

ELEKTOR ELECTRONICS

PC TOPICS:

master/slave mains switch

keyboard switcher

CD-R formats



RF signal generator



**versatile control
for MiniDisc player**

barometer/altimeter with integral data logger

smartcard-
operated
safety lock

RFID
systems



CONTENTS

INFORMATIVE ARTICLES

40 RFID Systems

Communications engineers have developed a new application for smart cards: Radio Frequency Identification (RFID) readers.
By our Editorial Staff

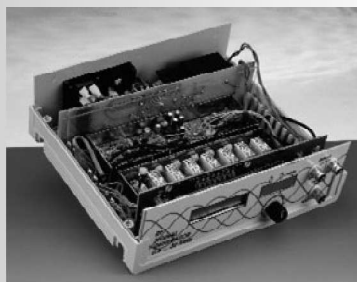
56 Programmable sensor interface MLX90308

An interface from Melexis which, with suitable software, can be readily adapted for use with a wide range of sensors.
A Melexis Application

CONSTRUCTION PROJECTS

10 RF signal generator – Part 1

This RF signal generator covers a frequency range of 0.5–30 MHz and has an output level range of 0 dBm to –79 dBm.
Design by Guido Brunner



30 PC control for Mini-Disc player

A design for controlling a Sony Mini-Disc deck via a PC.
Design by P. Glatzel



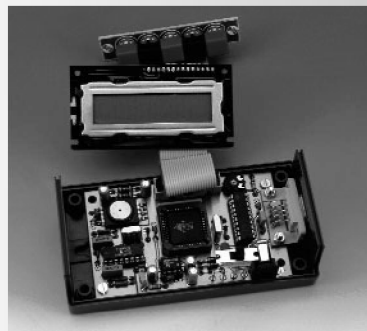
18 Versatile control system PLC87(A) – Part 2

The second and last instalment of the article describes the way commands are written and explains the operation of some of the more important instructions.
Design by R. Geugelin

24 Smart card operated safety lock

A design for giving expired smart cards a new lease of life.
Design by Patrick Gueulle

48 Barometer/altimeter



A design based on a Type MPXS4100A precision air-pressure sensor with integral signal processor from Motorola.
Design by H. Bonekamp

MISCELLANEOUS

- 65 Data sheets
- 54 Electronics on line:
Information on CD-R(W)
- 74 Index of advertisers
- 36 New books
- 60 New Products
- 74 Next month in Elektor Electronics
- 62 Readers' services
- 55 Safety guidelines

THIS MONTH IN PC-TOPICS:

- Atmel AVR-2
- PC keyboard switcher
- CD-R formats
- Compact display drive
- Master/slave switch

RF signal generator

part 1: circuit descriptions

An RF signal generator is used for repairing radio/TV circuits, checking filters, aligning receivers, and for comparative sensitivity tests on all kinds of receivers, whether home-made, restored surplus or off-the-shelf. The generator described here has an output frequency range of 0.5 to 30 MHz, making it suitable for many applications.



A rock-solid RF signal with an accurately known frequency and level is a must for anyone seriously involved in repairing radio receivers and other communications equipment like filters and even antennas. In particular,

receiver RF input and IF (intermediate frequency) sections can not be tested with any degree of certainty if a trustworthy RF signal generator is not to hand. Unfortunately, professional-grade RF signal generators (like the mighty Hewlett Packard 8640B in our design lab) cost an arm and a leg, even in the surplus trade. None the less, you will see at least one RF signal generator, home-made, thrown together from other bits and pieces, or ex-MOD, in the shack of the more advanced radio amateur, simply because this piece of test gear is as indispensable as the plain old multimeter.

The stability of the RF signal generator described in this article is such that it will meet the (moderate) demands of many amateurs. Offering a frequency range of 0.5 through 30 MHz and an output level down to -80 dBm, it is perfect for testing and aligning many receivers and their sub-circuits like RF/IF amplifiers, mixers

Main specifications

- Frequency range: 0.5 MHz to 30 MHz
- Output level: 0 dBm down to -79 dBm in 1-dB steps
- Max. output level: $0.63V_{pp}$ into 50Ω
- Output impedance: 50Ω
- AM input
- FM input
- LCD readout
- Microprocessor controlled
- Optional serial interface

Design by Guido Brunner

and demodulators.

What requirements can be mentioned in relation to an RF signal generator? The answer is very simple indeed: you need to be sure of (1) the frequency and (2) the level of the signal you feed into the circuit (receiver) under test. If either of these is unreliable, all testing and comparing of receiver specs becomes meaningless. In the present design, frequency stability is assured by a PLL (phase-locked loop), while the output level is determined by a switched pi (pi) attenuator, all under the control of a microprocessor.

BLOCK DIAGRAM

Because the actual circuit diagrams of the four modules that make up the signal generator are a fairly complex lot when presented together, it was decided to draw and discuss them as separate blocks. The basic interaction of these blocks is illustrated in Figure 1.



The block diagram shows that the heart of the circuit is a PLL synthesizer module keeping a VCO (voltage-controlled oscillator) in check. The VCO output signal is amplified and fed to the generator output as well to the synthesizer input and the input of the attenuator. The PLL obtains digital information on the target VCO frequency from a microprocessor module. The micro also takes care of the front-panel mounted user interface, which consists of 3 switches, a rotary encoder and an LCD (liquid-crystal display). It also controls the amount of attenuation at the generator output, across a range of -1 dB through -79 dB. An optional serial interface is available to enable the RF Signal Generator to be linked to a PC using an RS232 cable. Functionally, the instrument is completed by an internal power supply.

PLL BOARD

The circuit diagram of this first module to be discussed in detail is shown in Figure 2. It comprises three sub-circuits: VCO, synthesizer and output buffer. The VCO and the synthesizer together from the PLL.

VCO and buffers

The active element in the oscillator is a differential amplifier built around transistors T1, T2 and T3, whose gain

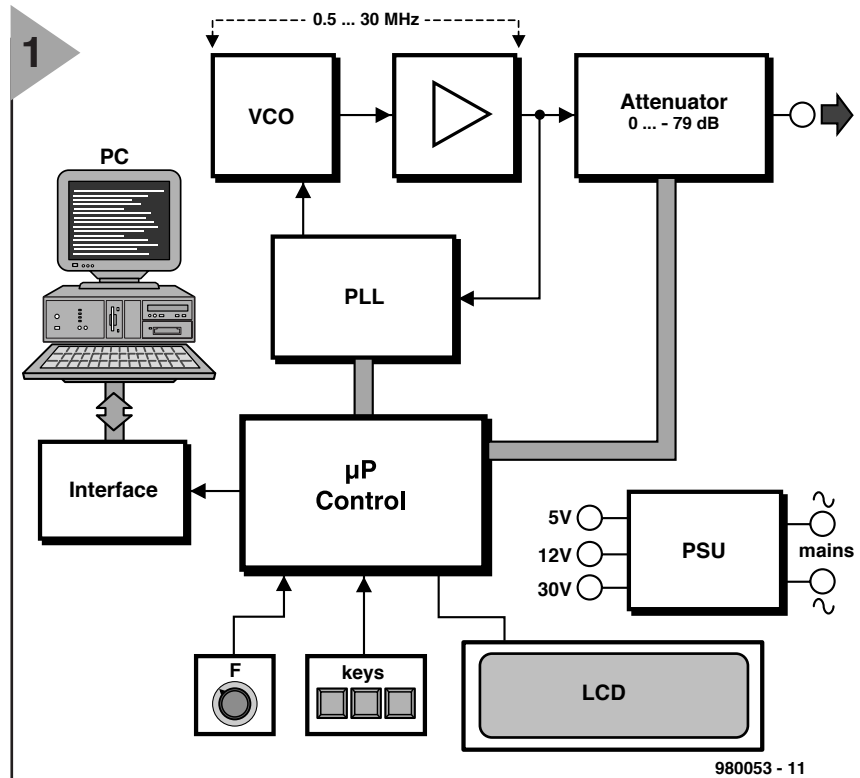


Figure 1. Block diagram of the RF Signal Generator. All intelligence is vested in a microcontroller.

depends on the current passed by T3. The actual resonating element in the oscillator is an L-C parallel tuned circuit connected to the input of the difference amplifier. The LC network consists of inductors L1-L5 in combination with variable-capacitance diodes (varicaps) D9 and D10. The other input of the oscillator is grounded for RF by capacitor C10. Depending on the desired frequency range, one or more inductors are switched into the oscillator. This is done by pulling the non-commoned terminals to RF ground using +5V control voltages on PIN diodes D2, D4, D6 and D8. In the highest frequency range, all inductors are effectively connected in parallel. This is necessary to make sure that the non-selected inductors and their parasitic capacitance can not form a series tuned circuit that would prevent the oscillator from operating at the desired frequency. All inductors are off-the-shelf miniature chokes. The frequency range switching takes place at 1.024 MHz, 2.304 MHz, 5.376 MHz and 13.056 MHz.

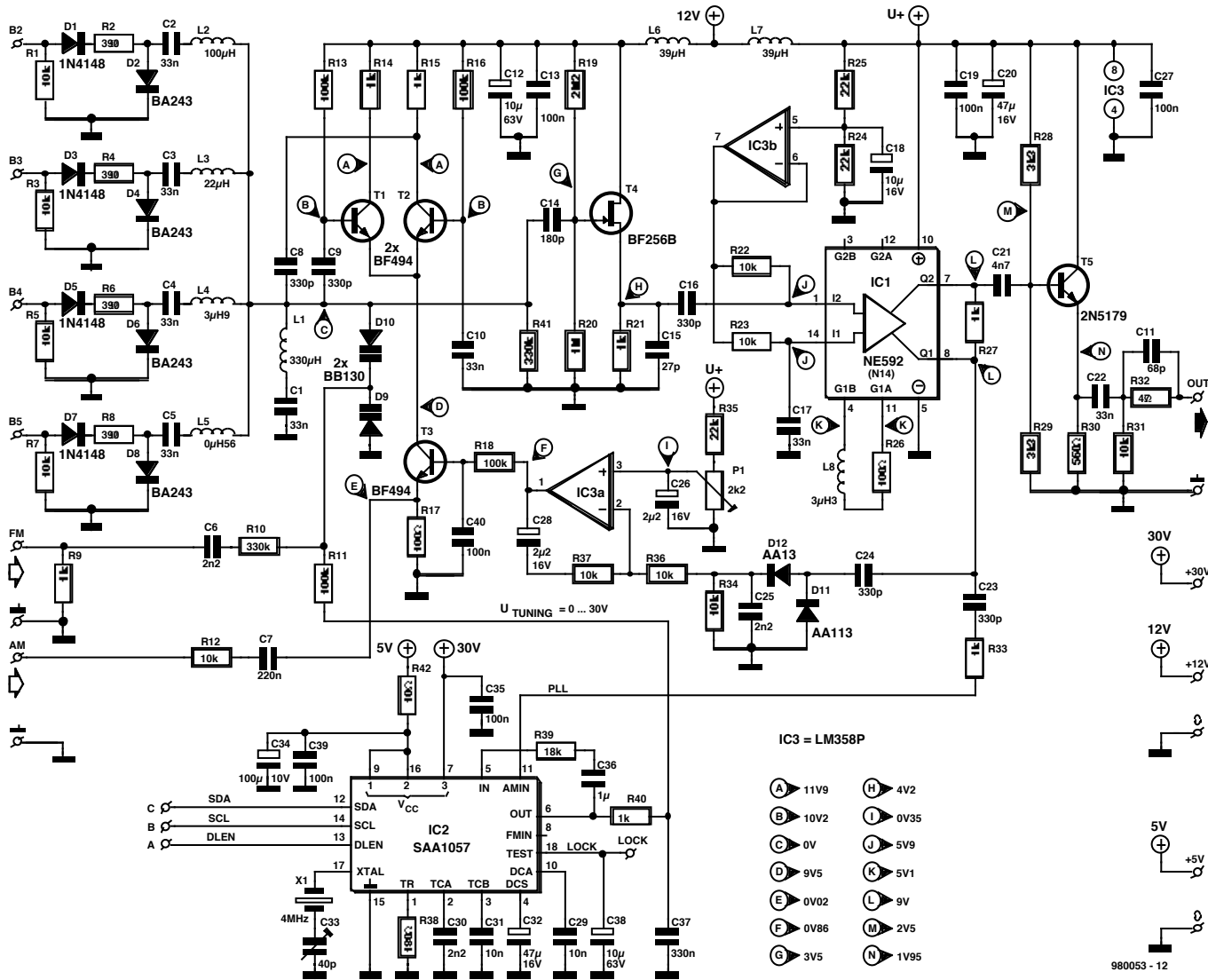
Capacitor C8 provides the necessary amount of positive feedback in the oscillator. An AF signal may be applied to the emitter of T4 to effect amplitude modulation (AM). Frequency modulation (FM) is also possible by superimposing an AF signal onto the varicap tuning voltage. Although FM will cause the PLL to drop out of lock, the average frequency remains constant because the time constant of the con-

trol loop is not capable of tracking the 'instability' caused by the modulation signal.

To make sure it is not too heavily loaded, the oscillator signal is first buffered by a FET (field-effect transistor), T4. Next comes the real amplifier, IC1, a type NE592 which some of you may know from baseband-video amplifiers in satellite-TV receivers. The amplifier is biased at half the supply voltage by opamp IC3b, and its gain is defined by series network R26-L8. Because of the inductor action, the gain decreases at higher frequencies. Because the VCO strives to maintain a stable output level, less gain on the NE592 automatically more gain in the differential oscillator. This purposely-created effect is essential for reliable starting of the oscillator at higher frequencies.

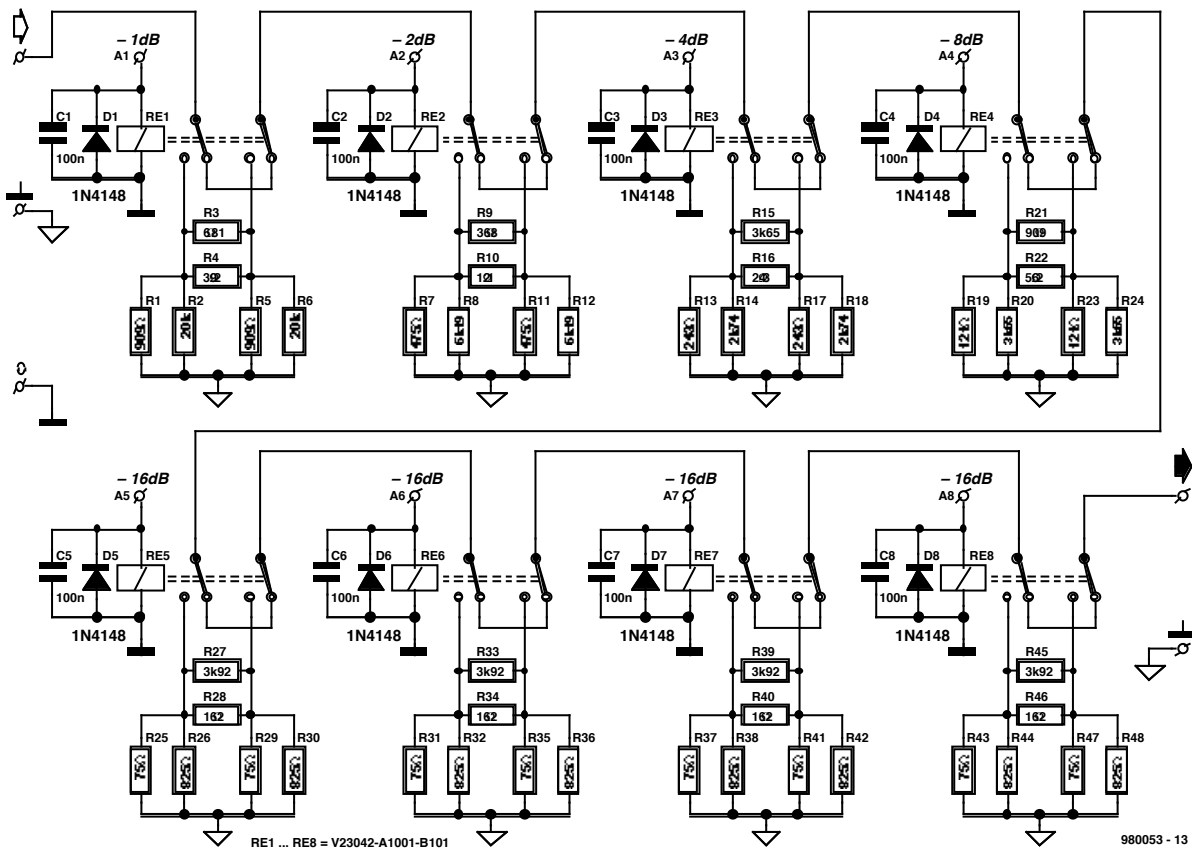
The NE592 being a differential amplifier, it has two inputs, but also outputs. Both are used here. The signal at the first output (pin 7) is applied to emitter follower T5 which supplies the actual generator output signal at an impedance of 50 Ω (the standard in RF test equipment). The other output signal supplied by the NE592 is used to drive two sub-circuits. One branch goes to the PLL chip via C23 and R33, the other is used to drive a voltage rectifier/doubler, D11-D12 which in turn drives amplitude-control opamp IC3a. The desired highest output amplitude may be set using preset P1. The author used a setting where 0 dBm (decibel-

2



980053 - 12

3



980053 - 13

Figure 2. Circuit diagram of the VCO/PLL board. The heart of the PLL is an I²C-controlled synthesizer chip type SAA1057.

milliwatt) into 50 Ω equals 0.63 V_{pp} at the generator output.

Synthesizer

The circuit of the synthesizer largely follows the Application Note for the SAA1057 as supplied by Philips Semiconductors. Some component values

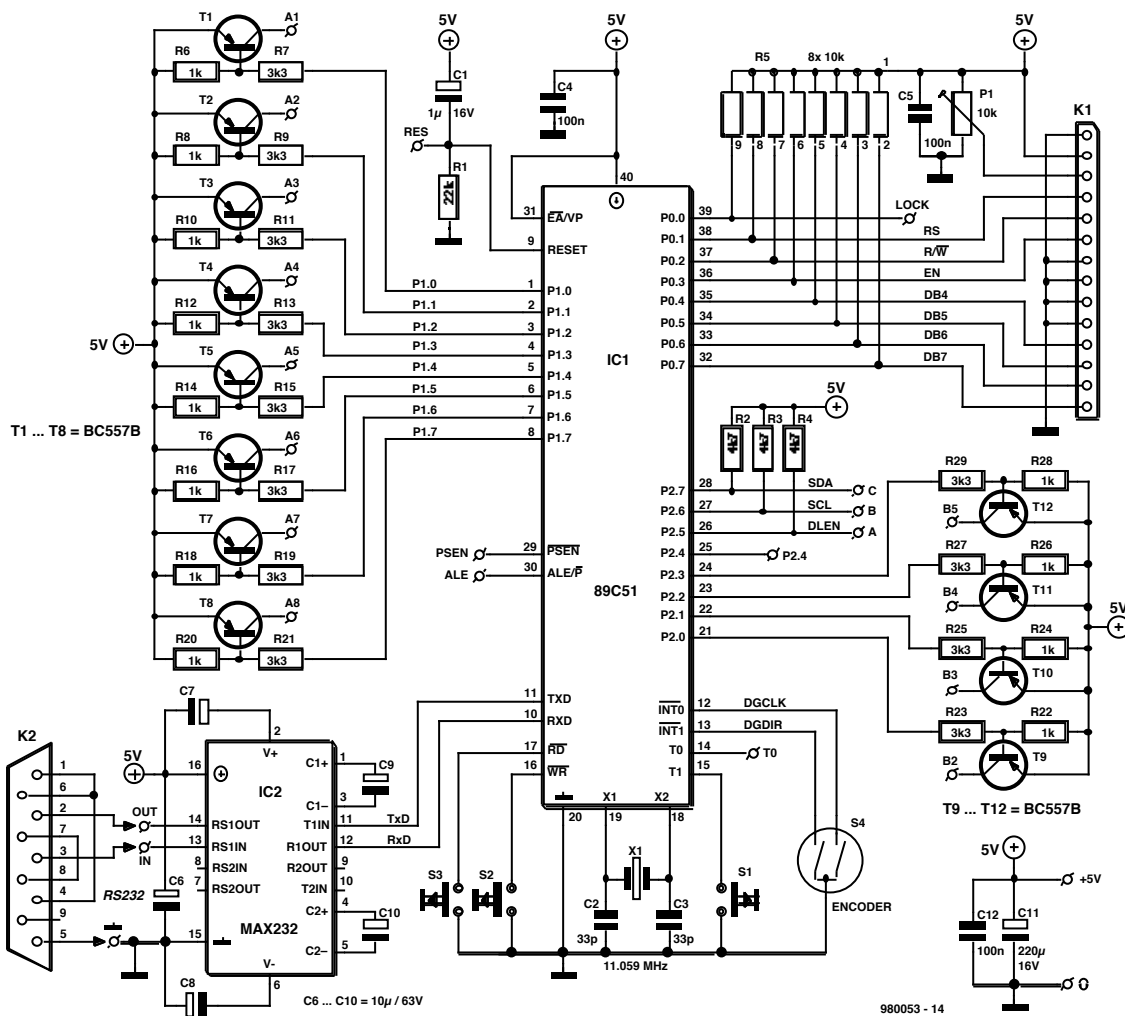
in the control loop had to be modified a little to optimise the behaviour of the PLL. The 'LOCK' output is only provided for test purposes. The SAA1057 receives its control information in I²C format via its SDA, SCL and DLEN inputs. These lines are connected to a microcontroller. Basically, the SAA1057 compares the frequency of the VCO with that of a reference signal derived from the external 4-MHz quartz crystal. For this purpose the VCO signal is internally divided by a factor determined by the microprocessor. The frequency difference produces an error signal which is converted into a corre-

sponding varicap control voltage. This control voltage is integrated by R40-C37 and has a range of 0-30 V. Remarkably, the SAA1057 does not require an external level converter for the varicap control voltage — a special amplifier is included on the chip for this purpose, as well as a direct connection for + 30 V (pin7).

Trimmer C33 allows the generator output frequency to be calibrated against a frequency standard.

The VCO/PLL board requires three supply voltages: + 5 V for the synthesizer, + 12 V for the VCO, and + 30 V for the varicap voltage.

4



980053 - 14

Figure 3. Circuit diagram of the digitally controlled attenuator. Range is -1 dB to -79 dB in 1-dB steps.

Figure 4. Circuit diagram of the controller board. An 89C51 sits between a number of input and output devices.

5

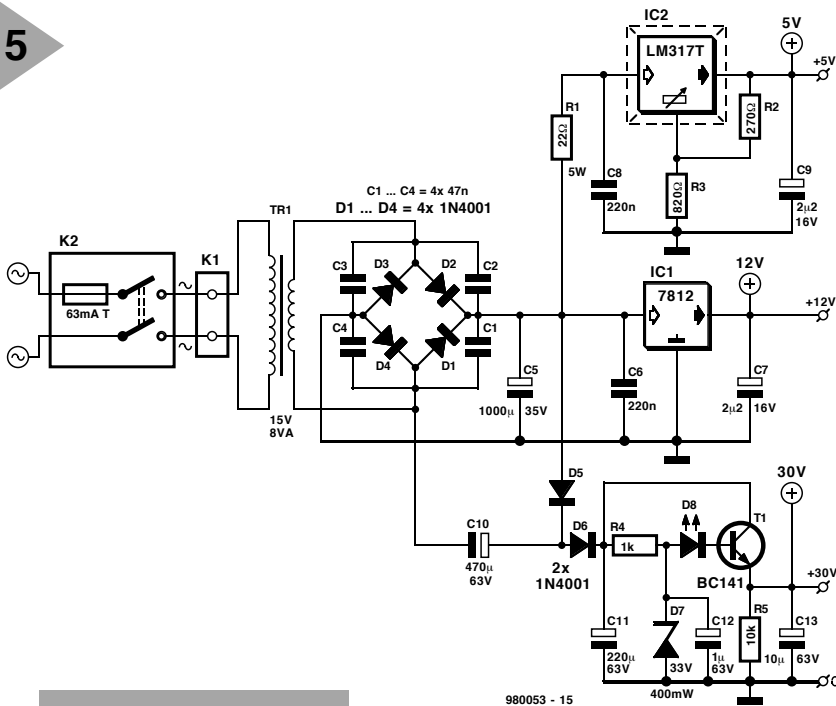


Figure 5. Circuit diagram of the power supply. Three voltages from one transformer!

ATTENUATOR BOARD

Figure 3 shows the circuit diagram of a digitally controlled 8-section pi RF attenuator with a range of -1 dB to -79 dB in 1-dB steps. The resistor combinations we need to realize each of the 79 discrete attenuation levels are connected into the circuit by means of relay contacts. The associated relays are actuated and de-actuated by microprocessor drive signals that form 8-bit combinations at the control inputs marked A1-A8.

The theoretical values of the resistors in the attenuator are realized by means of parallel combinations of 1% resistors from the E96 series.

Each relay coil is shunted by a back-emf suppressor diode and a decoupling capacitor.

MICROCONTROLLER BOARD

All the intelligence we need to implement a man/machine interface, i.e., establish communication between the user on the one hand, and the PLL and the attenuator on the other, is packed in a microcontroller type 89C51. This controller executes a program written

by the author and burned into the internal program memory by the Publishers. The 89C51 is available ready-programmed from the Publishers or certain kit suppliers advertising in this magazine.

The 89C51 accepts information and supplies information. Microcontroller freaks call this 'I/O' for input/output. Well, the input devices are a rotary shaft encoder, S4, which is used for the frequency setting, a small keyboard, S1-S2-S3, the SDA line of the I²C bus and (optionally) the Rx/D line of the MAX232 serial interface. The output devices to control are the LCD connected to port P0, the attenuator on port P1, the VCO inductors on port line P2.0 through P2.3 and, of course, the synthesizer chip, by way of the DDA and SCL lines (P2.6 and P2.7). Actually, the I²C bus is modified into a so-called CBUS by the addition of P2.5 (DLEN) and its pull-up resistor, R2.

The 89C51 is clocked at 11.0592 MHz by an external quartz crystal, X1. This frequency was chosen because it allows standard baud rates to be used on the serial interface.

A classic power-on reset network, R1-C1, completes the microcontroller circuit.

This board requires only + 5 V to

operate, the MAX232 having on-chip step-up converters for + 10 V and -10 V.

POWER SUPPLY BOARD

As you can see from the circuit diagram in Figure 5, the power supply for the RF signal generator is entirely conventional.

The 30-V varicap supply is based on a simple combination of a zener diode and a series transistor. Current drain on the 30-V rail will be very small, so extensive regulation is not necessary. None the less, a fair number of decoupling capacitors is used to keep the varicap voltage as clean as possible. After all, all hum, noise etc. on this rail will cause frequency modulation on the output signal. The input voltage for the 30-V regulator is supplied by a voltage doubler, C10-D5-D5.

The 5-V and 12-V supplies are based on two old faithfuls, the 7805 and the LM317 respectively. These ICs and their usual 'satellite' components have been used so many times in our published circuits that no further description will be necessary.

A single mains transformer rated at 15 V, 8VA, supplies all the necessary alternating voltages. The mains voltage at the primary side is applied via a double-pole switch and a fuse, both built into a Euro-style appliance socket.

NEXT MONTH

In next month's second and concluding instalment we will be discussing the construction of the instrument on four printed circuit boards. The article will be concluded with notes on the operation of the RF Signal Generator, miscellaneous matters and optional extras.

(980053-1)

versatile control system PLC87(A)

part 2: command list (CL) programming

The PLC87(A) is programmed using Simatic compatible command lists (CLs). A CL consists of a sequence of logic combinations and/or instructions for the PLC to execute. In this second and last instalment we look at the way command lists are written, and also explain the operation of some of the more important instructions.

The ports of the microcontroller on the PLC87(A) board are divided in inputs and outputs which form the links between the PLC and the 'real world'. The CL program, which is stored in an EEPROM, interrogates these inputs, and arranges for outputs to be set (actuated). Additionally, the PLC87(A) is capable of using flags (markers), software timers, flashers and counters. When the PLC is initialised or stopped, all outputs, flags, etc., are reset to zero.

A CL line consists of an operator and a parameter, or a command and a variable. Comment may be appended, or a jump label may appear ahead of operators or commands. After the program start, the CL interpreter, which is stored in the controller ROM, first reads the inputs, and stores a copy of the bit-image in a register (PAE). Next, the interpreter starts to execute the CL from the top line down to the bottom. While the processor executes the CL, it does not read the inputs, but their bit image. Likewise, the output states are first copied into a register (PAA), and not written into the respective ports until the PE command (program end) is encountered in the CL.

The CL interpreter reads a program line and executes it immediately. The results of the various logic combinations and the instructions are copied to internal registers (BCR, ACCU1, ACCU2, PAE, PAA), and may be called up for processing later. When the PE command is encountered, the interpreter returns to the start of the CL. This results in a delayed response to events (cycle time). Although the values of inputs and outputs are updated during every cycle, their status is not modified.

THE COMMANDS

The commands sent to the PLC87(A) by the PC may be divided into a number of groups. The Datasheets in this magazine list the English versions of the commands, together with their parameters, if applicable, and short descriptions. The Datasheets may be used as *Quick Reference Cards* for the PLC87(A) system.

Bit combinations

are the result of combining two bit-parameters.

A	z	x.y	AND
AN	z	x.y	AND NOT (NAND)
O	z	x.y	OR
ON	z	x.y	OR NOT (NOR)

The BCR (bit combination result) is the result of these logic operations. Valid parameters in position z are **I, Q, F, T, C** and **(**, while x represents the number of the byte, and y the number of the bit.

Allocations

These are used to allocate a certain bit parameter to the BCR.

S	z	x.y	set when BCR =1
R	z	x.y	reset when BCR = 1
=	z	x.y	equals (equals BCR)

Valid parameters are **Q, F** and **C**, while x represents the number of the byte, and y the number of the bit.

Timer functions

The timers **T n** (0-15) count down a previously loaded value. Various options are available. If the value becomes 0, interrogating the BCR will produce a '1'. Before the start command,

the timer value has to be loaded by means of L KT x.y (x=0-255, y=1, 2, 3)

SD T n start T n, on-delayed
 SF T n start T n, off-delayed
 SP T n start T n, as pulse
 SE T n start T n, as extended pulse

Counter functions

Each counter C n (0-7) has a count range of 0 through 255, and is capable of counting up or down. The counter value may be cleared, set and interrogated. A bit interrogation then produces a zero when the counter value is greater than 1.

CU C n C n counts up on positive edge of BCR
 CD C n C n counts down on positive edge of BCR
 R C n set counter to 0 when BCR=1
 S C n counter value to ACCU1 when BCR=1
 L C n load counter value into ACCU1
 A C n BCR=1 when counter value > 0

Byte combinations

These serve to perform bit-by-bit combinations of the contents of ACCU1 and ACCU2. The result of the byte-combination is left behind in ACCU1.

AB AND byte
 XB Exclusive-OR byte
 OB OR byte

Comparison commands

These allow you to compare the contents of ACCU1 and ACCU2.

!=B compare for: equal
 >B compare for: greater than
 <B compare for: smaller than
 >=B compare for: greater than or equal
 <=B compare for: smaller than or equal
 ><B compare for: not equal

Arithmetic operations

These are limited to 8-bit arithmetic. Invariably, the two accumulators are involved.

+B ACCU1 + ACCU2 = ACCU1 (carry to ACCU2)
 -B ACCU2 - ACCU1 = ACCU1 (borrow in ACCU2)
 *B ACCU2 * ACCU1 = ACCU1 (carry to ACCU2)
 /B ACCU2/ACCU1 = ACCU1 (rest in ACCU2)

As you can see, the result is always left in ACCU1, while the remainder, a carry bit or a borrow bit, ends up in ACCU2. To read ACCU2, you have to copy its contents to ACCU1 using the instruction TAK.

Load commands

The load commands offered by the CL interpreter enables variables or constants to be loaded into the ACCU(s). Before ACCU1 is actually loaded, its current value is moved to ACCU2.

L x y load
 L AI x load analogue input x (PLC87A only)
 L KB y load byte constant (y = 0-255)
 L KW y load word constant (y = 0-65535)
 L KT x.y load timebase (x= 1-255, y = 1-3)
 L C x load counter x with value
 L x load
 LI x load indirectly (x = IB, QB, FB; load

the contents of the input/output or flagbyte pointed to by ACCU1)

n = number of counter
 x = IB, QB, FB, FW
 y = number of byte

With word commands, both ACCUs are read (HighByte into ACCU2, LowByte into ACCU1).

Transfer commands

The transfer commands enable the ACCU contents to be moved to variables.

T x y (x= QB, FB, FW, y = no. of byte)
 T PR transfer to polling register
 TI x transfer indirectly (x = QB, FB)
 (transfers contents of ACCU2 to the output or flagbyte pointed to by ACCU1)

With word commands, both ACCUs are written to (HighByte into ACCU2, LowByte into ACCU1).

ACCU commands

The following commands only affect accumulator 1 (ACCU1):

SLB n rotate ACCU1 by n bytes to the left
 SRB n rotate ACCU1 by n bytes to the right
 INC n increment ACCU1 by n
 DEC n decrement ACCU1 by 1
 TAK swap ACCU1 and ACCU2 contents
 BCD convert ACCU1 contents into BCD number
 ADJ correct ACCU for temperature sensor AD22100K. The range -50 °C to +150 °C corresponds to an ACCU contents of 0-200.

Program-flow commands

The following commands control the program flow:

STP halt program at end of cycle when BCR=1
 STS halt program immediately when BCR=1
 NOP idle, causes empty line
 BE separate main program from subroutine
 PE must be included at end of program

Display commands

A number of commands are at your disposal to control a liquid-crystal display (LCD) with 2 lines of 16 characters. You can clear the display, position the cursor, output text strings as well as I, Q and F bytes. Furthermore, ACCU1 may be written into the display as a temperature value between -50 °C and +205 °C.

CLR clear display
 POS n position cursor at character n (start of second line at char. # 64!)
 PRT QB n write contents of output byte n into display
 PRT IB n write contents of input byte n into display
 PRT FB n write contents of flagbyte n into display
 PRT "STR" write character string (up to 20 characters)
 TMP write into display the contents of ACCU1 as temperature value from -50 °C to +205 °C

PRT commands are written at the current cursor position.

With cyclic refresh, every PRT command should be preceded by a POS command, if the same position is desired. If display commands are used without an LCD module connected, the PLC87(A) is likely to 'crash'.

Polling

The polling register, PR, writes values in the Polling Data Storage (PDS), which is basically a cyclic memory with a capacity of 512 bytes in EEPROM. Using the command TRG the value stored in the PR is written to the PDS at the end of the cycle, whereupon the next cell in the PDS is prepared for being written to. These are the relevant commands:

```
L IB 1 load input byte 1
T PR into polling register
A I 0.1 when I 0.1 = 1
TRG PR to PDS, one line on
```

For example, the PDS may capture certain events or states occurring as the program is being executed, and so enable the results to be read or processed at a later time. Once the PLC87(A) board is up and running, the PDS entry starts again at the first line. Only one value may be written into the PDS for every program cycle.

FOR EXAMPLE...

Under this sub-heading and helped by examples we will introduce CL programming. The PLC87(A) controls relays (via driver stages) which capture input states (via optocouplers) in a system using 24-V direct voltages.

Assignment: an immersion pump which is driven via output 0.4, should work when a switch at location 'A' or 'B' is switched on (High). It should stop pumping when the minimum water intake level ('dry') is reached.

```
:O I 0.1 switch A is on input 0.1
:O I 0.2 switch B is on input 0.2
:AN I 0.6 float switch at I 0.6 reports water
:= Q 0.4 output 0.4 switches on pump
:PE end of program
```

Assignment: a garage door is to be closed by a motor when a pushbutton is actuated, but only if there is no obstacle. When the end position is reached, or an obstruction is encountered, the motor should be switched off.

```
:A I 0.1 motor-on pushbutton connected to
input 0.1
:A I 0.2 light barrier at input 0.2 reports
all clear
:S Q 0.4 set output 0.4 (motor on)
:
:ON I 0.2 switch at I 0.2 reports occupied
(negative interrogation)
:ON I 0.4 switch end position reached at I 0.4
(opener)
:R Q 0.4 reset output (motor off)
:
:PE
```

Assignment: a horn should sound at a rate of 1 Hz when the motor powering a press is operating. Motor activity should also be indicated by a lamp. The motor runs when pushbuttons 1 and 2 are pressed simultaneously.

```
:O I 0.1 pushbutton 1
:O I 0.3 pushbutton 2
:= M 0.1 motor-on flag
:
```

```
:A F 0.1 motor-on flag
:= Q 0.4 motor output
:= Q 0.5 lamp output
:
A F 0.1 motor-on flag
A S 1.4 flash rate
:= Q 0.6 horn-on output
:
:PE
```

Assignment: when a switch labelled 'OPEN' is pressed, a valve should open after four seconds. After switching off, the valve should remain open for 20 seconds.

```
A I 0.1 switch, valve open
:L KT 40.1 load time constant 40x0.1s
:SD T 1 start timer 1, on- delayed
:
:A I 0.1 switch, valve open
:L KT 20.2 load time constant 20x0.1s
:SF T 2 start timer 2, off-delayed
:A T 2 when timer 2 reports 1
:AN I 0.1 and the valve was switched off
already
:O T 1 or the on-delay has elapsed
:= Q 0.4 then actuate valve
:PE
```

Assignment: Pulses applied to input 0.0 are to be counted and shown on a two-digit 7-segment display via output byte 0. Because the display is a common-anode type, the drive signals have to be inverted. When the counter value exceeds 99, or input 0.1 is actuated, the counter value is cleared.

```
:A I 0.0 count pulses
:CU C 1 on positive edge of BCR
:
:L KB 99 load constant 99
:L C 1 compare with counter 1
:<B when constant < counter
:O I 0.1 or input 0.1 is at 1
:R C 1 then reset count value
:
:L C 1 load count value into ACCU
:BCD change ACCU into BCD number
:CPL invert ACCU
:T QB 0 transfer to output byte 0
:
:PE
```

Assignment (for PLC87A with LCD only): a temperature sensor type AD22100K is connected to analogue input AI0. This sensor measures the temperature in an oven. The measured temperature is to be shown on a display, the target temperature is 120 °C. When the oven gets hotter than 150 °C, a warning should appear on the display.

```
:L AI 0 read analogue value
:ADJ ACCU do correction for AD22100
:T FB 1 store ACCU (temp.) in FB1 (0°C = ACCU
contents 50)
:
:POS 0 display position char. 0
:PRT "Temperature:"
:L FB 1 fetch value of FB1
:TMP and write to display as temperature
value
:
:L FB 1 when temp. < 110°C, switch on heater
```

```

:L KB 160
:<B
:S A 0.0
:
:L FB 1 when temp. > 130°C, switch off heater
:L KB 180
:>B
:R Q 0.0
:
POS 64 cursor in 2nd display line
:
:L FB 1 when temp. <150°C, jump to GOON
:L KB 200
:<B
:JC
GOON:
:PRT "OVEN ALARM!" show this alarm text when
temp. > 150°C

```

```

:JU
END
:GOON
:PRT "OVEN OKAY" else, display this text
:END
:PE

```

Further programming examples may be found in the subdirectory PLC87\EXAMPLE on the project diskette, order number 986026-1.

(989966-2)

Glossary

ACCU

The PLC87(A) board has two accumulators (ACCU1 and ACCU2) with a width of 8 bits. They are essential for logic byte combinations, for byte comparisons, for simple calculations and for shift operations. The ACCUs obtain their contents from load (L) commands. Before a byte is loaded into ACCU1, the current contents of ACCU1 are moved into ACCU2. An ACCU may be used to load variables or constants. The results of an operation are returned to the ACCU and may be copied to a variable using a transfer (T) command.

Analogue value processing

When an 87C550 processor is used, the PLC87A is also capable of processing analogue input values. The PLC system then offers six analogue inputs (AE0-AE5). The measurement range is from 0 to 5 V. At a resolution of 8 bits, that corresponds to a binary range of 0 to 255. Each of the PLC87A inputs may also function as a digital input. When the input voltage exceeds 2.7 V, the associated bit in input byte 0 is set to 1. When the value is smaller, the bit remains at 0.

BCR

The BCR (bit combination result) is the result of logic combining of two bit parameters, and is recreated as follows after each combination:

[BCR] = [BCR(old)] combined with bit parameter]

The current BCR is stored in a memory cell and may be copied to certain bit parameters with the aid of assignment commands.

```

: A I 1.3
: AN I 0.0
: = Q 0.0

```

The BCR may also be affected by timers, counters and comparisons. The result of logic combinations (BCR) may be 'noted' using a flag, which can be interrogated later. Similarly, results from byte assignments may be read from the ACCU. There exists a causal relation between bit parameters and byte variables, and the programmer may exploit this relation. The bit parameters are a subset of byte variables, which, in turn, are a subset of word variables. $FW0 = FB0 \& FB1 = F0.0$ to $F1.7$

Bit parameters

Valid bit parameters are

I	inputs	0.0 – 1.7
S	system flag	1.4
C	counters	0 – 7
Q	outputs	0.0 – 1.7
F	flags	0.0 – 15.7
T	timer	0 – 15

Brackets

Brackets allow bit combinations to be nested up to three levels deep. Make sure you have a closing bracket to match every opening bracket. As opposed to Simatic-S5 conventions, logic OR-ing of two blocks is not allowed without the use of brackets.

Byte variables

These are the parameters of byte operations, logic combinations, comparisons or storage of values.

Valid variables are

IB	input bytes	0 and 1
QB	output bytes	0 and 1
FB	flag bytes	0 – 15
FW	flag words	0 – 14
C	counters	0 – 7
PR	polling register	

Constants

Constants allow ACCUs or variables to be loaded with fixed values (con-

stants). Valid constants are:

KB	byte constant	(0 – 255)
KW	word constant	(0 – 65535)
KT	timer constant	(0 – 255, 1 – 3)
KC	counter constant	(0 – 255)

Counters

There are eight counters available (0-7), each capable of taking on values up to 255. No overflow is produced.

CU	C	x	count up on pos. edge of BCR
CD	C	x	count down on pos. edge of BCR
R	C	x	reset (sets count to 0 when BCR=1)
S	C	x	set (load ACCU1 into counter when BCR=1)
L	C	x	interrogate counter (fetch value, move to ACCU1)
A	C	x	check for > 0 (when count > 0 then BCR=1)

The highest count rate depends on the program cycle time.

Flasher

The system has available a flasher, S 1.4 (see bit parameters), which operates at a rate of 1 Hz. Its state may be interrogated by means of a logic combination.

Jumps

When the CL interpreter encounters a jump command, program execution continues at the CL line **before** the jump label, provided the jump condition is fulfilled. There are two types of jump:

JU	unconditional jump to label
JC	conditional jump if BCR=1

The label is entered in the right-hand column of the CL, and is defined by a '#' followed by four characters.

PLC87(A) vs. Simatic-S5

As already mentioned, the instruction set of the PLC87(A) is based on that of a Siemens Simatic-S5 system. It is not fully compatible with the S5! Because the PLC87(A) has accumulators with a width of just 8 bits, and no modules, a number of commands have been simplified to some extent. The essential differences with the S5 instruction set are:

1. comparisons use ...B instead of ...F;
2. shift operations use SLB, SRB instead of SLW, SRW;
3. different method of indirect addressing;
4. program end has to be marked with PE;
5. subroutines are placed at the end of the main program, they are called with UPA, UPC and closed off with RET;
6. BE is used to separate the main program from subroutines;
7. analogue inputs may be read directly
8. as opposed to a Simatic system, the PLC87(A) recognises commands for LCD control and for the polling register (PR).

The bulk of the commands is however identical, and that should enable you to teach yourself PLC programming using any of the vast number of courses and manuals covering the Simatic S5 standard.

When using jumps and subroutine calls, you should make sure that the jump label occurs only once in the program. The CL may contain up to 64 labels, no matter if simple jump labels or subroutine labels. Furthermore, subroutines may only begin after a BE command.

Programming errors

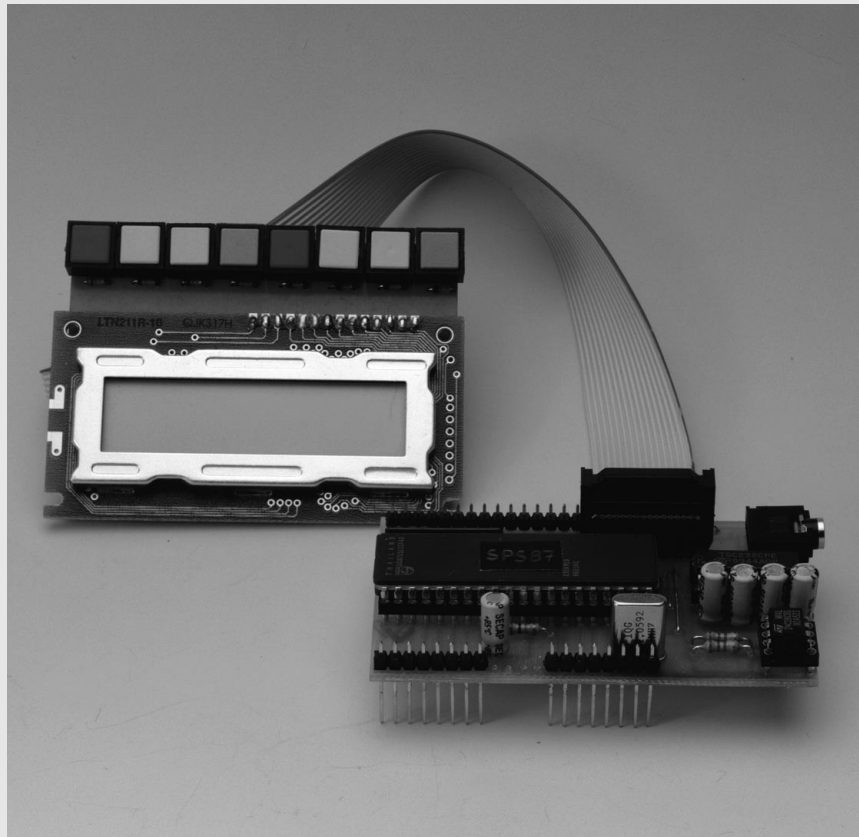
The following errors typically cause the PLC87(A) system to crash. Recovering from a system crash is only possible by removing the EEPROM and inserting it again.

JU, JC used to branch to subroutine label

1. UPA, UPC used to branch to jump label;
2. Multiple use of flags
3. Failure to use RET to close off subroutine;
4. Subroutine before a BE;
5. No BE before program end and the first subroutine;
6. Jumps from one subroutine into another;
7. Nesting of subroutines;
8. Too many opening brackets;
9. Failure to complement opening bracket with closing bracket.

Safety

According to various standards for industrial safety, errors in automated



equipment may not cause a risk to the intrinsic safety of persons, machines or goods. Equipment parts relevant to any aspect of intrinsic safety should be implemented in hardware (i.e., traditional circuit technology like relays and protective devices). An 'emergency-off' switch has to be implemented in hardware. With system reliability in mind, limit switches and other actuators should be connected to the PLC87(A) in the same way as to its hardware equivalent. For example, the 'off' switch of a motor has to be wired and programmed as a normally-closed contact:

```
AN I 0.3  switch: motor-off
R  Q  0.5  stop motor
```

Temperature measurement

The ADJ command is available for temperature measurement using the AD22100K sensor from Analog Devices (see Datasheets, October 1998 issue). A temperature range of -50 °C to +150 °C then corresponds to an output voltage of 0.25 V to 4.75 V. The contents of ACCU1 is automatically adjusted to give '0' at a temperature of -50 °C, '50' at 0 °C, and '200' at the maximum temperature, 150 °C.

The command TMP causes the measured temperature to be shown on the LC display at the cursor position, or the position set up by the POS command.

Subroutines

When the CL interpreter encounters a subroutine call, the program is continued at the CL line before the subroutine label, provided the call condition is fulfilled. There are two types of subroutine call:

UPA unconditional call
UPC conditional call (BCR=1)

The subroutine should be closed off with an RET command to enable the interpreter to continue executing the CL at the line following the subroutine call. A BE command must be inserted between the main program and the first subroutine.

Subroutines may be called as often as you like. They may not, however, be nested or call themselves.

Timers

The PLC87(A) command interpreter offers 16 timers (0-15), each having 4 user-selectable functions.

SD start with on-delay
SF start with off-delay
SP start as pulse
SE start as extended pulse

Before a timer may be started, a time constant has to be loaded into the ACCU. The value should be in the range 1-255 (in ACCU1), while the timebase goes to ACCU2 and takes a value of 1, 2 or 3 for 0.1 s, 1 s or 10 s respectively, in the format XXX.Y

Smartcard-operated code lock

give expired Smartcards a second life



Reading smartcards and writing to them, that's all very well, but building real-life application circuits, that's even better! Among the countless circuits that can be labelled 'electronic locks' an obvious, preferred, position is held by systems employing 'smartcards'.

This type of application is particularly well suited to programming into a microcontroller with on-board EEPROM (electrically erasable programmable read-only memory) such as the PIC16C84, even if you content yourself with the limited possibilities of a simple programming language like BASIC.

LEARN-AS-YOU PROGRAM

The essential point of a Smartcard-controlled lock is simply the sheer infinite number of possible codes.

A simple (expired) payphone or loyalty card, for example, is identified by a unique 'reference number' which is easy to read into a memory but almost impossible to modify.

Such a 96-bit identifier word allows nearly 80 trillion (million million million) different card numbers to be issued.

To exploit this almost inexhaustible stock of 'keys' represented by disposable chipcards, it is sufficient to design a reader unit capable of memorising the 'images' (electronic contents) of one or (better) several expired smartcards, and then preferably cards you

actually want to re-use.

Having passed this 'learning' phase, the card reader may well control, say, a relay if and only if you insert one of the smartcards the reader has been 'trained' to recognize as far as the electronic content is concerned.

Alternatively, you may want to consider 'personalizing' certain cards having one and the same 'ID', and originally intended for applications not related to telephones, for example, GSMs, car-wash cards (Mobil and BP), parking cards, loyalty cards, satellite-TV cards, etc. An expired car-wash card originally worth 24 units, for example, still contains 136 bits reading '0', which are easily and individually set to '1' with the aid of a chipcard reader/programmer like the one described in *Elektor Electronics* September 1997.

Although the most obvious choice is to use expired smartcards, an alternative is to order new (blank) cards in any quantity you want, from suppliers like Elea (France, type number ELEA100), or Crownhill Associates in the UK (see their advertisement elsewhere in this magazine).

As far as the author has been able to

1

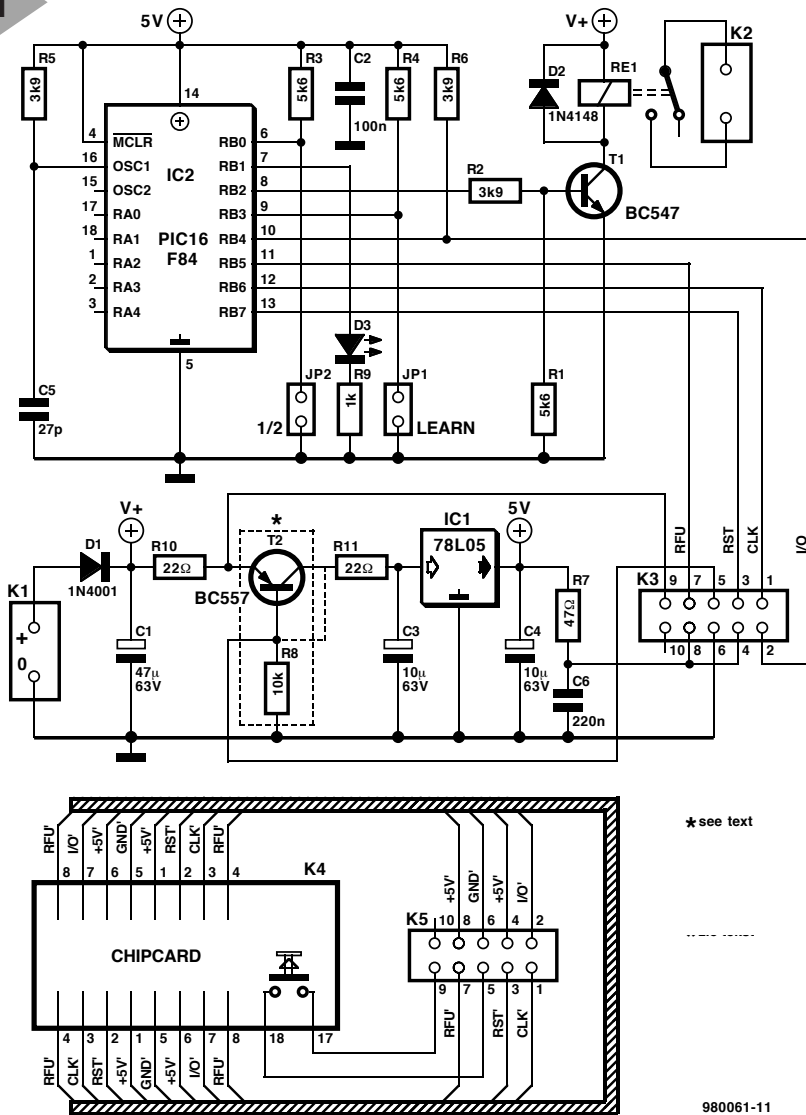


Figure 1. The circuit of the electronic code lock consists essentially of a PIC microcontroller, a Smartcard reader and a handful of other components.

find out, these 256-bit cards are not really blank, though, because they contain a 'lot' number in the memory are reserved for the first 96 bits. The next 160 bits are, however, at logic 0, and may be freely changed to logic 1s.

Depending on the application(s) you have in mind, the electronic lock will either recognize one or two cards that can not be forged, or, by contrast, any number of identical cards that may, under certain conditions, be duplicated.

A PIC16C84 PROGRAMMED IN BASIC!

The simplicity of the circuit diagram shown in Figure 1 may come as a surprise given the relative complexity of the functions it is designed to carry out.

Actually, that is because the microcontroller used, a PIC16C84, contains almost everything we need; to begin with, a data EEPROM with a size of

64 bytes.

This non-volatile memory is capable of recording two 'card images' of 256 bits each. These images may quietly remain in the PIC EEPROM for more than ten years, or be modified as many times as you like.

Because the circuit does not consume any significant current unless a card is inserted into the connector, it may operate from a simple 9-volt battery, although there is, of course, no objection against powering it from a mains adaptor or a rechargeable battery (for example, that of an alarm system).

Whatever power source is used, a type 78L05 voltage regulator is used to supply the PIC with its 5-volt supply

```
' DUAL smart card lock with PIC16F84
' copyright (c) 1998 Patrick GJEUILLÉ
symbol f = B0
symbol g = B1
symbol e = B2
symbol r = B3
symbol BLINK = B4
symbol h = B5
```

```
DI RS = %1100110
PI NS = %00000010
high 7
high 6
low 6
low 7
high 6
low 6
high 5
if PI N3 = 0 then learn
if PI N3 = 1 then check
learn:
for f = 0 to 31
e = 0
for g = 0 to 7
e = e * 2
e = e + PI N4
high 6
low 6
next g
h = 32 * PI N0
h = h + f
write h, e
next f
PI NS = 0
BLINK = 50
goto led
check:
for f = 0 to 31
e = 0
for g = 0 to 7
e = e * 2
e = e + PI N4
high 6
low 6
next g
read f, r
if e <> r then other
goto cont
other:
h = f + 32
read h, r
if e <> r then error
cont:
next f
PI NS = %0000110
goto forever
error:
PI NS = 0
BLINK = 255
goto led
led:
for f = 0 to 10
high 1
pause BLINK
low 1
pause BLINK
next f
forever:
goto forever
```

Figure 2. Listing of the program, before compilation, to be stored in the PIC.

voltage, while a transistor allows the relay coil to operate at a sufficiently high coil voltage.

It will be clear at this point that the wonderful simplicity of the circuit is due to an invisible program executed by the PIC microcontroller.

It may come as surprise to you that the control program was written in BASIC, or PBASIC to be precise, a 'dialect' originally developed by Parallax Inc. for its BASIC Stamp modules.

Although it would be easy to download the program shown in **Figure 2** into a BASIC Stamp 1 wired up to a Smartcard reader unit, we could not resist the temptation of an alternative solution which looks poised to become a buzz-word: *compiling*.

Several PBASIC compilers for the PIC microcontroller are available from reputed companies like Forest Electronic Developments and Micro Engineering Labs, but they may also be found in the public domain.

Whereas the most efficient of these compilers produce object code which is competitive to what can be produced with an assembler program, the simplest of them are no more than 'interpreters'.

That is not to say that the term

'interpreter' is inappropriate here because the BASIC program is actually stored in memory in the form of 'tokens' (a couple of bytes per instruction), which a resident piece of software starts to 'interpret' one by one. However, this process is usually called 'tokenising' instead of 'interpreting'.

The compiler we employed for the present project at the same time loads this 'firmware' and the 'tokenised' BASIC into an ordinary PIC16F84 of which about three quarters of the available program memory is occupied.

By contrast, the BASIC Stamp 1 comes with its interpreter mask-programmed into the (read-protected) EPROM on board an OTP (one-time programmable) PIC16C56, while the BASIC 'tokens' are stored in a separate EEPROM type 93LC56.

Whatever method is used, 256 bytes are available for the application program. Strange as it may sound, that is both a lot and very little to go on, mainly, because the tokenised BASIC is very compact.

For sure, the approach using the

BASIC Stamp is the most practical during the development phase. On the other hand, the use of empty PICs and a compiler is cheaper once the system software and hardware have passed the debugging phase.

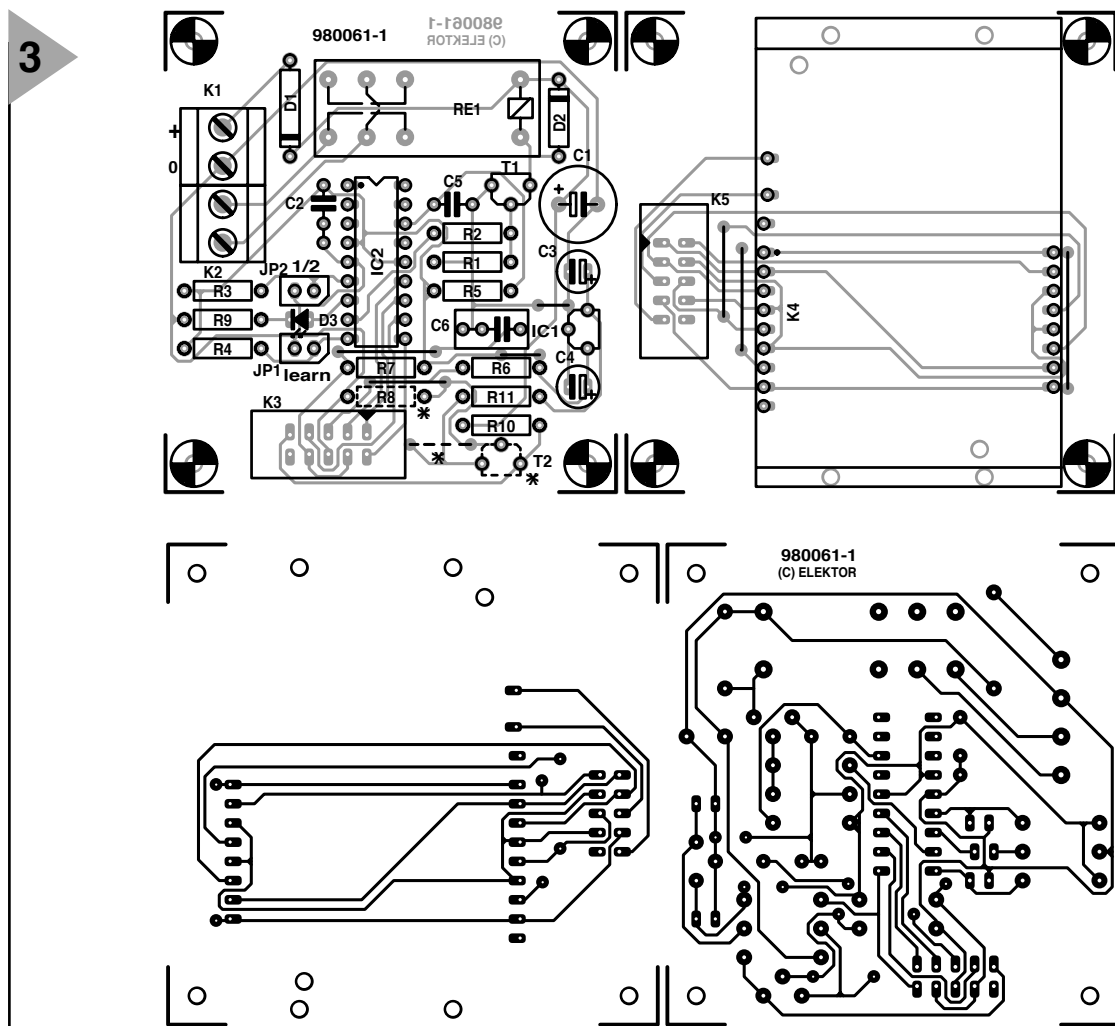
The fact that a compiler could be used which is compatible both with the BASIC Stamp and the PIC16F84 was a great help when it came to writing our program, particularly at the level of accessing the EEPROM, a function which is a bit difficult to implement in assembly language.

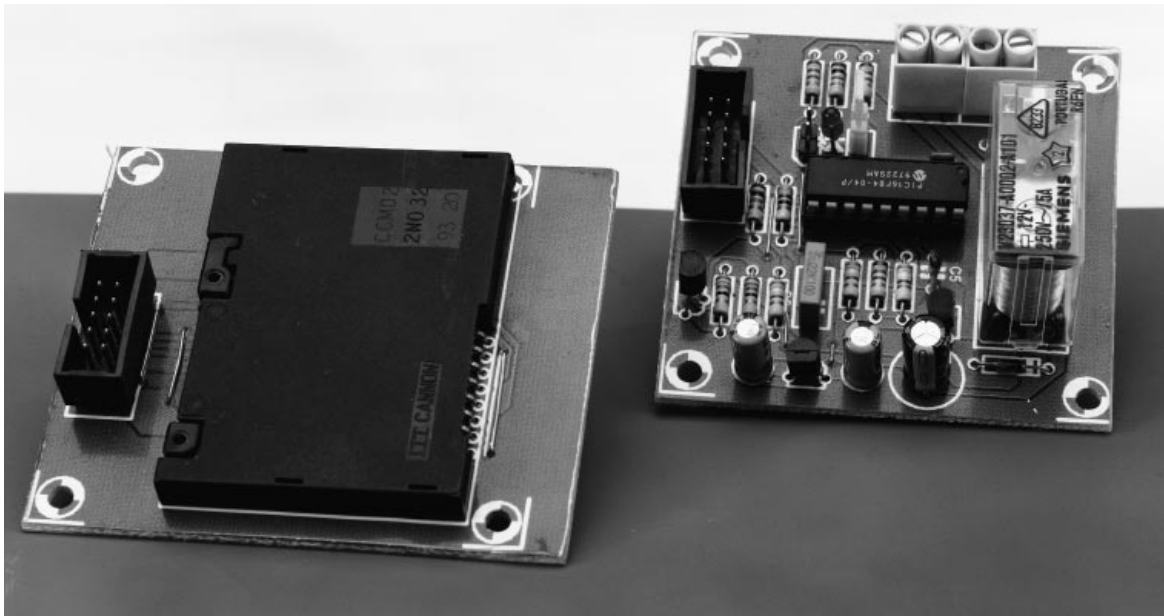
It is universally acknowledged that most 'interpreted' BASIC runs at snail's pace. Here, however, that turned out to be an advantage because delay loops are then not required!

The PIC is clocked at about 2 MHz by a simple RC network, and requires less than a second to read 256 bytes from any synchronous chipcard using the German or French protocols for data/command exchange.

A reset facility is provided to make the circuit compatible with payphone cards from almost any country.

Figure 3. PCB design for the Smartcard controlled electronics lock. The board consists of two sections that have to be separated with a saw.





PRACTICAL REALISATION

If you construct an electronic code lock, based on whatever circuit principles, you should observe a minimum number of elementary security measures.

It would have been (too) easy to throw all components that make up the circuit on the same board as the inevitable chipcard connector, and prescribe the use of SMA (surface-mount assembly) components to keep the size of the board within reason.

We know several examples of 'professional' equipment from reputable manufacturers, having the above-mentioned deplorable and over-simplistic construction.

In a number of cases, it was sufficient to remove just two screws to get access to two wires which, when short-circuited, open the locks...

The printed circuit board of which the design (artwork) is shown in **Figure 3** has been designed to facilitate fitting the electronics in a safe place, that is, inside a building or room to be protected (for example, on the inside of a door).

The only part which is accessible from the outside is the card connector,

whose ISO and AFNOR contacts may be wired in parallel if you want the system to be able to process both card types (the AFNOR pin arrangement is now obsolete even in France).

(Dotted) components R8/T2 are provided to allow card connectors with either a normally-open (NO) or normally-closed (NC) contact to be used.

In the first case, you do not fit R8/T2, and install a short wire linking the base and collector pads reserved for this transistor. This link is shown as a dotted line. If, on the other hand, you have an NC-style connector available, you have to install these two parts, and no wire link.

As illustrated in **Figure 3**, a 10-way flatcable is used to connect the card reader to the circuit. This cable is fitted with an IDC (insulation displacement connector) socket to be plugged on to the 10-way pinheader on the PCB.

Considering the (purposely-reduced) speed of the signals exchanged with the Smartcard (approx. 500 Hz), cable lengths of up to 50 cm may be used without problems.

A robust 4-way PCB mount terminal block allows the dc power supply (9 to 15 volts) to be connected to the board (K1), as well as the load to be

switched on and off by the relay contact (K2). This contact is only closed when the 'right' card is detected in the card reader.

The LED has to be mounted so that it is only visible when the system is being set up, because its very function is to inform you of the correct execution of each of the steps in the process.

Two jumpers, finally, allow you to select between the two available modes of operation: 'learn' or 'normal' use (1 / 2). It is not necessary or even desirable (for obvious reasons) to provide access to these jumpers for any purpose other than the process of 'teaching' the system to recognize new cards.

The programmed PIC for this project is available through our Readers Services under order code **986511-1**. It should not be installed in its socket before you have checked your construction and verified the presence of the correct voltages at several points in the circuit.

PRACTICAL USE

Easy to build, the electronic code lock is even easier to use.

As a matter of course, you should start by 'teaching' the PIC to recognize

COMPONENTS LIST

Resistors:

R1, R3, R4 = 5kΩ6
R2, R5, R6 = 3kΩ9
R7 = 47Ω
R8 = 10kΩ
R9 = 1kΩ
R10, R11 = 22Ω

Capacitors:

C1 = 47μF 63V radial
C2 = 100nF
C3, C4 = 10μF 63V radial
C5 = 27pF

C6 = 220nF

Semiconductors:

D1 = 1N4001
D2 = 1N4148
D3 = LED, red, 3mm, low-current
T1 = BC547
T2 = BC557
IC1 = 78L05
IC2 = PIC16F84-04P (pre-programmed, order code 9865111-1, see Readers Services page)

Miscellaneous:

K1, K2 = 2-way PCB terminal block,

pitch 5mm
Re1 = 12V PCB mount relay, e.g., Siemens V23037-A0002-A101 (ElectroValue)
JP1, JP2 = 2-pin header with jumper
K3, K5 = 1-way pinheader
K4 = smartcard reader, e.g. Canon CCM02 2NO 21 93 20.
10-way flatcable (max. 50cm)
2 10-way IDC sockets
Printed circuit board, order code 980061-1 (see Readers Services page).

a certain Smartcard. To do so, install JP1 to select 'LEARN' mode. That effectively pulls the RB3 port line of the PIC (pin 9) to ground. Next, insert the Smartcard or payphone card into the reader.

The LED should light up, indicating that the circuit is being powered. After about one second the LED should start to flash rapidly. This 'eye-winking' indicates that the card image (electronic contents) has been transferred to EEPROM, and that it remains there until another card is inserted, even if you turn off the power supply and leave it off for months.

Having supplied this signal, the LED indicator goes out, and you can safely withdraw the card, which from then on is promoted to 'master key'.

Now remove the 'LEARN' jumper and insert your Smartcard again. The LED indicator should light, and the relay should be energised the next second or so. It remains energised and the LED remains on, as long as the card is in the connector.

Remove the card and try another. Unless this one has exactly the same contents as the 'master key', the LED should flash slowly, and then go out. Of course, the relay should not be energised!

The above process may be repeated after changing the jumper in position '1/2', which enables the electronic lock to be 'taught' to recognize a second card.

In case you want the lock to recognize just one card, you have to present it to the system for both positions of the '1/2' jumper. This is done to avoid the system security being weakened by leaving one half of the EEPROM in an undefined state.

ENCODE YOUR OWN CARD SERIES

Although the ability of the system to recognize two different Smartcards will be sufficient for many applications, there will be cases in which many more users need to be authorized to open one and the same door or lock (as, for instance, in a laboratory with controlled access, a club or a block of apartments).

You then turn to cards whose contents was originally the same, within the same series, and you complete the encoding by a couple of freely chosen bits.

As an example, let's use car-wash

cards sold in France by BP or Mobil petrol stations. Once expired these cards are thrown away by the tens of thousands by their users.

A Smartcard worth 24 units (credits), all used up, contains the following 256 bits:

```
1000 1000 1000 0000 0010 0000 0000 0010
0011 1100 0111 0101 1000 0010 0010 0100
1010 0001 0000 0000 0000 0000 0000 0001
0000 0000 0011 1111 1111 1111 1111 1111
1100 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
```

In the first three lines we find the name of the card manufacturer, which can not be modified. The name string is identical on all 24-unit cards issued up to now, but different from the one on 12-unit cards, which, fortunately, are equally easy to obtain (in France).

Next come 10 bits all reading '0', followed by 24 '1's (these are the used-up credit units), and then 126 '0's which are not used by the application.

If you make the mistake of 'teaching' the electronic lock to recognize such a card, then the door can be opened by anyone having an expired car-wash card originally worth 24 credit units!

The obvious thing to do is of course to 'personalise' the card and so limit the number of authorized cards in accordance with the number of people (card users) who are authorized to open the lock. Cards may be 'personalised' quite easily using the software utility 'ELEKTIG' and the associated Smartcard Reader/Writer, both as described in the September 1997 issue of *Elektor Electronics*.

The following example goes to show how to change any number of the 136 '0's on the card into logic '1', simply by pressing the '+' key on the PC keyboard when the cursor is on top of the desired bit, after the card data have been read.

```
Space bar:  read next bit
              (auto-repeat)
+ key:      write 1
              (at current location)
ESCape key: quit
```

```
1000 1000 1000 0000 0010 0000 0000 0010
0011 1100 0111 0101 1000 0010 0010 0100
1010 0001 0000 0000 0000 0000 0000 0001
```

```
0011 0000 0011 1111 1111 1111 1111 1111
1100 0000 0000 0000 0000 0000 0000 0000
0000 0000 1010 0000 0000 0000 0000 0000
0000 0000 0101 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
```

WARNING: Switch the Vpp off!

For the other authorized cards you follow exactly the same procedure, and you use one of these cards to 'teach' the electronic lock the code by which the door is opened.

The same procedure may be applied with 'blank' (new) cards, provided they are of the GPM256 type or similar (256 bits of EEPROM) — ask your supplier.

Whether the card is new or expired, only the first 96 bits have already been personalised (by the manufacturer), the remaining 160 being at 0 and freely programmable by yourself.

Here is an example of what may be found on a card picked from an actual batch:

```
0000 0000 1000 0000 0000 0000 0000 0000
0100 1011 0000 0000 0000 0000 0000 0000
1111 1111 1111 0000 0000 1111 1111 1111
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
```

The buyer of a batch of these cards is totally free to employ ELEKTIG to write logic 1's into any location(s) in the last five lines, as illustrated below:

```
0000 0000 1000 0000 0000 0000 0000 0000
0100 1011 0000 0000 0000 0000 0000 0000
1111 1111 1111 0000 0000 1111 1111 1111
1111 1111 0000 0000 0000 0000 0000 0001
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 1100 0011 0000 0000 0000
1000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 1110 0010
```

WARNING: Switch the Vpp off!

Of course, you have to be aware of the fact that such a card may be copied, but only on to a card with the *same 96-bit manufacturer-ID*, which means that it has to be from the *same batch* bought from the *same manufacturer*. That seriously limits the possibilities of cards being forged!

(980061-1)

PC control for MiniDisc player

beam up track titles from your PC

Sure, it did take some time, but it is fair to say that Sony's MiniDisc has finally achieved market acceptance. The circuit presented in this article enables a MiniDisc deck to be controlled in a comfortable way when it comes to entering track titles. By using a PC and some clever electronics, you can put track titles onto your shiny little discs.



Specifications

- ⇒ For use with Sony MD decks types MDS-JE500 and later
- ⇒ Transmitter range: several metres
- ⇒ Up to 99 track titles
- ⇒ Up to 2032 characters per MD
- ⇒ Suitable for any PC with a free RS232 port
- ⇒ Standard 9-pin RS232 connection
- ⇒ No special software required
- ⇒ May be combined with a track-title database
- ⇒ Energy-saving design
- ⇒ Powered by one 9-V battery

MiniDisc (MD) audio systems are sold by the hundreds of thousands every year in various countries in which this *Elektor Electronics* is published. After a slow start, the MD system appears to

have made it to general recognition and commercial success after a number of years. Not surprisingly, several readers (mostly proud owners of MD decks) wrote to us with the request to

give some attention to the MiniDisc system. One frequently heard shortcoming of the MD system, we learned, is that entering a track title list is cumbersome. Although every MiniDisc has sufficient capacity for 256 tracks representing a total of 1,700 text characters, the process of actually entering the texts you want is very user unfriendly: lots of cursor and letter keys have to be pressed on the compact remote control before any meaningful text appears on the display. This text function is of special interest to music collectors. Tracks on old records (LPs), for example, can be efficiently filed together with the name of the album, the release number, name of the artist(s), etc. In most cases, the available space of 1,700 characters will be more than sufficient. A fine system, really, if only the process of entering texts was a little less awkward.

In this day and age, with the proliferation of PCs, it must be possible to simplify the process of conveying a track title to a MiniDisc deck, and, from there on, to the little disc itself. The designer of the present circuit fully agrees with this opinion, and set out to combine the functionality of a

PC with the power of a small microcontroller. This has become possible ever since Sony, from their model MDS-JE500, started to incorporate software in their MD decks that allows texts to be read by an infra-red receiver and then copied on to a MiniDisc. The PC control for MiniDisc players discussed in this article forms the link between a text file (document) written on your PC, and the infra-red 'eye' on the MD player. The system allows any PC to 'beam' up to 99 track titles to the MD deck, which writes them to a MiniDisc. If you are not sure if a particular MD deck will be compatible with the present circuit, ask the supplier if it is suitable for use with the advanced MD Title Generator supplied by Sony. If it is, then the present circuit can be used also.

THE CIRCUIT IN DETAIL

The circuit diagram of the MD control is shown in **Figure 1**. The input is formed by connector K1, a 9-way serial port. The output is formed by an infra-red sender diode, D3. IC4, a MAX232, converts the signals on the serial

port from TTL to RS232 levels, and the other way around. The electrolytic capacitors around IC4 help the internal circuits in the MAX232 to adapt the various voltage levels. In this way, the internal charge pump enables a symmetrical voltage to be derived from a direct voltage of +5 V.

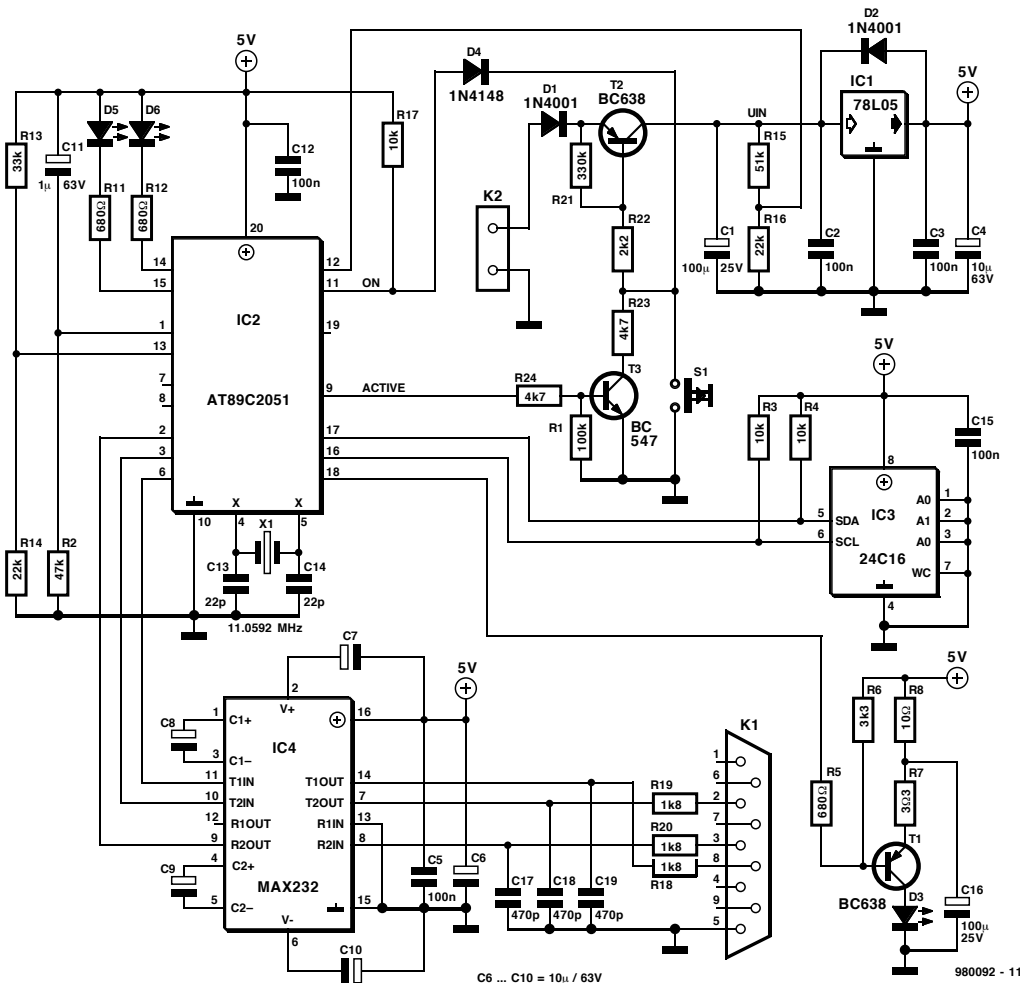
Although the circuit only uses the Tx, Rx and CTS signals on the RS232 port, these are sufficient for reliable data exchange between the MD control and the PC. The three logic signals appear at TTL level on three pins of the microcontroller (P3.0, P3.1 and P3.2) type AT89C2051 from Atmel.

The controller is an 8-bit machine whose instruction set is compatible with Intel's MCS-51 instruction set. It has 128 bytes of RAM and 2 Kbytes of Flash memory.

The circuit around the controller includes a standard crystal oscillator operating at 11.0592 MHz.

The line marked 'ACTIVE' (P3.5) plays an important role because it is used to implement software control of the power supply. You actuate the circuit by pressing switch S1. The base of transistor T2 is then pulled to

Figure 1. Circuit diagram of the MD title generator: simplicity is trumps!



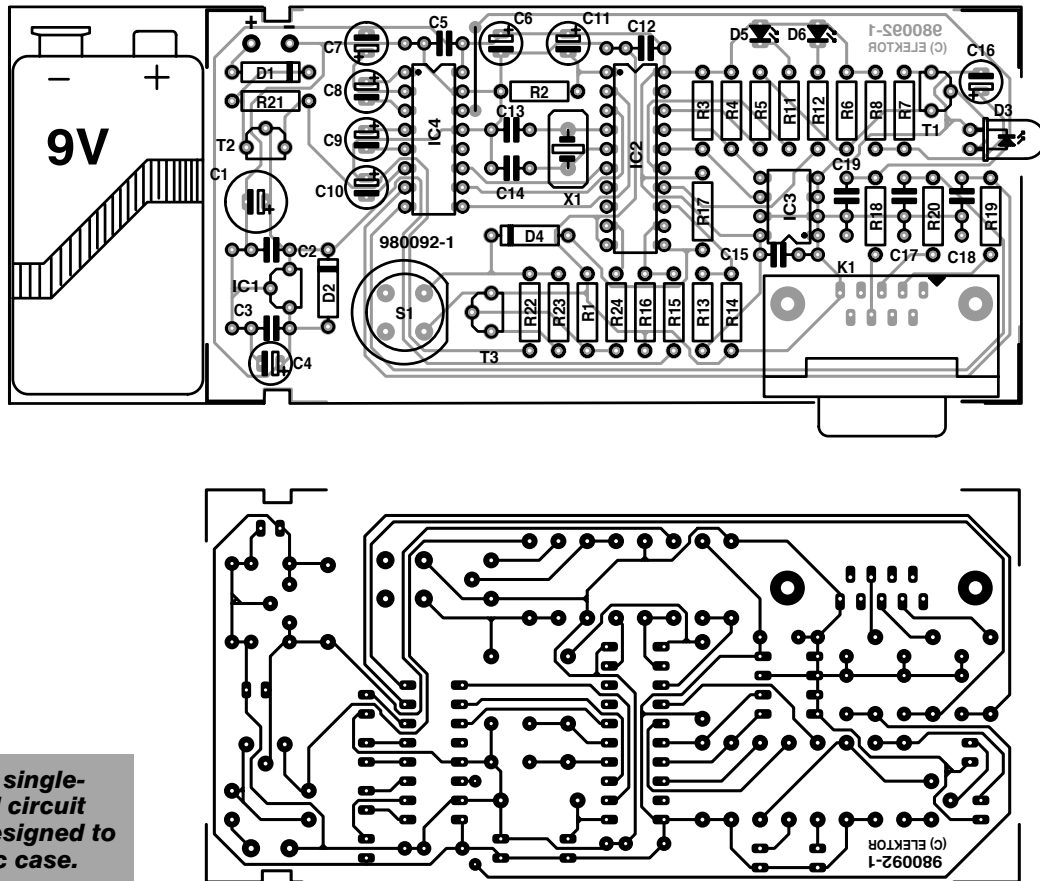


Figure 2. The single-sided printed circuit board was designed to fit in a Heddic case.

COMPONENTS LIST

Resistors:

R1 = 100k Ω
 R2 = 47k Ω
 R3, R4, R17 = 10k Ω
 R5, R11, R12 = 680 Ω
 R6 = 3k Ω
 R7 = 3 Ω
 R8 = 10 Ω
 R13 = 33k Ω
 R14, R16 = 22k Ω
 R15 = 51k Ω
 R18, R19, R20 = 1k Ω
 R21 = 330k Ω
 R22 = 2k Ω
 R23, R24 = 4k Ω

Capacitors:

C1, C16 = 100 μ F 25V radial
 C2, C3, C5, C12, C15 = 100nF ceramic
 C4, C6-C10 = 10 μ F 63V radial
 C11 = 1 μ F 63V radial
 C13, C14 = 22pF
 C17, C18, C19 = 470pF

Semiconductors:

D1, D2 = 1N4001
 D5 = red led 3mm
 D6 = yellow led 3mm
 D7 = IR-LED e.g. LD271 (LD274)
 D8 = 1N4148
 T1, T2 = BC638
 T3 = BC547
 IC1 = 78L05

IC2 = AT89C2051-PC, order code

986516-1
 IC3 = 24C16
 IC4 = MAX232

Miscellaneous:

K1 = 9-pin sub-D socket (female), PCB mount
 S1 = push-button D6-C-90 cap BTN-D6-90 (90=black; 40=red) ITC (Eurodis)
 X1 = 11.0592MHz
 Case: Heddic type 222
 PCB, order code 980092-1
 PCB and programmed microcontroller (set), order code 980092-C.

ground via resistor R22, whereupon the power supply is activated. The microcontroller then receives its supply voltage, and is initialised. Upon detecting that the ON line is active, the controller pulls the 'ACTIVE' line logic high, causing transistor T3 to be switched on. As a result, the power supply remains switched on even if you release S1. If the circuit has been inactive for some time, the microcontroller makes the 'ACTIVE' line low again and so causes the whole circuit to be switched off. In this way, the battery life is considerably extended. There is another benefit, namely that you can switch off the circuit by holding S1 depressed for about 2 seconds.

Incidentally, the circuit will only respond to this command if no communication activity is detected on the RS232 port.

Although it may appear rather complex at first blush, the power supply is of a straightforward design. Transistor T2 is the actual on/off switch which conveys the applied direct voltage to IC1, at the command of S1 or T3. The output of IC1 supplies a stable direct voltage of 5 V. Diode D1 protects the circuit against accidental supply reversals. Although a 9-V battery is the preferred power source, the circuit may also be powered by an inexpensive mains adaptor having an output voltage between 9 V and 12 V.

Via its input pins 12 and 13, the microcontroller continuously monitors the levels of the stabilised and non-stabilised voltages. Actually, the voltages at these pins are scaled-down versions of the full supply levels.

Two of the microcontroller's output lines are used to control LEDs D5 and D6. As soon as an output drops low, the associated LED lights up. Further, two port lines are in use to implement the communication with the EEPROM. Both lines (pins 16 and 17) simulate an I²C bus, and are connected to the +5-V rail via a pull-up resistor. The EEPROM, a type 24C16, is used to buffer the text which is eventually stored on the MiniDisc. After all, the whole point

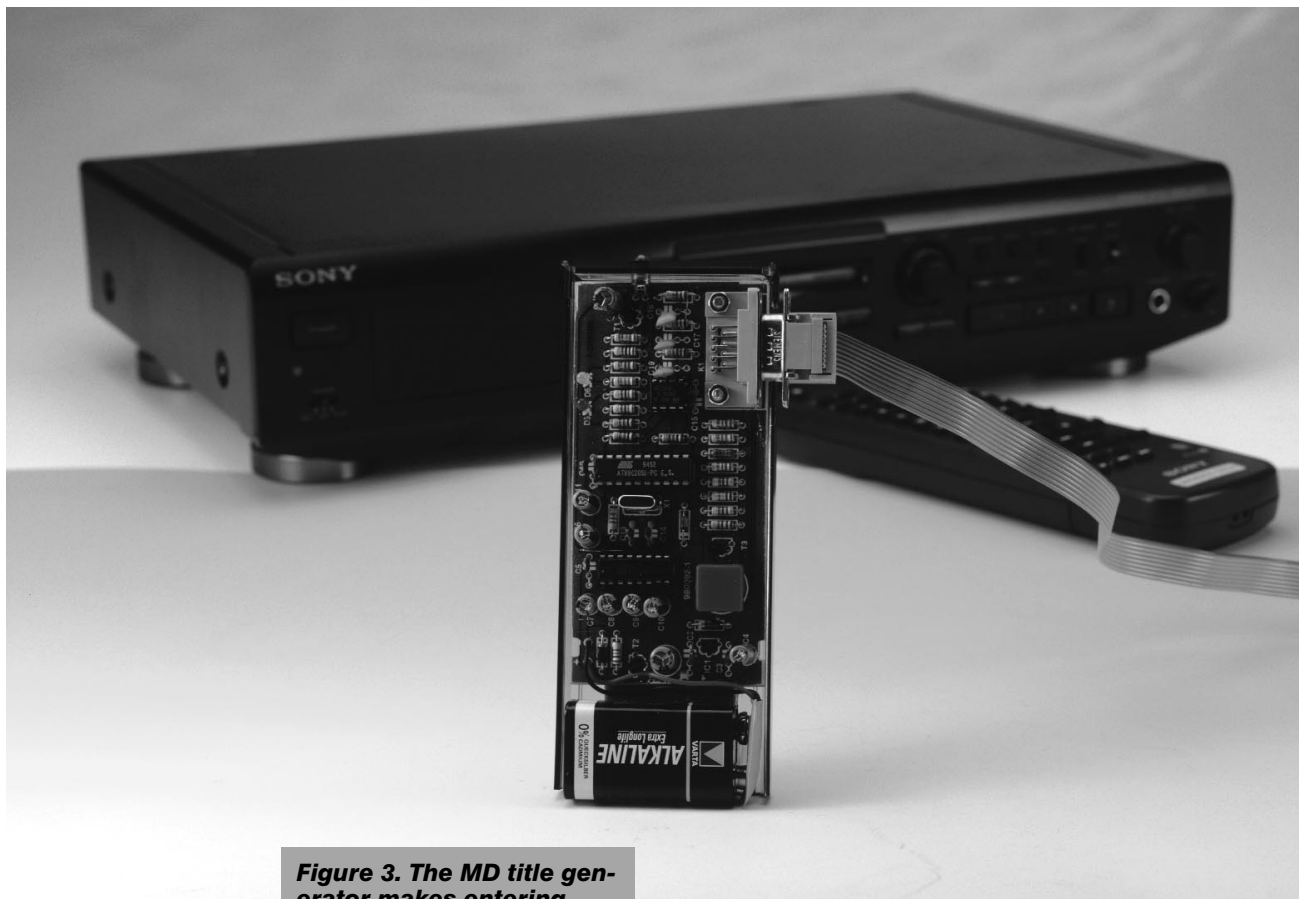


Figure 3. The MD title generator makes entering track titles into an MD deck much easier than with the remote control!

of the circuit is to be able to 'beam' data from the PC to the MiniDisc player.

The transmitting of data is accomplished by the transmitter stage built around transistor T1 and IR sender diode D3. This stage effectively converts pulses received from the micro-controller into infra-red light signals the MD is capable of detecting.

CONSTRUCTION: SIMPLE AND COMPACT

A circuit like this simply demands a compact construction, and we believe we have been successful in this respect. As you can see from the photographs in this article, the size of our prototype may be compared to that of a regular remote control. The PCB has been designed such that a standard case from Heddic is easily adapted for a perfect fit. The board, the battery, the sender diode, the control push-button, the serial connection and the two LEDs form a compact unit.

The copper track layout and component overlay of the circuit board may be found in Figure 2. The board is single-sided, contains one wire link, and accommodates all components. Before you start soldering, it is useful to first determine the exact locations of the holes for K1 and S1. The actual soldering work, we reckon, is not likely to cause problems. Once the circuit is fully built up, it may be fitted into the encl-

sure, and the battery may be connected. The unit is then ready for use.

ENTER THE PC

As already mentioned, the present control allows comfortable programming of track titles on a MiniDisc. In principle, there are two ways of storing your data in the EEPROM. Using a simple terminal program which may be found on almost any PC, laptop PC, organiser or, indeed, any other computer system, it is possible to 'get in touch' with the circuit. In all cases, select the following serial port settings:

Baudrate:	9600
No. of bits:	8
Parity:	none
Stop bits:	1
Handshaking:	hardware

After switching on the circuit (press the push-button briefly) the red LED lights up, and the circuit identifies itself by producing this message on the PC screen:

```
MD programmer V1.1
Title input help for Sony MD-Players
By Peter Glatzel (February 98)
Copyright Elektor 1998-09-08
```

Equipment types JE500...JE700

If this message appears, the circuit and

the PC link are fully functional, and the MD control is waiting for a command.

In command input mode, the following commands are recognised:

- d dump EEPROM contents to PC via RS232 port
- c clear EEPROM contents
- m manual input mode
- e1 select character set of equipment from MDS-JE series
- e2 select character set of equipment from SAJ series.

If you have an MD player from the SAJ series it is essential to be able to switch between the two equipment families supported by the control. In all other cases, the correct character set has already been selected for you.

To begin with, select 'm' via your terminal. Next, enter the MD title and the track titles:

```
00 MD title (CR+LF)
01 Title of track no. 1 (CR+LF)
...
99 Title of track no. 99 (CR+LF)
# End of input file
```

In this simple way, all data may be entered. Apart from letters and numbers, the following characters are also acceptable:

!"\$%&'()*+,-./:;<=>?@_`.

The 'hash' character (#) at the end of

each file is not transmitted to the MD player. Instead, it is used by the circuit to detect the end of the transmission. As soon as the # character is received, the circuit is switched off.

The dump ('d') command allows the EEPROM contents to be evaluated. If everything is all right so far, you may proceed to the next step.

The circuit is disconnected from the PC, and the IR sender LED is pointed at the MD deck — the system is ready to start transmitting data. At this point, it is essential to have a MiniDisc in the player, and that the player is ready for text entry. Press the button on the MD control, whereupon the yellow LED lights. The titles you programmed should appear one by one on the MD player's text display. After the MD player has updated the TOC (table of contents), the disc contains the new title list.

While testing and developing this circuit, it was discovered that some MD players insist on 3-digit track numbers. In that case, put an extra 0 before each track number. You can find out if your player requires the third digit by reading back the MiniDisc TOC — the first character of the track title is then missing.

The other method of programming the MD player is even user-friendlier. All you have to do is use a word processor to create an ASCII file which contains all track titles preceded by their track numbers. The final file will be identical to the text transmitted line-by-line in command input mode as discussed above. In ASCII mode, too, the end of your file is indicated by a '#' character.

Next, give the file a name, save it on hard disk, and start the terminal program. Make sure the program is properly configured, and select the option 'transmit file'. After entering the desired file name, the file is actually transmitted and stored in the EEPROM on the MD control board. Of course, this requires the control board to be switched on, and the yellow LED should light. 'Beaming' the file from the MD control to the MD player is then identical to the process outlined above.

(980092-1)

Note:
readers wishing to know more about the many possibilities of the MiniDisc system are advised to visit the a special Internet site dedicated entirely to the MiniDisc. The address is <http://www.czechin.com/minidisc>. On this site, extensive information is available on various technical aspects of the MiniDisc system, including track-title entry using a PC.

About terminal programs

Setting up a terminal ('communication') program may present some unexpected difficulties. Here, we provide some guidance to configuring the standard terminal program. In **Windows 95/98**, go to **Accessories and launch HyperTerminal.exe**. Enter a name for the new connection, say, 'MD Programmer'. Then, you are ready to make a configuration file for this connection.

In the first window, you select the serial port you are using — in the example, COM2 is used. Obviously, the port you want to use has to be available! Click on OK.

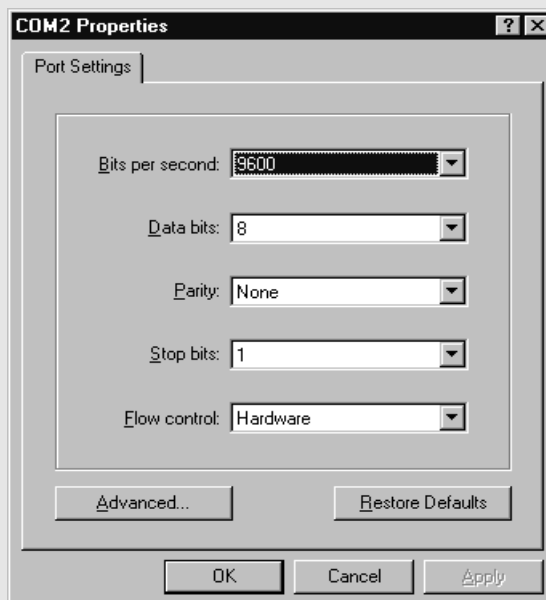
In the second window (Port Settings) the essential setting is '9600' for the bit rate.

Next, go to **File - Properties**, select 'ASCII Setup', and activate the option 'append line feeds to incoming line ends'. Click on OK and close off the configuration.

The next time you want to use HyperTerminal to communicate with the MD Control, all you have to do is double-click on 'MD programmer.ht' to enable the right configuration.

In **Windows 3.1**, things are a bit more complicated, mainly because the terminal program that comes with this operating system

works in a slightly different way. In the Settings menu, you use the 'Communications' option to configure the serial port. Apparently, the handshaking used in the Win 3.1 Terminal program is different from HyperTerminal, because Terminal is not capable of error-free file transmission with 'hardware handshaking' enabled. The only way to solve this problem is to go back to the Settings menu, select Text Transfers, and pick 'character at a time' for the flow control type. This results in characters being sent one by one to the 'remote' station, with 100-ms pauses. Obviously, this communication is much slower than with HyperTerminal, but there you are.



RFID systems

using smart cards

Now chip cards, and, perhaps more particularly, smart cards, are widely used in large parts of the world in a variety of applications, communications engineers have come up with another one: Radio Frequency Identification (RFID) readers. In this application, chipcards are used in conjunction with resonant circuits (antennas), but wireless and without physical contact, for the identification of persons, animals, objects or for process control. The technique is not new, but the means of applying it is.

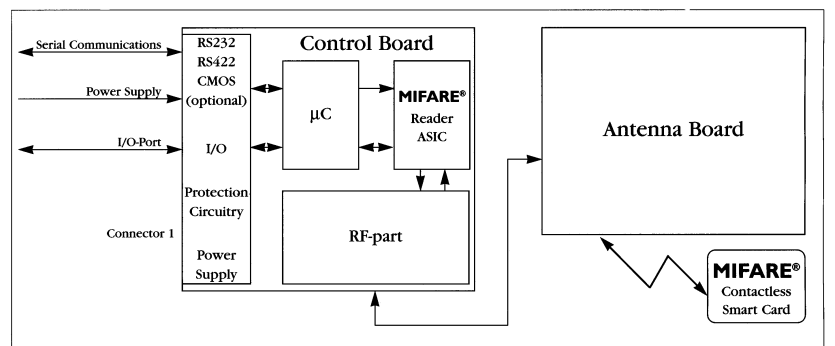


Figure 1. Block diagram of a Serial Proximity Reader consisting of a Philips/Mikron VLSI reader IC, an RF module, antenna, and microcontroller for the serial interface.

In essence, a Radio Frequency IDentification (RFID) system (a typical setup is shown in **Figure 6**) consists of a transmitter-receiver to transmit an interrogation signal and receive an acknowledgment from the responder in the interrogated reader (RFID device). A similar system in terrestrial communications is called a relay or repeater station, and in satellite communications, a transponder.

In many applications, the responder, that is, the part of the reader that replies automatically to a correct interrogation signal, is passive. This means that the radio frequency field generated by the antenna is used not only to transmit the requisite data, but also, by induction, to supply the necessary energy for the responder. This is the real novelty in the use of chipcards.

The use of chipcards in a contactless and wire-less manner has a further advantage in that it allows a variety of constructions to be employed. An RFID reader can be integrated in objects like rubbish bins or transport

containers, and, owing to its small size, be implanted into people and animals. They may also be incorporated in bracelets or wristwatches or, indeed, in cheque cards.

RFID READER

As in the case of chipcards in general, an RFID system has a reader for receiving and evaluating an interrogating signal from a transmitter-receiver. Only in key systems and similar applications is the reader a read-only device. In most other systems a reader's programmed information may be modified (written) by the user, so the reader is a read/write device.

A simple reader consists of a small module which only needs an antenna and a supply voltage. The antenna usually consists of a small inductor whose dimensions depend on the frequency range in which the system operates and the working range needed.

In a simple key system, the module has a switched output to which a relay may be connected or via which an elec-

tric door opener is controlled. All this happens, of course, only if the interrogation signal is evaluated by the microcontroller in the module as correct. In more complex applications, the evaluation is followed by the programming of the system by a small computer. The reader is then provided with a suitable interface.

The block diagram of such a reader is shown in **Figure 1**. This is a version of the MIFARE® Serial Proximity Reader from Philips/Mikron. It consists of a VLSI (Very Large Scale Integration) Reader Chip, an RF module, an antenna and a microcontroller for the serial interface. These are mounted on a printed-circuit board (PCB) with the RF part of the circuitry being shielded by a ferrite against the electromagnetic influence of the electronic part.

The reader operates at the standard ISO (International Organization for Standardization) frequency of 13.56 MHz. The operating range is about 10 cm (4 in). The signalling speed is 115 kilobaud.

This type of card is used primarily in slot or surface readers, such as ATMs (Automatic Teller Machines), vending machines, parking meters, and public telephones.

In most other RFID applications, the operating frequency is lower: 100–450 kHz, which enables larger operating ranges. The most popular frequency is 125 kHz, but 120 kHz, 134.2 kHz and 150 kHz are also encountered quite often. The frequency of 134.2 kHz is used primarily in Texas Instruments' TIRIS system.

The antenna is a round, rectangular or square, potted, air-cored inductor or, in the 100–200 kHz range, a ferrite rod.

Figure 2 shows a read/write module with integrated passive antenna for use in 13.56 MHz RFID systems. This module is controlled via a serial interface (RS232). The operating range is about 25 mm (1 in).

Larger operating ranges, up to a few metres, are achieved with operating frequencies in the 100–200 kHz range, and rectangular inductors that form effective frame antennas. The sides of such an antenna may be up to one metre long. Clearly, the electronic circuits are then separate from the antenna, which is located, for instance, in the entrance/exit gate of a shop, store or car park. When the distance between reader and antenna is large, active antennas or used.

CARD ICs

The heart of a contactless smartcard is a card IC. This single chip stores the data to be transmitted and ensures their secure transfer. The IC is usually mounted together with the capacitor for the resonant circuit on a small wafer as shown in **Figure 3**. All this

2

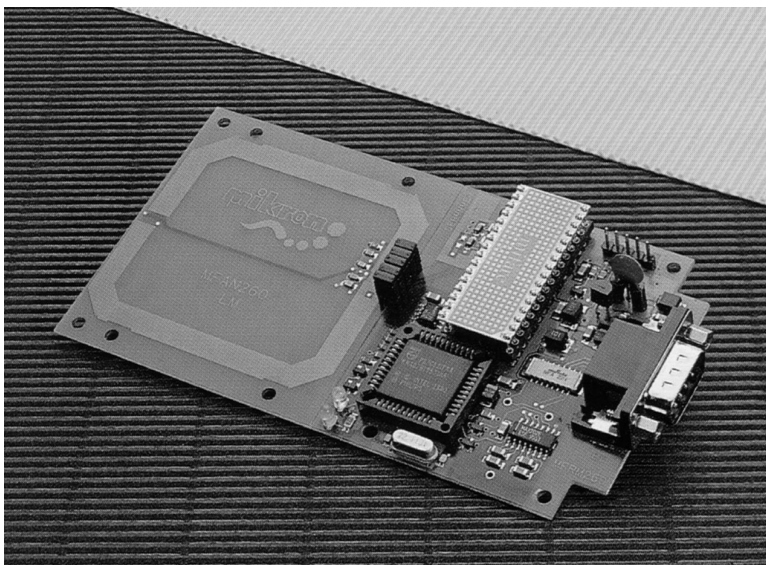


Figure 2. Short Range Serial Reader – a fully functional read/write module controlled via a serial interface (RS232). Note the integrated (printed) antenna on the wafer. The reader has an operating distance of 25 mm and auto baud rate selection (9 600–115 200 baud).

module needs is a coil (antenna) to be connected to its two terminals.

Figure 4 shows the block diagram of a typical 13.56 MHz card module. The functional groups at the righthand side of the diagram, that is, the digital section, are found in most smartcards. An exception is the anti-collision facility, which enables the reader to function with a number of cards in its capture range – which is, of course, impossible with a standard chipcard.

The RF interface, in conjunction with a tuned antenna connected to the

two terminals at the side, provides the wire-less connection (including the supply voltage) to the reader. The design of the antenna depends on the operating frequency, the dimensions of the enclosure, and the desired operating range.

FUNCTIONAL DESCRIPTION

Generation of the necessary supply voltage and modulation is achieved by two loosely coupled resonant circuits as shown in **Figure 6**.

3

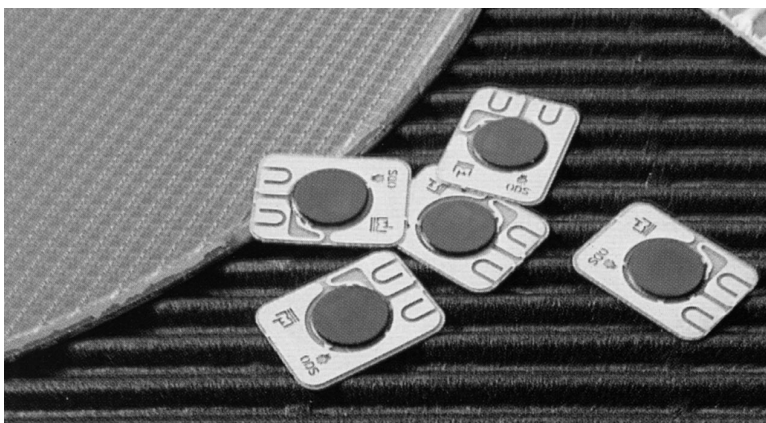


Figure 3. These standard card ICs form the heart of contactless smartcards. To produce one of these cards, the IC need only be connected to a rectangular coil with a few turns, which can be printed, etched or wire coil. The card operates at 13.56 MHz and has an operating distance of 10 cm. It is suitable for multi-application and has an anticollision facility. High-speed read/write communication (106 kbaud) ensures completion of ticketing transactions in less than 100 ms.

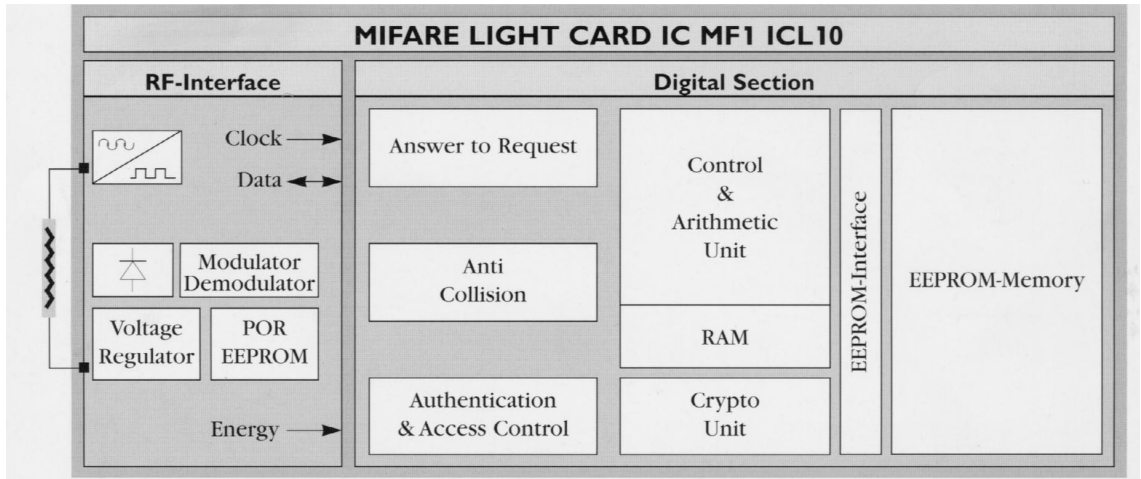


Figure 4. Block diagram of the 13.56 MHz light card IC

The supply voltage is obtained from the electromagnetic field induced on the attached (antenna) coil. The RF voltage is rectified to provide the requisite direct voltage. When this voltage crosses the power-on level, the RFID reader enters the standard read mode and sends data continuously. The data to be sent in this mode is user-defined by storing the first and last addresses to be output. When the last address is sent, the device will continue with the first address until the transmitter-receiver sends a request.

In the read mode, a listen window (LIW) is generated before each word. During this time, the reader will turn to the receive mode (RM) if it receives a valid RM pattern. The reader then expects a valid command.

A 1024 bit EEPROM in the RFID reader is organized in 32 words each of 32 bits. The first

three words are assigned to the password, the protection word, and the control word. To write one of these words, it is necessary to send a valid password. The password cannot be read out. The memory contains two extra words of laser ROM. These words are laser programmed during fabrication for every IC, are unique and cannot be altered.

The data modulator in the transmitter-receiver is driven by the serial data output from the memory in the RFID reader which is encoded. The modulator will draw a large current from both coil terminals, thereby amplitude modulating the RF field according to the memory data.

The RFID reader modulates the amplitude of the RF field to transmit data to the transmitter-receiver. Data are output serially from the EEPROM and Manchester encoded.

Different patterns are used to send status information to the transmitter-receiver. Their structure cannot be confused with a bit pattern sequence. These patterns are the listen window (LIW) to inform the transmitter-receiver that data can be accepted, the acknowledge (ACK) indicating proper communication and end of EEPROM write, and the no-acknowledge (NACK) when something is wrong.

The LIW, owing to its special structure, can be used to synchronize the transmitter-receiver during a read operation. It is sent before each word.

The RFID reader can be switched to the receive mode only during a listen window. The transmitter-receiver is synchronized with the incoming data from the reader and expects a LIW before each word. During the phase when the reader has its modulator on, the transmitter-receiver has to send a 0. A certain phase shift in the read path of the transmitter-receiver is acceptable since, when entering receive mode, the transmitter-receiver becomes the master.

At reception of the first 0, the RFID reader stops the LIW sequence and then expects another 0 to actuate the receive mode. Once the reader has received the first 0, the transmitter-receiver is imposing the timing for synchronization. The reader turns on its modulator at the beginning of each frame of a bit period.

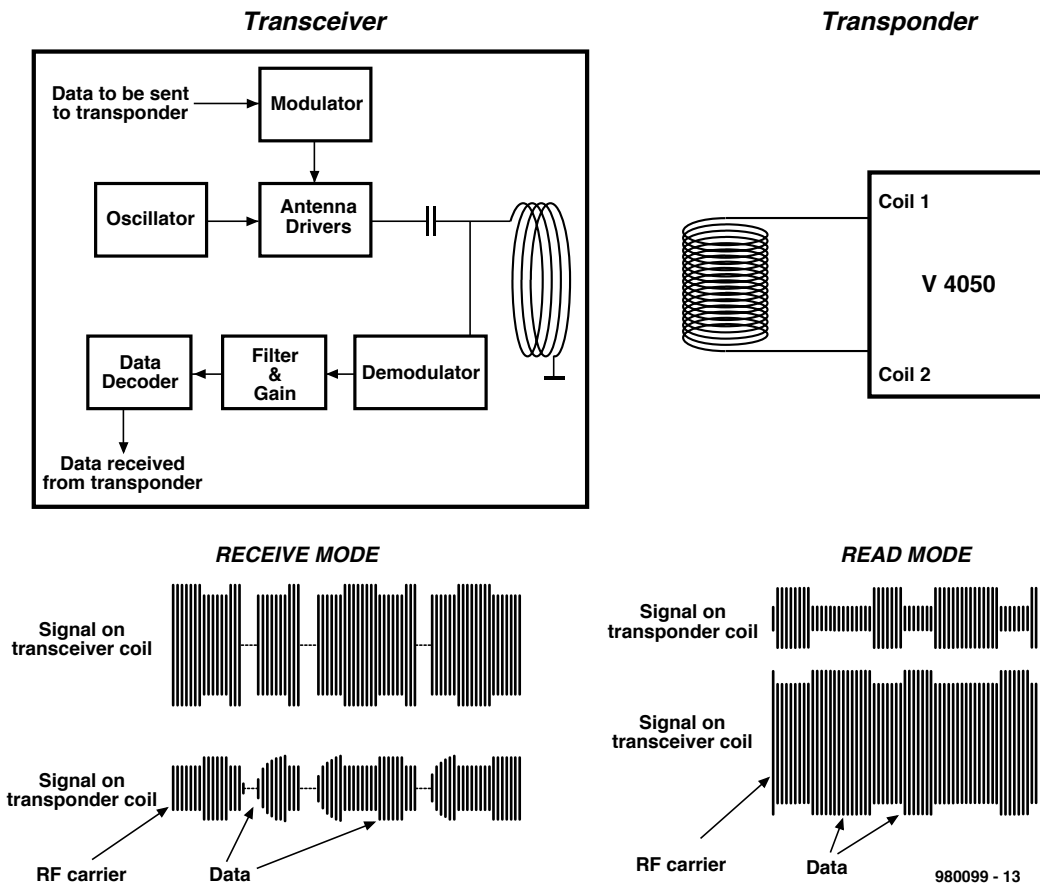
To send a logic 1, the transmitter-receiver continues to send clocks without modulation. Half a bit period later, the modulation device in the RFID reader is turned off, allowing the internal supply capacitor to be recharged.

To send a logic 0, the transmitter-receiver stops sending clocks (100% modulation) during the first half of a bit period. The transmitter-receiver does not turn off the field during the first 7/4 clocks of the bit period. The field is stopped for the remaining first half of the bit period and then turned on again for the second half of the bit period.

Figure 5. This contactless smartcard is the data carrier of the Philips/Mikron Mifare® RFID system designed for electronic ticketing. The card is suitable for use in public transport, parking meters, electronic keys, road tolls, ID cards, and more.



6



To ensure synchronization between the transmitter-receiver and the RFID reader, a 0 is transmitted at regular intervals. The RM pattern consists of two bits set to 0, so allowing initial synchronization. In addition, the chosen data structure contains even parity bits which will not allow more than eight consecutive 1s where no modulation occurs.

While the transmitter-receiver is sending data to the RFID reader, two different modulations will be observed on both coils. During the first half of the bit period, the reader switches on its modulation device causing the RF field to be modulated. This modulation can also be observed on the transmitter-receiver coil. When the transmitter-receiver sends a 0, the RF field is switched off, causing 100% modulation on the reader coil.

The transmitter-receiver may consist of a single IC, such as the 90106 from Melexis UK, the block diagram of which is shown in Figure 7.

In this, there is a modulation input (MODU) for the oscillator. The RF signal can be shifted by a direct voltage of 0-3 V applied to this input (0 V is peak value; 3 V oscillator disabled). If the input is left unterminated, the (internal) modulating voltage is 1.5 V. This ensures the best sensitivity for the RFID reader, whose data are available at the DATA pin. The coil parameters are calculated as shown in the box.

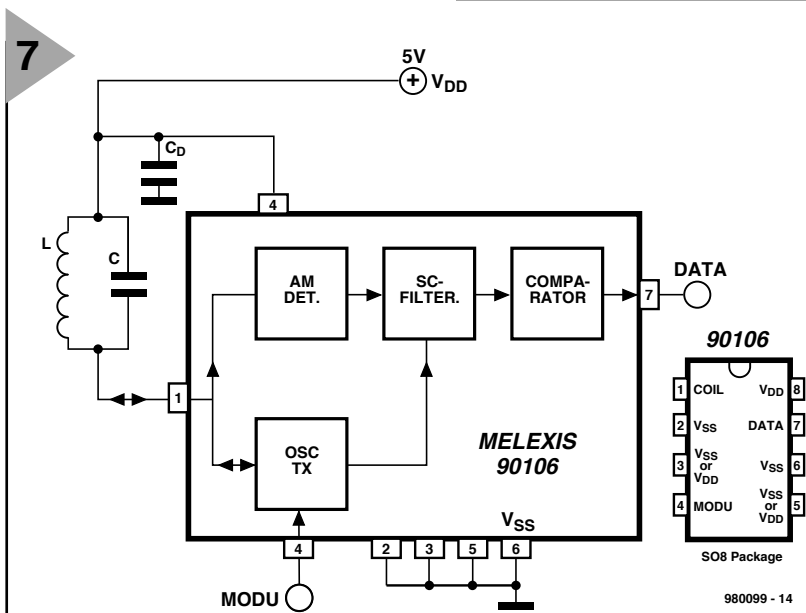
Apart from amplitude modulation (AM), RFID systems may use Manchester encoded AM, phase-shift keying (PSK), frequency shift keying (FSK), and frequency modulation (FM).

CONSTRUCTION

Since RFID readers consist of an IC and an antenna only, their enclosures can take many shapes and be of many types of material. Polyester and plastic

Figure 6. Functional diagram of a passive, inductive RFID system. The 4050 is a 1 kbit read/write contactless RFID device (reader) from Microelectronic-Marin.

Figure 7. The transmitter-receiver used in RFID systems may be a single IC as shown by the block diagram of the 90106 from Melexis UK.



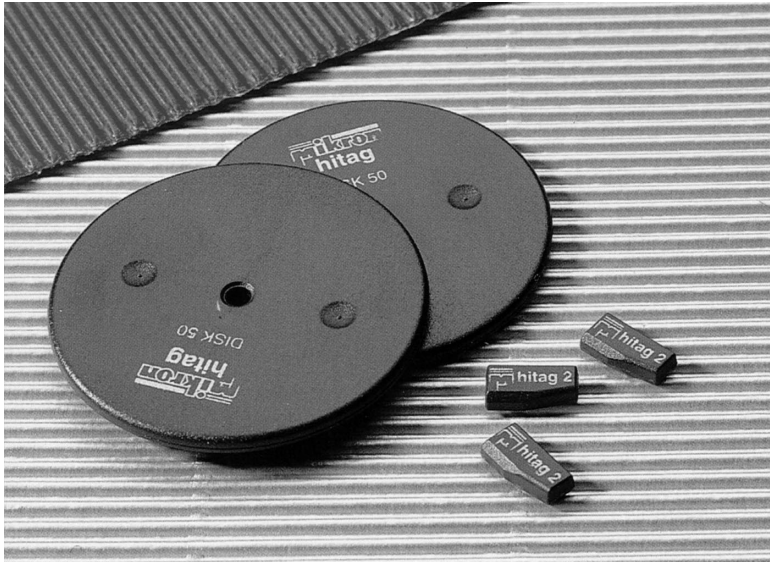


Figure 8. Multimode RFID readers may be housed in disk-shaped or rod-shaped plastic enclosures. These units are particularly suitable for use in car immobilizer systems, as animal ID, in asset tracking, in electronic keys, and many more.

cases dominate in the small units, whereas the more robust types use metal carrier plates. Units for use in damp conditions or in chemical plants are hermetically enclosed in glass cases. Some are very small (2 mm diameter and 12 mm long) and therefore suitable for implanting into farm animals or domestic pets for recognition purposes. Implanting is carried by (virtually painless) injection under the skin with the aid of a special veterinary gun.

RFID readers come in the shape of cards, disks, pills, or rods. Pill-shaped units are particularly suitable for identification of pallets, containers, and motor vehicles, while small rod-shaped ones are used for integrating into car keys. Small units (glass or plastic) used for animal implantation, called animal-IDs are designed and produced to a standard ISO protocol: ISO 11784/85.

RFID systems and components are available from a number of manufacturers as listed below. It should be

noted, however, that, owing to the specialized equipment needed, this field is not suitable for home design and/or construction. It may, of course, become so during the next few years, but the signs are not encouraging.

Ready-made small systems for, say, electronic locks are, however, available from some stockists. Also available from several manufacturers for the professional designer are development and demonstration kits.

MISCELLANEOUS

Information from the following manufacturers was used in the preparation of this article.

Philips/Mikron (The Netherlands)

Melexis UK (United Kingdom)

Siemens (Germany)

Temic (Telefunken Microelectronics) (Germany)

Texas Instruments (USA)

EM Microelectronic-Marin (Switzerland)

Philips, in association with Mikron, has a range of RFID products under the name Mifare®. This range includes readers for use at 13.56 MHz with an operating range of about 10 cm. The company also produces Hitag units for operation in the 125 kHz band with an operating range of up to one metre.

Telefunken Microelectronics offers a range of RFID readers in the 100–400 kHz range. These units are inexpensive and particularly intended for large-quantity production. The range includes a read-only version that has a 128-bit PROM and can be used in the 100–400 kHz range at temperatures from –40 °C to +125 °C. The write unit is intended for use in the 100–150 kHz band.

Texas Instruments has a family of RFID readers, named TIRIS, that ranges from contactless 64-bit chip-cards to heavy-duty units for use in goods vehicles and containers. All units operate on the same frequency of 134.2 kHz, and use frequency-shift keying (FSK). Many units in the range can operate at distances of up to two metres, which necessitates the use of large frame antennas for the reader and ferrite rod antennas for the transmitter/receiver. The largest unit in the TIRIS range is housed in a glass tube 31 cm long and 4 cm in diameter.

More information may be obtained from 'Transponder News', Internet News Services:

www.rapidftp.com/transponder

In the United Kingdom contact Melexis UK, 3 Anglesey Close, Chase-town, Staffs WS7 8XA; telephone/fax 01543 670391.

[980099]



Figure 9. Evaluation and demonstration kits of RFID systems are available from several manufacturers. Those shown are from Philips/Mikron.

Calculating the inductor

In the following it is assumed that the (RFID) reader operates at a frequency of 120 kHz. So as to minimize the influence of parasitic capacitances on the resonant frequency of the RC circuit, it is advisable to keep the value of the inductance as low as feasible (since this requires a larger circuit capacitance, C).

The fieldstrength produced by the inductor, H , is directly proportional to the number of turns, n , and the current, i , flowing through the coil:

$$H = ni = nv/\omega L \quad [\text{Eq. 1}]$$

where v is the alternating voltage across the coil, L is the self-inductance of the coil, and ω is the angular frequency, that is, $2\pi f$.

The self-inductance of the coil is directly proportional to the square of the number of turns:

$$L = L_0 n^2, \quad [\text{Eq. 2}]$$

where L_0 is the self-inductance of a single turn of the coil.

Combining the two equations yields:

$$H = v/\omega L_0 n \quad [\text{Eq. 3}]$$

Equation 3 shows that a reduction of the number of turns increases the fieldstrength (and simplifies winding the coil).

Practical values for a 120 kHz resonant circuit (antenna) are 0.025 μF for the capacitor and 70 μH for the inductor. Such an (air-cored) inductor would consist of 30 turns with a diameter of about 30 mm.

TERMS USED IN RFID SYSTEMS

Anticollision

A facility to enable the system to recognize the correct RFID units when a number of these units are within the operating range.

Delta modulation (DM). In delta modulation, the level of the sampled analogue signal is coded by one bit, positive or negative, based upon whether a signal sample is greater or smaller than the previous value. Because only one bit per sample needs to be transmitted, the sampling rate can be increased significantly to reduce the quantization noise, simplify the anti-aliasing filter, and reduce the bandwidth required for transmission.

Delta-sigma modulation (DSM). Delta modulation systems cannot handle signals with a d.c. component and have a signal-to-noise ratio that is sampling and modulation frequency dependent. Both of these disadvantages are overcome by using the delta-sigma technique, which employs an integrator circuit at the input to the modulator. The amplitude of the output pulses then conveys information about the signal amplitude and not just its variation.

Frequency shift keying. Non-coherent demodulation produces two channels in the decoder, each processing the 0 and 1 frequency respectively. Envelope detection in each channel then produces a pulse to coincide with the transmission of each 0 or 1. The output bit stream is finally produced by combining the two channel outputs. The coherent technique uses two product detectors, one for each channel or bit carrier frequency. The decoded data stream is obtained again by

combining the two channel outputs. **Manchester code.** This is a bi-phase format in which each bit in the original signal is represented by two bits in the derived format. The basic rule for the transform is that 0 is represented by 01 and 1 by 10. This ensures that there are never more than two identical bits in series.

Multifunctionality. This term pertains to a card that can be used for several different purposes; for instance, a bank card combined with a credit card

Multipage transponder (MPT). A transponder whose memory is organized in several pages, for instance, 1360 bits in 17 pages each holding 80 bits.

Multiple security (access). Access control to a communications system according to the security status of the user, and the degree of classification of the facility required.

Phase shift keying. Only coherent decoding can recover the data from such a modulated signal. It is, therefore, pre-filtered to narrow the detection bandwidth and minimize the noise before being input to a product detector stage. Low-pass filtering is used to recover the data stream from the unwanted harmonic components of the carrier frequency.

Read only. An RFID device whose programmed information cannot be modified.

Read/write (R/W). An RFID device whose programmed information may be altered by the user.

Responder. That part of an RFID reader that automatically transmits a reply to the interrogating transmitter-receiver.

Responder. The receiver in an RFID reader that accepts and interprets the signals from the interrogating transmitter-receiver.

Selective addressable multipage transponder (SAMPT). A transponder similar to an MPT but whose first page holds a 24-bit address data field, which can be programmed by the user to enable one of a group of transponders to be accessed

Smart card. A plastic credit card sized device in which some simple computing facilities are embedded. The card contains typically a purpose-developed 8-bit microprocessor accompanied by up to 20 kbytes of ROM to carry the application program, 4 kbytes of EPROM that provides non-volatile memory of the card user status, 512 bytes of RAM to hold the data associated with the calculation for a single transaction, and registers that may be 512 bits long. There are currently two types of this card: one with contact pads along one edge surface, and the other that is accessed via radio-frequency signals. The latter was developed for use in the GSM mobile telephone service in which the user has to input a personal identity number (PIN), and if this is accepted, the user can proceed to the next stage.

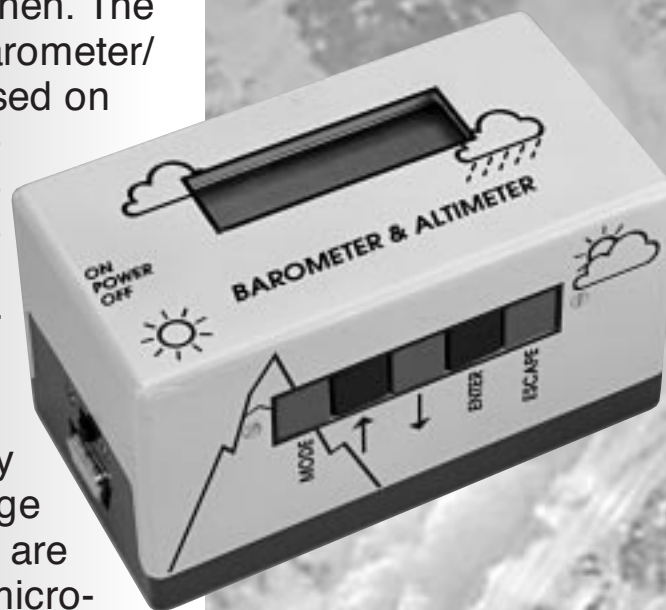
Tag. A device attached to an item to be tracked. There are two basic concepts: active and passive. An active tag is powered by a long-life battery, whereas a passive device is signal-powered. An RFID reader may be used as a tag.

Transponder. (1) The name given in satellite communications to what is called a relay or repeater system in terrestrial communications. (2) In radar engineering a form of transmitter-receiver that automatically transmits signals when the correct interrogation is received.

barometer/altimeter

Part 1

It is just about ten years ago that this magazine published the design of a barometer/altimeter. In these ten years electronics technology has progressed to developments that were unimaginable then. The present barometer/altimeter is based on a precision air-pressure sensor with integral signal processor from Motorola. Evaluation, storing, display and interchange of data are effected by a micro-controller system.



meter with many facilities

Features:

- X Barometer function
- X Altimeter (absolute or relative height measurement)
- X Height measurement from -2000 m to + 10 000 m
- X Relative height measurement referred to arbitrarily chosen height
- X 1 or 2 point calibration
- X EEPROM data logger for 1000 samples
- X Log interval from 10 seconds to 8 hours
- X Log data retrievable sample by sample from the computer

Design by H. Bonekamp

Gently tapping and looking at the usual (aneroid) domestic barometer does not really say much about the weather that can be expected. Much more telling is the drift of the barometer, which shows the speed, magnitude and direction of variations in atmospheric pressure. This information gives a good idea of the weather to be expected over the next few hours. Unfortunately, an aneroid barometer is not very accurate: a mercury type (not often found in domestic premises) is better, but is nowhere near as robust and convenient to use. Note that frequent tapping does not do the barometer much good.

ATMOSPHERIC PRESSURE AND HEIGHT

If a barometer is to be used in a weather station or in conjunction with an external temperature controller in a heating installation, it should be a microcontroller-driven model. Such a unit is very accurate and offers a number of additional facilities as in the present design.

The present design has a total maximum error of not more than two per cent, which is very small for a barometer. It also enables the measurements to be stored or to be supplied to a computer.

Apart from atmospheric pressure, a barometer can, at least in principle, show the height above sea level. But in this, other factors such as humidity and temperature also play a role. Unfortunately, the present design cannot compensate for these factors, since that would make it much more complex. To use a barometer as an altimeter, it needs a different scale. The relationship between the real atmospheric pressure, p , at a certain height, h , and the atmospheric pressure at sea level, p_s , is given by Equation 1:

$$h \left[\frac{p}{p_s} \right] = \frac{1 - \left[\frac{p}{p_s} \right]^{5.2563}}{22.555 \times 10^{-6}} \quad [\text{Eq. 1}]$$

A distinction must be made between relative height and absolute height. A relative height, h_r , is referred to an arbitrarily chosen height, h_o :

$$h_r = h \left[\frac{p}{p_s} \right] - h \left[\frac{p_o}{p_s} \right] \quad [\text{Eq. 2}]$$

where h_r is the relative height, p_o is the atmospheric pressure at h_o , and the other parameters are as in Eq. 1.

Measuring relative height can only be done accurately when the atmospheric pressure at the reference height is virtually constant.

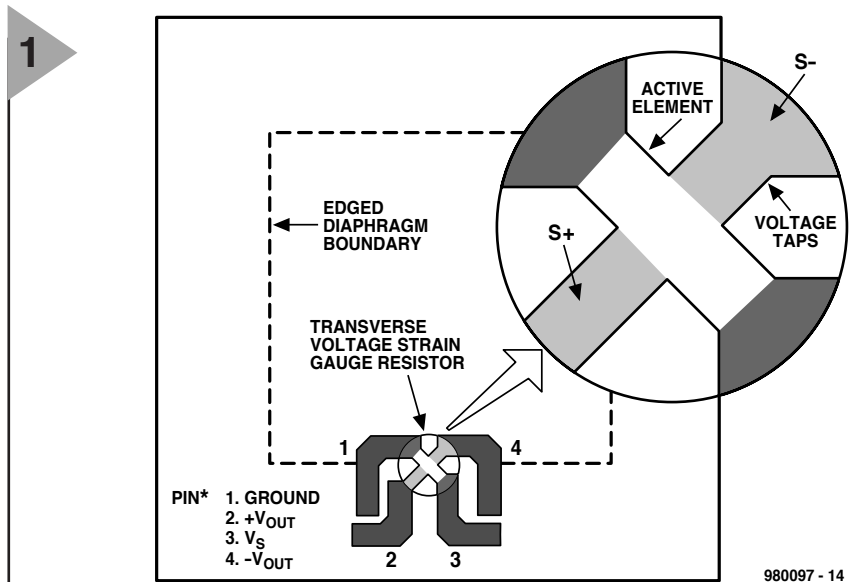


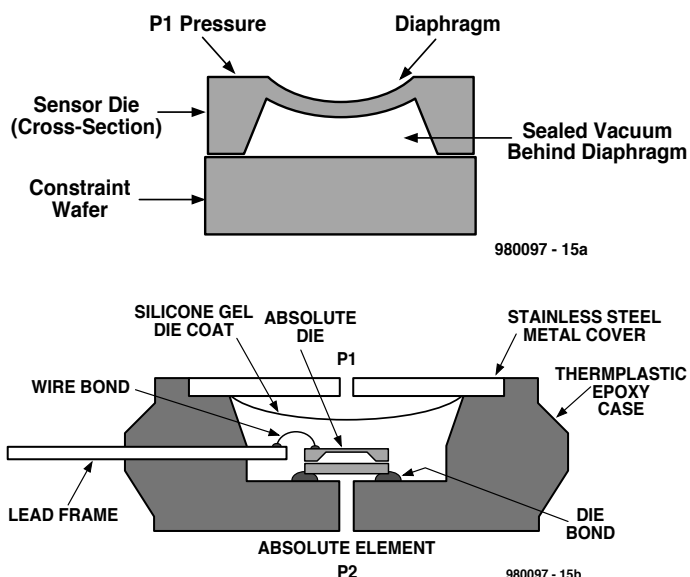
Figure 1. The sensor consists essentially of a diaphragm onto whose edge a piezoelectric element is fused.

The absolute height, h_a , is referred to sea level and is calculated with Equation 3:

$$h_a = h \left[\frac{p}{p_s} \right] \quad [\text{Eq. 3}]$$

Since the measured absolute height

Figure 2. The pressure sensor for absolute measurements has an evacuated space behind the diaphragm.



depends solely on the atmospheric pressure at sea level, p_s , must be regularly checked and adjusted in accordance with the actual pressure at sea level when extended measurements are carried out.

SENSOR

The atmospheric pressure is evaluated by a Barometric Absolute Pressure (BAP) sensor Type MPXS4100A from Motorola. The sensor contains a thin diaphragm that is depressed to a greater or lesser degree by atmospheric pressure (Figure 1). A piezoelectric element fused along the edge of the diaphragm faithfully follows the movements of the diaphragm. In doing so, its resistance varies in direct proportion to the movements and thus to the atmospheric pressure. In other words, the sensor behaves as a potentiometer.

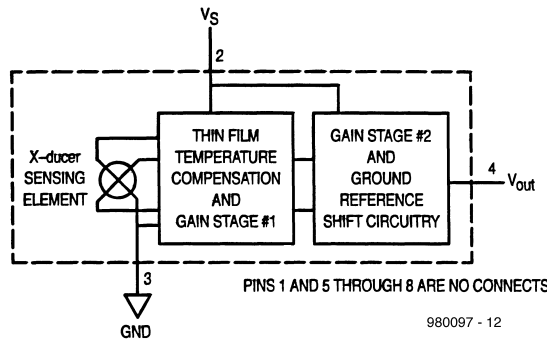


Figure 3. Block diagram of Motorola's Type MPXS4100A Barometric Absolute Pressure sensor that includes a signal processing facility.

When a supply voltage is applied across it, the resulting output voltage varies in direct proportion to the variations in pressure.

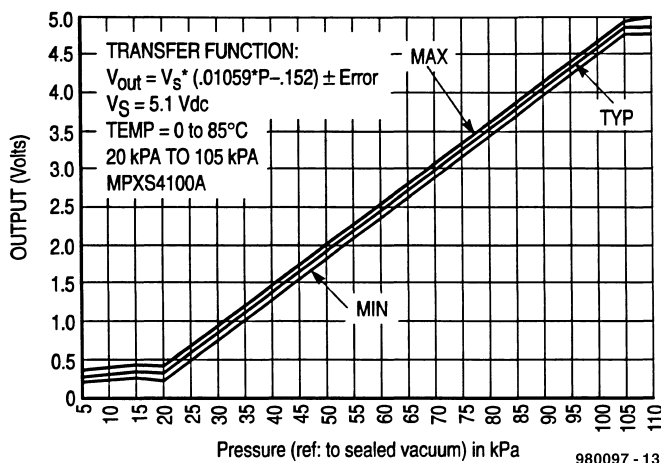
There are three kinds of pressure sensor. In a differential sensor pressure is applied to both sides of the diaphragm, whereupon the device provides a voltage that is directly proportional to the difference in pressure at the two sides.

The standard pressure sensor is constructed along similar lines as the differential model. This type uses the ambient pressure as reference, so that the output voltage is proportional to the relative pressure.

In the absolute-value sensor, used in the present design, the space behind the diaphragm is isolated from the environment and evacuated to a reference pressure (Figure 2). The output voltage is proportional to the absolute atmospheric pressure, which makes it suitable for use in a barometer, altimeter or mobile weather station. The sensor was originally developed for use in motor vehicles and is consequently readily available at a reasonable price.

The sensor provides an output voltage of

Figure 4. The pressure vs output voltage characteristic of the sensor is linear between pressures of 200 millibar and 1050 millibar.



0.3–4.9 V, which is directly proportional to a pressure range of 20–105 kPa (200–1050 millibar). It has an error of only 1.8 per cent over the temperature range 0–85 °C. It can operate at temperatures down to –40 °C and up to +125 °C. It can withstand pressures of 400 kPa (continuous) and 1000 kPa (burst).

The sensor element (Figure 3) also contains temperature compensation circuitry, two amplifiers, and shift circuitry that converts the differential measurement potential into one referred to ground. Altogether, it provides a linear output signal over the pressure range 20–105 kPa with a sensitivity of 54 mV/kPa (Figure 4). The relevant transfer function is

$$V_{out} = V_s(0.01059p - 0.1518) \pm E,$$

where
 $V_s = 5.1 \text{ V} \pm 0.25 \text{ V}$
 $p = \text{pressure in kPa}$
 $E = \text{total error}$

The linear relationship and the output voltage range allow easy connection to an analogue-

to-digital converter (ADC) and a microcontroller.

ANALOGUE - TO - DIGITAL CONVERTER

The analogue-to-digital converter is based on a 16-bit sigma-delta converter Type AD7715-5 from Analog Devices. Although at first sight a resolution of 16 bit seems a little over the top, it should be noted that when the device is used as altimeter this provides a reliable height resolution of one metre, provided that the ambient pressure is fairly constant.

The AS7715 has a balanced input buffer and a programmable amplifier (PGA). Nevertheless, since input AIN– as well as REF IN– are at ground potential, the converter operates in unbalanced mode and requires an input signal of 0–2.5 V (when the amplification of the PGA is ×1). This setup requires two potential dividers between sensor and converter to halve the input signal (R₁–R₂) and the reference voltage (R₃–R₄). The voltages are decoupled by capacitors C₁ and C₂.

The input signal and reference voltage are applied to a sigma-delta modulator, which is terminated into a digital filter. The digital output, DOUT, is available in serial form, thus allowing easy connection to a microcontroller.

The converter provides the controller with various facilities for influencing the overall operation. For instance, apart from the 16-bit data register, there are three 8-bit registers that can be read serially or set via DIN (Communication, Setup and Test).

It is important to note that the converter has a self-calibrate facility which is instigated by a relevant instruction from the microcontroller.

The converter is clocked by an integral crystal (X₁) oscillator that operates at 2.4576 MHz. The clock signal is available at SCLK.

MICROCONTROLLER

The microcontroller is an 8051-derived Type AT89S53-12 from Atmel. The use of a single IC for this purpose requires it to have RAM and ROM on board. These are both relatively small: a 12 kbyte flash-ROM in which the larger part of the program and all the constants are stored, and a 256×8 bit RAM that holds the stack and the variables. This means that the computations cannot be carried out with floating-point arithmetic, but have to be effected by quite cumbersome mathematical conversions. This will be reverted to in Part 2 next month.

Port 1 is used for controlling the ADC, reading the measurement data, and communication with the computer via the RS232 interface (MAX233).

Ports 0 and 2 are used for driving a

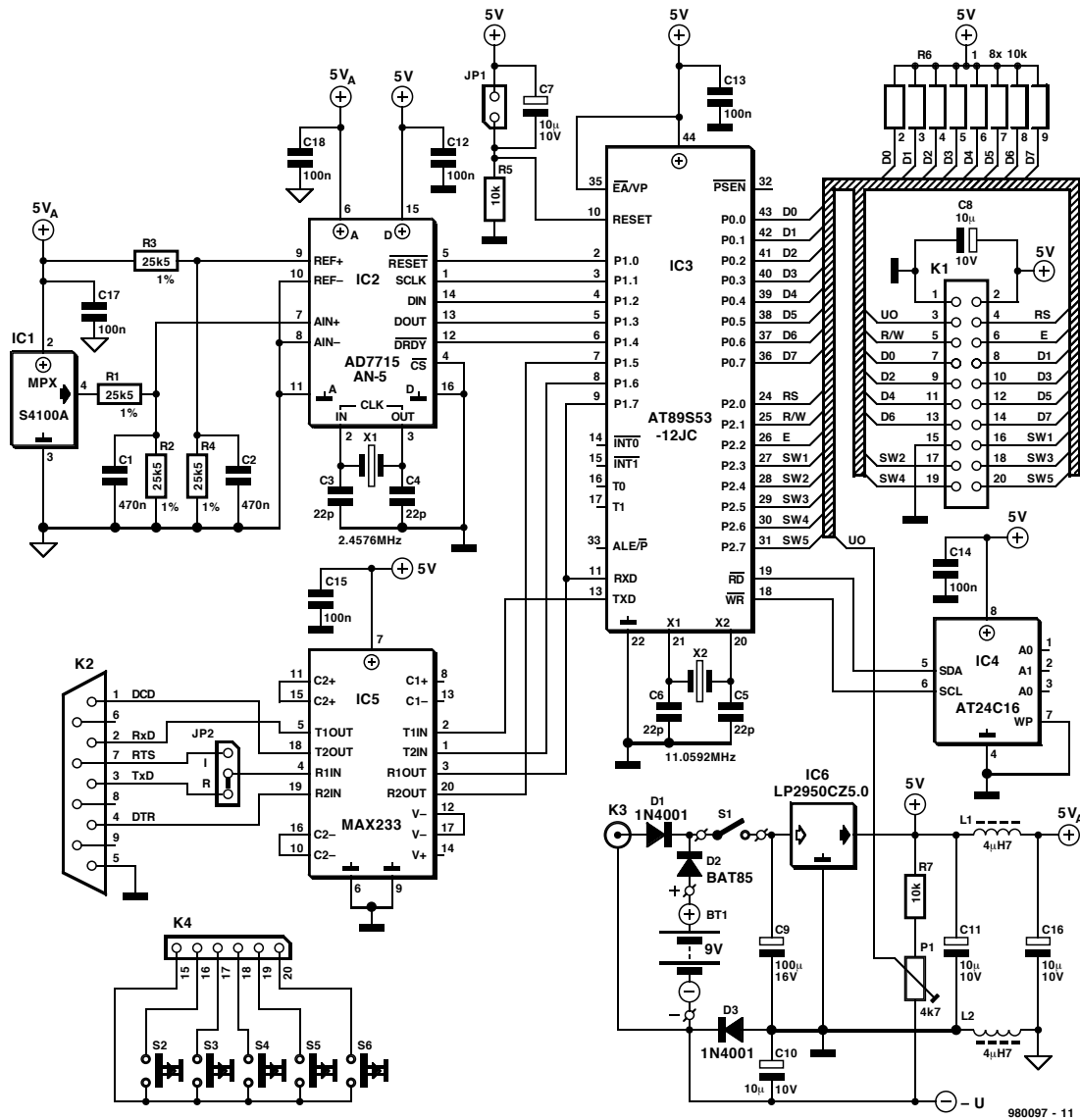


Figure 5. Circuit diagram of the barometer/altimeter, including power supply.

liquid-crystal display (LCD) and for inquiring after the operational settings of switches S_2 – S_6 located on a separate interface board.

The controller is also connected to an EEPROM that is driven serially via a two-wire link. This device has a storage capacity of 16 kbit to enable up to one thousand 16-bit samples to be stored.

The controller has a serial programming interface and can be erased and programmed in circuit (thanks to flash). The use of an in-system programmable (ISP) controller is very effective in this kind of application. Apart from the programming facility, it allows modifications to be entered into the software. More about this in Part 2 next month.

MEASUREMENT ERROR

As in all electronic measuring equipment, each and every part and component contributes, in however small degree, to the overall measurement error. As already mentioned, the sensor proper has a maximum error, E , of

$\pm 1.5 \text{ kPa } (\pm 15 \text{ millibar})$ over the temperature range 0 – 85°C . The error is up to three times as great (because of the temperature-error factor—TF) over the temperature ranges -40°C to 0°C and $+85^\circ\text{C}$ to $+125^\circ\text{C}$. The overall error is

$$E = \pm (1.5 \times \text{TF} \times 0.01059 \times V_s)$$

which is taken account of in Figure 4.

The input signal is halved by potential divider R_1 – R_2 . The tolerance of these resistors may cause an additional error of up to two per cent in the output voltage. The same is true of the resistive divider halving the reference voltage. From this it is clear that these resistors should be carefully selected, if at all possible.

Another possible source of error is the ADC. Fortunately, as mentioned earlier, this has a self-calibrate facility which sets the reference point and the maximum output correctly every time

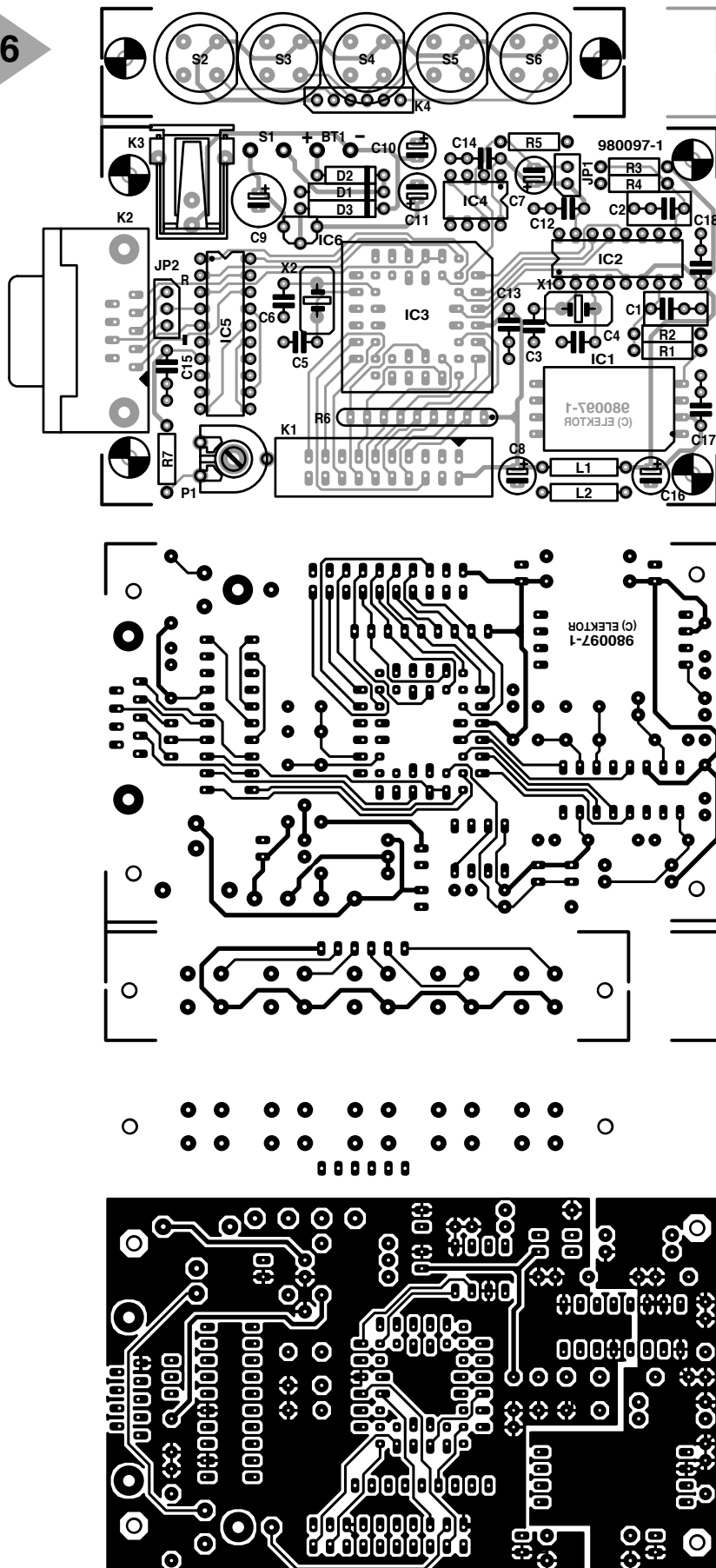
the unit is switched on and every thirty minutes thereafter.

Any errors caused by the amplifiers, temperature variations and missing codes may be ignored. Only the non-linearity of the ADC (0.0015 per cent) must be included in the error computation.

All in all, the controller is supplied with data that have an error of not more than 1.8 per cent if the unit is carefully balanced, and up to 5.8 per cent in an unbalanced unit.

SUPPLY VOLTAGE

The supply voltage is derived from a 12 V mains adaptor or a 9 V dry or rechargeable battery. Since the unit draws a current of up to 25 mA, operation from a dry battery is, however, not recommended. The mains adaptor and rechargeable battery may be used together: diodes D_1 and D_2 prevent any interaction.



Parts list

Resistors:

R_1 – R_4 = 25.5 k Ω , 1%
 R_5 , R_7 = 10 k Ω
 R_6 - resistor array 8 \times 10 k Ω
 P_1 = preset 4.7 k Ω

Capacitors:

C_1 , C_2 = 0.47 μ F, MKT (metallized polyester)
 C_3 – C_6 = 22 pF, ceramic
 C_7 , C_8 , C_{11} , C_{16} = 10 μ F, 10 V, radial
 C_9 = 100 μ F, 16 V, radial
 C_{10} = 10 μ F, 16 V, radial
 C_{12} – C_{15} , C_{17} , C_{18} = 0.1 μ F, ceramic

Inductors:

L_1 , L_2 = 4.7 μ H choke

Semiconductors:

D_1 , D_3 = 1N4001
 D_2 = BAT85

Integrated circuits:

IC_1 = MPXS4100A (Motorola)
 IC_2 = AD7715AN5 (Analog Devices)
 IC_3 = AT89S53-12JC (Atmel)
 IC_4 = AT24C16 (Atmel)
 IC_5 = ADM233L (MAX233) (Analog Devices)
 IC_6 = LP2950CZ5.0

Miscellaneous:

X_1 = quartz crystal 2.4576 MHz
 X_2 = Quartz crystal 11.0592 MHz
 JP_1 = 2-terminal 2.54 mm pin strip and pin jumper (Maplin)
 JP_2 = 3-terminal 2.54 mm pin strip and pin jumper (Maplin)
 JP_2 = 3-terminal 2.54 mm pin strip and pin jumper
 K_1 = 2 \times 10 pin box header
 K_2 = 9-pin female sub-D connector
 K_3 = low-voltage socket for board mounting
 K_4 = 6-way header for board mounting with mating part for cable fitting
 BT_1 = 9 V rechargeable battery with connecting clip
 S_1 = switch, single on contact
 S_2 – S_6 = push-button switch D6-R-RD with cover D6Q-RD-CAP (ITC)
 Enclosure Bopla E435 (120 \times 65 \times 65 mm) (available from Phoenix 01296 398355)
 Liquid-crystal display 2 \times 16 digits
 5 cm of 20-core flatcable
 1 flatcable connector
 Socket for IC_3
 PCB Order no. 980097*
 Software Order no. 986031*
 * See Readers Services towards the end of this issue.

Figure 6. Double-sided printed-circuit board for the barometer/altimeter.

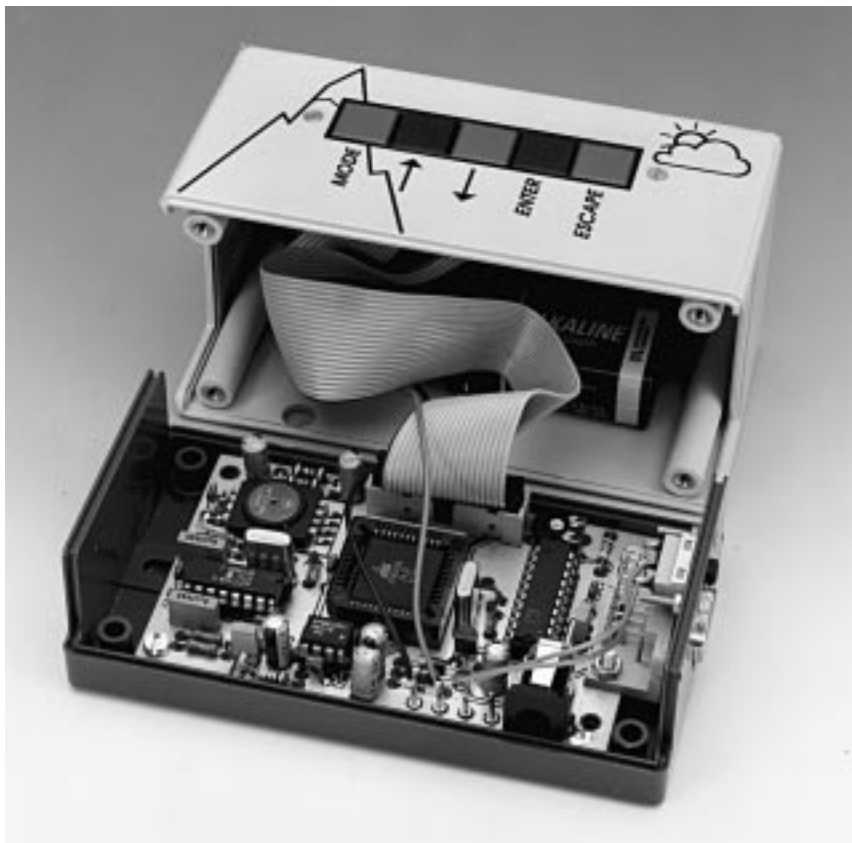
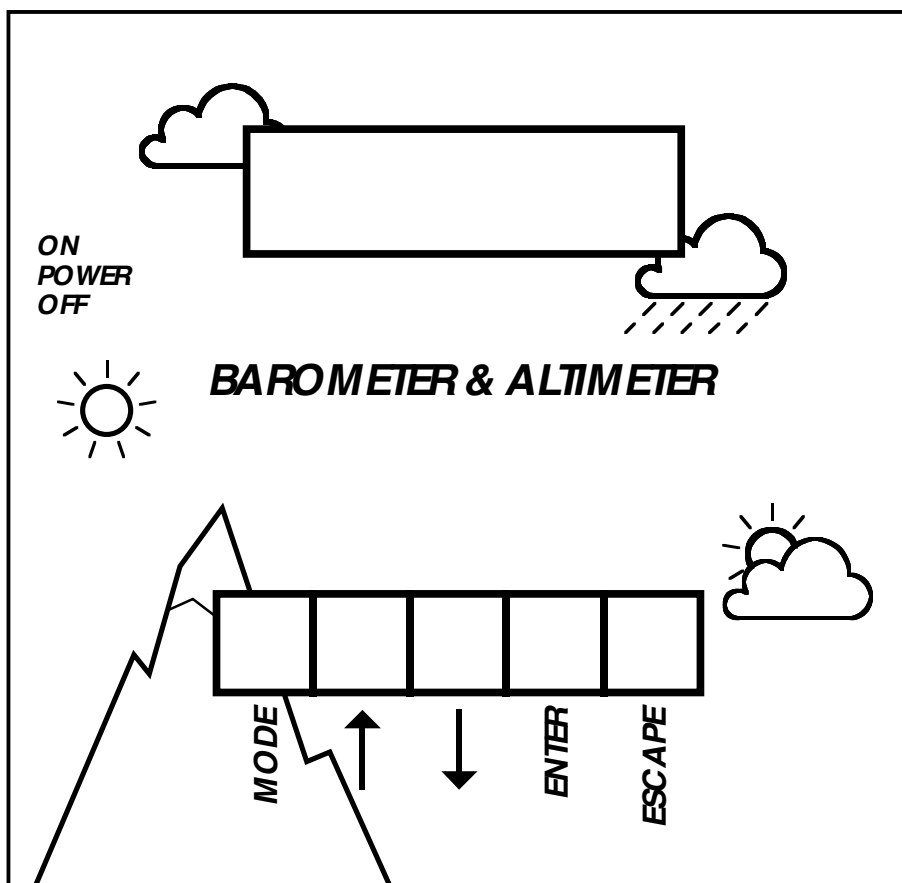


Figure 7. Completed prototype printed-circuit board.

Figure 8. Suggested front panel layout for the barometer/altimeter.



The voltage is stabilized at 5 V by micropower voltage regulator IC₆, which requires a quiescent current of only 75 μ A and creates a voltage drop of not more than 400 mV. The device can handle output currents of up to 100 mA.

Immediately following the regulator, preset P₁ serves to set the contrast voltage, U_o, for the LC display.

The supply voltage to the various digital ICs is taken direct from the regulator output pin, whereas that to the sensor and the reference voltage input of the DAC is decoupled by chokes L₁ and L₂.

CONSTRUCTION

Before any construction work is started, the double-sided printed-circuit board in Figure 6 must be cut into two. The smaller portion houses five push-button switches and connector K₄. When these have been put into place, fix it to the cover of the enclosure after this has been prepared according to the front panel layout in Figure 7.

Owing to the narrowness of the tracks and the dense population of the larger portion, it is imperative to use a small soldering iron and to take even greater care than usual to avoid solder splashes on the board. It hardly needs saying that great care must be taken also to ensure correct positioning of all components, especially polarized ones.

When the board has been completed (it should look more or less as the prototype shown in Figure 8), fit it into the enclosure, and close this with the cover prepared as described earlier.

In next month's Part 2, the programming, calibration, software and operation will be described.

[980097]

electronics on-line

information on CD-R(W)

The article on CD-R formats in this month's Supplement discusses the main formats and features of the CD-R and CD-RW. If you want to know more about this interesting subject, there is, of course, the worldwide web...

If you want to know exactly which characters are allowed in an ISO9660 file name, or the exact contents of the Red Book, or what happens when pits are 'burned' into a CD-ROM, then this information is very likely available somewhere on the Internet. In our search for background information for the above mentioned article we came across a number of web sites with lots of information on this storage medium as well as many associated aspects. On this page, we want to share this information with you.

One of the main forces behind the development of the rewritable CD is the Japanese company Ricoh. Not surprisingly, their Japanese web site explains the basic construction and operation of this medium. You can find it at

http://ext.ricoh.co.jp/cd-r/e-e_europe/shikumi/index.html

For a detailed technical explanation of the CD-R medium and related matters, you should pay a visit to an HP web site at

<http://www.hp.com/isgsupport/cdr/devokit/techover.html>

Although this page starts with some information on HP's own products in this area, this is just a short section, and a very interesting article follows!

The web site of Plasmon, a manufacturer of all sorts of magneto-optical products, is not lacking in useful information on the CD-R medium:

<http://www.plasmon.co.uk/home1.htm>

General information on CD-Rs also abounds on the Internet, particularly with companies offering products

where to find background facts

related (one way or another) to CD-R(W).

For example, Adaptec supplies software for CD-ROM burners — their site offers clear explanations of various CD-R related terms, as well as an extremely useful CD-R glossary:

<http://www.adaptec.com/support/overview/cdrecbackground.html>

The web site run by Audio Images Interactive Group (AIIG) also has a wealth of information, especially on the CD-Extra:

<http://www.bway.net/aiig/>

Another web site you should not forget to visit is that of Octave, an American company specialised in selling CD recorders and ancillaries. Their site allows you to browse the on-line version of Ricoh's extremely interesting 'Recordable Handbook':

<http://www.cd-r.com/library/ricoh/cdrhandbook.html>

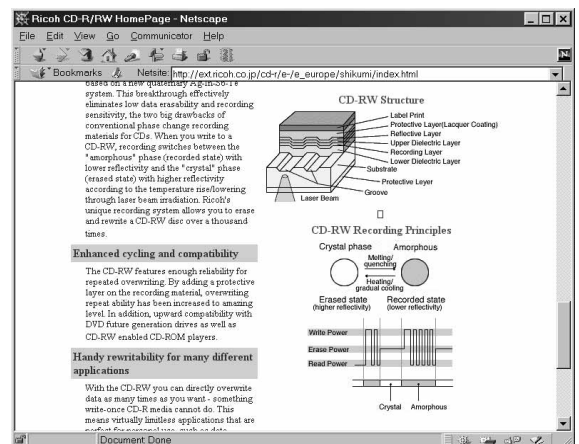
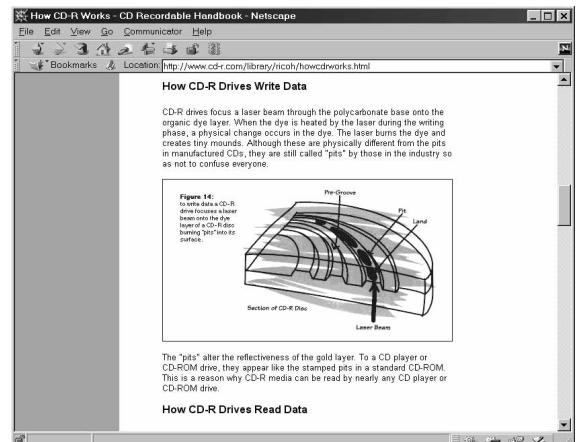
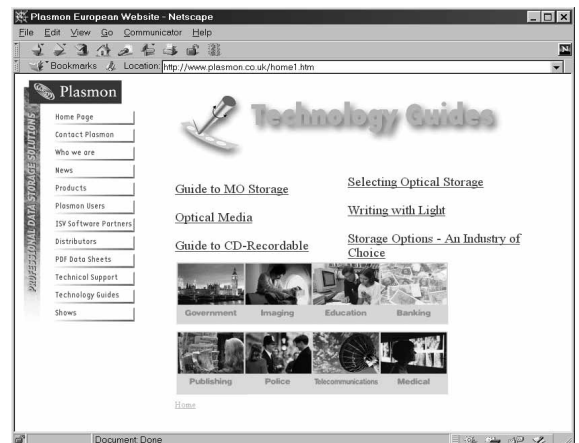
Octave's site contains more interesting stuff, by the way, at:

<http://www.cd-rom/library.html>

Finally, a useful address for those who feel like exploring the intricacies of the ISO9660 format without having to wade through the official documentation:

<http://www.coe.neu.edu/~riker/ISO9660.html>

At this site, Philip J. Erdelsky in his article 'ISO9660 Simplified for DOS/Windows' ventures to explain in plain words the details of this ISO format.



(985081-1)

programmable sensor interface

MLX90308

Owing to its integrated controller, the MLX90308 from Melexis is a very flexible interface that, with suitable software, can be readily adapted for use with a wide range of sensors. Since most other needed components are also integrated on the IC, the device is not only user-friendly but also relatively inexpensive.



Features

- ⇒ Microprocessor-controlled signal conditioning for bridge-type sensors
- ⇒ Suitable for low-cost sensors: reduction of non-linearity by programmable coefficients
- ⇒ External or internal temperature sensor for compensating temperature errors
- ⇒ 64 byte RAM, 4 kbyte program space (ROM) available
- ⇒ 48×8 bit for calibration data available (EEPROM)
- ⇒ Versatile output signal ranges: 4 V, ..., 11 V or 4–20 mA loop
- ⇒ Mass calibration easy with 2400 or 9600 baud UART
- ⇒ Power supply from 5 V to 35 V

DESCRIPTION

The most noteworthy property of the MLX90308 sensor interface is the integrated controller. This enables the device to be used for a very wide range of applications. The non-linear operation of many bridge-type sensors can be largely reduced, eliminated or corrected simply by programmable coefficients.

Temperature errors can be compensated with the aid of an internal or external sensor.

The program memory consists of 4 kbyte ROM. For the interface, there are 64 bytes of RAM, and for the calibration data, there is a 48 × 8 bit EEPROM available.

ROM available.

The output may be used as required in the voltage mode (4–11 V), or current loop (4–20 mA). Since the IC has an internal voltage regulator, the supply voltage may range from 4 V to 35 V.

The MLX90308 also contains a complete oscillator, a power-on reset facility, a watch dog, and a UART (2400 or 9600 baud).

The interface is housed in a 16-pin DIP case. The internal block diagram and pinout are shown in **Figure 1**.

MODES OF OPERATION

There are two major modes of operation: the *digital mode* and the *analogue mode*.

In the analogue mode, the output of the pressure sensor bridge is amplified via IA and GN and then converted to the correct output signal form in one of the output stages.

The sensitivity and offset of the analogue signal chain are defined by numbers passed to the digital-to-analogue converter (DAC) interfaces from

A Melexis Application

Melexis UK
3 Anglesey Close
Chasetown
Staffordshire WS7 8XA
Telephone/fax 01543 670391

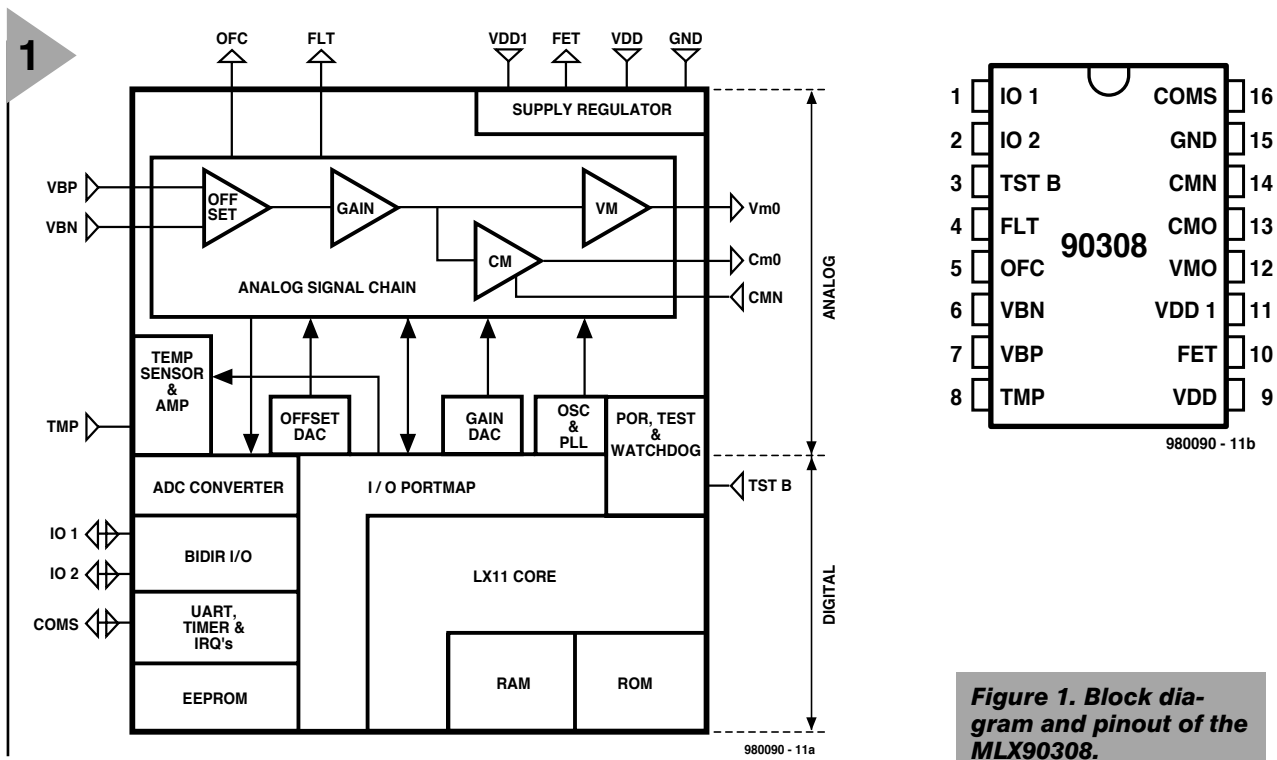


Figure 1. Block diagram and pinout of the MLX90308.

the microcontroller (GN[9:0] and OF[9:0]). The wide range of bridge offset and gain is accommodated by means of a 2-bit 'coarse adjustment' DAC in the offset adjustment (CSOF[1:0]), and a similar one in the gain adjustment (CSGN[2:0]).

The temperature signal, TPO, generated from the external or internal temperature sensor, drives an analogue-to-digital converter (ADC) interface to provide a 10-bit digital representation of temperature for use in calculating the main 10-bit DAC inputs. A similar 2-bit 'coarse adjustment', GNTP[1:0] is also required for the temperature signal gain and offset.

In the digital mode, the analogue sensor signal is converted to digital information before being processed by the microcontroller. It is then converted with the 10-bit DAC and finally directed to the output.

ANALOGUE SECTION

A bandgap-stabilized supply regulator is provided internally; the pass transistor is external. The bridge-type sensor is typically powered by the regulated supply (typically 4.75 V). For ratiometric operation, the supply regulator may be disabled by shorting the unregulated and regulated supply pins (VDD1 and VDD respectively).

The 90308 contains a programmable on-chip RC oscillator. No external components are needed to set the frequency (87.8 kHz ± 1%). The microprocessor clock is generated by a phase-locked loop (PLL) (614 kHz or 2.46 MHz) that locks on the basic oscillator.

A/D and D/A conversions using only one DAC

For saving chip area, the single offset DAC is multiplexed in various ways. Both fine offset and digital mode signals are stored on a capacitor. An ADC loop is available via a 10-bit multiplexer (see also Figure 1).

Digital-to-analogue

Before changing to another capacitor, the DAC output must be settled to the new value. So, MODSEL moves the analogue multiplexer to the so-called 'open-state 0'. At the same time, the 10-bit multiplexer selects OF[9:0] for the offset DAC. After the DAC settling time, the analogue multiplexer is moved to its final state and the DAC output is stored on a capacitor.

Analogue-to-digital

The S/W signal MODSEL connects the SAR output to the DAC and the DAC output to the comparator. The SAR register is initialized by a leading edge of STC (S/W signal). At the end of the A/D conversion, the EOC flag is set to 1 and the controller can read the ADC values.

Power-on reset

The power-on reset initializes the state of the digital part after power up. This reset procedure has two phases—see Figure 3. The reset level is typically 3.5 V, while the delay time is given by:

$$T_{\text{delay}} = 2^6 / f_{\text{RCosc}}$$

Test mode

For 100 per cent testability, a TEST pin

(3) is provided. When this pin is pulled low, the monitor program is entered and the chip changes its functionality. In all applications, this pin should be pulled high or left floating (there is an internal pullup resistor).

MICROPROCESSOR AND DIGITAL SECTION

The Type LX11 microcontroller core is well specified elsewhere (for instance, in Melexis' *Standard Products Book*) and will therefore not be discussed in detail in this short article. It has the following features:

- ⇒ two accumulators, one index, and two interrupt accumulators
- ⇒ 8-bit I/O ports (see below)
- ⇒ wait mode is not implemented
- ⇒ 64 byte RAM
- ⇒ 4 kbyte ROM; 3 kbyte is available for the user's application firmware (changed with one mask only); 1 kbyte is reserved
- ⇒ 48 × 8-bit EEPROM, which is accessible only via the I/O ports (see below)

I/O ports

The 90308 microcontroller has 15 port addresses. Read and write operations to the same address do not access the same port. In other words, there are 15 input ports (only six used) and 15 output ports. See Table 1.

Interrupt controller

The 90308 has four interrupt sources: two UART interrupts and two timers. When these sources are enabled (output port), the LX11 core is sent to

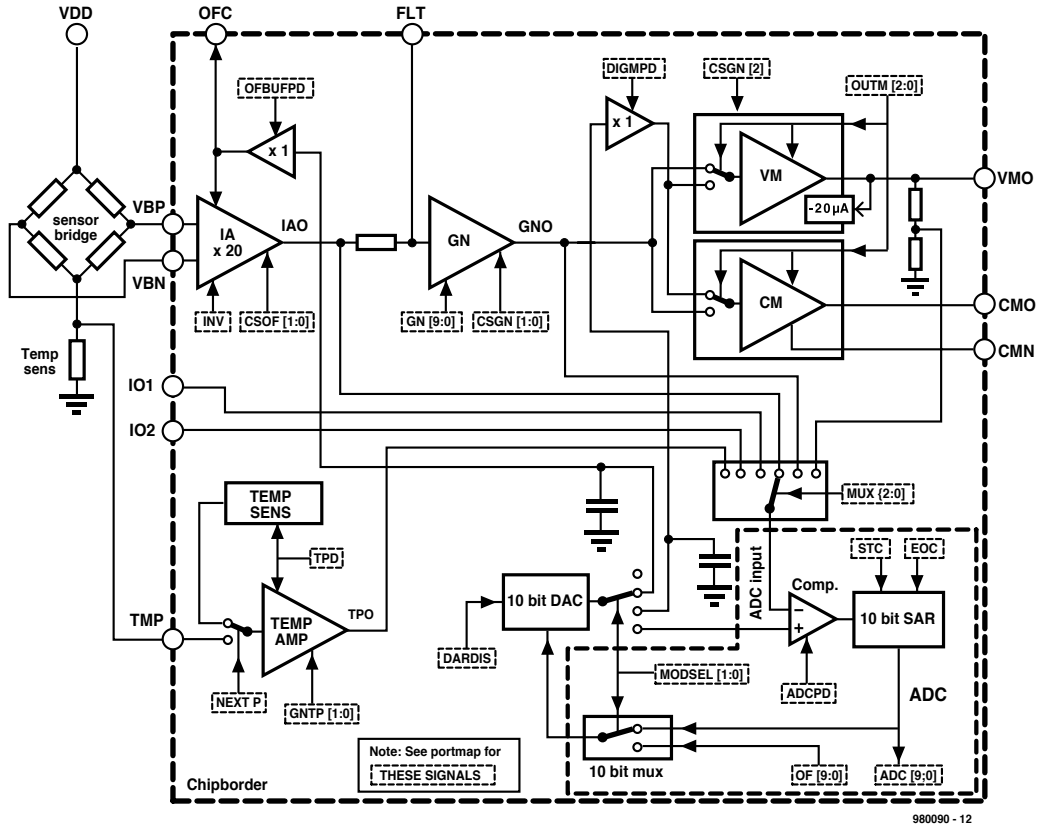


Figure 2. The main signal path in the 90308. The bridge sensor is supplied from the regulated 4.75 V line (VDD).

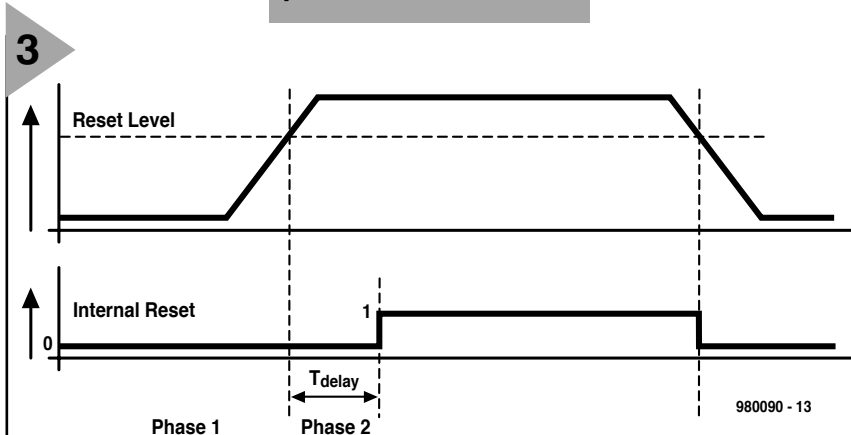
interrupt address 21h upon interrupt occurrence. The interrupt signals are situated in input/output port 2. All interrupts are reset to zero upon a write or read access to port 3. It is up to the user to read the interrupt bits or not.

Note that if the interrupt is not read, the interrupt controller will be in a state being the OR of the previous states.

UART

The serial link is a potentially full-duplex UART. It is receive-buffered, which means that it can receive a second byte before a previously received byte has been read from the receiving register. However, if the

Figure 3. The power-on reset of the 90308 takes place in two phases.



first byte is not read by the time the reception of the second byte is completed, the first byte will be lost. The UART's baud rate depends on the RC oscillator's frequency and the TURBO bit (see Table 2). The transmitted and received data has the following structure: start bit = 0; eight bits of data; stop bit = 1. Sending data:

- Writing a byte to port 1 automatically starts a transmission sequence.
- The TX interrupt is set when the STOP bit of the byte is latched on the serial line.

Receiving data:

- Reception is initialized by a 1 to 0 transition on the serial line (i.e., a START bit).
- The baud rate

period (i.e., the duration of one bit) is divided in 16 phases. The first six and last seven phases of a bit are of no use. The decision on the bit-value is then the result of a majority vote of phases 7, 8 and 9 (i.e., the centre of the bit).

- Spike synchronization is avoided by debouncing on the incoming data and a verification of the START bit value.
- The RX interrupt is set when the STOP bit is latched in the UART.

Timer

The clock of timers TMI and TPI is taken directly from the main oscillator. The timers are never reloaded, so the next interrupt will take place 2^x oscillator pulses after the first interrupt.

Watch dog

An internal watch dog will reset the whole circuit in case of a software crash. When the watch dog counter is not reset at least once every 26 ms (with a main clock frequency of 2.46 MHz), the microcontroller and all the periphery will be reset.

Demonstration material

A complete set of demonstration material is available to bona fide designers upon request. It contains:

- 10 samples 90308 with versatile demonstration firmware
- a demonstration board
- a serial cable for connection to a PC
- special software for communication and calibration

- a demonstration description

With the demo set, a working sensor interface can be built and calibrated without too much effort.

Application specific firmware

For the development of application specific firmware, a special version of the 90308 with an external ROM interface is available. This version also fits in the demonstration board and can be used as such to develop and debug new firmware. The external ROM is then replaced typically by an external EPROM or an emulator of 'Crash Barrier'. A complete software development toolset is also available—see **Figure 4**. It contains:

- ⇒ Complete demonstration set: see above
- ⇒ An assembler: AS
- ⇒ A relocater and linker: LYN
- ⇒ A ROM file generator: OBSEND
- ⇒ A customizable simulator: SIM
- ⇒ An emulator: Crash Barrier

[980090]

4

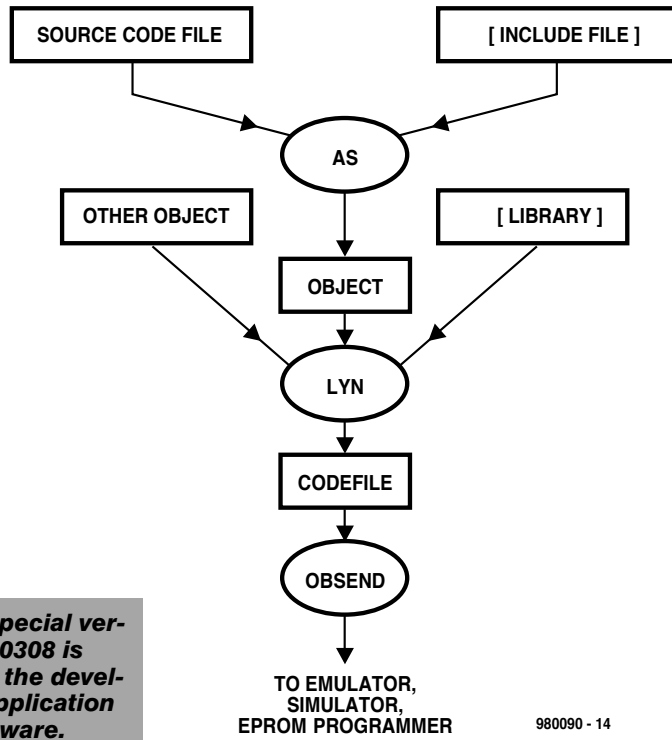


Figure 4. A special version of the 90308 is available for the development of application specific firmware.

980090 - 14

Table 1. Input-ports

	7	6	5	4	3	2	1	0	bitnumbers
port	msb	msb-1	msb-2	msb-3	msb-4	msb-5	msb-6	lsb	portname
0									
1	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	UART receive
2					PTPI	PTMI	PUTI	PURI	interrupt register
3				IRQ1	IRQ0	EOC	TESTB	WCB	input register
4	EEI7	EEI6	EEI5	EEI4	EEI3	EEI2	EEI1	EEI0	EEPROM read
5									
6									
7	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC input 1
8			IBP1	IBP0			ADC1	ADC0	ADC input 2

Table 3. Absolute maximum ratings

VDD1 (regulator active): 35 V
 VDD (regulator disabled): 7 V
 Die temperature: +160 °C
 Storage temperature range: -55 °C to +150 °C
 Working temperature range: -40 °C to +140 °C
 Supply current (at 25 °C ambient):
 current mode: 2.1 mA
 voltage mode: 5 mA

Table 2. Output-ports

	7	6	5	4	3	2	1	0	bitnumber
port	msb	msb-1	msb-2	msb-3	msb-4	msb-5	msb-6	lsb	portname
0	DIGMPD	OUTM1	OUTM0	WDTSTA	ETPI	ETMI	EUTI	EURI	UARTMR control
1	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	UART transmit
2					ATPI	ATMI	AUTI	AURI	interrupt register
3	AWD	MUX2	MUX1	MUX0	STC	ADCPD	MODSEL1	MODSEL0	output register
4	EEO7	EEO6	EEO5	EEO4	EEO3	EEO2	EEO1	EEO0	EEPROM write
5			EEA5	EEA4	EEA3	EEA2	EEA1	EEA0	EEPROM address
6		VEE1	VEE0	EBW	EBE	EWR	ERD	EER	EEPROM control
7	CADJ7B	CADJ6	CADJ5	CADJ4	CADJ3	CADJ2	CADJ1	CADJ0	clkadj register
8	OFBUFPD	DARDIS	OBP1	OBP0	BPDSEL1	BPDSEL0	BPASEL1	BPASEL0	output II register
9	GN9	GN8	GN7	GN6	GN5	GN4	GN3	GN2	DACGN I register
10							GN1	GN0	DACGN II register
11	OF9	OF8	OF7	OF6	OF5	OF4	OF3	OF2	DACOF I register
12							OF1	OF0	DACOF II register
13	IINV	GNTP1	GNTP0	CSOF1	CSOF0	CSGN2	CSGN1	CSGN0	output III register
14					TURBO	OUTM2	INEXTP	TPD	output IV register

PLC87 (A) Instruction Set

Reference Card - 1

Syntax	Parameter(x,y)	BCR	AC1	AC2	Meaning	Function
A I xy	00-17	>, =	-	-	AND input xy	BOR = BOR + Ixy
A Q xy	00-17	>, =	-	-	AND output xy	BOR = BOR + Qxy
A F xy	00-15,7	>, =	-	-	AND flag xy	BOR = BOR + Fxy
AK					AND bracket open	BOR = BOR + (.....)
A T x	0-15	>, =	-	-	AND timer x	BOR = BOR + Tx
A C x	7	>, =	-	-	AND counter x	BOR = BOR + Cx
A S xy	14	>, =	-	-	AND flasher	BOR = BOR + S14
O I xy	00-17	>, =	-	-	OR input xy	BOR = BOR + Ixy
O Q xy	00-17	>, =	-	-	OR output xy	BOR = BOR + Qxy
O F xy	00-15,7	>, =	-	-	OR flag xy	BOR = BOR + Fxy
Q					OR bracket open	BOR = BOR + (.....)
O T x	0-15	>, =	-	-	OR timer x	BOR = BOR + Tx
O C x	7	>, =	-	-	OR counter x	BOR = BOR + Cx
O S xy	14	>, =	-	-	OR flasher	BOR = BOR + S14
ON I xy	00-17	>, =	-	-	OR NOT input xy	BOR = BOR + /Ixy
ON Q xy	00-17	>, =	-	-	OR NOT output xy	BOR = BOR + /Qxy
ON F xy	00-15,7	>, =	-	-	OR NOT flag xy	BOR = BOR + /Fxy

PLC87 (A) Instruction Set

Reference Card - 3

Syntax	Parameter(x,y)	BCR	AC1	AC2	Meaning	Function
TI					transfer indirect FB	FB[AC1] = AC2
T FB					transfer into Fword x	FBx = AC1, FBx+ 1 = AC2
T FV x	0-14	-	>	>	transfer into poll-leg	PR = AC1
T PR		-	>	-	complement accuum.	AC1 = 1st complement of AC1
ORL		-	=	-	BOD-adjust accuum.	Adjust AC1 to BOD-numbers
BOD		-	=	-	adjust accuum.	AD22/100 temp. correction to AC1
ADU		-	=	-	transfer accuum.	AC2 = AC1, AC1 = AC2
TAK		-	=	=	AND byte	AC2 = AC1, AC1 = AC1 and AC2
AB		-	=	=	OR byte	AC2 = AC1, AC1 = AC1 or AC2
OB		-	=	=	XOR byte	AC2 = AC1, AC1 = AC2 xor AC1
XB		-	=	=	shift-left accuum.	shift AC1 left, x bits
SXB x	1-7	-	=	-	shift-right accuum.	shift AC1 right, x bits
SFB x	1-7	-	=	-	increment accuum.	increment AC1, x times
INC x	1-255	-	=	-	decrement accuum.	decrement AC1, x times
DEC x	1-255	-	=	-	equal	AC2 = AC1, BOR= 1 if AC1 = AC2
I= B		=	=	-	greater than	AC2 = AC1, BOR= 1 if AC1 > AC2
> B		=	=	-	smaller than	AC2 = AC1, BOR= 1 if AC1 < AC2
< B		=	=	-	greater than or equal	AC2 = AC1, BOR= 1 if AC1 >= AC2
>= B		=	=	-	smaller than or equal	AC2 = AC1, BOR= 1 if AC1 <= AC2
<= B		=	=	-	not equal	AC2 = AC1, BOR= 1 if AC1 > < AC2
> < B		=	=	-	add	AC1 = AC2+ AC1, AC2= overflow
+ B		=	=	-	subtract	AC1 = AC2- AC1, AC2= borrow
- B		=	=	-	multiply	AC1 = AC2* AC1, AC2= overflow
* B		=	=	-	divide	AC1 = AC2/AC1, AC2= rest
/ B		-	=	-	jump	jump to label M
JU		-	-	-	conditional jump	jump to label M if BOR= 1
JC		>, ?	-	-	absolute subr. call	continue w. subroutine at label M
UPA		-	-	-	conditional subr. call	continue w. subroutine at label M if BOR= 1
UPC		>, ?	-	-	return	return to main program
PET						



Syntax	Parameter(x,y)	BCR	AC1	AC2	Meaning	Function
ON(OR NOT bracket open	BCR= BCR+ / (...)
ON T x	0-15	>,=	-	-	OR NOT timer x	BCR= BCR+ / T x
ON C x	7	>,=	-	-	OR NOT counter X	BCR= BCR+ / O x
ON S xy	1,4	>,=	-	-	OR NOT flasher	BCR= BCR+ / S 1,4
)					bracket close	close bracket-level
S Q xy	0.0 - 1.7	>,?	-	-	set output	Ox=y= 1 if BCR= 1
S F xy	0.0 - 15.7	>,?	-	-	set flag	Fx=y= 1 if BCR= 1
S C x	0-7	>,?	>	-	set counter contents	Ox= AC1 if BCR= 1
R Q xy	0.0 - 1.7	>,?	-	-	reset output	Ox=y= 0 if BCR= 1
R F xy	0.0 - 15.7	>,?	-	-	reset flag	Fx=y= 0 if BCR= 1
R C x	0-7	>,?	-	-	reset counter contents	Ox= 0 if BCR= 1
= Q xy	0.0 - 1.7	>	-	-	output = BCR	Ox=y= BCR
= F xy	0.0 - 15.7	>	-	-	flag = BCR	Fx=y= BCR
L IB x	0-1	-	=	=	load input byte x	AC2 = AC1, AC1 = IBx
L OB x	1	-	=	=	load output byte x	AC2 = AC1, AC1 = OBx
L FB x	15	-	=	=	load flag byte x	AC2 = AC1, AC1 = FBx
LJ IB		-	>,=	=	load indirect, I-byte	AC2 = AC1, AC1 = IJ[AC1]
LJ OB		-	>,=	=	load indirect, O-byte	AC2 = AC1, AC1 = OJ[AC1]
LJ FB		-	>,=	=	load indirect, F-byte	AC2 = AC1, AC1 = FJ[AC1]
L FW x	0-14	-	=	=	load flagword x	AC1 = FBx, AC2 = FBx+ 1
L C x	0-7	-	=	=	load content counter x	AC2 = AC1, AC1 = Zx
L AI x	0-5	-	=	=	load analogue input x	AC2 = AC1, AC1 = AIx
L KB x	0-255	-	=	=	load constant, byte	AC2 = AC1, AC1 = x
L KW x	0-65535	-	=	=	load constant, word	AC2 = x[HB/je], AC1 = x[LB/je]
L KT xy	0-255,0-3	-	=	=	load constant, timer	AC2 = y, AC1 = x
L KC x	0-255	-	=	=	load constant, counter	AC2 = ~, AC1 = x
T CB x	0-1	-	>	-	transfer into Q-byte x	CBx = AC1
T FB x	0-15	-	>	-	transfer into F-byte x	FBx = AC1
TI CB		-	>	>	transfer indirect CB	CB[AC1] = AC2



Syntax	Parameter(x,y)	BCR	AC1	AC2	Meaning	Function
CJ C x	0-7	>,?	-	-	count up	Cx = Cx+ 1 on pos. edge of BCR
CD C x	0-7	>,?	-	-	count down	Cx = Cx- 1 on pos. edge of BCR
SD T x	0-15	>,?	-	-	start Tx, on-delayed	start Tx as on-delayed Timer if BCR= 1
SF T x	0-15	>,?	-	-	start Tx, off-delayed	start Tx as off-delayed Timer if BCR= 1
SP T x	0-15	>,?	-	-	start Tx as pulse	start Tx as pulse if BCR= 1
SE T x	0-15	>,?	-	-	start length, pulse	start Tx as lengthened pulse if BCR= 1
PRIT IB x	0-1	-	-	-	print I-byte	IBx output to display
PRIT CB x	0-1	-	-	-	print C-byte	CBx output to display
PRIT FB x	0-15	-	-	-	print F-byte	FBx output to display
PRIT "..."	max 20 chars	-	-	-	print string	text output to display
POS x	0-127	-	-	-	pos. cursor	set cursor position (2nd row = 64)
TMP		-	>	-	print temperature	temperature output of AC1 to display
CLR		-	-	-	clear	clear display
TRG		>,?	>	-	trigger	write AC1 into PDS, INQ^, PDS
STP		>,?	-	-	stop	stop PLC if BCR= 1 at program end
STS		>,?	-	-	stop immediate	stop PLC immediately if BCR= 1
BE		-	-	-	subroutine follows	separates main program from subroutines
PE		-	-	-	end of program	marks the end of program

The following instructions are only executable with a PLC87A-L AI x, ADJ and TMP.

Abbreviations/symbols:

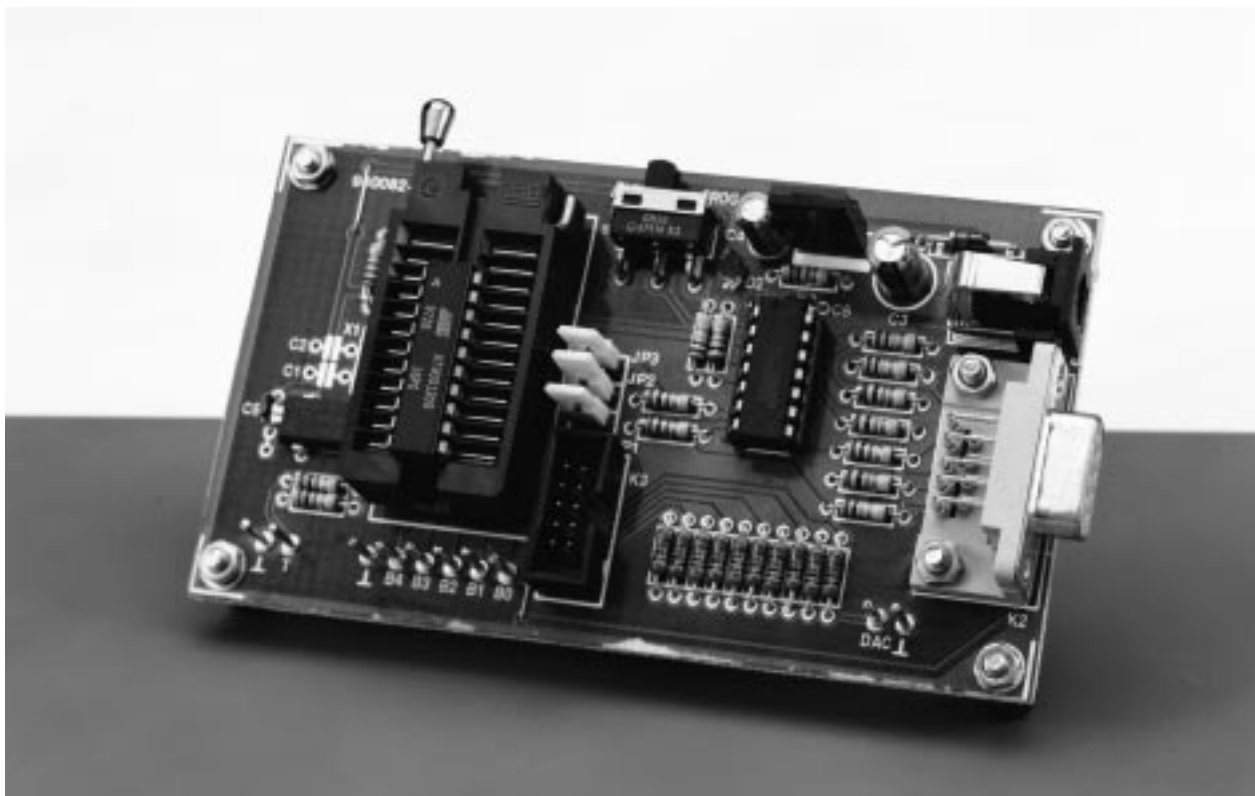
- x,y = number of variable(s) or value
- AC1 = accumulator 1
- AC2 = accumulator 2
- BCR = bit-combination result
- > = operation reads register
- ? = operation depends on register status
- = = operation writes to register
- = no charge to register

A small experimental board for the Atmel AT90S1200 8-bit RISC processor was already presented in the October issue, together with the necessary application data. In this, the second and last part of the article, we present sample programs, including a frequency divider, a numerically-controlled oscillator (NCO) and programs for measuring time and frequency.

by Dr M. Ohsmann

AVR-RISC evaluation system (2)

experiment with
Atmel's latest RISC processors



Due to its high clock frequency (up to 16 MHz) and its RISC structure, which allows it to execute most instructions within one clock cycle, the very inexpensive Atmel AVR controller can take on applications that would normally require the use of more costly programmable logic devices. The easiest way to learn how to program such

applications is to study easily implemented sample programs. The program for the first application example, a simple signal-pattern generator, was included in the first part of this article. The step from a this simple pattern generator to a frequency divider is not very great. You can construct simple 'wait loops' for relatively large divider

ratios. For example, the 12 MHz clock must be divided by 625 to produce the well-known serial bit rate of 19200 pulses per second. A divider based on a microcontroller has the great advantage that it is often more flexible than a classical design. The program listing shown in **Figure 6** (SCDIV625.ASM), for example, not only divides by 625, but

also generates non-overlapping pulses as illustrated in **Figure 7**. By varying the parameters n1 and n2, you can adjust the divider ratio and the duty cycle of the outputs over a wide range.

Fancy frequencies: the NCO

With a frequency divider, it is only possible to simply divide a clock rate (as the name suggests). But what can you do if you need to derive a frequency of 77.5 kHz from a 12 MHz clock? (Let's assume that the derived frequency need not be exactly 77.5 kHz, so that for example an error of 0.005 Hz is allowed – this represents an error of less than 1 ppm!)

Modern clock synthesizers employ DDS (direct digital synthesis) oscillators for this purpose (see **Figure 8**). A DDS oscillator consists of the actual numerically controlled oscillator (NCO), a sine look-up table and a D/A converter. As can be seen from the figure, the NCO itself is built using a relatively wide adder (often 32 bits). This function can be emulated by the microcontroller, using the program listing shown in **Figure 9**. One cycle of the DDS oscillator lasts for eight controller clock cycles. With a clock rate of 12 MHz, the clock rate of the NCO is thus $12 \div 8 = 1.5$ MHz. The width of the NCO is 29 bits. To generate a clock with a frequency of 77,500 Hz, the value of the repeatedly added constant term must be chosen as follows:

$$(77500 \text{ Hz}) \cdot (2^{29}) \div (1.5 \text{ MHz}) = 554766627738330 = 1A740DA \text{ (hex)}.$$

This is exactly what the sample program does, with the result that a 77.5 kHz clock appears at the Port B, bit 4 output (pin 16). The error is less than $(1.5 \text{ MHz}) \div (2^{29}) = 0.00279... \text{ Hz}$ – which is quite negligible. A simple D/A converter as shown in **Figure 10** also allows a sawtooth waveform to be generated from the NCO.

You can see that a microcontroller that costs only a few pounds can be used to generate frequencies up to 100 kHz with an error of 0.003 Hz or less. A version of the sample program that allows you to select the value of the frequency from the PC via the RS232 interface is included on the diskette (NCO2.ASM).

There are many varied applications for such an NCO. It can for example be used in a PLL circuit as a 'non-integer' divider that reduces the controlled frequency to the rate of the reference

frequency (see **Figure 11a**), or it can be used directly as a reference oscillator with very fine adjustment capability (see **Figure 11b**).

Time measurements

The next application is measuring the duration of the HI interval of a logic signal. This is accomplished by first waiting in a program loop (labeled waitL) until the signal goes LO and then waiting again until it goes HI (in the loop labeled waitH). While the signal is HI, the program stays in a counter loop (labeled m1) that keeps track of how many times the loop has been executed and senses whether the signal is still HI. As soon as it goes LO, the loop is exited and the result is output (see the program listing in **Figure 12**).

The inner counting loop consists of 6 cycles and thus takes $0.5 \mu\text{s}$ with a 12 MHz clock. This means that the pulse duration can be measured with a resolution of $0.5 \mu\text{s}$. The loop counter is 3 bytes wide (24 bits), so it can count up to $2^{24} = 16,777,216$. Since $2^{24} \cdot (0.5 \mu\text{s}) = 8 \text{ s}$, this short program can measure times from 1 μs to 8 s with a resolution of $0.5 \mu\text{s}$. If you need to measure longer intervals, you only have to increase the width of the counter. This will slightly reduce the resolution, since the loop will take longer to execute, but that usually does not matter with long time intervals. The program XTIMES1.ASM on the diskette includes both the actual measurement loop shown in **Figure 12** and a function to output the result in normal decimal form via the RS232 interface, so that

Listing figure 6

```

.equ    n1 =2
.equ    n2 =203

ldi    r16,$ff ; set output B to totem pole
out    DDRB,r16

LOOP:  cbi    PORTB,0 ; [2]      first bit0 goes LOW
        sbi    PORTB,1 ; [2]      then bit1 goes HIGH
        ldi    r16,n1 ; [1]      now wait a little
wt 1:  dec    r16
        br ne wt 1 ; [3*n1-1]
        cbi    PORTB,1 ; [2]      bit 1 goes LOW
        sbi    PORTB,0 ; [2]      bit 0 goes HIGH
        ldi    r16,n2 ; [1]      again wait a little
wt 2:  dec    r16
        br ne wt 2 ; [3*n2-1]
        rjmp  LOOP ; [2]      total : 10+3*(n1+n2)

```

Figure 6. Listing of the program for dividing by 625.

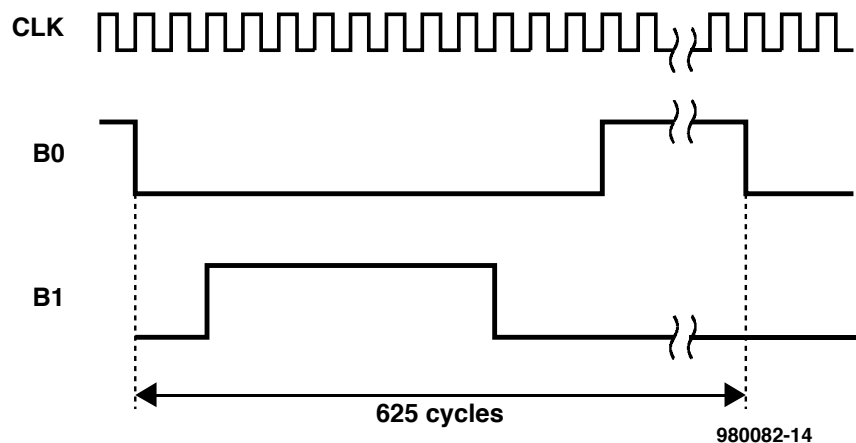
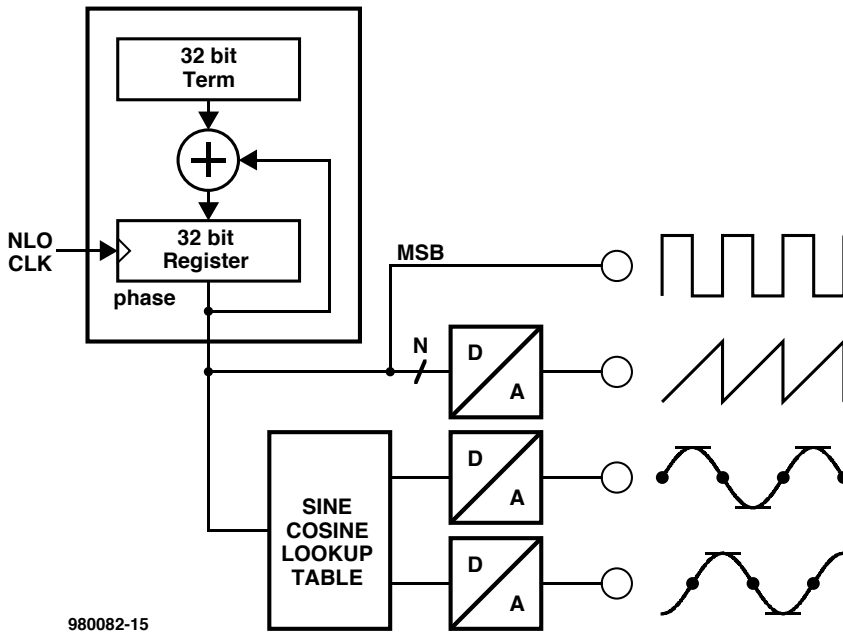


Figure 7. Non-overlapping pulses, which are also generated by the program listed in **Figure 6**.



980082-15

Figure 8. A NCO is fairly simple (at least in block-diagram form)...

you can build a precision time-interval measurement device with serial data output using only a single IC. **Figure**

13 shows the complete schematic diagram. Thanks to its built-in logic, the AT90S1200 does not even need any

Listing figure 9

```

.def    d0    =r 16
.def    d1    =r 17
.def    d2    =r 18
.def    d3    =r 19

.def    nco0  =r 25
.def    nco1  =r 26
.def    nco2  =r 27
.def    nco3  =r 28

RESET:  l di   r 16, $ff    ; set output to tot emp ole
        out   DDRB, r 16

        l di   d0, $DA     ; 01A740DA gives 77.5 kHz at 12
                           MHz XTAL
        l di   d1, $40
        l di   d2, $A7
        l di   d3, $01

LOOP:   add   nco0, d0     ; 29 bit nco summation
        adc   nco1, d1
        adc   nco2, d2
        adc   nco3, d3

        andi  nco3, $01F   ; [1] 5 MSB bits go out
        out   PORTB, nco3 ; [1]
        rjmp  LOOP        ; [2] total 8 cycles

```

Figure 9. ... and the program for the NCO function is just as simple.

Program assembly and device programming

The assembler is started by typing:
a < file name >

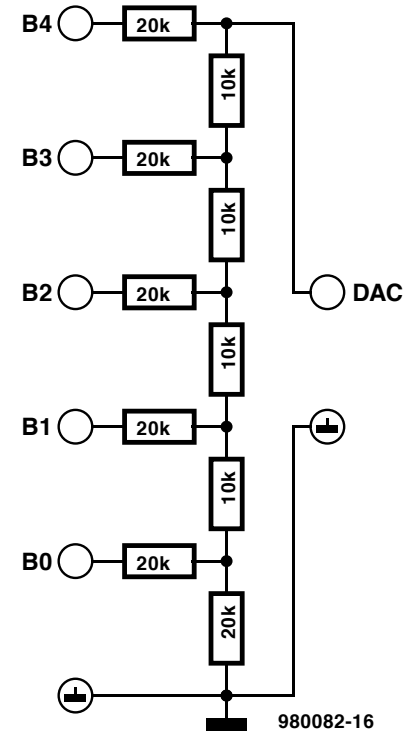
To program the IC, enter 'p' together with a second letter, as follows:
z to erase
p to program
v to verify

This applies equally well to the EEPROM (if used).
The batch files are:
a.bat (assembler)
p.bat (programmer)

The second of these must be modified if a non-standard COM port is to be used (COM2 is the standard setting).

A COM port other than COM2 can also be used with the Pascal programs; the desired port is selected by typing:
progname-COMx

With the program V24COM.EXE, the baud rate is set to a standard value of 9600.



980082-16

Figure 10. The circuitry needed to build a discrete D/A converter for the microcontroller.

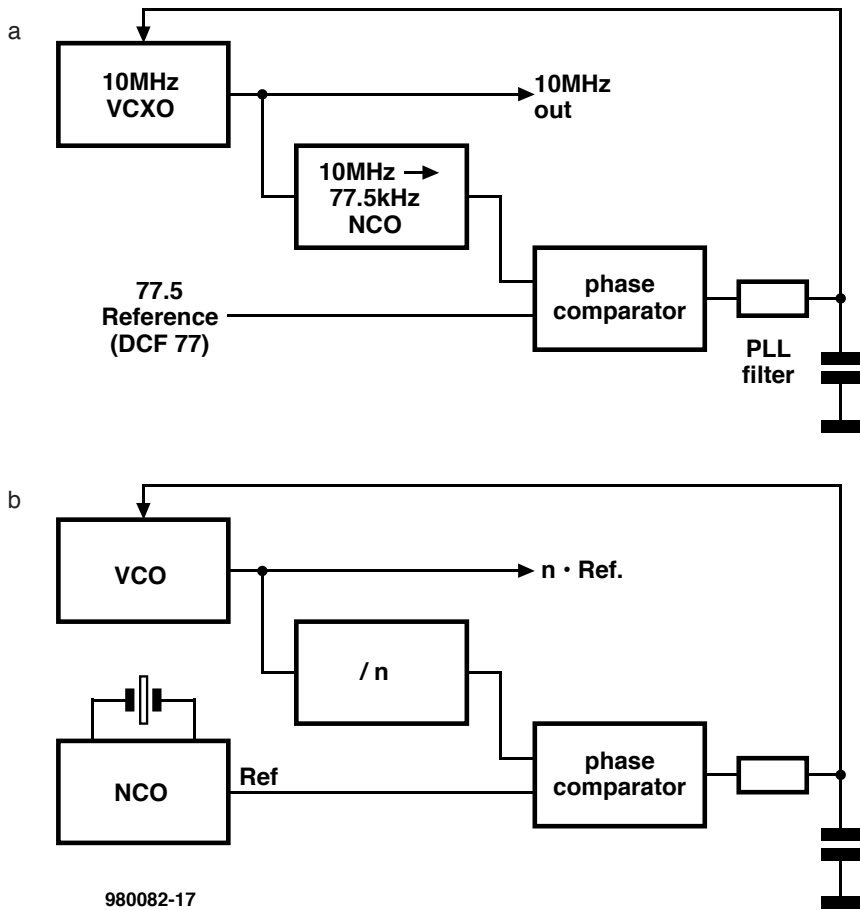


Figure 11. Using an NCO as a divider in a PLL (a) and as a reference frequency source (b).

external circuitry to generate the reset signal. Truly, it would be difficult to construct a simpler device for measuring time intervals, and when you don't need it any more you can 'recycle' it as a frequency meter, as illustrated in

the next example.

Frequency measurements

The AT90S1200 has an internal 8-bit counter that (with suitable program-

ming) can be clocked via Port D bit 4 (pin 8). The external clock signal is sampled by the processor clock, so that the maximum clock rate for the counter is half the crystal frequency – which is still more than 5 MHz. The diskette includes a program (XFRES1.ASM) that uses this counter to implement a frequency counter. A program loop that executes in exactly 12 million cycles (1 second) is used for this purpose. The internal counter runs for the duration of this loop, and a 16-bit software counter is incremented each time the internal counter overflows, so that a 24-bit result (8 + 16) is available at the end of the 1-second interval. This result is subsequently output via the serial interface in decimal form. In this way, the circuit shown in the schematic of Figure 13 can be used to construct a simple frequency counter with serial data output and a resolution of 1 Hz for frequencies up to 5 MHz.

Further possibilities

Naturally, it is possible to do a lot more with the AT90S1200, in particular with its interrupt facilities and the built-in EEPROM, but that is outside the scope of this article. The AT90S1200 also has a big brother, called the 90S1200, which unfortunately can be used at clock rates up to only 8 MHz (instead of the originally announced 20 MHz). The 90S1200 has an 8 kB program memory, a 512-byte EEPROM and an additional 256 bytes of on-chip static RAM, plus an UART. Now, I wonder what we can do with this processor...

(982082-2)

Listing figure 12

```

measure:
    clr    sum0
    clr    sum1
    clr    sum2
waitL:   sbic   PIND, 4    ; wait until we are in a LOW
state
    rjmp  waitL
waitH:   sbis   PIND, 4    ; wait until line goes high
    rjmp  waitH
ml:      add    sum0, one   ; [ 1 ]    one loop= 6 cycles
          adc    sum1, zero  ; count 24 bits wide
          adc    sum2, zero
          sbic   PIND, 4    ; [ 1/2 ]
          rjmp  ml         ; [ 2 ]    jump back if line still
                                HIGH
; end of time measurement
  
```

Figure 12. Listing of the time-interval measurement program.

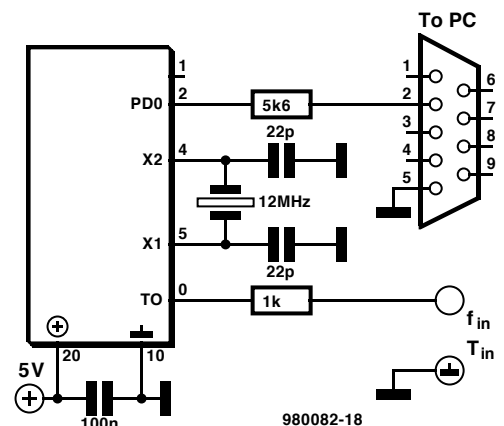


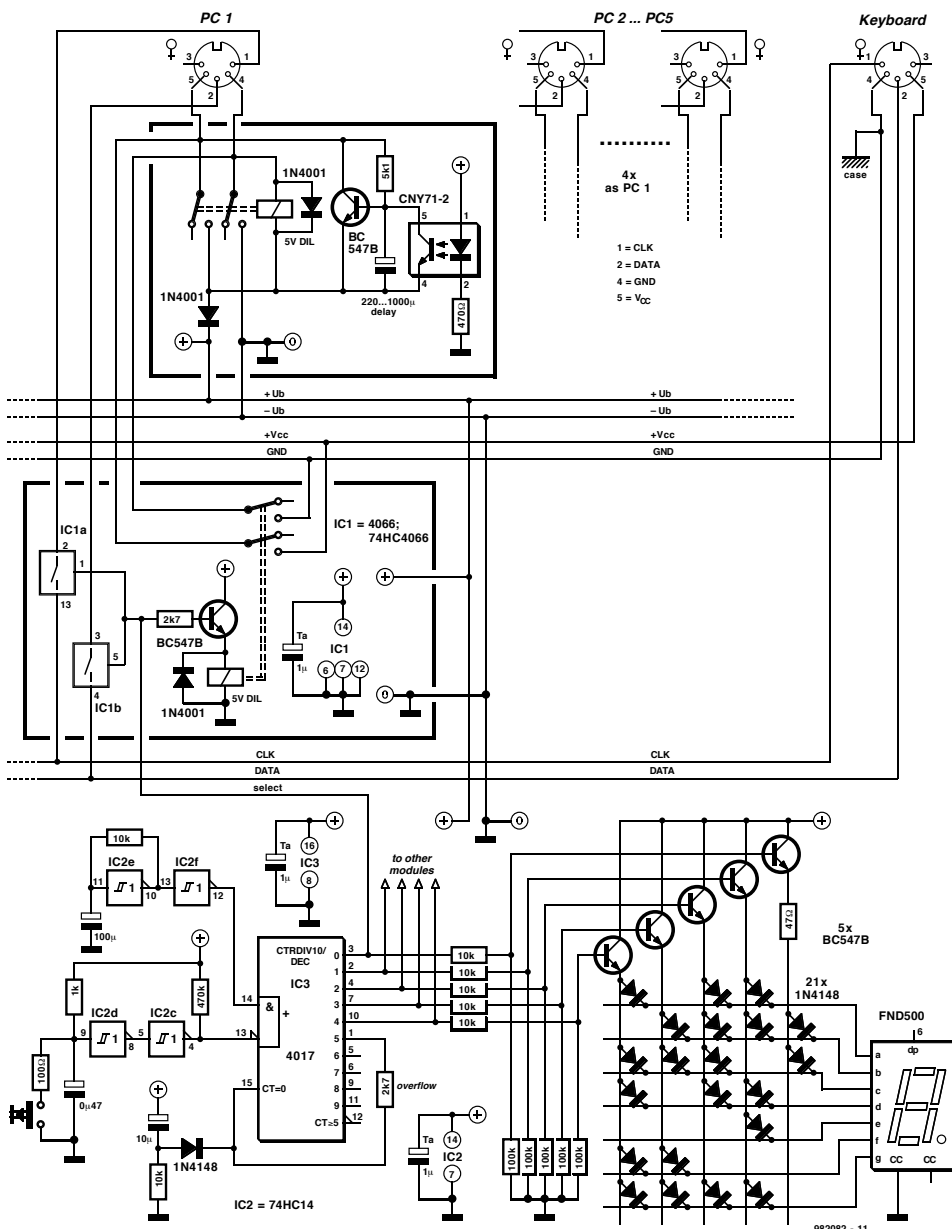
Figure 13. Complete schematic diagram of a precision time-interval measurement device (or a frequency counter) with serial data output. The IC is naturally an AT90S1200.

If you employ several networked computers, you probably don't have enough desk space, since most of the system components are duplicated or triplicated. The circuit presented here deals with the main culprit by reducing the number of keyboards to only one.

Design by Thomas Bandenstein

keyboard switch

many computers, one keyboard



There are a number of possible ways to operate several networked computers using only one keyboard. The least expensive is no doubt a rotary switch, but this is not all that robust mechanically and is also not easily extended. On the other hand, you could buy one of the professional switchover boxes for keyboards, mice and monitors. Although these are practically maintenance-free (since they work electronically) and can be modularly extended, they will put you out of pocket for at least £100 per connected PC (!).

Pros and cons

A nearly ideal solution to this problem is a specially designed unit that you can build yourself. Although the design is modular, you can fit it out to meet your own particular needs. By dispensing with unnecessary features that are found in expensive commercial units, the keyboard switch can be made both inexpensive and reliable. A quick glance at the schematic (Figure 1) shows that it does not require any exotic or unusually expensive components – in fact, you may already have most

Figure 1: The schematic diagram of the keyboard switch, showing one full set of the circuits to be installed in each PC. The circuit in the lower outlined box is required for each PC connected to the switch; that in the upper outlined box is needed for each PC only if PC1 is not always switched on.

of them on hand. The circuit can be easily constructed on a Eurocard and can thus be fitted in the enclosure of a standard mechanical changeover switch (expensive non-mechanical units are usually even larger).

First come, first served!

The circuit of the keyboard switch is designed to be modular, as can be seen from **Figure 1**. It is suitable for connecting a single keyboard to up to ten PCs.

The most difficult task is, surprisingly enough, handling the supply voltage. This is because the individual, possibly unequal voltages of the various PC power supplies must not be allowed to come into conflict. For this reason, the keyboard connection of each PC is fitted with a circuit (shown within the upper outlined box) consisting of an PC pair, a switching transistor, a two-pole reed relay and an optocoupler. When PC1 is first switched on, the emitter of the transistor is connected to earth via the coil of the relay. The electrolytic capacitor is charged via the 5k1 resistor until the base voltage is high enough to cause the transistor to

conduct. This activates the relay, which connects the supply voltage to the +Ub and -Ub lines. This situation will be maintained, even after the optocoupler shorts out the capacitor. You can guess the purpose of the optocoupler: its counterparts in all of the other PCs prevent the associated relays from being activated. The supply voltage is thus always provided by the PC with the shortest PC time constant. If PC1 is switched off, its optocoupler is blocked and the next-shortest time constant determines which PC takes over the role of power source. If PC1 is always on in a given network, the outlined circuit can simply be omitted and the ±Ub lines can be permanently connected. In this case, the connections to ±Ub should be omitted in all of the other PCs.

Selection and control

The switch circuit now has power, but not the keyboard, which of course must be supplied with power from the PC to which it will deliver clock and data signals.

Switching the keyboard between the various PCs is handled by a CMOS

decade counter (4017). When power is first applied, the network connected to the reset input ($CT=0$) forces the Q0 output to be initially active. An oscillator is connected to the clock input of the IC (its frequency is just under 1 Hz with a 100-μF capacitor and a 10-kΩ resistor). If the pushbutton switch (with its 'debouncing' circuit) is pressed, the clock input is enabled and the IC counts up at a rate of one count per second.

The outputs of the counter IC control two things: a seven-segment display that indicates the number of the PC that is connected to the keyboard, and (via transistors) a group of (reed) relays (one for each PC). The two contacts of the selected relay connect the +Vcc and earth lines from the selected PC to the appropriate pins of the keyboard connector. At the same time, the active counter output enables two CMOS switches, which connect the clock and data lines of the keyboard with those of the selected PC. The circuitry shown in the lower outlined box must be fitted in each of the PCs.

(982082-1)

In spite of the slow but steady inroads being made by DVD, the CD-ROM is by far the most popular computer data storage medium. Due to the decreasing prices of both CD-R disks and CD-R writers, more and more computer users are 'burning' their own CDs. Modern CD writers offer a wealth of options, so it makes sense to describe the most important formats and specifications for CD-R and CD-RW.

CD-R formats

all about ISO, Romeo, Joliet and UDF



When Philips and Sony laid down the CD standard in 1980, they probably had no idea that the CD would become so popular as a computer data storage medium. Nearly every PC these days has a built-in CD-ROM drive, and most software is supplied on CD as a matter of course. CD-ROM

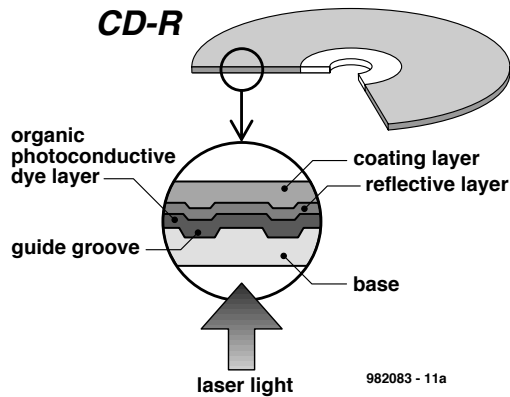
players have become dirt cheap as a result of a rapid series of technical innovations and new models developed for the computer market. Even the 'CD-burners', which allow the user to make his or her own audio and data CDs, have recently dropped significantly in price, so that acquiring a CD

writer has become increasingly attractive for the normal computer user. After you have purchased and installed a CD writer, you will find that it offers an especially large variety of options for making various types of CDs. Multi-session, CD-Extra and Joliet are only a few of the terms that immediately come to mind. We have described all of the options and formats together here, in the hope of helping many (aspiring) owners of CD writers to be able to distinguish the forest from the trees.

From bits to pits

First let's step back for a moment and look at how a CD is made. A standard CD consists of a relatively thick polycarbonate disk that contains *tracks* with *pits*. A thin, reflective aluminium coating is applied on top of this, and a protective lacquer layer is applied on top of the aluminium. The laser beam that shines on the CD when it is installed in a CD player is focussed through the polycarbonate layer onto the aluminium layer. A pit changes the amount of light that is reflected, and this change is detected by a receiver in the laser assembly. In this way, ones and zeros can be read from the CD. A recordable CD (CD-R) uses an entirely different construction (see **Figure 1a**). The basic CD again consists of a polycarbonate layer, but a complete track is moulded into this layer in the form of a long spiral that runs from the hub to the rim of the disk

a



b

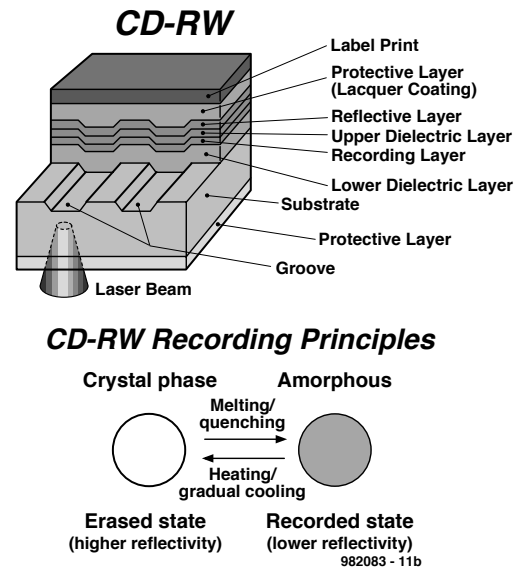


Figure 1. Physical construction of CD-R and CD-RW disks.

(just as with a 'normal' CD). This track serves to guide laser assembly. A coating of a special organic material is deposited on the polycarbonate layer, and this organic layer is covered by a reflective layer (gold, for example). The reflective layer is covered by a protective lacquer layer.

When a CD-R disk is written, the organic material is locally heated by the laser beam. This changes the reflectivity of the gold layer at the location that has been heated, and this change can be sensed as a pit by a CD reader. Although this is not a 'true' pit as with a normal CD, it can be adequately detected by a CD-ROM drive or an audio CD player.

In principle, the relatively new rewritable CD (CD-RW) uses the same construction as the CD-R (see **Figure 1b**). It also has a pre-moulded spiral track that serves as a guide for the laser assembly. On top of this there are two dielectric layers that sandwich a special material in which the pits are 'burned'. A reflective layer of gold or silver sits on top of these layers, and finally there is the usual protective lacquer layer. The storage layer can be switched between two different phases (an amorphous phase and a crystalline phase), depending on how quickly it is heated and cooled by the laser beam. A lifetime of more than one thousand phase changes is possible with present-day CD-RW disks.

Each phase of the storage layer material has a different reflectivity. Since the difference between the two reflectivities of a CD-RW is less than that of a normal CD or a CD-R, CD-RW disks cannot be read in all CD-ROM drives. Only relatively modern types (and DVD

players) can handle CD-RW disks properly; suitable players can be recognized by the designation 'multi-read'.

Data volume

There is a lot of confusion regarding the amount of data that can be stored on a CD-R(W). One manufacturer specifies a capacity of 650 MB, another claims 700 MB and yet another specifies '74 minutes'. In principle, all CD-Rs have the same capacity, plus or minus a few megabytes. The actual amount of data that can be stored depends on the type of information that is to be put on the disk. **Figure 2** compares the various possibilities. The length of the track on a CD is divided into minutes and seconds (which comes from audio CDs). The 'basic length' of a standard CD is 74 minutes. Each second consists of 75 sectors that contain data. Each sector of an audio CD contains 2352 bytes (see **Figure 2a**), which would amount to almost 747 MB for a 74-minute CD (if we calculate using the 'computer thousand' = 1024). If we take 'mega' to simply mean one million (as hard-disk drive makers like to do), we come up with the even more impressive figure of 783 MB!

However, if we want to store computer data on a CD, it turns out that extra synchronisation data and error-correction codes are needed. The original standard defines *mode 1* and *mode 2*, where mode 1 includes a significant amount of correction data. The result is that each sector can effectively contain only 2048 bytes of 'user' data. A CD-ROM can thus con-

tain at most (74 minutes) x (60 seconds) x (75 sectors) x (2048 bytes) = 650 MB. From this figure, we must subtract a few megabytes that are reserved for session and directory information. The XA standard, which is also shown in **Figure 2**, has a somewhat different structure but still has a net data capacity of 2048 bytes per sector for mode 2 form 1.

If you now think that you can use mode 2 (or XA mode 2 form 2) to put more data on a CD-ROM, you are in for a disappointment. These modes are only intended to be used for CDs where error correction is not particularly important, such as video CDs.

Sessions

In order to later add more data to a CD that already contains some data, we can utilize multiple sessions. A session is a block of data that is written to the CD by the CD-writer in one pass, and it includes a directory section placed after the data. A multisession CD contains one or more additional sessions located after the first session. Each time that a new session is added, the directory must be updated. This is done during the writing process by reading the directory information for the previous sessions and adding the information relating to the current session, to form a new directory that is then written at the end of the current session. This, and the fact that a lead-out (termination of the previous session) and a lead-in (start of the current session) must also be added, cause a certain amount of storage capacity to be 'lost': 22 MB for the first session and 13 MB for each following session. If the

a 1 sector: 2352 bytes

CD-DA (audio-CD):

2352 bytes audio-data

b mode 1

CD-ROM:

sync 12 bytes	header 4 bytes	user data 2048 bytes	EDC 4 bytes	8 empty bytes	ECC 276 bytes
------------------	-------------------	-------------------------	----------------	------------------	------------------

mode 2

sync 12 bytes	header 4 bytes	user data 2336 bytes
------------------	-------------------	-------------------------

(not used in practice)

c mode 2 form 1

CD-ROM XA:

sync 12 bytes	header 4 bytes	subheader 8 bytes	user data 2048 bytes	EDC 4 bytes	ECC 276 bytes
------------------	-------------------	----------------------	-------------------------	----------------	------------------

mode 2 form 2

sync 12 bytes	header 4 bytes	subheader 8 bytes	user data 2324 bytes	EDC 4 bytes
------------------	-------------------	----------------------	-------------------------	----------------

(EDC = error detection code
ECC = error correction code)

982083 - 12

Figure 2. The available data storage volume on a CD-R(W) is determined by the type of information to be stored and the format that is used.

disk is nearly full or you do not want to add any additional information, you can 'close' the CD so that no more sessions can be added. After this, such a CD-R can be read without problems in most CD-ROM players. Modern players, by the way, also have no problem reading multisession CDs that have not been closed.

Make sure that you choose the proper format before you start with the first session, since the format cannot be changed for subsequent sessions. The XA format (mode 2 form 1, often simply called 'mode 2') offers the most options and is also handled well by older-model CD readers.

It is also possible to put a new session on a CD and 'throw away' the old session at the same time. Of course, the old session actually remains on the disk, but it is rendered invisible to the CD player by omitting its directory information in the new session.

File names

You can choose from several file naming options when recording a CD-R. *ISO9660* is the best-known standard. It is the original CD-ROM standard, and it represents an attempt to devise a universal format that is suitable for the

majority of operating systems. In *ISO9660*, a file name consists of eight characters ahead of a period and three after the period. Only capital letters, underscores and numerals are allowed.

The *DOS format* also uses the 8.3 format, but allows all characters in the ASCII set to be used. Long file names (Windows 95) are truncated to eight characters before the period and three after, with a tilde (~) and a numeral placed in the name.

Long file names are also possible in the *ISO9660* format, but they are limited to 30 characters. Names that are longer than this are truncated and also marked by a tilde and numeral. This system is a good choice for CD-Rs that must be readable under Windows 95 and Unix.

The *Joliet* format, which has become fairly well known, was developed by Microsoft for Windows 95 to allow long file names to be used with CDs. A file name may contain at most 65 characters (including spaces) taken from the Unicode set.

The *Romeo* format offers even more room — up to 128 characters per name — but it does not support the Unicode character set. This format can be read by Windows 95, NT and

Macintosh systems (but only if the name is not longer than 31 characters, for the Macintosh).

The *HFS* (Hierarchical File System) format is the standard format for Macintosh machines. This format is also used by Mac systems for the hard and floppy disks.

UDF

UDF (Universal Disk Format) is fairly new file system that can be used for CD-R, CD-RW, DVD-ROM and DVD-Video. Compared to the older *ISO9660* standard, it offers many more options and makes it easier to use CD-R and CD-RW. For example, UDF allows files to be added to an existing CD one at a time without any loss of usable capacity. With CD-RW, UDF even allows you to delete individual files and to store new data in the space that is thus made available.

The two most important suppliers of software that uses UDF for writing and reading CD-R(W) disks are Adaptec (with DirectCD) and Cequadrat (with PacketCD).

A CD writer can put data onto a CD-R disk in UDF format practically without any extra effort, but things are more complicated with a CD-RW disk. A CD-RW disk must first be formatted, just like a hard disk, to divide the disk into 32-kB sectors. This procedure, which only has to be performed once, takes roughly one hour with the current models of CD writers. After it has been formatted, a CD-RW disk can be used in the same way as a hard disk.

A lot of storage capacity is lost as a result of the formatting, so that the net available capacity of a formatted CD-RW disk is just under 500 MB. (This naturally does not apply to CD-R disks).

With UDF, it is possible to write files to the CD-RW and to delete them. Current programs are also smart enough to regularly choose a different part of the CD to use for writing, to avoid overwriting certain portions of the disk so often that the disk thereby becomes unusable (the number of write/erase cycles is after all limited).

As long as a CD in UDF format has not been closed, it can only be read in the CD writer. This is because the directory is only written to the disk when it is closed. After it has been closed, the CD can be read by any modern CD player with multiread capability. If problems occur with certain drives, the above-mentioned programs allow a copy of the CD to be made in *ISO9660* format.

Writing methods

There are three basic methods for transferring data to a CD.

In the *track-at-once* method, the laser beam is switched off after each track has been written, even if several tracks are written to the CD in one operation. This is particularly important when the CD contains both data and audio tracks or when you are producing an audio CD. Normally, the gap (empty section) between the tracks lasts around two seconds. Some software and newer-model CD writers allow the length of the gap to be adjusted.

Some audio CD players have problems with the gaps between the tracks and may produce noise when they encounter a gap. In such cases, the *disk-at-once* writing method is preferable. In this method, all selected tracks are put on the CD in a continuous chain, without switching off the laser beam between tracks. This is of course only possible with a CD-R(W) that contains only a single session.

A variation of disk-at-once is *session-at-once*. This can be used when mak-

ing a CD-R in CD-Extra format, to write a session containing several audio tracks to the CD in a single operation. Since the CD is not closed after the session is written, additional sessions can subsequently be added to the CD.

The final method is called *packet writing*. This allows data to be written in small packets (normally according to the UDF standard). With CD-RW, a fixed packet length of 32 kB is used, since the whole file system would become too complicated (and thus too slow) if it had to deal with variable-length packets. Variable-length packets can however be used with CD-R since it is not possible to overwrite files and the file system thus does not have to 'remember' where space has been made free by deleting individual files.

Practical considerations

Currently, most CD writers are supplied with a program from Adaptec or Cequadrat that makes it easy for you to burn your own CDs. Often, it is only

a question of dragging the selected files from one window to another, or clicking on a button to copy an entire CD.

With the information presented in this article, you will probably find that you have no problems understanding the instructions for the 'advanced' settings of such programs. Now you know what they mean and what you have to watch out for.

A CD-R is a wonderful medium for storing more than 600 MB of data at a cost of only a few pounds — it has truly become an inexpensive medium. If you do not yet have a CD writer, maybe this will be sufficient incentive for you to acquire one. Whether CD-RW will become as popular as CD-R remains to be seen. The price of the CD-RW disk is currently the main obstacle — for the amount that you pay for one CD-RW disk (around £15) you can buy ten CD-R disks. Also, in practice it turns out that there is not actually all that much data that has to be frequently rewritten.

(982083-1)

The CD books

The specifications of the various types of CD format are defined in a number of books that are named after the colours originally used for their jackets.

Red Book

This is the original standard for audio CDs, as defined by Philips and Sony more than 20 years ago. Here you will find (among other things) specifications for the audio data format, the structure of the tracks on the CD, the maximum number of audio tracks (99) and the methods used for error detection and correction.

Yellow Book

This is a supplement to the Red Book, which describes how computer data must be put on a CD. It defines two modes: mode 1 and mode 2. A later supplement describes the XA format (eXtended Architecture), which allows audio, video and computer data to be placed on a single CD. This supplement was made necessary by the development of the Photo CD.

Orange Book

This book specifies the formats for CD-R, magneto-optical disks and other types of (re)recordable CDs. It also includes the multisession standard. Many of the specifications in the Orange Book are also reused in other books.

Blue Book

This defines the organizations of the CD-Erasable and CD-Extra formats (also known as CD-Plus or CD-Enhanced). In CD-Extra, audio and computer data are stored on two separate parts of the CD, with the computer data always placed after the audio data to avoid problems with normal audio-CD players. The audio portion is structured according to the Red Book standard and the computer data portion according to the Yellow Book standard. Multisession capability is derived from the Orange Book standard.

Green Book

The Green Book contains the standard for CD-i, which was brought to life by Philips. One of the most important features of this standard is the synchronization of separate audio and video tracks. Both CD-ROM players and audio-CD players can usually not read such CDs.

White Book

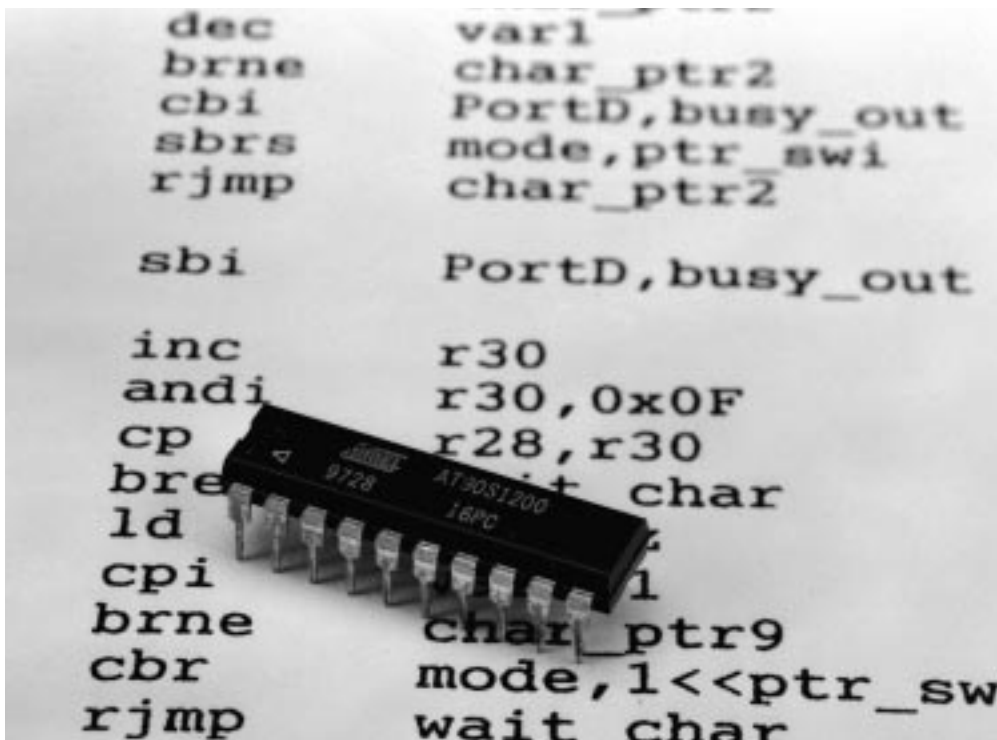
This defines the standards for putting video data on CDs. It also contains the MPEG specification for compressing video and audio data. Such CDs are organized according to the XA format.

We've used superlatives often enough in *Eektor Electronics*, but there's no question that this is the most compact computer design we have ever published. Just one controller IC and one TTL buffer chip are all that is needed to build a microcomputer with a printer port, a serial port and a connection for an LCD.

Design by H. Böhling & D. Wulf

A compact display controller

driving a display or printer using only two ICs



The AT90S1200 is the key to this design. You can easily program it yourself, especially since the necessary software can be downloaded free of charge.

Specifications

Processor:	AT90S1200
Clock rate:	16 MHz
Serial input:	TTL
Baud rate:	2400, 4800 or 9600
Serial handshaking:	DTR
Parallel port:	8-bit bidirectional; Centronics-compatible output
Parallel handshaking:	busy & strobe
Display:	standard LCD module, 2 rows of 16 characters

It's simple, really, once you think of it: start with a compact but powerful controller IC (the AT90S1200) with built-in ROM and RAM, add an HCT245 TTL buffer chip and write a few clever software routines. The result: a microcontroller with a printer port, a serial interface and provision for driving a liquid-crystal display. The potential applications are too numerous to mention. What about supervising a parallel interface or monitoring the communications with a serial device, for instance?

Naturally, there are some other components needed in addition to the two ICs, but only a few. A 16-MHz crystal and two ceramic capacitors take care of generating the clock frequency, and a trimmer potentiometer allows the display contrast to be adjusted. In addition, a few small capacitors provide the necessary decoupling for the supply voltage (see **Figure 1**).

The designers have used this circuit as a display driver for the printer port of an MCS-51 system (note: it can be used either as a display driver or a printer driver, but not both at the same time). It can be connected to the MCS-51 system using only three leads: +5 V, earth and LPout (Port 1.7, located at pin 8 of the 8032). With only a small additional effort, it can be used as a universal serial-to-parallel converter. An interesting feature is that the circuit draws so little current (4.6 mA when idle, 13.6 mA in the operating mode) that the serial port can be used as its power source. If it is necessary to monitor the connected printer via a 'busy' line, then an interrupt input can

be made available.

Driving the controller from an MCS-51 system is relatively simple. First send the BASIC instruction BAUD 2400 to select the desired serial data rate (rates of 4800 and 9600 are possible with suitable modifications to the software). After this, you can send the data plus the instruction PRINT# or LIST# to the controller via the serial port. The controller will then look after presenting the information on a display or printing it on an attached printer.

Your own applications

It goes without saying that this circuit lends itself to experimenting. The source code for the software is available through our Readers Services (order number 986030-1). This is extensively commented, so it should be possible for you to adapt the controller for your own applications, independent of the example presented here. If on the other hand the sample application is just what you need, a ready-to-use, programmed version of the controller (IC1) is also available through the Readers Services (order number 986521-1). The software that you will need to do your own programming, such as an assembler and a simulator, is available free of charge from Atmel's Internet site. You will also find supplementary information and sample programs at this site; the address is <http://www.atmel.com>. Look under the heading 'products' for 8-bit AVR processors. There you will find AVREXE in the list of downloadable software. This package includes extensive information the form of PDF documents, as well as the assembler and debugger software. If you do not have access to the Internet, the software is also available through the Readers Services (order number 976017-1). The Handyman programmer (published in the December 1997 issue of *Ektor Electronics*) is (among others) a suitable programmer for the controller IC.

Your own construction

A ready-made circuit board for this design is not available, since it is extremely compact and also particularly well suited for integration into larger designs. In addition, a small piece of prototyping board should be all that you need in most cases.

As already mentioned, the circuit can in principle draw its operating power directly from the serial interface. In this case it is essential to use a three-terminal regulator (such as a 78L05) and two capacitors to regulate the supply

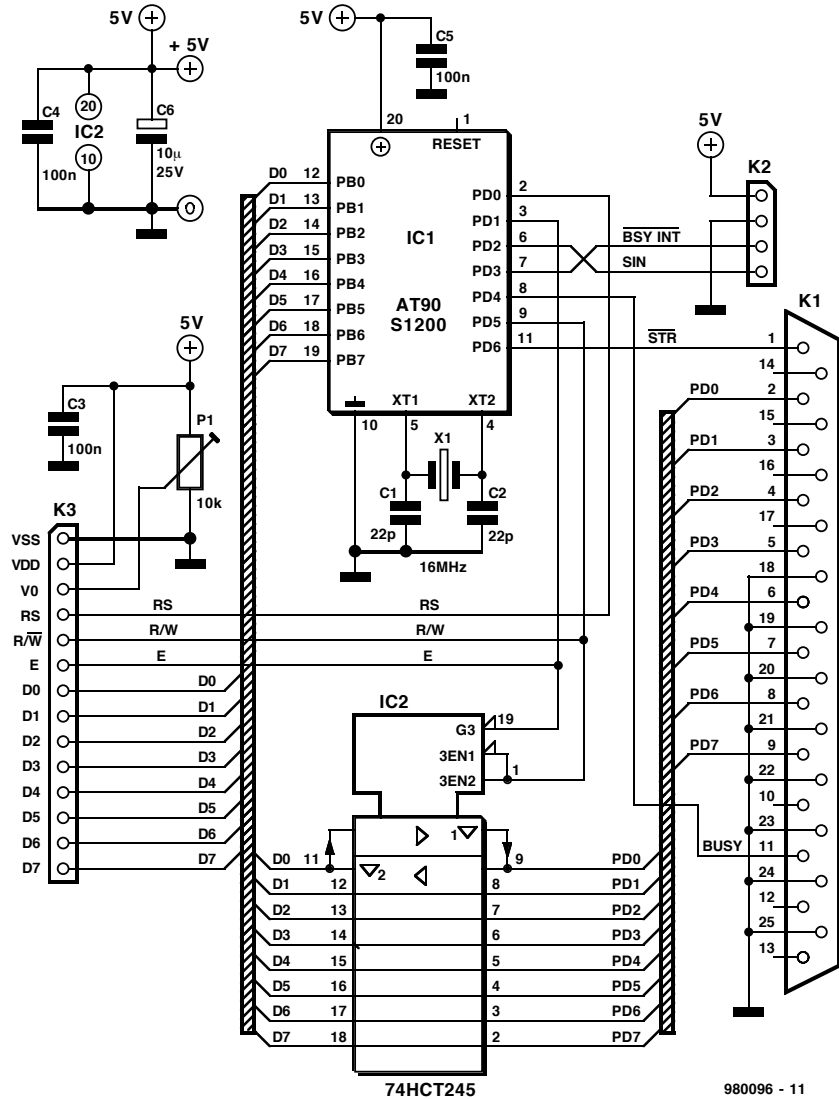


Figure 1. The complete schematic. Since the entire controller consists of only two ICs, it can justly be called extremely compact.

voltage at exactly 5 V. If the controller is used with the previously-mentioned MCS-51 system, the necessary voltage is already present; a few mA more or less shouldn't make any practical difference.

A standard two-row LCD module can

be used as a display. Adjust the display contrast using P1 after power has been applied — the adjustment is correct as soon as the cursor is clearly visible. Your circuit is now ready for use.

(980096-1)

Control codes

PRINT# CHR(1)	switch from printer mode to display mode *
PRINT# CHR(6)	circuit reset
PRINT# CHR(7)	switch from display mode to printer mode *
PRINT# CHR(8)	backspace
PRINT# CHR(9)	tab character
PRINT# CHR(10)	line feed (LF)
PRINT# CHR(11)	home
PRINT# CHR(12)	erase screen
PRINT# CHR(13)	new line (CR)
PRINT# CHR(14)	cursor off
PRINT# CHR(15)	cursor on

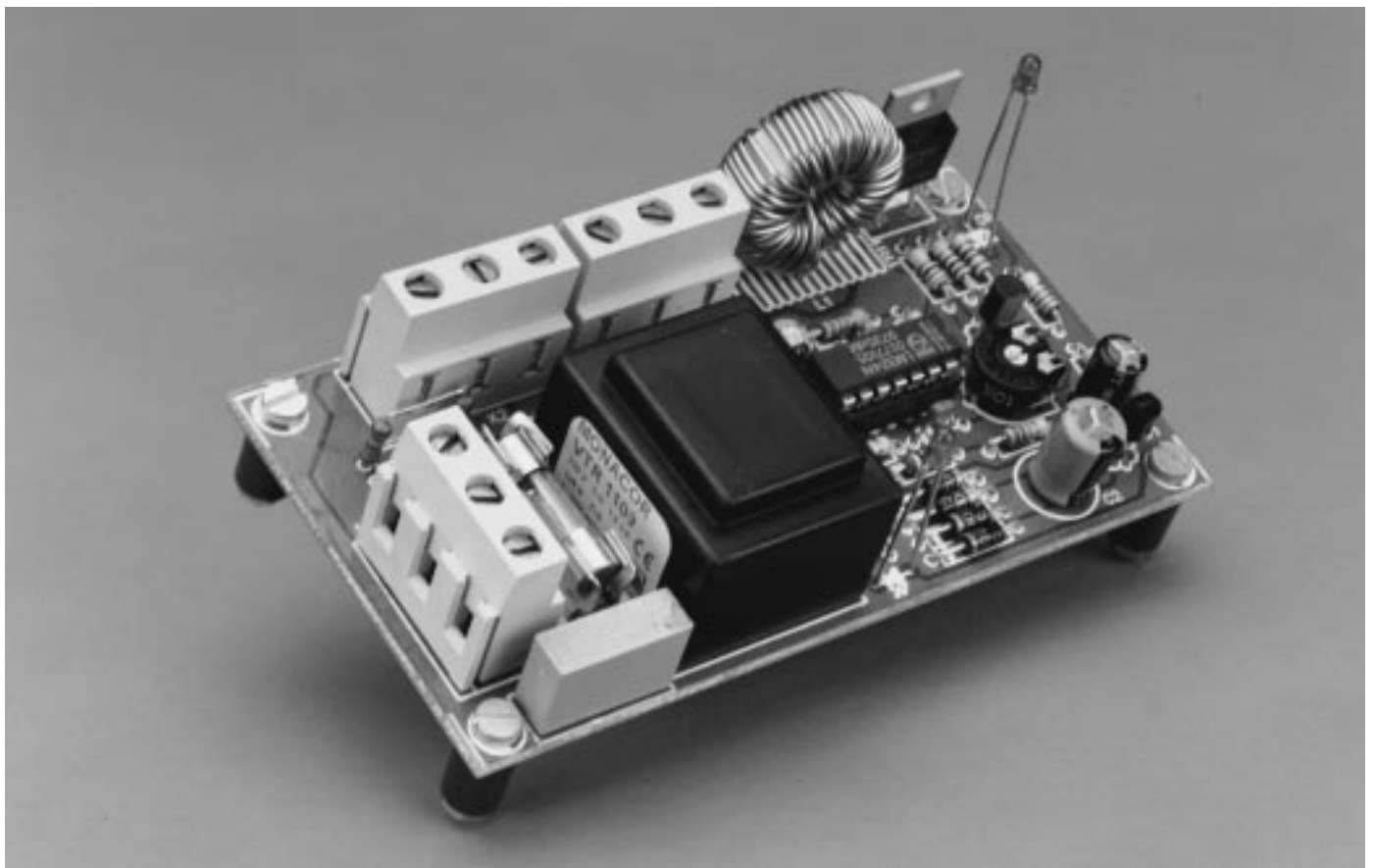
* The control code CHR(7) causes all subsequent data to be sent to the printer. As soon as the control code CHR(1) is received, the printer is disabled and all subsequent data are sent to the LCD module.

Many PCs have a switched mains outlet that enables your computer and monitor to be switched on at the same time. However, because it is not designed to supply high currents, this outlet should not be used to switch on too many peripherals. If you want to use the on/off switch on your PC to power up the whole computer workstation, you should use a so-called master/slave switch. Unfortunately, most commercial models and kits offer poor performance when switch-mode power supplies are hooked up as the master equipment.

Design by T Scherer

master/slave switch 'deluxe'

switch on slave devices
safely and automatically



To enable the large number of peripherals connected to his Macintosh to be switched on together with the computer, the author "got hold of a master/slave switch DIYkit that came com-

plete with a printed circuit board, and an enclosure with the necessary plug and sockets. Alas, such gizmos do not work very well when the master device is a switch-mode power sup-

ply. I heard crackling and ticking sounds, and sometimes the monitor and other peripherals would suddenly come to life with the computer switched off."

Musing on this problem the author started to design a master/slave switch which is virtually immune to voltage surges and other irregularities, and sent his practical circuit to *Ektor Electronics*.

The circuit

The circuit shown in **Figure 1** may be thought of as consisting of three parts: an ordinary 5-V power supply, a current sensor and a comparator with a power output driver. The mains voltage is connected to K1, the master, to K2, and the slave, to K3. Apart from a fuse, F1, there is only the current sense resistor, R5, between K1 and K2. As soon as current starts to flow to the master, R5 drops an alternating voltage which is limited to ± 0.7 V by D5 and D6. R5, incidentally, allows a master power consumption of up to 600 watts. Low-pass filter R7-C4 removes high-frequency noise from the alternating voltage. As mentioned above, it is particularly this type of noise that wreaks havoc in commercial master/slave switches. IC2a amplifies the positive half-cycles about 3 times, while D7 blocks the negative half-cycles. In this way, a pulsating direct voltage is created whose peak value is stored in capacitor C6. Buffer IC2b completes the current-sense sub-circuit.

The next comparator, IC2c, defines the switch-on threshold of the circuit. The green low-current LED D8 not only acts as an on/off indicator (i.e. circuit connected to the mains), but also provides a stable reference voltage of about 2 volts. Preset P1 allows a threshold to be set that corresponds to any master power consumption between 10 W and 100 W. Resistor R9 adds a little hysteresis to the comparator, preventing its output from 'chattering'. The red LED, also a low-current type, lights when the comparator output drops low. This means that the circuit is active. Inverter stage T1, finally, is included to provide sufficient gate current for triac T1.

The triac used here is an SGS-Thomson device marked by an extremely sensitive gate — the device is triggered by a current as low as 5 mA. Moreover, the BTA04-600T does not require a so-called snubber network that is normally indispensable with triacs. The triac is capable of switching up to 4 A, allowing slave equipment with a power consumption of just under 1 kW to be connected. A ready-made suppressor coil, L1, is included to attenuate transients and RF interference. The same triac family

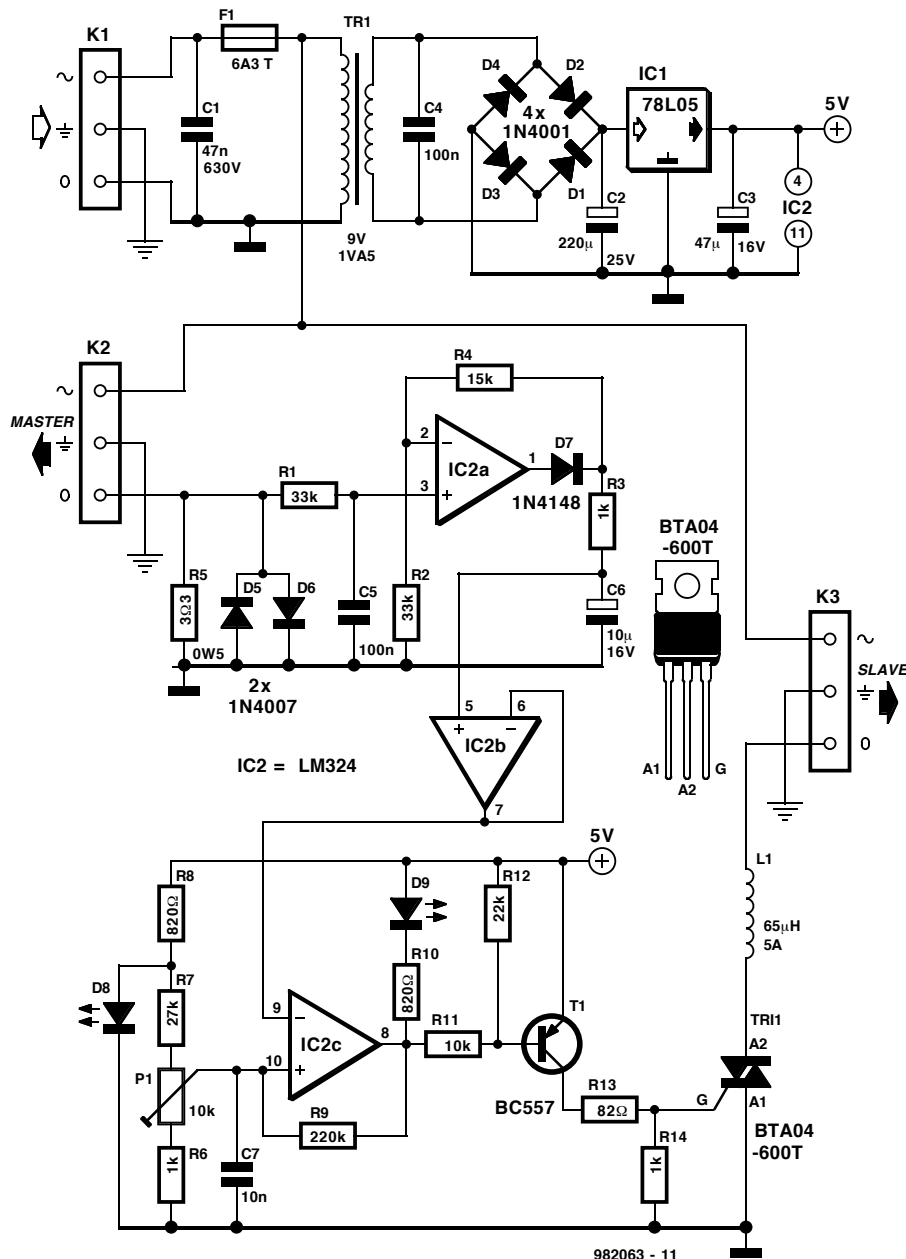


Figure 1. The circuit of the master/slave switch consists of three functional sections

includes types with a switching capacity of 6 A (BTA06-600T), and 8 A (BTA08-600S). Although it is possible, in principle, to use these devices in the present circuit, the PCB then has to be modified to cope with the higher currents: strengthen the high-current tracks by soldering solid wire over them, increase the wire diameter of L1 from 1.0 mm to 1.5 mm, and exchange the fuse with a type rated for the higher current.

The circuit has a conventional power supply. A small mains transformer, TR1, a rectifier, D1-D4, and a reservoir capacitor, C2, provide a raw direct voltage which is stepped down and stabilised at 5 V by fixed-voltage regulator IC1. By the way, the circuit itself has a modest power consumption of just 200 mW.

Construction: safety first

Before you start soldering away, do take note of the following: the circuit, including the low-voltage section, is not electrically isolated from the mains. The circuit must, therefore, not be built on any other circuit board than the one shown in **Figure 2**, which has been designed to comply with every requirement in respect of electrical safety.

Start the construction of this project by fitting the low-profile parts, and work your way 'up' to the taller parts, with the transformer soldered as the last component. The two LEDs are mounted at such a height that they fit snugly in the holes drilled in the enclosure top panel. Two properly isolated, transparent lenses not only contribute

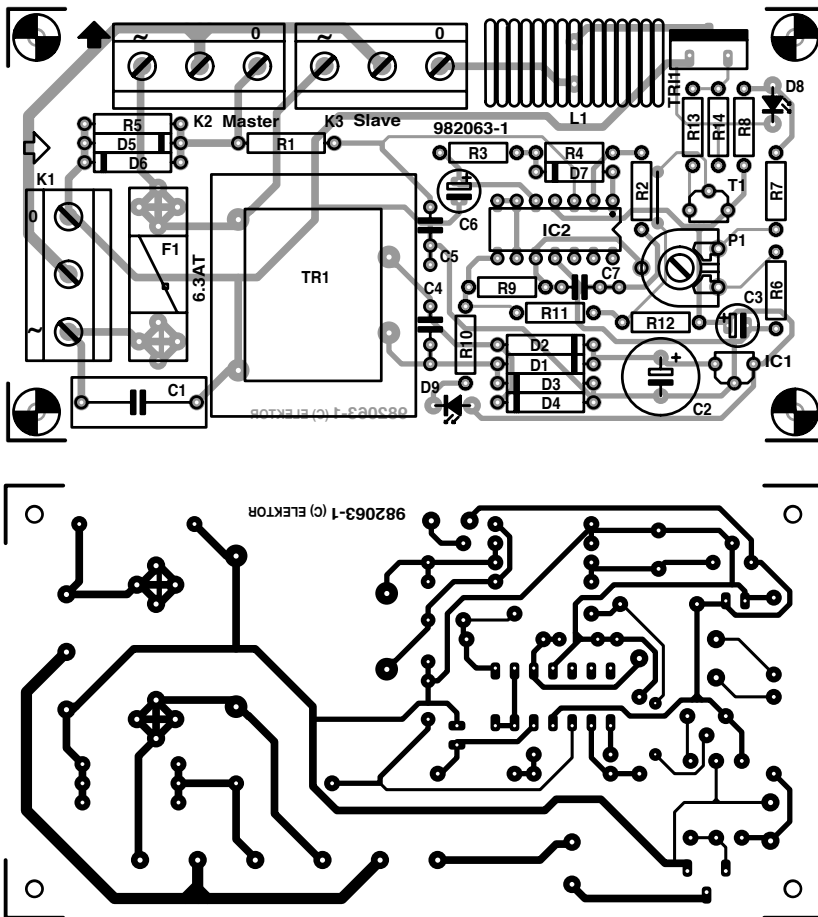
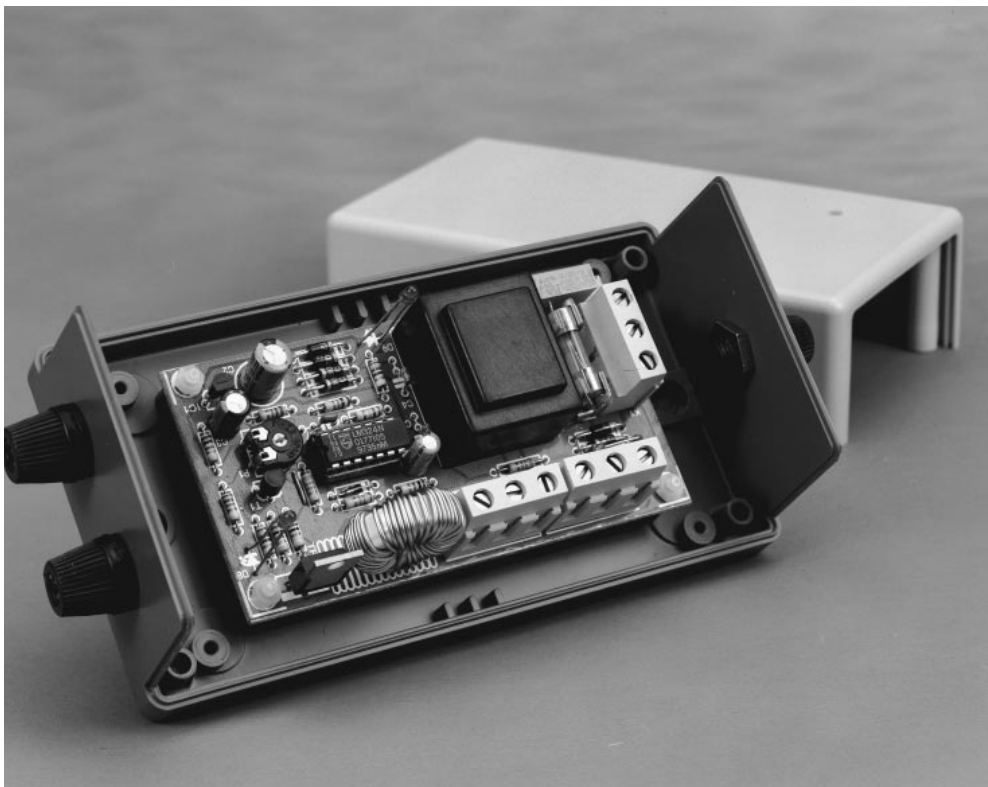


Figure 2. The carefully designed PCB ensures safe constructing.

Figure 3. It is essential that the circuit be mounted so that it can not be touched when in operation.



COMPONENTS LIST

Resistors:

R1,R2 = 33k Ω
 R3,R6,R14 = 1k Ω
 R4 = 15k Ω
 R5 = 3 Ω , 0.5 W
 R7 = 27k Ω
 R8,R10 = 820 Ω
 R9 = 220k Ω
 R11 = 10k Ω
 R12 = 22k Ω
 R13 = 82 Ω
 P1 = 10k Ω preset H

Capacitors:

C1 = 47nF 630V (Class X2)
 C2 = 220 μ F 25V radial
 C3 = 47 μ F 16V radial
 C4,C5 = 100nF
 C6 = 10 μ F 16V radial
 C7 = 10nF

Semiconductors:

D1...D4 = 1N4001
 D5,D6 = 1N4007
 D7 = 1N4148
 D8,D9 = LED
 T1 = BC557
 Tri1 = BTA04-600T (SGS-Thomson)
 IC1 = 78L05
 IC2 = LM324

Miscellaneous:

F1 = Fuse 6.3A slow with PCB mount holder
 K1,K2,K3 = 3-way PCB terminal block, pitch 7.5mm
 Tr1 = mains transformer, PCB mount (Conrad Electronics o/n 506052)
 L1 = suppressor choke 65 μ H/5 A (Conrad Electronics o/n 534439)
 Heatsink for Tri1 (Fischer SK59)
 2 LED lenses, transparent (Conrad Electronics o/n 539910)
 PCB, order code 982063-1 (see Readers Services page).

to the 'professional' look of the master/slave switch, but also afford touch protection — after all, the LEDs, too, may be at mains potential!

The completed and (visually inspected) circuit board is mounted 'touch-proof' in an ABS case. Our lab prototype is shown in **Figure 3**. Please be sure to undertake the adjustment of P1 with an insulated screwdriver — we do want to keep you as a reader!

(982063-1)