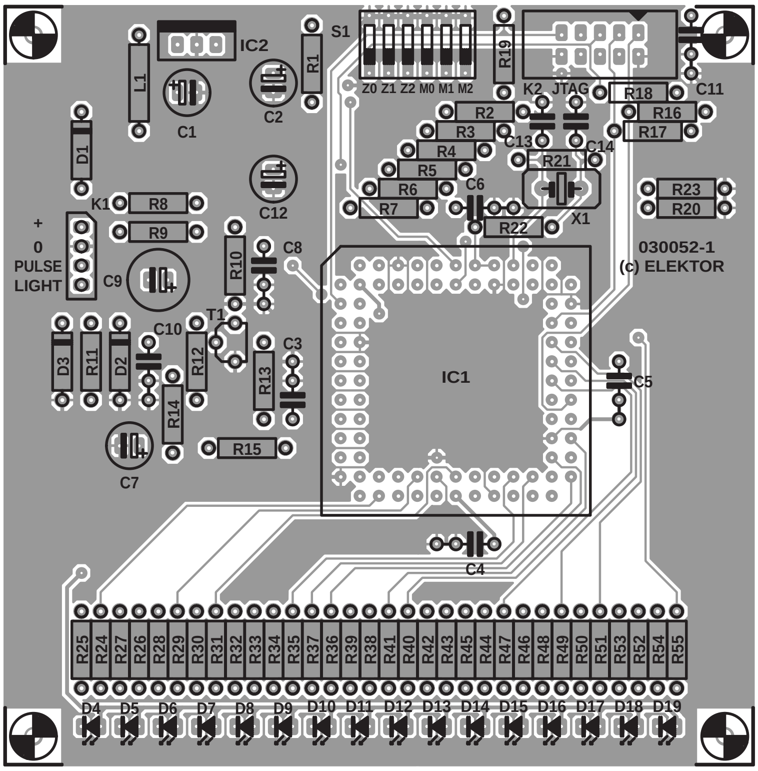

IC2
S1
R1
L1
C1
C2
R19
Z0 Z1 Z2 M0 M1 M2
K2 JTAG
C11
R18
R16
R17
R2
R3
R4
R5
R6
R7
C13
C14
R21
C6
X1
R22
R23
R20
D1
K1
R8
R9
C12
C8
+
0
PULSE
LIGHT
C9
R10
030052-1
(c) ELEKTOR
T1
C10
D3
R11
D2
R12
R13
C3
IC1
C5
R14
C7
R15
C4
R25 R24 R27 R26 R28 R29 R30 R31 R32 R33 R34 R35 R37 R36 R39 R38 R41 R40 R42 R43 R45 R44 R47 R46 R48 R49 R50 R51 R53 R52 R54 R55
D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17 D18 D19

Figure 3. Component mounting plan for the double-sided PCB.

COMPONENTS LIST

Resistors:
R1-R8,R16-R20,R23 = 10kΩ
R9,R22 = 100Ω
R10 = 3kΩ9
R11 = 1kΩ2
R12,R13,R15 = 4kΩ7
R14 = 1kΩ
R21 = 100kΩ
R24...R55 = 270Ω

Capacitors:
C1, C12 = 10µF 16V radial
C2 = 100µF 16V radial
C3-C6,C8,C11 = 100nF
C7 = 22µF 16V radial
C9 = 220µF 16V radial
C10 = 820nF
C13,C14 = 33pF

Semiconductors:
D1,D2,D3 = 1N4001
D2 = zener diode 3V9, 500 mW
D4...D19 = LED, green, 3 mm
IC1 = EPM7128SLC84-15
IC2 = 7805CP
T1 = BC547A

Miscellaneous:
K1 = 4-way SIL pinheader
K2 = 10-way pinheader
L1 = 68µH choke
S1 = 6-way DIP switch
X1 = 4.9152MHz quartz crystal
PCB, available from The PCBShop
Disk, order code **030052-11** or
Free Download

Web Addresses

www.xilinx.com
www.coolpld.com

www.altera.com

www.latticesemi.com

Resistors R14 and R15 control the hysteresis of a Schmitt trigger input. This is quite important because any noise on the ignition input signal would cause the tachometer to give a false reading. R15 provides the feedback from the output side of the input buffer. When the buffer changes state a small proportion of its output is fed back out from pin 30 on IC1 to reinforce the input signal via R15. This has the effect of shifting the voltage threshold at which the buffer can switch back (hysteresis) and so masking the effects of noise on the input.

Free Downloads

Excel spreadsheet.
File number: 030052-11.zip
PCB layout in PDF format.
File number: 030052-1.zip

www.elektor-electronics.co.uk/dl/dl.htm,
select month of publication.

The PCB and future developments

A double-sided PCB has been produced specifically for this tachometer project. The PCB and component placement diagram is shown in **Figure 3**. Component mounting should be fairly straightforward but take care not to allow any solder debris to enter underneath the PLCC socket where it could short between PCB tracks. Correct orientation of the IC is important so ensure that the clipped corner of the package outline corresponds to the corner of the socket and silk-screen printed IC outline on the PCB. Ensure correct polarity of the LEDs, capacitors and diodes when they are fitted to the board.

When any circuit uses a line of discrete LEDs the finished project always looks a bit amateurish if the LEDs are not uniformly in-line. A good tip here is to firstly solder just a single leg of each LED in place. The board can now be flipped over and all the LEDs carefully bent or re-soldered into exactly the right position before the second leg of the LEDs are soldered.

The layout is not too cramped so there is no reason why this PCB could not also be used as a development platform to prototype other CPLD designs. Those of you however who are looking for the flexibility of a true evaluation board should not be disappointed by our EVA-board that we shall be featuring in a forthcoming article in *Elektor Electronics*.

(030052-1)

In the second part of this article we will take a closer look at the CPLD internals and device programming.

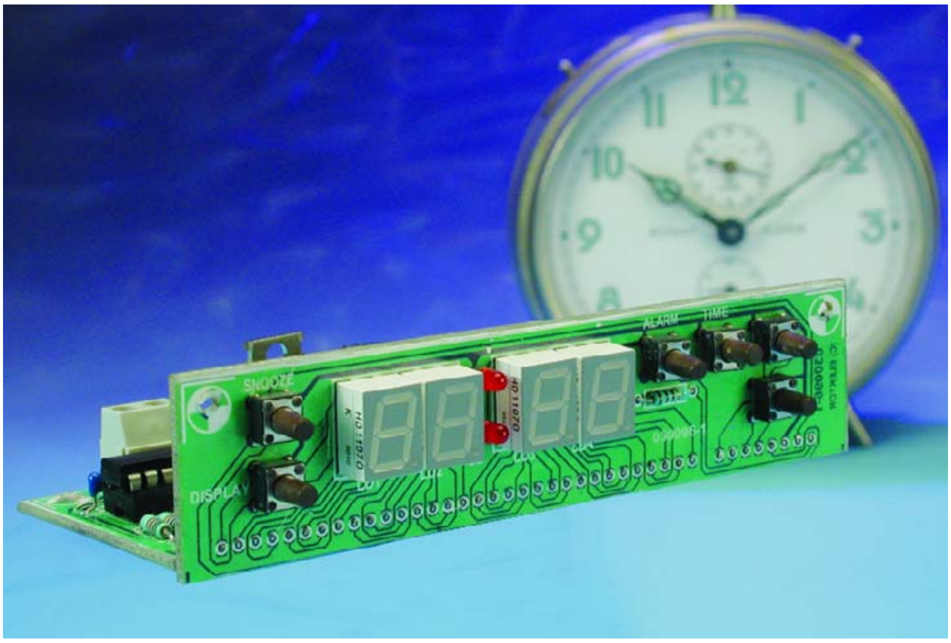
Digital Alarm Clock

Based on a PIC micro

Design by M. Conde de Almeida

maconde@terra.com.br

Although digital alarm clocks have been around for years, most of today's off the shelf products suffer from serious design limitations. For example, many don't keep track of weekdays and can only store one alarm time. High time for a home-brew design that does a better job.



Off the shelf alarm clocks present a serious limitation if, for instance, you and your partner have different wake-up times or if you have to give or take medication at regular intervals. This limitation gets more serious if you want alarms to go off only on specific days of the week or only during work days. The time setting process on most clocks is also subject to improvement. In the majority of cases you adjust time by incrementing minutes and hours. If your clock is at, say, 06:15h and the correct time is 20:58h you need to keep a key pressed for quite some time and release it well before the desired hour/minute, switching to 'slow' setting. If not, you're past the desired time and have to start all over again.

This article proposes a solution for these design weaknesses by discussing a low-cost circuit based on the popular PIC 16F84A microcontroller from Microchip.

Our digital alarm clock keeps track of weekdays and has eight alarms that can be individually set to go off every day, only on working days or, if you want, on a specific day of the week.

The time setting process allows the adjustment of each digit of the clock separately by means of Up and Down (+ and –) keys. It also incorporates some other interesting features.

Like all good digital alarm clocks our project has a battery that keeps the clock ticking in the event of an AC power failure. In battery-powered mode the display is turned off to reduce energy consumption. However, if you want to check the time you may still enable the display by pressing a 'display on' button. In battery mode the alarms will continue to operate normally. When an alarm goes off the display will be turned on to show the current time.

The eight different alarm times are kept in the PIC's EEPROM. So, you won't have to adjust any of the alarms again even if the battery fails. A 'snooze' key will temporarily turn an alarm off. The alarm will be triggered again after a minute until it is definitely turned off. A master Alarm On/Off key will enable/disable all the alarms, independent of their individually set states. A bright LED display makes it easy to check the time from a distance or in the dark.

The 4-MHz crystal oscillator guarantees a pretty accurate timebase for the clock (error = 0.000427%).

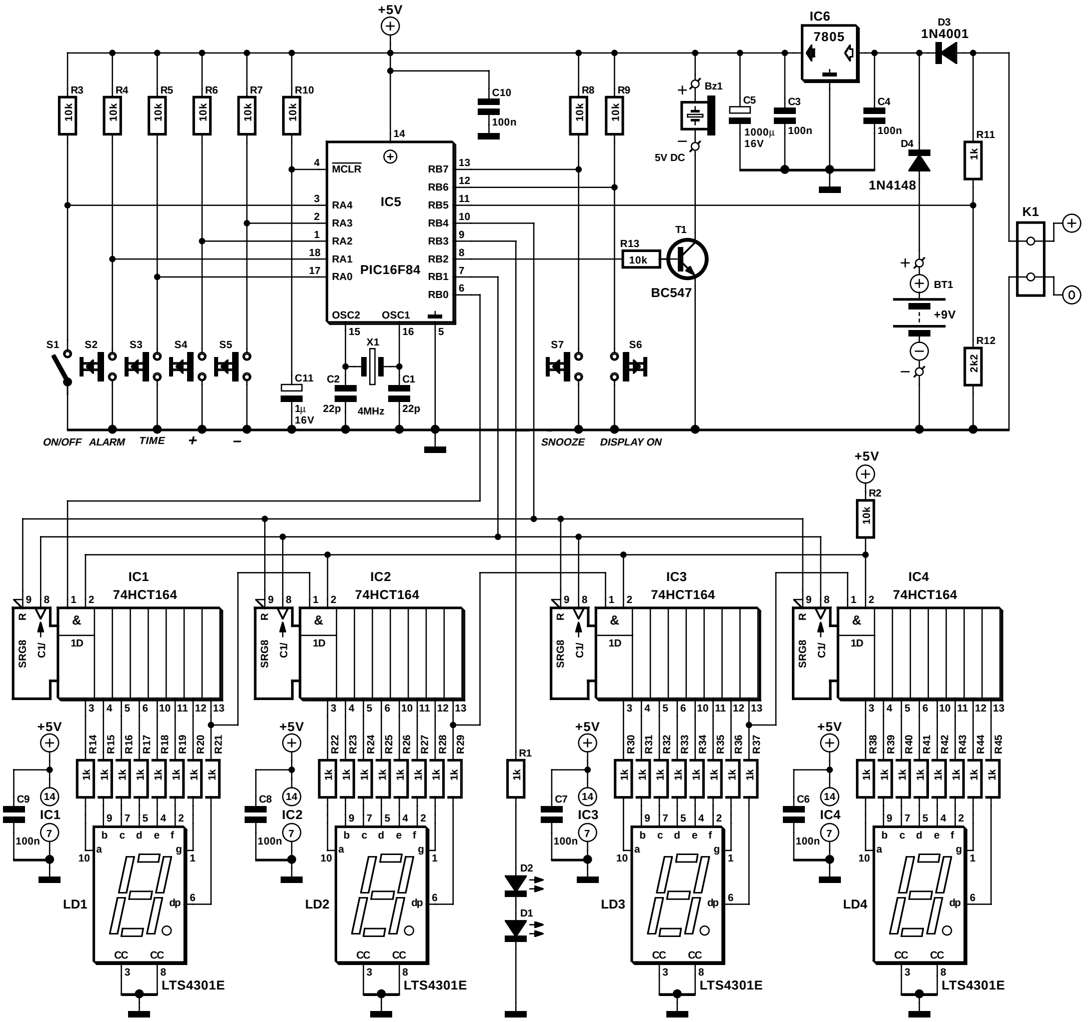


Figure 1. Circuit diagram of the PIC-based Digital Alarm Clock.

The hardware

Figure 1 shows the complete circuit diagram of the clock. All intelligence (and a lot of logic) is vested in the PIC16F84 MCU in position IC5. Using RA0-RA4 and RB5-RB7 as input port lines and RB0-RB4 as output port lines, a fair amount of executable code run from the on-chip memory is able to take total control of the circuit, requiring just a 5-V supply voltage and a clock signal generated with the aid of an external 4-MHz quartz crystal, X1.

Components R10 and C11 guarantee that the microcontroller is reset at power-on.

Switches S1-S5 are connected to the microcontroller's PORTA pins (all configured as inputs). Resistors R3-R7 guarantee a High logic level at the PORTA pins when the switches are open. When closed, these keys will force a LOW state on the PORTA pin they are connected to, triggering the execution of specific clock control routines.

PORTB.6 and PORTB.7 are also configured as inputs. R8 and R9 guarantee a High logic level at these inputs when the associated switch S6 or S7 is open. S7, when closed, will activate the 'Snooze' function. Similarly, S6 will temporarily activate the display. This function is only available when the clock is operating in the battery mode.

Resistors R11 and R12 form a voltage divider fed by the main 12 VDC source. They guarantee a High logic level at PORTB.5 when the main supply voltage is available. In this situation the display will be permanently on. If for some reason the main power is not available (AC power failure) there will be a Low logic level at PORTB.5 and the entire display will be turned off by the software.

Looking at the output devices controlled by the PIC, the alarm buzzer Bz1 is driven via PORTB.2 while PORTB.3 controls LED1 and LED2, the Hour/Minute separator in the readout.

Four common-cathode 7-segment displays, LD1-LD4 constitute the clock readout. These displays are driven by four 74HCT164 shift registers (IC1-IC4) connected in series. R2 only serves to guarantee a High level to the enable pins of the shift registers. PORTB.1 drives the CLOCK pin (8) of the shift registers while PORTB.0 drives the DATA line of the first shift register in the chain (IC1, pin 1). The microcontroller program will clock 32-bit strings into the shift registers whenever a display update is necessary. PORTB.4 controls the RESET pin of the shift registers. This port line will be held Low when the display is to be turned off.

The regular 12 VDC power may be supplied by a standard AC or DC adapter. Diodes D3 and D4 will guarantee exclusive operation of either the 9 V battery or the 12 VDC power source. They will also guarantee that R11 and R12 are only powered by the 12 VDC source.

The 5 V supply voltage for the logic circuitry is provided by a 7805 regulator (IC6). Capacitors C3, C4, C5 and C6-C9 help to keep the supply voltage as clean as possible.

Current consumption of the clock in normal use (i.e., powered by the mains adapter) is of the order of 75 mA. The current from the 9-V backup battery amounts to about 5 mA.

Control software

The entire clock program was written in the PIC Assembly Language using the MPLAB Integrated Development Environment (v. 5.70.40) supplied free of charge by Microchip.

The source code and Hex files containing the program ready to be flashed into the PIC microcontroller can be obtained free of charge from the Publisher's website, see 'the Free Downloads' inset. The file number is **030096-11**. For those without access to a PIC programmer, the microcontroller is also available ready-programmed under number **030096-41**.

The flowchart in **Figure 2** summarizes the operation of the program. After an initialisation routine where I/O pins, Interrupt register, Timer_0/Prescaler operation are configured and alarms information is read from the EEPROMs into the PIC data memory, the program enters a loop in which control keys are read and, based on their status, a specific branch is taken.

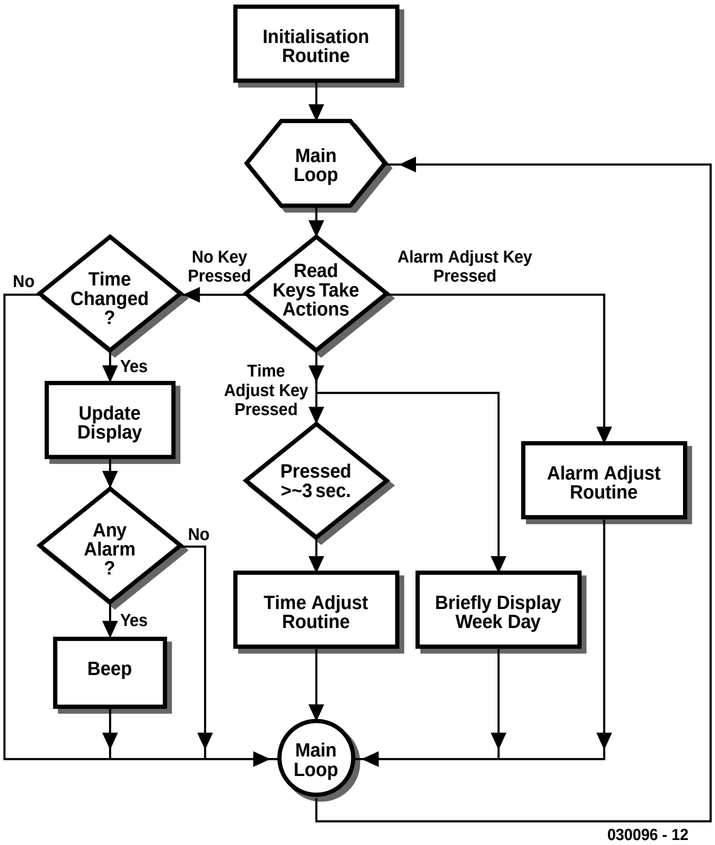
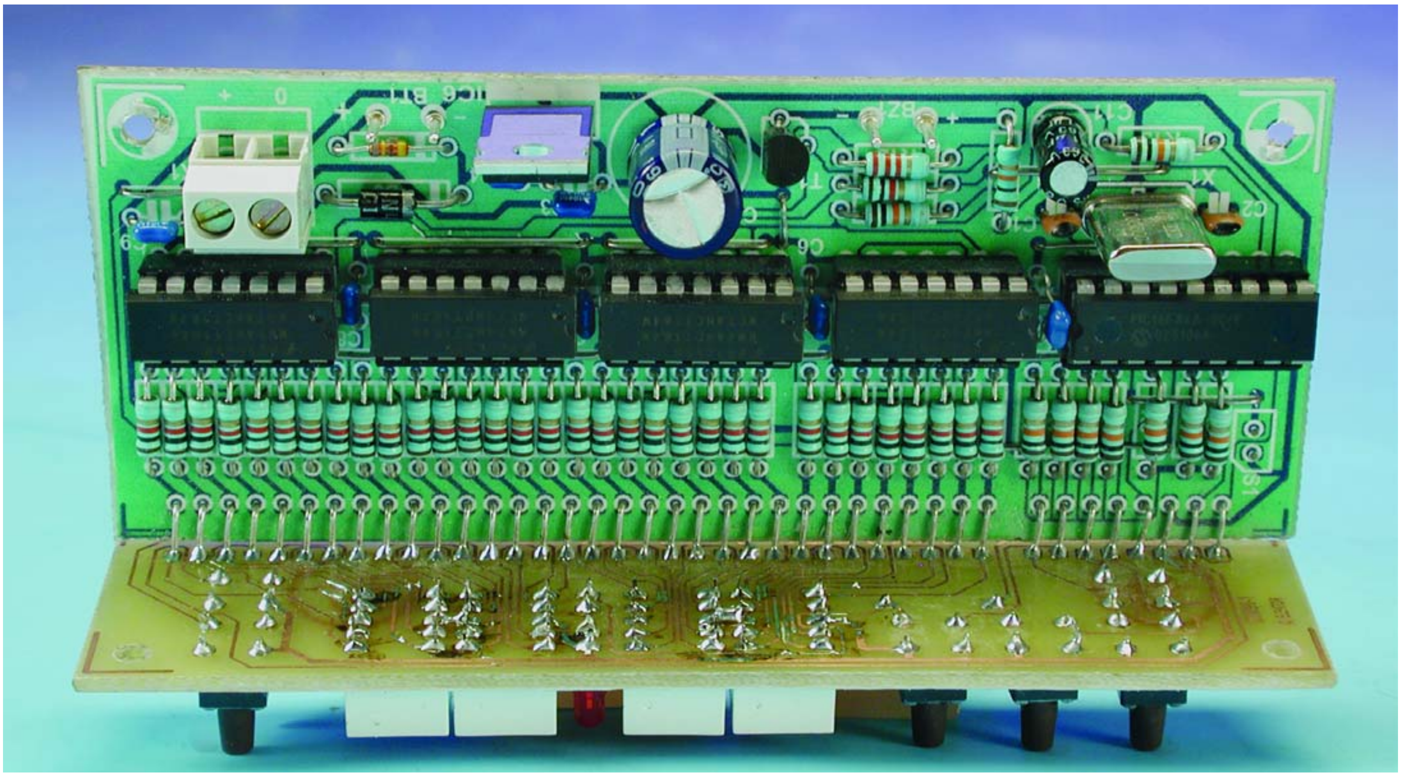


Figure 2. Flow chart of the program run by the PIC.

For example, if the Alarm Adjust key is pressed and held down longer than about 3 seconds, the program will execute the Alarm Adjust Routine and, once finished, return to the main loop.



COMPONENTS LIST

Resistors:
R1, R11,R14-R45 = 1kΩ
R2...R10,R13 = 10kΩ
R12 = 2kΩ2

Capacitors:
C1,C2 = 22pF
C3,C4,C6-C10 = 100nF
C5 = 1000µF 16V radial
C11 = 1µF 16V radial

Semiconductors:
D1,D2 = LED, 3mm, red, low-current
D3 = 1N4001
D4 = 1N4148
T1 = BC547
IC1-IC4 = 74HCT164
IC5 = PIC16F84-04/p, programmed, order code **030096-41**
IC6 = 7805

Miscellaneous:
K1 = 2-way PCB terminal block, lead pitch 5mm
S1 = on/off switch
S2-S7 = miniature PCB mount pushbutton, type DTS65N
LD1-LD4 = LTS4301E (Lite-On)
BZ1 = 5VDC buzzer (active)
X1 = 4MHz quartz crystal
BT1 = 9V battery with and clip-on lead
PCB, available from **The PCBShop**
Disk, PIC source and hex (object) code files, order code **030096-11** or Free Download

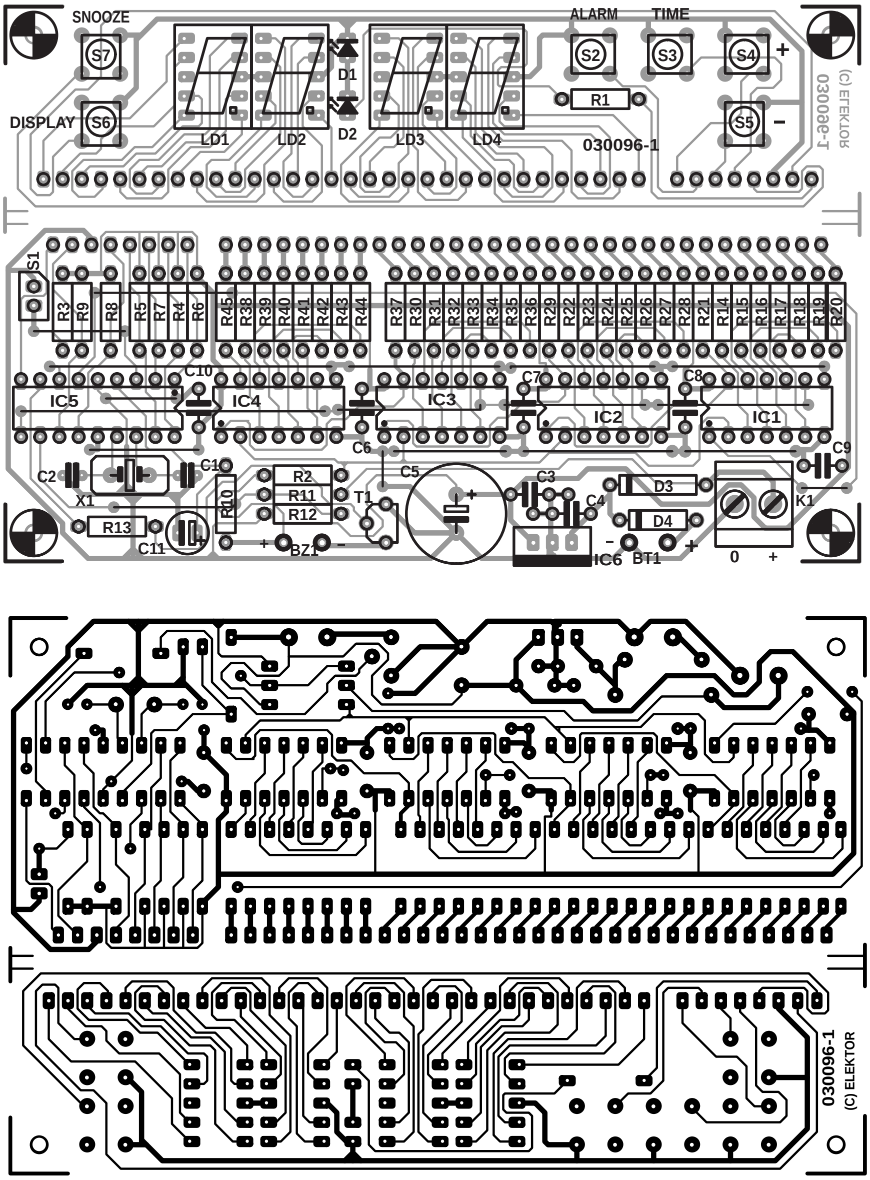


Figure 3. Copper track layout and component mounting plan of the PCB designed for the clock (board available from The PCBShop).

If the Time Adjust key is held depressed for less than about 3 seconds, the display will show the current weekday. If the key remains pressed longer than about 3 seconds, the program will execute the Time Adjust routine and, once time and weekday are adjusted, return to the main loop.

When no key is pressed the program keeps updating the display, checking the current time against the alarm entries in memory and triggering the beeper if a matching times are found.

Although it has not indicated by flowchart, the display updating process is affected by the status of the PORTB.5 line (12 VDC supply monitor) while the alarm verification and triggering processes are subject to the status of Alarm On/Off and Snooze keys.

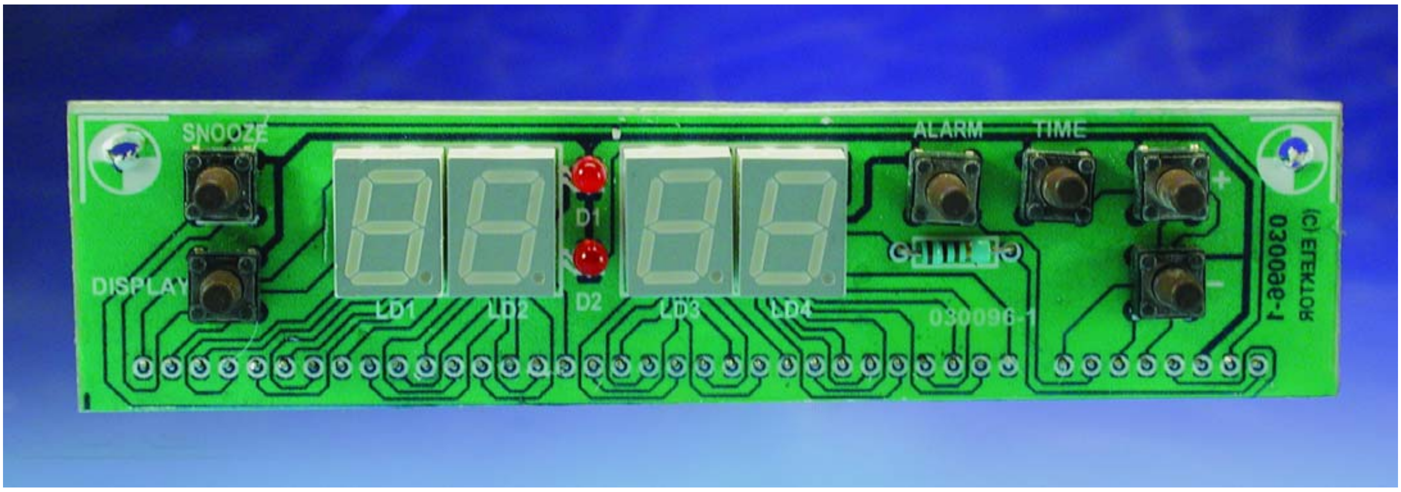
One important piece of the software is the time counting process. It is based on the microcontroller TIMER_0 which has the ability to interrupt the main program and deviate its execution to an interrupt handling routine whenever its count overflows (FFh → 00h).

The microcontroller is configured so that the TIMER_0 overflow will take place every 2.048 ms. This is accomplished by setting the prescaler to divide the TIMER_0 clock by eight.

Whenever called, the interrupt handling routine, will increment a 16-bit counter. This counter has a maximum count of 29,297 (7271h). This means that every minute (29,297 × 2.048 ms = 60,000.256 ms or very close to one minute) it will overflow.

The overflow of the 16-bit counter increments the minute-counter (modulo-10) which, after overflowing, increments the tens-of-minutes counter (modulo-6) which increments the hour-counter and sure it will finally increment the tens-of-hours counter.

As shown above, the clock's minute is slightly longer than it should be. This error, amounting to 0.000427%, will cause a time difference of less than two minutes per year which is pretty low compared to what is achieved by most the off-the-shelf clocks.



Construction

The artwork of the PCB designed for the clock is shown in **Figure 3**. The PCB has to be cut in two to separate the main board from the display board.

Start by soldering the wire links onto the main board. Then follow the five IC sockets and all low-profile components. The rest of the construction should be mostly plain sailing as only regular components are used. As always, take care with the polarity of electrolytic components like capacitors, diodes and (yes!) the displays. The display board is secured to the main board at an angle of 90 degrees. The interconnections between the boards are made using angled SIL pin-headers or 40 short pieces of stiff, bare wire. The pushbutton actuator rods will protrude a little above the display tops to make them accessible from the outside once the PCB assembly has been mounted behind the front panel. A red bezel may be used as a finishing touch to enhance the appearance of the clock.

Using a multimeter, make sure that there's no short circuits between the +5 V supply rail and ground.

When inserting the integrated circuits always check the correct 'pin 1' position. Finally, avoid ESD damage to your circuit by keeping yourself grounded by means of wrist straps.

Clock operation

The Time adjustment is shown pictorially in **Figure 4**. Other drawings are available but not printed here for lack of space. They include:

- clock display;
- keys and their functions;
- alarm on/off key operation;
- buzzer control procedure;

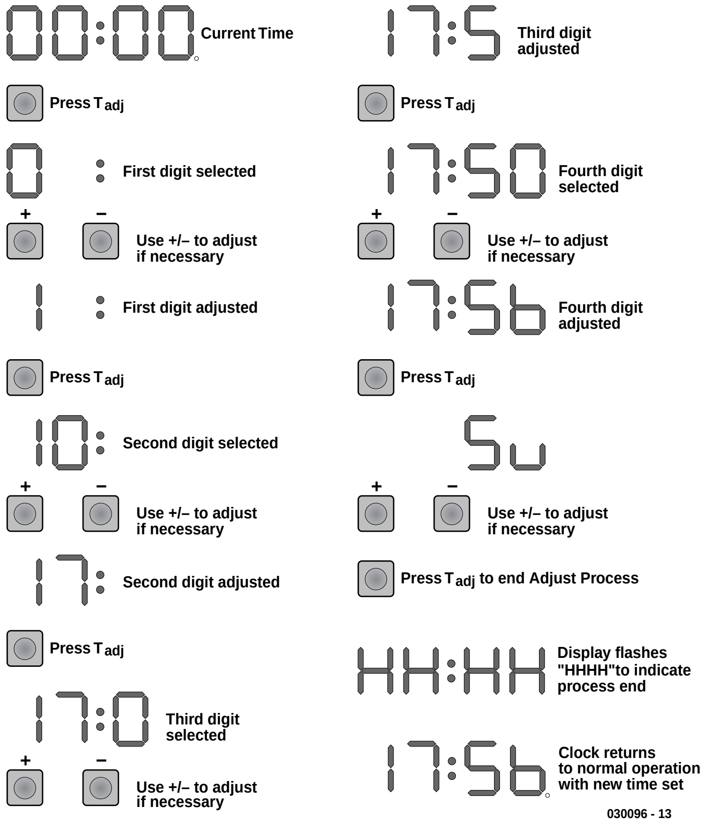


Figure 4. Time adjustment procedure.

- operation in battery mode (mains power failure);
- alarm adjustment,

The pictures are supplied in the form of a (zipped) pdf document — see this month's Free Downloads, number **030096-12**.

(030096-1)

Free Downloads

PIC source and hex (object) code. File number: **030096-11.zip**
Clock operation pictograms. File number: **030096-12**
PCB layout in PDF format. File number: **030096-1.zip**
www.elektor-electronics.co.uk/dl/dl.htm, select month of publication.

DIY RF Inductors

The ABCs of inductors

By B. Kainka

Electronics hobbyists often wish to copy a circuit for which suitable coils or fixed inductors are not readily available. However, you can wind just about any type of inductor if you only know how. Or you can take inductors from old equipment and modify or adjust them. All you have to do is determine is how many turns you need.

Low-value inductors are primarily used in RF circuits. A general distinction must be made between inductors with magnetisable cores (made from ferrite or iron) and 'air-core' inductors, which are wound on insulating forms or entirely without any sort of coil form.

Air-core inductors

Let's first turn our attention to air-core inductors. **Figure 1** shows an example of an inductor for a short-wave resonant circuit, which has 20 turns, a diameter of 16 mm and a length of 35 mm. It has an inductance of around 3 μH, and with a variable capacitor having a maximum value of 300 pF it has lower frequency limit of approximately 5.3 MHz. How can this be calculated? Read on to learn more… (and by the way, there's also a simple utility program to make things easier).

For a 'long' inductor with $l > D$ and n turns, a cross-sectional area A in m^2 and a length l in m, the following relationship generally holds true:

$$L = (\mu_0 \times n^2 \times A) \div l$$

where μ_0 is the *magnetic constant* or *permeability of free space* and has a value of

$4\pi \times 10^{-7}$ henry/metre
1.2466×10^{-6} henry/metre.

Although this formula is strictly true only for infinitely long inductors, it can be used as a satisfactory approximation for inductors with lengths down to $l = D$.

For an inductor with a given number of turns, the magnetic coupling between the individual turns increases as the length of the inductor decreases, which yields a greater inductance. By reverse token,

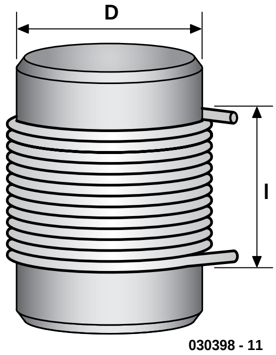


Figure 1. Structure of an air-core inductor.

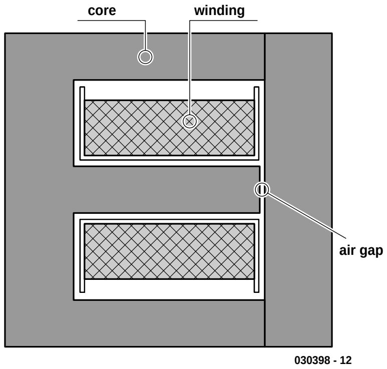


Figure 2. Structure of an inductor with an E-I core.

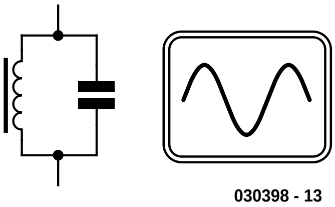


Figure 3. A inductor in a resonant circuit.

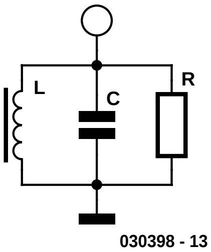


Figure 4. A resonant circuit with a loss resistor.

increasing the spacing between the turns of an inductor decreases its inductance, and this is sometimes used to tune inductors.

For inductors having a circular cross section, the above formula can be simplified to the following approximate formula, where the diameter D and length l of the coil are given in millimetres:

$$L = n^2 \times D^2 \div l \quad \text{[nH]}$$

This formula includes the approximation $\pi^2 \approx 10$, which introduces a small error (approximately 1.3 %). Extreme accuracy should anyhow not be expected, since the inductance depends in part on the shape of the coil, particularly the ratio of its length and diameter, as well as the thickness of the wire and even its surroundings. Consequently, for many purposes it is adequate to be able to calculate the inductance of an air-core inductor within a 10 percent tolerance margin.

Inductors with cores

RF inductor forms with threaded ferrite cores are often used in practice. The core increases the inductance, typically by a factor of four or even more. The inductor can be tuned by adjusting how far the threaded core is screwed into the inductor. Ferrite cores are made for specific frequency ranges, within which they exhibit small energy losses.

Significantly higher inductances can be obtained using closed cores with or without air gaps. Although an air gap reduces the inductance, it allows a higher magnetisation levels to be used, since it prevents the core from becoming magnetically saturated even at high currents. The types of cores commonly used are toroidal (ring) cores, E-I transformer cores (**Figure 2**) and closed pot cores.

With such cores, the inductance strongly depends on the material used and the geometry of the core, as well as the number of turns. This means that it is not possible to give a general formula for calculating the inductance, as with air-core inductors. Instead, manufacturers state an 'A_L value' in nH/n^2 for each core, such that

$$L = A_L \times n^2 \quad \text{[nH]}$$

For example, an Amidon T37-2 ring core has an A_L value of 40 nH/n^2. If you wind a coil of 10 turns on such a core, you will obtain an inductance of L = 4000 nH = 4 µH.

Ring-core inductors, like air-core inductors, are suitable for building RF resonant circuits. Besides the A_L value, the design frequency range of the core is also important. Amidon Type xxx-2 cores (with red marking) are suitable for frequencies up to 30 MHz. The calculation program, which is described in more detail below, can be used to quickly determine the inductance of air-core inductors and inductors with known A_L values.

Resonant circuits

Although resonant circuits are the most important application for inductors, resonant frequency and damping are also significant when inductors are used for other purposes. For one thing, it's important to recognise undesired resonances, and for another thing, it's very easy to determine the value of an unknown inductor by using frequency measurements.

If an inductor and a capacitor are connected together as shown in **Figure 3**, the result is a resonant circuit. Electrical energy can 'swing' back and forth between the inductor and the capacitor, similar to the motion of a pendulum, and such a circuit has a characteristic resonant frequency. After being excited by a short current pulse, a resonant circuit will oscillate freely at a frequency given by the formula

$$f_0 = 1 \div [2\pi \times \sqrt{(LC)}] \quad \text{[Hz]}$$

Resonant circuits are often used in circuits where several different frequencies are present and in frequency mixers. This allows currents and voltages to be distinguished according to their frequencies. A parallel resonant circuit has a complex impedance Z whose peak value occurs at the resonant frequency f_0. At this frequency, $R_C = R_L$, and the currents through the inductor and the capacitor exactly cancel each other since they have a 180-degree phase difference. An ideal parallel resonant circuit with no damping would have infinite impedance at its resonant frequency.

However, energy losses always occur in practice, due to the ohmic resistance of the coil, magnetic losses in the core of the inductor and electromagnetic radiation. The resonant impedance thus remains finite. This causes the oscillation to be damped. For simplicity, the losses can be grouped into an equivalent parallel ‘loss resistance’ *R*, as shown in **Figure 4**.

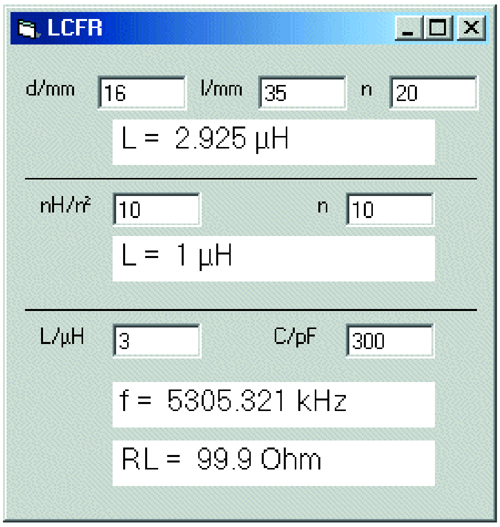
For every resonant circuit, it is possible to specify a quality factor, or ‘Q factor’, or simply ‘*Q*’, which is inversely proportional to the bandwidth of the circuit. *Q* is dimensionless and can easily be determined by taking the ratio of the parallel damping resistance *R* to the inductive impedance $R_L = 2\pi f L$ or capacitive impedance $R_C = 1/(2\pi f C)$ at the resonant frequency:

$$Q = R / R_L = R / R_C$$

If a resonant circuit is excited by an alternating current *I* with constant amplitude and variable frequency, for example using an AC generator with a high internal impedance, the voltage across the resonant circuit will be proportional to the magnitude of the complex impedance *Z*. The voltage will reach its maximum value at the resonant frequency.

The amount that the voltage increases at resonance is inversely proportional to the extent to which the oscillations are damped by any sort of energy loss, and thus directly proportional to the Q factor of the resonant circuit. At either side of the resonant frequency, points can be found at which the voltage is reduced from its maximum value by a factor of $1/\sqrt{2} = 0.707$ (–3 dB). The difference between the frequencies of these two points is defined to be the bandwidth *BW* of the circuit. The relationship between the bandwidth *BW* and the resonant frequency f_0 and Q factor of the circuit is:

$$BW_{(-3\mathrm{dB})} = f_0 / Q$$



d/mm 16 l/mm 35 n 20

L = 2.925 µH

nH/n² 10 n 10

L = 1 µH

L/µH 3 C/pF 300

f = 5305.321 kHz

RL = 99.9 Ohm

Figure 6. User interface for the inductor calculation program.

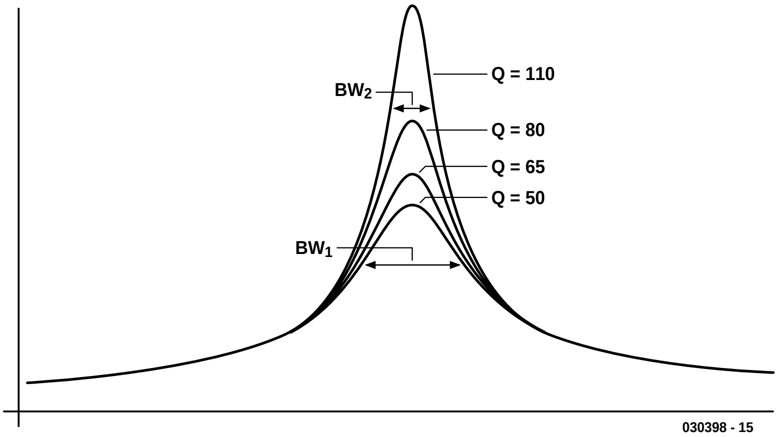


Figure 5. Resonance curves with various Q factors.

Figure 5 shows resonance curves for several different Q factors. A circuit with $Q = 50$ has a greater bandwidth (BW_1) than one with $Q = 110$ (BW_2). You can also see that the peak value at resonance increases as the *Q* factor increases. This means that the resonant circuit oscillates more strongly at its resonant frequency. By contrast, the various circuits show nearly the same behaviour in regions far away from the resonant frequency.

In practice, the circuit damping, and with it the *Q* factor, almost always arises from a combination of series and parallel resistances. The series resistance comes from the wire used to form the coil, and at a given frequency it is greater than the DC resistance of the inductor, due to the ‘skin effect’. The parallel resistance is determined by the matching impedance in the circuit. However, iron cores and ferrite cores also have losses that can be expressed in the form of a parallel resistance. For a given inductance, an inductor with a core requires fewer turns and thus has smaller copper losses, but this comes at the price of core losses.

At high frequencies (above approximately 100 MHz), pure air-core inductors wound using thick, silver-plated wire give the best results, while at medium frequencies (around 10 MHz) the best *Q* factor can be obtained using closed cores, such as ring cores. However, air-core inductors can also be used down to frequencies of approximately 1 MHz. By contrast, inductors and transformers for use at low frequencies use almost always require cores.

With careful coil construction, *Q* factors of around 100 can be achieved. However, resonant circuits can also be damped by connected circuitry or an aerial. This damping can be counteracted by using loose coupling to the resonant circuit via a small auxiliary winding, a coil tap or a suitable capacitor. When a resonant circuit is connected directly to an amplifier, the internal impedance of the amplifier must be very high in order to minimise the damping.

Inductor calculations using software

A small Visual Basic (VB) program called **LCFR** has been written to make calculations for inductors and resonant circuits. The source and executable code for this program can be obtained free of charge from the *Elektor Electronics* website. The number is **030398-11**, see this month’s Free Downloads. This program, whose user interface is shown in **Figure 6**, calculates the inductance of air-core inductors and inductors using cores with known A_L values. In addition, it can determine the resonant frequency and inductive impedance R_L of the inductor at resonance if a capacitance value is entered in addition to the inductance value.

The program consists of three parts that independently perform calculations, which for purely practical reasons are merged into a single user interface. Calculations for air-core inductors are made using the upper portion of the window, and magnetic-core calculations are made using the middle portion. At the bottom you can see the calculated resonant frequency and inductive impedance. If a new value is entered in any of the boxes, the results will change immediately. The most recently calculated inductance is automatically used for the calculations in the bottom section.

This program is ideal for quickly 'trying out' new parameters. The fact that the inductance is shown to three decimal places should not be taken as an indication of the accuracy of the result. Instead, it is intended to allow calculations to be made for inductors covering a wide span of inductances, ranging from a few nanohenries (1 nH = 0.001 μH) to many millihenries (1 mH = 1000 μH).

If you want to build a resonant circuit for a specific frequency, you can start by entering the capacitance, then calculating the inductance and finally determining the number of turns for a type given type of core or coil form. However, a less systematic approach often works better. You can simply select a type of inductor and then try several inductance and capacitance values until you find a satisfactory result. For instance, you may wish to determine which standard component values of fixed inductors and capacitors can be used to build a resonant circuit that will have a particular resonant frequency in a specific circuit. Here a trial-and-error approach often yields results faster than systematic calculation.

Some practical examples

Suppose you want to wind a 330-μH inductor for a medium-wave detector radio on a cardboard roll with a diameter of 42 mm. Further suppose that the wire diameter is 0.5 mm, so 100 turns will yield a coil length of 50 mm. Now you can simply try several different values, which ultimately yields a result of approximately 80 turns. For tuning the medium-wave (MW) band starting at 530 kHz, the variable capacitor must have a maximum capacitance of at least 45 pF.

For higher frequencies, you will need fewer turns. An inductor for a VHF FM receiver, for example, will have only five turns, with D = 8 mm and l = 10 mm. The calculated inductance is 0.16 μH. With a 20-pF capacitor, this inductor will resonate at 88.9 MHz, which is almost exactly the lower limit of the VHF FM broadcast band.

The above examples use air-core inductors. But how can you use a ferrite core? Usually, you won't have any exact data for the core. You will thus have to estimate how much it will increase the inductance or reduce the frequency. An inductor for the short-wave band, for example, might have n = 18 turns, D = 8 mm and l = 12 mm. For a pure air-core inductor, this gives a calculated inductance of 1.7 μH. But with a 275-pF variable capacitor, this inductor achieves a lower frequency limit of approximately 5 MHz with the core fully threaded in, which corresponds to an inductance of approximately 3.7 μH.

The frequency can thus be reduced by a factor of two using a threaded core, and the inductance can be up to four times as large. A relatively long medium-wave ferrite rod can in turn increase the inductance by a factor of approximately ten. Roughly speaking, we can say that an inductor on a ferrite rod only needs to have approximately one third as many turns as a similarly dimensioned air-core inductor having the same inductance.

The resonant frequency of a resonant circuit can change considerably when it is built into a circuit. Particularly at relatively high frequencies, wiring capacitances have a significant effect. This means that it is often necessary to make adjustments after assembly or build tuning capability into the circuit by means of a threaded core or trimmer capacitor.

For major modifications, it is often helpful to use a few rules of thumb that can be directly derived from the formulas give above and simulated using the LCFR program. For instance, doubling the number of turns quadruples the inductance and cuts the frequency in half if the capacitance remains the same. The frequency is thus inversely proportional to the number of turns and inversely proportional to square of the capacitance. This means that twice the frequency can be attained with one fourth of the capacitance. In order to tune over a frequency range of 1:3 using a variable capacitor, you need a capacitor with a capacity ratio of at least 1:9.

Inductors are not necessarily limited to RF circuits. They are also used in interference filters, low-frequency/audio filters and voltage converters. Schematic diagrams often show only the inductance value, without any other data for the inductor. Particularly in blocking-type voltage converters, the saturation current level and resistance of the inductor are also important factors.

It is also certainly possible to use a fixed inductor with the correct inductance but still not obtain the optimum result. Consequently, it is often worthwhile to wind your own inductors, even if only for initial testing. For instance, a 1.5-mH inductor for a voltage converter can be wound on a ferrite rod from an old medium-wave radio. If you have a relatively small ferrite rod that originally had 100 turns (which can be easily counted when you unwind the coil), it must have had an inductance of 300 μH, since the commonly used variable capacitors have a maximum capacitance of approximately 300 pF. You can thus calculate an A_L value of 30 nH/n^2. From here it's only a small step to the desired result: you will have to wind approximately 220 turns on the rod to obtain 1.5 mH.

Damping, Q and bandwidth

If you know the value of the damping resistance for a resonant circuit, it's easy to calculate the circuit's Q factor, and thus its bandwidth. This following example shows how a specific problem can be solved using the LCFR program.

Suppose you want to build an aerial filter for the medium-wave frequency of 1296 kHz (BBC, AM and DRM) using a fixed inductor. A possible solution can be found using the standard values of 100 μH and 150 pF. The program calculates a resonant frequency of 1299.5 kHz, and the small deviation of 3.5 kHz lies within the allowed tolerance. The program also indicates an inductive impedance of approximately 800 Ω.

If the DC resistance of the inductor (1.7 Ω) is taken as the series resistance and divided by the inductive impedance, the resulting Q factor is approximately 500, which is unrealistically large. In actual fact, you should assume a Q factor of around 50 for small fixed inductors. The extra damping arises from the skin effect and core losses.

In order to avoid excessive loss of energy

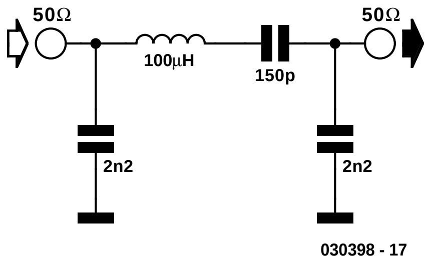


Figure 7. A filter circuit.

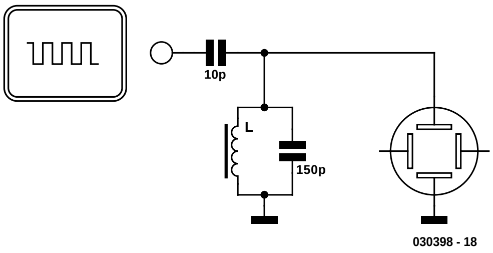


Figure 8. Exciting free oscillations.

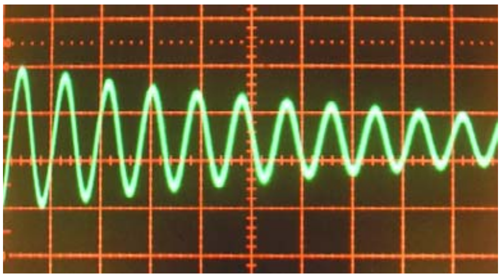


Figure 9. Damped self-resonant oscillations.

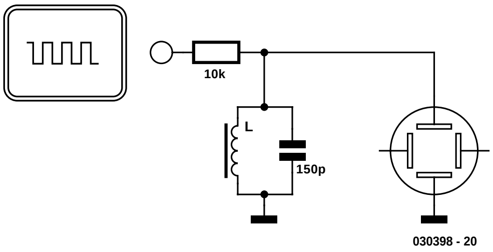


Figure 10. Resistive coupling

from the aerial, the filter should primarily be damped by the connected circuit. We thus chose a low working value of $Q = 10$. This also means that there's a reasonable chance that frequency deviations due to component tolerances will lie within the bandwidth of 130 kHz. A Q factor of 10 can be obtained with a parallel resistance of 8 kΩ.

If the aerial input of the receiver has an impedance of 50 Ω, an impedance conversion ratio of 160 is necessary. This corresponds to a voltage conversion ratio of $\sqrt{160} = 12.6$. If you used a inductor that you wound yourself, you could achieve this using a suitable tap or coupling coil. For a resonant circuit using a fixed inductor, a capacitive voltage divider can be used. The initial form of the resulting circuit is shown in **Figure 7**.

The actual resonant frequency has now been increased slightly, since the capacitance of the resonant circuit has been reduced by the two additional capacitors. You could thus suitably modify the values of the capacitors or provide a supplementary trimmer. However, it may be better to first try out the circuit. It may well be that the self-capacitance of the fixed inductor partially compensates for this error. In fact, this circuit proved to be satisfactory in practice in a DRM receiver without subsequent adjustment.

Would it be possible to use a T37-2 ring core with $A_L = 4$ nH/n^2 to make a DIY inductor? This idea can be quickly simulated and equally quickly rejected, since 158 turns is an unrealistic value for a small ring core. However, an air-core inductor with $D = 8$ mm, $l = 8$ mm and 110 turns is conceivable. If you have a suitable threaded core, you can manage with approximately half as many turns. This also allows the exact frequency of the filter to be adjusted using the core.

Measurements

If you do not have a suitable signal generator or dip meter, the only way to measure the resonant frequency of a resonant circuit is to use an oscilloscope. The frequency can be measured by exciting the circuit so it oscillates freely (**Figure 8**).

This requires a steep-edged square-wave signal with a frequency well below the resonant frequency. Many oscilloscopes have a square-wave output for calibration purposes (usually 1 kHz). This should be coupled to the resonant circuit as loosely as possible via a small capacitor. With suitable adjustment of the time base, you will be able to observe free oscillations. Besides the resonant frequency, damping can also be measured in this manner.

Figure 9 shows a measurement made using the previously calculated resonant circuit. What is important is to affect the circuit as little as possible. This means you should use a 10:1 probe with an internal impedance of 10 MΩ. The figure shows the result of a measurement made with a horizontal deflection factor of 1 µs/division. The measured frequency is slightly greater than 1100 kHz.

The Q factor can be determined from the number of oscillations required for the amplitude to drop to 0.37 (1/e) of the initial value. Here the Q factor is approximately 10.

The frequency and Q factor are affected by the measurement set-up. However, they will be affected even more strongly in an actual circuit.

A 10-kΩ coupling resistor can be used instead of a 10-pF coupling capacitor (**Figure 10**). In this case, the resulting oscillations will be somewhat weaker, so a more sensitive vertical scale must be selected. Using a resistor for coupling generates higher damping. However, the advantage of this approach is that it avoids any shift in the resonant frequency due to the coupling capacitor, so the frequency can be measured more accurately. In addition, measurements can also be made over a wide frequency range, from around 10 kHz to many megahertz.

This simple measurement technique can also be used to measure the values of unknown inductors. If you use a known capacitance value and measure the frequency, you can then determine the inductance. It is often necessary to try several different capacitors before you obtain readily measurable oscillations. With relatively large inductances, it is necessary to use correspondingly large capacitors. It is also possible to determine the value of an unknown capacitor using a known inductance.

The A_L values of unknown cores can also be determined using this technique. To do this, wind a small test coil on the core and determine the resonant frequency with this coil connected to a known capacitor. From the number of turns and the inductance, you can then determine the approximate A_L value.

(030398-1)

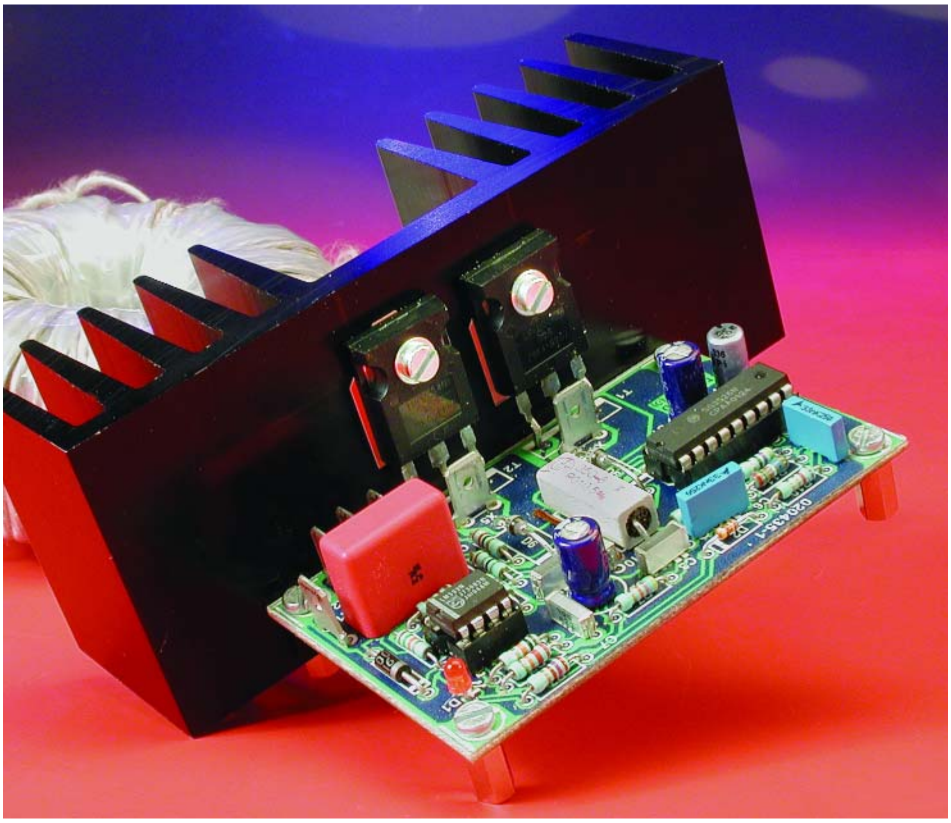
Simple 12-to-230V Power Inverter

A mobile power outlet

Design by G. Gerards

www.aixcon.de

The absence of a mains power outlet is often keenly felt on camping sites, with car repairs in the middle of nowhere and with picnic or party events in the countryside. In some cases, mains power can only be brought to a remote site by running a very long cable — which either dangerous, impossible or not available. Whatever the occasion, planned or unexpected, it is great to have a power inverter available capable of changing the 12-V car battery voltage into 230 volts AC.



The idea for a simple, portable 'power outlet' was first suggested by a trainee at the Aixcom company, which is normally involved with high-tech power inverters and special high-current power supplies. The trainee, called Dirk, had been trying for quite some time to build a power inverter for his model aircraft club. In his enthusiastic attempts he ran into problems obtaining the special integrated circuit that was to form the heart of his project. When he was finally successful in obtaining the elusive chip, albeit at horrific costs, all the circuit did was produce a loud bang at switch-on, wrecking a lot of components.

The company decided to continue the design, and the result is presented here: a power inverter that was not only successfully reproduced by nearly all trainees at Aixcom, but also presented as a Christmas or anniversary gift to dad, used on a camping site and, last but not least, deployed in a (very loud) music parade. A beefed up version of 1,000 watts was developed and installed by Dirk at his model aircraft club, where it has performed beautifully for over a year despite rough conditions.

Concept

Arguably, the circuit represents the simplest way of creating a power outlet for on the road. In the design phase, the aim was a 100% bare bones circuit, stripped of anything that could be, well, stripped! For example, there's no voltage regulation, and a sagging battery voltage also causes the ac output voltage to sag. However, because most mains powered equipment continues to work just fine in the face of ac voltage variations of ±10-15%, the mobile power outlet does so too. Rather than perfecting the design for performance, Aixcom went for simplicity, low component count and utter reliability in practical use. None the less, the 230-Vac output is short-circuit resistant and an undervoltage protection switches the inverter off before the battery has been drained to level that would no longer allow the car to be started. The circuit is simple enough to be reproduced successfully by beginners, too, provided they realise that 230 Vac is a really dangerous voltage.

Pulsewidth modulation

The central part in the circuit is an SG3526 low-cost switch-mode regulator, which is supplied by a number of manufacturers under the component identifier xx3526, where xx is a manufacturer-specific letter combination. The 3526 supports all known switch-mode PSU topologies. Its complete datasheets may be obtained free of charge from www.unitrode.com (part search: UC3526 and Datasheet).

The basic operation of the power inverter is illustrated in **Figure 1**. The SG3526 alternately switches the current through the 12-V windings of a mains transformer, the two central ends of the windings having been taken together and connected to the positive battery terminal (+12 V). At each switching action, the direction of the current changes and with it the direction of the magnetic field in the transformer core. The result is a square-wave(-like) alternating voltage at the 230-V side of the transformer.

In real life, the switch consists of two FETs in complementary arrangement (push-pull). The source connections of the FETs are taken to ground by way of very low resistances (compare the circuit diagram in Figure 3).

The internal architecture of the SG3526 is shown in **Figure 2**. The input voltage $+V_{in}$ may be between 7 V and 35 V, and is used to create a reference voltage V_{REF} of 5 V. A voltage guard blocks the drivers stages when the input voltage drops below 7 V. The drivers are separately powered via the $+V_c$ connection. Using resistor RT and the capacitor at CT (again compare with circuit diagram in Figure 3) the frequency is determined, which is 50 Hz in this case. The resistor at R_D causes a fixed dead time between the driver's Output A and Output B. This is done to eliminate the risk of the two drivers (and consequently the two power FETs) conducting at the same time when the switch-over takes place.

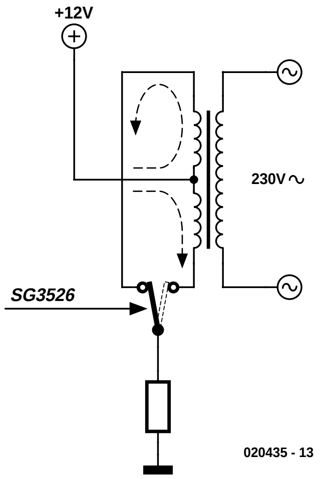


Figure 1. Block diagram of the power inverter. In the actual circuit, the switching element consists of two power FETs, while the resistor to ground acts as a current sense (shunt) for the current limiter circuit inside the SG3526.

The capacitor at the $C_{SOFTSTART}$ pin (Css, pin 4) allows the pulse mark/space (on/off) ratio of the outputs to be slowly raised to 48% after the supply voltage is switched on, or after a reset. The 'Amp' voltage regulator is not used as such in our application, alternatively it takes the role of an impedance converter using the reference voltage as the controlling quantity. In this way it is assured that the outputs supply the full mark/space ratio after the start-up phase.

The current limiter using shunt resistor R8 triggers a shutdown sequence when the voltage between +CS and –CS (in other words, the drops across R8) exceeds 100 mV. However, the shutdown control may also be used externally by connecting it to ground. Because shutdown and Reset (pins 8 and 5 respectively) are interconnected in this circuit, the modulator starts again with a soft start after an overload condition or an external disconnect.

More design thoughts

The transformer for the project should be a toroidal type with a primary of 230 V and two 12-V secondary windings. Readers in countries with 110 V, 117 V or 127 V mains voltage are, of course, advised to use a matching 200-

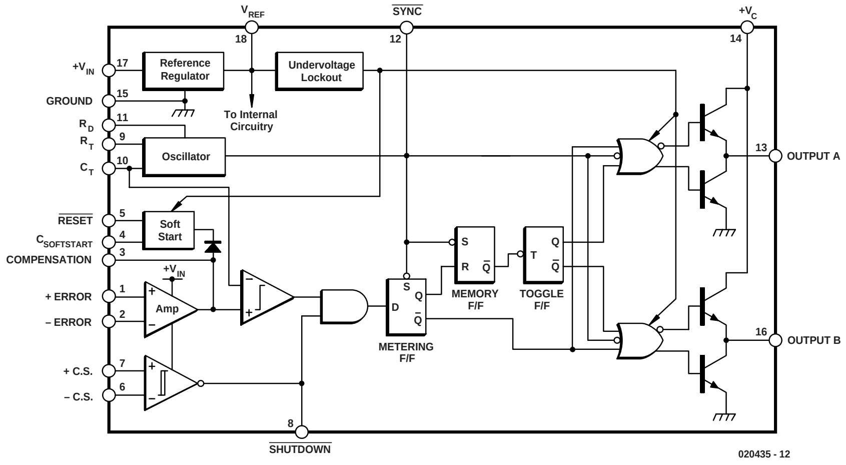
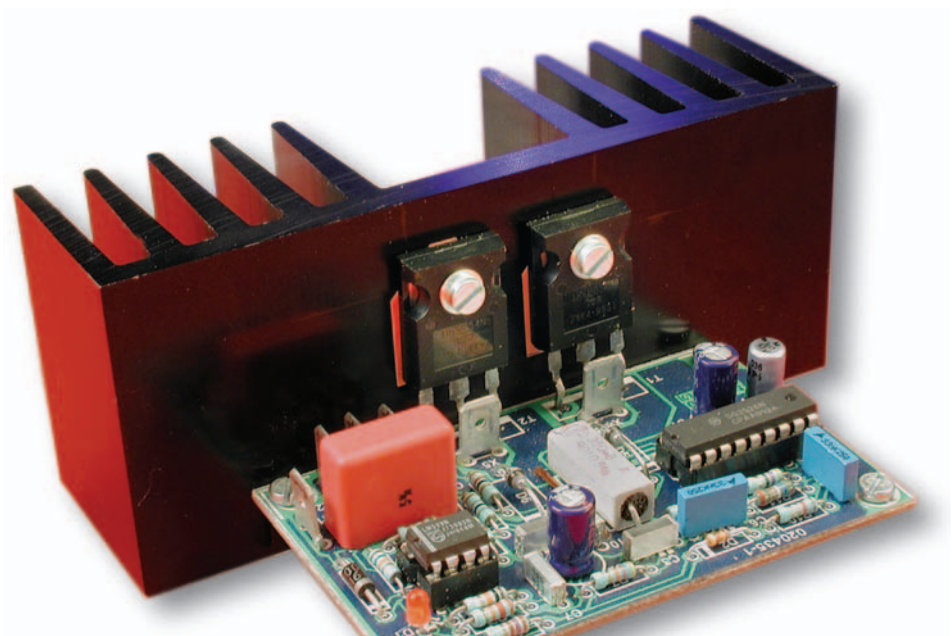


Figure 2. Internal diagram of low-cost SMPSU regulator type SG3526.

watt transformer. If you are lucky to have an old toroidal transformer lying around in a drawer or a junk box, it should not be too difficult to ‘retro-fit’ two 12-V windings. Simply wind ten turns of litze wire around the core and connect the primary to the mains. Measure the voltage across your new winding and then calculate how many turns you need to get at 12 V. At an output power of 200 watts, the average current will be about 10 A, so the cross-sectional area (c.s.a.) of the litze wire you’re using should be 1.5 mm^2 or greater.

It is vital that the two 12-volt windings have **exactly the same number of turns**. If there is a difference of just one turn then the transformer core will saturate the instant the 12-volt battery is connected, causing the regulator to ‘hang’ in shutdown mode. The sense direction) of the windings is equally important. Before installing the transformer, connect the ends of the two 12-volt windings in series and apply 230 Vac to the primary. You should measure 24 Vac across the free ends of the secondaries.

The FETs used in the circuit can handle up to 72 A at 55 V, and are marked by an $R_{D\text{-}S(ON)}$ of just 12 mΩ. Of course, other types may be used provided you are sure they can handle at last 40 A at 40 V, and have an $R_{D\text{-}S(ON)}$ not exceeding 50 mΩ. Usually, power FETs may also be connected in parallel, but please make sure each one gets its own gate resistor. The parallel configuration is of interest if you wish to configure the inverter for output powers greater than 200 watts. In that case, the current limiter has to be adapted, which is easiest done by



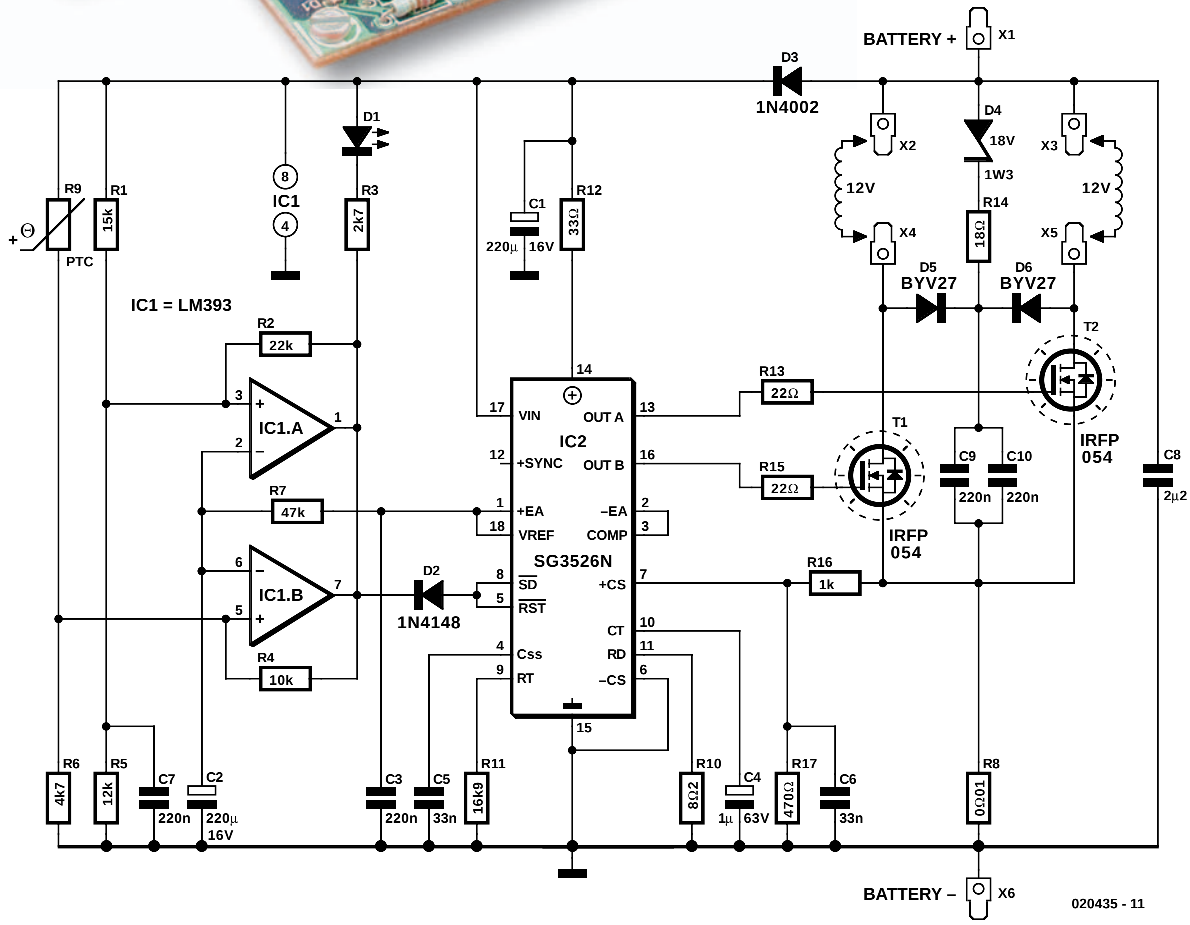


Figure 3. Circuit diagram of the 12V-to-230V Power Inverter. Comparator IC1 acts as a guard for temperature as well as battery voltage.

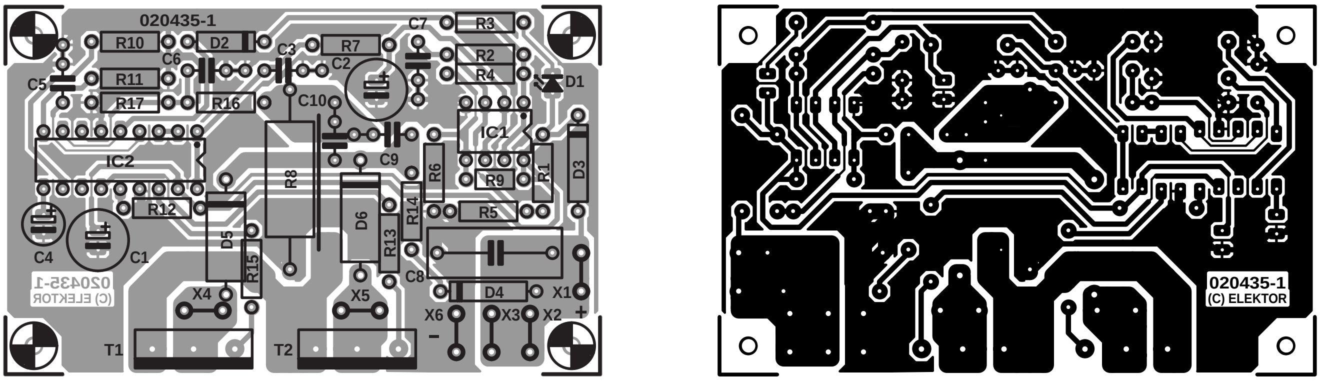


Figure 4. Copper track layout and component mounting plan of the PCB designed for the project. The board is single-sided and available ready-made through Readers Services.

using a smaller value for shunt R8 and/or by modifying voltage divider R16-R17.

Ordinary bulbs and halogen (flood-) lights are sure sources of trouble with most power inverters. Both present a very low 'cold' resistance, causing the inverter to reduce its output voltage or even actuate the shutdown. The result is a lock-up with isufficient voltage to heat up the filaments to their normal operating temperature. Fortunately, the 200-watt version of the inverter described here should be capable of turning on lamps of up to 150 watts without problems. Should problems arise, the value of capacitor C6 may be increased — but not to any extent because the ability of the circuit to withstand short-circuits may well suffer. Using C5 it is also possible to increase the soft-start time considerably, or do without it altogether. That, in all likelihood, is the safest solution.

Comparator IC1 monitors the battery voltage and ambient temperature and compares its measured results with the 5-V reference voltage from the 3526. The two open-collector outputs pull the shutdown control input (pin 8) to ground in case of an error. The PTC used determines the switch-off temperature. Depending on the exact type in your circuit, R6 may need slight redimensioning. Early Aixcom prototypes of the inverter used a D901-D60-A40 from Epcos (trip temperature 60 degrees C). However, it should also be possible to use a temperature switch of 60 to 80 degrees C or a temperature fuse of 90 degrees C. Although the latter component is extremely cheap at just a few pence, you'll need to exchange it when it has 'gone off'.

Provided a large enough heatsink is used, a simple wire link may be used instead of the PTC. The voltage monitor switches off at about 12 volt and this may be adapted to suit other levels by changing R1 and R5. On the comparators, R2 and R4 define an amount of hysteresis that prevents the power inverter from switching itself on again after a fault condition. After switching on, the reference voltage rises slowly as determined by the charge time of C2, hence the monitors are only activated a few seconds later.

Car batteries supply dangerously high currents. To prevent the inverter going up into flames and causing a fire, you **must** protect it with a car fuse of between 25 A and 35 A. The 230 Vac output voltage is also **very dangerous** even if it is generated by means of a battery.

Construction

The design of the printed circuit board is shown in **Figure 4**. Despite large ground areas and wide tracks it may be necessary to strengthen the tracks carrying the transformer current by tinning them. It is recommended to start by mounting the AMP ('fast-on') lugs (spade terminals), because they require considerable force to insert into the board. After all, a mishap with the use of pliers at this point could cause con-

COMPONENTS LIST

Resistors:

R1 = 15kΩ
R2 = 22kΩ
R3 = 2kΩ7
R4 = 10kΩ
R5 = 12kΩ
R6 = 4kΩ7
R7 = 47kΩ
R8 = 0Ω01 (max. lead pitch 24mm)
R9 = 1kΩ (PTC, see text)
R10 = 8Ω2
R11 = 16kΩ9
R12 = 33Ω
R13,R15 = 22Ω
R14 = 18Ω
R16 = 1kΩ
R17 = 470Ω

Capacitors:

C1,C2 = 220µF 16V radial
C3,C7,C9,C10 = 220nF
C4 = 1µF 63V radial
C5,C6 = 33nF
C8 = 2µF2 63V, 15mm lead pitch, MKS4 (Wima)

Semiconductors:

D1 = LED, red, low current
D2 = 1N4148
D3 = 1N4002
D4 = 18V 1.3W zener diode
D5,D6 = BYV27-200
IC1 = LM393N
IC2 = SG3526N
T1,T2 = IRFP054N (IRF)

Miscellaneous:

X1-X6 = AMP spade terminals, PCB mount
PCB, order code **020435-1** (see Readers Services page)
Toroidal mains transformer, see text, e.g., Aixcon 230V /12-0-12 V / 200W (www.geist-electronic.de)

Table 1

Output voltage vs. battery voltage (150 watt load)

Battery voltage [Vdc]	Output voltage [Vac]
11.5	182.4
12	194.6
12.5	202.4
13	214.3
13.5	223.0
14	231.2

siderable damage to other components on the board. The wire link beside the shunt resistor R8 should not be forgotten. R8, by the way, should be mounted a little above the board surface to help it stay as cool as possible. If desired a higher-wattage resistor may be substituted (5 watts). Finally, do make sure you mount all polarised components (transistors, electrolytic capacitors, diodes and ICs) the right way around on the board. Insulating washers must be used when fitting the transistors onto the heatsink.

Powering up

Commissioning this project only requires a multimeter. Initially, you use the inverter **without the transformer connected**. Connect it to an adjustable bench supply and check the two guard circuits: the voltage guard by adjusting the input voltage, and the temperature guard with the aid of your soldering iron, a potentiometer or any other means you see fit. In any case, the outputs will switch to ground and the LED will light when the voltage at the positive input of the comparators drops below that at the negative input. If the guard circuit appears to work, you proceed by measuring the two gate signals. If an error is present, both will read 0 V. In the case of an error-free circuit, an oscilloscope will show two clean rectangular-wave signals with 10-ms long pulses. Using your multimeter, the same measurement yields a readout of about half the supply voltage.

All approved so far, you are in a position to connect the toroidal transformer. At this point, it makes sense to remove IC1 from its socket, as in that case the shutdown can only be triggered by the current limiter. If an ordinary 100-watt bulb does not light up within a few seconds, measure the voltage at the shutdown control (pin 8 on the 3526 or the anode of D2). If you measure less than 5 V, the current limiter or the soft-start time has to be tweaked as described above.

Once the bulb lights, you may (carefully!) check if the inverter is short-circuit resistant. If an oscilloscope is available, the FET current may be measured (= the voltage across R8) and use R16 to increase the current limit point to about 20% below the permissible drain current. This is of course done with the 230-Vac output short-circuited.

It is normal for the transformer to make more noise under no-load conditions than you would expect when in normal use. This is caused by the rectangular wave switching the magnetic field hard and fast. Core saturation under no-load conditions is signalled by ugly sounds from the transformer. Measured with an oscilloscope the currents will not rise in sawtooth-wise but with peaks (overshoot). In that case, the 12-V windings on the transformer require just a few more turns. If that is problematic, the alternative is to raise the oscillator frequency a little by using a slightly lower value for R11. The resulting output frequency may well be 55 Hz, but that is immaterial for most loads and the circuit is not suitable anyway to power an alarm clock.

Practical results

Because a voltage regulation loop omitted for the sake of simplicity and cost, the output voltage is dependent on the battery voltage. The output voltage of the author's prototype loaded with a 150-watt halogen lamp is shown in **Table 1**, as a function of battery voltage.

The output voltage is dependent on the transformer's winding ratio and output current. If you want to reach the nominal output voltage of 230 Vac at 13 Vdc input, you should consider using a transformer with two 11-volt windings. On the prototype, a maximum efficiency of 94% was measured and the circuit was found to be Dirk-proof.

(020435-1)

Foil Capacitor Polarity

Which terminal of a foil capacitor acts as the screen?

By H. Friedli

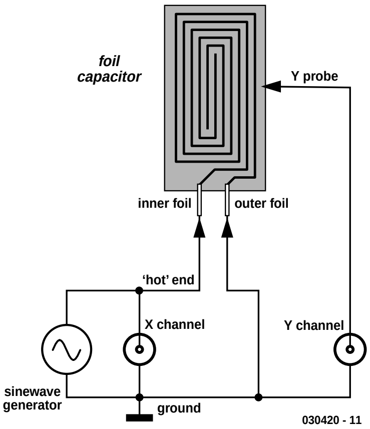


Figure 1. Method of connecting a foil capacitor to a sinewave generator and an oscilloscope.

As most of you will be able to confirm, an electrolytic capacitor has to be connected the right way around in an electronic circuit. If you get the connections wrong, the innards of the capacitor will greet you when the power is switched on — the capacitor can will rupture, spilling its liquid or partly vaporised contents across the circuit board or (in the case of older high-voltage electrolytics) spouting it high up into the air. The rupture is not a silent process either as most of you will have learned the hard way. Depending on the size of the capacitor, a small explosion may occur. By contrast, fixed capacitors are non-polarized components and therefore not prone to explode unless they are operated way beyond their specifications in respect of voltage or surge current.

The is however one type of fixed capacitor that may still be qualified as 'polarised'. We're talking about foil capacitors, which can pick up noise when connected the wrong way around. This may affect the operation of sensitive electronic circuits like preamplifiers, the negative effects consisting of hard to explain levels of hum or cross-talk, or a tendency to oscillate. Take, for example, the wire of the coupling capacitor connected to the (very high-impedance) input grid of a valve. This particular wire must be connected to the inner foil of the coupling capacitor, the outer foil being connected to the anode of the driving valve, which represents a much lower impedance.

With many types of capacitor, the outer foil is marked by, for example, a dash, a dot or some other indication. Unfortunately this is not common practice among manufacturers and even the print on the capacitor may not always provide a definitive clue to how the foils are arranged inside the device.

Trick of the trade

A simple tricks is available that allows you to relate the capacitor wires to the rolled up foils inside. A sinewave voltage is applied to the capacitor as well as to the X (horizontal) input of a dual-trace oscilloscope. The test signal should have a frequency of about 1 kHz and a level of a couple of volts peak to peak. The probe connected to the Y channel is held close to the capacitor's plastic encapsulation. On the 'scope display you watch the amplitude changes in a very sensitive input range. With the 'scope set to X-Y mode, a more or less opened, widened or possibly sloping ellipse should be seen. If the ellipse widens in the Y direction when the probe is moved closer to the capacitor body, the outer foil is connected to the 'hot' (signal) end of the generator output. If it becomes narrower, the same foil is connected the ground. To verify the outcome, swap the capacitor connections and repeat the test.

If you are unable to see a clear shape change to the ellipse, either the signal voltage or the test frequency is too low, the Y channel is not sensitive enough or the effect of your hands is too large. By fitting the tip of the Y probe with a small disc results in a larger capacitance between the outer foil and the Y-channel input. This may help to improve the reliability of the test described above.

Once you've unequivocally identified the connections that go with a capacitor's inner and outer foil, that does not mean the outcome can be applied to other, identical capacitors! The author found inconsistencies even among capacitors from a batch produced by one of and the same (reputable) manufacturer. In practice, each foil capacitor has to be tested individually if you want to be absolutely sure where its wires are best connected to in your circuit.

(030420-1)

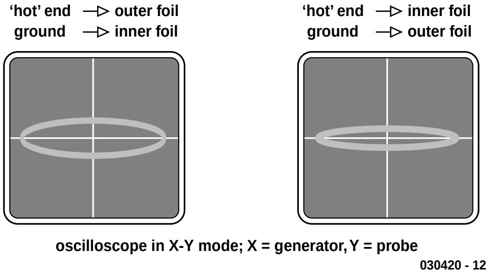


Figure 2. Left: wrong polarisation; right: correct polarisation.

Touch-controlled Switch

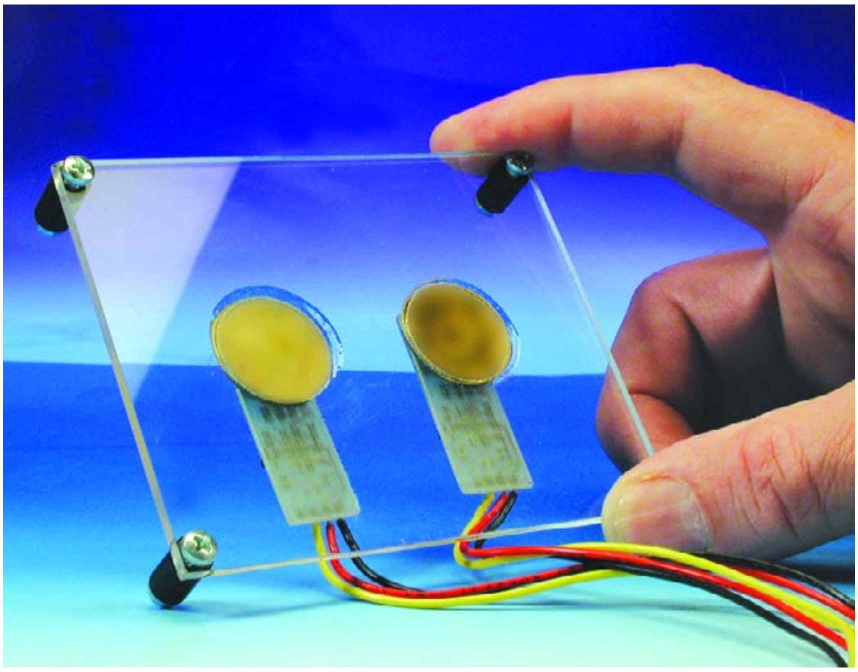
With a PIC

Design by J. Wickenhäuser

www.wickenhaeuser.com

You don't need special ICs when a capacitive (or 'touch') switch can also be realised with an inexpensive PIC controller. This article shows how it's done.

Over the past few years several manufacturers have come up with special integrated circuits for touch control switches. These ICs usually operate on the principle of capacitive change, work perfectly but are hard to obtain as well as relatively expensive. Fortunately, a capacitive switch may also be realised using 'traditional' means, i.e., a little physics and a microcontroller.



The human body

may act as one plate of a capacitor, with the other plate formed by a coin soldered onto a copper plane. The principle is illustrated in **Figure 1**. Assuming the coin has a surface area of 3.2 cm^2 and is placed at a distance of 4 mm (using an acrylic plastic disc as a spacer) and a dielectric constant of 8, a theoretical capacitance of 8 pF is obtained, or 2-3 pF when the pad is not touched.

A second capacitor (C2) is charged via two resistors. In reality, the three switches are microcontroller port lines with GP1/GP2 representing signal 'A' and GP0, signal 'B'.

The measurement comprises the closing of the two switches for 2 μs using signal 'B'. The 2-μs period allows capacitor C1 to almost fully charge via R1. A very small portion of the charge (the amount is negligible) also flows to ground via R2. Next the two switches are opened and C1 is allowed to charge C2. As soon as the voltage on C2 exceeds a certain level (here, about 0.7 V) the measurement is finished. Signal 'B' disappears and C2 is discharged. This marks the return of the measurement cycle to its start state. The number of iterations it takes to fully charge the second capacitor is counted — it's as simple as that.

The simultaneous charging and discharging of C2 may appear a bit of a contradiction. However, a small portion of the charge in C2 is also lost, because the switches are open much longer than closed.

The smart button

Ideally, about 20,000 cycles are required to enable C2 to be charged by the user's finger. Consequently a measurement takes about 50 ms. As soon as the software has recognised a 'key press' action on GP3, output GP4 is pulled Low, causing the indicator LED to light up. At the same time, the number of cycles for each measurement is output via pin GP5. An oscilloscope or a pulse counter connected to GP5 will clearly indicate the approach of the finger.

As you can see from the circuit diagram in **Figure 2**, the microcontroller used is a cheap and easy to obtain type PIC12C508 which has 512 bytes of memory. If you happen to have a '509 lying around (with 1024 bytes) it can be used also. The R-C oscillator inside the PIC operates at about 4 MHz.

The printed circuit board shown n **Figure 3** is single-sided. Normally the copper plane on the board is sufficient for reliable operation. In cases

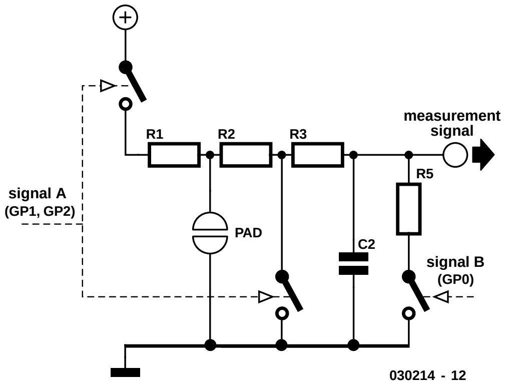


Figure 1. Principle of operation.

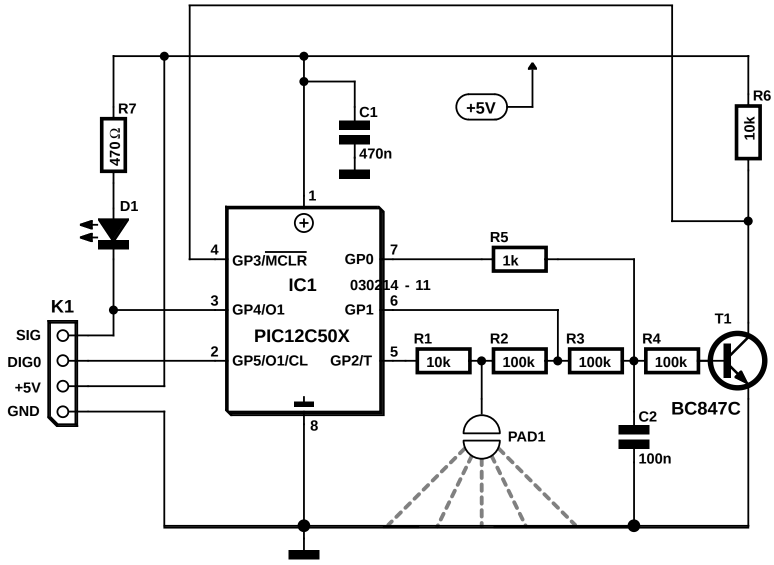


Figure 2. Circuit diagram of the Touch-controlled Switch.

Free Downloads

PIC source code for use with Microchip MPLAB assembler.
File number: 030214-11.zip

PCB layout in PDF format.
File number: 030214-1.zip

www.elektor-electronics.co.uk/dl/dl.htm, select month of publication.

COMPONENTS LIST

Resistors:
R1,R6 = 10kΩ (SMD shape 0805)
R2,R3,R4 = 100kΩ (SMD shape 0805)
R5 = 1kΩ (SMD shape 0805)
R7 = 470Ω (SMD shape 0805)

Capacitors:
C1 = 470nF (SMD shape 1206)
C2 = 100nF (SMD shape 0805)

Semiconductors:
D1 = Chip LED, red, (SMD shape 0805)
T1 = BC847 (SOT23 case)
IC1 = PIC12C508A04/SO8, programmed, order code **030214-41**

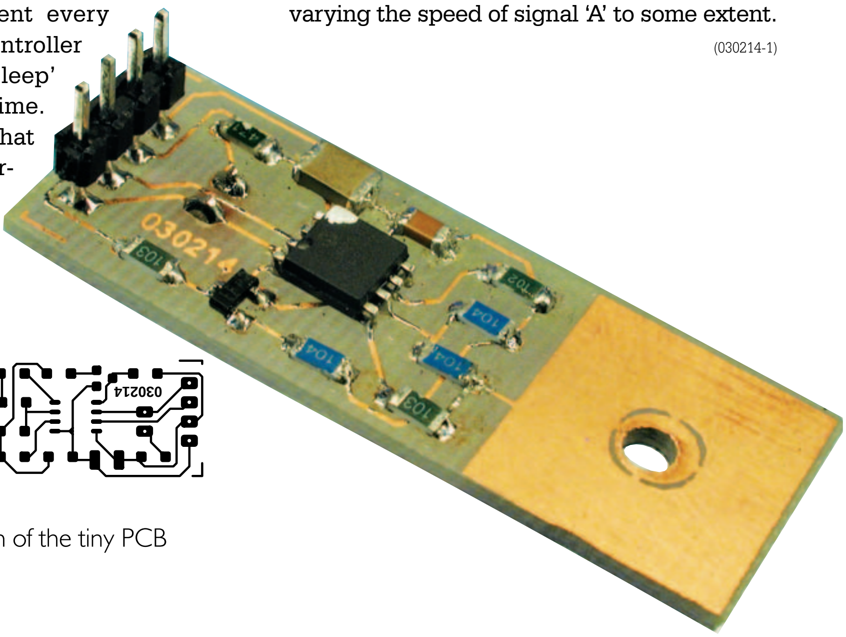
Miscellaneous:
K1 = 4-way mini PCB terminal block, lead pitch 2.5mm
PAD = sensor area (see text)
PCB, order code **030214-1**
Disk, PIC source code file, order code **030214-11** or Free Download

where higher sensitivity is required, the surface area may be increased.

The circuit is capable of adaptive operation. This means that the trigger point is shifted (within limits) to compensate for effects like dirt on the touch pad. Only relatively fast capacitive changes are recognised as valid actions. Also, the fact that the measurement is completely done in software can be exploited by making clever use of the energy reduction features offered by a PIC. Because a measurement takes only 50 ms and it is sufficient to do measurement every 500 ms, the controller could be 'asleep' most of the time. The upshot is that the average current consumption of the circuit can be reduced to 0.1-0.2 mA without problems.

The circuit is made relatively immune to constant external RF signals by the software varying the speed of signal 'A' to some extent.

(030214-1)



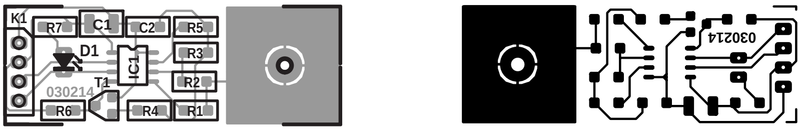


Figure 3. Copper track layout and component mounting plan of the tiny PCB designed for the circuit.

Parallel Port Mains Switching Interface

switch radio-controlled mains sockets on and off from a PC

by K. Heiden

The wide availability of low-cost switched mains sockets with radio remote control provides a simple way to control electrical equipment, lights, etc. using a PC. Using radio not only means that we do not need a cable, but also avoids problems with electrical safety as the interface is electrically isolated from the mains supply.

As USB interfaces become more widespread on PCs the serial and parallel ports are becoming less and less important.

Most computers, however, still offer the ‘legacy’ ports which are available for new applications, now that devices such as printers tend to use USB. The parallel port, for example, is ideally suited for controlling electrical devices remotely. Using a simple switch interface it is possible to turn household appliances on and off under PC control.

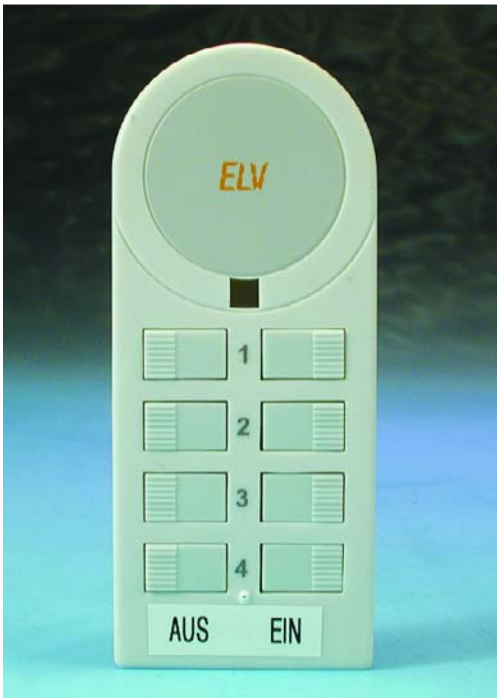


Figure 1. The handheld transmitter (ELV part code FS 10 S8-2) before modification.

So as not to add to the cable spaghetti already associated with the PC, the remote control described here uses a modified radio-controlled mains switch. These are available from various outlets including specialist electronics shops as well as DIY and housewares shops. Generally two or three switched sockets are sold as a set along with a single hand-held remote control (see **Figure 1** and **Figure 2**).

In order to control the switches from a PC we will need an interface to connect the parallel port of the PC to the hand-held remote control transmitter. If the facility for manual control must be retained, the best approach is to purchase another transmitter, as the unit connected to the interface will be attached to the parallel port using only a short cable and so it will not be convenient to operate it manually.

Interface Circuit

The parallel port interface circuit (**Figure 3**) is essentially divided into two parts: the part that drives the remote control transmitter using optocouplers, and two optional extra switching stages built around IC1 and two relays. This switching circuit allows for the control of two additional devices, located near to the PC, directly from the interface. If this feature is not required, the switching stages can be left out, and signals D0, D1 and D3 (port pins 2, 3 and 5 of the 25-way sub-D plug) left unconnected.

Port signals D4 to D7 (pins 6, 7, 8 and 9 of the 25-way sub-D plug) are connected to the remote control transmitter via a quad optocoupler device (IC2, type ILQ74). Figure 1 also shows the pinout of this device.

The circuit diagram only shows the part of the remote control transmitter circuit that is connected to the interface circuit, namely the pushbuttons and the connections for the power supply. The pushbutton connections for the transmitter are brought out on four wires labelled A to D and connected to the output transistors of the optocouplers (see

Figure 2. A radio-controlled switched mains socket (ELV FS 10 system).

also **Figure 4**). When the port pin driving an optocoupler goes high, its output transistor conducts, bridging the corresponding pushbutton contacts on the transmitter. The effect is the same as if the button had been pressed (channel 2 on/off, channel 3 on/off and so on). We can arrange for the pin on the parallel port to be taken high using suitable software. The components between the parallel port and the optocoupler (the series resistor and the diode) serve to limit current flow and protect both the input LED of the optocoupler and the output transistors of the parallel port.

The interface circuit also includes a power supply for the remote control transmitter. The transmitter batteries (two button cells in the type we used) should be removed and connections made to the points marked '+' and '–' in the circuit diagram. A voltage of 3 V for the transmitter is derived from the 12 V supply of the interface using D2, R11, C8 and Zener diode D3. The interface circuit itself can be powered from a 12 V mains adaptor connected to K1 or by connecting to the PC's internal 12 V supply.

The optional switching stages connected to port pins D0, D1 and D3 are equipped with one normal relay (with one coil) and one latching relay (with two coils). They are controlled by driver IC1 (type ULN2803).

The latching relay (RE1) is controlled using port signals D0 and D1 and has the feature that its state is preserved when power is switched off. In contrast, RE2 is an ordinary, non-latching, relay and is controlled using D3. Its contacts are closed only while current flows in the coil. Switching pulses can be generated using this relay with a duration under software control. Thus it is suitable, for example, for driving an electrically-operated door latch from the PC, connecting the relay contacts in parallel with the button for opening the door.

You may be wondering why there are no protection diodes across the relay coils: this is because suitable diodes are already built into the ULN2803.

A 25-way male sub-D connector is required to connect the interface circuit to the parallel port. A cable with at least eight conductors is required for the connection, with the 25-way sub-D connector at one and suitable connector at the other. If it is desired to use a spare ordinary printer cable, a 36-way Centronics-style socket will be required.

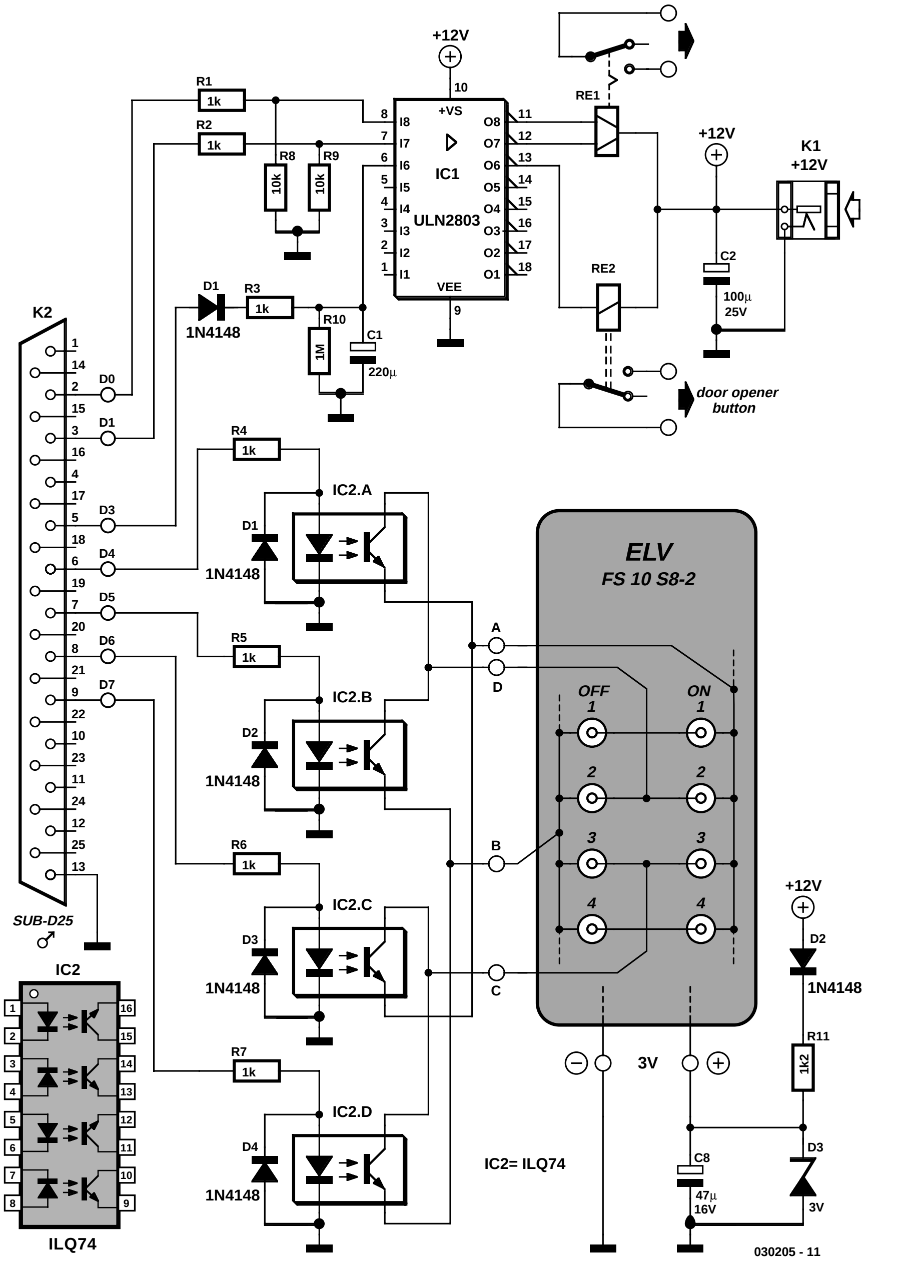


Figure 3. The circuit of the parallel port interface connects to the handheld transmitter via optocouplers. Two relays are also available.

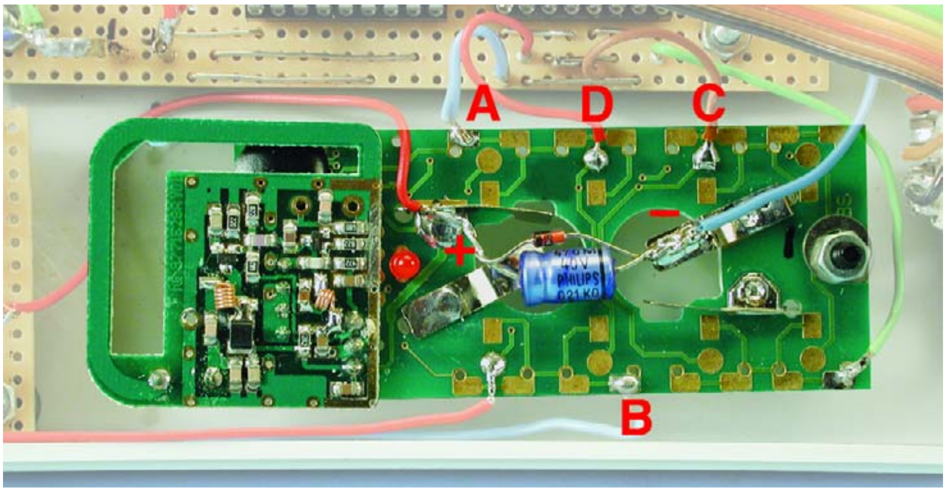


Figure 4. The printed circuit board (author's design) of the handheld transmitter with wires soldered to it to connect to the interface circuit.

Construction

The simplest way to construct the interface circuit is to use matrix stripboard. The components are all widely available and so should be easy to obtain. A suitable latching relay for RE1 is available for example from Conrad Electronics (order code 50 40 60). For RE2 practically any small 12 V relay can be used.

As can be seen from the circuit diagram of the parallel port interface and from the photographs, a few connections need to be made to the remote control transmitter using short lengths of insulated wire. The connection points are easy to find: they are simply the contacts of the pushbuttons. The circuitry inside the transmitter can be quite tightly packed, and so soldering can be a little fiddly. Things are not too difficult with the FS 10 remote control system available from the German company ELV that we used in our prototype: as soon as the enclosure of the transmitter (part code FS 10 S8-2: see Figure 1) is opened up and the caps of the pushbuttons removed, the contact points of the pushbuttons are visible (Figure 4). It is very easy to solder wires to these contacts, connecting the transmitter to the optocouplers (points A to D in the circuit). The pushbuttons can no longer be used.

The completed assembly can be protected by a plastic enclosure (**Figure 5**, which shows a slightly modified version of the interface).

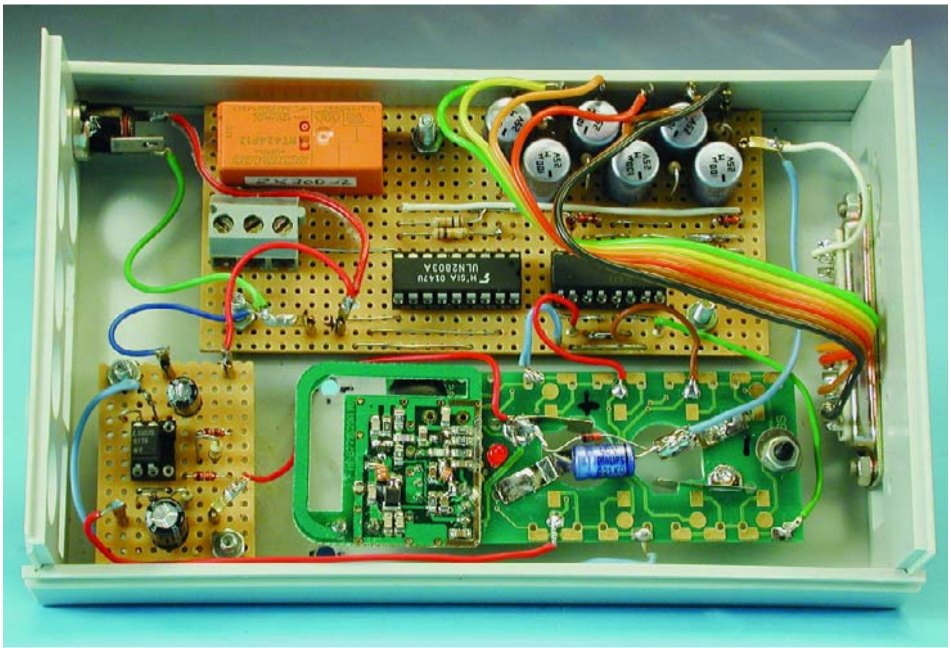


Figure 5. The switching interface built into a plastic enclosure.

The Software

The control software is as straightforward as the interface circuit. All it has to do is set the individual bits of the parallel port high for a period of about one second, and then low again. We do this using a COM file, which is created in advance with the aid of a VB (Visual BASIC) script. First type in the program Mk_COM.vbs shown in the text box using any suitable text editor (such as WordPad or Notepad), and save it under this name in any directory on the hard disk. Now run the program by double-clicking on the filename: this will create the file SETLPT.COM in the same directory. The VB script can now be deleted.

The various switching actions can be carried out using the following commands:

SETLPT 1:01 switches the latching relay on.
SETLPT 1:02 switches the latching relay off.
SETLPT 1:08 switches the non-latching relay (briefly) on.
SETLPT 1:10 switches remotely-controlled mains socket 2 on.
SETLPT 1:20 switches remotely-controlled mains socket 2 off.
SETLPT 1:40 switches remotely-controlled mains socket 3 on.
SETLPT 1:80 switches remotely-controlled mains socket 3 off.

The space after 'SETLPT' must not be omitted. The digit '1' after 'SETLPT' and the space refer to parallel port LPT1. The characters after the colon are hexadecimal digits which define the value which will be sent to the parallel port: two digits must always be specified. The assembler routine sets the parallel port to the value given after the colon for about 1 s, and then sets it back to zero again so that only a brief positive pulse will appear on the data line. If you have a second parallel port ('LPT2') on your computer which you wish to use with the interface, then the command should read ‘SETLPT 2:XX’. A note for PC experts: the I/O base address used for LPT1 is 0378 hex, that for LPT2 is 0278 hex.

You can create a shortcut to the file SETLPT.COM with parameter ‘1:08’ on the desktop, and call it `Open Door’. Then, when someone rings the doorbell while you are busy at the computer, just double click on the icon and the door opener will be activated. To prevent the DOS window appearing, right-click on the icon and select ‘Properties’. Then, under ‘Shortcut’ select ‘Run minimised’ and ‘Close window on exit’.

The following file, ‘**Mk_COM.vbs**’, can be typed in and run to create the assembler file ‘**SETLPT.COM**’ automatically. Alternatively, the file ‘**SETLPT.COM**’ can be downloaded directly from the *Elektor Electronics* website at www.elektor-electronics.co.uk (select this issue).

```
    d = “186,120,003,138,030,130,000,128,251,049,”
d = d & “116,007,128,251,050,117,030,254,206,160,”
d = d & “132,000,044,048,177,004,210,224,138,038,”
d = d & “133,000,128,236,048,008,224,177,001,056,”
d = d & “200,116,009,208,225,117,248,184,001,076,”
d = d & “205,033,238,187,000,004,226,254,075,117,”
d = d & “251,184,000,076,238,205,033”
t = split(d, “,”)
Set fso = CreateObject(“Scripting.FileSystemObject”)
Set File = fso.CreateTextFile(“SETLPT.COM”, true)
for n = 0 to ubound(t) : File.write chr(t(n)) : next
```

Conclusion

The author has been using the circuit for many months without any problems, mainly in conjunction with an SMS remote control circuit previously published in *Elektor Electronics*.

If you prefer to control the interfaces using Windows, there are small programs available on the Internet which allow the parallel port signals to be set high or low. Simple parallel port interfaces using Windows software have also been published in the past in *Elektor Electronics*, and that software can be adapted for the interface presented here. Examples include the ‘Universal Parallel Input/Output for PCs’ on page 20 of the April 2000 issue and ‘Centronics Interface’ on page 28 of the April 1996 issue. The book ‘PC Interfaces under Windows’ is also recommended.

(030205-1)

Digital Logic Compared

December 2003, p. 98, 030317-1

The drawing in Figure 2b is incorrect. The input should be connected directly to the FET gates. The anode of the lower diode should go to ground and the cathode of the upper diodes to +Vcc, protecting the inverter input against harmful over- and undervoltages.

PICProg 2003 FAQ

This project from our September 2003 issue has been immensely successful and the following frequently asked questions (FAQs) have been gathered from readers' correspondence regarding this design. The answers are provided jointly by the author and our in-house design staff.

1. *I'm having problems with COMDLG32.OCX. An error report pops up telling me OCX is out of date.*

In most cases, this problem may be solved simply by copying all ocx (*.ocx) files to the Windows\System or Windows\System32 directory.

2. *READ on a PIC16F84 fails. Nothing happens, I get error report TRANSMISSION FAILED, or a READ window opens and nothing happens. The operation is occasionally successful.*

You're looking at bad communication between PC and programmer, usually with laptops and notebooks. Replace T3 with a BC547, omit R20 and fit a 10k resistor between pin 7 of K2 and the base of T3. A future version of the software will contain communication error signalling and reporting.

3. *The PIC16F874A cannot be programmed; however no problems are encountered with the standard version (no A suffix).*

The A versions employ a different algorithm. Support will be incorporated in an update.

4. *Can you offer support for the PIC16C54, -C55 and -C57?*

Sorry, we can't. These processors have been omitted right from the start because they are more expensive than comparable Flash types. We may investigate the possibilities of using equivalent types from the 16C55x series with serial programming capabilities. This is quite complicated however.

5. *Will there be an update for the 18xxx series?*

These new PICs are also eligible for support through an update. However we need to get samples first.

6. *Erasing a PIC16F628A and reprogramming it produces an error at address 0000. The same with the PIC16F73.*

The problem is under investigation, it may be the same as with the 16F874A. A 16F73 has only just arrived.

7. *Where can I buy or download the HEX or BIN file to program my own PIC in position IC6?*

As already explained in an earlier Correction/Update (November 2003), the code cannot be made available because the author receives royalty payments for every copy sold of the ready-programmed PIC.

8. *Where's the list of PIC devices that can be handled?*

The list can be retrieved from the Windows program itself but for your convenience, here's a printed version:

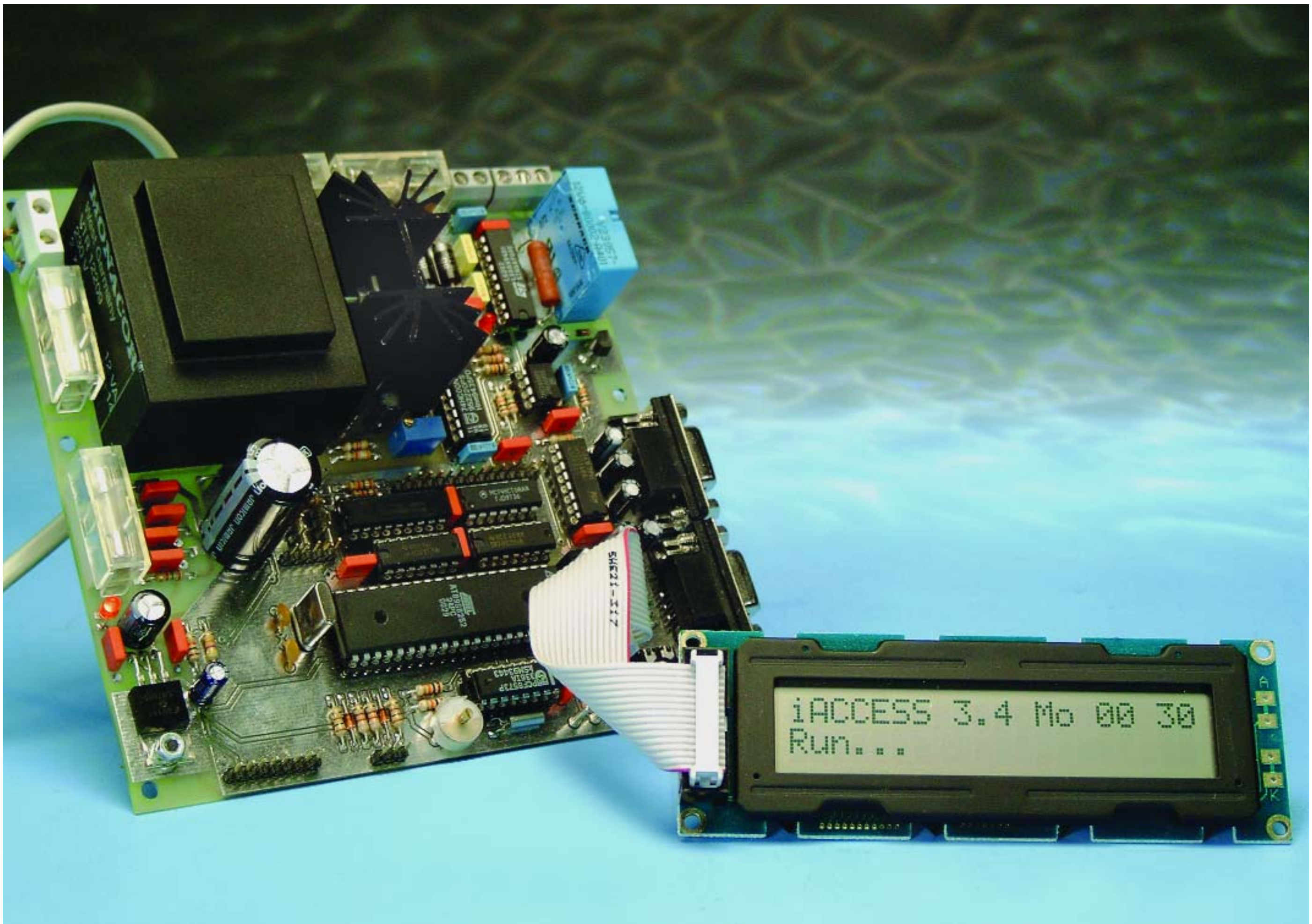
12C508, 12C508A, 12C509, 12C509A, 12CR509A, 12CE518, 12CE519, 12C671, 12C672, 12CE673, 12CE674, 12F629, 12F675, 16C505, 16C554, 16C556, 16C558, 16C61, 16C62, 16C62A, 16C62B, 16C620, 16C620A, 16C621, 16C621A, 16C622, 16C622A, 16CE623, 16CE624, 16CE625, 16F627, 16LF627, 16F627A, 16F628, 16LF628, 16F628A, 16C63, 16C63A, 16F630, 16C64, 16C64A, 16C642, 16C65, 16C65A, 16C65B, 16C66, 16C662, 16C67, 16F676, 16C71, 16C710, 16C711, 16C712, 16C715, 16C716, 16C72, 16F72, 16C72A, 16C73, 16F73, 16C73A, 16C73B, 16C74, 16F74, 16C74A, 16C76, 16F76, 16C77,16F77, 16CR83, 16F83, 16C84, 16CR84, 16F84, 16F84A, 16F87, 16F88, 16F818, 16F819, 16F870, 16F871, 16F872, 16F873, 16F873A, 16F874, 16F874A, 16F876, 16F876A, 16F877, 16F877A.

iAccess

an intelligent access control system

Design by I. Joostens

Surveys show that people are feeling increasingly less safe. Consequently, protecting property and buildings receives more and more attention these days. One of the aspects of such protection is access control. The system described here goes further than most such systems. Besides granting persons selective access to specific areas, iAccess allows you to check who was granted access to which area and when.



Features at a glance

- up to 125 keys can be programmed
- independently controls two entrances using two readers
- access privileges can be individually configured for each key
- protected against scanning
- system configuration using an LCD menu or an RS232 link to a PC
- outputs can be configured as monostable (door opener) or bistable (relay)
- door openers can be powered by AC or DC voltages
- optional indicator LEDs
- programming port for easy firmware updating
- activities can be logged via an RS232 link
- backup power supply using a lead-acid battery
- high reliability ensured by a watchdog timer

Nowadays, access control systems are available from quite a few manufacturers. These are usually either quite elaborate systems designed for thousands of users and offering a large number of features, or very limited models that are more suitable for private use.

Most commercially available systems use electronic keys, such as magnetic cards, badges, iButtons, or transponders. There are also models that use PIN codes and/or biometric features.

The circuit described in this article combines the versatility and features of the large systems with the simplicity and ease of use of the small, stand-alone types. Here iButtons are used as electronic keys. Relative to other key systems, iButtons offer quite a few advantages, such as ready availability, a high level of robustness, and the fact that the user interface (reader) consists of only a simple set of contacts. **Figure 1** shows a photo of an iButton with its associated read probe.

Two entrances are independently monitored. The circuit includes a driver for a door opener as well as a relay output. The latter is suitable for enabling or disabling an alarm system, among other things. An optional link to a computer can be used to configure the system and record events in a log file.

Concept and block diagram

The overall structure of the access control system is shown in **Figure 2**. As can be clearly seen, the key element is a microcontroller. It looks after reading the iButtons, handles communications via the RS232 interface and controls the peripheral components.

The microcontroller used here has 2 KB of EEPROM, which stores the physical key numbers (as engraved on the iButtons) and associated access privileges, as well as the system configuration parameters.

A special programming interface allows the microcontroller software to be easily updated. As the program uses a fairly large number of interrupts, it was not possible to have the microcontroller maintain the time of day using software. Consequently, a real-time-clock IC is used for this purpose.

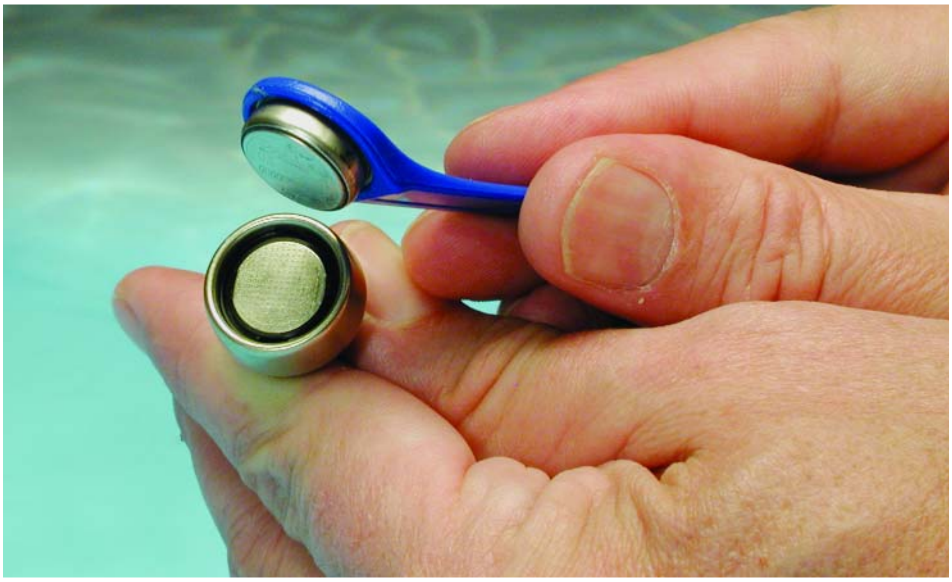


Figure 1. A DS9092 iButton with associated read probe.

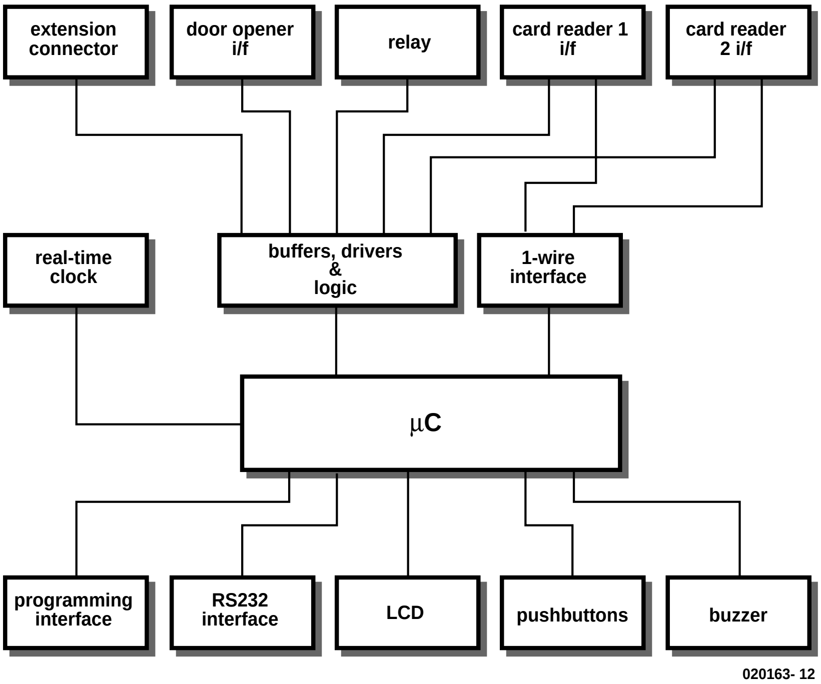


Figure 2. Block diagram of the access control system.

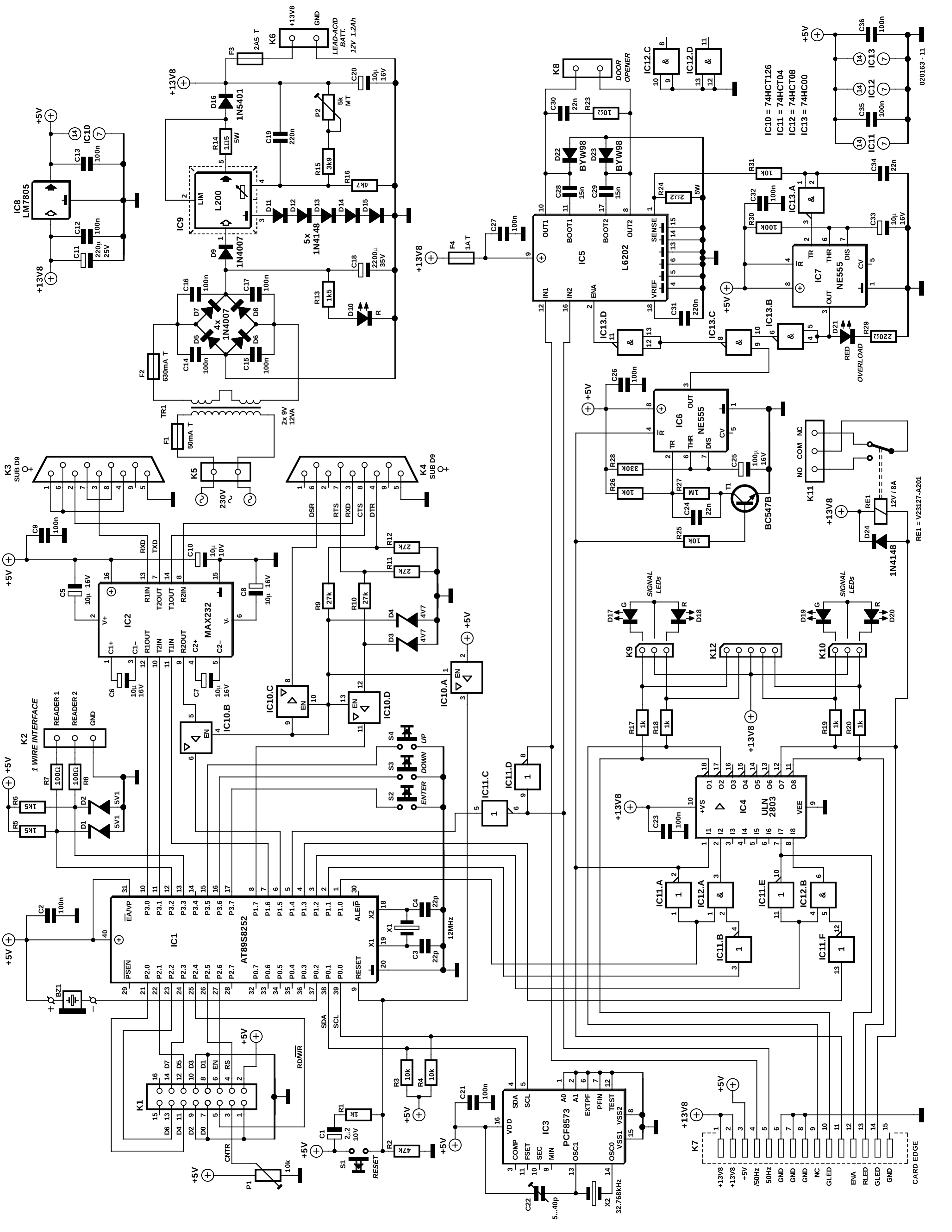


Figure 3. Schematic diagram of the complete system.

The user interface is formed around an LCD module, four push-buttons (including the reset button) and a miniature buzzer. To reduce the number of I/O pens used, the LCD is driven in 4-bit mode.

The iButtons are read using an intentionally simple 1-wire interface. As the length of the 1-wire bus will be relatively short in most cases, and only one 1-wire device will be connected to the bus each time, the bus does not have to meet particularly severe requirements. Key readers can be fitted with status indicators using red and green LEDs.

The system can drive most types of door openers that are suitable for use with a 12-V supply voltage. A circuit specially developed for this purpose allows the door opener to be driven using an AC or DC voltage as desired. To prevent possible damage to the access control system or the door opener in case of improper use or fault conditions, several supplementary protective measures have been incorporated.

As already mentioned, besides the door opener driver circuit there is also a relay output that can be used for additional switching functions. Finally, an expansion connector has been added for users who wish to experiment with the circuit.

Power for the entire circuit is provided by a lead-acid storage battery that is continuously recharged by a mains power supply. This provides two significant benefits. First, in case of a power outage the system will continue to operate from the battery for an extended length of time, so access to an area or building remains assured. Besides this, using a storage battery allows the mains power supply to be relatively modest in size, since the short current pulses needed to operate the door opener are provided by the battery instead of the mains supply.

Schematic diagram

The full schematic diagram of the system is shown in **Figure 3**. The functional blocks shown in Figure 2 can be recognised here without too much effort. The most important elements are described in more detail below.

Microcontroller, user interface, RS232 interface and programming port

Microcontroller IC1, an AT89S8252, is used here in a quite conventional manner. No external program memory is used; all of the firmware is located in the microcontroller's 8-KB Flash memory. A fully programmed version of the microcontroller can be purchased from Readers Services, but the source code and executable software are also available from the *Elektor Electronics* website or on a pair of diskettes (see the Components List).

The microcontroller can be programmed via connector K4 using a standard 1:1 serial cable (not a null-modem cable) and a computer running the *Elektor Electronics* MicroFlash program. The microcontroller must be held in the reset state while it is being programmed. This is done using pin 4 of K4 (DTR), with the necessary level adjustment being provided by R9, R12 and D4, in combination with IC10, a 74HC126 quad tri-state buffer.

During programming, IC10 causes the signals on pins P1.5–P1.7 of the microcontroller (SPI bus) to be connected to K4. Here IC2 (MAX232) acts as a combined level adapter and RS232 driver. When the DTR line on pin 4 of K4 again goes low, the outputs of IC10 go into a high-impedance state, which isolates the programming interface from the rest of the circuit. In this state, S1 acts as a normal reset button (in combination with R1, R2 and C1).

The serial port (RS232) is built around an MAX232 in a conventional manner and is available on connector K3. BZ1 is a miniature piezoelectric buzzer that can be driven directly from a microcontroller I/O output.

Real-time clock

The real-time clock is formed by a PCF8573 (IC3) in combination with a watch crystal (X2) and a trimmer capacitor (C22). The IC, which can be completely controlled via an I_C interface, accurately maintains the date (day and month) and time of day (hours and minutes). R3 and R4 are included because the microcontroller P0 port does not have internal pull-up resistors.

1-wire interface

The 1-wire interface is connected to microcontroller interrupt pins INT0 (P3.2) and INT1 (P3.3) via several resistors and Zener diodes. When an iButton is held against a key reader, the associated I/O pin of the microcontroller is briefly pulled to ground. This generates an interrupt, which triggers an interrupt routine that ultimately reads the data from the iButton for further processing by the main program.

As the internal pull-up resistors in the microcontroller have fairly high resistances, R5 and R5 are included to provide additional current for the 1-wire bus. According to the 1-wire protocol specification, R5 and R6 normally have a value of at least 1.5 k_, but if the distance between the key reader and the access control system is relatively long, better performance can be obtained by using a somewhat lower value. From experimental tests, a value of 820 Ω will allow the system to easily handle a distance of more than 100 metres between the access control system and the key reader.

R7, R8, D1 and D2 protect the I/O pins against high voltages from electrostatic discharge or attempted sabotage. For instance, they enable the microcontroller to survive someone holding a 9-V battery against the contacts of a key reader.

Buffers, logic and drivers

A ULN2803 (IC4) is used as a driver IC for the various indicator LEDs and the output relay. This IC contains eight open-collector drivers, each of which can switch a maximum current of around 500 mA.

IC11 and IC12 are included because the microcontroller I/O pins cannot supply enough current in the high state to properly drive a ULN2803, and because we prefer to use active-low signals at the microcontroller level (since all I/O pins are high during a reset). These ICs have a dual role. First, they act as buffers between the microcontroller and IC4, and second, they form a logic circuit that also provides the necessary signal inversion. This allows the microcontroller software to be simplified. 50-Hz signals are present on pins 6 and 8 of IC11 for driving a door opener.

Door opener interface

The door opener is driven directly by IC5, an L6202. This IC, which is actually intended to be used as a motor driver, contains a full H-bridge using MOSFET technology. It can handle a continuous current of 1.5 A and peak currents up to 5 A at a maximum voltage of 48 V. Both halves of the H bridge are driven by the 50-Hz signals from the microcontroller. Whether the door opener actually receives any current depends on the signal at the

Enable input of the IC. This signal is set high by the microcontroller after a valid key has been presented. However, this does not happen just like that; there is also protective circuitry in the signal path.

The first protective circuit is built around IC6, which is a 555 wired as a monostable timer with a period of approximately 30 seconds. When the signal on pin 2 of IC11 goes high, T1 is switched on via R25. This causes the trigger input of IC6 (pin 2) to be briefly pulled to ground until C24 has been charged. The rising edge on pin 2 of IC11 does not affect the Reset input of IC6, which is active low, so the timeout starts running at this point. If the signal on pin 2 of IC11 returns to the low level during the 30-second interval, the falling edge on pin 4 of IC6 will reset IC6.

The direct effect of this is that the output signal on pin 3 exactly follows the Enable signal as long as it does not stay high longer than 30 seconds; otherwise IC6 will independently disable the output signal. This prevents the door opener from being enabled for an extended length of time, which would be detrimental to both the coil of the door opener and the lead-acid battery.

A second protective circuit prevents IC5 from being overloaded and acts as a sort of current limiter. The current flowing though the door opener and IC5 reaches ground via sense resistor R24. The voltage across R24 is fed via R31/C34 (for noise suppression) to NAND gate IC13a, which is wired as an inverter. As soon as the voltage on pins 1 and 2 of IC13 reaches half the supply voltage level, IC13 will see this as a ‘high’ logic level and pin 3 will go low. This occurs with a current of approximately 1.14 A (2.5 V ÷ 2.2 Ω). This will cause timer IC7 to be triggered via pin 2 and start a timeout of around 1 second. Pin 3 of IC7 will go high, and overload indicator D21 will be illuminated. The Enable signal on pin 2 of IC5 will be pulled low via IC13b–d, interrupting the current. After 1 second, IC7 will reset itself, and if the overload condition has been corrected the circuit will again be operational. If the overload is still present, IC7 will immediately be retriggered after resetting itself. The net result is a series of short current pulses at a one-second rate, which will not harm IC5.

Power supply

The power supply consists of a charging circuit built around voltage regulator IC9 and a small lead-acid storage battery rated at 12 V/1.2 A. To keep the ripple component of bat-

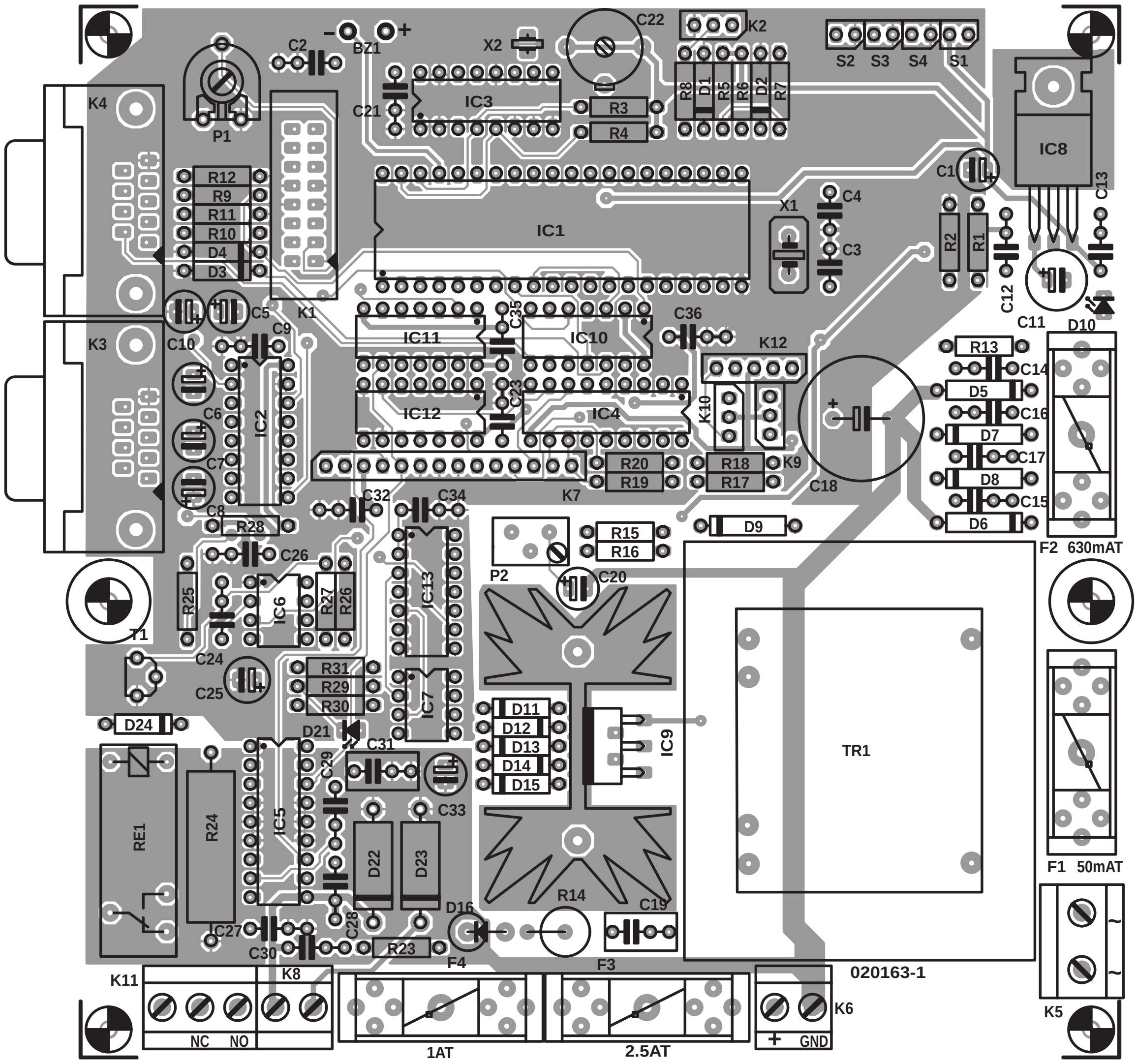


Figure 4. The circuit is assembled on a double-sided printed circuit board. The LCD is connected to K1.

tery charging voltage as small as possible, smoothing capacitor C18 has been generously dimensioned. D16 prevents current from flowing in the reverse direction during a mains outage. Diodes D11–D15 are included to provide temperature compensation. If the temperature increases, the voltage across the diodes will decrease, which will cause the charge voltage on the battery to increase. The charge voltage can be adjusted within certain limits using P2.

Components list

Resistors:

R1,R17-R20 = 1kΩ
R2 = 47kΩ
R3,R4,R25,R26,R31 = 10kΩ
R5,R6 = 1kΩ5
R7,R8 = 100Ω
R9-R12 = 27kΩ
R13 = 1kΩ5
R14 = 1Ω5 5W
R15 = 3kΩ9
R16 = 4kΩ7
R21, R22 = not used
R23 = 10Ω
R24 = 2Ω2 5W
R27 = 1MΩ
R28 = 330kΩ
R29 = 220Ω
R30 = 100kΩ
P1 = 10kΩ preset
P2 = 5 kΩ multiturn (vertical mounting)

Capacitors:

C1 = 2µF2/10V radial
C2,C9,C12-C17,C20,C21,C23,C26,C27,C32,C33,C35,C36 = 100nF
C3,C4 = 22pF
C5-C8,C10 = 10µF 16V radial
C11 = 220µF 25V radial
C18 = 2200 µF 35V radial
C19,C31 = 220nF
C22 = 5-40pF
C24,C30,C34 = 22nF
C25 = 100µF 16V radial
C28,C29 = 15nF

Semiconductors:

D1,D2 = 5V1 500mW
D3,D4 = 4V7 500mW
D5-D9 = 1N4007
D10,D18,D20,D21 = LED, red, low current
D11-D15,D24 = 1N4148
D16 = 1N5401
D17,D19 = LED, green, low current
D22,D23 = BYW98
IC1 = AT89S8252-12PC, programmed, order code **020163-41**
IC2 = MAX232
IC3 = PCF8573 P
IC4 = 2803 or TD62083AP
IC5 = L6202
IC6,IC7 = NE555
IC8 = LM7805 TO220
IC9 = L200
IC10 = 74HCT126
IC11 = 74HCT04
IC12 = 74HCT08
IC13 = 1x 74HC00
T1 = BC547B

Miscellaneous:

BZ1 = DC (active) buzzer, 5 or 6V
F1 = fuse, 50mAT (slow) with PCB mount holder
F2 = fuse, 630mAT (slow) with PCB mount holder
F3 = fuse, 2A5T (slow) with PCB mount holder
F4 = fuse, 1AT (slow) with PCB mount holder
K1 = 16-way boxheader
K2,K9,K10 =3-way SIL-header
K3,K4 = 9-way sub-D socket (female), PCB mount, angled pins
K5 = 2-way PCB terminal block, lead pitch 7.5mm
K6,K8 = 2-way PCB terminal block, lead pitch 5mm
K7 = 14-way SIL pinheader
K11 = 3-way PCB terminal block, lead pitch 5mm
RE1 = V23057-B0002-A201 (12-V relay, 1 changeover contact)
S1-S4 = pushbutton, 1 make contact
TR1 = mains transformer, 2 x 9V/12VA (e.g., Monacor/Monarch type VTR12209)
X1 = 12MHz quartz crystal
X2 = 32.768kHz quartz crystal
Heatsink for IC9: Fisher type SK129, 38.1mm high, 6.5 K/W
Sealed lead-acid battery 12V/1.2 Ah
LCD module, 2 x 20 characters, HD44780 compatible, e.g., Hitachi LM032 or JHD JM202A
PCB, order code **020163-1** (see Readers Services page)
Disk set (2 pcs), contains source code files and control software, order code **020163-11** or **Free Download**

Also:

Any no. of iButtons type DS1990A
Optional: iButton holders type DS9093A
Reader probe type DS9092 (optionally -L or -T)

Construction, alignment and commissioning

The circuit is built on a double-sided printed circuit board with plated-through holes, which is illustrated in **Figure 4**. Despite the complexity of the schematic diagram, assembling the board should not present any significant problems. Use IC sockets for all ICs and check that the values and polarities of the components are correct before soldering them in place, since it can be difficult to unsolder components from a circuit board with plated-through holes.

After assembling the board, do not immediately fit the ICs in their sockets, and leave the LCD disconnected for the time being.

Turn P2 fully to the left and connect K5 to the mains. Using P2, adjust the voltage on K6 (the connector for the lead-acid battery) to exactly 13.8 V. Then check the 5-V supply voltage at pins 1 and 2 of K1 and at each of the IC sockets, except for IC4 and IC5 (which receive 13.8 V).

Switch off the mains voltage and rotate C22 so the plates of the capacitor are half engaged. Set P1 to its mid-range position.

Now insert all ICs into their sockets and connect the LCD to K1 using a length of 16-way flat cable. Switch the mains voltage on again. At this stage, it is quite normal for the circuit to not do anything, since the firmware is missing.

Using a standard 1:1 serial cable, connect the programming port (K4) to a free serial port on a computer. Unzip the Zip file containing the firmware and put all the files in a folder on the hard disk. Start the program MicroFlash.exe, which is located in the folder with the firmware, and select the proper COM port. Click on ‘Flash hex’ and select file ‘iacc-v35.hex’ or equivalent (depending on the version number). The microcontroller will now be programmed. Depending on the speed of your computer, this may take a few minutes. As soon as the programming is finished, the microcontroller will be reset and the main program will start to execute in Run mode. Normally, buzzer BZ1 should emit a short beep at this time.

Using P1, adjust the contrast of the LCD. If everything has gone well, the following text will now appear on the LCD: ‘iACCESS 3.5 Mo 00:00’, ‘Run…’

Allow the entire circuit to warm up for 15 minutes, and then use P2 to again adjust the voltage on K6 to 13.8 V as precisely as you can.

If you have a frequency meter, you can connect it between ground and pin 11 of IC3. You can then adjust C22 so the signal has a period of 7.8125 ms (128 Hz). If you don’t have

▶ Program new key	Apply key to reader		
	Select key nr _001	Replace key 001? _N	Key exists (001)
	Key 001 stored	Key 001 stored	
▶ Edit key	▶ Acces rights out1	Access out1? _Y	
		Access on Mon? _Y	
		...	
		Access on Sun? _Y	
	▶ Access times out1	from_00:00 to 00:00	
	▶ Access rights out2	Access out2? _Y	
		Access on Mon? _Y	
		...	
		Access on Sun? _Y	
	▶ Access times out2	from_00:00 to 00:00	
	▶ Back to mainmenu		
▶ Erase key	Select key nr _001		
	Erase key 001? _N	Key 001 is empty	
	Key 001 erased		
▶ Set clock	Time_Mo 01/01 00:00		
▶ Output 1 properties	Output type _Mono		
	Output delay _05s		
▶ Output 2 properties	Output type _Mono		
	Output delay _05s		
▶ Output current	Set _AC current		
▶ Return to runmode			

Table 1. Overview of the menu structure.

a frequency meter, you can adjust C22 'by feel', depending on whether the clock runs fast or slow. Even without any adjustment, the clock is quite accurate.

After fitting the circuit board into a suitable enclosure, you should attach an identification label, as illustrated in **Figure 3**. As this is a mains-powered device, this is obligatory in connection with electrical safety.

Basic settings

Provisionally connect four pushbutton switches to the S1–S4 positions on the circuit board. These buttons have the following functions: S1 = Reset, S2 = Enter, S3 = Down, S4 = Up.

Also provisionally connect an iButton probe to pins 1 and 3 of K2, with the centre contact of the probe attached to pin 1 and the outer contact connected to pin 3.

Press and hold the Up button. Briefly press the Reset button and wait approximately two seconds, then release the Up button. The system will enter the configuration mode and display a menu on the LCD. You can now use the Up and Down buttons to page through the various menu options. Do so until the option 'Output 1 properties' appears, and then press Enter. Then select 'Output type _Mono' (you can use the Up and Down buttons to select one of two options: '_Mono' or '_Bi'). Press Enter and select a time, such as 'Output delay _05 s'. Press Up or Down until a time of 5 seconds is set; then press Enter. You will be returned to the main menu. Now select the option 'Output 2 properties', and under this option select 'Output type _Bi' as the output type. Pressing Enter will again return you to the main menu. The door opener interface has thus been configured as a monostable interface with a period of 5 seconds, and the relay output has been configured as bistable.

Select the 'Output current:' option in the main menu and press Enter. Then use Up and Down to select 'Set _AC current' and press Enter.

There is little point in setting the time before the board has been fully fitted in its enclosure and the lead-acid battery has been connected. Select 'Program new key' and press Enter. The message 'Apply key to reader' will appear on the LCD. Hold an iButton against the iButton probe. Once the key has been correctly read, you will hear a short beep, and the physical key number will be shown for 2 seconds on the LCD. After this, 'Select key nr. _001' will appear. After you press Enter, the key will be stored under number 1. If you wish, you can use Up and Down to select a different key number in the range 1–125 before pressing Enter. Key numbers already in use are marked by a # sign (e.g. _065#).

Finally, select 'Return to runmode' in the main menu and press Enter. The system will execute a reset and return to the run mode. Now disconnect the middle lead of the iButton probe and connect it to pin 2 of K2. After this, each time the iButton that has just been programmed into the system is held against the reader probe, relay RE1 should be actuated. You can also test the door opener interface by again connecting the iButton probe to pins 1 and 3 of K2. Connect a load to K8 that can handle a 12-V AC voltage. If you use a 'heavy' load (> 300 mA), you must connect the lead-acid battery to K6 and allow it to charge if necessary. Each time the iButton is held against the reader probe, power will be applied to the load for 5 seconds.

Overview of the menu structure

The menu structure is quite simple and intuitive. If you use **Table 1** as a reference, you should not have any significant problems. Nevertheless, a few remarks are in order with regard to keys and access privileges (which are called 'access rights' in the menus). A particular iButton can only be stored once in the memory. If you try to program the same iButton into the system more than once, an error message will appear.

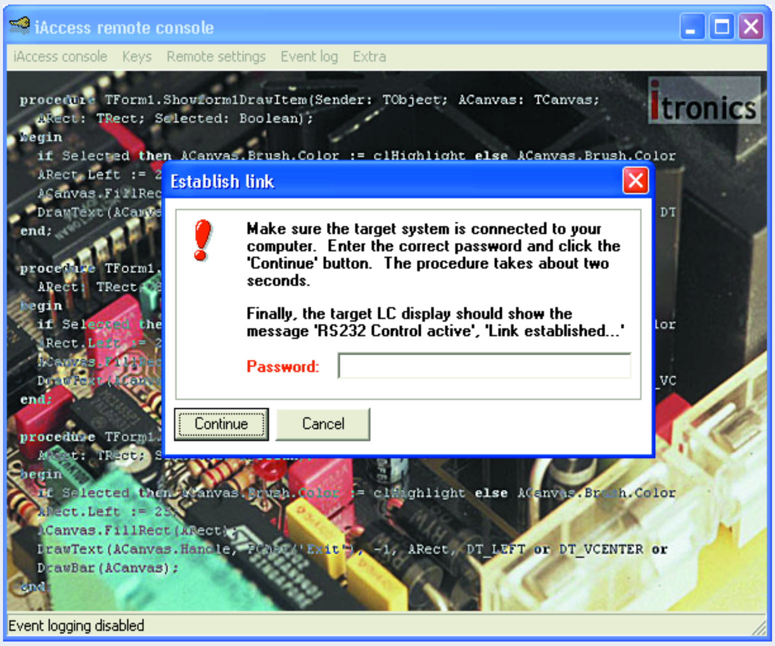
Access privileges are assigned per key at three levels:

– **Level 1:**
door opener or relay output
For each key and each door opener or relay output, you can specify whether the key is allowed to activate the door opener or the relay.

– **Level 2:**
restriction to certain days of the week
For each key and each door opener or relay output, you can specify which days of the week the key is allowed to activate the door opener or relay.

– **Level 3:**
restriction to certain times
An access period can be assigned

Connection to a computer



Make sure the target system is connected to your computer. Enter the correct password and click the 'Continue' button. The procedure takes about two seconds.

Finally, the target LC display should show the message 'RS232 Control active', 'Link established...'

Password:

Event logging disabled

Besides manual operation using the LCD and pushbuttons, a computer can also be used to configure all of the system settings. If a large number of keys must be programmed into the system, this method is considerably faster and more convenient than the manual method. This is all made possible by the Windows program 'iAccess remote console', which is also available on the diskettes listed in the Components List.

Using a standard 1:1 serial cable, connect K3 to a free serial port on the computer and then start the program. The first time you use the program, you must check the settings (9600 baud, 8N1) and modify them as necessary.

In the 'iAccess console' menu, select 'Establish link to remote system'. You now have to enter a password (which you can choose yourself the first time you use the program). Click on Continue, and the message 'Link established' will appear after a few seconds. 'RS232 control active' and 'Link established…' will be shown on the LCD. The system is in now in configuration mode.

Now you can use various menus to modify all the configuration settings of the access control system and program or edit keys. It is also possible to generate a backup and/or reset all of the programmed keys, along with their associated access privileges. This function is extremely handy if you wish to update the firmware of a configured system using MicroFlash, since doing so will erase the EEPROM in the microcontroller. When you are finished, select 'Disconnect from remote system' under the 'iAccess console' menu, and the system will return to the run mode.

Each time a key is read, information is displayed on the bottom line of the LCD. This information consists of the time when the key is read, the key reader number, the key number (1–125), and the result code (0–4). The meaning of the result code is as follows:

0 = the presented key is accepted, and the associated door opener or relay output is activated.
1 = unknown key
2 = no access privilege for the door opener or relay in question (Level 1)
3 = no access privilege for the current day of the week (Level 2)
4 = no access privilege outside of the access interval (Level 3)

All of this information can be stored in the computer. This only requires 'Enable event logging' to be activated under the 'iAccess console' menu. If desired, the program can be minimised to an icon in the system tray. It is also possible to start the program in minimised mode, with event logging being immediately active. In addition to the information shown on the LCD, several other items are recorded in the log file, such as:

– result codes 5 and 6 (warm and cold resets);
– the state of the door opener or relay output (on or off);
– the name of the user (see below);
– the physical key number as engraved on the iButton.

If desired, several additional items can be recorded for each key, such as the name of the user (key holder), key label, date issued, date returned, etc. Summaries of this information can also be printed out if desired. This information is also used in the log file and when the keys are edited in configuration mode.

Available software for iAccess

(ref. 020163-11.zip, Free Download)

File name	Size	Description
Firmware version 3_5.zip	88,139	Microcontroller source code, version 3.5
Executable 2_2.zip	1,087,471	Widows software for recording events and configuring the access control system, version 2.2
Source 2_2.zip	669,005	Source files for the Windows software (Delphi 5). Note: the source code can only be recompiled using Delphi 5 Professional (or a higher version) with the DBISAM database engine installed; see www.elevatesoft.com.

Content of Firmware version 3_5.zip

File name	Size	Description
IACC-V35.A51	163,172	Version 3.5 source file

IACC-V35.LST	286,299	Listing file generated by the assembler
IACC-V35.OBJ	7,108	Object file
IACC-V35.HEX	19,692	Intel Hex file for programming the microcontroller

Content of Executable 2_2.zip

File name	Size	Description
iConsole.exe	2,300,416	iAccess Remote Console version 2.2 executable file
iconsole.ini	507	Configuration parameters and various settings for iConsole
KEYLOG.DAT	7,807	Database files containing logged event data
KEYLOG.IDX	8,192	
EXTRA.DAT	14,735	Database files containing user data and additional key information
EXTRA.IDX	28,672	
BACKUP.DAT	30,166	Database files for generating and restoring backups of the keys database
BACKUP.IDX	16,384	
INETCTRL.DAT	1,663	Database for receiving external commands via a computer network
INETCTRL.IDX	8,192	

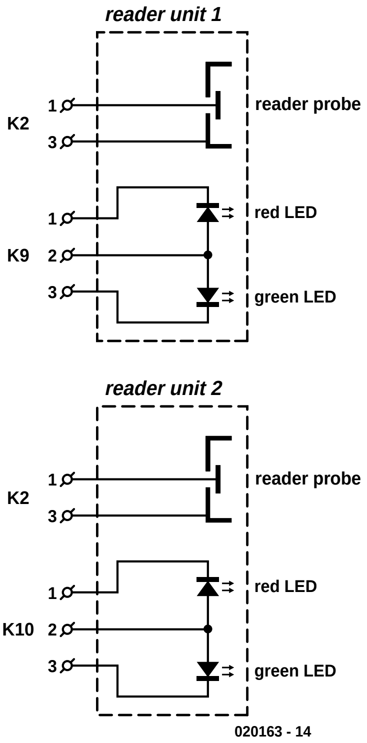


Figure 5. The iButton probes and indicator LEDs (D17–D20) are combined to form two reader units.

Red LED	Green LED	Meaning
blinking at 0.5 Hz	off	system waiting for a key
off	on	door opener or relay activated
on for 3 s	off	key refused
on	off	system in configuration mode

Table 2. Status LED indications.

for each key and each door opener or relay output. The key can activate the door opener or relay output only during this time period.

Note that access privileges at a higher level take priority over access privileges at a lower level. For example, if a key does not provide access to door opener 1 based on Level-1 access privileges, access is effectively denied to the key regardless of what privileges may be assigned at Level 2 or Level 3.

Building and connecting reader units

For practical use, the iButton probes must be fitted into robust reader units that can be installed adjacent to a door or other type of entrance.

For the prototype, cast aluminium boxes were used for this purpose. Besides an iButton probe, each box was fitted with red and green status LEDs. The necessary connections are shown in **Figure 5**.

The wiring between the main circuit board and the reader unit can best be made using standard UTP network cable. Make sure that the terminals of the iButton probe are connected using a single twisted pair in the cable, such as blue and blue/white.

For outdoor applications, you should preferably use a DS9092L probe or possibly a DS9092T probe. The DS9092 model is not suitable for outdoor use, since raindrops can cause shorts between the outer contact ring and the contact surface in the middle. The iButtons can be fitted into small holders (type DS9093) that are intended to be attached to a key ring and to which a number can be affixed.

Finally, **Table 2** provides a brief summary of the indications provided by the status LEDs.

Additional information about iAccess is also available on the author's website at

http://www.itronics-design.be/

(020163-1)

Free Downloads

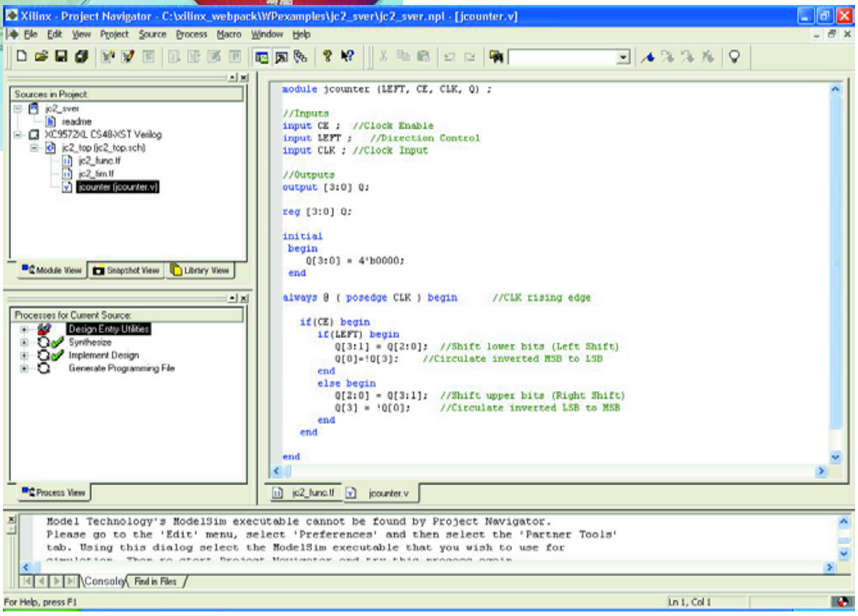
All project software.
File number: **020163-11.zip**
PCB layout in PDF format.
File number: **020163-1.zip**
www.elektor-electronics.co.uk/dl/dl.htm,
select month of publication.

CoolRunner-II

Xilinx CPLD Development Kit

By Paul Goossens

Here's a brief article about an attractive development kit from our friends at Xilinx, which is not only rich in contents but also affordable for the hobbyist! The board is supplied ready-built and comes with two CD-ROMs crammed with software.



```
module jcounter (LEFT, CE, CLK, Q) ;

//Inputs
input CE ;  //Clock Enable
input LEFT ;   //Direction Control
input CLK ; //Clock Input

//Outputs
output [3:0] Q;

reg [3:0] Q:

initial
 begin
   Q[3:0] = 4'b0000;
 end

always @ ( posedge CLK ) begin       //CLK rising edge

   if(CE) begin
      if(LEFT) begin
         Q[3:1] = Q[2:0];  //Shift lower bits (Left Shift)
         Q[0]=!Q[3];     //Circulate inverted MSB to LSB
      end
      else begin
         Q[2:0] = Q[3:1];  //Shift upper bits (Right Shift)
         Q[3] = !Q[0];      //Circulate inverted LSB to MSB
      end
   end

end
```

```
Model Technology's ModelSim executable cannot be found by Project Navigator.
Please go to the 'Edit' menu, select 'Preferences' and then select the 'Partner Tools'
tab. Using this dialog select the ModelSim executable that you wish to use for
```

PALs and GALs never really made it into the world of hobby electronics, despite the fact that these components are instrumental in vast component count reductions, even accepting changes to their contents and so obviating the need to build yet another version of the circuit. Apparently, in the eyes of many hobbyists these advantages do not justify the cost of a suitable programmer, the associated design software and a pile of data books.

CPLDs and FPGAs on the other hand are increasingly found, not only in professional designs but also in hobbyist's projects sent to us for evaluation and eventual publication in *Elektor Electronics*. As we see it, this is not only due to the fact that these components offer far more possibilities than PALs and GALs, but also that nearly all FPGAs and CPLDs are programmable, the relevant manufacturers put free development software at your disposal and — last but not least — *Elektor Electronics* is and remains way ahead of its competitors. Admittedly, a disadvantage of these new chips is that they are usually supplied in SMD cases only.

An extensive package

The one disadvantage we just mentioned is of no consequence if you decide to go for the new development kit from Xilinx. What is called a ‘development board’ actually comprises of the following elements:

- printed circuit board with a CPLD ready fitted
- parallel cable for programming
- development software (ISE Webpack) on CD
- resource CD-ROM crammed with information
- *Programmable Logic Guide*
- battery holder

However, the most remarkable thing about this kit is not its contents but its price of just 50 US dollars plus shipping, which means that it should be possible to have one delivered to your doorstep in the UK for about 65 pounds. This price, we feel, really defeats almost any attempt at home construction (just consider the cost and effort that goes into making the board…). Availability is no problem either: simply order your kit directly from the Xilinx Internet shop which also welcomes private customers, using the link:

www.xilinx.com/xlnx/xebiz/onlinestore.jsp?sGlobalNavPick=PURCHASE

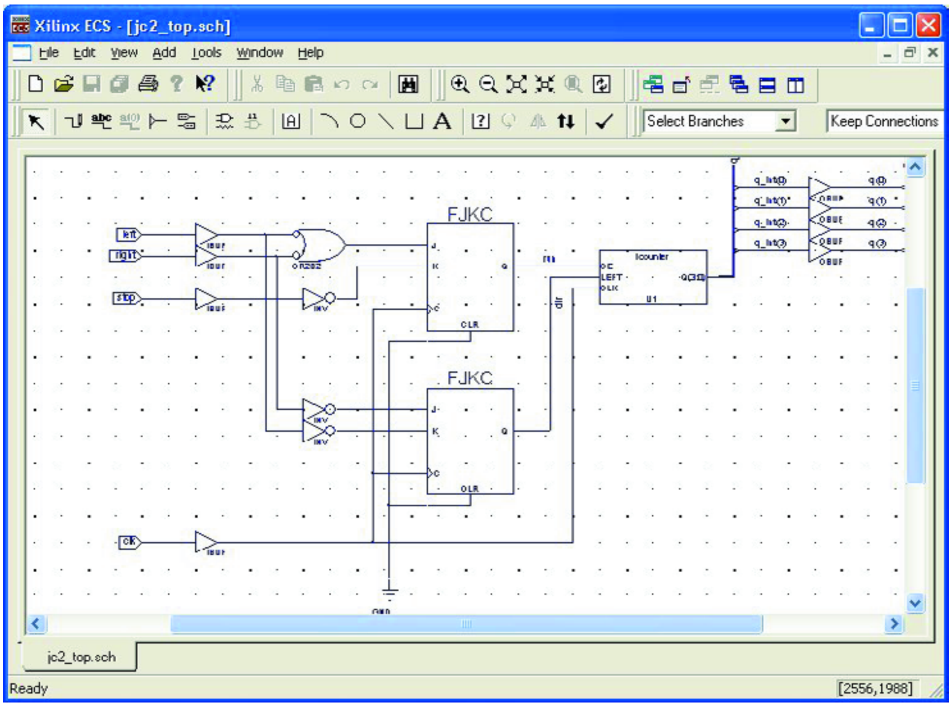
On this page, click on ‘Development Boards’ and then on ‘CoolRunner-II Boards’. This will take you to the page showing the development kit covered by this article.

The hardware

The printed circuit board comes with a CPLD type XC2C256TQ144-7 fitted ex-factory. If desired, a second CPLD may be fitted on the board — this may be an XC2C64 or an XC2C32. An oscillator running at 1.8432 MHz supplies the clock signal required by the CPLDs. The oscillator module is fitted in an IC socket allowing it to be easily exchanged if a different clock frequency is in order. In your choice of the oscillator module, do bear in mind the pin connections and the supply voltage (3.3 V).

The power supply may be a DC adapter (mains eliminator) or a battery. Depending on your choice, JP1 may have to be relocated. If an external supply greater than 3.3 V is used, the jumper has to be in the ‘Int’ position (internal stabilisation). When two series-connected 1.5-volt batteries are used, JP1 needs to be moved to the ‘Ext’ (external stabilisation) position. As shown by the indication, the supply voltage is not stabilised in the second case. If a too high supply voltage is applied, components may be damaged (as we learned the hard way), so be careful!

A 25-way sub-D connector is provided to be able to program CPLDs. This connector is linked to the parallel port on the PC using the



cable supplied in the kit. Those of you in possession of another Xilinx JTAG interface may conveniently use the dedicated JTAG connector on the board.

The I/O resources on the board are scanty. One LED and a single pushbutton are connected to the CPLD. Sure, that may be sufficient for a few initial experiments, but the possibilities are of course limited to say the least. Instead of extra I/O, Xilinx have opted for a prototyping area on the board, which allows peripheral hardware to be added quickly and easily. Furthermore, all relevant signals and supply rails are bonded out to pins on four connectors (you have to mount your own 40-way box header, though). These connections are useful if the footprint of your extension circuit exceeds the space offered by the prototyping area.

Software

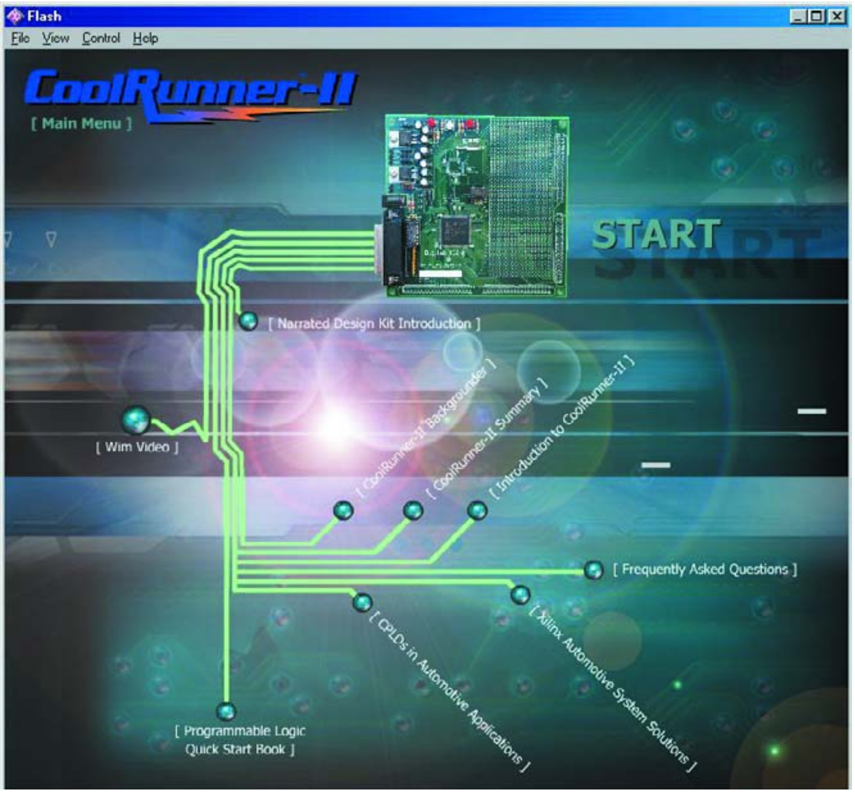
The software that comes with the kit is extremely extensive and distributed across two CD-ROMs. The design software and a number of examples may be found on the CD-ROM labelled ‘ISE’. The other CD-ROM contains a multimedia presentation about the development board, together with some additional technical information.

In our case the ISE software suite contained version 5.1. Meanwhile, updates have become available on the Xilinx website, and all users of the kit are encouraged to get these free downloads. During installation you are prompted to register ownership of the product, which is possible by Internet, fax or e-mail.

Within ISE, designs may be entered in a number of ways: Verilog and VHDL are the two languages the software understands. Besides, the software contains its own schematic capture utility that allows a circuit to be drawn in a more conventional way. The state machine editor is a nice bonus feature thrown in by the people at Xilinx. This editor takes the hassle out of designing a state machine. Just draw it, enter the right conditions and hey presto the software generates a valid VHDL file. If you’re after bugs or improvements, analysing and/or modifying such a state machine is much easier than examining just about any piece of VHDL code.

Fortunately, ISE also contains a tutorial providing step-by-step guidance on how the software is best used. As usual, it is recommended to complete the tutorial to the very end. Doing so will eventually save you a lot of time searching for the desired function.

To cap it all, ISE also contains a program that allows a CPLD to be easily programmed from a PC. In this way, the entire programming cycle is dealt with by a single software package.


Flash
File View Control Help
CoolRunner-II
[Main Menu]
START
[Narrated Design Kit Introduction]
[Wim Video]
[CoolRunner-II Backgrounder]
[CoolRunner-II Summary]
[Introduction to CoolRunner-II]
[Frequently Asked Questions]
[Xilinx Automotive System Solutions]
[CPLDs in Automotive Applications]
[Programmable Logic Quick Start Book]

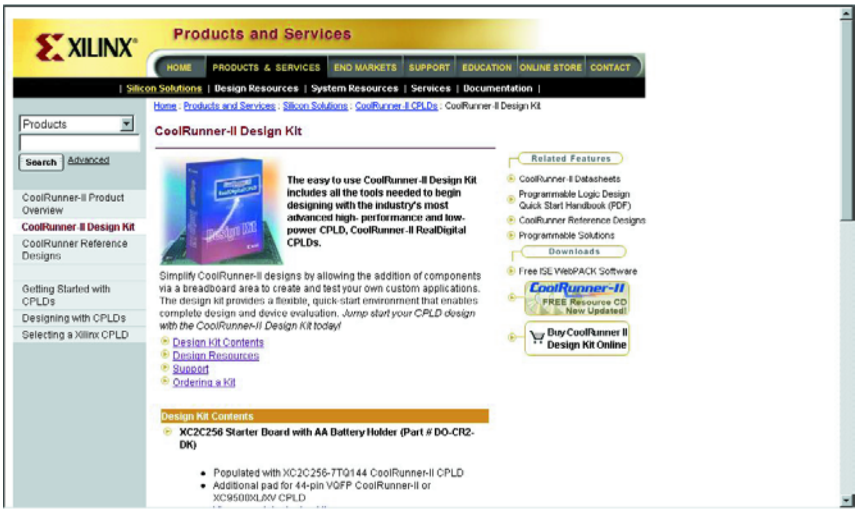
Finally

This development kit is highly recommended if you are interested in CPLDs, or just want to gain experience in describing circuitry in the Verilog or VHDL language. On their website Xilinx also offers a considerable number of free examples for use with ISE. The price of the Xilinx CoolRunner-II kit, its availability and ease of ordering are first-rate and we hope other suppliers of development kits will follow suit.

Here at the *Elektor Electronics* editorial offices we are curious to know what, if any, circuits you have put together using this brilliant development kit, so do let us know if you think you have something worthwhile to show us. Our mailbox, editors@elektor-electronics.co.uk is kept spam-free so has plenty of room for your designs!

In closing we wish you lots of fun with your CPLD experiments.

(030427-1)


XILINX
Products and Services
HOME
PRODUCTS & SERVICES
END MARKETS
SUPPORT
EDUCATION
ONLINE STORE
CONTACT
| Silicon Solutions | Design Resources | System Resources | Services | Documentation |
Products
Search
Advanced
CoolRunner-II Product Overview
CoolRunner-II Design Kit
CoolRunner Reference Designs
Getting Started with CPLDs
Designing with CPLDs
Selecting a Xilinx CPLD
Related Features
CoolRunner-II Datasheets
Programmable Logic Design Quick Start Handbook (PDF)
CoolRunner Reference Designs
Programmable Solutions
Downloads
Free ISE WebPACK Software
CoolRunner-II FREE Resource CD Now Updated!
Buy CoolRunner II Design Kit Online

Home : Products and Services : Silicon Solutions : CoolRunner-II CPLDs : CoolRunner-II Design Kit

CoolRunner-II Design Kit

The easy to use CoolRunner-II Design Kit includes all the tools needed to begin designing with the industry's most advanced high- performance and low-power CPLD, CoolRunner-II RealDigital CPLDs.

Simplify CoolRunner-II designs by allowing the addition of components via a breadboard area to create and test your own custom applications. The design kit provides a flexible, quick-start environment that enables complete design and device evaluation. *Jump start your CPLD design with the CoolRunner-II Design Kit today!*

- Design Kit Contents
- Design Resources
- Support
- Ordering a Kit

Design Kit Contents

XC2C256 Starter Board with AA Battery Holder (Part # DO-CR2-DK)

- Populated with XC2C256-7TQ144 CoolRunner-II CPLD
- Additional pad for 44-pin VQFP CoolRunner-II or XC9500XL/XV CPLD

Electronic Valentine Heart

LEDs are red my dear…

Design by F. Wolling

Although this month's Mini Project is not exactly stunning from a technical point of view, there are a number of unusual sides to it, mainly regarding why and how it came to be published in this issue. Read on…

Unusual aspect no. 1: This is the first Valentine circuit ever to be published in Elektor, and no, the highstreet flower shops are not to blame if that is the first you associate with Valentine's Day.

Valentine's Day is reportedly the day we 'celebrate' the beheading of a priest called Valentinus on 14 February 270. The execution orders came from an oppressive emperor called Claudius Goticus (a.k.a. Claudius II). According to some sources, Valentinus was a bishop and the 14th of February a popular Roman holiday. The legend of Saint Valentine tells us the martyr was executed for illicit marrying of Christian couples, the occasions being marked by Valentinus presenting lots of flowers to the newly weds.

The oldest literary evidence for Valentine's Day being associated with (secret) love may be found in 14th century writings, as well as in some of Shakespeare's sonnets and plays.

Unusual aspect no. 2: The designer of this circuit, Florian Wolling, was 14 years old when he sent us his article proposal — making him the youngest external author ever to make it to publication of a Mini Project.

Unusual aspect no. 3: Originally, this project did not stand a chance of getting published in *Elektor Electronics*. Just like all contributions and other article proposals we receive from free-lance authors, Florian's papers were 'evaluated for publication value' during a meeting of editors and design staff. Although the project met with praise for the meticulous way the circuit, the circuit board and the documentation had been designed and presented, and despite the fact that the small heart-shaped LED ornament was 'charming', the manuscript was rejected for publication because the complete

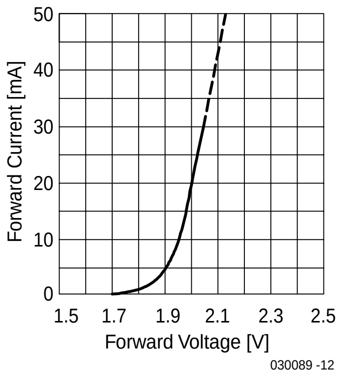


Figure 1. Typical U/I characteristic of a red LED.

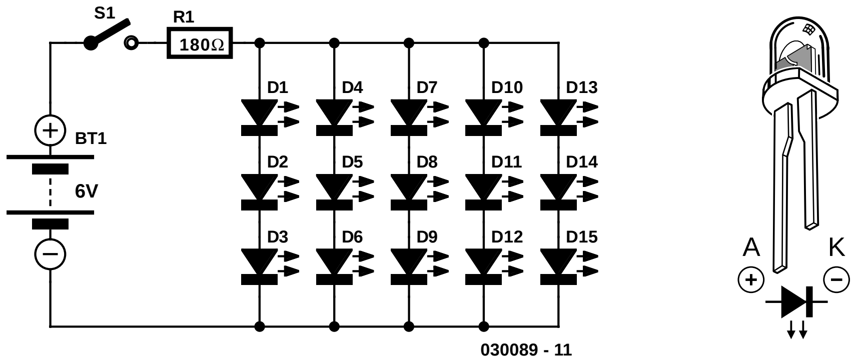


Figure 2. The circuit: one resistor and 15 LEDs.

project was found rather too simple for our demanding readership. However, when writing a letter informing Florian about the rejection of his project, still encouraging him to send further projects in the future, one of our fellow editors suddenly recognised the simple arrangement of LEDs on a board as the perfect circuit for Valentine's Day. Having discovered that such a circuit had not been 'done' before in our magazine, and because Elektor editors are always open to novelties, the LED Heart was added to our forward planning schedule. So, here it is, although one more thing should be noted…

Unusual aspect no. 4: The Elektor designer assigned with the technical preparation of Florian's project (mainly to make it meet our design standards), could not resist doing his bit of post-engineering. Admittedly, there's not you can change drastically when looking at one resistor and a bunch of paralleled LEDs, but improvements and fine tuning are always possible. The difference between Florian's LED Heart and the Elektorised version boiled down to the components used. Apparently, Florian was encouraged to use SMD LEDs by our two-part article 'SMDs — Don't Panic!' In the January and February 2003 issues of *Elektor Electronics*. The resulting project, he thought, would make a excellent gift as well present a convincing exercise to win over those readers still hesitant about their own skills in handling these tiny components. The latter aspect has been crossed out, however, because the Elektorised version of the project employs normal size components, not only with an aim to make the project easier to reproduce for beginners, but also to prevent component sourcing problems. Because all components except the LEDs are soldered at the rear side of the board, only the LED faces are visible from the front, allowing the front side of the board to be painted or sprayed in whatever way you like. For our own prototype, we happened to have some red paint left…

Circuit and component calculations

As most of you will know, there are few, if any, applications of LEDs that do not include some form of current limiting resistor. After all, LEDs are simply diodes that light when a forward bias voltage is applied across them. They also 'exhibit' the typical diode characteristic shown in **Figure 1**. If you apply a slowly rising voltage starting at 0 V to a LED, such that the device conducts (+ to anode; – to cathode), you will notice that nothing happens for quite a while — no current appears to flow. However, once the voltage exceeds a critical level, current starts to flow, rising very quickly when the voltage is increased further. This so-called threshold voltage occurs at 0.6 V with normal silicon diodes and at 1.6 V with most LEDs (as in Figure 1). A value of 1.7 V will be noted when testing low-current LDs in this way. Without a current limiting resistor, a small increase of the voltage beyond the threshold voltage would mean imminent destruction of the LED because the forward current rises extremely fast and soon exceeds the 'safe' level.

With a resistor in series with the LED, things look different because according to Ohm's law, the resistor becomes the main current determining factor. In the circuit of the Electronic Valentine Heart (**Figure 2**) five strings of series-connected LEDs are connected in parallel. Each string consists of three LEDs. The voltage across each of these strings equals three times the forward conducting voltage of a single LED. Assuming a device voltage of 1.7 V we get

'High Efficiency' LEDs

Light Emitting Diodes (LEDs) have been around for a long time — the first LEDs were mass-produced around 1975. The luminosity of red LEDs is specified by the manufacturer at a current of 10 mA or 20 mA (depending on type). These current eaters are hardly used anymore (at least not in Elektor circuits), because red LEDs with a much higher efficiency are now available. These successor types light happily at a current of just 2 mA (similar green LEDs are specified at 4 mA or 5 mA). Because of their improved efficiency, these LEDs are referred to as 'High Efficiency' types in some professional publications. The classification 'Low-Current' is however more commonly found in catalogues, datasheets and articles. Such LEDs are perfectly suitable for low currents of the order of 1-3 mA and are clearly visible in indoor applications when operated at just under 1 mA (as in the project described in this article). Besides, who would use the Electronic Valentine Heart in the glare of the February sun?

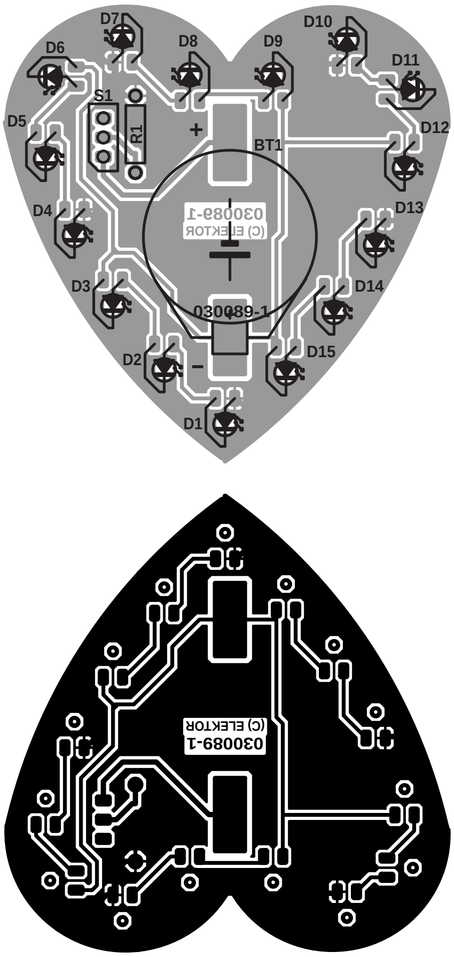
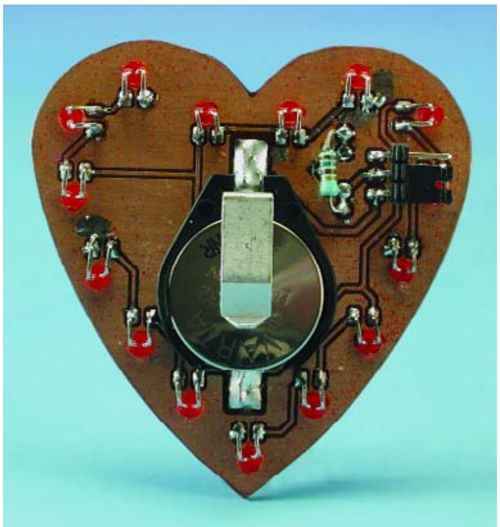


Figure 3. The heart-shaped board (insert LEDs from track side).



COMPONENTS LIST

R1 = 180Ω
D1-D15 = LED, red, low-current
S1 = 3-way SIL pinheader, angled, with jumper (alternative: miniature slide switch)
Battery holder = SMD button cell holder (e.g., Farnell # 302-9785)
Two button cells type CR2032
PCB, available from **The PCBShop**

$$3 \times 1.7\ \mathrm{V} = 5.1\ \mathrm{V}$$

The LEDs are connected to the battery by way of a 180-Ω resistor. Although the battery voltage is 6 V nominally (2 × 3 V), we should use the value 5.9 V to account for the load on button cells. If the voltage across the LED strings is 5.1 V, the resistor is accountable for the voltage difference between the two voltages at its terminals. The difference works out at 5.9 V – 5.1 V = 0.8 V. From this value we can calculate the current as

$$0.8\ \mathrm{V} / 180\ \Omega = 0.00444\ \mathrm{A} = 4.44\ \mathrm{mA}$$

This current is supplied by the battery, flowing through the resistor and the LEDs. Because all LEDs are of the same type and the voltage across each string is also equal, it is safe to assume that the current is equally divided across the five LED strings. Hence, the current through each individual LED equals

$$4.44\ \mathrm{mA} / 5 = 0.89\ \mathrm{mA}$$

Not a lot, you may think, but sufficient for a low-current LED to light with reasonable brightness. If the circuit was modified such that all LEDs are connected in parallel, the current consumption from the battery would be three times as high (assuming the same LED current is maintained). The load on the battery would then become about 13 mA, which is rather heavy on the button cells used. Besides, the difference in brightness between individual LEDs would be far more noticeable than with the series-connected version.

If you would like to know more about LEDs and how they can be used in large clusters, you should get a copy of our article ‘LED Arrays’ from the April 2003 issue of *Elektor Electronics*. The Internet is also a good source for additional information on LED calculations, see ‘Web Pointers’ at the end of this article.

Printed circuit board and construction.

The printed circuit board designed for the project is single-sided. The 3-mm holes in it enables the LED faces to be seen from the front side. For this to work the LEDs have to be inserted from the solder side of the board, their terminals being cut, bent and then soldered to the copper pads on the board. The other three parts in the circuit (battery holder, resistor and a 3-way SIL pinheader) are soldered in SMD fashion to the track side of the board. The battery holder is actually the only SMD component. Instead of the soldered battery holder you may, of course, use a separate holder (for example, containing two AA cells) and connect it to the board via wires.

When soldering the LEDs onto he board, make sure you get their polarity right. As indicated in the circuit diagram, the cathode is the shorter of the two wires. This (negative) connection is marked by the bar in the diode symbol.

Our prototype is switched on by relocating a jumper on the 3-way SIL pinheader. For a more sophisticated approach to on/off switching, a miniature slide switch is recommended.

Finally

If you have been wondering all the time why the Electronic Valentine Heart does not blink just like most other LED ornaments, brooches and trinkets, well, it happens to be *Unusual Aspect no. 5*. There’s nothing to stop you making the LEDs blink, though, just consider borrowing the flip-flop element from our ‘LED Christmas Decoration’ (December 2003).

(030089-1)

Web Pointers

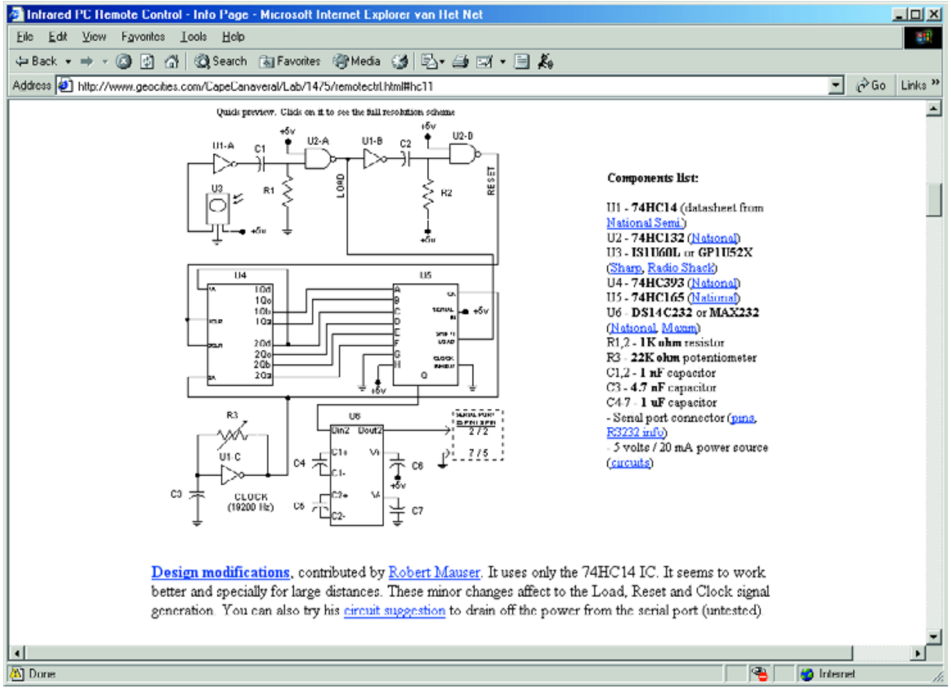
ourworld.compuserve.com/homepages/Bill_Bowden/led.htm
http://linear1.org/ckts/led.php
www.electronics2000.co.uk/calc/calcled.htm
www.hamradioindia.com/HRI-Calc/LEDcalc.htm

Remote Control for your PC

With a home-made infra-red receiver

By Harry Baggen

With the PC seeing increasing use as a multimedia machine, recording and playing back video and audio, users will soon want to control a number of functions remotely, just as with their TV or hi-fi set. All you need to realise this is a standard infra-red remote control unit, a home-brew receiver circuit and some software.


Infrared PC Remote Control - Info Page - Microsoft Internet Explorer van Het Net
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.geocities.com/CapeCanaveral/Lab/1475/remotectrl.html#hc11
Go Links »
Quick preview. Click on it to see the full resolution scheme
U1-A
C1
+5v
U2-A
U1-B
C2
U2-D
U3
R1
LOAD
R2
RESET
U4
U5
R3
U1 C
CLOCK (19200 Hz)
C3
C4
C5
C6
C7
U6
Din2 Dout2
Done
Internet

Components list:

U1 - **74HC14** (datasheet from National Semi.)
U2 - **74HC132** (National)
U3 - **IS1U60L** or **GP1U52X** (Sharp, Radio Shack)
U4 - **74HC393** (National)
U5 - **74HC165** (National)
U6 - **DS14C232** or **MAX232** (National, Maxim)
R1,2 - **1K ohm** resistor
R3 - **22K ohm** potentiometer
C1,2 - **1 nF** capacitor
C3 - **4.7 nF** capacitor
C4-7 - **1 uF** capacitor
- Serial port connector (pins, R3232 info)
- 5 volts / 20 mA power source (circuits)

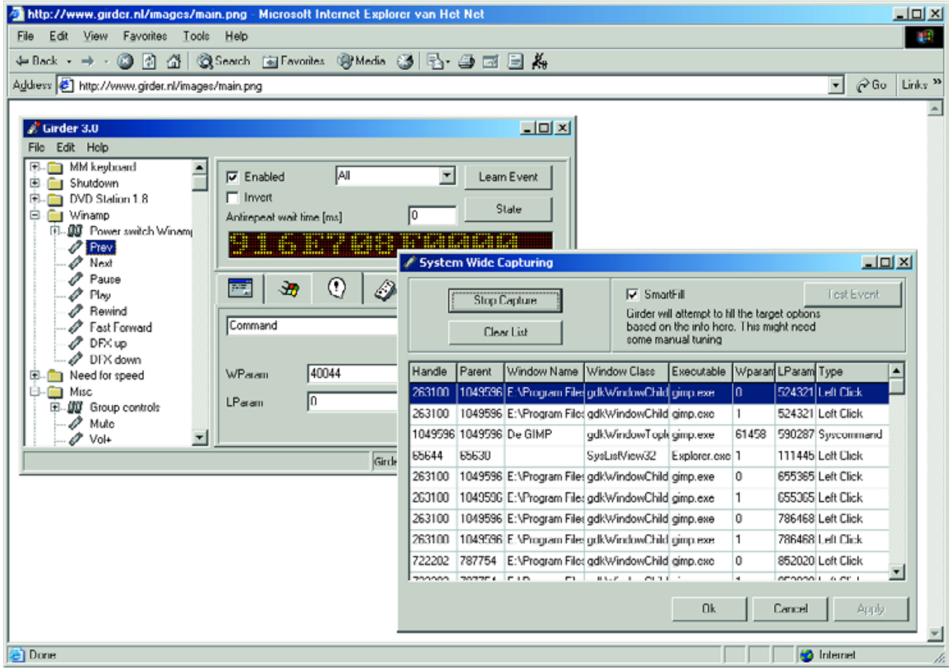
Design modifications, contributed by Robert Mauser. It uses only the 74HC14 IC. It seems to work better and specially for large distances. These minor changes affect to the Load, Reset and Clock signal generation. You can also try his circuit suggestion to drain off the power from the serial port (untested).

A number of manufacturers in the consumer electronics sector supply remote controls with special receivers — the complete kit allowing a PC to be controlled from a distance. In many cases, the remote control includes a kind of mouse replacement so you can even move the cursor around on the screen. The infra-red (IR) receiver usually consists of a small receiver module for connecting to the PC's serial port by way of a cable. The supplied software then looks after the processing of received IR signals. In many cases, a remote control set is included with a TV card for the PC, although the features then remain limited to a number of essential TV and radio functions.

Fortunately, several alternatives are now available to link a PC to a remote control. Thanks to a number of clever programmers and some really simple hardware it has become possible to use a dead standard remote control transmitter to realise remote control on your PC.

The receiver is, of course, home built, which — we are sure — is unlikely to present problems to readers of *Elektor Electronics*. Incidentally, if you happen to have an IR remote control lying around, say, from a TV card system, you'll find that this may often be used without problems in combination with the programs discussed below.

On the Internet you'll find a variety of articles and web pages discussing home construction of an IR receiver. The designs vary in complexity from five-component 'interfaces' right up to microcontroller-driven gadgets. In all cases, however, you'll have to build up a small circuit. When a microcontroller is used, we



http://www.girder.nl/images/main.png - Microsoft Internet Explorer van Het Net

Address: http://www.girder.nl/images/main.png

Girder 3.0

☑ Enabled — All

□ Invert

Antirepeat wait time [ms]: 0

Command

WParam: 40044

LParam: 0

System Wide Capturing

☑ SmartFill

Girder will attempt to fill the target options based on the info here. This might need some manual tuning

Handle	Parent	Window Name	Window Class	Executable	Wparam	LParam	Type
263100	1049596	E:\Program Files	gdkWindowChild	gimp.exe	0	524321	Left Click
263100	1049596	E:\Program Files	gdkWindowChild	gimp.exe	1	524321	Left Click
1049596	1049596	De GIMP	gdkWindowTopl	gimp.exe	61458	590287	Syscommand
65644	65630		SysListView32	Explorer.exe	1	111445	Left Click
263100	1049596	E:\Program Files	gdkWindowChild	gimp.exe	0	655365	Left Click
263100	1049596	E:\Program Files	gdkWindowChild	gimp.exe	1	655365	Left Click
263100	1049596	E:\Program Files	gdkWindowChild	gimp.exe	0	786468	Left Click
263100	1049596	E:\Program Files	gdkWindowChild	gimp.exe	1	786468	Left Click
722202	787754	E:\Program Files	gdkWindowChild	gimp.exe	0	852020	Left Click

found that in the majority of cases the hex code to program your own MCU is freely available.

How to build an IR Receiver by Alessio Sangalli [1] provides extensive explanations and clear photographs, allowing even beginners to assemble their own IR receiver without too much of an effort. Alessio's design is simple by any standard, consisting of an IR receive module, a voltage regulator and three passive components.

Similar designs, also rich in illustrations and with detailed descriptions may be found on the **Heatseekerz** website [2]. The author, Tom de Keninck, sets a standard how an electronic construction can be described on web pages. The photographs Tom shows us, in combination with a step-by-step approach and the assumption that the reader knows virtually nothing all contribute to the excellent reproducibility of the circuit.

The Infrared PC Remote Control — Info page [3] aims a bit wider. Here, we found several receivers discussed — designs with and without microcontrollers. The site also discusses the necessary software and proposes some ready to go solutions. We also found a download link to the shareware program 'PC Remote Control'.

A receiver circuit based on a PIC12C508/509 is discussed at the **UIR page** [4]. The author shows how the circuit may be built from standard components as well as SMDs (surface mount devices).

The **Universal Remote Control Receiver** [5] is a project built around a PIC16F84. According to the description, it is suitable for nearly all commercially available remote controls. Both the microcontroller hex code and the PC layout are available to everyone wishing to build this design. The designer also offers the associated software, DRCS (Dalton Remote Control Software), to run on your PC.

A similar circuit based on the previous design is called the **Universal IR-receiver for serial interface** [6] and may be found on the web pages of the electronics faculty at the University of Ulm, Germany.

Besides hardware (a remote control transmitter and a matching receiver) you'll also need software to detect and process the serial data, as well as convert it to commands the PC can understand. A universal program that looks after these tasks is called **LIRC** (Linux Infrared Remote Control) [7]. A Windows version of the program is available from the **WinLIRC** homepage [8].

Girder [9] is also a well-known *Windows automation tool* (shareware) capable of recognising and employing lots of remote control signalling formats for linking to Windows programs and PC hardware. The sheer popularity of this program has resulted in many exotic IR formats now being supported thanks to the contributions of active users. If you don't believe us, just take a look at the list of available plug-ins!

IRAssistant [10], which comes free of charge for private use, also allows mouse movement to be mimicked using the remote control, as well as launch and control programs, run macros, all from the comfy chair.

Finally, we should mention **uICE** [11] by MediaTexX. This shareware is also rich in features when it comes to driving hardware and software. The program is marked by an Office-like user interface, offering an on–screen menu and on-screen display looking very much like the ones you see on modern TV sets.

(045012)

Internet Addresses

[1] How to build an IR receiver: http://lnx.manoweb.com/lirc/

[2] Heatseekerz - remote control:
www.heatseekerz.net/index.php?page=articles&id=4&pagenum=1

[3] Infrared PC Remote Control - Info Page:
www.geocities.com/CapeCanaveral/Lab/1475/remotectrl.html

[4] UIR - Universal Infrared Receiver: http://fly.cc.fer.hr/~mozgic/UIR/

[5] Dalton Electronics - URCR: www.dalton-electronics.com/URCR/URCR.aspx

[6] Universal IR-receiver, Microelectronics Department, University of Ulm:
http://mikro.e-technik.uni-ulm.de/research/urcr.html

[7] LIRC - Linux Infrared Remote Control: www.lirc.org/

[8] WinLIRC Homepage: http://winlirc.sourceforge.net/

[9] Girder: www.girder.nl/

[10] IRAssistant: www.irassistant.com/dotcom/v_current/

[11] MediaTexX Software Solutions: www.mediatexx.com/