ELEKTOR ELECTRONICS

THE ELECTRONICS & COMPUTER MAGAZINE

FEBRUARY 2001
£2.95

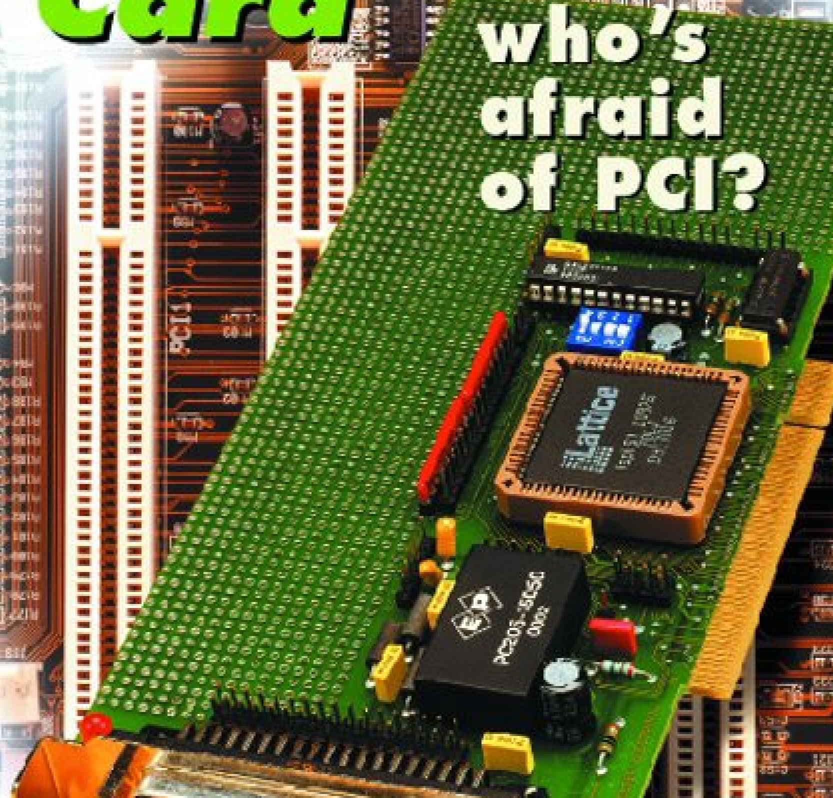
www.elektor-electronics.co.uk

0-1.8 MHz
AM Receiver

DS1621
Programmer

PCI Bus Prototyping Card

who's afraid of PCI?



Revamped
BASIC-52

Home Bus
Systems

Low-Power
Electronics

Micro-
controllers
Overview

PCI Bus Prototyping Card (1)

Explore the PCI-Bus

By H. Kolter (Kolter Electronic)

The development of a PCI based expansion card is relatively complex and usually needs a large investment of time and money before success is achieved. The PCI prototyping card presented here offers a more convenient solution. Its pre-configured PCI decoder and 16 bit wide data bus together with a series of port addresses allows you to concentrate on development and testing of your prototype circuit without getting bogged down in the detail of the PCI bus standard.

The Peripheral Component Interconnect or PCI Bus is a relative newcomer to the computer interface world but has rapidly replaced the older and well known ISA Bus. One attractive feature for peripheral designers is that this bus is used not only for IBM compatible PCs but also for workstations and Apple Macintosh systems. Basically, the interface is a very powerful multiplexed 32/64 bit wide address/data bus. One of the original aims of this bus standard was to ensure that it would not become obsolete overnight and would still be in use after many generations of processor have come and gone.

The PCI bus is independent of the system processor and is almost a stand-alone system bus capable of a relatively high data transfer rate.

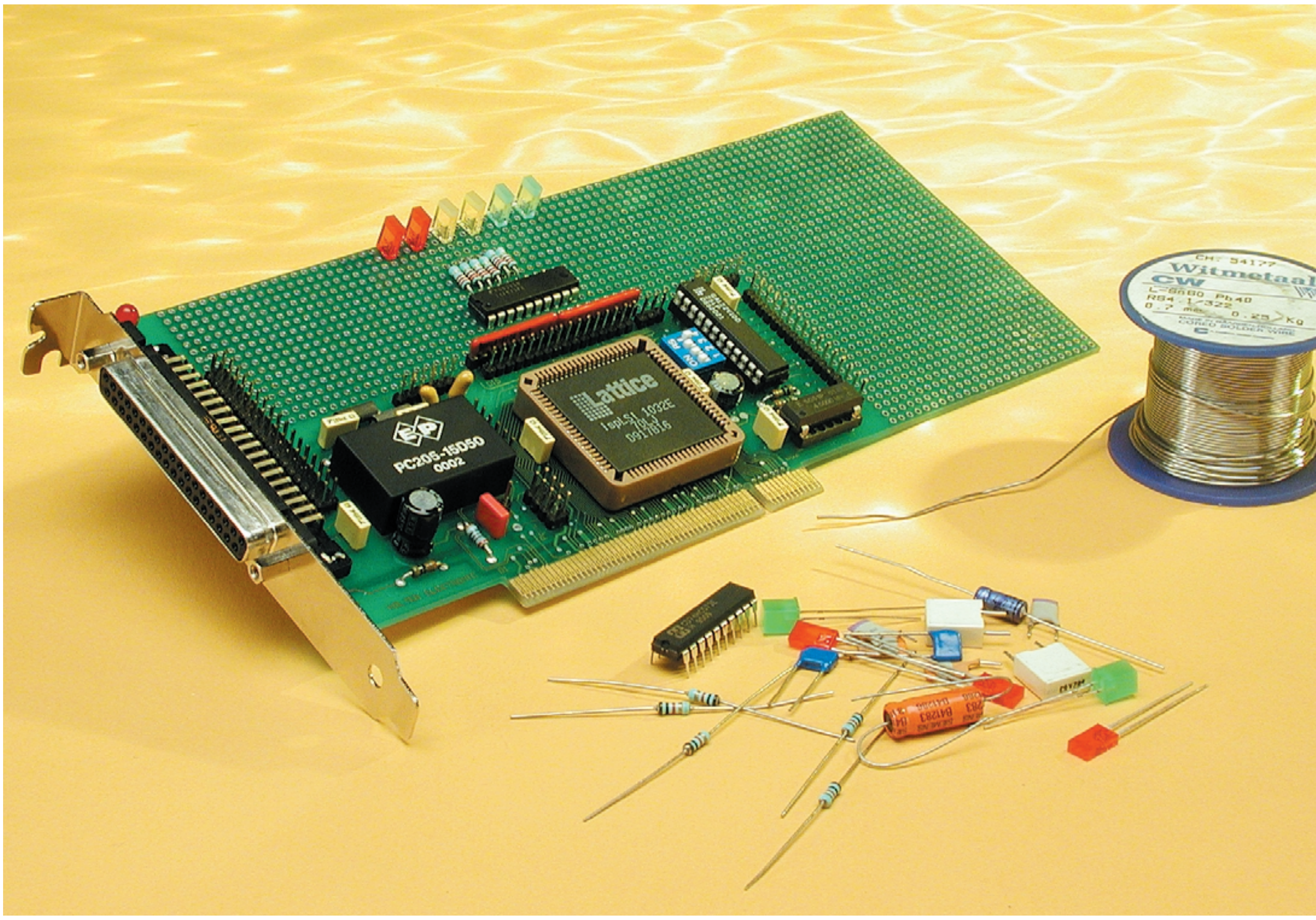
Modern PC motherboards use PCI slots and no longer provide any of the older ISA slots. The disadvantage for the designer of a PCI plug-in card is that the design process is a bit more complex. It is no longer possible to fiddle with hardware jumpers to select the address space of the expansion card — instead, the card configuration is performed with software. The PCI card contains its own operating system that is initialised by writing to internal registers over the PCI interface. The PCI BIOS will recognise each of the PCI expansion cards fitted to the computer system and will initialise them during the BIOS set-up procedure. Once the card has been configured under BIOS, it is assigned an access address. There are different PCI routines that allow this information to be read or altered.

PCI board features:

- 32-bit PCI Controller implemented with ispLSI
- 16 bit User Interface similar to ISA-Bus
- Accessible Decoder GAL22V10 (Synario)
- Large 2.54-mm matrix of through-plated holes
- DC/DC converter +/- 15 volts
- Suitable for prototyping digital or analogue circuits
- Selectable Product ID
- 8/16 bit Bus with CS signals

PCI Controller and Vendor ID

The central core of the PCI expansion card is the PCI decoder or controller chip. This chip forms the link between the PCI bus of the computer and your custom circuitry. Most commercial PCI controller chips on the market are not only very complex (and expensive) but are also relatively unwieldy beasts because of their large package outlines. For our PCI experimentation card in this project, the PCI controller has been implemented in an ispLSI1032 pro-



grammable chip from Lattice. This solution is not optimal for every PCI application — it was developed to be used for current 16 bit ISA applications. The decoder is intended to be used for the slow and middle speed data rate of the PCI specification less than or equal to 33 MHz, supporting the specifications 2.1 and 2.2 of the PCISIG group of the PC98 and PC99 specification. The PCISIG group is also responsible for assigning a unique Vendor Identity (**VID**) code to each manufacturer of PCI compatible products and also a Device ID code for each type of PCI product that the manufacturer produces. These codes, along with information of the cards I/O and memory address range and interrupts, are contained in a 64-byte information header stored on the card. This information is read by the PCI BIOS whenever the computer is re-booted.

Technical Data

Digital Output:	1 x TTL level (0/+5 V)
Modes:	TTL OUT port polling, or program controlled
PCI Decoder:	1 x ispLSI 1032E (Lattice)
Vendor-ID:	KOLTER 0x1001
Product-ID:	KOLTER 0x0017 (adjustable)
Addressing:	variable, PNP (Plug and Play)
Bus:	32/64 bit PCI
User Interface:	8/16 bit

Connectors:
 37 pin sub-D connector (K4), connector on slot bracket
Pin strips for the breadboard wiring:
 1 x 9 pin. (K9), ±15 V, AGND and +5 V, GND, power supply for analogue prototyping components.
 1 x 6 pin. (K8), interrupt select
 2 x 17 pin (K5), patch pins for the output socket. (see Figures 1 and 5)
 1 x 3 pin (K6), not defined. Useful for application-specific prototyping signals.
 1 x 6 pin (K10), 3 way power supply +5 V, GND, for prototyping digital circuits.
 1 x 16 pin (K3), data bus
 1 x 16 pin (K2), I/O-Bus
Dimensions:
 95 x 215 mm (card dimensions without bracket)

Temperature range	0-70 °C typical. (continuous operation)
Storage temperature	–20 to +85 °C

Card features

The PCI prototyping card provides an area of through-plated holes of 87 × 37 solder points (minus 46 × 21 for the PCI decoder). This breadboard area provides a useful space to lay out prototype hardware rapidly and easily for your specific PCI application, e.g., A/D or D/A converters, TTL I/O circuitry or general-purpose interfaces. A power supply providing ±15 V from a separate DC/DC converter is also included on the card and a 16 bit data bus together with a series of port addresses ensures that you will be able to access your hardware in under 90 ns. All the necessary components are provided on the card and configured to transfer data to and from the PCI bus.

Configuration of the card address occurs automatically with Plug and Play (PNP). Extensive PCI tools running under Windows 95/98/NT and 2000 enable the address I/O address range of the card to be determined. No further initialisation of the components is necessary to be able to read and write to the card at register level.

The circuit

A quick look at the block diagram in **Figure 1** shows that the PCI prototyping card consists of four main parts:

IC1: PCI bus adapter chip type ISPLSI1032

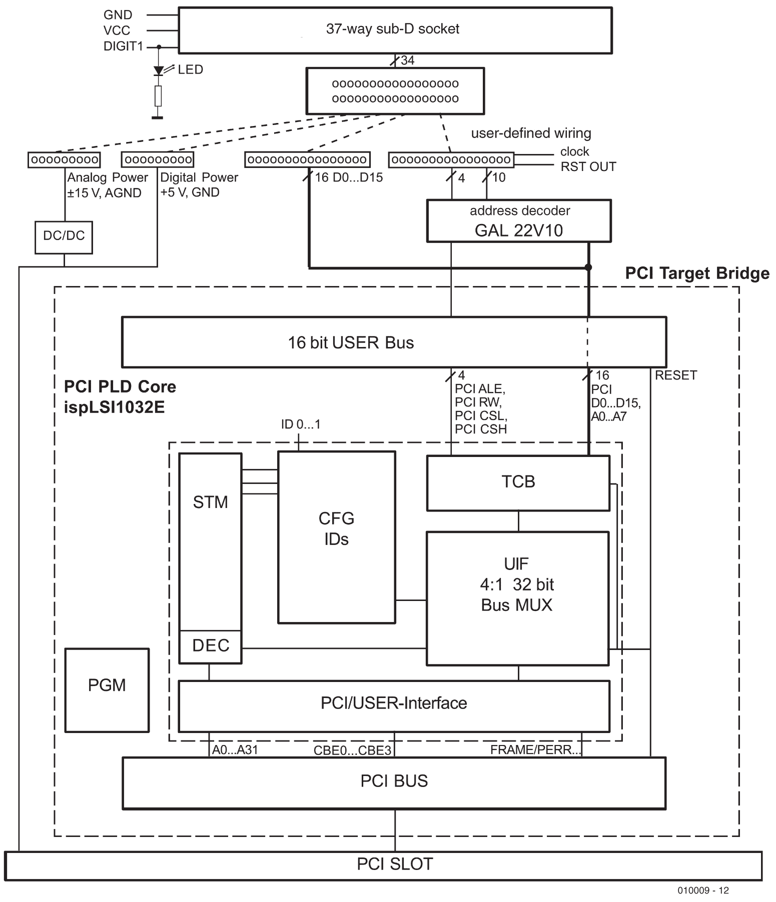


Figure 1. Overview of the functional blocks of the PCI prototyping card. An explanation of the abbreviations used is given in the inset.

Abbreviations used in the Block Diagram

PCI/USER Interface: Bus interface with bi-directional port and bus driver for the signals: CLK, System Pin, Address/Data Pins, CBE and other Interface-Control Pins along with Error Reporting Pins.

STM, DEC and CFG-ID: STM stands for State Machine, this reads out the information to the PCI bus over the 4:1 32-bit multiplexer of all the 64 bytes of CFG registers when the information request is made by the PCI BIOS, e.g., during computer boot-up. The CFG register is 'pre-wired'. The product ID inputs ID0, ID1 and ID2 allow selection of one of eight ID's. The Vendor ID is pre-programmed and not selectable.

UIF and Bus-MUX: Bus driver and Turnaround are generated here for the PCI target. The bus multiplexer co-ordinates the address and data in both directions. All the read and write instructions are used in this interface.

TCB: The Timing Correction Block generates pre-defined Bus Cycles so that the user bus runs with precise timing. This block is necessary to cater for motherboards with different bus timings.

PGM: Programming Interface for the ispLSI chip (not freely accessible).

ALE: Address Latch Enable, switches the multiplexed user bus between address and data.

D0 - D15: 16-bit wide data bus. Data is latched with ALE and CSL/CSH.

A2 - A7: Address lines used to generate the memory mapped chip selects. Address line A0 is not used (decoded from CSL and CSH) Address line A1 is also not used. Only the lower 16 bits of the 32-bit data word can be decoded.

RW: User Bus Signal Read/Write: for decoding the read and write cycles for the I/O timing

CSL / CSH: Chip Select Low and Chip Select High signal. When an 8-bit input or output is performed CSL will be active when the lower 8 bits of the 16 bit data word are used and CSH will be active when the upper 8 bits are used. If a 16-bit instruction is used then both lines are active so that a 16-bit word can be read or written in one cycle. It is always the lower 16 bits of a 32-bit data word that are decoded.

RESET: Activating this line will force the chip into its initialisation configuration.

with ID switch.

IC2: address decoder gate array type GAL 22V10.

IC4: DC/DC voltage converter chip together with its Radio Frequency Interference (RFI) suppression components.

K2 to K10: these rows of pin strips and connectors are free to be wired to the prototyping area of the card.

K4 is a sub D type connector providing connections to external signals.

Looking (ahead) at the circuit diagram (**Figure 2**) and the PCB layout in **Figure 3** you'll

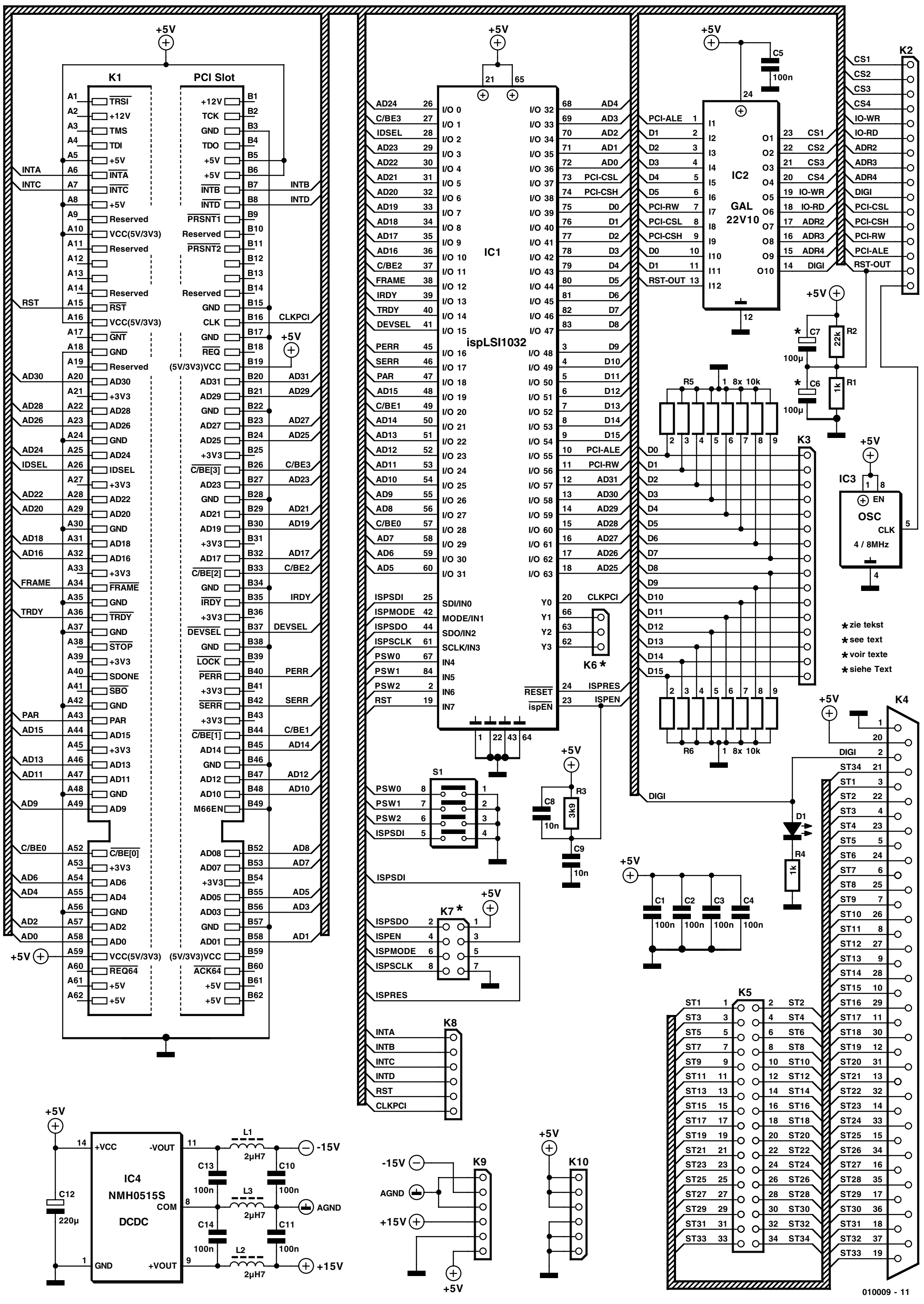
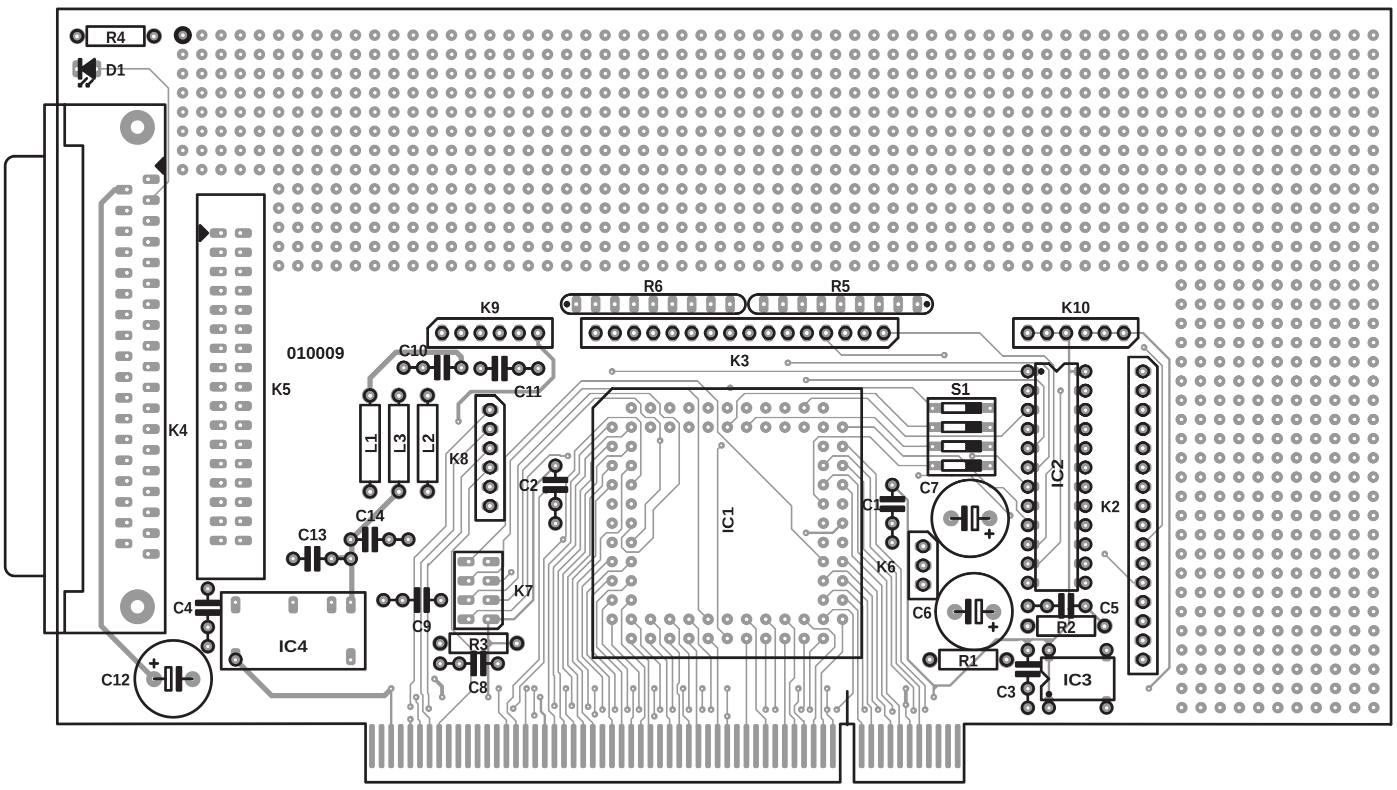


Figure 2. Circuit diagram of the PCI prototyping card. The biggest part of the circuit, the PCI bridge which links the PCI bus to the prototyping area circuits, is implemented in the ispLSI-Chip.



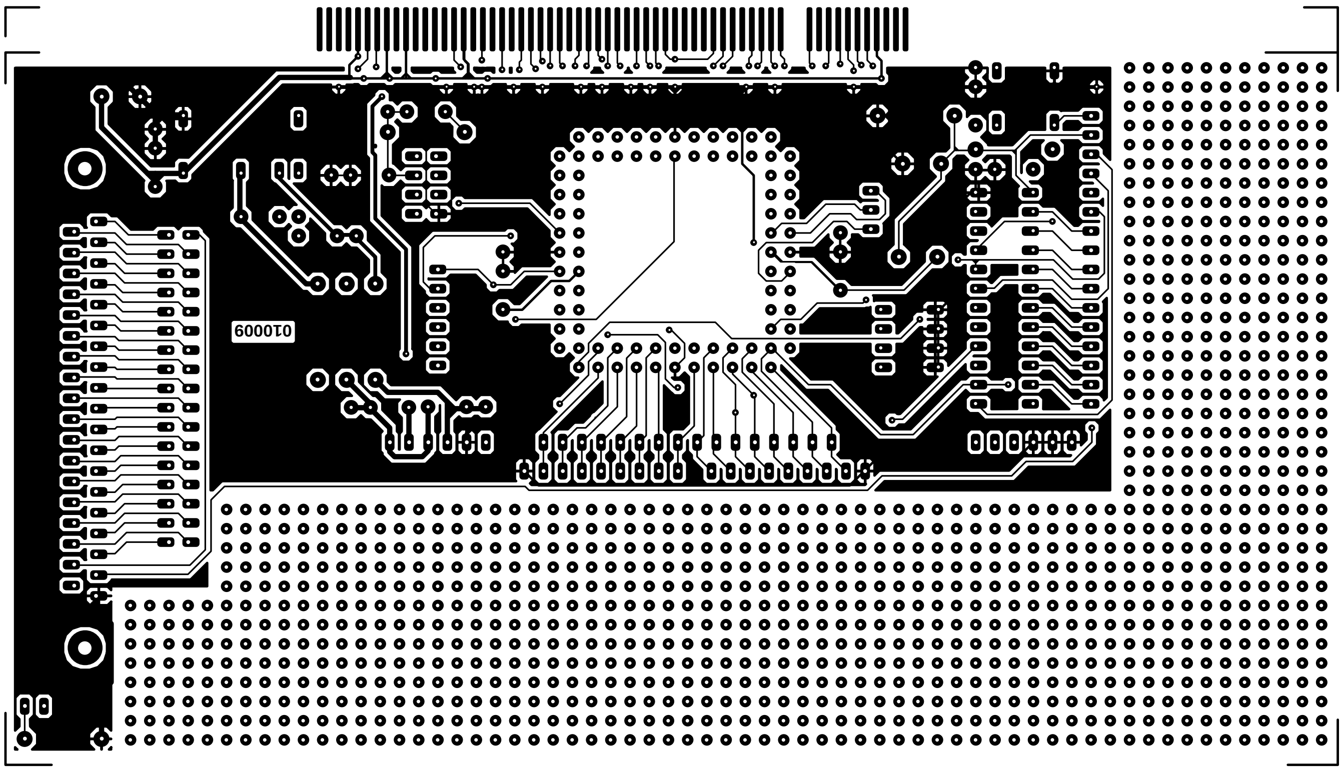
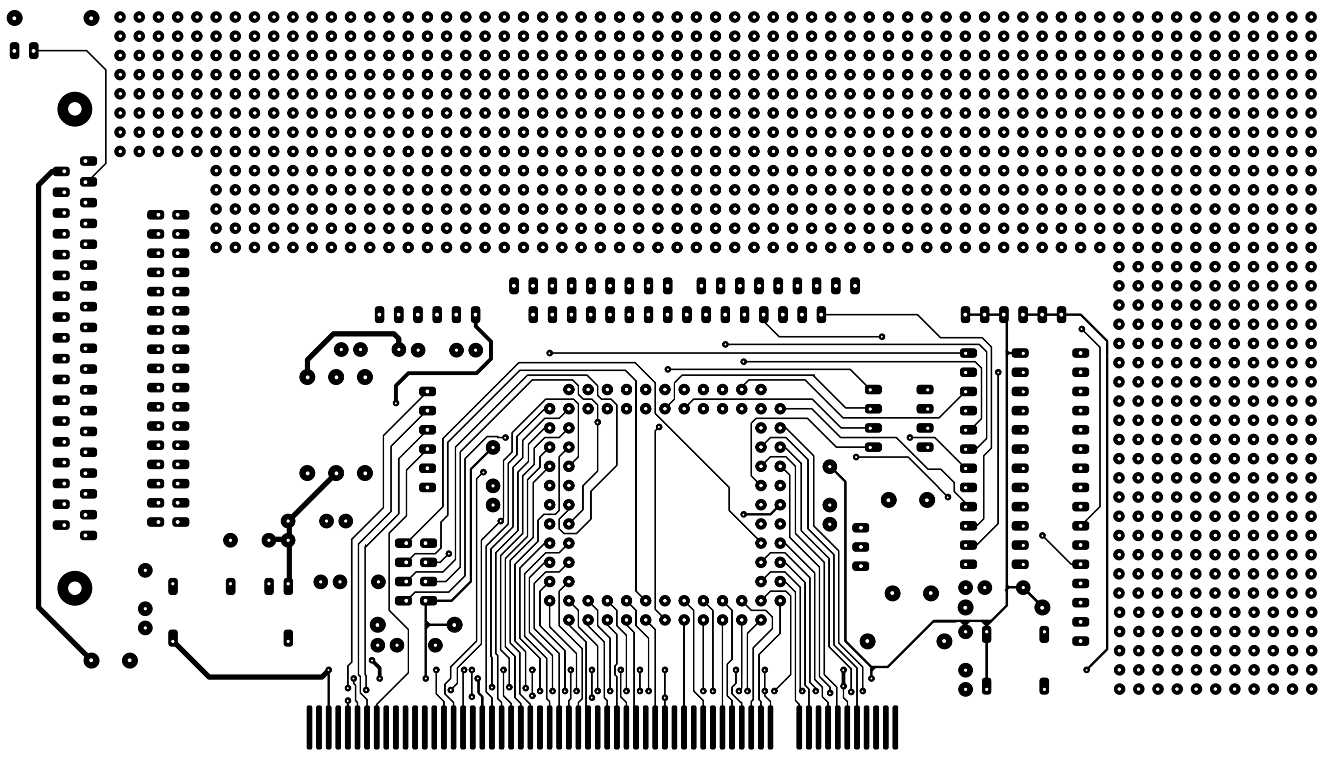


Figure 3. PCB layout and placement plan.

notice that the most complex function of the circuit is performed by IC1, the pre-programmed LSI chip ispLSI1032 from Lattice.

The block diagram for the present PCI card in shown in **Figure 1.** IC1 is shown here as the 'PCI target bridge'. A key to the abbreviations used on this diagram is given at the end of the text.

The Vendor ID code mentioned earlier is pre-programmed into IC1 and in this case has the value 0x1001. This is the VID of Kolter Electronic, Germany, and can be used on all the applications in which the ispLSI-chip from Kolter is employed.



COMPONENTS LIST

Resistors:
R1,R4 = 1kΩ
R2 = 22kΩ
R3 = 3kΩ9
R5,R6 = array 8 x 10kΩ

Capacitors:
C1-C5,C10,C11,C13,C14 = 100nF
C6,C7 = 100µF 16V
C8,C9 = 10nF
C12 = 220µF 16V

Inductors:
L1,L2,L3 = 2µH7 (see text)

Semiconductors:
D1 = LED
IC1 = ispLSI1032 (order code **010009-41**)*
IC2 = GAL22V10 (order code **010009-42**)*
IC3 = oscillator module in 8-pin DIL case (see text)
IC4 = NMH0515S (see text)

Miscellaneous:
K1 = PCB edge connector for PCI bus
K2,K3 = 16-pin SIL pinheader
K4 = 37-pin sub-D socket
K5 = 34-pin boxheader or pinheader
K6 = not fitted**
K7 = not fitted **
K8,K9,K10= 6-pin pinheader
S1 = 4-way DIP switch
PCB, order code **010009-1***

* For programmed controllers, GALs and PCB, see Readers Services pages elsewhere in this issue, or website at http://www.elektor-electronics.co.uk.
** See text

IC1 forms the bridge between the computer PCI bus and the 16-bit wide bus of this prototyping card. If your application only requires an 8 bit bus then it is a simple matter to leave the other 8 bus bits unused. The data transfer rate will be halved, but this is not always an important criterion and may be perfectly adequate for your application. Expanding the bus to 32 bits wide is unfortunately not possible with this LSI.

DIP switch S1 has two functions. The fourth of the four switches, S1-4, allows the switching between the PCI local specification 2.1 and 2.2 while switches S1-1, S1-2 and S1-3 allow the device ID to be changed (**Figure 4**). This allows several PCI prototyping cards to be plugged into the same PCI bus, and the control software will be able to distinguish between each card on the basis of its vendor ID, device ID and its slot number.

IC2 is a programmed gate array containing an address decoder which produces four chip selects outputs $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$ and $\overline{CS4}$ and generates a $\overline{RD}$ and $\overline{WR}$ signal along with three decoded address lines ADR2, ADR3 and ADR4. A 1-bit digital output also provides an indication that the circuit is working correctly by its output to LED D1 via resistor R4. It is important to note here that when chip select $\overline{CS1}$ and $\overline{CS3}$ are active, data transfer occurs over the data bus pins D0 to D7. When $\overline{CS2}$ and $\overline{CS4}$ are active, data transfer occurs over D8 to D15. When a 16-bit transfer occurs, $\overline{CS1}$, $\overline{CS3}$ and $\overline{CS2}$, $\overline{CS4}$ are active.

The programming data for the GAL is freely available and is shown in **Figure 5.** The GAL listing is also given in one of the files that can be downloaded from the Free Downloads section of the *Elektor Electronics* website at http://www.elektor-electronics.co.uk. Using this information it is possible to change

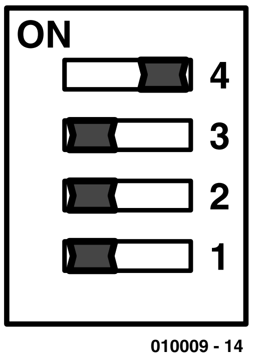


Figure 4. Selecting the Product ID with the DIL switch (example shown 17 hex).

the chip selects or generate new I/O signals for your own applications or special processor control signals.

The programming tools for the Lattice GAL are available from the Internet. If you intend to redesign the address decoder, it should be noted that half of the 256 addresses are used because we only use 16 of the 32 bits available. All addresses with address bit A1 set to a ‘1’ are invalid.

Additional features

In addition to the basic ICs necessary to perform the PCI interface, there are also some pre-wired IC positions on the PCB to accept additional ICs that may be useful for prototyping.

Components R1, R2 and C6, C7 can be fitted to build a reset circuit in one of two variants: either active-Low or active-High reset. Fitting R1, R2 and C7 will produce an active-High reset signal at the RST-OUT pin while fitting R1,R2 and C6 will produce an active-Low reset signal. In this case, R2 is 1 kΩ and R2, 22 kΩ. It should be noted that either C6 or C7 should be fitted but not both.

The position for IC3 will accept a clock generator chip and that for IC4, a DC/DC converter type NMH 0512S or NMH 0515S for generating either ±12 V or ±15 V respectively. These supply voltages are the most useful when prototyping with analogue devices.

Capacitors C10 to C14 and inductors L1 and L2 will filter out any nasty supply spikes.

Placing jumpers across the pins of K7 will allow the ‘In Circuit Programming’ mode to be enabled of the LSI IC1. This is primarily intended as an aid for the development of the LSI. Those of you who are not familiar with these features of this LSI are advised to leave K7 unused, otherwise you run the risk that the PCI card may no longer function.

Building your own circuits

When constructing your own circuits on the breadboard area of the card it is possible to use IC’s from the 74ALS, 74ACT, 74AHC or 74F families. The 74HCT family of devices has a switching speed that is just on the limit of acceptability when used on the PCI card, and reliable operation of the hardware cannot be guaranteed. A test circuit was built using HCT devices in the *Elektor Electronics* laboratory, and when these chips were swapped with HCT devices from a different manufacturer, the circuit failed to work. In the interests of reliability, it is wise not to include HCT devices in your designs built on this prototyping card.

If you are interfacing to any of the PCI bus signals, it is important to note that the specification for this bus allows a maximum of one TTL load on each of the PCI bus signals with a maximum track capacitance of 10 pF. This means that apart from the PCI decoder chip it is not possible to make any further connections to the PCI bus connector. If the PCI decoder chip IC1 is used in another development circuit, it is vital that the track layout to this chip is the same as in this PCI prototyping card. Changes to the track layout or track length can cause timing problems in the PCI decoder chip. The 16-bit data bus is provided for circuit expansion.

The Software Driver

The driver for the PCI prototyping card can be downloaded from the Free Downloads section, February 2001 items, on the *Elektor Electronics* internet site http://www.elektor-electronics.co.uk. Along with the drivers are also example programs together with their source code running on DOS, Windows 9x, Windows 2000/NT and Linux. This software is also available on a CD-ROM called ‘PC Card Software CD’ from Kolter Electronics. Also included on the CD are shareware programs, demos, datasheets, PCI card documentation and useful Internet links.

Table 1

Outputs of the address decoder implemented in GAL IC2

CS1	Base + 0
CS2	Base + 1
CS3	Base + 4
CS4	Base + 5

Use with DOS

For operation of the PCI prototype card on a computer using DOS, it is not necessary to add any drivers in the CONFIG.SYS or AUTOEXEC.BAT files. DOS software can directly access the hardware using the assembler instructions IN and OUT.

PCIVIEW

This DOS program displays all the installed PCI bus components together with information showing each Vendor ID and Device ID. It also shows (in as far as possible) the manufacturer’s name and a description of all devices used on the card. A more important feature of this program is that is shows the base address of the card connected to the PCI bus. This information allows you to determine the I/O address of the card and use this value in your own software to read and write to the PCI card.

One important point to watch here is that the address allocation of the card may be altered automatically if there is a change in the PCI system, e.g., installing new components or removal of hardware. In that case, you will also need to change the corresponding address in your software.

Using the PCI card base address in your program is the most direct method of programming the card, although it means that the resulting program will not be portable, that is, it will only run in this one environment and not on another PC or if changes are made to the hardware configuration. To make the program portable it is necessary to make use of the PCI functions of the PC BIOS because DOS does not have PCI Applications Programming Interface, API.

PCI_ADR

This program is similar to PCIVIEW

but only shows information of PCI devices with the Kolter Electronics vendor ID (including this PCI prototyping card).

Use with Windows 95/98/NT

The installation of the PCI prototyping card is just as simple as with any other PCI cards under Windows 95/98. After fitting the card in the PC it will be recognised by Windows when the PC is turned on. You will be presented with the message that the new PCI card has been detected and then asked which driver to install. You now need to refer to the file `KOLTER.INF` (see the chapter on drivers). Windows now looks for the `KLIBNDRV.VXD` file and installs the driver. When the computer is restarted the PCI prototyping card will be installed.

HWT

The Hardware Test Program HWT is a PCI diagnosis program running under windows. Besides the manufacturer's vendor ID and the equipment device ID it also shows the I/O base address of the card. The displayed base address has been incremented by one so that for example a address shown as E001 is actually E000. The address space of the PCI card extends from this base address over a 256-byte area. In our example, the actual address range would be from E000 to E0FF. Using the base address, it is possible to access the ports using 8- or 16-bit data width.

The functions Read*106 and Write*106 are included and can be used to determine the maximum data flow through the port of your computer. Clicking on this function will perform 1000000 Port cycles. Measuring the time taken will give the throughput of your system. The hardware table is a standard text file and can be easily expanded to your requirements.

With Windows 95/98/ME you don't need any further registration to use HWT — all the data files can be stored in any subdirectory. If you are using Windows NT4/2000 then it is necessary to copy SYS data into the directory c:\Winnt\system32\drivers and then re-boot the system. HWT can then be launched from Explorer.

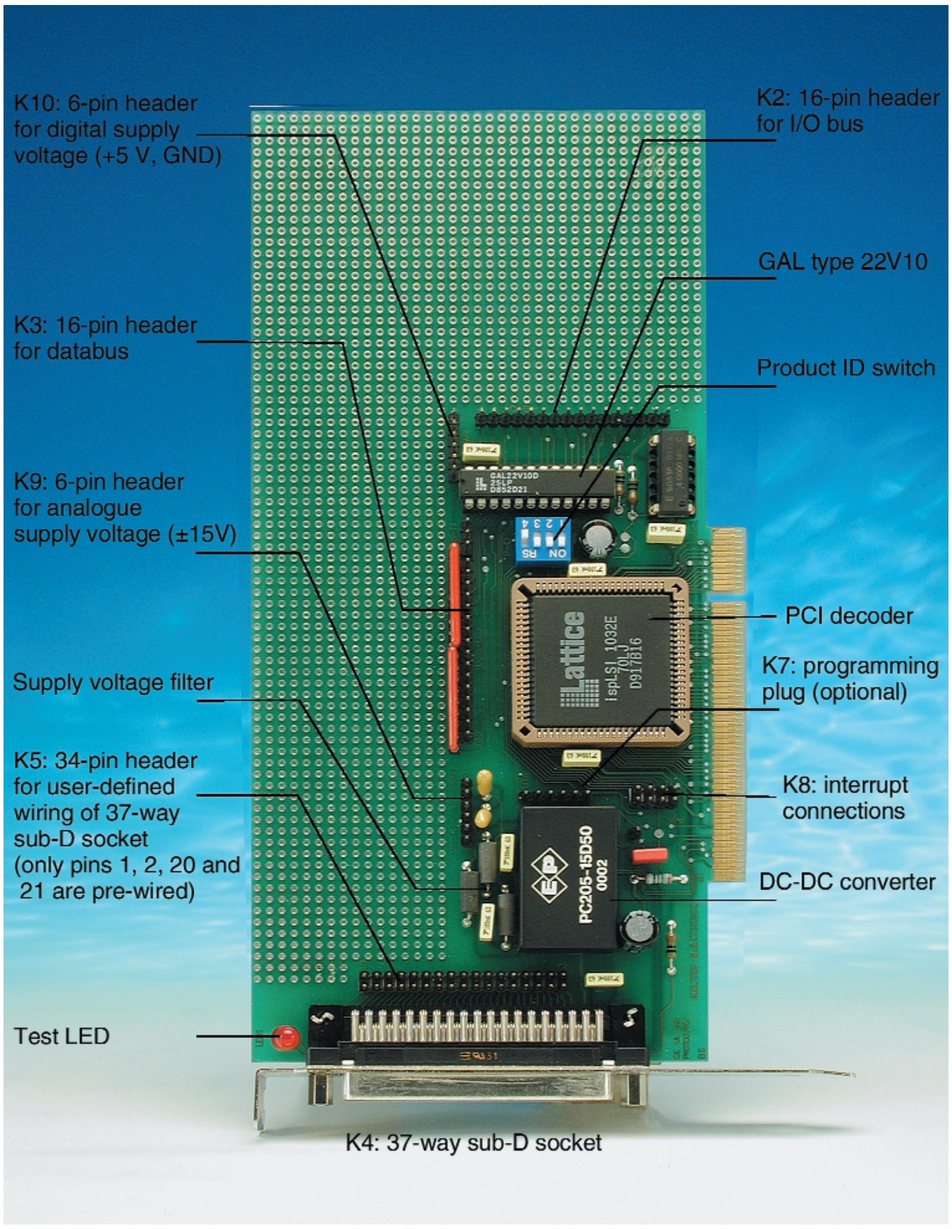

K10: 6-pin header for digital supply voltage (+5 V, GND)
K3: 16-pin header for databus
K9: 6-pin header for analogue supply voltage (±15V)
Supply voltage filter
K5: 34-pin header for user-defined wiring of 37-way sub-D socket (only pins 1, 2, 20 and 21 are pre-wired)
Test LED
K4: 37-way sub-D socket
K2: 16-pin header for I/O bus
GAL type 22V10
Product ID switch
PCI decoder
K7: programming plug (optional)
K8: interrupt connections
DC-DC converter

Figure 5. Overview of the pinheaders and main elements on the board.

Example Program

This program, complete with the Delphi source code, allows you to rapidly test your own circuitry in the breadboard area of the PCI prototyping card. It also allows you to see if the PCI card itself is functioning correctly by flashing the test LED on the card.

An NT driver is available for Windows 2000/NT. Details are available at the web address http://www.pci-card.com. (see web links at the end of this article).

Use with Linux

Users of the Linux operating system will find it generally very simple to program hardware cards such as I/O cards, A/D and D/A converters using the GNU-C compiler. Comparing it with the MS operating system you will find that there are no hidden access mechanisms or tricks that you normally find necessary with MS operating system programming. The Linux operating system and Linux application software is openly available and more often than not free. Programming in Linux is so much simpler that anyone familiar with ANSI-C should have no problems. The instructions `ioperm` and `iopl` give the program permission to access the I/O ports without the need for any additional device drivers or the need for any extra cryptic `include` statements in the compilation process. The programming process is similar to ANSI-C running in DOS but instead of only an 8-bit wide data access you can also use 16 or 32 bit. To actually move data through the ports, instructions such as `outb(value,address)` and `inb(address)` are contained in the directory /usr/include/asm/io.h. These routines are all inline macros so it is only necessary to `#include<asm/io.h>`. No additional libraries are required. Compilation of the program can be performed with XWPE under KDE or with gcc from the Linux 'bash' shell. Other compilers can also be used to compile the code and they generally seem to suffer fewer problems than when under other operating systems. Extensive information on the use of the Linux operating system is available as a download from the *Elektor Electronics* website.

Data transfer, timing and addressing

The second instalment of this article will explore in detail the high speed data transfer, timing and addressing of the PCI bus architecture, it will also look at the BIOS configuration and the PCI bus connector pin assignments in 5 V and 3.3 V systems.

(010009-1)

Suggested literature:
PCI Bus Demystified, by Doug Abbott, LLH Technology Publishing, ISBN 1-878707-54-x.

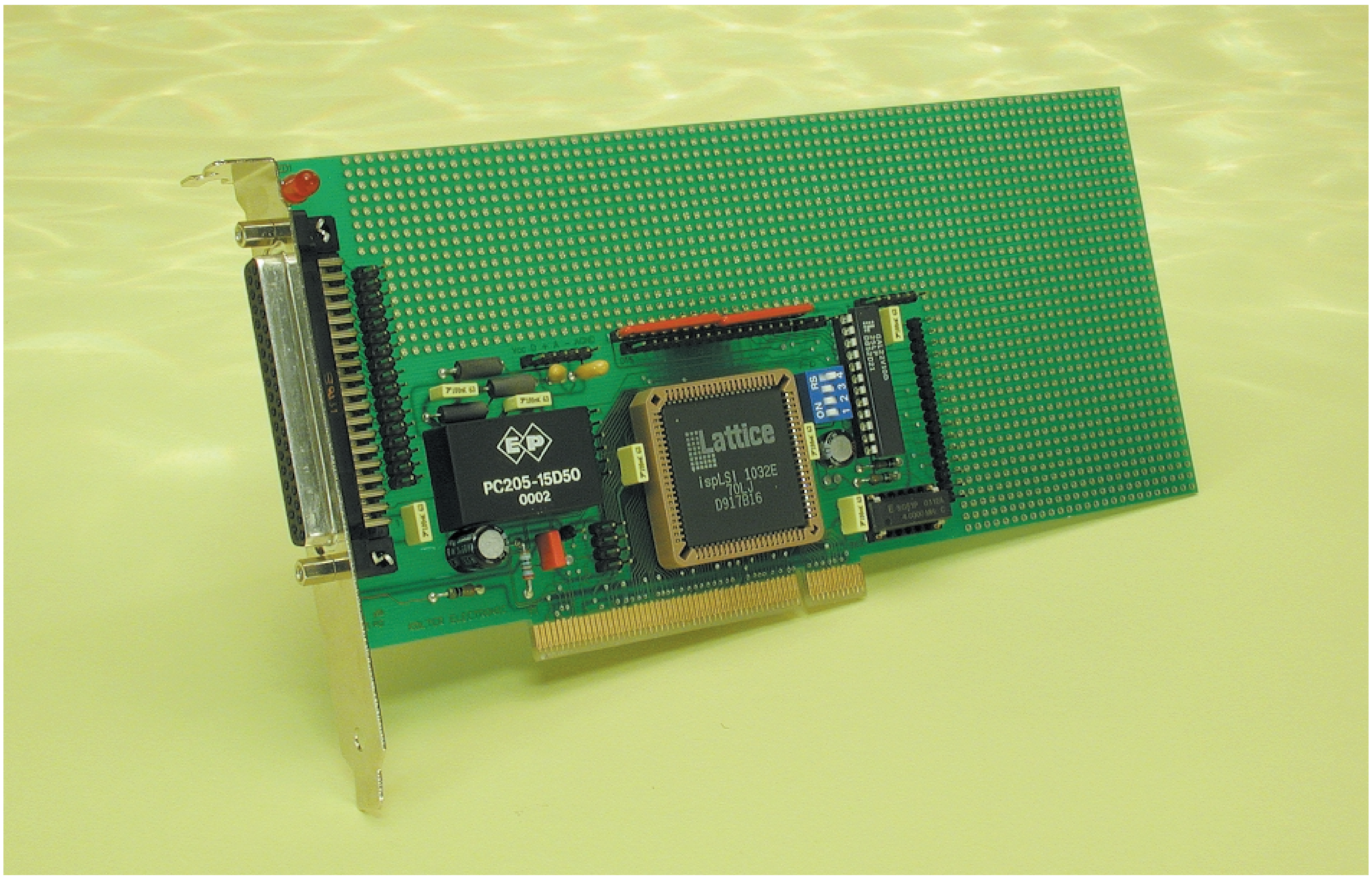
Supplier reference:

Kolter Electronic,
PO Box 1127,
D-50362 Erftstadt,
Germany.
Tel. (+49) 2235 76707,
Fax (+49) 2235 72048.

Web Links

Links for the PCI prototyping card and PCI decoder:

http://www.pci-card.com/pcintern.htm
http://www.pci-card.com/neuep.htm#proto3
http://www.pci-card.com/Prototim.pdf
http://www.pci-card.com/pcideko.pdf
http://www.pci-card.com/pcideko.html
http://www.pci-card.com/pciversion.html
http://www.pci-card.com/pciboards.html
http://www.pci-card.com/Hwt.zip
http://www.pci-card.com/pci_pins.htm
http://www.pci-card.com/pro3pci.zip



MCS BASIC–52 V1.3

rejuvenating a popular interpreter

By H.-J. Böhling and D. Wulf

The MCS BASIC–52 interpreter, developed by Intel, is still very popular, but in its 15th year it needs a thoroughgoing facelift.

The MCS BASIC–52 V1.0 interpreter was specially developed by Intel in 1985 for the 8052-AH microcontroller. With its code size of only 8 kB, it fitted into the internal ROM storage of this controller. Unfortunately, the 8052-AH has gone out of production. Nonetheless, the interpreter still enjoys considerable popularity in its final 'official' version (1.1), in spite of a few errors. Nowadays, the 8052 family, with its many derivative versions, has become the most widely used microcontroller family. Regrettably, MCS BASIC 1.1 will not run directly on the fastest members of this family. The authors have made numerous modifications to the source code to arrive at version 1.3, with the following results:

- The interpreter no longer has any known errors.
- BASIC–52 V1.3 can program EEPROMs, even if they are located in external memory.
- The new command 'ERASE' erases an EEPROM.
- The maximum value for 'XTAL' can now be set as high as 78 MHz, so the interpreter can even run on a Dallas 80C320 at 33 MHz!
- A new reset routine recognises the baud rate and various types of controllers. This means that BASIC–52 V1.3 can run on many 8052-family members (80616, 517(A), 528, 535, 537, 575 etc.).
- The new OPCODE 43h reads a value from the argument stack into a variable.
- The size of the interpreter is still only 8 kB, which means that it will fit into an 87C52 with 8 kB of internal EPROM.

Intel has now released the interpreter as freeware. It can thus be legally obtained via the Internet as commented source code. In spite of the comments, the source code is not easy to read, so the authors were forced to use a simulator to unravel the functions of the interpreter. However, this need not concern us any further here.

Tailor-made improvements

The authors' work started with an extensive Internet search for proposed improvements and error descriptions related to MCS BASIC–52. Unfortunately, the available source code was not relocatable, which meant that its instructions could not be freely moved around. The insertion of even a few extra bytes of code immediately resulted in incorrect operation. This was primarily due to the fact that the source code was contained in two separate files, one for the interpreter itself and another for floating-point calculations. Consequently, the first task was to merge the two files into a single, common file. Labels were also added to all unlabelled absolute and relative jumps and calls. Some parts of the source code were coded as literal data, because the assembler used at the time did not support the new 8052 instructions. The literal data were translated into equivalent mnemonics. One routine in the source code differed from the original Intel ROM code, so it was replaced by the mnemonics corresponding to the ROM code. On completion of these modifications, the code was relocatable, so a number of ORG directives, in particular those linking the interpreter and floating-point code, could be eliminated. However, the primary benefit was that memory space gained by improvements to the code could now be used for further error corrections (see the October 1991 issue of *Elektor Electronics*). Still, the amount of recovered space was insufficient to allow all error corrections and improvements to be fitted into the code. Consequently, the 'ego' message of the Intel programmer J. Katausky had to be deleted (sorry!). Bit 40h in the internal bit-addressable memory, which was made free as a result, was used to detect a multiplication underrun. Previously, this had not been handled properly.

As early as 1993, D. Karmann rewrote the interpreter as version 1.2 for an 8031 controller, and in the process he eliminated two known errors, so that:

– a BASIC Autostart EPROM could work together with command extensions, and
– variable names with 'F' could be used without any restrictions.

Dan Karmann was kind enough to provide the authors with the source code for his improvements.

Running BASIC–52 V1.3 on Elektor Electronics boards

The hardware requirements for running BASIC–52 on the 80C537 or '537-Lite' boards are fully described in the article 'Basic-537', which appeared in the February 2000 issue of *Elektor Electronics*. The information in that article is equally applicable to V1.3, with the following exceptions:

– It is no longer necessary to manually set the value of MTOP, since V1.3 automatically detects the amount of available RAM.
– BASIC–52 V1.3 automatically recognises the baud rate being used, so a space character must be sent to the board after a restart!

The 80C32 Control Computer described in the February and March 1998 issues of *Elektor Electronics* can be used with BASIC–52 V1.3, including using a Dallas 80C320 'Speed it μP'. The Dallas controller is software and hardware compatible with the 8052. However, the processor design is completely new, and it takes only four clock cycles to execute an instruction instead of twelve. It can be run at clock frequencies up to 33 MHz. To revise the board for this controller, you can replace the quartz crystal with a 22.118400 MHz crystal, and the resulting processing speed will be approximately six times greater than that of a standard 80C32. The controller can handle 33 MHz, but this is not feasible here, in part due to the lack of a sufficiently fast EPROM (which would have to have an access time of around 40 ns) and in part because a suitable crystal is not available. For technical reasons, fundamental-mode crystals are only made for frequencies up to around 22 MHz. Above this frequency, overtone crystals are used, with a supplementary resonant network to force the crystal to oscillate at a harmonic frequency. Even at 22 MHz, sufficiently fast program memory (70 ns) must be used for the interpreter. The data memory can be populated with devices a having a standard access time of 150 ns, since the controller automatically inserts wait states when accessing data memory. The two TTL ICs (74HC573 and 74HC00) should be replaced by fast 74AC or 74F types. A 28C64 can be used for the EEPROM (IC5 on the circuit board). However, you must first cut the connections between +5 V and pins 1 and 27 of the socket for the EEPROM. Pin 1 may be left open, but pin 27 must be connected to the /WR signal from the controller (pin 27 of IC3).

Terminal MCS–51

If you order BASIC–52 V1.3 from the Publishers' Readers Services on diskette (order number **000121-11**), you will also receive a full edition of the latest version of Terminal MCS–51. This is a terminal emulator program for DOS computers, which naturally can also be run in a DOS window under Windows. Terminal MCS–51 was written to compensate for the known weaknesses of BASIC–52. For example, BASIC–52 does not have a RENUMBER command for renumbering BASIC command lines, and it has only very limited functions for editing BASIC programs. The functions provided by Terminal MCS–51 include:

– easy loading and storing of BASIC–52 programs,
– renumbering BASIC–52 programs,
– integrated line editor for BASIC–52 command lines,
– any desired ASCII editor can be linked in.

A shareware version of Terminal MCS–51 can also be downloaded from the Free Downloads section on the *Elektor Electronics* website at http://www.elektor-electronics.co.uk. The shareware version is fully functional and not time-limited, but it does not allow BASIC program lines to be renumbered.

```
4    REM *******************************************************
5    REM * Program for controlling A/D converter in 80535/537 *
6    REM *******************************************************
10   RANGE=5 :                   REM Controller reference voltage
20   ADCO=0D8H :                 REM Register to check A/D state etc.
30   ADDAT=0D9H :                REM Register for measurement output
40   DAPR=0DAH :                 REM Register for meas. range and start
50   RDSFR (ADCO)BUSY :          REM Read other function bits
60   BUSY=BUSY.AND.0C0H :        REM Isolate function bits
70   WRSFR (ADCO)BUSY :          REM Pick measurement input 0 (Port 7.0)
80   WRSFR (DAPR)0 :             REM Start measurement in highest meas. range
90   RDSFR (ADCO)BUSY :          REM Read A/D converter state
10   IF (BUSY.AND.020H)>0 THEN GOTO 90 :         REM wait if busy
110  RDSFR (ADDAT)VOLTAGE :      REM Read measurement value
120  PRINT "Voltage on Input 0 equals:",
130  PRINT VOLTAGE*RANGE/256,"Volt" :    REM Adapt value to meas. range
140  PRINT "Another measurement required in 1-V range (Y/N)?",
150  I=GET.AND.0DFH
160  IF I=ASC(Y) THEN GOTO 190
170  IF I=ASC(N) THEN GOTO 260
180  GOTO 150
190  RANGE=1 :                   REM Measurement range = 1 Volt
200  WRSFR (DAPR)040H :          REM Set range and launch measurement
210  RDSFR (ADCO)BUSY :          REM Read A/D converter state
220  IF (BUSY.AND.020H)>0 THEN GOTO 210 :        REM Wait if busy
230  RDSFR (ADDAT)VOLTAGE :      REM Read measurement value
240  PRINT "Voltage on Input 0 equals exactly:",
250  PRINT VOLTAGE*RANGE/256,"Volt" :    REM Adapt value to meas. range
260  END
```

The A/D converter in the 80517A has a resolution of 10 bits and increased measurement accuracy. However, with this type of controller it is not possible to modify the internal reference voltage via a control register. The following sample BASIC program shows how the input can be selected for the signal to be measured.

```
4    REM ****************************************************
5    REM * Program for controlling A/D converter in 80517A *
6    REM ****************************************************
10   RANGE=5 :           REM Controller reference voltage
20   ADCO=0D8H :         REM Register for measurement input selection
30   ADDATL=0DAH :               REM measurement value, LSB
40   ADDATH=0D9H :               REM measurement value, MSB
50   PRINT "Please enter number of measurement input (0-7):",
60   INPUT CHANNEL
70   CHANNEL=CHANNEL.AND.7 :     REM Mask superfluous bits
80   RDSFR (ADCO)BUSY :          REM Read function bits
90   BUSY=(BUSY.AND.0C0H).OR.CHANNEL :   REM add channel selection
100  NRSFR (ADCO)BUSY :          REM Copy channel selection to controller
110  WRSFR (ADDATL)0 :           REM Launch measurement
120  RDSFR (ADCO)BUSY :          REM Monitor A/D Busy state
130  IF (BUSY.AND.020H)>0 THEN GOTO 120 :        REM Wait if busy
140  RDSFR (ADDATL)UOLTPART :    REM Read least significant part
150  RDSFR (ADDATH)VOLTAGE :     REM Read most significant part
160  VOLTAGE=VOLTAGE+UOLTPART/256 :      REM Compute maesurement values
170  PRINT "Voltage at input:",CHANNEL,"equals:",
180  PRINT VOLTAGE*RANGE/256,"Volt" :    REM Adapt value to meas. range
190  END
```

EPROM programming

Unfortunately, MCS BASIC–52 V1.1 cannot program an EPROM when it is run from external memory. The improvement made to remedy this shortcoming is based on a suggestion made by R. Skowronek in 1996. BASIC–52 V1.3 can directly program hardware EEPROMs using the 'PROG(1–6)' and 'PGM' commands, even when it is run from external program memory. The EEPROM is written in the same way as RAM, but with a suitably longer cycle time. With this change, port pins 1.3 ($\overline{\text{ALEDIS}}$), 1.4 ($\overline{\text{PRGPLS}}$) and 1.5 ($\overline{\text{PRGEN}}$), as well as bit 15h (INTELB) in the internal memory and memory locations 12Ah and 12Bh in external RAM, are no longer needed and thus can be freely utilised for other purposes. The 'PROG' commands are also redundant, so they have been deleted. This yields additional space savings, and the BASIC token 0F9h that has been made free is used for the new 'ERASE' command. This command erases an EEPROM by overwriting all memory locations between 08000h and 0C000h with 0FFh. 'ERASE' is called in command mode, and it takes around 2.75 minutes to erase this 16-kB region.

A special feature of BASIC–52 is the possibility of extending the command set using up to 16 user-defined commands programmed in assembly language. 'OPCODEs' are provided in order to allow system routines to be used in these extensions. Unfortunately, there was no OPCODE available to assign a value placed on the argument stack to a variable defined in BASIC text. Consequently, it was always necessary resort to awkward routines using the POP command. In BASIC–52 V1.3, OPCODE 43h has been defined for this purpose. It is used, for example, in the command extension 'RDSFR (address) variable'.

Previously, the instruction 'TIME = 0' did not reset the millisecond counter to zero, so that the value of TIME was not reset to exactly zero. This unfortunate error has been corrected in V1.3.

The BASIC–52 instruction 'ASC(x)' always returned an incorrect value for any character position occupied by a BASIC operator (*, +, –, /, <, =, > or

?). The interpreter translated these characters into equivalent tokens during the tokenisation process. For example, a question mark was translated into the token for 'PRINT'. Consequently, the value of the token was returned instead of the correctly computed ASCII value of the character. BASIC–52 V1.3 now supplies correct values for all ASCII characters.

In order to allow BASIC–52 V1.3 to run on as many 8052-family derivatives as possible, and in particular on the Dallas Semiconductor 80C320 'Speed it µP' controller, the reset routine has been extensively reworked, as follows:

– A new baud rate detector utilises Timer 2 instead of 'code loop' timing to measure the utilised baud rate. This makes the measurement fully independent of the controller type and clock frequency.
– Timer 0, which is used for time measurements (BASIC Software Clock), is now operated in the 16-bit mode instead of the 13-bit mode that was previously used. This means that the maximum value for XTAL is now 78,627,473 Hz. The power-up default value is still 11,059,200 Hz, as before, but this value can be modified as necessary using 'XTAL = value'. Alternatively, the default value can be changed by patching the following locations in the EPROM with suitable new values:

17F1h	=	11
17F0h	=	05
17EFh	=	92
17EEh	=	00

For baud rate detection, the user still has to send a space character to BASIC–52 via the serial interface. The character length is then measured by the detection routine using Timer 2. In principle, BASIC–52 utilises only Timer 2 as the baud rate generator. This applies to 8052 controllers and the Dallas 80C320. The 80535, 80515 and 80517 controllers cannot activate Timer 2 as a baud rate generator. They have a special baud rate generator that is fully independent of Timer 2, and this allows only two different baud rates to be generated with a specific relationship to the processor clock frequency. If the clock frequency is exactly 12 MHz, the available rates are 4800 and 9600 baud. If the BASIC–52 V1.3 set-up routine detects this type of controller, it activates the baud rate generator with a baud rate that corresponds to that of the measured timing of the space character (4800 or 9600 baud).

The 80C517A controller, which is also frequently used, has an enhanced special baud rate generator. This supports arbitrary data transmission rates, but it still operates independently of Timer 2. BASIC–52 V1.3 also automatically recognises this processor type and activates it with the appropriate baud rate. With all of these special types of controller, Timer 2 is freely available to the user.

I^{2}C and SFR

BASIC–52 can easily be extended in assembly language with additional BASIC commands. Naturally, these extended commands cannot be located within the 8-kB code region of the BASIC interpreter. The authors have implemented six new commands. Four of these are related to I^{2}C communications and have already been described in the article "Implementing the I^{2}C Bus" in the July/August 2000 issue of *Elektor Electronics*, as follows:

I2CSTART
Transmits a Start condition on the I2C bus.

I2CSTOP
Transmits a Stop condition on the I2C bus.

I2CPUT (value)
Transmits a byte on the I2C bus.

I2CGET (value)
Reads a byte from the I2C bus into a variable.

Two additional commands allow Special Function Registers (SFRs) to be read and written. The various derivative versions in the controller family employ supplementary SFRs. The following commands have been implemented to allow these registers to be accessed via BASIC:

WRSFR (address) value
Writes 'value' to the indicated SFR address.

RDSFR (address) variable
Reads the content of the indicated SFR address and writes it to 'variable'.

Using these commands, it is easy to address internal extensions of the various processor derivatives, such as the additional I/O ports of an A/D converter. The two sample BASIC programs in the text boxes (AD-535.LIS and AD-517A.LIS) illustrate how the SFR commands can be used.

Since BASIC–52 V1.3 also runs on the 80535, 537 and 517 controllers, as well as other types, and since the external command extensions allow extended SFRs to be used without any problems to access new hardware features, it is now easy to write programmable control and adjustment programs in BASIC, instead of wrestling with the intricacies of assembly language programming. The sample programs illustrate the use of the A/D converter of an 80535 and an 80517(A). The A/D converter in an 80535/537 controller has a resolution of 8 bits. The external reference voltage for the converter is normally set to +5 V. This yields a minimum resolution of 20 mV. Since the internal reference voltage of the converter can be programmed using the 'DAPR' SFR, it is possible to achieve a higher resolution (for example, 4 mV) by making a second measurement using a smaller measurement range. The SFRs listed in **Table 1** can be used for this purpose.

(000121-1)

Table I. 80535/537 A/D converter SFRs

SFR	Address	Bit(s)	Name(s)	Function
ADCON	0D8h	2–0	MX2,MX1,MXO	Select analogue input
ADCON	0D8h	3	ADEX	Start via hardware/software
ADCON	0D8h	4	ADM	Single/continuous conversion
ADCON	0D8h	5	BSY	Conversion in progress
DAPR	0DAh	3–0	DAPR.3-.0	Lower reference voltage
DAPR	0DAh	7–4	DAPR.7-.4	Upper reference voltage
ADDAT	0D9h	7–0		8-bit measurement result

Microcontrollers

the latest developments

By Guy Raedersdorf

It's already been around 18 months since we last looked at the state of affairs in the microcontroller world (February 1999). Time doesn't stand still, especially in this area, so we thought it was again time to make an up-to-date 'family photo'.

What we want to do here is to peek into the kitchens of the most important manufacturers, in order to see if they have anything new. We conclude the article with some general observations regarding the most important developments, since these naturally have an effect on the course of hobby electronics.

Atmel

www.atmel.com

Atmel are continuing on the course that they set with their successful **AT90SXXX** product, in this case with the ATiny series. This consists of the ATiny 12, 15, 22 and 28. The **ATiny 22**, for example, has 2 kB of flash memory, 128 bytes of RAM and 128 bytes of EEPROM on board.

The **AT43320** and **AT43USB321** can be mentioned as the most recent products. These are microcontroller-based USB hubs.

The **AT8X** series includes the **AT89S8252** (an 80C32 with 8 kB of flash memory and 2 kB of EEPROM) and the AT89S4D12 (an 80C31 with 128 kB of flash memory). The PS version of the former type is an ISP (in-system programmable) part. The clock frequencies, like the memory sizes, are being steadily increased, and they now lie around 33 MHz.

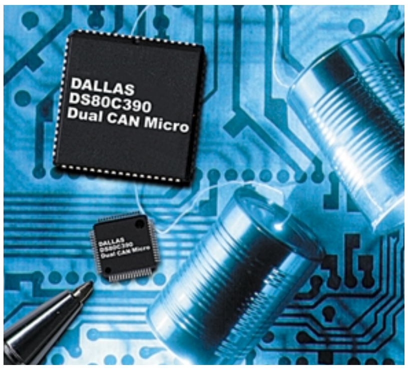
As regards 32-bit microcontrollers, Atmel has the **AT91 Thumb** family. The youngest scion of this clan is the **AT91F40416**, which is a combination of an AT91M40400 microcontroller and an AT49BV16x4 flash memory.

Dallas Semiconductor

www.dalsemi.com

With regard to microcontrollers, Dallas Semiconductor do not have much news to report.

The latest component is the **DS80C390**. This is a beefed-up 8051 with two integrated CAN controllers and extensive peripherals. It brushes aside the usual memory limitations, since it can address 4 MB of data memory and 4 MB of program memory (all external). With a clock frequency of 40 MHz, the **DS80C390** runs at a speed of around 120 MHz, which is ten times faster than the original design.



Another newcomer at Dallas Semiconductor is the **DS87C550**. This muscular derivative of the 8051 has built-in analogue/digital (A/D) converters, namely an 8-channel, 10-bit A/D converter and a 4-channel, 8-bit pulse-width modulator (PWM). Finally, we can briefly mention the **DS80C323**. This is a low-power version of the popular DS80C320. Running at 18 MHz, this controller delivers the performance of a 50-MHz 8051, while consuming no more than 10 mA.

Infineon (ex Siemens)

www.infineon.com

Infineon's product line includes several important microcontrollers, some of which were already mentioned in the previous article. The latest arrivals from Infineon are described below.

The **SAB C508** comes with 32 kB of OTP (One Time Programmable) ROM, 1280 bytes of RAM and a 10-bit A/D converter. This is an improved version of the very well known C504 microcontroller.

The **SAB C541U**, which is an 8-bit microcontroller in the C500 family, achieved a breakthrough in the microcontroller field by integrating a USB controller into the chip. It combines an SAB C511 with new extensions, such as an SPI-compatible interface and a USB module.

A newcomer in Infineon's 16-bit family is the **SAF C165UTAH**. This IC is a derivative of the Infineon C166, in which the fully static 16-bit core is used in combination with a USB module, four HDLC controllers, an IOM-2/PCM interface and 3 kB of on-chip dual-port RAM. The entire device works with 3.3 V, while the outputs are TTL compatible.

Next we have the **SAK-C167CR**. The C167CR and C167SR are the top-end products of the C166 family. This is a 16-bit CPU with a 4-stage pipeline, which works with a clock frequency of up to 33 MHz. It can handle 4 kB of on-chip SRAM, up to 128 kB of ROM, a version 2.0B CAN module (C167CR only), a four-channel MLI unit and a 16-channel, 10-bit A/D converter.

Another member of the C166 family is the **SAF C164CI-8EM**. This also has the features of the C167, but with only 64 kB of program memory and 3 kB of RAM. However, it does have an on-board CAN module.

Infineon describes the **SAF C161U** as 'the USB controller for the new millennium'. This inexpensive 16-bit digital workhorse has a C166 core combined with extensive on-chip peripherals, such as a UART, five timers and eight bi-directional, software configurable USB endpoints.

It appears that the idea of developing a single core that can be used for almost all applications is becoming an established technique in microcontroller design. This is how the **SAB TC10GP** came to be. It is the first 32-bit single-core digital signal processor (DSP). It is based on the TriCore-1, runs at 66 MHz (soon to be raised to 88 MHz) and has some interesting features, such as a configurable SRAM/cache memory. It can have from 0 to 8 kB of SRAM program memory with 8 to 16 kB of cache memory. For data, there is 32 to 16 kB of SRAM with 0 to 16 kB of cache memory. This microcontroller is blessed with three 8-bit parallel ports and can achieve 100 MIPS.

The TC1775 is a 32-bit microcontroller that forms the main stem of the AUDO microcontroller family, which has been developed by Infineon especially for automotive applications. A 32-bit TriCore chip is also used here, equipped with (among other things) a Peripheral Control Processor (PCP) and very flexible and powerful peripherals. It also incorporates two synchronisable 12-bit A/D converters and two CAN modules (Twin-CAN).

Infineon continue to be intensively active in the Smart Card area, and one of their latest products is the **SLE66CX640P**, the first Smart Card controller with a 64-kB EEPROM.

Intel

www.intel.com

Although Intel are no longer active in the area of classic microcontrollers and appears to prefer 'embedded' controllers, they have still brought out a very interesting and currently unique controller, the 8XC196, which works on the basis of fuzzy logic. This makes it a very contemporary design.

Microchip

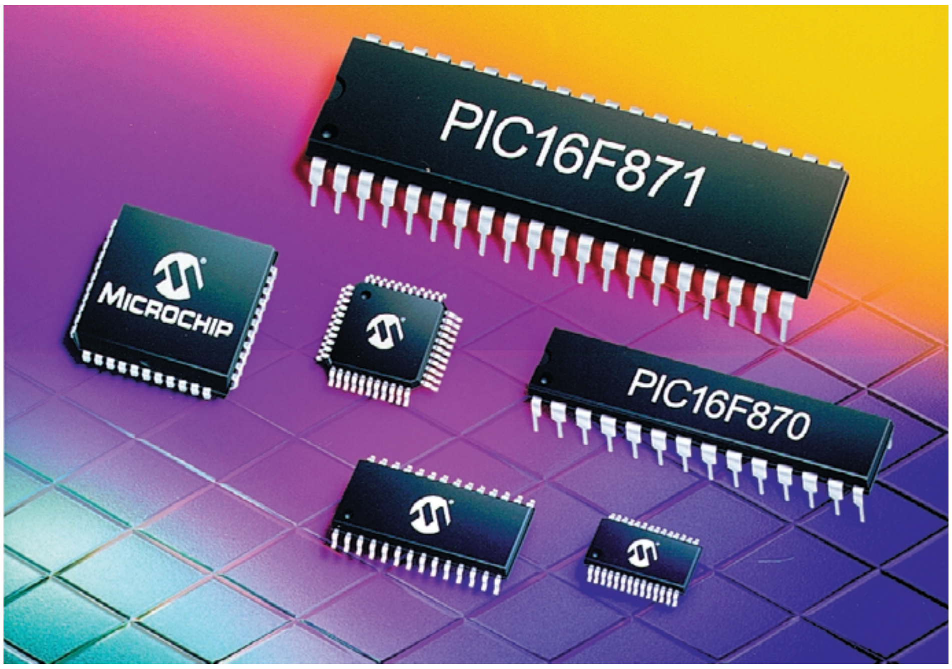
www.microchip.com

Without any doubt, Microchip are one of the most enterprising of the microcontroller manufacturers. They thus had no hesitation in bringing 37 (!) new models of flash controllers to the market in May 2000. Of course, we cannot discuss all of them, but in general they contain between 1 and 64 k words of flash program memory, up to 4 kB of data SRAM and 256 bytes of data EEPROM. They are housed in packages with 8 to 84 pins.

The most important new products from Microchip are the 16F87X family (based on flash technology), the **PIC16C7X5** family (including the **PIC16C745** and the **PIC17C765** with a USB interface), the PIC16C7XX family (the **PIC16C717, PIC16C770 and PIC16C771**), which can handle analogue signals, and the completely new 18C family, which is a series of powerful 8-bit microcontrollers.

PIC16F87X

This series of RISC controllers has only 35 instructions in its instruction set. The size of the flash program memory ranges from 4096



to 8192 14-bit words, and there are 192 to 368 bytes of RAM available, as well as 128 to 256 bytes of EEPROM. The processing power is 5 MIPS at 20 MHz. The peripherals consist of A/D converters, an SSP port with SPI and I^{2}C, as well as a Parallel Slave Port (PSP) and a USART.

PICmicroFLASH

The new **PIC18CXXX** employs an improved RISC kernel that is compatible with the architectures of three different families: PIC16C5X (12-bit), PIC12C508CXXX and PIC16XXX (14-bit) and PIC17CXXX (16-bit).

Up to now, the PIC18CXXX family consists of the **PIC18C242**, **C422, C252** and **C452**. The first two of these have 8 k words of 16-bit OTP program memory and 512 bytes of RAM. There are also versions with flash memory. The processing power is 10 MIPS.

The final item in the Microchip line-up is the **PIC12CR509A**, which is presently the smallest 8-bit ROM microcontroller. In spite of its dimensions, it has room for no less than 1 kB of program memory, 41 bytes of RAM and 6 I/O lines.

Motorola

www.mcu.motsps.com

In last year's article, we discussed the 68HC908GP20, and now we can introduce the **68HC908GP32**, which is a universal controller with ISP flash memory. The family has also been expanded with the **908JL3**, **JK3** and **JK1**, and – lest we forget – recently with the **68HC908JB8**.

Based on the 68HC08, this new flash controller with a USB or IBM/PS2 interface provides a flexible solution for PC peripheral devices, such as keyboards, mice, wireless HF receivers, USB converters, security measures for e-commerce and joysticks. Its 8-kB flash memory can be reprogrammed up to 10,000 times, and it includes 256 bytes of RAM, 37 I/O lines and a two-channel, 16-bit timer.

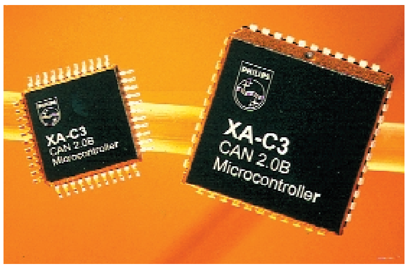
The real news is the **MMC 2107**. It is the first member of a new general-purpose microcontroller family, based on the M210 MicroRISC M-CORE. These new 32-bit integrated circuits with flash memory, which we are seeing increasingly often, include numerous digital and analogue peripherals, such as a powerful 16-bit clock, communications interfaces, an A/D converter with a resolution of 10 bits and 8 kB of RAM. The processing power is 31 Dhrystones or 2.1 MIPS at 33 MHz. The MMC 2107 has a 128-kB flash EEPROM and 8 kB of SRAM with battery backup for programs or data. Additional distinctive features are an interrupt controller with 40 programmable lines and an 8-bit SPI interface with error detection.

Philips Semiconductors

www-us.semiconductors.philips.com

Philips, one of the major players among the European component manufacturers, continue to expand their range of 16-bit XA controllers. If you have been keeping an eye on developments in this area, you surely will be aware of the XA-G3 controller family. The controllers in the XA (extended architecture) family are backwards compatible with the 80C51.

The **XA-G3** combines standard peripherals with the PeliCAN 2.0B bus. This 16-bit controller can address its program and data memory at the 24-bit level. It runs at 32 MHz and has an SPI. The memory consists of a 32-kB (EP)ROM and 1 kB of RAM. The PLCC or LQPF package has 44 pins. The controller runs at 32 MHz and has 42 interrupt vectors, in addition to 1 kB of SRAM for data. As with the other members of the family, 32 kB of EPROM/ROM is available for program memory.

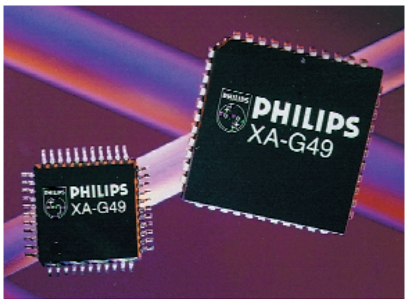


XA-S3

Another member of the XA family is the XA-S3. This controller runs at 30 MHz, can address 32 kB of ROM-less memory, and has 32 kB of EEPROM/ROM and 1 kB of RAM on chip. The XA-S3 is equipped with an I^{2}C interface. Note the presence of a 5-channel, 16-bit programmable counter array (PCA) and 50 I/O lines with four programmable output configurations. In addition, this device has two UARTs with independent baud rates and an 8-channel, 8-bit A/D converter.

The latest addition to the family is the **XA-G49**, a controller with IAP flash memory (more on this later on), which has the special feature that it offers an Internet interface via the iComponent from Connect One. The **XA-G49** has 64 kB of In-Application Programmable (IAP) flash memory, which allows the program code to be modified while the program is being executed. The combination of the XA-G49 and the 561AD-S/P iComponent from Connect One offers the

possibility of obtaining firmware updates via the Internet in a simple, fast and inexpensive manner. This is an application that we will certainly hear more about in the future. This chip also has 2 kB of RAM.
For more information about iChip, you can have a look at the following Internet address:
www.connectone.com



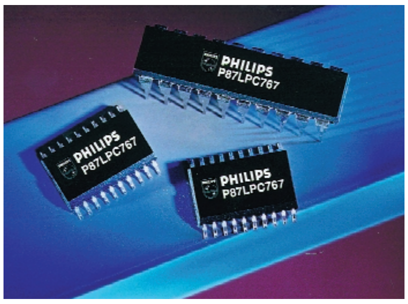
P8xC591
Here again we have an 80C51 single-chip controller with a CAN bus that complies with the CAN 2.0B standard. The powerful instruction set of the 80C51 is here combined with the PeliCAN functionality of the SJA1000, a CAN controller that has been used in several *Elektor Electronics* designs.

P89C51Rx2
The controllers in the P89C51Rx2 series satisfy the very latest demands. They are naturally compatible with the 80C51, have up to 64 kB of flash program memory and 1 kB of RAM for data (P89C51RD2); the B2 and C2 versions have respectively 16 kB and 32 kB of flash memory and 512 bytes of RAM. The flash memory can be programmed in parallel as well as by means of in-system programming (ISP). In-application programming (IAP) is also possible. Since the Rx2 has a boot ROM, the flash memory can even be modified while an application is running.

Philips have recently brought out a new controller family whose name will no doubt call up memories, namely the **NEW!80C51+**. These newcomers are compatible with their 80C51 predecessors, but their power consumption is a factor of two lower, they work at significantly higher clock rates, they have a greater supply voltage range and they largely satisfy EMC requirements, which nowadays is a highly valued feature.

51LPC family
The latest offspring of this family are the **P87LPC768**, **67**, **64** and **62**. This family, based on the 80C51 kernel, is target for applications in which low cost is the most important criterion. A number of modern touches are present, such as brownout detection, analogue functions and an integrated RC oscillator, so that the number of external components can be considerably reduced.

The **P87LPC768** has a 4-kB OTP memory, a 4-channel, 8-bit A/D converter and a PWM. These features also apply to the **P87LPC767**. The **P87LPC762** has only 2 kB of OTP memory, but it has 128 bytes of SRAM. The P87LPC764 is the twin brother of the former controller, but it has 4 kB of OTP memory. Running at 20 MHz, it is just as powerful as a C51 running at 40 MHz. A full-duplex UART is integrated into each of these components.



Scenix

www.scenix.com
Scenix are known for their SX series of controllers. They combine a pseudo-RISC architecture with on-chip data and program memory, which allows instructions to be processed in a single cycle. In combination with a deterministically programmable architecture, this allows the functions of certain real-time hardware to be replaced by program modules (virtual peripherals).

SX18/20/28AC
Scenix have plunged into communications and offers the SX18/20/28AC for such applications. These devices are distinguished by their high clock rates, which range up to 50 MHz for some types.

SX52/42BD
While the SX18/20/28AC can deliver a performance of 50/75 MIPS, the SX52DB achieves 100 MIPS. However, the surprising news from Scenix is the change of course they have made in the direction of communications. We are curious to see what the future will bring.

ST Microelectronics

www.st.com
ST Microelectronics are well known for having one of the most extensive controller families to be found, ranging from the inexpensive ST62 series to 32-bit and 64-bit cores, which are intended to be used in embedded systems. ST are one of the three most important producers of 8-bit controllers.
ST's primary product line consists of the 8-bit ST family and the 8/16-bit ST9 family

ST6
The ST6 series includes the ST62 and ST63 families. These are 8-bit controllers. The ST62 family can be divided into the **ST620X**, **ST622X**, **ST623X**, **ST624X**, **ST625X**, **ST626X** and **ST628X** series. The ST63 family has no less than 17 different members. The interesting feature of this family is the presence of an EEPROM. However, these families have not been particularly successful. This is partly due to the fact that ST have chosen to give more attention to the ST7 family.

ST7
The **ST72XXX** family offers many choices with regard to on-chip memory, ranging from 4 kB of program memory and 128 bytes of RAM (in the ST72101) to 32 kB of program memory and 1 kB of RAM plus 256 bytes of EEPROM (in the ST7231). Program memory was originally provided in the form of ROM, OTP and EPROM, but there are presently no fewer than 27 members of the family with flash memory. The different family members

have various features, including a multi-channel, 8-bit A/D converter and SPI, I^{2}C, USB and CAN interfaces. An interesting example of the ST72 family is the **ST72171**, which is a controller with analogue functions that allows the gain to be programmed. This is the first controller to be equipped with a software programmable gain amplifier (SPGA). In addition, the ST72171 has flash memory and powerful analogue and digital functions.
Another new and interesting family is the **ST7263**, which has a low-speed USB interface. The chip with an 8-bit CAN bus has 4 to 16 kB of ROM/OTP memory. One of the latest additions is the **ST72141**, which is optimised for driving DC and induction motors.

ST9
The motto of the ST9 family could well be '16-bit performance at an 8-bit price'. The newest versions include the **ST92F120 family, which has on-chip flash memory and RAM. The ST9 family is presently fabricated in 0.5-μm technology, but the 0.35-μm boundary should be crossed in the course of this year.**

The final family that has been announced is the **ST92163** family. It has an integrated high-speed USB interface, which makes it suitable for MP3 applications, among others. It has 20 kB of program memory and 2 kB of RAM.

Temic Semiconductor

http://www.temic-semi.com
The latest scion of the Temic clan is the **T89C51RD2**, an 8-bit controller with 64 kB of ISP flash memory, 2 kB of EEPROM, a boot loader in the flash memory, 256 bytes of RAM and 1 kB of XRAM. In addition to its 80C51 architecture, this controller has all the features of the Temic TS8xC51Rx2, namely 256 bytes of on-chip RAM and 1 kB of XRAM, a PCA and the advantage of a fast core (X2 mode). ISP makes it possible to store data directly in the flash memory of the T89C51RD2 using the standard supply voltage. This chip also has a WDT.
The C51 series also has other members, such as the **T8XC510X**. This new 8-bit controller family has 16 kB of ROM/OTP (8 kB for the **T83C5102**) and 512 bytes of RAM. It is housed in a 24-pin package. It can also run at twice the clock frequency (X2).
Some other members of the family are the **T87C5101** (16-kB OTP), the **T83C5101** (16-kB ROM) and the **T83C5102** (8-kB ROM). The clock frequency is 66 MHz at 5 V and 40 MHz at 3 V.
The **TS80C51U2** is a faster controller with more peripherals than its predecessors. The improvements that have been made include the addition of a second UART and a baud-rate generator that makes the internal timers independent of the clock frequency.
New models include the **TS80C52X2** (8 kB), the **TS80C54X2** (16 kB), the **TS80CC58X2** (32 kB) and the **TS80C51RX2**. The 'X2' suffix indicates that these devices work with an internal clock that is twice the external clock frequency, which means that 60-MHz performance can be achieved with a 30-MHz external clock. The size of the memory varies between 16 and 64 kB of ROM/OTP and 512 to 1024 bytes of RAM. Up to 48 I/O lines are available.

The 8/16-bit architecture is characteristic of the C51 family. A three-stage pipeline provides performance that is five times as good as that of a standard 80C51. Using the new instruction set of the C51, it is possible to increase the performance by a factor of 15 at the same clock rate.
In the present C51 family, we can mention in passing the TSC80251, the TS83251 and the TS87251.

Texas Instruments

http://www.ti.com
Although Texas Instruments do not have a big name in the area of controllers, they still offer a very broad product range.
The 16-bit RISC controller series was designed to be extremely energy efficient, so that it can work as long as possible with a battery supply. Some of the more striking features of the **MSP430C336** are: 24 kB of ROM/OTP/EPROM, 1 kB of RAM, a 120-segment LCD driver, an extended supply voltage range (2.5 to 5.5 V) and a current consumption of only 400 μA at 1 MHz and 3 V, or 2 μA in standby mode and 0.1 μA without RAM battery backup. It has a USART and a CAN bus interface and works with an 'old-fashioned' 32-kHz crystal, while the internal clock rate can run as high as 3.8 MHz.

TMS370 family
Some members of this family, designated as the **TMS370Cx8x**, **TMS370C080/380/686** and **SE370C686**, are still in the prototype stage. These are all 8-bit controllers, of which only the SE version might be interesting to readers of *Elektor Electronics*, since its EPROM allows it to be reprogrammed.
Another new development is the TMS370Cx2x series, including the **TMS370C020A**, **022**A, **320A**, **322A**, **722** and **SE370C722**. Their features are the same as those of other modern chips: ISP, SCI1 (Serial Communication Interface 1) and so on. The **772** version has an EPROM.

Zilog

http://www.zilog.com

MUZE Z86E1XX family
The Z8 is the godfather of the **MUZE Z86E1XX** controller family, which is intended to be used for universal applications. These are OTP controllers with on-chip ISP memory ranging from 4 to 64 kB. In-circuit serial programming (ICSP) allows the devices to be programmed after they have been soldered to a circuit board. A standard industrial UART makes it easy to communicate with other controllers or even a PC host. In addition, the MUZE chips have two analogue comparators, two timers, a watchdog timer (WDT) and 237 bytes of RAM. The 13x series has a 28-pin package, while the 14x series has a 40 or 44-pin package. Some technical details: the **Z86E132** has 4 kB of OTP ROM and 237 bytes of RAM, while the **133 has** 8 kB, the **134** 16 kB, the **135** 32 kB and the **136** 64 kB. The devices in the **Z86E14x** series have the same memory capacities as those in the 13x series, but they have 24 I/O lines instead of 32.

Another recent development is the **Z80S183**. This controller shows how the modern embedded controller will appear. It has an 8-channel, 10-bit A/D converter, a 10-bit D/A converter and a Programmable Output Generator (POG). It also includes a real-time clock, 2 kB of SRAM and 1 kB of boot ROM, and it can address up to 1 MB of external memory. It can run in various power-saving modes at clock rates up to 33 MHz.
Another interesting object is the **Z80S188**, which can address an external memory region of 8 MB – currently the maximum amount for an 8-bit controller. The Z80S188 is code-compatible with the Z80 and the Z8X180.
The Z8/Z8Plus is another commonly used Zilog family. We can conclude the Zilog report by noting that few days after its writing, Zilog plan to announce the **Z86E130X** family, consisting of the **136**, **146**, and **126** series.

Conclusions and developments

The facts and developments that we encountered during our research for this article allows us to draw certain conclusions, as follows:

The distinction between controllers and other types of chips with a ‘core’, such as the **PSD835G2** and **8-935G2** ICs from Waferscale Integration (WI), is becoming increasing blurred. The ST1200 DSP MCU from ST Microelectronics, for example, provides evidence of this. As its name suggests, it is a combination of a digital signal processor (DSP) and the core of a microcontroller unit (MCU).

ROM, regardless of whether it is EPROM or one-time programmable (OTP) ROM, is being replaced by flash memory and, to a lesser degree, EEPROM. Flash technology is still in its infancy. Every chip producer recognises its advantages, but not all of them are presently able to integrate it into their products. However, development is proceeding at a rapid pace, as is always the case with electronics.

Another development is in-circuit serial programming (ISP), which happens to be derived from the previously described technology. The use of EEPROM or flash memory is necessary to allow the memory contents of an IC to be (re)programmed after it has been soldered into the circuit board. Flash memory is enjoying steadily increasing popularity, due to its high density and low cost.

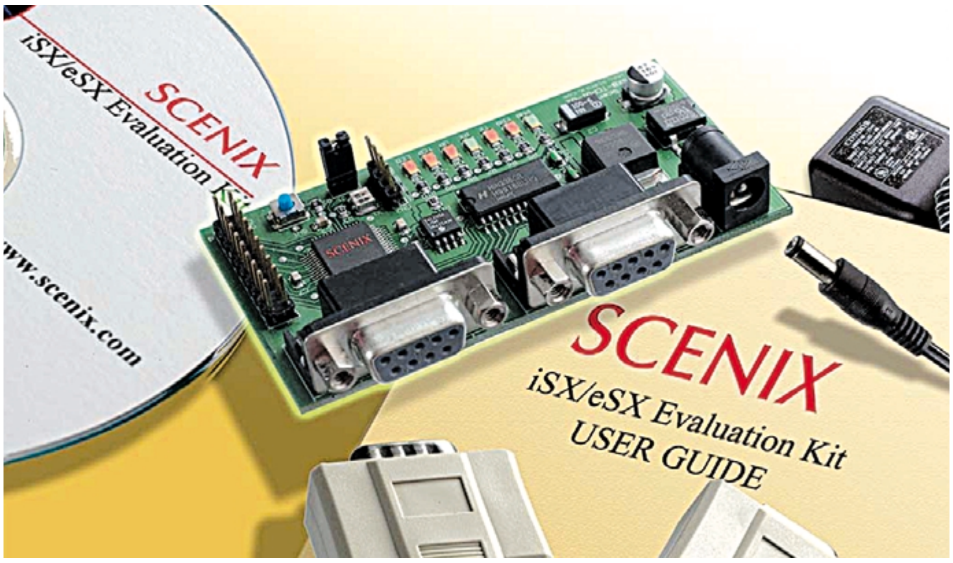
Yet another development is the integration of as many peripheral functions as possible on the chip, such as flash memory and an EEPROM. Motorola, for example, do this with their HC08 and HC12 series. We also see that an increasing number of controllers, such as those from Infineon, ST Microcontrollers and Microchip, have USB ports.

The final development that we want to mention is that the fabrication process width is becoming steadily smaller, with 0.25 μm already standard. The integration of a controller core into an IC to make it ‘intelligent’ is also becoming standard practice. One of the popular cores at the moment is the **ARM7TDMI** from ARM. This is found in the products of many manufacturers, including Atmel, Epson, LSI Logic, OKI Electric and Philips.

As you have probably gathered from reading this article, the controller world is highly dynamic, with new models appearing quite regularly. Although we have attempted to make this report as complete as possible, we unfortunately cannot mention all manufacturers and all of their products, due to lack of space.

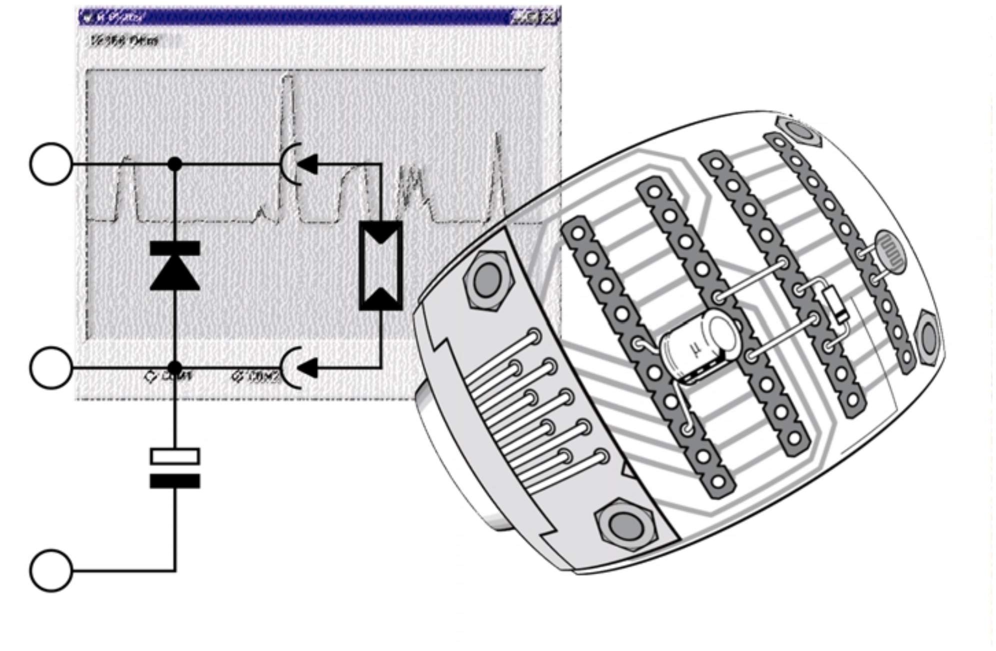
(000142-1)

Recommended website for further information:
Chip Directory
http://www.chipdir.com/chipdir/chipdir.htm



PC Serial Peripheral Design (6)

Measurements with sensors



By B. Kainka

In the previous instalment we measured capacitances and resistances. You may have wondered if this was all worth the effort, when accurate and inexpensive digital multimeters are readily available, but we shall see that we are able to take readings from a wide variety of sensors by measuring their resistance.

The best demonstration involves a light-dependent resistor (LDR). **Figure 1** shows a simple light-measuring circuit; **Figure 2** shows how it might be constructed. In this case we are frequently interested in how the reading changes over time, and this is best shown on a graph.

Light measurement

In order to produce a graph on the screen using Visual Basic, we need to use a so-called PictureBox, which can be dragged from the form's toolbar. It is very important to set the correct size. Visual Basic can use a variety of units and here we want to measure the size in pixels. The ScaleMode property should be set to '3 - pixel', and the box can be

Listing I. Resistance Plotter

```
Dim y1, y2, x1, x2, n
Private Sub Form_Load()
 i = OPENCOM("COM2,1200,N,8,1")
 If i = 0 Then
    i = OPENCOM("COM1,1200,N,8,1")
    Option1.Value = True
 End If
 If i = 0 Then MsgBox ("COM Interface Error")
 TXD 0
 RTS 0
 DTR 0
 Counter1 = 0
 Timer1.Interval = 1000
 n = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
  CLOSECOM
End Sub

Private Sub Option1_Click()
 i = OPENCOM("COM1,1200,N,8,1")
 If i = 0 Then MsgBox ("COM1 not available")
End Sub

Private Sub Option2_Click()
 i = OPENCOM("COM2,1200,N,8,1")
 If i = 0 Then MsgBox ("COM2 not available")
End Sub

Private Sub Timer1_Timer()
  DTR 1
  REALTIME (True)
  TIMEINITUS
  While (DSR() = 0) And (TIMEREADUS() < 900000)
  Wend
  T = TIMEREADUS()
  DTR 0
  T = T * 0.932
  R = 2200 + 7800 * (T - 76300) / (294600 - 76300)
  REALTIME (False)
  R = Int(R)
  Label1.Caption = Str$(R) + " Ohm"
  y2 = 300 - R / 100
  If n = 0 Then y1 = y2
  x1 = n
  n = n + 5
  x2 = n
  Picture1.Line (x1, y1)-(x2, y2)
  y1 = y2
End Sub
```

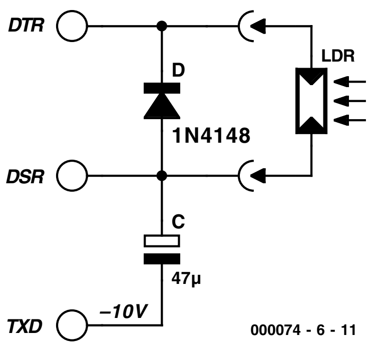


Figure 1. Measurement using a light-dependent resistor

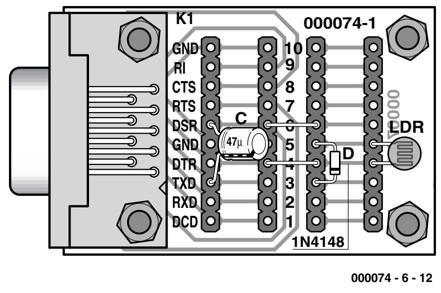


Figure 2. Construction of the LDR circuit

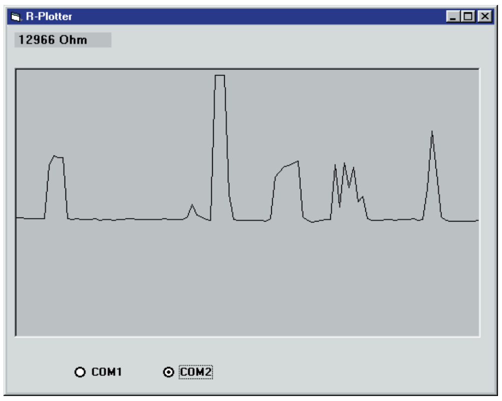


Figure 3. Results from the LDR measurement

dragged out to the correct size using the mouse. For the present example (plotter1.frm) a PictureBox 300 by 500 points is required.

We can draw in the PictureBox using the Line command: see **Listing 1** for an example. The plot is drawn over a period of 100 seconds, the x-coordinate advancing by 5 pixels as each reading is taken. A global variable n must be specially declared to store the current position across calls to the timer procedure. On the y-axis resistance (R/100) is plotted, with a maximum value of 300 (30000 Ω=30 kΩ). Every second a new value is plotted in the graph in **Figure 3** and a new line segment is drawn from the last point to the new one. The old coordinate values must be stored, and global variables are used for this purpose.

Skin resistance measurements

Another interesting 'sensor' is the human skin: a pair of bare wires can be wound around two fingers. The resistance of the skin is not constant,

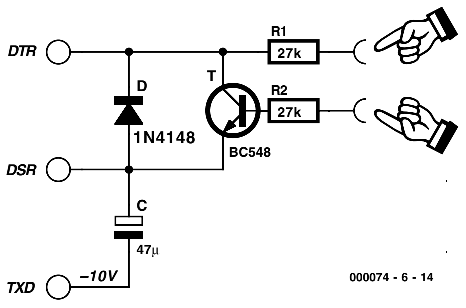


Figure 4. Amplifier for measuring skin resistance

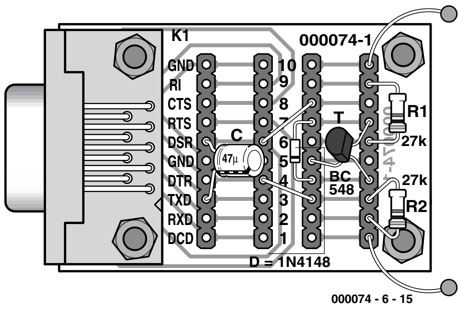


Figure 5. Measuring skin resistance

Listing 2. Extensions for temperature measurement

```
Private Sub Timer1_Timer()
  DTR 1
  REALTIME (True)
  TIMEINITUS
  While (DSR() = 0) And (TIMEREADUS() < 1500000)
  Wend
  T = TIMEREADUS()
  T = T * 1.0000000001
  R = 2200 + 7800 * (T - 76300) / (294600 - 76300)
  REALTIME (False)
  R = Int(R)
  Temp = 1 / (Log(R / 10000) / 4300 + 1 / 298) - 273
  Temp = Int(Temp * 10) / 10
  Label1.Caption = Str$(Temp) + "°C"
  DTR 0
End Sub
```

Listing 3. Conversion to the Fahrenheit scale

Although the German physicist Gabriel Daniel Fahrenheit died in 1736, some people are still more comfortable working with temperatures measured in Fahrenheit. They need only change a couple of lines of the program, as follows:

```
  Temp = 32 + (Temp / 100 * 180)
  Temp = Int(Temp * 10) / 10
  Label1.Caption = Str$(Temp) + " F"
```

but varies according to its moisture levels. When people lie, they sweat: so we can build a lie detector by determining when the skin resistance falls. The problem is that skin resistance is typically rather higher than the maximum value we can measure with our circuit: we need to use a transistor for amplification (**Figure 4, Figure 5**).

An NPN transistor amplifies the small current flowing through the skin, allowing our simple A/D converter to measure relatively high resistances. The input to the circuit is protected by two additional resistors, so that the measured resistance will not be too small even if the wires are shorted. The reading obtained depends on the contact area of the wires and the moisture level of the skin. The results can be viewed using the resistance plotter program.

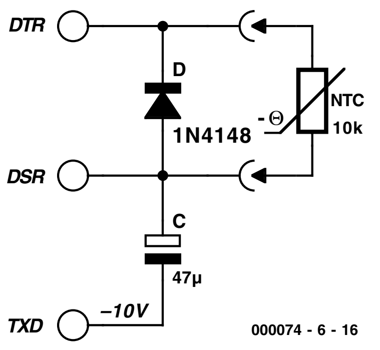


Figure 6. Connecting an NTC thermistor

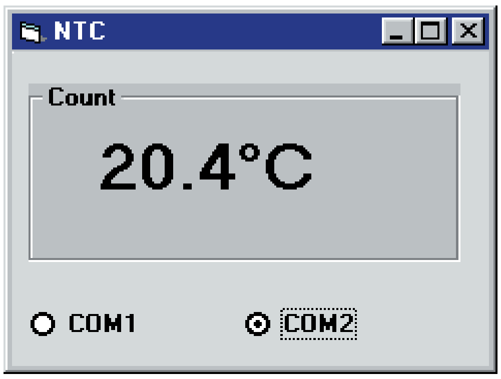


Figure 7. Temperature display

Temperature measurement

Temperature can also be measured using our resistance measuring circuit. The circuit of **Figure 6** shows how easy it is to connect a 10 kΩ NTC thermistor to build a very simple yet perfectly usable thermometer.

The resistance characteristic of an NTC thermistor can be approximated by an exponential curve. In the following formula, T is the absolute temperature in Kelvin (T/°C + 273), while B is a value, typically between 2000 and 500 Kelvin, provided by the thermistor manufacturer. In fact the value of B is not perfectly constant but rises gradually with temperature: for this reason the value $B_{25/85}$ is often quoted, where calibration has been performed at 25°C and 85°C.

$$R(T) = R_{25} \cdot e^{B \cdot \left(\frac{1}{T} - \frac{1}{298K} \right)}$$

An NTC thermistor is therefore specified to a first approximation by two values: B and the thermistor's nominal resistance R_{25}. For a typical 10 kΩ NTC thermistor the value of B might be 4300 K. Rearranging the expression above and substituting R_{25} = 10 kΩ we can derive the following line of Visual Basic to calculate the value of T in Celsius:

Temp = 1 / (Log(R / 10000) / 4300 + 1 / 298) – 273

The VB 'Log' function calculates the natural logarithm, often abbreviated to 'ln' in mathematics textbooks.

In the program shown in **Listing 2** we first carry out a resistance measurement and then convert the measured value into temperature. The results appear to one decimal place as shown in **Figure 7**. Finally, **Listing 3** shows how to convert the reading from the Celsius scale into Fahrenheit.

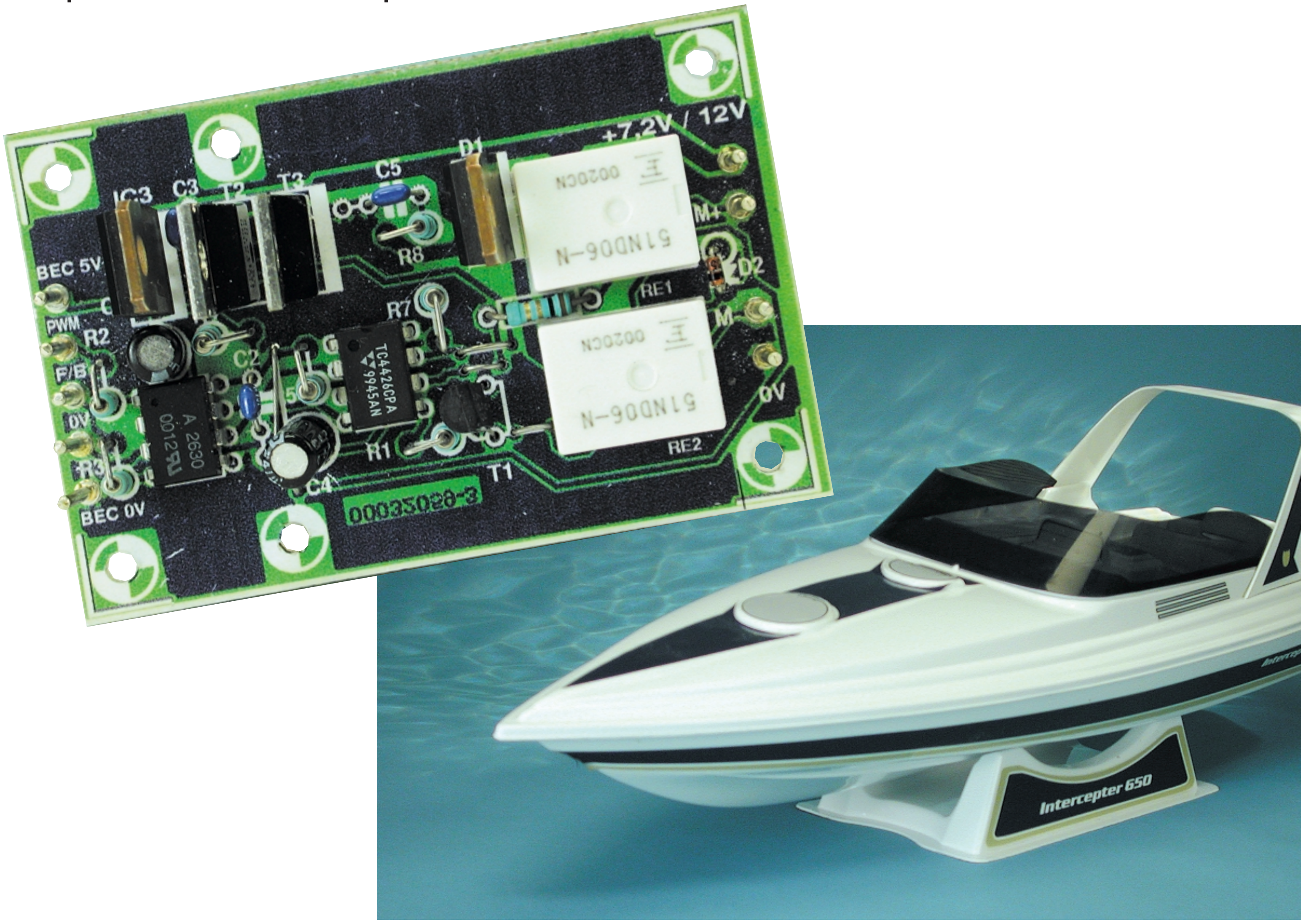
(000074-6)

Speed Controller Duet (2)

part 2: construction, programming and the power units

Design by B. Stuurman

Part 1 presented an overview of the system. In part 2, we deal with the construction of the SpeedControl unit, user programming with a PC (via the RS232 interface) and finally the construction of the two power units: SpeedPower 1 and SpeedPower 2.



COMPONENTS LIST

Resistors:
R1,R4,R8,R13 = 10kΩ
R2 = 100kΩ
R3,R7 = 2kΩ2
R5,R6,R9-R12 = 1kΩ
R14,R15 = 560Ω
P1 = multiturn preset, vertical mounting (e.g., Bourns 3386W)

Capacitors:
C1 = 220pF ceramic, lead pitch 2.5 mm
C2 = 100µF 16V, radial
C3 = 100nF
C4,C5 = 18pF ceramic

Semiconductors:
D1 =zener diode 2V3
D2,D3 = high-efficiency-LED (3 mm) red
D4 = high-efficiency-LED yellow
D5 = high-efficiency-LED green
D6,D7 = BAT85
T1,T5 = BC547
T2,T3,T4 = BC557
IC1 =ST62T60BB6, programmed, order code **000070-41** (see Readers Services-page)

Miscellaneous:
S1 = pushbutton, make contact
X1 = 4MHz quartz crystal
Bz 1 = AC buzzer
Ferrite suppressor coil
PCB:
- in combination with SpeedPower1: order code **000070-5**;
- in combination with SpeedPower2: order code **000070-4**
(see Readers Services-page).

SpeedControl construction

The printed circuit board for the SpeedControl unit is shown in **Figure 4**. For economic reasons, this board is only available in combination with one of the two SpeedPower boards (see the Components List), so the first task is to saw apart the two boards (if you so desire).

If you work calmly and carefully with a fine-tipped soldering iron and keep to the components identified in the Components List, you shouldn't have any problems assembling the circuit board. A servo extension cable that has been cut in two can be used for the connecting cable. It is a good idea to fit a ferrite toroidal core onto the cable to the receiver, since this will isolate the circuit from high frequency signals from the receiver, and the remaining length of the cable will not degrade signal reception.

When everything is finished, wait a bit before placing the microcontroller in its socket. First connect the SpeedControl to an adjustable power supply with a range of 0–6 V, and connect a voltmeter between 0 V and a needle inserted into the connection for the reset signal at the IC socket. Slowly increase the applied voltage. The reset signal should remain Low until around 3.7 V, where it should immediately go High. From around 3.75 V onwards, it should track the supply voltage. If this is all OK, you

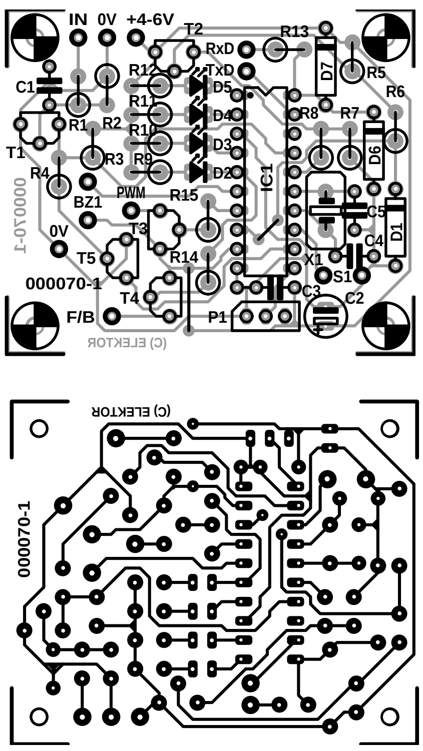


Figure 4. SpeedControl printed circuit board layout (scale 1:1)

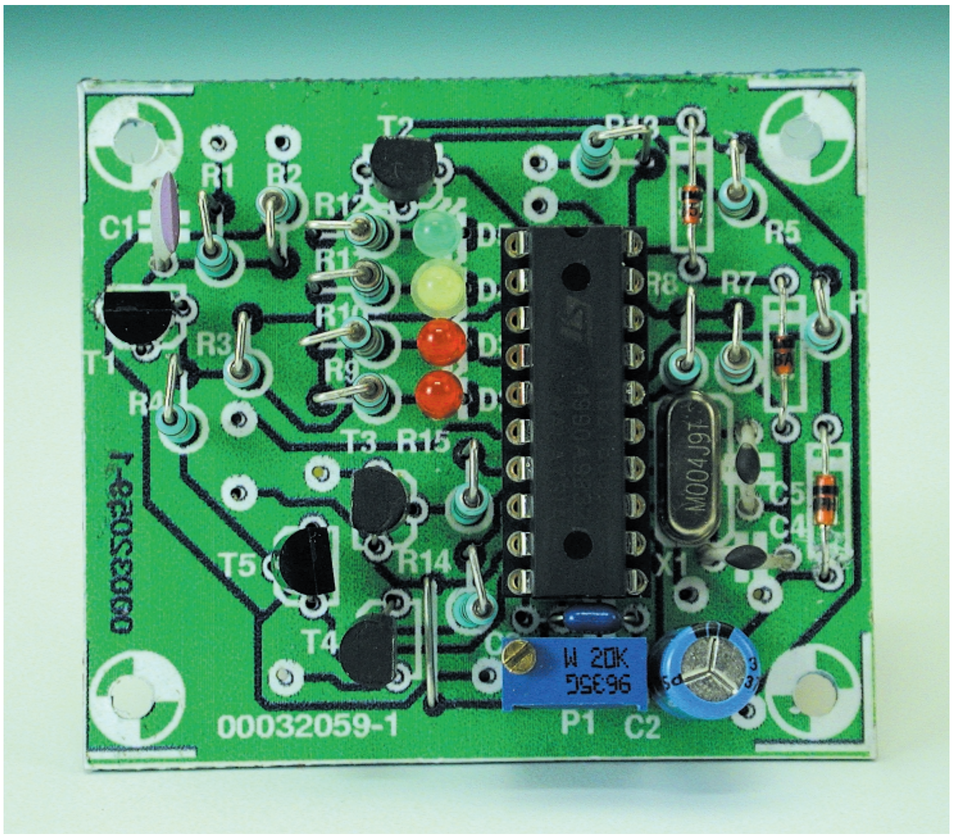


Figure 5. Fully assembled SpeedControl unit.

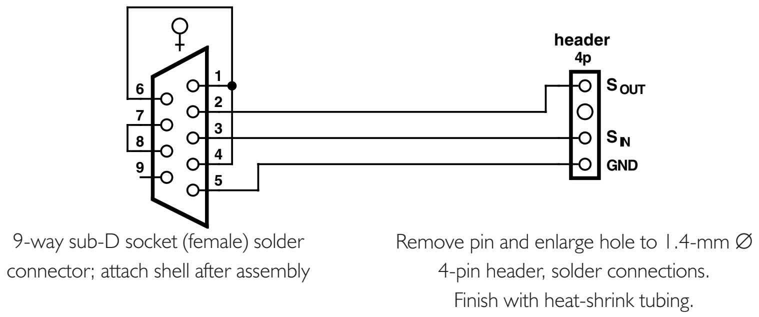

header
4p
S OUT
S IN
GND
1
2
3
4
5
6
7
8
9
9-way sub-D socket (female) solder connector; attach shell after assembly
Remove pin and enlarge hole to 1.4-mm Ø 4-pin header, solder connections. Finish with heat-shrink tubing.

Figure 6. RS232 cable for communicating with a PC.

can insert the microcontroller in its socket. The ready-made circuit board for the SpeedControl is shown in **Figure 5**.

Programming

Before going any further, it's helpful if you first match the SpeedControl to the transmitter using the three-point timing setup, and in particular the setup described in Example 2 of the first part of this article. Once this has been done, you can test the operation of the 'forw/back' output.

After this test, the receiver and the SpeedControl can be switched off. Set trimpot P1 to the 'speed' position and hold the pushbutton down while switching on the power. Now you can test the operation of the trimpot. Its behaviour should be essentially the same as that of the control stick of the transmitter.

If no programming has been carried out, default values are used. The microcontroller copies these value from the ROM to the EEPROM when it is first started up. A 'virgin' EEPROM cell contains either 00h or FFh. After the values have been copied, the 'test cell' contains 08h, so the program knows that the EEPROM has already been written. In the descriptions of the commands, the default values are shown in parentheses.

A special cable is needed to allow the SpeedControl to communicate with a PC. The schematic of this cable is shown in **Figure 6**. The three loopback connections trick the RS232 interface into being able to function with only three leads ('gnd', 'in' and 'out'). In addition, you will need a terminal emulator, such as the Windows accessory HyperTerminal program. If you don't use Windows, an 'old-timer' communications program, such as ProComm, is perfectly usable. Set the protocol parameters to 19200 baud, 7 data bits, no parity and 2 stop bits. With HyperTerminal, you can save these settings on the desktop under the name 'Speed.ht' (for example). In our case, the new settings took effect only after HyperTerminal had been stopped and then restarted via Speed.ht.

In order to enter the programming mode, you must set the trimpot to the 'program' position and then switch on the power while pressing the pushbutton. The SpeedControl unit will then announce itself with the message 'command?' on the PC monitor. Now you can enter commands. If an incorrect command or incorrect parameter is entered, a beep is generated and a message appears on the screen. You can exit this situation by pressing the Escape key. If you type the command '?' followed by the Enter key, the command set is displayed as illustrated in **Figure 7**. Only lower-case letters may be used for entering commands. If the parameter is omitted, the currently active setting is displayed.

Commands

reptest (on) This switch command enables or disables testing the repetition interval (pulse rate). Disable this test only if the repetition interval of the transmitter is highly unstable. Note that this command disables the test, so to restore the original setting, the command must be issued twice. This applies to all switch commands!

cutoff [parameter] This command is used to set or disable the cutoff voltage. This is indispensable if the supply voltage for the receiver, servos and SpeedControl is taken from the BEC of a power unit, so that there is no separate battery for the receiver. When the supply voltage drops below the set-point value, the motor voltage is switched off. The power supply to the motor can be restored by putting the control stick in the neutral position, and it will remain on until the voltage again drops below the set-point value. In general, the cutoff voltage is referenced to the full-load battery voltage, for which a value of 0.7 V per NiCd cell is used (Sub-C, 1500–2000 mAh).

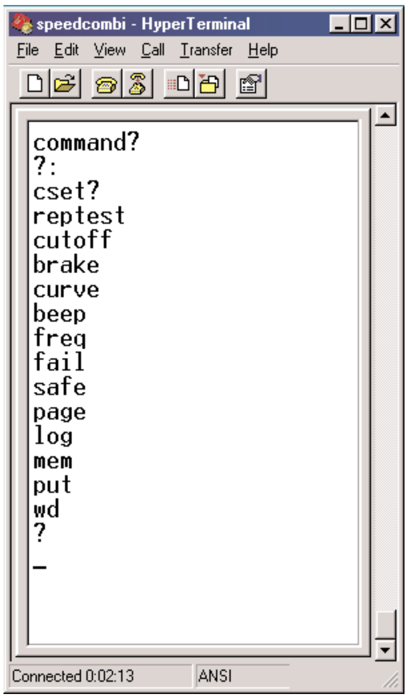
brake [parameter] The 'brake' command is intended for electric gliders with folding propellers that are fitted with a SpeedPower 2 unit. As soon as the control stick (or the trim) is set to the 'reverse' position, the brake FET is activated after a set time delay. 'Reverse' to 'forward' always has priority.

curve (linear) This switch command alternates between a linear and an exponential power curve. In the latter case, the amount of power supplied to the motor is a better match to the position of the control stick.

beep (high) This sets the volume of the beep to 'loud' or 'soft'.

freq [parameter] (2 kHz) This allows the frequency of the PWM signal to be set to four different values.

fail [parameter] (1 s) The combination of this command and the 'safe' command forms the 'fail-safe' provision of the SpeedControl. The 'fail' command can be used to specify the maximum interval allowed for receiving no pulses (or bad pulses) before reducing the motor power to zero.



```
command?
?:
cset?
reptest
cutoff
brake
curve
beep
freq
fail
safe
page
log
mem
put
wd
?
_
```

Figure 7. The SpeedControl command set.

safe (resume) The options for this command are 'resume' and 'go neutral'. If the motor power has been set to zero in response to a 'fail' condition and the 'resume' option is selected, normal operation will be resumed as soon as control pulses are again received. If the 'go neutral' option is selected, the control stick must first be set to the neutral position before normal operation is resumed.
log (off) If 'log' is active, bad pulses are recorded in the 'logbook'. This is a 16-bit counter that provides an indication of the reliability of the link. The logging function costs a lot of extra time in the controller, so it should normally be disabled.

Diagnostic commands

wd (on) This can be used to enable or disable the digital microcontroller's watchdog timer. If the watchdog times out, the microcontroller is reset. When the watchdog is disabled, the command processor in the main program loop is enabled and we can examine all memory locations in the microcontroller and modify them as desired.
mem [parameter] (off) This command, followed by a hexadecimal address, causes the content of the selected memory location to be displayed on the screen, updated approximately three times per second (exit with the Escape key).
put [parameter] The parameter byte is written to the memory location specified by the 'mem' command.

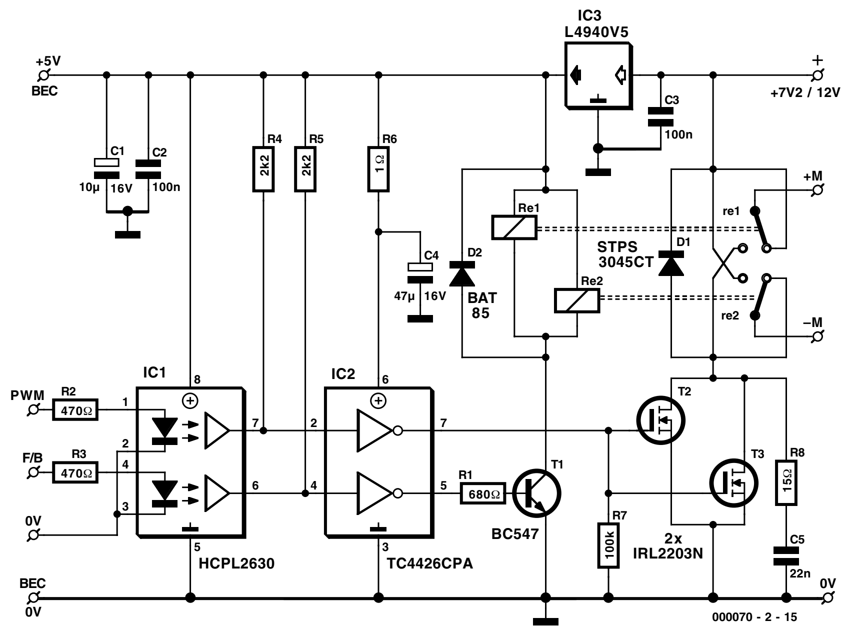


Figure 8. Schematic diagram of the SpeedPower 1.

page [parameter] 0 = EEPROM page 0, 1 = EEPROM page 1, 2 = RAM page 2.

The diagnostic commands can be used to restore the default values. This is done as follows. First enter 'mem 00' to select memory address 00h. Next, enter 'page 0' to select EEPROM page 0, and then enter 'put 00' to write the byte 00h to that address. Complete each entry with the Enter key. When the SpeedControl is again switched on, the default values will again be written to the EEPROM.

SpeedPower 1

The schematic diagram of the SpeedPower 1 unit, which has forward/reverse capability, is shown in **Figure 8**. The 5-V voltage regulator (IC3), which also supplies power to the BEC, is a low-drop type that can supply 1.5 A. With

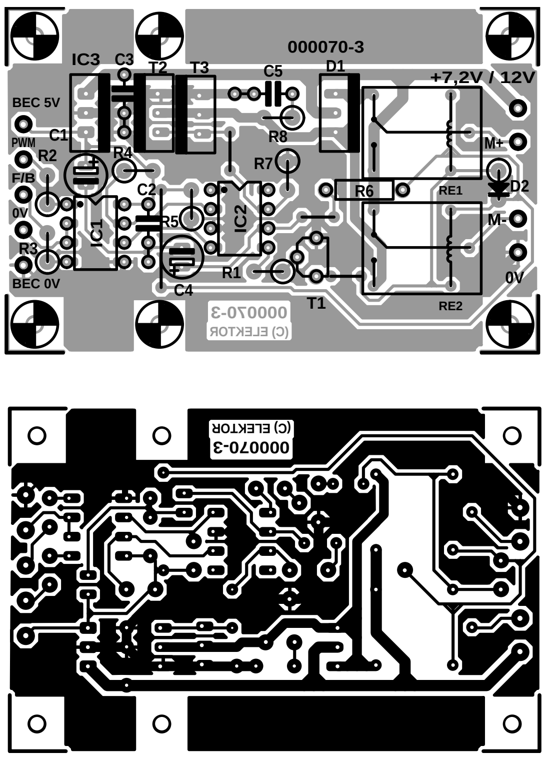


Figure 9. Printed circuit board layout for the SpeedPower 1.

COMPONENTNS LIST
SpeedPower 1

Resistors:
R1 = 680Ω
R2,R3 = 470Ω
R4,R5 = 2kΩ2
R6 = 1Ω
R7 = 100kΩ
R8 = 15Ω

Capacitors:
C1 = 10µF 16V radial
C2,C3 = 100nF
C4 = 47µF 16V radial
C5 = 22nF

Semiconductors:
D1 = STPS3045CT
D2 = BAT85
T1 = BC547
T2,T3 = IRL2203N
IC1 = HCPL2630
IC2 = TC4426CPA
IC3 = L4940V5

Miscellaneous:
Re1, Re2 = relay, Fujitsu 51ND06-N (Conrad Electronics #505099-11)
PC1-PC9 = solder pin
PCB, order code **000070-5** (in combination with SpeedControl – see Readers Services page)

two standard servos, the average current consumption from the BEC is around 150 to 250 mA, but if reception is lost and the servos start to jitter, the current consumption can rise to as much as 1.2 A, and the receiver must continue to work reliably under these conditions. The PWM and 'for/back' signals are electrically isolated by a dual digital optocoupler (IC1). IC2 is a dual inverting driver for power MOSFETs. Each output can source or sink 1.5 A, and the voltage can range from 0 V on the ground lead (pin 3) to the supply voltage (pin 6). The inputs are TTL compatible. One driver output (pin 7) is used to control two power MOSFETs wired in parallel (T2 & T3), while the other output controls a switching transistor (T1) that activates the relay. A pair of Schottky diodes in a dual package is connected across the MOSFET outputs to handle flyback currents, along with the 'snubber' network (R8 & C5) that protects the MOSFETs. The circuit is designed such that the motor cannot run in the absence of an input signal.

Figure 9 shows the printed circuit board for the SpeedPower 1. This is supplied in combination with the SpeedControl circuit board, and it can be separated from the latter board if desired. The same remarks apply to the assembly of this board as for the SpeedPower board. The current-carrying tracks are laid out extra wide, and if necessary they can be electrically 'beefed up' by soldering lengths of copper wire on top.

Finally, fasten the interconnecting cables and BEC cable. If the distance between the power unit and the receiver is greater than 20 cm, it is a good idea to fit a ferrite toroidal core on the BEC cable, close to the receiver.

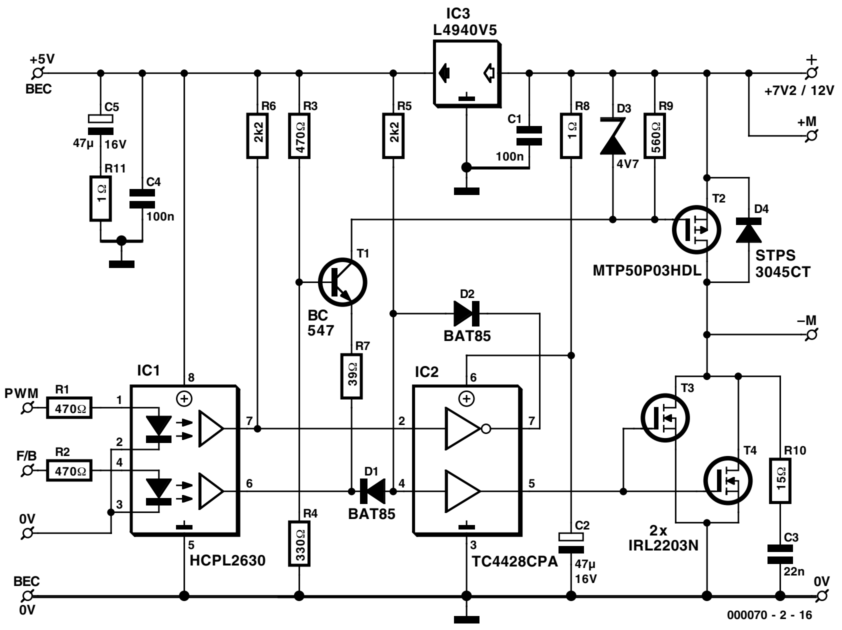
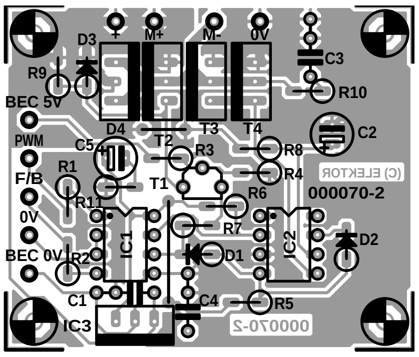


Figure 10. Schematic diagram of the SpeedPower 2.

SpeedPower 2

Figure 10 shows the schematic diagram of the SpeedPower 2 unit, which includes a brake FET. The power MOSFET driver IC2 is a different type than the driver used in the SpeedPower 1 unit; it has an inverting and a non-inverting stage. When 'for/back' is not active, output pin 6 of the optocoupler (IC1) is High, which blocks diode D1. The 'PWM' signal reaches the non-inverting driver, which controls the power MOSFETs, via the inverting driver and diode D2. When 'for/back' is active, output pin 6 of the optocoupler is Low, so that diode D1 conducts. Input pin 4 of the optocoupler remains Low, and consequently also the gates of T2 and T3.

The base of T3 is connected to a fixed voltage of 3 V. When 'for/back' is Low, the emitter of T1 is connected



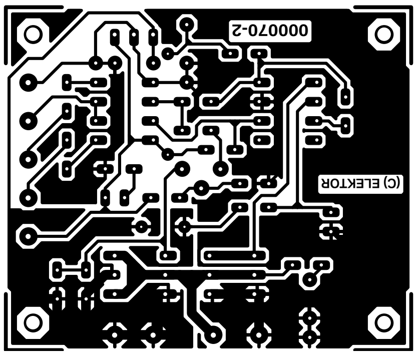


Figure 11. Printed circuit board layout for the SpeedPower 2.

COMPONENTS LIST
SpeedPower 2

Resistors:
R1,R2,R3 = 470Ω
R4 = 330Ω
R5,R6 = 2kΩ2
R7 = 39Ω
R8,R11 = 1Ω
R9 = 560Ω
R10 = 15Ω

Capacitors:
C1,C4 = 100nF
C2,C5 = 47µF 16V radial
C3 = 22nF

Semiconductors:
D1,D2 = BAT85
D3 = zener diode 4V7
D4 = STPS3045CT
T1 = BC547
T2 = MTP50P03HDL
T3,T4 = IRL2203N
IC1 = HCPL2630
IC2 = TC4428CPA
IC3 = L4940V5

Miscellaneous:
PC1-PC9 = solder pin
PCB, order code **000070-4** (in combination with SpeedControl – see Readers Services page)

Power MOSFETs

Due to the low nominal value of the supply voltage (7.2 V), it is not possible to use ordinary MOSFETs in the power units without resorting to a lot of trickery. The threshold voltage of such devices, $V_{gs(thresh)}$, lies at around 4 V, and the saturation resistance $R_{ds(on)}$ shown in the data sheets is only achieved at a voltage of 7 to 8 V. With logic power MOSFETs, the threshold voltage is much lower (2 to 2.5 V), and the on resistance is already extremely low at 'logic voltage' levels (5 V). The price that must be paid for this is that the gate capacitance is much greater than that of a standard power MOSFET. This means that more power is needed for the (PWM) drive signal, which is why special driver ICs are used in the power units. We have selected three types of logic power MOSFETs that are very well suited for use in the power units. They are listed in the accompanying table.

All three of these power MOSFETs have a TO-220 package. This type of package can dissipate 1 W without supplementary cooling. With a small cooling fin (a piece of 2-mm thick aluminium, around 10 × 27 mm), this can be increased to 2 W. The maximum continuous current that a single MOSFET can handle is thus the amount that produces a dissipation of 2 W. For the IRL2203, this is $\sqrt{(2/0.01)}$ = 14 A. A power unit with two IRL2203s can thus handle a continuous current of 28 A. The highest current level (36 A) can be achieved using a pair of IRL3803s, but these are significantly more expensive. Don't forget that we're talking about continuous current here; considerably higher currents can be handled for short intervals. The table shows the current handling capacities of power units fitted with pairs of the listed types of power MOSFETs, both continuously and for a 30-second interval.

MOSFET type	$R_{DS(on)}$ V_{gs}=5 V	$R_{DS(on)}$ V_{gs}=10 V	$I_{(cont)}$ 2 pieces	$I_{(30s)}$ 2 pieces
BUZ100L	18 mΩ	12 mΩ	26 A	39 A
IRL2203	15 mΩ	10 mΩ	28 A	42 A
IRL3803	9 mΩ	6 mΩ	36 A	54 A

to 'ground' via R7. T1 then acts as a constant-current source, and its collector sinks 20 mA. This pulls the gate of MOSFET T2 towards ground until Zener diode D3 starts to conduct. A stable state then arises, in which T2 'shorts out' the motor with a resistance of around 20 mΩ. This is the 'brake' state. When 'for/back' stops being active, T1 cuts off and the gate of T2 is again pulled to the source level, which causes T2 to cut off.

If the brake state does not work effectively, the value of R9 may be increased as necessary, up to around 1 kΩ.

The printed circuit board designed for the SpeedPower 2 unit is shown in **Figure 11**. This is also provided in combination with a SpeedControl board. Here again, the relevant printed circuit tracks have been kept as wide as possible, but it is recommended to reinforce the connections between T2–T4 and the M+ and M– connections by soldering copper wires on top.

Both power units should be initially tested using a power supply that is current limited to two ampères. After this, you can connect them to the battery pack, but be careful on account of the high currents that can flow. There is a risk of fire in case of a short circuit, and explosions are even possible.

And finally: the controls are in your hands!

(000070-2)

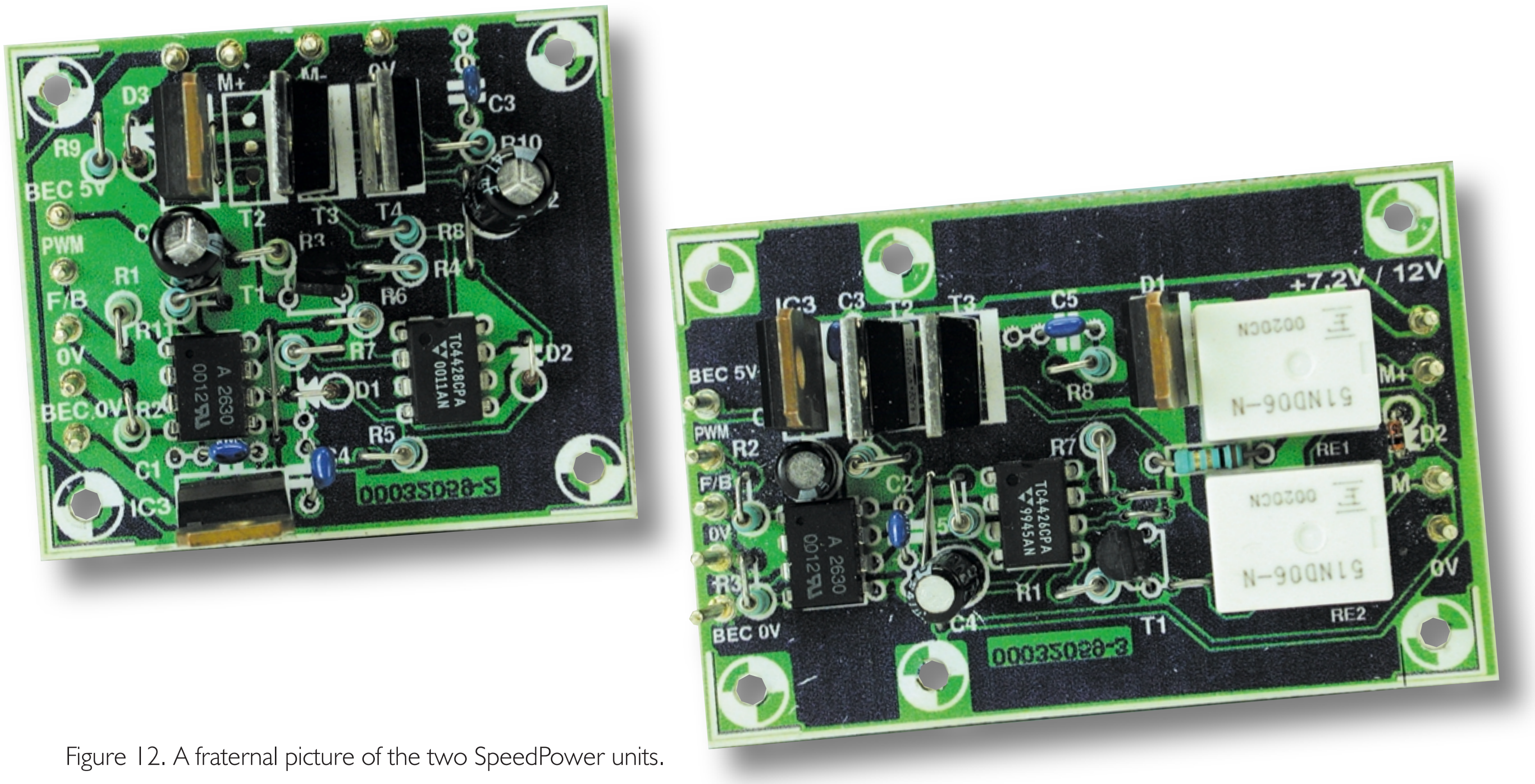


Figure 12. A fraternal picture of the two SpeedPower units.

Home Bus Systems

By H.-J. Peifer, G. Krause and J. Rütten

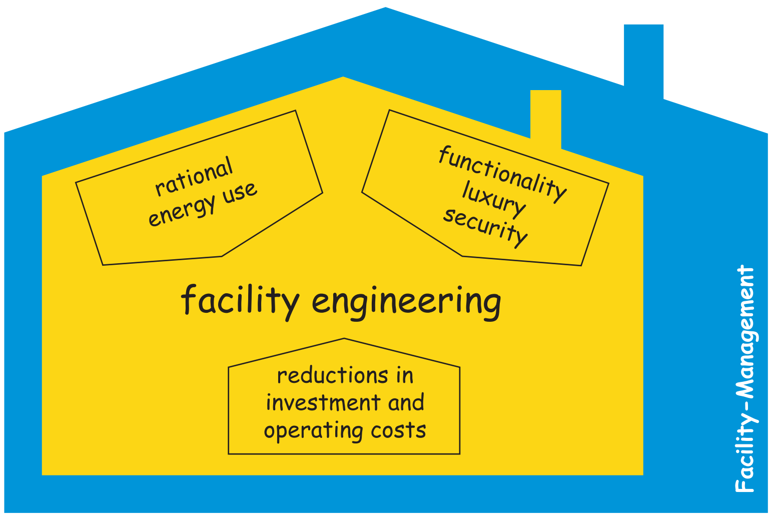
Aachen Polytechnic, Germany

The future of automatic control of home and office environments is with Building Management Systems. The dream is of greater comfort, functionality and security with more efficient use of energy and optimised building running costs. These systems however have found a slow uptake and difficult Marketing.

The idea of intelligent buildings conveys images of some future utopian dream where the environment inside the home is controlled for maximum comfort and convenience. Cameras outside watch for any unauthorised movements and silently convey your alarm status to some distant security monitor. At the office each area senses occupancy and reduces the heating and lighting to empty rooms. Altogether the promise is of more comfort, better functionality, flexibility, security and energy saving. This is precisely the aim of current Building Management Systems (BMS).

For most complex control situations it is usual to employ a bus system so that control and data can be transferred using relatively few interconnecting wires. In the beginning of bus based Building Management Systems, industry standard field buses were used. These were later phased out as improved dedicated BMS buses were developed.

We already see buses used to link several intelligent sub-systems together with one main management controlling unit. The central management unit analyses conditions within the building gathering data from peripheral sensors and sending control commands to individual equipment. This is an example of a centralised energy management system. One such open communication system built on this principle is the BACnet (Building Automation and Control Networks). The network protocols are designed for use at the management and automation level of the network and provide a comprehensive set of objects for organising the data in building controllers and a set of communication services to convey data between devices. The protocol has also been extended to include trend logs.


rational energy use
functionality luxury security
facility engineering
reductions in investment and operating costs
Facility-Management

000144 - 18

This standard should not be seen as competing with current systems but as a kind of common denominator while it can be interfaced with all the other existing standards.

One level deeper at the field level where terminal units, controllers and sensors connect to the bus, there is another entirely different set of industry standards.

Current Systems

In the building control technology there are some propriety systems (with only one supplier) and also systems supported by many companies that conform to generally agreed standards.

The proprietary systems include the Local Control Network (LCN) and Peha House Control (PHC). The LCN uses an additional line in the mains socket to pass control and data information around the building. Installation of this system is really only feasible if it is installed in a new building where the mains cabling has yet to be installed. Segments of the network can be linked to an Arcnet backbone allowing a bus capacity of 2.5 Mbit/s. This system is therefore suitable for large commercial installations. In contrast, PHC is designed more for residential and home environments while it has a limited number of connections and the equipment available is more suited to these applications. Both of these systems have, so far, not gained wide acceptance.

Globally accepted standards include the systems X10, CEBus, LON, EHS, BatiBus and EIB. X10 and CEBus have been developed specifically for the U.S. market and adaptation of these standards for Europe is unlikely.

Likewise from the USA the company Echelon developed the Local Operating Network (LON). This system was originally conceived as a general control standard for automated tasks. It is well suited to building management applications. Worldwide it is estimated that over 10 million LON based units are currently in use.

The European standards are currently undergoing a convergence procedure whereby three of the main European standards the EIBA, Batibus Club International (BCI) and the European Home Systems Association have pooled their experience and resources to form the Konnex Association. This association is now responsible for defining the transfer protocols for these three systems. Currently there are over 10 million EIB compatible units in use. Globally, this puts its market share as roughly equal to the LON based systems.

Bosch-Siemens Appliances have introduced household equipment that feature an EIB approved bus interface for their Home Electronic System (HES). Using a software package called Home Assistant, it is possible for the user to control these appliances and receive status information either using a PC or via a touchscreen.

Based on the technical capabilities and the open architecture of both LON and the EIB (Konnex) systems, they are likely to remain the main players in the field of building management systems.

EIB and LON

The EIB standard was originally conceived jointly by a group of companies headed by Siemens. In 1990, the European Installation Bus Association (EIBA) was formed and took over responsibility for the specification and certification of EIB compliant appliances. Similarly in 1993 the LonMark Organisation took responsibility for the LON standard. Unlike the EIB, the LON Certification procedure is not obligatory, consequently some LON equipment on the market from diverse manufacturers are not fully compatible in every aspect.

EIB and LON broadly offer similar functions but have strengths in different areas of application. Many companies support the two standards. Producers of electrical installation technologies who supply a wide range of sensors and actuators supply EIB systems. LON based systems are supplied by companies with a background in control technology, lighting control etc. It could be said that the EIB standard has grown up from the installation side of the industry gradually building more complex control functions whereas the LON system has come from the other direction. There is little information available to compare the relative performance of the two systems in different control environments, what is probably more important is the cost and availability of the system.

The most important and most widely deployed communication medium for both systems is the twisted pair cable.

The EIB system allows a maximum distance between users in the same segment of 700 m. The bus topology allows users to be connected in a line, star or tree configuration provided that no loops are formed in the signal routing.

The data rate of 9.6 kbit/s gives a bit length of 104 μs which is much longer than any signal propagation delay down the bus, so data is sent directly onto the bus without the need for terminating resistors at the ends of bus branches.

A network interface unit connects the user to the bus and an internal inductor matches the signal to the bus impedance. **Figure 2** shows how the data appears on the bus. The bus voltage is normally at 24 V. When a '0' is sent, the bus voltage is pulled low to 10 V for a period of 35 μs. To send a '1', the bus voltage remains high at 24 V.

The bus protocol used is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This method allows each bus user to only send data onto the bus when they know it

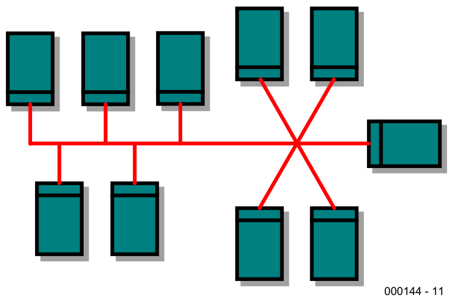


Figure 1. EIB Topology.

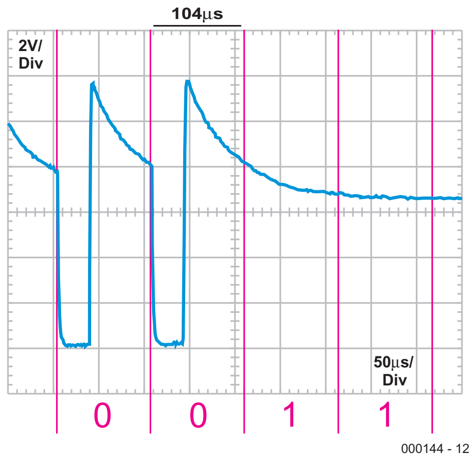


Figure 2. EIB Twisted Pair Signal.

will be free, thereby avoiding collisions between users which would otherwise occur if two or more send data simultaneously.

The most widely used LON network configuration uses the medium of a twisted-pair cable in a free topology. This allows loops to be made in the signal routing. The maximum distance between users in a segment depends on the cable type employed and can be up to 500 m. Using a data rate of 78 kbit/s means that the signal bit duration of 12.8 μs is in the order of the propagation delay down the bus. LON uses special signal coding to overcome this potential problem. Signal encoding employs the DC-free Differential Manchester coding (**Figure 4**). A logic '1' is represented by only one edge change within a bit period whereas a logic '0' has two edge changes, the additional change occurs in the middle of the bit period. This gives a line frequency of 39 kHz if a series of logic ones is sent, and 78 kHz for a series of zeroes. Equipment connected to a LON network can be powered from the net but this is not obligatory although it removes the necessity for a mains unit and is the preferred option. The Link Power net unit provides bus termination.

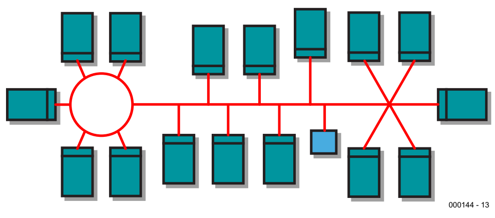


Figure 3. LON Free Topology.

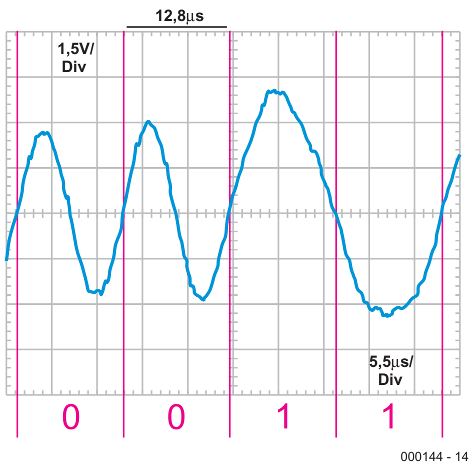


Figure 4. LON Twisted Pair Signal.

The bus protocol used here is Carrier Sense Multiple Access with Collision Detection or CSMA/CD. This method allows any user to send data at any time. Collisions only occur when two or more users send packets of data at the same time. When a collision is detected, each user backs off for a random time interval before re-sending the data packets. Special algorithms are employed to reduce the likelihood of collisions thereby ensuring a good bus bandwidth even at times of high traffic density.

A further medium for sending data is over the mains wiring in a building. This solution is particularly attractive when considering the installation of networks into existing buildings (retro-fitting). Here the cost of the additional cabling can be prohibitive so this method incurs no additional cabling costs. The disadvantage is the relatively poor bandwidth available. The EIB specifies a data rate of only 1.2 kbit/s while LON offers 4.8 kbit/s. This may be sufficient for simple control applications but when you consider the bandwidth necessary for internet, speech and video information, it is far too limiting to consider. The actual mains signalling bands are defined by CENELEC, the electrical standards agency of the European Union.

The EIB have specified frequency modulation as the modulation method (Binary Frequency Shift Keying) while LON (at least in Europe) uses phase modulation (Binary Phase Shift Keying). The frequencies used by the EIB are in the B-Band (95 kHz - 125 kHz) or for LON and EHS in the C-Band (125 kHz - 140 kHz). This description applies to only one of the methods of data transfer allowed for EIB and LON equipment. Other options are available

How near are we to the Intelligent House?

There are many reasons why the intelligent building has not gained widespread acceptance. If you read the sales leaflets of the equipment manufacturers it is clear that all you need to satisfy your need for comfort, security flexibility and energy efficiency is the latest building control system. But when you get down to the nuts and bolts you will most probably find that the functionality that is important to your application is not available or at least not without a greater investment in more equipment. In the early 90's the technique for building bus based equipment was to copy the functionality of the existing installation units and simply add a bus interface. More recently, the principle of shared functionality has been introduced into the design. This refinement process is by no means complete, experimenting with existing systems always creates new ideas and possibilities that influence the next generation of equipment.

The principle of an open system ensures that each manufacturer can compete in the market place with similar products but offering improved features. The effect of this competition is to increase the sophistication of equipment. The same process also applies to the LON equipment.

From the point of view of the installation engineer with the ever-growing numbers of equipment from diverse manufacturers, the situation must be somewhat daunting. A one-week introductory course in Buildings System Management is surely not sufficient. There may be a role here for an independent advisory organisation to be set up that will offer expertise and advice to the installation industry. A further hurdle to overcome if this bus technology is to become acceptable is the natural scepticism of architects, builders and administrators. Not least we also have to convince the user that these systems offer real advantages. Part of the problem here is the user-unfriendliness of past systems. It is important for a user to be able to exercise some control over the equipment rather than being a helpless bystander when the equipment refuses to work properly. Even when the equipment is working perfectly, it is useful to have some method of user input.

A more important aspect of user acceptance may be the system cost. There is a division here between commercial buildings and residential buildings. Most new houses are purchased on a mortgage and the loan institutions are not convinced of the necessity of the additional outlay for a building management system, the cost of which is not likely to be recouped at the time when the house is later sold. By contrast, commercial buildings are likely to see the installation of a building management system as an investment that will repay itself in reduced running costs and maintenance over the building's lifetime.

Current developments

The development causing the greatest excitement for LON and EIB standards is the coupling of the bus with the Internet. LON compliant equipment is already in existence, allowing remote monitoring and control using a normal web browser. The system also allows for remote maintenance and system reconfiguration via an Internet connection.

For the EIB this development is not so far advanced and the Internet connectivity using the software tools ANubis (Advanced Network Technology for Unified Building Integration Services) is partly defined while some equipment is already available. iETS is the internet version of the EIB software tools that provide routines to allow remote monitoring and remote equipment re-programming over the Internet. Software drivers are available to assist developers who intend to program their own applications. Falcon drivers are available for EIB environments and there are many development kits available in the market place for LON applications. With intra and Internet connections there is no reason why the status and control of several buildings cannot be managed remotely.

The Konnex Association that we mentioned earlier is not just a single organisation that looks after the three bus systems, it also specifies the technical development of the systems. Three modes of programming and operation of Konnex systems are planned. The so-called system mode is the same as the previous EIB programming procedure using the EIB programming tools (ETS). This mode offers the greatest flexibility and power but is relatively complex, so requires a certain amount of investment in personnel training. Easy mode requires no start-up tools but offers fewer features than the system mode. It is intended that the installation engineer will use this level. The simplest mode is the auto configuration mode and is designed for the end user of the equipment. The use of this mode is very easy (plug and play philosophy) but offers limited flexibility for the equipment setup. It remains to be seen if these features will make a big impact in the industry.

The future looks bright

The intelligent house is already here. The technology you choose is dependent on the complexity and requirements of each particular installation. A good knowledge of all currently available systems is also necessary. Uppermost in the building planner's mind will surely be the cost of such a system compared with a conventional electrical installation. Here is a bit of a chicken and egg situation in that high initial costs mean that systems are not widely used. Once they are widely deployed, sales revenues should ensure that increased development leads to cheaper and more convenient systems. One method to encourage greater acceptance may be to rent or lease out complex installations similar to those we see today for car leasing or the rental of expensive office equipment. Alternatively, the industry could go down the path taken by the mobile phone operators, here the high cost of producing the phone is partly paid for by the contract that you sign-up for when you purchase the phone. This allows the shop price of the phone to be kept low and is partly the reason for its phenomenal success.

More recently we see the emergence of Bluetooth technology which threatens to replace the wired bus with a radio network. It remains to be seen what impact this will have on the market. What is more certain is that the intelligent building is here to stay and we can look forward to seeing more sophisticated systems accepted as part of everyday life.

(000144-1)

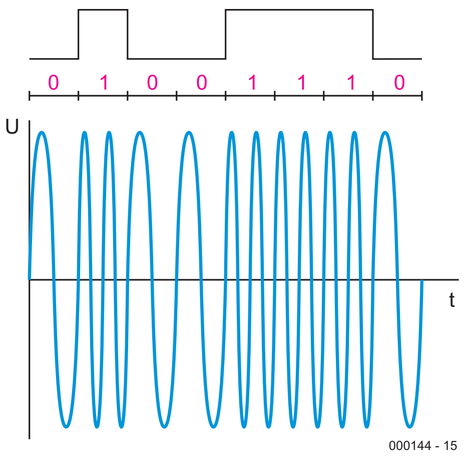


Figure 5. Frequency modulation (EIB).

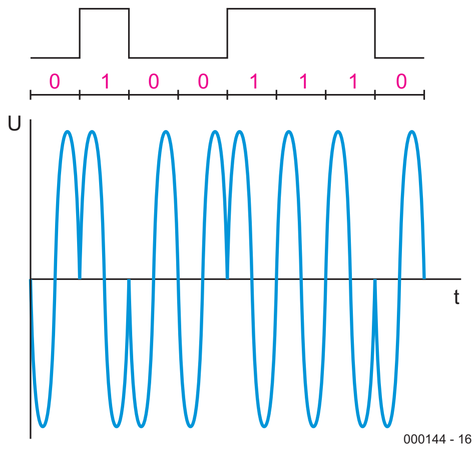


Figure 6. Phase modulation (LON).

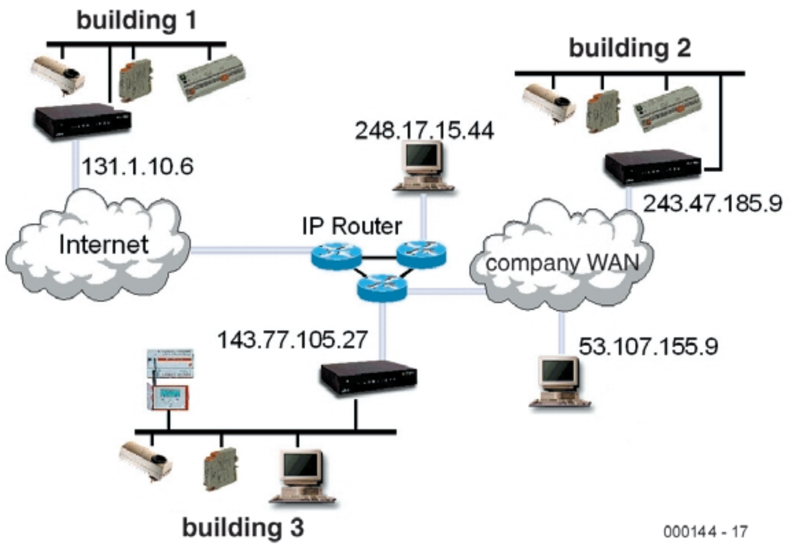


Figure 7. A LON Internet connection.

Some Web Links

Author contact:

peifer@fh-aachen.de

For more information:

http://www.eiba.org
http://www.lonmark.org
http://www.echelon.com
http://www.batibus.com
http://www.ehsa.com
http://www.lcn.de
http://www.peha.de
http://www.bacnet.org
http://www.ukosa.org/

AM Receiver

From DC to 1.8 MHz

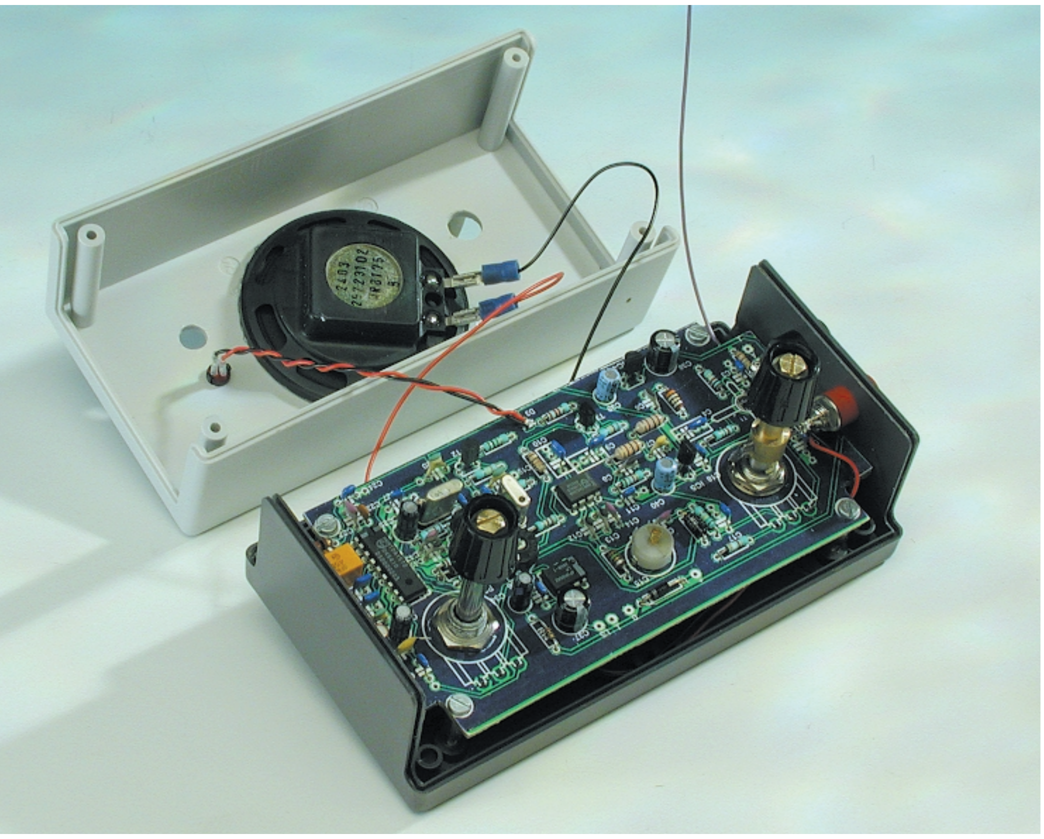
Design by G. Baars

Not all AM receivers are created equal. The receiver described here is definitely not a run-of-the-mill type. The tuning range is not limited to the medium or long wave bands but extends down to extremely low frequencies. This opens up new possibilities.

The receiving range of the receiver described here is continuously variable from DC to 1.8 MHz. That makes this receiver special, since most AM-receivers are only suitable for the long wave band (150-433 kHz) and the medium wave band (approximately 520-1612 kHz). What does this widened range have to offer?

Those who live near the coast will especially appreciate the extra section from 1600 to about 1800 kHz. Several coast guard stations operate in this range and these often provide very interesting information, usually weather conditions.

At the low end of the tuning range, particularly the so-called VLF-band from 9 to 148.5 kHz enjoys increasing interest. The activity within this range includes, among other things, communications with submarines. Several services also operate on these super low frequencies, including teleprinter (RTTY) stations, meteorological services and the famous time transmitter DCF77 (on 77.5 kHz). For the enthusiast, the VLF-band is a fascinating range. **Table 1** offers a sampling of the stations that operate in this band.



Double Superhet

First a brief overview of the receiver. Because of our requirements regarding sensitivity and selectivity, the receiver has been designed as a double superhet. For those who would like to refresh their memory of what a superhet, or more precisely, superheterodyne receiver, exactly is, are referred to the sidebar.

For the antenna, a very short vertical was selected; a length of 10 cm suffices!

Tuning of the receiver is achieved with a potentiometer and a varicap diode. The filters employ only standard ceramic filters and ready-made E12-chokes. There is no need to wind any coils! A lot of people will normally shy away from winding coils for themselves. They will applaud the fact that, in this respect, the receiver is very easy to build.

In addition, it can be reported

that the receiver is quite integrated, consisting mainly of three ICs, supplemented with a handful of standard parts. And finally, because the adjustment is limited to a single trimmer, anyone with at least moderate soldering skills can be expected to successfully construct the receiver on the specially designed PCB.

Blocks in a row

We limit ourselves first to the description of the block diagram, because this way it will quickly become clear how all the pieces of receiver work together.

As shown in **Figure 1**, the received signal, with a frequency of 0 to 1.8 MHz, arrives at an amplifier stage, continues on to be filtered and is fed to a mixer. Here, it is mixed with the signal from a variable oscillator with a frequency which is 10.7 MHz higher than the received signal. This variable oscillator is used to tune the receiver.

The difference signal resulting from the mixing operation has a fixed frequency of 10.7 MHz (the first intermediate frequency). This signal is, once again, amplified and filtered and applied to another mixer. This time with a signal from a fixed oscillator at 10.245 MHz. The mixer difference signal (the second intermediate frequency) has a frequency of 455 kHz and is also filtered, amplified and finally demodulated. The signal then arrives, via a low pass filter, at an audio frequency amplifier that raises the signal to loudspeaker level.

The line that starts behind the demodulator and runs back, via a low pass filter, to the intermediate frequency amplifiers is part of the AGC (automatic gain control) circuit. This controller ensures that during the reception of strong broadcast stations, the amplification of the intermediate frequency amplifiers is reduced so that strong stations and weak stations will appear nearly equally loud.

Schematic

It is now time to look at how the various blocks are realised in practice. **Figure 2** depicts the complete schematic diagram for the receiver.

A dual gate MOSFET (T1) is used for the input amplifier. The most important function of this stage is impedance matching – this is because we choose a very short and therefore high impedance antenna. This is followed by a low pass filter, which eliminates signals above about 2 MHz. This filter, consisting of L2, L3/C6, C7 and C8, has been designed in such a way that it provides an extra 'notch' at the second intermediate frequency of 455 kHz. Practical experience shows that some stations have significant residual field strength at this frequency.

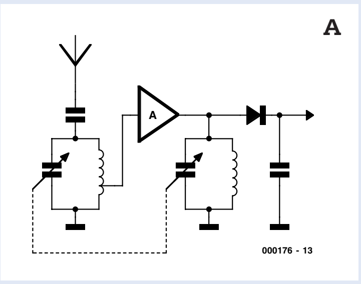
As can be observed, the filtered signal is applied to pin 1 of IC1. This IC (type

Table 1 9.0 to 148.5 kHz Some Utility Stations

Freq. (kHz)	Call	Description
18.3	UNID	Teleprinter
53.6	RTO	Moscow Meteo
60.0	MSF	Rugby TS
75.0	HBG	Nyon TS
77.5	DCF77	Mainflingen TS
82.8	MKL	RAF Edinburgh
111.3	SOA211	Warsaw Meteo
117.4	DCF37	Offenbach Meteo
129.1	DCF49	BMPT Bonn
139.0	TBA	TN Ankara
147.3	DDH47	Hamburg Meteo

Superheterodyne

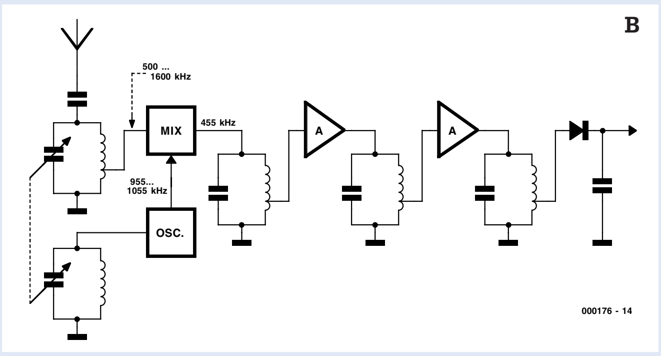
The earliest receivers consisted of little more than a tuneable RF stage for transmitter selection and a detector. When the number of broadcast transmitters increased, it quickly became clear that a single RF stage was insufficient to separate the stations. To increase the selectivity, a second RF stage was added. Such 'double tuned radio frequency (TRF) receivers' (**Figure A**) were popular for a long time.



But eventually, the double TRF also failed to provide sufficient selectivity. Of coarse, the concept can be extended to three, four or five stages, but all the LC tuned circuits have to be synchronously adjusted. This is not a trivial task.

The arrival of the superheterodyne receiver (**Figure B**) resulted in a huge step forwards. The trick is to mix the input signal with a variable oscillator, which is synchronously adjusted with the RF stage (by using two variable capacitors on one spindle, just as with a double TRF). The oscillator frequency is chosen such that the difference between the input- and oscillator frequencies is constant. In the example of figure A, when the input frequency changes from 500...1600 kHz, the oscillator frequency changes simultaneously from 955…2055 kHz. Therefore, independent of the selected input frequency, the difference frequency (the intermediate frequency) will, after mixing, always be 455 kHz.

The advantages are clear. To increase selectivity, as many stages as are desired may be added after the mixer, without the need to tune them synchronously with the input stage. They can simply be of fixed frequency (455 kHz in this example).



The 'superhet' is often extended to a 'double superhet', by adding another mixer and oscillator. This results in a further increase in performance.

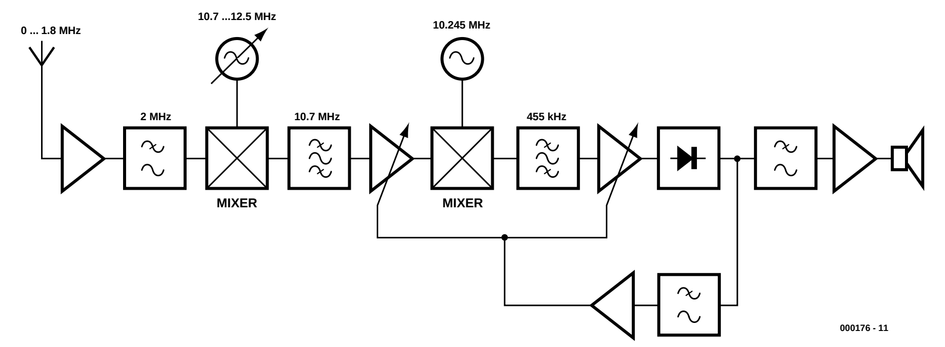


Figure 1. Block diagram of the 'double superhet' configured AM-receiver.

SA612AN) consists of a balanced mixer and an oscillator and is ideal for the first mixing ceremony. The ring oscillator is connected to pins 6 and 7. It is dimensioned such that the oscillator frequency, with the aid of varicap diode D1 and multi-turn potentiometer P1, can be varied from 10.7 to 12.5 MHz. P1, therefore, serves to tune the receiver.

We would like to quickly add that the choice of varicap diode D1 is extremely critical. The MV1401 has a capacitance variation of 30-600 pF when the tuning voltage ranges from 1 to 10 V; a device with different capacitance will result in a change of the receiver frequency range.

A 10.7 MHz band pass filter follows the first mixer. This filter provides the desired selectivity and also removes the undesirable frequencies resulting from the mixing process. We are only interested in the difference signal between the oscillator and the input frequency (12.5 MHz –1.8 MHz = 10.7 MHz) and not the sum (12.5 MHz + 1.8 MHz = 14.3 MHz).

Emitter follower T2 provides impedance matching from the output of the 10.7 MHz filter and the input of IC2. This IC, a TDA1572, takes care of practically all the remaining functions in the block diagram. In fact, the TDA1572 is a complete integrated AM-receiver circuit. It contains, among other things, a mixer, two IF amplifiers with AGC control and a demodulator.

The 10.7 MHz signal is first amplified and then mixed with a (fixed) 10.245 MHz oscillator, resulting in a second intermediate frequency of 455 kHz. This signal is amplified again, then filtered with the aid of a ceramic filter, demodulated and routed to the output via a low pass filter.

The required oscillator signal is generated with the circuit around FET T4. The crystal used here has a screened enclosure, which has to be grounded. The advantage of this external oscillator is that there is no need to adjust it.

Apart from the ceramic 455 kHz filter (X2) and a few compulsory passive components (mostly capacitors), the circuitry around T3 and D3 is the only external circuit. This is a simple tuning indicator, the sensitivity of which can be adjusted with P3.

The demodulated signal is available on pin 9 of IC2. C32 forms part of a low pass filter, after which the signal via C33 and volume control P2 is presented to a small IC audio amplifier. The well-known LM386 is used here; it can provide a few hundred milliwatts at full output to a

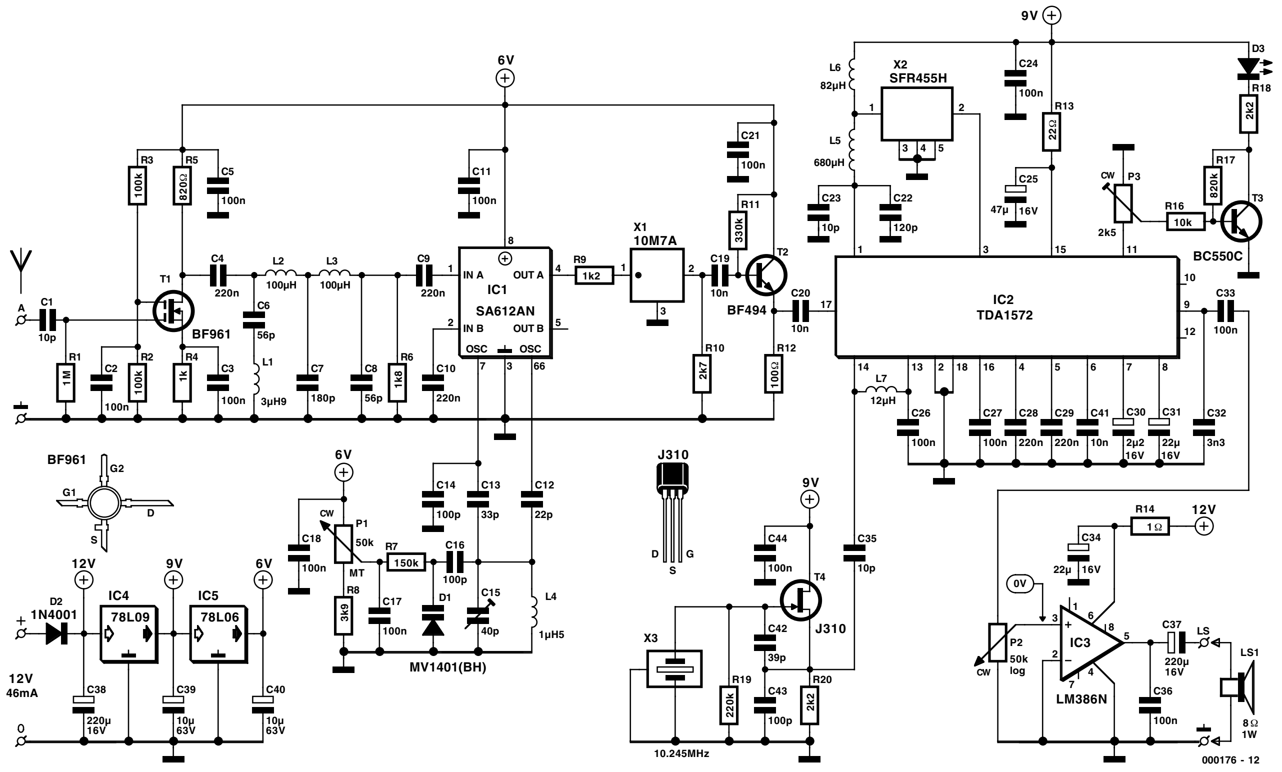


Figure 2. The size of the circuit is small as a consequence of a special AM-receiver IC (IC2)

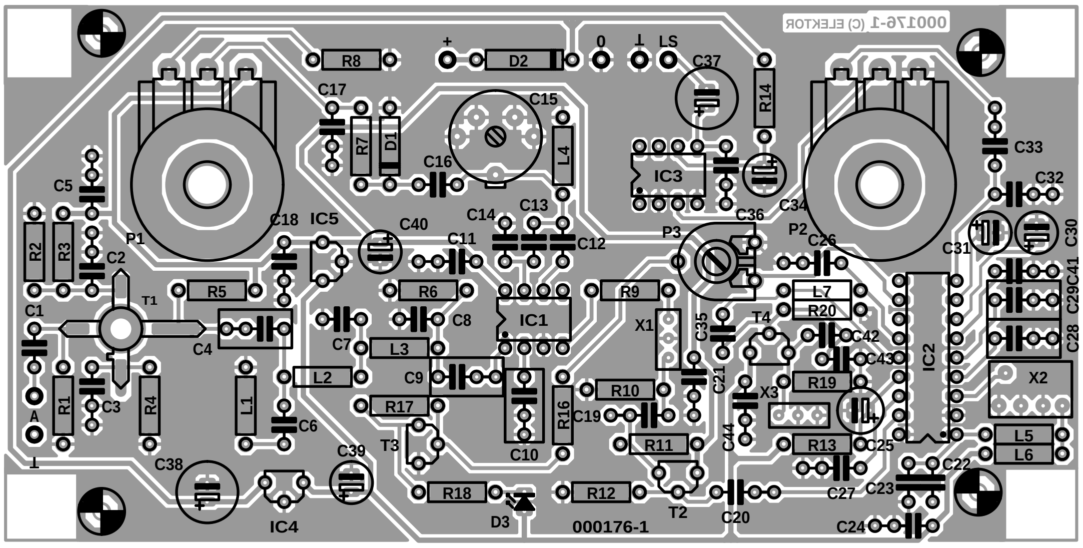
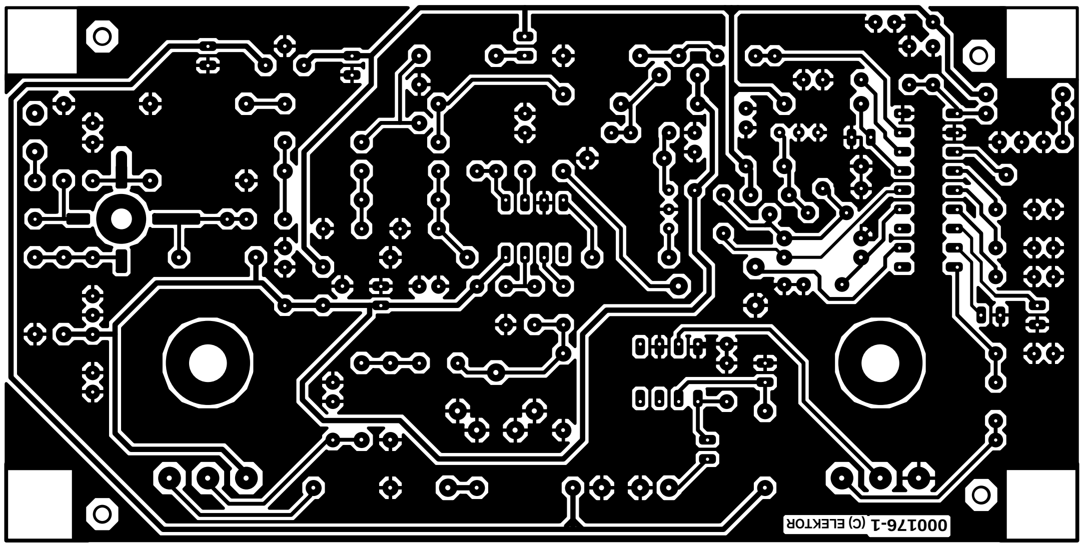
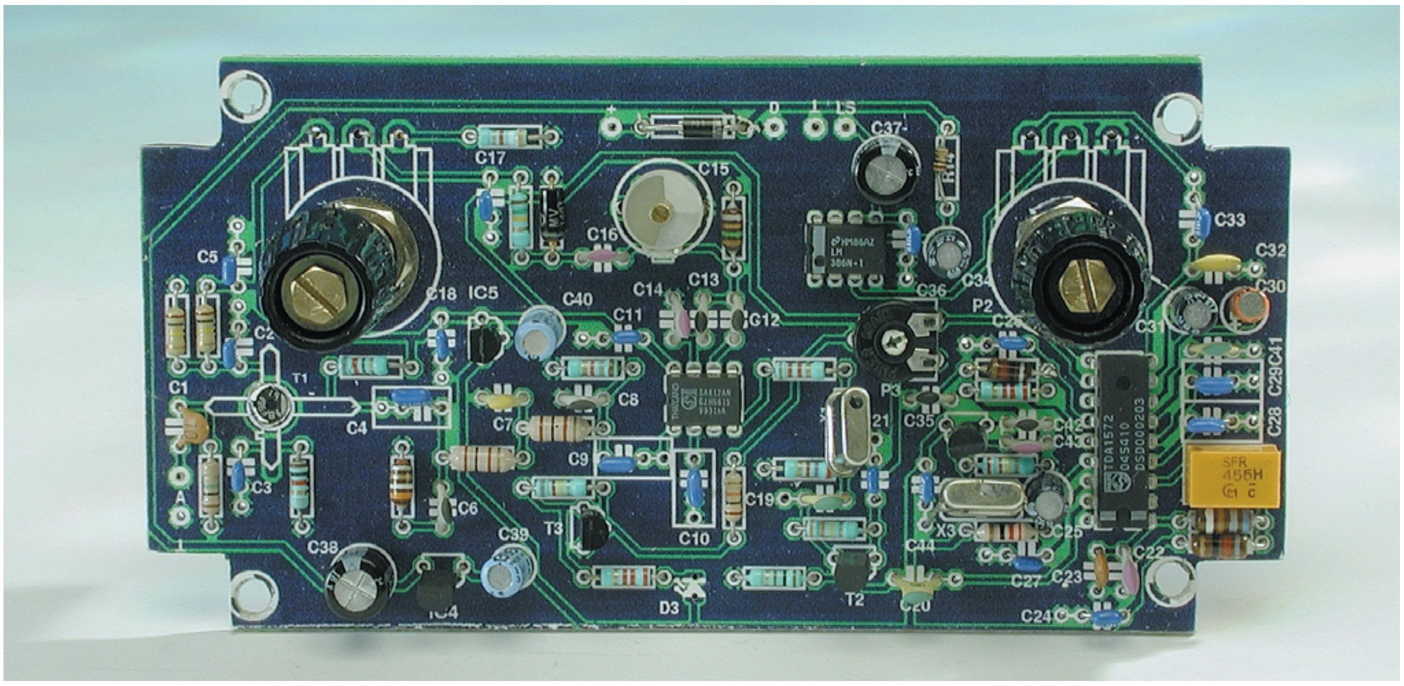



Figure 3. The cleverly designed PCB layout makes construction a simple task.



small 8 Ω loudspeaker. This is sufficiently loud for living room use.

Finally, a few remarks about the power supply. There are three power supply voltages: 12 V, 9 V and 6 V. A normal 12 V mains adapter may be used, the voltage of which is applied directly to the audio amplifier, IC3. Voltage regulator IC4 provides a stable 9 V for IC2, and the remainder of the receiver runs from the stabilised 6 V generated by IC5.

Construction

The construction of radio-frequency circuits is traditionally a little more critical than low frequency or digital designs. However, the construction of this receiver is remarkably easy and the few critical areas have been carefully considered during the PCB layout stage. In this case that means, for instance, short tracks around T1 and the oscillator connections of IC1, decoupling via the shortest possible paths to ground, and generally a lot of copper that functions as a ground plane.

There remain, therefore, very few problems for the builder. It can confidently be said that, if you do a tidy job of the soldering and stick to the parts list of **Figure 3**, very little can go wrong.

Pay careful attention to the polarity of semiconductors and other polarised components. This applies in particular to T1: the short leg with the protruding part (source) connects to C3 and the longest leg (drain) to C4.

The coils are, without exception, standard chokes shaped like resistors — that makes it easy. The input of filter X1 is marked with a black dot; this side connects to R9. X2 possesses an asymmetrically placed pin, which makes getting it wrong almost impossible. Crystal X3 has no centre leg. The third leg is made by (quickly!) soldering a short piece of bare wire to the crystal.

The dimensions of the PCB are such that it fits nicely in, for instance, a Bopla enclosure type E440. Because both potentiometers are also fitted on the circuit board, wiring is a simple affair: only the antenna and loudspeaker need to be connected. The lengths of the potentiometer spindles and the leads of LED D3 need to be selected in such a way as to enable them to protrude through the front of the enclosure.

The power supply for the receiver can be a standard 12 V mains adapter; since the power consumption is barely 50 mA there are no further requirements.

Adjustment and Usage

The adjustment is limited to the calibration of the oscillator range using C15. The simplest method is as follows. Temporarily short circuit the antenna connection (point A) to ground and turn tuning potentiometer P1 fully to the left. Then, slowly adjust C15, starting at the centre position, until a signal is observed; the oscillator is now exactly at 10.700 MHz.

You may also use a frequency counter to adjust the signal on pin 7 of IC1 to 10.700 MHz. The disadvantage however, is that the measuring instrument presents a small load and causes a change in frequency. This can be prevented by temporarily connecting a 10 kΩ resistor from pin 2 of IC1 to ground; this causes a buffered version of the oscillator signal to appear on pin 5 of IC1.

The vertical antenna can now be connected and the test drive can commence. The prototype was found to be very sensitive and selective. In addition to the British broadcast stations, Dutch German, French and Belgian stations were received during the day. During the evening, many Scandinavian and eastern European stations were added to that. In the range below 150 kHz several data stations were heard.

The ideal length of the antenna is somewhere between 10 and 20 cm. A longer antenna is not recommended because of increased interference levels caused by atmospheric disturbances.

(000176-1)

COMPONENTS LIST

Resistors:
R1 = 1MΩ
R2,R3 = 100kΩ
R4 = 1kΩ
R5 = 820Ω
R6 = 1kΩ8
R7 = 150kΩ
R8 = 3kΩ9
R9 = 1kΩ2
R10 = 2kΩ7
R11 = 330kΩ
R12 = 100Ω
R13 = 22Ω
R14 = 1Ω
R16 = 10kΩ
R17 = 820kΩ
R18,R20 = 2kΩ2
R19 = 220kΩ
P1 = 50 k 10-turn cermet
P2 = 50 k logarithmic potentiometer
P3 = 2kΩ5 preset

Capacitors:
C1,C23,C35 = 10pF
C2,C3,C5,C11,C17,C18,C21,C24,C26,C27,C33,C36,C44 = 100nF
C4,C9,C10,C28,C29 = 220nF
C6,C8 = 56pF
C7 = 180pF
C12 = 22pF
C13 = 33pF
C14,C16,C43 = 100pF
C15 = 40pF trimmer
C19,C20,C41 = 10nF
C22 = 120pF
C25 = 47µF 16V radial
C30 = 2µF2 16V radial
C31,C34 = 22µF 16V radial
C32 = 3nF3
C37,C38 = 220µF 16V radial
C39,C40 = 10µF 63V radial
C42 = 39pF

Inductors:
L1 = 3µH9
L2,L3 = 100µH
L4 = 1µH5
L5 = 680µH
L6 = 82µH
L7 = 12µH

Semiconductors:
D1 = MV1401 (Motorola)*
D2 = 1N4001
D3 = LED high-efficiency
T1 = BF961
T2 = BF494
T3 = BC550C
T4 = J310*
IC1 = SA612AN (Philips Semiconductors)*
IC2 = TDA1572
IC3 = LM386N
IC4 = 78L09
IC5 = 78L06

Miscellaneous:
LS1 = miniature loudspeaker 8Ω 1 watt
X1 = 10M7A* ceramic filter
X2 = SFR455H* ceramic filter
X3 = 10.245 MHz quartz crystal
Enclosure: e.g., Bopla type E440
PCB, order code **000176-1** (see Readers Services page)

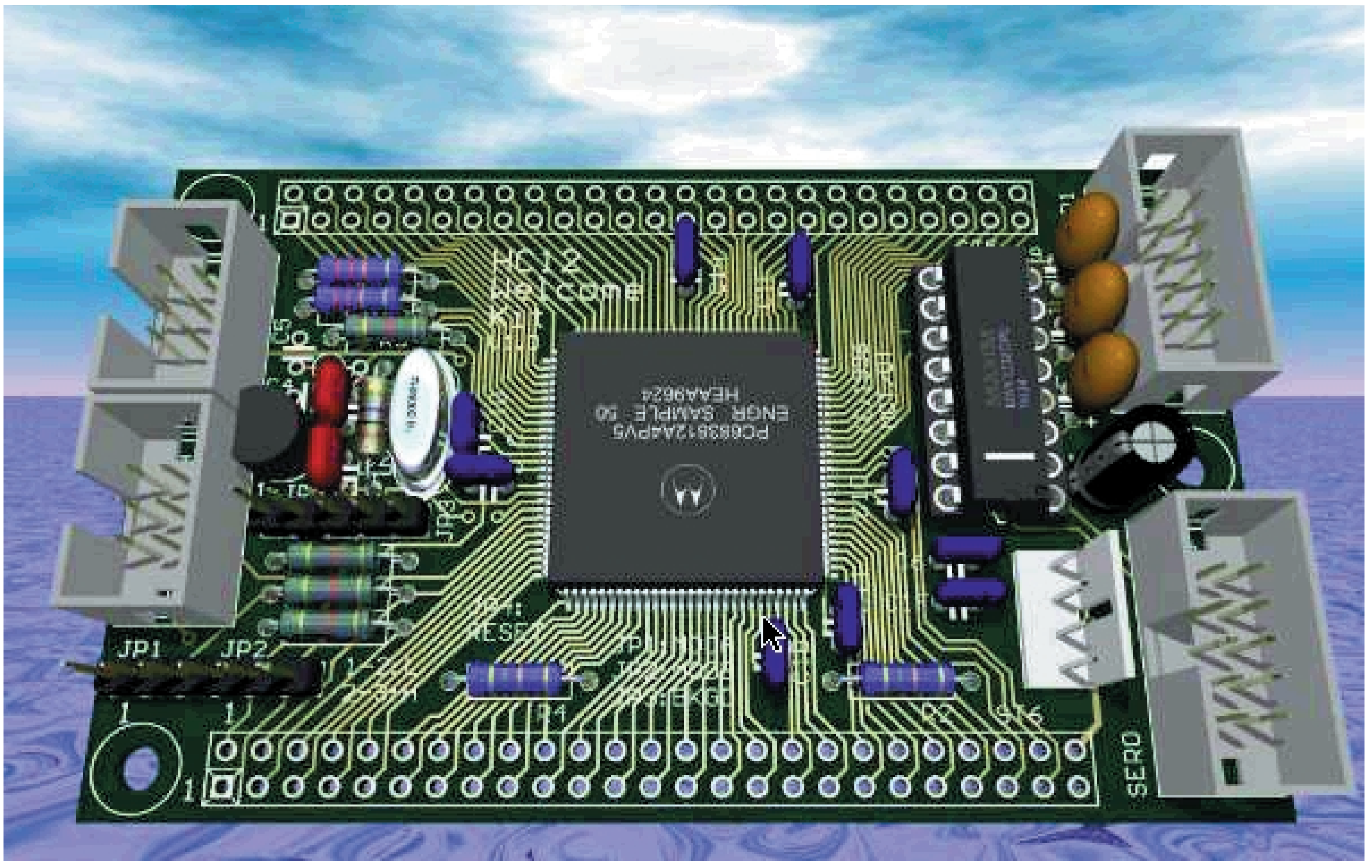
*) suggested supplier:
Barend Hendriksen, website at *http://www.xs4all.nl/~barendh/*
Email: *barendh@xs4all.nl*

Upgrading to the 68HC12 16-bit microcontroller (2)

Part 2: programming and tools

By Oliver Thamm

Having dealt with the hardware in Part 1 of the HC12 article, we now turn our attention to a summary of how to program the HC12, along with a look at the necessary tools.



The HC12 is a typical CISC microcontroller, which means that it has a relatively large number of instructions with relatively complex modes of operation. In general, this type of microcontroller is easier to program in assembly language than are representatives of the RISC camp. The latter type (necessarily) induces the generation of 'spaghetti code', which consists of long and correspondingly incomprehensible code sequences. This does not have to be a disadvantage, as long as the code generation is left to a compiler. However, since we want to start with assembly language, the HC12 is just what we need.
The assembly-language instruction set of the HC12 consists of nearly 200 different instructions — which may sound like a discouragingly large number for a beginner. However, this is by no means a hopeless undertaking, since the instructions are arranged in logical groups and some instructions are not used all that often. If you deal with the problem a step at a time, after a while you will have no trouble making use of this tangle of instructions.

From HC11 to HC12

Previous HC11 users have it especially easy. Since all the familiar HC11 instructions are present in the HC12 set, no initial adjustments are necessary. Existing HC11 programs can be re-translated using a suitable HC12 assembler without generating any error messages. With (mathematical) algorithms, at least, you can save a lot of time by reusing existing program sources. However, the rehosting of routines that address specific control registers of the microcontroller has a lower chance of success. Obviously, a HC11 program that addresses the serial port (SCI) will no longer work properly on the HC12 if the address and meaning of the SCI control register have changed.
The HC12, however, should not be considered to be just a faster, com-

New HC12 Instructions

MOVB #$0D,$1234

In order to write a constant to a particular memory location with the HC11, you always had to do the LDAA/STAA dance. This meant that the value in the accumulator was always lost afterwards! One remedy was to bracket the instructions with PSHA and PULA. Fortunately, the HC12 comes with the long-desired Move instruction, which shortens this procedure.

DBNE B,Loop

Loops are everywhere! With the indicated 'Decrement and Branch if Not Equal' instruction, it is very easy to implement this important basic structure, which you will find in dozens of places in every assembly-language program.

LDAA 1,X+

What is being loaded into the accumulator here? The value that is pointed to by the index register X. Old news, you say? Not at all, since X is also incremented (increased by 1) in the process. This means that you can just about do without the old HC11 INX instruction – especially when you want to increment by 2, 3, 4 or more!

LBRA Label

With the HC11, you could always jump using BRA – but always too short. LBRA eliminates the 8-bit jump region. Of course, this instruction needs one more byte.

EXG D,X

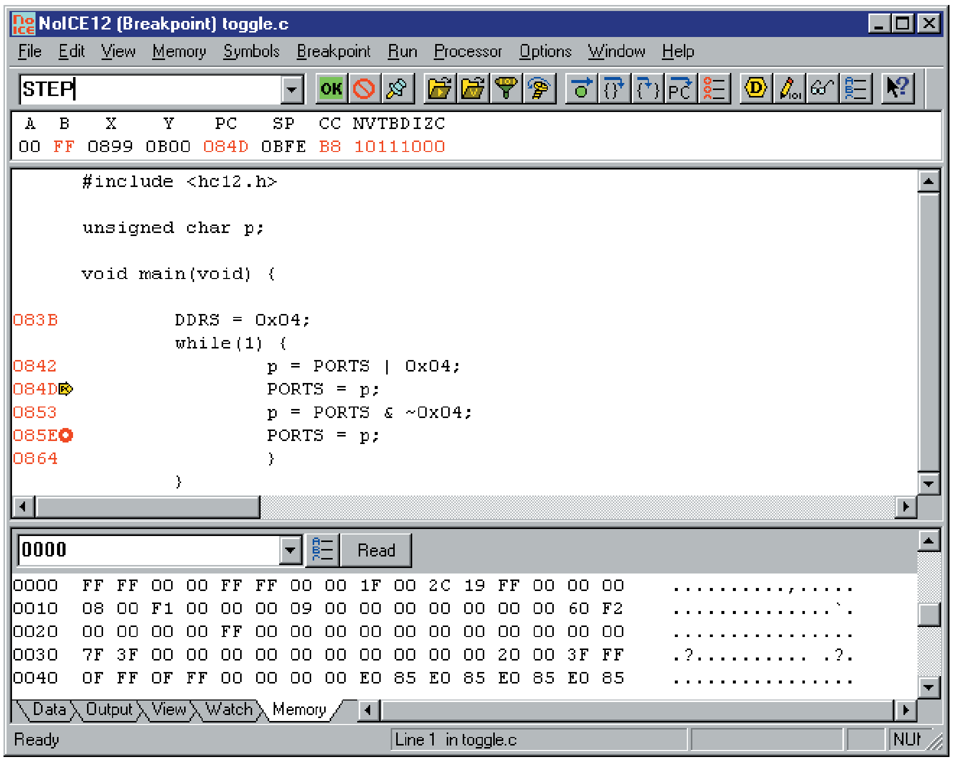
Do you know XGDX? This is the same thing. What's the advantage? EXG works with any combination of registers; you can even mix 8-bit and 16-bit registers!

PSHD

Now you can make PSHB, PSHA faster. PSHD does the whole 16 bits in one go.

SEX

This is included to boost the sales of the CPU12 Reference Manual. Besides that, the 'Sign EXtend' instruction is also handy in C compilers.



```
 A  B    X    Y   PC   SP  CC NVTBDIZC
00 FF 0899 0B00 084D 0BFE B8 10111000
```

```
      #include <hc12.h>

      unsigned char p;

      void main(void) {

083B          DDRS = 0x04;
              while(1) {
0842                  p = PORTS | 0x04;
084D                  PORTS = p;
0853                  p = PORTS & ~0x04;
085E                  PORTS = p;
0864                  }
              }
```

```
0000
0000  FF FF 00 00 FF FF 00 00 1F 00 2C 19 FF 00 00 00   ..........,.....
0010  08 00 F1 00 00 00 09 00 00 00 00 00 00 00 60 F2   ..............`.
0020  00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00   ................
0030  7F 3F 00 00 00 00 00 00 00 00 00 00 20 00 3F FF   .?.......... .?.
0040  0F FF 0F FF 00 00 00 00 E0 85 E0 85 E0 85 E0 85   ................
```

Figure 1. NoICE – a typical example of a graphical BDM debugger. ▶

Programming Tips and Tricks

If the program is finally complete, but it still doesn't work properly, then it's often hard to know what to do. However, there are a few particularly common errors or problems. The following are the five most important ones with HC12 programming.

COP

Many programmers find themselves at a loss if the code works perfectly in the monitor program or debugger, but then shows no signs of life when running 'stand-alone'. The most common cause is that the watchdog timer (COP) has been activated after the reset. Even if you don't use this feature, you can't ignore it. At the very least, you have to disable it immediately after the reset.

Program Counter

Following a reset, the CPU fetches a word from addresses $FFFE and $FFFF. This word contains the reset vector, which determines where the CPU starts with the execution of machine instructions. If nothing at all works in your program, you should immediately bring to mind the helpful existence of the reset vector.

Stack Pointer

The stack pointer is used by the CPU each time a subroutine is called. The CPU saves temporary data in this memory region. If undefined crashes occur (especially after a period of successful programming effort), you should have a close and critical look at the stack region. You should check the following: (a) is the stack pointer initialised for the correct RAM region, (b) are there overlaps with other uses of the RAM and (c) is the size of the stack adequate?

Interrupts

Interrupts can suspend the operation of a running program asynchronously, or in other words, at almost any arbitrary location. Problems only occur if the interrupt routine never returns to the main program. If you suspect that this is happening, you should without fail immediately check two things: (a) does the interrupt routine conclude with the correct instruction (ISR) and (b) are local interrupt flags correctly cleared in the routine? If the local interrupt flags are not reset, everything runs in a circle — the interrupt is still pending on completion of the ISR, and everything starts all over again from the beginning.

The monitor versus the interrupt vectors

If a monitor program is used in the target system, it usually 'blocks' the interrupt vectors located at the end of the address space. In order to allow these vectors to still be used by the application program, it is common practice for the monitor to move the vectors (that it does not use itself) into a RAM region. Here the user can enter 'pseudo interrupt vectors', each of which consists of a three-byte jump instruction of the type *JMP Addr*. This technique should be well known to anyone who is familiar with the HC11, where it is used in the Special Bootstrap mode.

Calling the AS Assembler

```
@echo off
rem
rem -L generates xxx.lst
rem -E generates xxx.log (error list)
rem
c:\as\bin\as.exe %1.a -L -E -g noice
rem
rem -F Moto specifies S-Record output
rem +5 deletes S5 records
rem -r expands the image window
rem
c:\as\bin\p2hex %1.p %1.s19 -F Moto +5 -r $0000-$ffff
if exist %1.p del %1.p
```

patible successor to the HC11. This new microcontroller family includes a lot of additional improvements, such as new instructions and addressing modes. Some of the major improvements are described in the box 'New HC12 Instructions'.

Programming tools

In order to program the HC12, you need a set of tools, as with any other microprocessor. This starts with a PC, with which you can write source programs using a (program) editor and then translate them into machine language using a cross-assembler or cross-compiler. After this, the finished program (the 'executable') must be transferred to the target system. Finally, your self-crafted program may not work properly right off the bat. This means that it is advisable to have some ideas and aids available for doing debugging.

Everything always starts with the source program. To help you devise programs, you should acquire a good program editor. As to what 'good' means, you should not judge the quality of the software by its price alone. What is more important is that you can personally work well with the software. The author has good experience with PFE (Programmer's File Editor), which is a freeware program from Alan Phillips.

Translators

The next step is to hand over the source text to an assembler, which attempts to find syntactical errors in the program. If everything is error-free (up to this point), you will receive a program in machine language. The S-Record format is the established standard here for Motorola processors. An S-Record file is a text file that represents not only all the code bytes of the generated machine program, but also information regarding the addresses where the various parts of the program are to be loaded.

For the HC12, there are already a hefty number of assemblers available from various producers. If you have a C compiler for the HC12 (for example,

Example program for serial interface

```
;=================================================================;

; File: SCI12.A
; Func: Serial-I/O via the Serial Communication Interface (SCI0)
;=================================================================;

;       CPU 68HC12
;       include “reghc12.inc”


SC0BD   equ SC0BDH
SC0CR   equ SC0CR1


; Func: Initialize SCI (9600 Baud, 8N1, Polling Mode)
; Args: -
; Retn: -
; Dest: D
;
initSCI         ldd   #26                 ; 19200 Bd
                std   SC0BD
                ldd   #$000c              ; 8N1, TE + RE
                std   SC0CR               ; A:B -> SC0CR1:SC0CR2
                rts


; Func: Test if any character available (received)
; Args: -
; Retn: A =  0 (Z = 1) -> no char
;       A <> 0 (Z = 0) -> char available
; Dest: -
;
testSCI         ldaa  SC0SR1              ; read status
                anda  #$20                ; receive data reg
full?
                rts                       ; returns 0, if no data


; Func: Get character from SCI (wait for)
; Args: -
; Retn: A = char
; Dest: -
;
getSCI          brclr SC0SR1,$20,getSCI ; receive data reg
full?
                ldaa  SC0DRL              ; read out data
                rts


; Func: Send a character via SCI
; Args: A = char
; Retn: -
; Dest: -
;
putSCI          brclr SC0SR1,$80,putSCI ; transmit data reg
empty?
                staa  SC0DRL              ; send data
                rts
```

the inexpensive ICC12 from Image Craft), you need look no further, since an assembler is always present in the cross compiler. In this case, you can choose from assembly language, C or a combination of the two.

However, as a beginner you should always start with assembly language, since this is the only way you can directly recognise how the interactions between the hardware and software components of the microcontroller take place. The insights you gain in this way will be very useful when you turn to higher-level languages later on. If you are looking for a suitable assembler, you will quickly encounter the name ‘Alfred Arnold’. This author of the universal assembler AS has supported the HC12 since 1996. AS has two distinctive advantages: it is universal, which means that it can be used for all types of processors (including the HC11 and a whole series of other microcontrollers), and it is available free of charge — which is a persuasive argument for private users in particular.

Writing programs in assembly language is certainly not especially easy, and you will have to gain experience in actual practice. If you do not know anything at all about it, you should spend some time with one or two introductory books on HC11 programming. Everything you learn about HC11 programming can be directly applied to the HC12.

Using the assembler is quite simple, at least in comparison to the development of the source code. The box ‘Calling the AS Assembler’ shows a sample sequence in the form of a DOS batch file that calls the assembler, passes the desired assembler file and finally generates the loadable S-Record file (in a separate step in the case of AS). The illustrated batch file can be called directly from the PFE editor using a hotkey, which makes programming a bit more convenient.

Transferring code

You soon will come to the moment of truth, when you start the finished program for the first time. First, however, the program must be loaded into the memory of the target system. There are several ways to do this.

The first is the classic monitor approach. In this case, a monitor program is present on the microcontroller board. This piece of firmware is located in a small region of the memory of the target system, such as in the EEPROM of the MCU. When the target system is connected to the PC via a serial link, you can see the start-up message of the monitor program. On the PC side, you need a terminal program to display the incoming characters on the screen and to send your keyboard entries to the target system.

The terminal program should also have an Upload function, so that a file can be transferred from the hard disk of the PC to the target system. This function is needed to load the HC12 program in the form of an S-Record file. In practice, this goes as follows: first you start the monitor program, and in the terminal window you see the welcome message. Now you can type the load command on the keyboard (for example, *L<ENTER>*). Following this, the program announces that it is ready to accept an S-Record file. You then start the Upload function in the terminal program and select the desired file, and the PC then sends this file to the target system. When the end of the file is reached, the monitor returns the system prompt as notification, and the program is now located in the memory of the HC12 system.

The only critical factor with regard to selecting a suitable terminal program is that it allows the timing requirements of the target system to be accommodated. Setting the data transfer rate (baud rate) should certainly not be a problem, but it should also be possible to activate a waiting function. If transferring the program involves a significant amount of programming time in the target system memory, the terminal program must wait for an acknowledgement from the monitor after sending each line. This ensures that the required programming time (for example, 10 ms per EEPROM location) is not prematurely interrupted by the arrival of the following S-Record line. The old standby TERMINAL.EXE from Windows 3.1 meets this requirement. If you want something more modern, you can download Uwe Alterburg's freeware terminal program OC-Console from Elektronikladen's web site:

(http://www.elektronikladen.de).

Via the back door

We have just seen how the program transfer works when a monitor program is already present in the target system. Clever programmers will probably now ask how the monitor program itself is loaded into the HC12. This question deserves an answer.

Motorola has come up with a special 'back door' for the HC12. It is possible to gain access to all memory addresses via this back door. To a certain degree, it allows you to 'look into' the microcontroller. However, this is not all; it is also possible to actively change memory locations, halt the CPU, modify processor registers and run microcontroller programs in single-step mode.

The operating mode of the HC12 for using this back door is called Background Debug Mode, or BDM for short. The BDM interface consists of a single bidirectional signal line, which is designated BKGD. The signal interactions on the BDM interface are quite complex (see Part 1 of this article), and above all fast. This is also why a BDM pod must always be connected between the software development PC and the BDM interface of the target system. The BDM pod translates the data stream, with regard to contents, signal levels and timing, from BDM to RS232 and vice versa.

There are two forms of debugging solutions that work via the BDM. The first of these is based on a command-line oriented approach. Here a terminal program is used to control the pod, and thereby the attached target system. This is all very similar to the previously described monitor solution for program downloading, with one critical difference: the monitor firmware runs in the pod instead of the target system, so it does not utilise any resources of the target system! Since the limitations of the target system do not have to be taken into account with regard to the size of the code, it is possible to implement very powerful BDM-based monitor programs.

A typical example of this sort of BDM tool is B32EVB from Motorola. This is a palm-sized circuit board with an RS232 connector (9-pin sub-D) and a BDM output in the form of a standard 6-pin header. Amusingly enough, an HC12 is also used on this board (an HC912B32), but this is not relevant to its function as a BDM pod. Lately, there have frequently been supply problems with this quite inexpensive Motorola tool, but the ZWERG32 from Elektronikladen can be used as an alternative.

And through the window

The second category of BDM tools consists of debuggers with graphic user interfaces. These mostly come as Windows-based software that uses a pod with a serial or parallel connection for communication with the target hardware. With the graphic approach, it is naturally possible to display a large amount of data simultaneously in a much better manner, while at the same time hiding the details of the BDM control from the user. As a rule, memory regions can be displayed in the form of hex dumps, while the current status of the processor registers is displayed in a separate region of the window. In addition, another window shows the contents of the program memory in disassembled form. Breakpoints can be entered in this window, and program execution can also be followed one step at a time (see **Figure 1**). Of course, all this convenience comes at a price. Professional-level graphic debugging tools (for example, ZAP from Cosmic) run to a few thousand pounds. This is naturally fully justified if you wish to have recourse to the largest possible number of powerful programming features. If your demands are somewhat more modest, however, you can find attractive debugging solutions for the HC12 at significantly lower prices. The author has had good experience with StingRay (written by Sid Price), among others, and with John Hartman's NoICE Debugger.

Summary

It is worthwhile to take the plunge into HC12 programming for two reasons. First, the Motorola 16-bit family offers solid performance without placing excessively high hurdles in front of the programmer with regard to the complexity of the microcontroller. Secondly, all imaginable software tools are available, including ones that can be obtained as freeware, shareware or 'low-price' ware. The only other thing you have to factor in is the cost of a microcontroller board. The 'serious' user should also include a BDM debugger in his toolkit.

You can find links to the HC12 software mentioned here at the author's website (http://hc12web.de), along with a vast amount of additional information and resources relating to the HC12.

(000077-2)

TPS60100

Charge Pump for 3.3-V Systems

by G. Kleine

Inside the latest state of the art DC/DC converter from Texas Instruments, two synchronous charge pumps work at 180 degree phase shift on one output. This clearly reduces the output ripple voltage.

The TPS60100 has been designed especially for battery operated equipment using 1.8 V to 3.6 V batteries (two NiCd or NiMH cells). It delivers an output current up to 200 mA at 3.3 V and for this it needs only four external (SMD) capacitors. In this design there is no need for any over dimensioned, complicated or hard to come by chokes.

The IC is contained in a thermally improved TSSOP housing, what the company calls a PowerPAD. By using a heatsink, a power loss of up to 25 W can be reached.

Internal circuit of the Tandem charge pump

The inner working of the TPS60100 can be seen in **Figure 1**. Both Charge pumps work with High Power MOSFET switches, whereby the pump capacitor C1F, in the first half cycle, is combined with T12 and T13 and C2F respectively with T22 and T23, with IN and PGNG linked together. Thus the capacitors are charged to the input voltage. For that to be achieved, MOSFETs T11/T14 and T21/T24 should have a high resistance.

In the second half cycle, the transistor pairs conduct, to enable the pump capacitors to be connected to IN and OUT, whereby T12 and T13 (T22 and T23) are held at high resistance. The voltage at the pump capacitor stores the charge of the input IN potential. Theoretically this gives rise to an output voltage twice as large as the voltage at the input IN. The charge distributes itself during this cycle phase between the pump capacitor CxF and the output capacitor Cout.

If both charge pumps were to work in push-pull then the output would be charged every half cycle.

Consequently, a constant output current can flow, in comparison to other state of the art single phase charge pump ICs, where the current can only flow during the charge dissipation phase. If the pump capacitors are to be charged from the IC supply then the output capacitor has to supply the current flow. This causes a ripple voltage. Thus a push-pull configuration of the TPS60100 clearly reduces the ripple voltage effect. For the designs discussed further on, Texas Instruments state typical values of 5 mV$_{pp}$, which can still be further reduced using output filters.

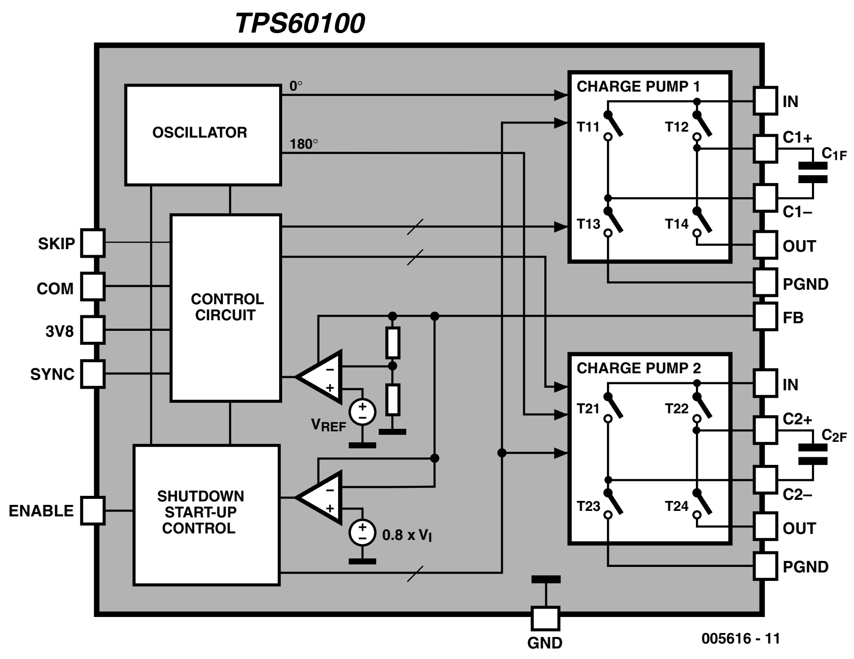


Figure 1. Internal circuit diagram of the TPS60100.

In order to control the output voltage, there is a feedback pin FB, which allows the output voltage to be compared with an internal reference voltage and then via the *Control Circuit* further regulates the charge pumps. Subsequently it is feasible to skip charge cycles in a controlled manner if too large an input voltage appears, by applying the Pulse Skip configuration (SKIP=IN). Equally possible is the Constant Frequency configuration (SKIP=GND) where the internal MOSFETs are so arranged that the output voltage remains at the chosen value.

The block *Shutdown/Start-Up Control* controls the IN and OUT switch sequencing by controlling ENABLE and driving the input voltage high. When switched, both MOSFETs T12 and T14 (T22 and T24) conduct, so that the output OUT is linked to IN. This allows the output capacitor Cout to directly charge itself from the input voltage. At 0.8 V_{IN} the *Start-Up-Control* finally frees up the charge pumps. This is aranged by using the lower comparators shown in Figure 1. Thus the TPS60100, even with a load resistance as low as 16 Ω, can still be relied on to start oscillating.

Finally the DC/DC-Converter also possesses the *Undervoltage-Lockout*-Function. With an input voltage less than 1.6 V the chip automatically shuts itself down.

TPS60100 Mode Operation

The charge pump IC allows a wide variety of operational configurations using the five control inputs.

- **ENABLE**: activates DC/DC-converter
- **SKIP**: switches between Constant-Frequency and Pulse-Skip mode
- **3V8**: switches between regulated 3.3 V mode and unregulated 3.8 V mode
- **SYNC**: switches between an internal and external clock signal
- **COM**: switches between Push-pull and Single-end mode

The tandem pump charger is activated by taking the ENABLE Pin high. This is done by linking the ENABLE connection to IN. By applying a pull-up resistor at IN, a LOW (open-drain-output) at ENABLE will disable the converter, so that it consumes less than 1 µA (typically 0.5 µA) and the load is separated from the battery. This method puts the ENABLE to good use, switching the TPS60100 on and off. The internal Oscillator works with a clock frequency of about 300 kHz. The Pin SKIP controls switching between the Constant Frequency mode and Pule-Skip mode. Notably, the Pulse-Skip-Mode economises on battery energy by switching with a low current consumption during the Constant-Frequency-Mode, giving rise to a lower ripple Voltage. Voltage Control is somewhat better in Pulse-Skip mode than in Constant-Frequency mode, where one has to take the variable ripple frequency into account. Connect SKIP to IN and the Pulse-Skip mode is activated. A Low at the SKIP pin switches over to Constant-Frequency-Mode.

The 3V8 Pin allows a choice between the regulated 3.3 V mode and higher 3.8 V voltage output (for an ancillary Low-Drop-out

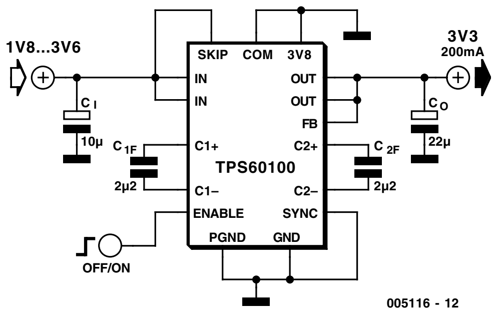


Figure 2. 3.3V DC/DC converter in Push-pull and Pulse-skip mode.

Regulator). Connect the pin to IN and the TPS60100 produces 3.8 V rather than the regulated 3.3 V.

The control input SYNC configuration depends upon whether the internal 300 kHz clock or an external clock is used (fed back into the 3V8 connection). The IC will now remain in 3.3 V mode. A 3.8 V simultaneous output for an auxiliary Low-Drop-Regulator using an external clock is not available. When SYNC is linked to the IN potential then the external clock is enabled. SYNC to ground always means internal clock.

The COM connection allows a fundamental operation which lies between Push-Pull- and Single-ended mode. Push-Pull means that both internal charge pumps are working in anti-phase and thus minimizing the lowest possible ripple voltage. Single-ended mode corresponds to the conventional simple state of the art charge pump. Both charge pumps in the TPS60100 work synchronously, so that one can dispense with one charge capacitor and link the connections of both charge pumps together. At C1+ and C2+ as well as C1– and C2– there is then only one capacitor, thus the number of components and printed circuit board size can be further reduced.

Applications

The TPS60100 circuit diagram arrangement in **Figure 2** depicts a 3.3 V DC/DC converter in Push-Pull and Pulse-Skip mode. During the regulation of 3.3 V the charge pumps are working in low distortion mode by the intended release of pulses. The capacitors are minimally dimensioned. The output capacitor should be about 5 to 50 times larger than the pump capacitor.

The input capacitor improves the input impedance of the circuit. It should be 2 to 4 times larger than the pump capacitor C1F/C2F. It‘s recommended that for the pump capacitors only ceramic types should be used. Tantalum electrolytic capacitors are not suitable for the purpose.

Figure 3 shows the same circuit for Constant-Frequency-mode, where by control of the voltage regulation occurs by means of modulating the MOSFET internal resistance. Here SKIP is connected to ground.

The 3.3V DC/DC converter in **Figure 4** requires the smallest amount of components. It works in Single-ended- and Pulse-Skip mode. The charge pumps work synchronously and not in anti-phase. The pump capacitor must be shared and is simultaneously connected on both charge pumps via C1+/C2+ and C1–/C2–. It should have double the capacitance (here, 4.7 µF instead of 2.2 µF) and a ceramic type.

An example of a circuit with an external clock supply is shown in **Figure 5.** TheTPS60100 works as a 3.3-V DC/DC converter in Push-Pull (Anti-Phase) and Constant-Frequency-Mode. The external clock is fed into Pin 3V8. The frequency must lie between 400 kHz and 800 kHz, where the clock frequency of the charge pump is half the value of the input frequency. The voltage level of the external clock signal must lie below 0.3 Vin for a Low and above 0.7 Vin for a High.

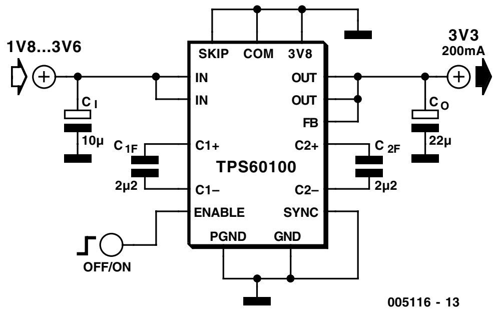


Figure 3. 3.3V DC/DC converter in Push-pull aund Constant-Frequency mode.

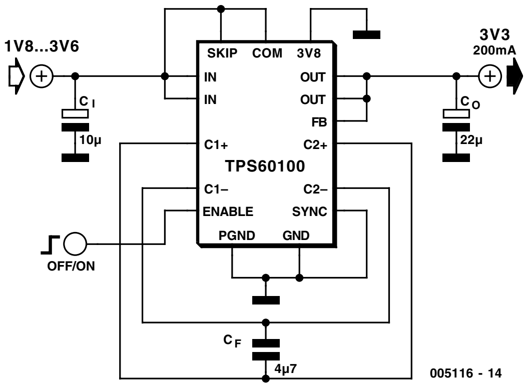


Figure 4. 3.3V DC/DC converter in Single-ended and Pulse skip mode.

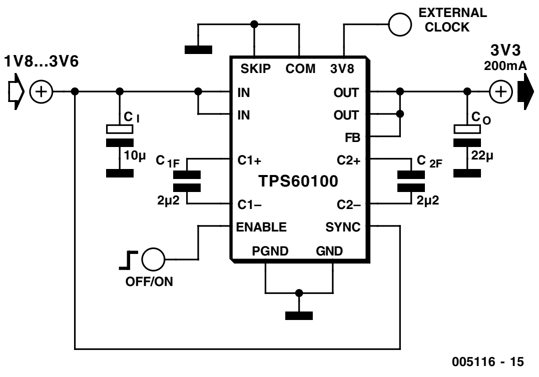


Figure 5. 3.3V DC/DC converter in Push-pull and Constant-Frequency-mode with external clock.

Improved Efficiency

Although the output current of 200mA from the TPS60100 is considerably well above the standard charge pumps, it is possible through direct parallel switching of two TPS60100 to double it, so that 400mA is available for use. As **Fig-**

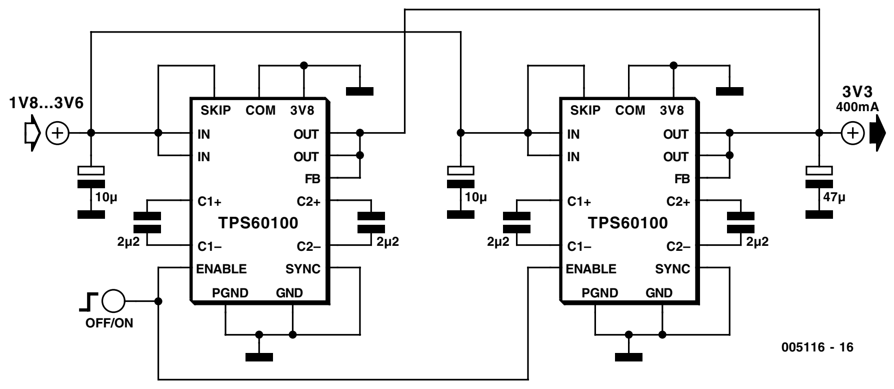


Figure 6. Parallel switching multiplier 3.3V DC/DC converter.

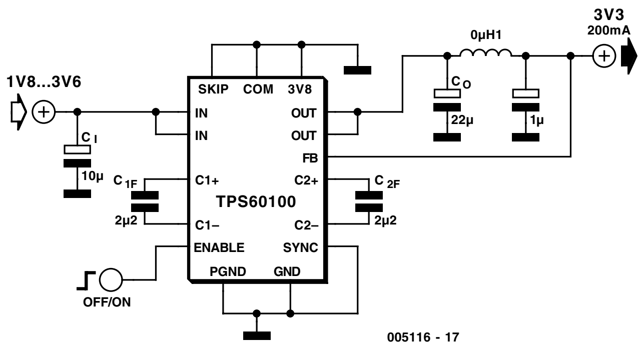


Figure 7. 3.3V DC/DC converter in Push-pull and Constant-Frequency-mode with LC-Filter to reduce ripple effect.

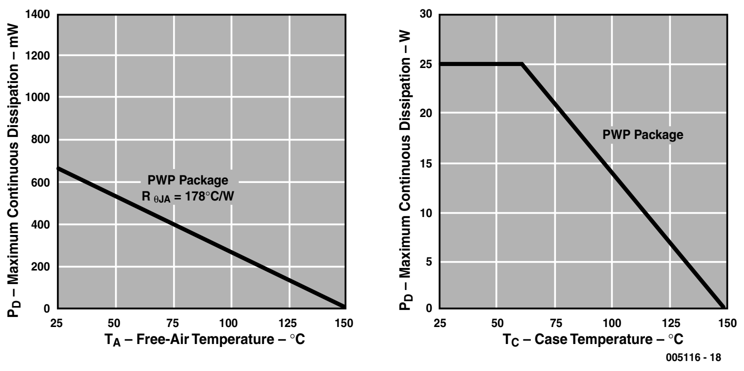


Figure 8. Diagram to calculate maximum power dissipation.

ure 6 shows, the connections ENABLE, IN, OUT, PGND and are all parallel connected. Both ICs have their own input capacitor, which like all other capacitors should be placed as near a possible to the chip. Naturally, both chips must be configured and run in the same operational mode (SKIP, COM, 3V8, SYNC). Further synchronisation of both chips is however not possible. The common output capacitor ought to be double the capacitance of the opposite single switched capacitor (i.e. 47 µF instead of 22 µF).

Reduction of Ripple Voltage

The ripple voltage of standard applications lies round about 5 mV_{pp}. When even this value is too high it is possible, with an ancillary LC-filter, to reduce ripple even further. However this also requires a low drop-off frequency to be selected (for example, 50 kHz), in order to suppress the internal switching frequency of 300 kHz. A possible disadvantage of using this filter is increased voltage loss.

Figure 7 shows a recommended Texas Instruments variation. Here, the LC-Filter is in the feedback loop. The Feedback-connection FB taps off from the load behind the low-pass LC filter. Due to stability reasons a high roll-off frequency (500 kHz) must be selected, otherwise instability can occur.

Note: Maximum Dissipation Loss

To end with, here is the formula to calculate the power dissipation of the TPS60100.

$$P_v = 2{\cdot}I_{out} \cdot V_{in} - I_{out} \cdot V_{out}$$

Here it is assumed that the output current I_{out} far exceeds the internal current consumption of 90 µA. **Figure 8** shows two diagrams which define the maximum possible power dissipation in relation to the ambient temperature and case temperature. Without a heatsink, a power dissipation of only 700 mW is possible. However, with a heatsink the IC can cope up to 25 W , so long as the case temperature can be maintained at 60 °C, using the size of the heatsink (heat resistance). At higher case temperatures the maximum allowable power dissipation is reduced (derating effect)

The efficiency of these charge pump IC's may be computed from the ratio of input power to output power. In practice, values of up to 90% may be obtained.

(005116)

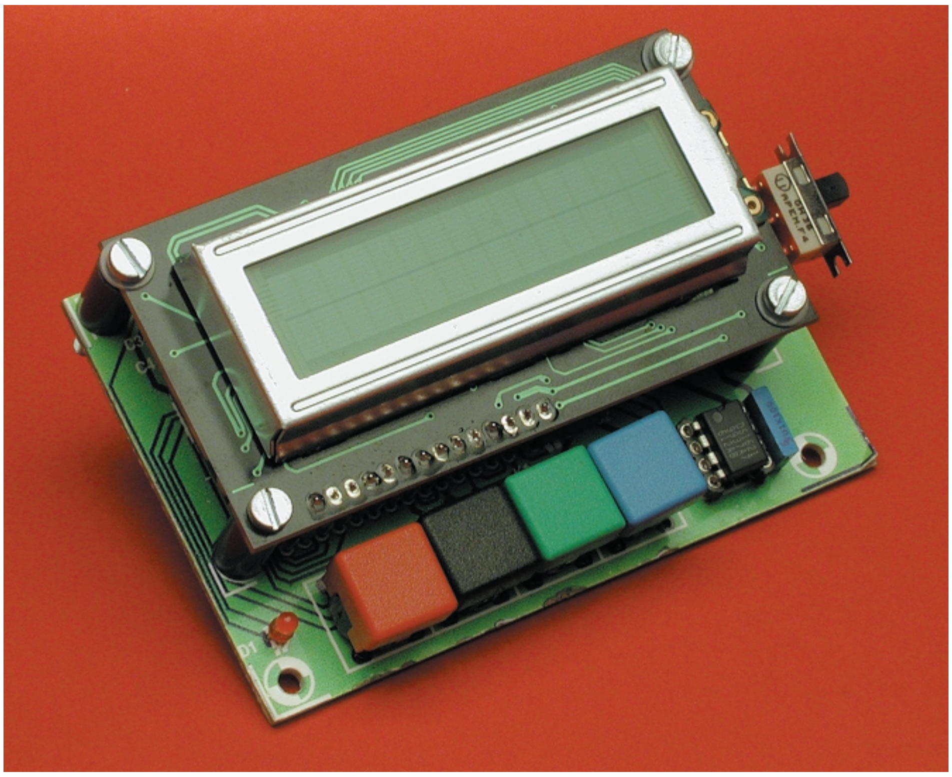
Literature:
Datasheet and further information at *http://www.ti.com/sc/60100*

DS1621 Programmer

sensor operation without a PC

Design by A. Tüchter

The Dallas Semiconductor DS1621 thermometer/thermostat IC can be programmed and read via a two-wire bus. Instead of using a full-blown PC for this purpose, you can use this handy, battery-operated programmer.



The Dallas Semiconductor DS1621 is a general-purpose thermometer with thermostat functions. The features of this device have already been fully described in the article 'Temperature measurements with the DS1621' in the March 2000 issue of *Elektor Electronics*. If you missed this issue, you can find the necessary information on the device data sheet, which you can obtain from www.dalsemi.com/datasheets/pdfs/1621.pdf. Here we will only repeat the essentials. All read and write operations for both the thermometer and the thermostat take place via an I^{2}C bus. Dallas Semiconductor also provides a (free) program for this at ftp://ftp.dalsemi.com/pub/thermal/ds1621k.zip, although this can only be used with the Windows operating system. This may be perfectly OK for experimental purposes, but in everyday life it may be quite cumbersome to use a PC every time you want to read the configuration register, configure the thermostat set-points, or just start a measurement sequence without slope time. Consequently, the author has developed a programmer that is small, inexpensive and above all battery-operated. This device, which is half the size of a standard Eurocard, is based on an Atmel microcontroller and can access all functions of the DS1621. This means that you don't need Windows or a mains supply, and you don't have to lug around a laptop.

The programmer allows you to program a DS12621 that is integrated into an application circuit without switching off the circuit or removing the IC from the circuit. Its operation has been kept simple, but it still offers a wide range of features. Thanks to its design, the programmer can be used not only for programming the DS1621, but also as a complete data logger if it is fitted with its own DS1621. In the latter case, the thermometer IC can either be attached directly to the board or connected remotely via an adapter cable.

A microcontroller and four switches

The schematic diagram in **Figure 1** shows little more than a microcontroller with a power supply, power-

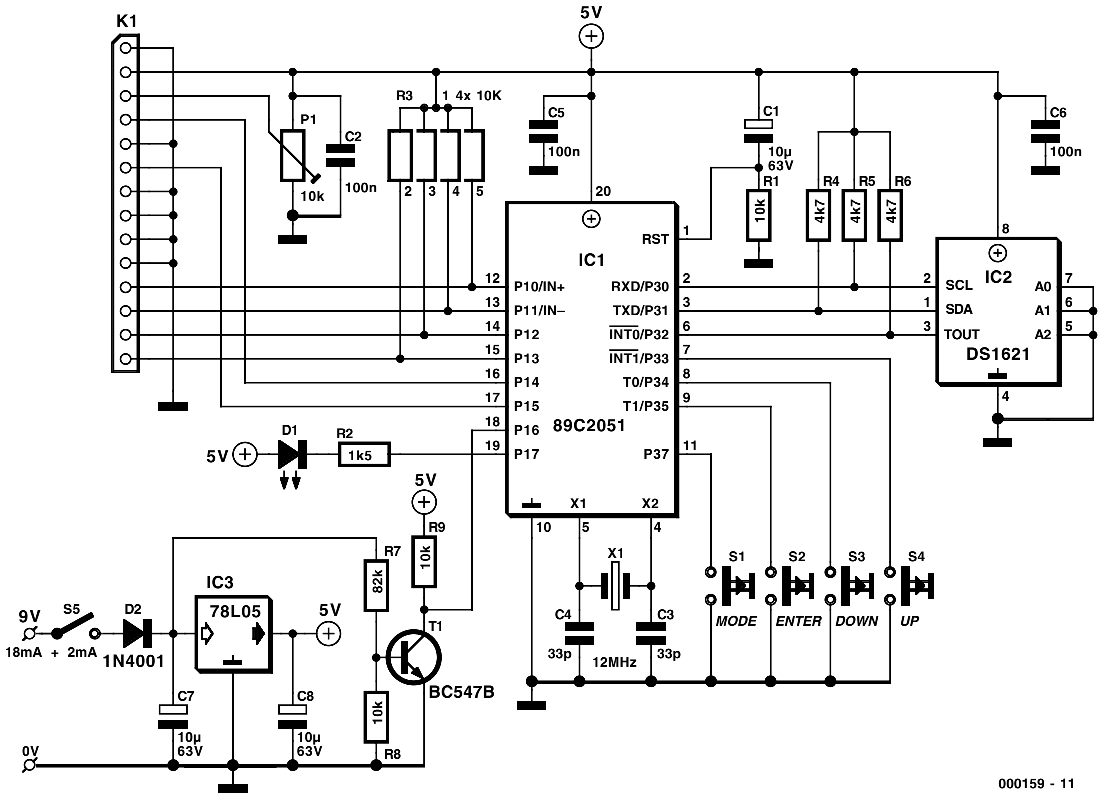


Figure 1. The microcontroller, which reads and programs the sensor IC and drives an LCD module, is controlled by four pushbuttons.

up network, crystal and four pushbutton switches. We'll have a look at the various DS1621 options and adapter cables later on, but first let's look at the controller circuit.

The AT89C2051 microcontroller drives the LCD module via connector K1, evaluates the four pushbuttons S1–S4 and communicates with the DS1621. The power-up network C1–R1 ensures that the microcontroller starts up reliably. The clock is generated by X1, C3 and C4. The controller is connected to a DS1621 (IC2) via ports P3.0–P3.2 and a socket. These three leads are held High by the three pull-up resistors R4-R6. The I^{2}C bus is formed by P3.0 (SCL) and P3.1 (SDA), while P3.1 only serves to receive a notification whenever the temperature rises above or drops below the preset thermostat thresholds during a measurement. C6 suppresses noise impulses on the DS1621 supply line.

The Error LED (D1) is driven via port P1.7, with resistor R2 limiting the current to 2 mA since a low-current LED is used here. Port P1.6 monitors the battery voltage. If the battery voltage (attenuated by the voltage divider R7-R8) is high enough, T1 conducts and a Low level is present at P1.6. As the capacity of the battery becomes exhausted, the input voltage drops and T1 blocks, which causes a High level to be applied to P1.6 via R9.

Only four pushbutton switches (S1-S4) are used to operate the programmer. These are connected to ports P3.3–P3.7. External pull-up resistors are not necessary, since the ports are pulled High internally. The LCD module is connected to the controller via ports P1.0-P1.5. It works in 4-bit mode in order to save port connections. The four data lines (D4–D7 in the LCD module) are pulled High by resistor array R3, while the unused data lines and the R/W control line are tied to ground. The display contrast can be adjusted using trimpot P1.

Power is supplied to the programmer by a 9-volt battery. Voltage regulator IC3 provides a stable +5 V supply from the battery voltage. The voltage for the battery test is tapped off after the on/off switch (S5) and reverse-polarity protection diode (D1), ahead of the regulator. The current consumption is around 18 mA, plus 2 mA for the LED (when it is on).

Connection options

In order to keep the programmer as general-purpose as possible, there are several options for connecting the sensor. The simplest of these is unquestionably inserting the DS1621 directly into the IC2 socket on the circuit board, as shown in **Figure 1**.

However, this socket can also accept a second socket as a plug (as long as the latter has thin, turned pins!), to which a cable with five leads has been soldered as shown in **Figure 2a**. A second socket is connected to the far end of this cable, which may be up to 4 m long. This socket could for example be mounted on a small piece of prototyping board. The connections between the ground lead (pin 4) and the address inputs (pins 5-7) should be made at this socket. In essence, this option amounts to an extension of the connections on the printed circuit board.

The option shown in **Figure 2b** is different. Here the adapter cable connects the programmer to an application circuit, which can be built on the small, detachable section of the printed circuit board. Only the I^{2}C bus and the ground lead are extended in this case, while the DS1621 receives its operating voltage via its own separate voltage regulator (IC5). The output of the DS1621 drives its own Error LED (D3). The IC address is set to

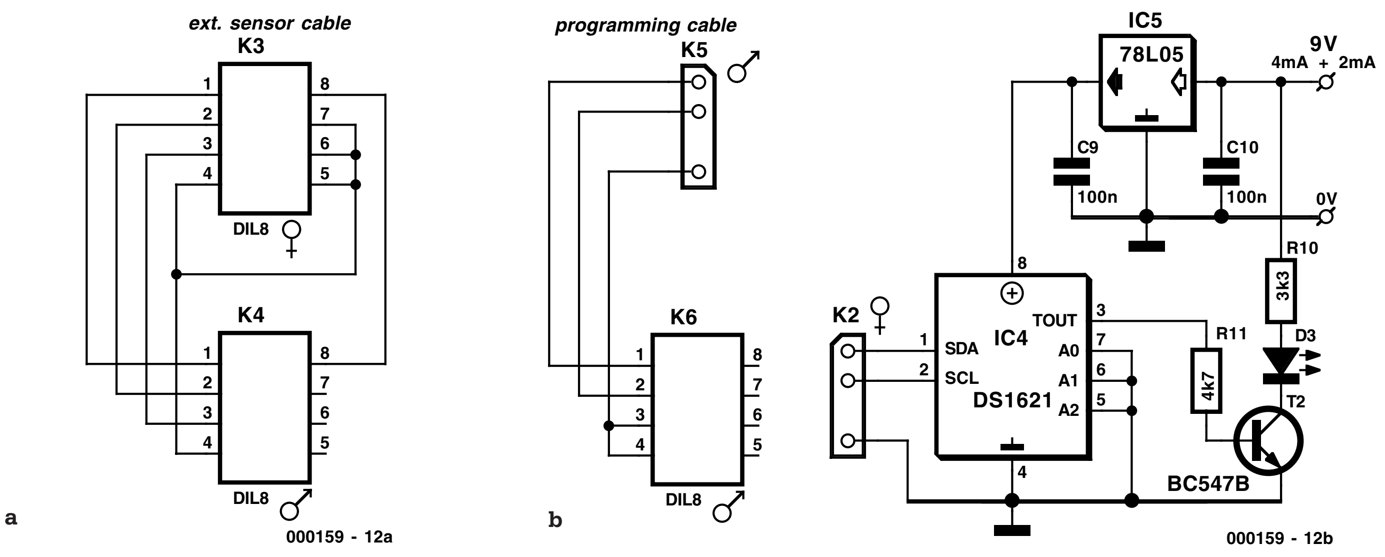


Figure 2. If the sensor is not mounted on the main circuit board, it can be connected via an external sensor cable (a). Alternatively, the sensor can be programmed in an application circuit. This requires a programming cable as shown in (b).

zero in this case as well. Since the IC output cannot be polled, pin 3 of the socket must be connected to ground. A four-pin SIL plug strip is adequate for use at the other end of the cable, with the unused pin (pin 3) broken off. If you solder pin 3 of the matching socket strip closed, it will not be possible to insert the plug the wrong way around.

Operation

The programmer must be connected to a DS1621 before being switched on. After it has been switched on, or after the Enter button (S2) has been pressed, the message *DS1621 – Programmer* appears on the display. After a short delay, communication with the DS1621 is tested. It is important to note that the programmer only works with device address 0. If communication is faulty, *Chip <ERROR> push –Enter–* is displayed and LED D1 blinks. This will always occur if no connection to a DS1621 takes place or the connection is faulty. If this happens, verify the DS1621 address and check the communications link to the sensor. If the connection is successful, the sensor is reinitialised. In this case, the display shows *Chip Testing!*. The TH and TL registers are read, as well as the Config register, and then a temperature conversion is started and read. The programmer reports the successful transfer of all data with the message *Chip <<OK>>*.
The sensor is next configured for continuous measurements and started. If the DS1621 is integrated into an application circuit, the programmer can now be disconnected from the circuit if it is not necessary to make any

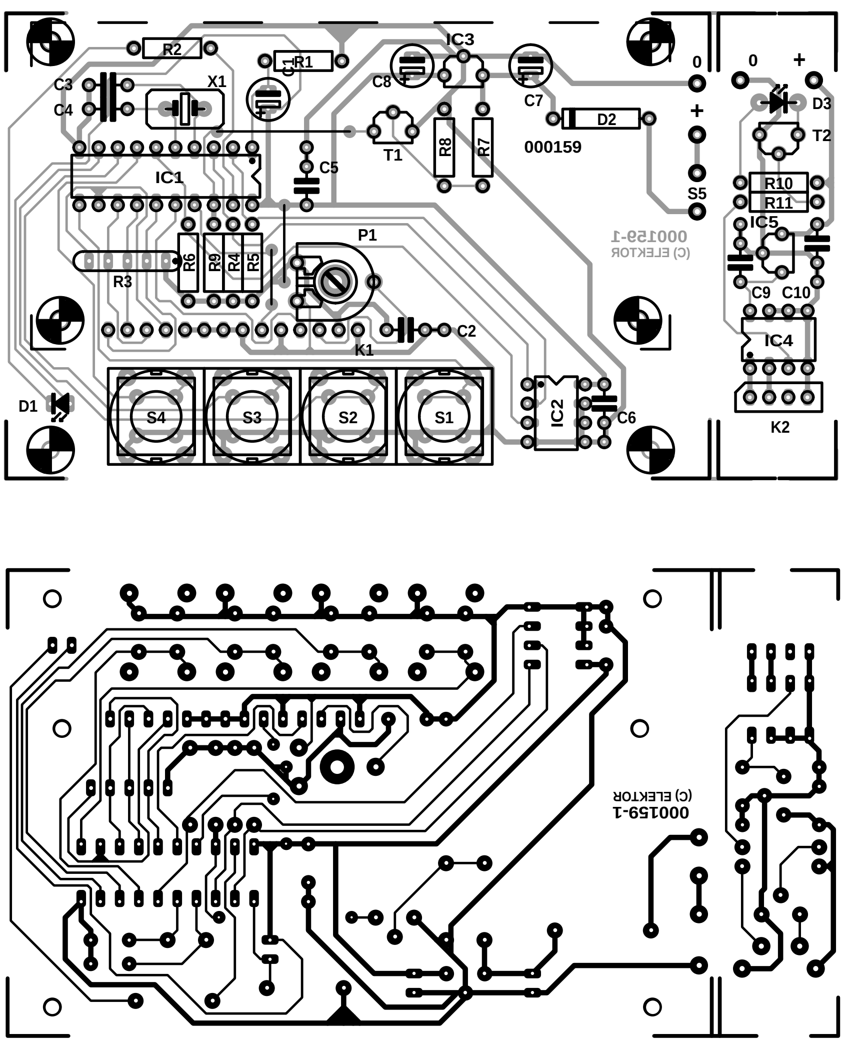


Figure 3. The relatively straightforward printed circuit board for the DS1621 programmer, with mounting holes for the LCD module and a sensor circuit board that can be cut off before the board is assembled.

changes to the settings.
Following the chip test, you reach the menu with the options *Thermometer, TH-Register, TL-Register* and *Output Polarity*. The MODE button (S1) can be used to select the desired menu item, and the selected function can be started by pressing the ENTER button.

Thermometer

In this mode, the programmer operates as a sort of data logger. Here the manner in which the DS1621 is connected does not matter. The first line of the display shows the currently measured temperature. This value is periodically updated. The second line of the display shows the absolute extreme values of the temperature, with the lower extreme on the left and the upper extreme on the right.
In addition to these values, the behaviour with respect to the hysteresis range is also displayed. If the current temperature lies below the threshold value, two dashes are displayed (– –). If the temperature rises to a level equal to or greater than the value in the TH register, *TH* appears in the display. Similarly, if the temperature drops to a level equal to or less than the value in the TL register, *TL* appears in the display. The hysteresis settings can thus be verified. LED D1 or D3, respectively, is on when the I/O output of the sensor is active. This allows the switching behaviour of the DS1621 to be simulated. The thermometer mode can be exited at any time by pressing the ENTER button.

TH and TL registers

This menu item can be used to check or modify the settings of the TH and TL registers

Parts and software via Readers Services and the WWW

A blank printed circuit board for the DS1621 programmer is available from Readers Services under order number **000159-1**. The diskette (order number **000159-11**) contains the source code and a hex file for the controller program. The circuit board layout and both software files can also be downloaded free of charge from the *Elektor Electronics* Internet site at http://www.elektor-electronics.co.uk. If you do not want to be bothered with programming the controller, you can obtain a ready-programmed device from Readers Services under order number **000159-41**.

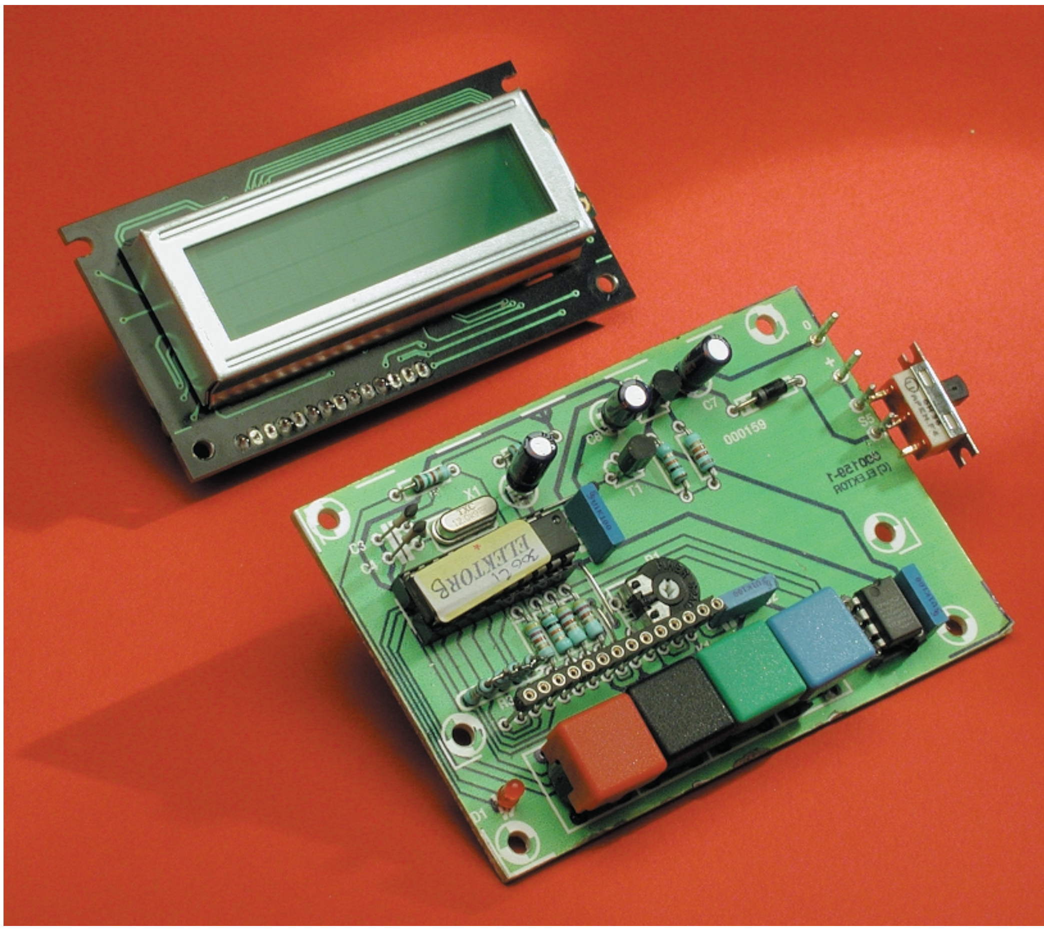
COMPONENTS LIST

Resistors:
R1,R8,R9 = 10kΩ
R2 = 1kΩ5
R3 = 10kΩ 4-way·SIL array
R4,R5,R6,R11 = 4kΩ7
R7 = 82kΩ
P1 = 10kΩ preset

Capacitors:
C1,C7,C8 = 10µF 63V radial
C2,C5,C6,C9,C10 = 100nF
C3,C4 = 33pF

Semiconductors:
D1,D3 = LED, high efficiency
D2 = 1N4001
T1,T2 = BC547B
IC1 = 89C2051-12PC (order code **000159-41**)
IC2,IC4* = DS1621 (Dallas) (Farnell # 670704) with holder
IC3,IC5* = 78L05

Miscellaneous:
K1 = LCD module, 2x16 characters (LTN211R-10, LM16A211, LM16A212)
K3,K4,K6 = 8-way IC socket with turned pins and thin connecting leads (Conrad Electronics #189600)
K2,K5 = 4- way SIL pinheader
S1-S4 = Pushbutton, 1 make contact (Multimec D6-Q or Marquardt, Conrad Electronics #706892 with cap)
S5 = slide switch, 1 changeover contact
X1 = 12 MHz quartz crystal
PC1-PC4 = solder pins
9-V PP3 battery with clip-on leads
PCB, order code **000159-1** (see Readers Services page).
Disk, all project software, order code **000159-11** (see Readers Services page).



and to verify the register flags (which indicate that he limit value has been exceeded). The register name (e.g. *TH – Register*) is shown in first line of the display, and the current register value is shown in the second line. If an exclamation point is shown following the register name (e.g. *TH – Register!*), it means that the flag bit is set, and thus that the limit value has been reached. This menu item can be exited by pressing the ENTER button, but the flags and register values remain as they were. The register value can be modified using the UP and DOWN buttons (S4 and S3). The value changes in steps of 0.5 °C over the range of –55.0 °C to +125.0 °C. If either one of the buttons is pressed, the LED is illuminated to indicate that the value has been changed, which also causes both (!) flags to be cleared. This also causes any exclamation point that may have been visible on the display to disappear. If the previous value is subsequently restored, only the two flags are reset.
Pressing ENTER exits the new programming mode and returns you to the menu.

Output polarity

This menu item can be used to check and modify the I/O output of the DS1621. If *OUTPUT = 1* is displayed, the output is active-High, which means that it goes High when a threshold value is exceeded. Correspondingly, *OUTPUT = 0* indicates that the output is active-Low. The switching behaviour of the output can be changed using the UP of DOWN button. ENTER stores the selected setting in the DS1621 and exits this menu item.
The programmer has a battery voltage monitor that is activated if the voltage is too low. The display then shows *Low Batt. – insert new batt*. The LED also blinks in this case, since this is an error condition. This error message can be eliminated by replacing the battery; it cannot be acknowledged.

Construction

The printed circuit board shown in **Figure 3** provides adequate space for the compact DS1621 programmer. You should first cut off the sensor section on the right hand side. Inserting the components into the board should not present any problems, although you may need to use a horizontal package for crystal X1 and mount C1 flat on the board, so that the LCD module will not have to be mounted too high above the board. The mounting holes match the module shown in the Components List. The K1 connections can be made using either thin wires or an inline pin connector strip with extra-long wire-wrap pins. Don't forget to adjust the display contrast (immediately after completing the assembly, naturally) using P1, via the back side of the circuit board.
Normally, it will be sufficient to solder in a standard IC socket for IC2. However, if you plan to program a large number of DS1621s, it wouldn't hurt to use a test socket (not to be confused with an expensive ZIF socket). With such a socket, the IC can be lifted out of the precision gold-plated socket pins using a wedge. This is beneficial for the service life of the socket and the mechanical symmetry of the IC pins. Unfortunately, 8-pin test sockets are not available, but a 14-pin model can be used if you cut off three pairs of pins.

(000159-1)

CONSTRUCTION GUIDELINES

Elektor Electronics (Publishing) does not provide **parts and components other than** PCBs, fornt panel foils and software on diskette or IC (not necessarily for all projects). Components are usually available form a number of retailers – see the adverts in the magazine.

Large and small values of components are indicated by means of one of the following prefixes :

E (exa) = 10^{18}	a (atto) = 10^{-18}
P (peta) = 10^{15}	f (femto) = 10^{-15}
T (tera) = 10^{12}	p (pico) = 10^{-12}
G (giga) = 10^{9}	n (nano) = 10^{-9}
M (mega) = 10^{6}	μ (micro) = 10^{-6}
k (kilo) = 10^{3}	m (milli) = 10^{-3}
h (hecto) = 10^{2}	c (centi) = 10^{-2}
da (deca) = 10^{1}	d (deci) = 10^{-1}

In some circuit diagrams, to avoid confusion, but contrary to IEC and BS recommandations, the value of components is given by substituting the relevant prefix for the decimal point. For example,
3k9 = 3.9 kΩ   4μ7 = 4.7 μF

Unless otherwise indicated, the tolerance of resistors is ±5% and their rating is ⅓–½ watt. The working voltage of capacitors is ≥ 50 V.

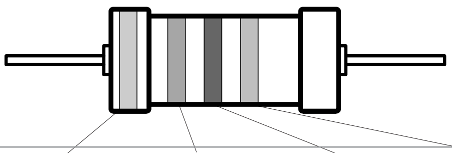
In **populating a PCB**, always start with the smallest passive components, that is, wire bridges, resistors and small capacitors; and then IC sockets, relays, electrolytic and other large capacitors, and connectors. Vulnerable semiconductors and ICS should be done last.

Soldering. Use a 15–30 W soldering iron with a fine tip and tin with a resin core (60/40) Insert the terminals of components in the board, bend them slightly, cut them short, and solder: wait 1–2 seconds for the tin to flow smoothly and remove the iron. Do not overheat, particularly when soldering ICS and semiconductors. Unsoldering is best done with a suction iron or special unsoldering braid.

Faultfinding. If the circuit does not work, carefully compare the populated board with the published component layout and parts list. Are all the components in the correct position? Has correct polarity been observed? Have the powerlines been reversed? Are all solder joints sound? Have any wire bridges been forgotten?
If voltage levels have been given on the circuit diagram, do those measured on the board match them – note that deviations up to ±10% from the specified values are acceptable.

Possible corrections to published projects are published from time to time in this magazine. Also, the readers letters column often contains useful comments/additions to the published projects.

The value of a resistor is indicated by a **colour code** as follows.



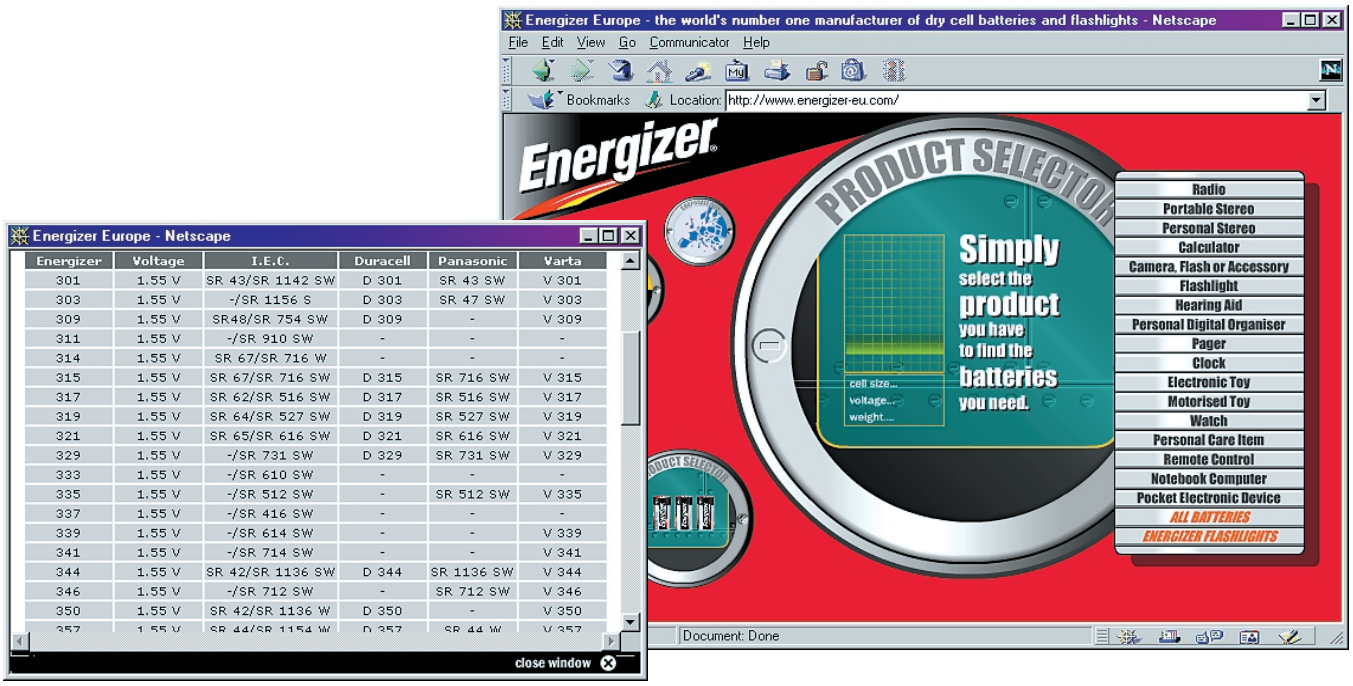
color	1st digit	2nd digit	mult. factor	tolerance
black	–	0	–	–
brown	1	1	$\times 10^{1}$	±1%
red	2	2	$\times 10^{2}$	±2%
orange	3	3	$\times 10^{3}$	–
yellow	4	4	$\times 10^{4}$	–
green	5	5	$\times 10^{5}$	±0,5%
blue	6	6	$\times 10^{6}$	–
violet	7	7	–	–
grey	8	8	–	–
white	9	9	–	–
gold	–	–	$\times 10^{-1}$	±5%
silver	–	–	$\times 10^{-2}$	±10%
none	–	–	–	±20%

Examples:
brown-red-brown-gold = 120 Ω, 5%

Battery Selection Charts

find the right type

Electric torches, watches, pocket calculators, toys and personal stereos, they all need a particular type of battery to do their job. In some cases, it is surprisingly hard to find fresh ones when the original batteries are exhausted. Fortunately, battery manufacturers offer help — via the web.


Energizer Europe - the world's number one manufacturer of dry cell batteries and flashlights - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://www.energizer-eu.com/
Energizer
PRODUCT SELECTOR
Simply select the product you have to find the batteries you need.
cell size...
voltage...
weight....
Radio
Portable Stereo
Personal Stereo
Calculator
Camera, Flash or Accessory
Flashlight
Hearing Aid
Personal Digital Organiser
Pager
Clock
Electronic Toy
Motorised Toy
Watch
Personal Care Item
Remote Control
Notebook Computer
Pocket Electronic Device
ALL BATTERIES
ENERGIZER FLASHLIGHTS
Document: Done
Energizer Europe - Netscape
close window

Energizer	Voltage	I.E.C.	Duracell	Panasonic	Varta
301	1.55 V	SR 43/SR 1142 SW	D 301	SR 43 SW	V 301
303	1.55 V	-/SR 1156 S	D 303	SR 47 SW	V 303
309	1.55 V	SR48/SR 754 SW	D 309	-	V 309
311	1.55 V	-/SR 910 SW	-	-	-
314	1.55 V	SR 67/SR 716 W	-	-	-
315	1.55 V	SR 67/SR 716 SW	D 315	SR 716 SW	V 315
317	1.55 V	SR 62/SR 516 SW	D 317	SR 516 SW	V 317
319	1.55 V	SR 64/SR 527 SW	D 319	SR 527 SW	V 319
321	1.55 V	SR 65/SR 616 SW	D 321	SR 616 SW	V 321
329	1.55 V	-/SR 731 SW	D 329	SR 731 SW	V 329
333	1.55 V	-/SR 610 SW	-	-	-
335	1.55 V	-/SR 512 SW	-	SR 512 SW	V 335
337	1.55 V	-/SR 416 SW	-	-	-
339	1.55 V	-/SR 614 SW	-	-	V 339
341	1.55 V	-/SR 714 SW	-	-	V 341
344	1.55 V	SR 42/SR 1136 SW	D 344	SR 1136 SW	V 344
346	1.55 V	-/SR 712 SW	-	SR 712 SW	V 346
350	1.55 V	SR 42/SR 1136 W	D 350	-	V 350
357	1.55 V	SR 44/SR 1154 W	D 357	SR 44 W	V 357

Batteries — you can't seem to avoid them in electrical or electronic equipment that's not connected to the mains. Increasingly, people will want to go 'mobile' and use as much equipment as possible on the road, without having to plug into the mains for power. Fortunately, notebooks, mobile telephones and lots of other equipment contain batteries that can be topped up many times over. In some equipment, however, rechargeable batteries make little or no sense. For example, if you have a portable radio that's only used occasionally, you're better off using a set of alkaline batteries than any type of rechargeable battery, simply because the latter would be empty (due to self-discharging) after a few months even if the radio is not used at all. By contrast, a good battery will last several years!

Batteries are also commonly used in watches and calculators. These come in a staggering variety so equivalent types may be very hard to find. Fortunately, many battery manufacturers and suppliers have extensive product information available enabling customers to find an equivalent type quickly and easily.

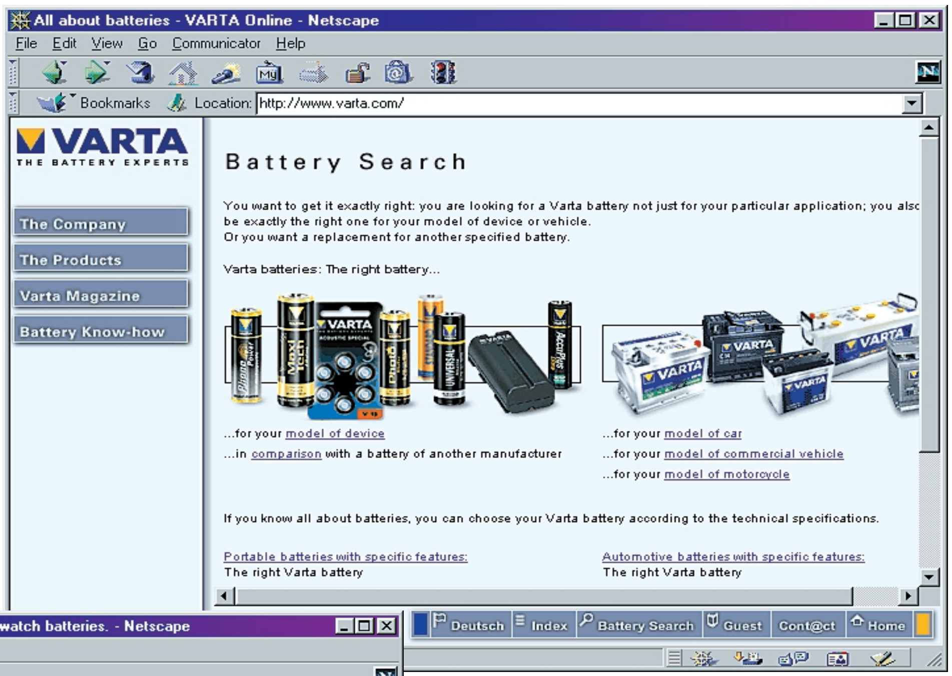
On the **Varta** main website [1] you can press the 'battery search' button to start searching for an equivalent if you have the type number, or a suitable type if you know the equipment or its requirements. An extensive list is available covering most commonly sold equipment.

In Europe, **Eveready** batteries are usually sold under the brand name 'Energiser'. The European website [2] starts with a fairly heavy but pleasant flash animation. From there, you can go to the Product Selector, which is subdivided into different types of equipment.

Electronic component giant **Philips** have set up a separate website for their battery products [3]. To find suitable batteries, you click on one of the battery types. This produces an overview of alternative type numbers.

Duracell, the manufacturer of those well-known gold & black alkaline cells, also supplies a fair amount of product information via the web [4]. Here we find the products in an overview arranged by battery type and user applications (alkaline, lithium, watches, hearing aids, etc.). Having clicked on one of these, an extensive and clearly laid out overview appears with equivalent type numbers of other manufacturers.

Renata is a Swiss battery brand that belongs to the Swatch group.


All about batteries - VARTA Online - Netscape

Location: http://www.varta.com/

VARTA THE BATTERY EXPERTS

The Company | The Products | Varta Magazine | Battery Know-how

Battery Search

You want to get it exactly right: you are looking for a Varta battery not just for your particular application; you also be exactly the right one for your model of device or vehicle.
Or you want a replacement for another specified battery.

Varta batteries: The right battery...

...for your model of device
...in comparison with a battery of another manufacturer

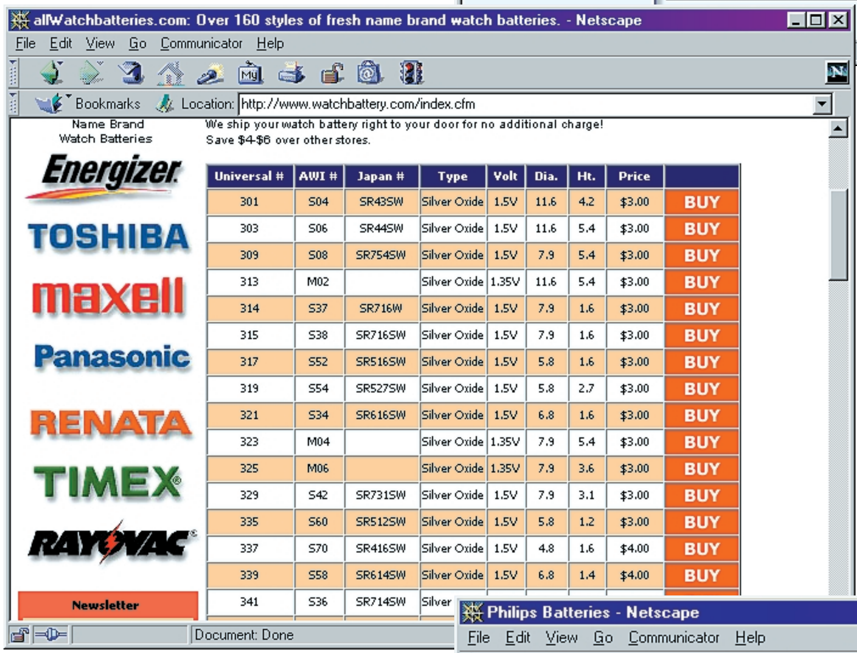
...for your model of car
...for your model of commercial vehicle
...for your model of motorcycle

If you know all about batteries, you can choose your Varta battery according to the technical specifications.

Portable batteries with specific features:
The right Varta battery

Automotive batteries with specific features:
The right Varta battery

Deutsch | Index | Battery Search | Guest | Cont@ct | Home


allWatchbatteries.com: Over 160 styles of fresh name brand watch batteries. - Netscape

Location: http://www.watchbattery.com/index.cfm

Name Brand Watch Batteries

We ship your watch battery right to your door for no additional charge!
Save $4-$6 over other stores.

Universal #	AWI #	Japan #	Type	Volt	Dia.	Ht.	Price	
301	S04	SR43SW	Silver Oxide	1.5V	11.6	4.2	$3.00	BUY
303	S06	SR44SW	Silver Oxide	1.5V	11.6	5.4	$3.00	BUY
309	S08	SR754SW	Silver Oxide	1.5V	7.9	5.4	$3.00	BUY
313	M02		Silver Oxide	1.35V	11.6	5.4	$3.00	BUY
314	S37	SR716W	Silver Oxide	1.5V	7.9	1.6	$3.00	BUY
315	S38	SR716SW	Silver Oxide	1.5V	7.9	1.6	$3.00	BUY
317	S52	SR516SW	Silver Oxide	1.5V	5.8	1.6	$3.00	BUY
319	S54	SR527SW	Silver Oxide	1.5V	5.8	2.7	$3.00	BUY
321	S34	SR616SW	Silver Oxide	1.5V	6.8	1.6	$3.00	BUY
323	M04		Silver Oxide	1.35V	7.9	5.4	$3.00	BUY
325	M06		Silver Oxide	1.35V	7.9	3.6	$3.00	BUY
329	S42	SR731SW	Silver Oxide	1.5V	7.9	3.1	$3.00	BUY
335	S60	SR512SW	Silver Oxide	1.5V	5.8	1.2	$3.00	BUY
337	S70	SR416SW	Silver Oxide	1.5V	4.8	1.6	$4.00	BUY
339	S58	SR614SW	Silver Oxide	1.5V	6.8	1.4	$4.00	BUY
341	S36	SR714SW	Silver					

Newsletter

Document: Done

[1] Varta:
http://www.varta.com
[2] Energizer:
http://www.energizer-eu.com
[3] Philips:
http://www.philipsbatteries.com
[4] Duracell:
http://www.duracell.com/Our_Products/comprehensive.html
[5] Renata:
http://www.renata.com/watch-and-calculator/cref-guide-en.html
[6] WatchBattery:
http://www.watchbattery.com/index.cfm

Philips Batteries - Netscape

Location: http://www.philipsbatteries.com/temp_def_nn.html

PHILIPS Search Talk Global home Let's make things better

HOME

Choose your use
FAQ
Products
Choose Language
Commercial (5.6Mb)
Doe mee met de --Music Power Quiz
Enter the --Nordic Competition

choose a lifestyle
Streetlife • Sportlife • Glamourlife

Visit our new 'Music Power For Life' website where you can WIN FANTASTIC MUSIC PRIZES EVERY DAY!

POWERLIFE XXL • LONGLIFE • MULTILIFE • MINICELLS • TORCHES

Order online!

PHILIPS POWERLIFE

Document: Done

They are best known for their watch and calculator batteries. Renata's website [5] has extensive information on batteries for these applications, with tables showing equivalent type numbers of practically all battery manufacturers and watch brands.

Finally, we should not forget to mention an overview of particular interest to those of you who are on the lookout for a fresh battery for a watch. Under the name Watchbattery.com [6], **JewelryService** runs a web site exclusively covering watch batteries. The site lists more than 160 different types of watch battery in a carefully arranged overview.

(015017-1)

RF Matching Network Software

Calculations using ADLAB

By Dr S. Weber

stephan@weberconnect.com

The universal RF matching network described in this article is particularly easy to use, adapt using (free) software and last but not least, easily constructed from readily-available components. We give example component values suitable for use in the 70 cm (440 MHz) band.

Matching networks are used in RF design to avoid mismatch where different impedances are involved. Calculation of component values is in practice left to a dedicated program such as the ADLAB shareware package used here. This package was developed by the author and is available for download from his website.

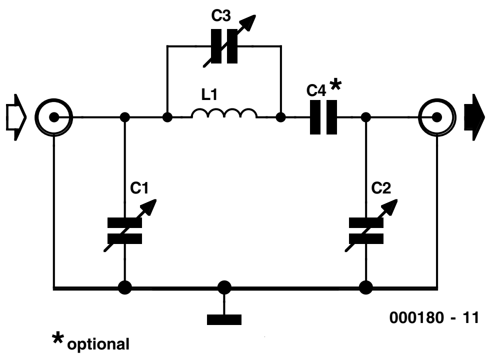
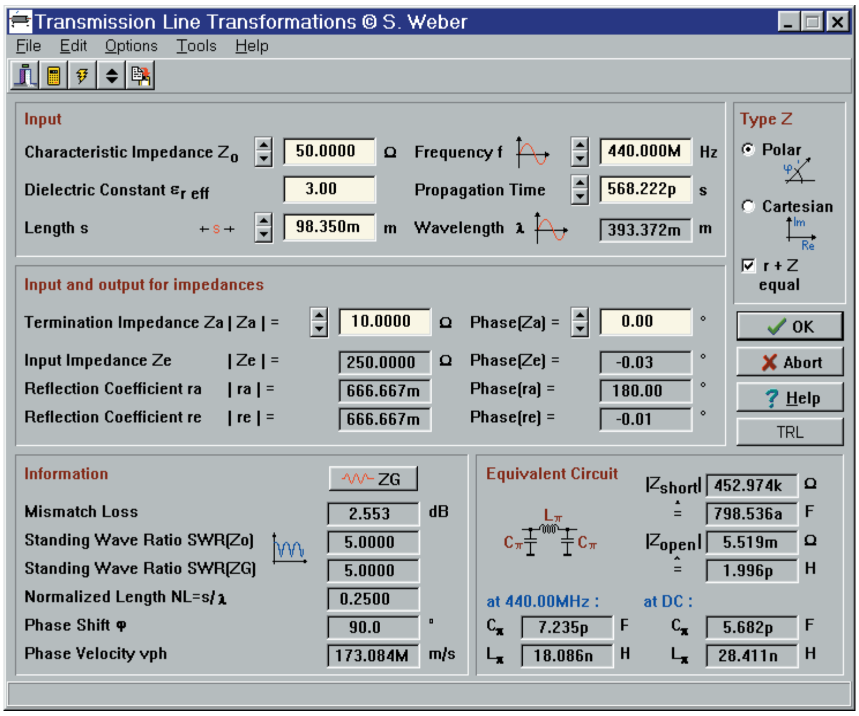


Figure 1. Complete circuit (C4 optional).



Input

Characteristic Impedance Z_0 50.0000 Ω — Frequency f 440.000M Hz

Dielectric Constant $\varepsilon_{r\ eff}$ 3.00 — Propagation Time 568.222p s

Length s 98.350m m — Wavelength λ 393.372m m

Type Z: (● Polar) ○ Cartesian; ☑ r + Z equal

Input and output for impedances

Termination Impedance Za | Za | = 10.0000 Ω — Phase(Za) = 0.00 °

Input Impedance Ze | Ze | = 250.0000 Ω — Phase(Ze) = -0.03 °

Reflection Coefficient ra | ra | = 666.667m — Phase(ra) = 180.00 °

Reflection Coefficient re | re | = 666.667m — Phase(re) = -0.01 °

Information

Mismatch Loss	2.553	dB
Standing Wave Ratio SWR(Zo)	5.0000	
Standing Wave Ratio SWR(ZG)	5.0000	
Normalized Length NL=s/λ	0.2500	
Phase Shift φ	90.0	°
Phase Velocity vph	173.084M	m/s

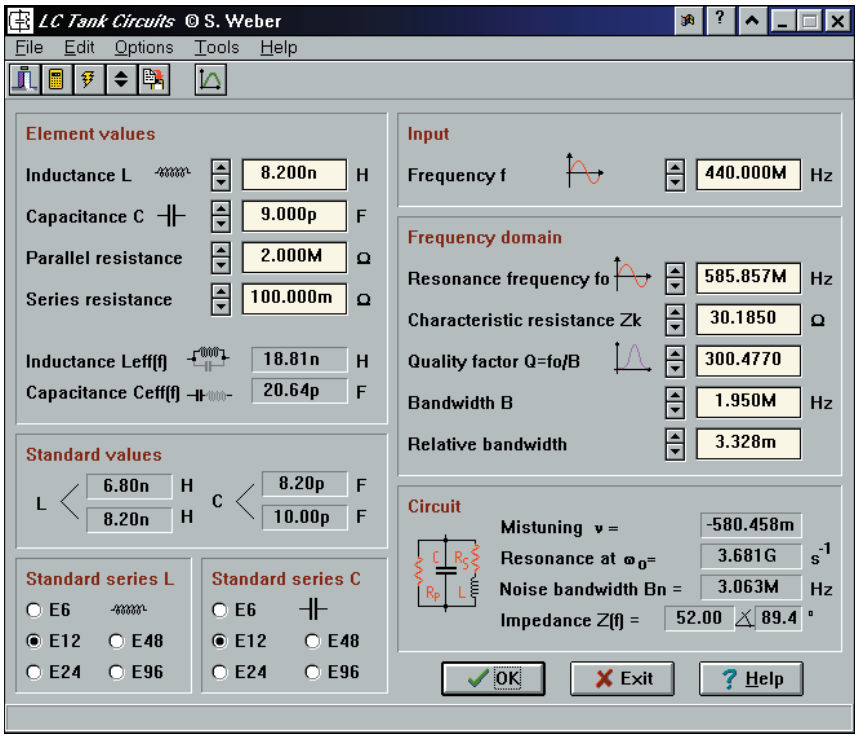
Equivalent Circuit

$|Z_{short}|$ 452.974k Ω ≙ 798.536a F

$|Z_{open}|$ 5.519m Ω ≙ 1.996p H

at 440.00MHz :			at DC :		
C_π	7.235p	F	C_π	5.682p	F
L_π	18.086n	H	L_π	28.411n	H

Figure 2 : Equivalent circuit of a 50Ω λ/4 transmission line at 440MHz (calculation with LTRANS : L_1=18,08nH, C_1=C_2=7,235pF)



LC Tank Circuits © S. Weber

File Edit Options Tools Help

Element values

Inductance L	8.200n	H
Capacitance C	9.000p	F
Parallel resistance	2.000M	Ω
Series resistance	100.000m	Ω
Inductance Leff(f)	18.81n	H
Capacitance Ceff(f)	20.64p	F

Standard values

L: 6.80n H / 8.20n H — C: 8.20p F / 10.00p F

Standard series L: E6 [✓ E12] E24 E48 E96

Standard series C: E6 [✓ E12] E24 E48 E96

Input

Frequency f	440.000M	Hz

Frequency domain

Resonance frequency fo	585.857M	Hz
Characteristic resistance Zk	30.1850	Ω
Quality factor Q=fo/B	300.4770	
Bandwidth B	1.950M	Hz
Relative bandwidth	3.328m	

Circuit

Mistung v =	-580.458m	
Resonance at ω_0=	3.681G	s^{-1}
Noise bandwidth Bn =	3.063M	Hz
Impedance Z(f) =	52.00 ∠ 89.4	°

Figure 3 : Calculation of the effective inductance of a parallel LC circuit (LCFILT from Elekta Professional : 8,2nH||9pF results in 18,8nH at 440MHz).

Low-Pass Structure

A low-pass structure was chosen for the example matching network described in this article. This has the normally desirable side-effect that harmonics are attenuated. DC is passed with unity gain; if this is not desired, a suitably large series capacitor can be added to the circuit. In order to accommodate virtually any impedance combination, three alignment points are provided in the form of standard trimmer capacitors. **Figure 1** shows the complete circuit, which comprises a third-order elliptic low-pass filter. Construction of the circuit depends on the frequency band it is intended for. It is advisable to build the circuit as small as possible in a metal enclosure using SMA connectors, attaching the components to one another directly by their leads. For higher power applications, N-type connectors are recommended, along with air-dielectric trimmers with an adequate plate gap.

Component Values

The circuit is similar to the equivalent circuit of a transmission line, and indeed it is well known that a λ/4 line can be used to make a good matching circuit. The equivalent circuit of the λ/4 transmission line for the frequency band of interest gives us a good starting point for calculating the component values for our circuit. For the 70 cm band the relevant values are shown in **Figure 2**.

Trimmers C1 and C2 should be chosen to have a maximum capacitance of about three times the calculated value of 7.2 pF. L1 should be about half the calculated value (here 8.2 nH instead of 18 nH) since the effective inductance is increased by the parallel capacitance C3. C3 should be the same value as C1 and C2.

The value of C4 should be at least five times the value of C1. If a very large value, say 470 nF, is chosen (to extend the lower limit of the passband), a ceramic capacitor of around 470 pF should be added in parallel. If the circuit is to be modified for the 2 m or FM bands, the calculations can be carried out from scratch or the component values can simply be scaled. For example, for the FM band, all the capacitance and inductance values should simply be multiplied by 440 MHz / 100 MHz = 4.4.

Alignment

It might be thought that with three trimmers alignment of the circuit will be complicated. However, the situation can be simply illustrated in the Smith chart of **Figure 4**. If all three trimmers are set to their mid-positions, we know that the circuit will behave approximately like a λ/4 transmission line, with both an input impedance and an output impedance of 50 Ω. Thus these settings make a good starting point.

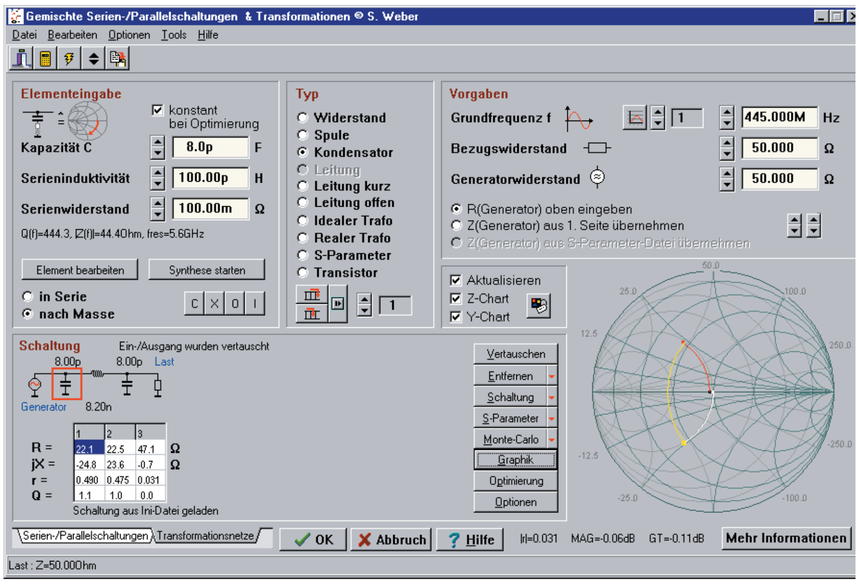
However, in contrast to a transmission line, the frequency response of the circuit is not periodic: instead it exhibits a low-pass characteristic (**Figure 5**). Here the second harmonic is attenuated by 14 dB.

With the aid of the Smith chart we can see the effect of adjusting the values. We discover that C3 chiefly affects the real part of the impedance, while C1 and C2 affect the imaginary part. In practice we start from the basic settings above and adjust the trimmers in turn, beginning with the one that has the greatest effect. If the required impedance is already known (for example if it has been measured using a network analyser), the component values can be got directly from the Smith chart and the trimmers set appropriately. Final adjustments can then be made on the actual circuit.

Although the bandwidth shown in Figure 5 is rather wide, it narrows considerably when a large shift in impedance is required. **Figure 6** shows an example where an 8 Ω signal generator is to be matched to a 50 Ω impedance (VSWR 6.25:1). A small mismatch of 1.45:1 remains, since in practice the trimmers cannot be adjusted down to zero. Nevertheless, the resulting attenuation is only 0.3 dB (compared to 3.3 dB without a matching network). The –1 dB bandwidth is reduced to around 200 MHz and attenuation of harmonics is improved, especially above 1.5 GHz.

For many required impedances there are various combinations of trimmer settings with the same matching effect at the design frequency. These settings will, however, produce different characteristics in terms of bandwidth and harmonic attenuation. When used in conjunction with a power amplifier, one setting may be as good as another, although the resulting efficiency can vary. If C3 is given a high value, the frequency response falls off more steeply, because of the zero introduced into the response by the parallel resonant circuit. Smaller capacitors result in a flatter response and a wider bandwidth, and make the circuit less sensitive to component tolerances.

The component values used were:



Gemischte Serien-/Parallelschaltungen & Transformationen © S. Weber

Datei Bearbeiten Optionen Tools Hilfe

Elementeingabe

[✓] konstant bei Optimierung

Kapazität C	8.0p	F
Serieninduktivität	100.00p	H
Serienwiderstand	100.00m	Ω

Q(f)=444.3, |Z(f)|=44.40hm, fres=5.6GHz

Element bearbeiten | Synthese starten

○ in Serie [● nach Masse]

Typ: ○ Widerstand ○ Spule [● Kondensator] ○ Leitung ○ Leitung kurz ○ Leitung offen ○ Idealer Trafo ○ Realer Trafo ○ S-Parameter ○ Transistor

Vorgaben

Grundfrequenz f	445.000M	Hz
Bezugswiderstand	50.000	Ω
Generatorwiderstand	50.000	Ω

[● R(Generator) oben eingeben] ○ Z(Generator) aus 1. Seite übernehmen ○ Z(Generator) aus S-Parameter-Datei übernehmen

[✓] Aktualisieren [✓] Z-Chart [✓] Y-Chart

Schaltung — Ein-/Ausgang wurden vertauscht

8.00p 8.00p Last, Generator 8.20n

	1	2	3	
R =	22.1	22.5	47.1	Ω
jX =	-24.8	23.6	-0.7	Ω
r =	0.490	0.475	0.031	
Q =	1.1	1.0	0.0	

Schaltung aus Ini-Datei geladen

Vertauschen | Entfernen | Schaltung | S-Parameter | Monte-Carlo | Graphik | Optimierung | Optionen

Serien-/Parallelschaltungen | Transformationsnetze | OK | Abbruch | Hilfe | |r|=0.031 MAG=-0.06dB GT=-0.11dB | Mehr Informationen

Last : Z=50.000hm

Figure 4 : Circuit with its default values in the Smith diagram (calculation with CSERPAR).

L1 = 8.2 nH (air core; approximately 3 turns 0.5 mm enamelled copper wire; internal diameter 2.5 mm)

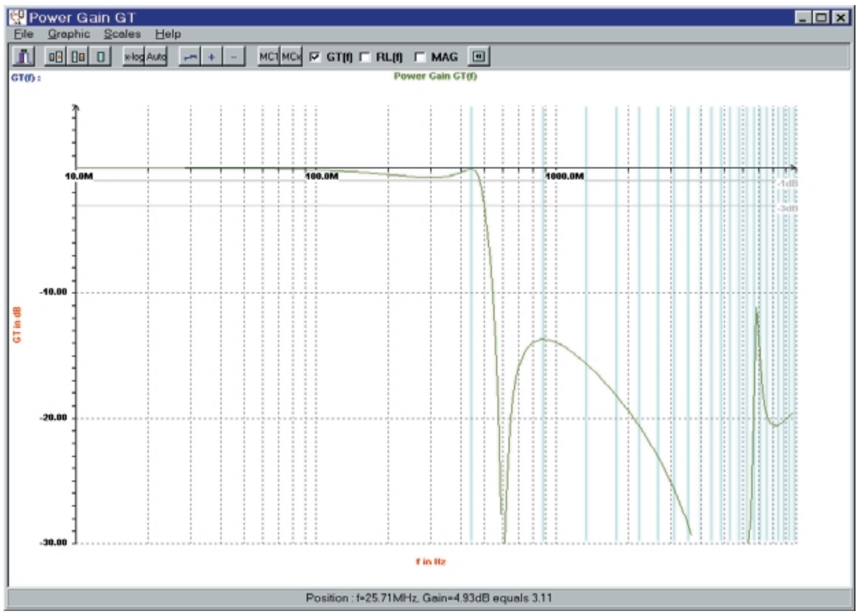
C1 = C2 = C3 = 1-22 pF

C4 = 470 pF

Application Example

Matching networks are frequently used with RF amplifiers or antennas. One application might be to match a small 70 cm handheld transceiver to an RF transmit amplifier. Without matching, the transceiver will not be able to drive the amplifier adequately: with a direct 50 Ω connection the output power will be unsatisfactorily low, and the automatic transmit/receive switch (VOX) may no longer work. The matching circuit can be fitted between the transceiver and the amplifier and the trimmers adjusted for maximum output power into the antenna. Adjust the trimmers in small increments, because when the trimmers are at the end of their travel a small movement can easily make matching worse rather than better. Take particular care when working with RF power amplifiers, which can be destroyed in cases of extreme impedance mismatch.

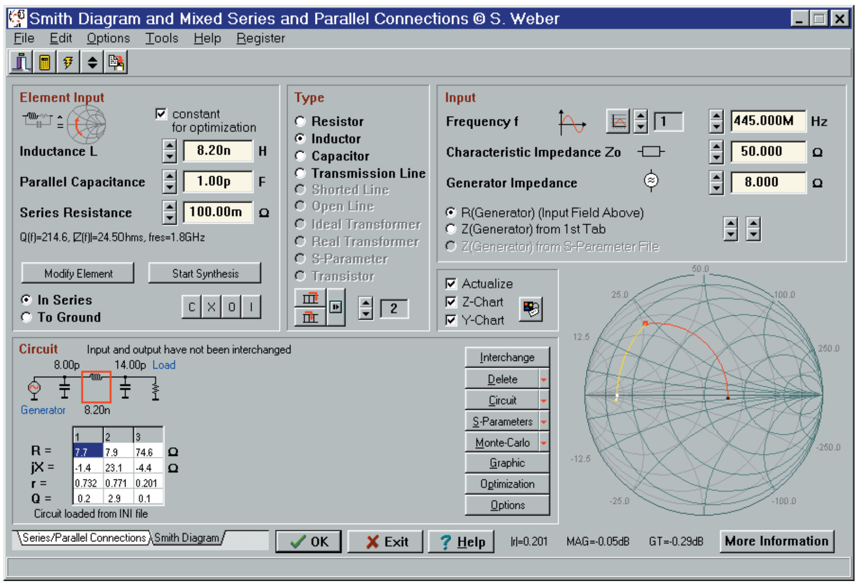
(0000180-1)



Position : f=25.71MHz, Gain=-4.93dB equals 3.11

Figure 5 : Frequency response with default values (calculation with CSERPAR).

The ADLAB software package, developed by the author, consists of eleven programs. It can be downloaded free of charge from [*http://home.t-online.de/home/weberconnect/adlab2.htm*](http://home.t-online.de/home/weberconnect/adlab2.htm)
This shareware version will run 100 times; the registration fee is US$ 35 for the unrestricted version.



Smith Diagram and Mixed Series and Parallel Connections © S. Weber

File Edit Options Tools Help Register

Element Input

[✓] constant for optimization

Inductance L	8.20n	H
Parallel Capacitance	1.00p	F
Series Resistance	100.00m	Ω

Q(f)=214.6, |Z(f)|=24.50hms, fres=1.8GHz

Modify Element | Start Synthesis

[● In Series] ○ To Ground

Type: ○ Resistor [● Inductor] ○ Capacitor ○ Transmission Line ○ Shorted Line ○ Open Line ○ Ideal Transformer ○ Real Transformer ○ S-Parameter ○ Transistor

Input

Frequency f	445.000M	Hz
Characteristic Impedance Zo	50.000	Ω
Generator Impedance	8.000	Ω

[● R(Generator) (Input Field Above)] ○ Z(Generator) from 1st Tab ○ Z(Generator) from S-Parameter File

[✓] Actualize [✓] Z-Chart [✓] Y-Chart

Circuit — Input and output have not been interchanged

8.00p 14.00p Load, Generator 8.20n

	1	2	3	
R =	7.7	7.9	74.6	Ω
jX =	-1.4	23.1	-4.4	Ω
r =	0.732	0.771	0.201	
Q =	0.2	2.9	0.1	

Circuit loaded from INI file

Interchange | Delete | Circuit | S-Parameters | Monte-Carlo | Graphic | Optimization | Options

Series/Parallel Connections | Smith Diagram | OK | Exit | Help | |r|=0.201 MAG=-0.05dB GT=-0.29dB | More Information

Figure 6. Circuit adjusted for matching between 8 Ω and 50 Ω (calculated using CSERPAR).

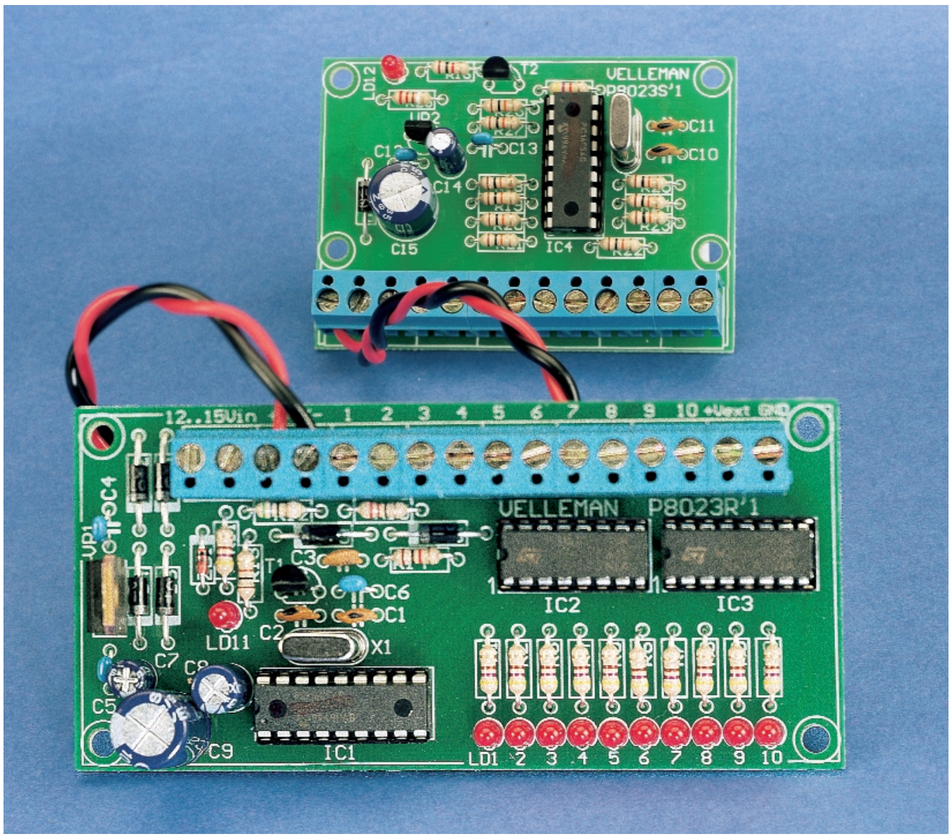
Two-Wire Remote Control

Switch ten channels via just two wires

By H. Steeman

This time we'll look at a kit for a remote control that allows you to send ten digital control signals over a distance of more than 50 metres. How? Using infrared? No, with just two wires!

When using a microcontroller or PC to control various pieces of equipment it is vital that the digital control signals can be sent reliably over larger distances, and in such a way that we do not need lots of cables. This circuit, which has been developed by Velleman, allows us to send ten digital signals through two wires. It can be used to cover distances up to 50 m.



Current loop

Figure 1 shows the circuit diagram of the transmitter. The circuit is built around a PIC16C54, a microprocessor used previously in *Elektor Electronics* projects. The ten inputs (1-10) are connected to the bus. All inputs have a pull-up resistor of 10 kΩ, which keeps them at +5 V when they are not connected. Seven inputs are connected to port B and three to port A of the processor. One output of port A drives transistor T2. The waveform used to drive T2 depends on the state of the logic levels found on the ten inputs. When T2 conducts, a current of about 50 mA (10 V/220 Ω plus the LED current) flows through the transistor. The receiver connected to the VTX lines (SK9) detects these current pulses. The use of a current loop is a good choice for this type of application because it is much less susceptible to interference than a voltage-driven signal.

The rest of the circuit is fairly straightforward. The voltage across the two control lines is 10 V. Diode D7 keeps capacitor C15 charged to this level in between the pulses. This is fed to voltage regulator VR2, which

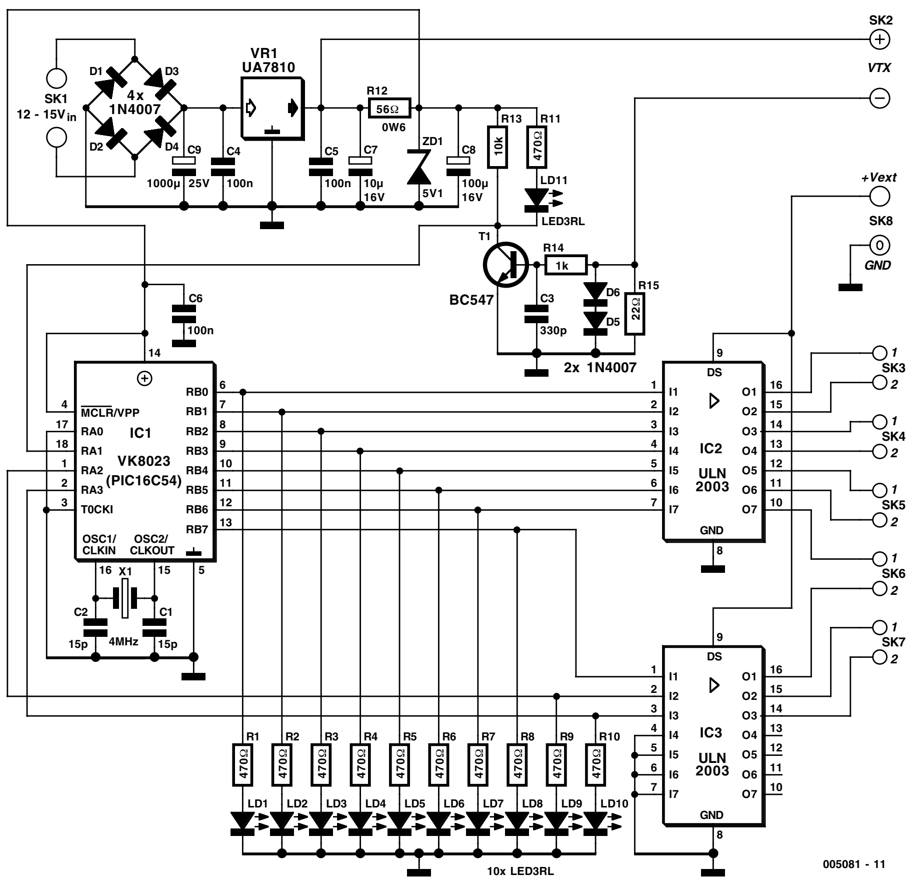


Figure 1. The circuit diagram of the transmitter. The use of PCB terminal blocks simplifies the connections.

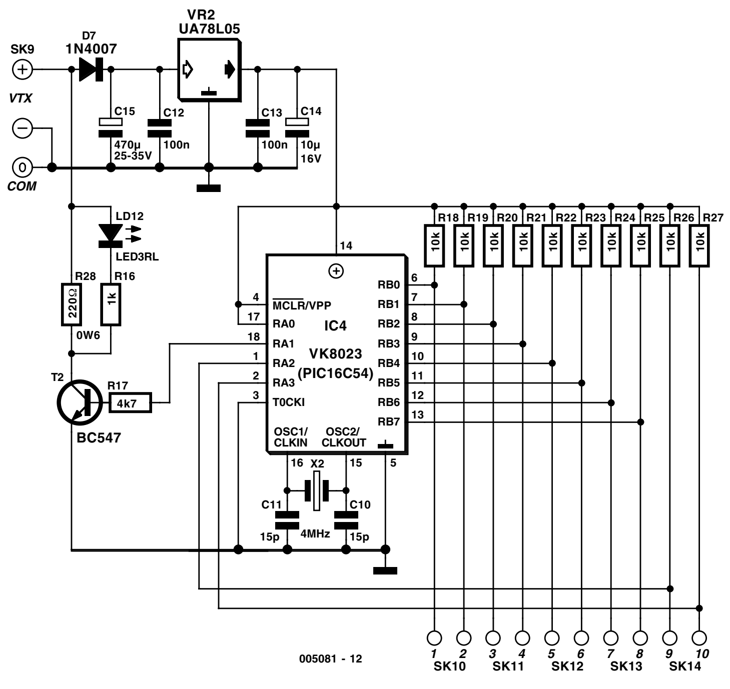


Figure 2. The receiver also uses a PIC processor to decode the current pulses.

supplies the microprocessor of the transmitter (IC4) with a stable 5 V. Quartz crystal X2 and capacitors C10 and C11 provide a clock signal of 4 MHz.

Decoding

Figure 2 shows the circuit diagram of the receiver. A current detector is connected to SK2. The current pulses from the transmitter are converted to a voltage by R15, which drives the base of T1. Diodes D5 and D6 protect the transistor against any voltage surges that may be induced on the line. LED LD11 lights when a control signal is being received.

The received pulses are decoded by microprocessor IC1 (also a PIC16C54) into ten digital signals, again to ports A and B. Ten LEDs show the state of the digital outputs. The ten digital signals are fed to a ULN2003 driver IC. This chip contains power drivers with open collector outputs and internal protection diodes. These can switch relatively large voltages and currents (50 V/100 mA). The positive supplies of the switched loads have to be connected to +Vext.

The receiver supplies power to the whole system. The supply required is about 12 V AC, for example from a simple mains adapter (a DC adapter is also possible, as long as it supplies a minimum of 15 V). A bridge rectifier turns it into a DC voltage which is fed to VR1 (a 7810), which outputs a stabilised 10 V. This voltage is fed directly to the control cable. Resistor R12 and zener diode ZD1 provide a supply of 5 V to the microprocessor.

The type of cable chosen to make the connection depends mainly on where it will be used. Indoors we can use a simple (shielded) cable —outdoors it is better to use a more sturdy cable. We have tested the circuit over a distance of more than 100 m using telephone cable and found that it still worked reliably.

The complete kit comprising the transmitter and receiver is available under **order code K8023** from authorised Velleman dealers. For further information, see Vellemans' website at http://www.velleman.be or email info@velleman.be

(005108-1)