

Automotive CANtroll

Car electronics exposed

Chris Vossen (Elektor Labs) and Ep Gernaat (Timloto, The Netherlands)

This universal microcontroller board was designed, in the first instance, for use by students studying automotive technologies, but it can also be used for other applications, of course. The heart of this board is an Atmel AT90CAN32 with a fast RISC core.



Technical Specifications

- Microcontroller: Atmel AT90CAN32
- Fast RISC -architecture with 133 instructions
- Clock speed: 12 MHz
- 32 KB flash, 2 KB RAM and 1 KB EEPROM available
- 53 programmable I/O lines
- Integrated CAN2.0 controller
- 8-channel 10-bit A/D converter
- SPI interface
- JTAG interface
- 2 USARTs
- Two-Wire interface
- 8 DIP switches and 8 LEDs available for experimental applications
- Power supply: 5 V

Since cars contain an ever increasing amount of electronics, students learning about motor vehicle technology also need to know more about electronics and microcontrollers. In collaboration with the Timloto o.s. Foundation in the Netherlands, Elektor designed a special controller PCB, which will be used in schools in several countries for teaching students about automotive technologies. Particular attention was paid to issues such as universal design, cost, connection options, expandability and the availability of free development software for various platforms.

Cars and electronics

About 20 years ago, teachers of the subject of motor vehicle technology introduced the topic of microcontrollers into the curriculum of automotive technicians. In those days they used a teaching kit that was based on the Z80, which was appropriately named the Microprofessor.

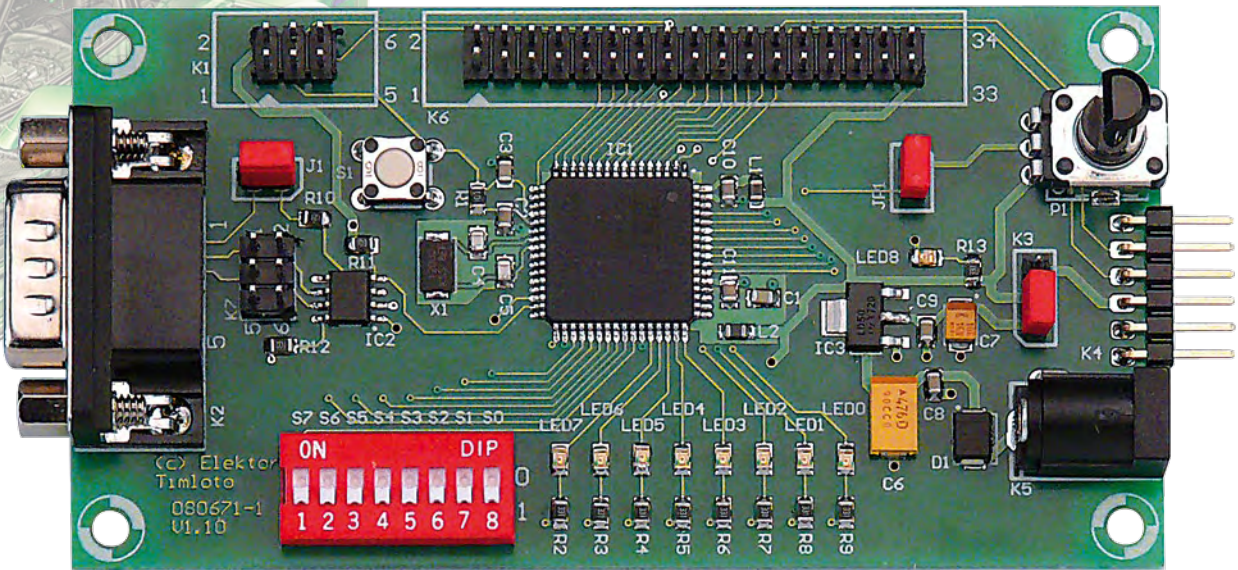
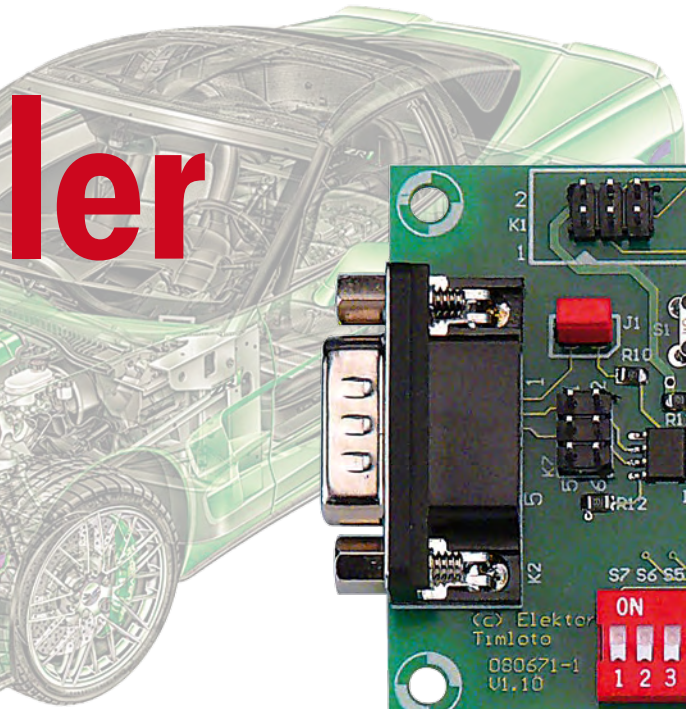
This kit has been used intensively for at least 10 years, at the higher grades of technician training and teachers united in the TIM working group (TIM = Technical Informatics Motor Vehicles) have made many automotive applications over the years.

However, at some stage a 'real' automotive microcontroller was selected, the Motorola 68HC11, which at that time was frequently used in various automotive computers. The educational programs which were originally developed for the Z80 were ported across and expanded. An engineering consultant developed a 68HC11 controller board based on the specifications from the TIM group. One of the prerequisites was that the already developed Z80 hardware applications could be used again. A textbook was written and the teachers were given further training. Even now the 68HC11 is still used successfully as an educational controller within the context of motor vehicle technology.

The TIM working group evolved into the Timloto o.s. foundation, a working group of teachers which sets itself the purpose of closely monitoring the technical developments in cars and make these available as teaching resources to other teachers and students as soon as possible and at no cost. New times, new opportunities: the Timloto website with open-source licence became a fact. See www.timloto.org.

In the meantime the CAN bus became common and car computers started receiving Flash memory upgrades while the cars were being serviced. Again there was a call for a new(er) controller and this time the editors at Elektor were approached for advice. The choice for the Atmel AT90CAN32 was quite quickly made because of its reasonable purchase price, the many features and the programmability under Windows and Linux (Ubuntu). Thanks to the ingenuity of the Elektor designers all the requirements from the Timloto specifications could be met. The demands were considerable. Because Timloto works together with another automotive teachers initiative (the 'GoforAfrica' foundation) it also had to be possible to use the controller in technical schools in Senegal and Gambia. The computers there run the Ubuntu operating system so that a Linux development environment was an absolute requirement.

Costs also play an important role. The approach was to keep the cost of the controller board as low as possible so that it could be added to



A significant amount of teaching material is already available, which is freely available to anyone via the Timloto website. You can therefore also use this project at home. But the design of this circuit is so universal that it will also be excellent for all kinds of other home, or should we say, garage, projects.

Choice of microcontroller

When searching for a suitable microcontroller we soon arrived at the AT90CAN32 made by Atmel. This controller is packed with many features. It

has 32 KB of Flash memory and 2 KB of RAM. An EEPROM of size 1 KB is also available. In addition to a 10-bit A/D-converter with eight channels, the controller also has multiple timers, an SPI interface and two USARTS (one of which is used as the programming interface).

There is also a TWI interface and a CAN2.0 controller. The latter makes this controller eminently suitable for applications in an automotive environment. The core of this controller has a RISC architecture with an instruction set consisting of 133 instructions.

The AT90CAN32 is available in both 64-pin TQFP and QFN packages. For this design we choose the TQFP version. This package type has all the pins accessible around the outside edge of the package, which makes it much easier to solder by hand.

Finally we would like to mention that this controller is completely compatible with its bigger siblings the AT90CAN64 and AT90CAN128. For a detailed description of all its features (such as the TWI) we refer you to the datasheet [1].

the book list of the automotive science students and in this way each student also has the opportunity to practise in his or her own time. The switches in particular were a hurdle initially. Eight toggle switches would increase the price of the design considerably. A clever Elektor solution was found by using a separate expansion board for use in class, which contains the switches. The module is plugged into the expansion board. In this way, during classes at school the robust switches on the expansion board are available.

Using the first Elektor prototypes, members of the Timloto working group could translate the first (educational) 68HC11 programs into AT90CAN32 assembly language. For this purpose, use was made of AVR Studio 4 and gcc-avr, avrdude and kontrollerlab for Linux. The lesson materials are now organised in a matrix and can be found at www.timloto.org/nl/matrix/matrix_atmel.html



In the spring and autumn Timloto will organise training courses for automotive technology teachers in Gambia, Senegal and the Netherlands to show them the educational use of the AT90CAN32. Timloto aspires to international cooperation between all (automotive) technical education and would like to see as many people as possible supporting this Elektor-Timloto project. Help is required to translate programs (comments and questions) into English and French. Program ideas and new applications are also very welcome. Consideration can also be made to use the C programming language instead of assembly language.

We would like to appeal to teachers, electronics and information technology experts to cooperate and develop things further, all in the spirit of open-source.

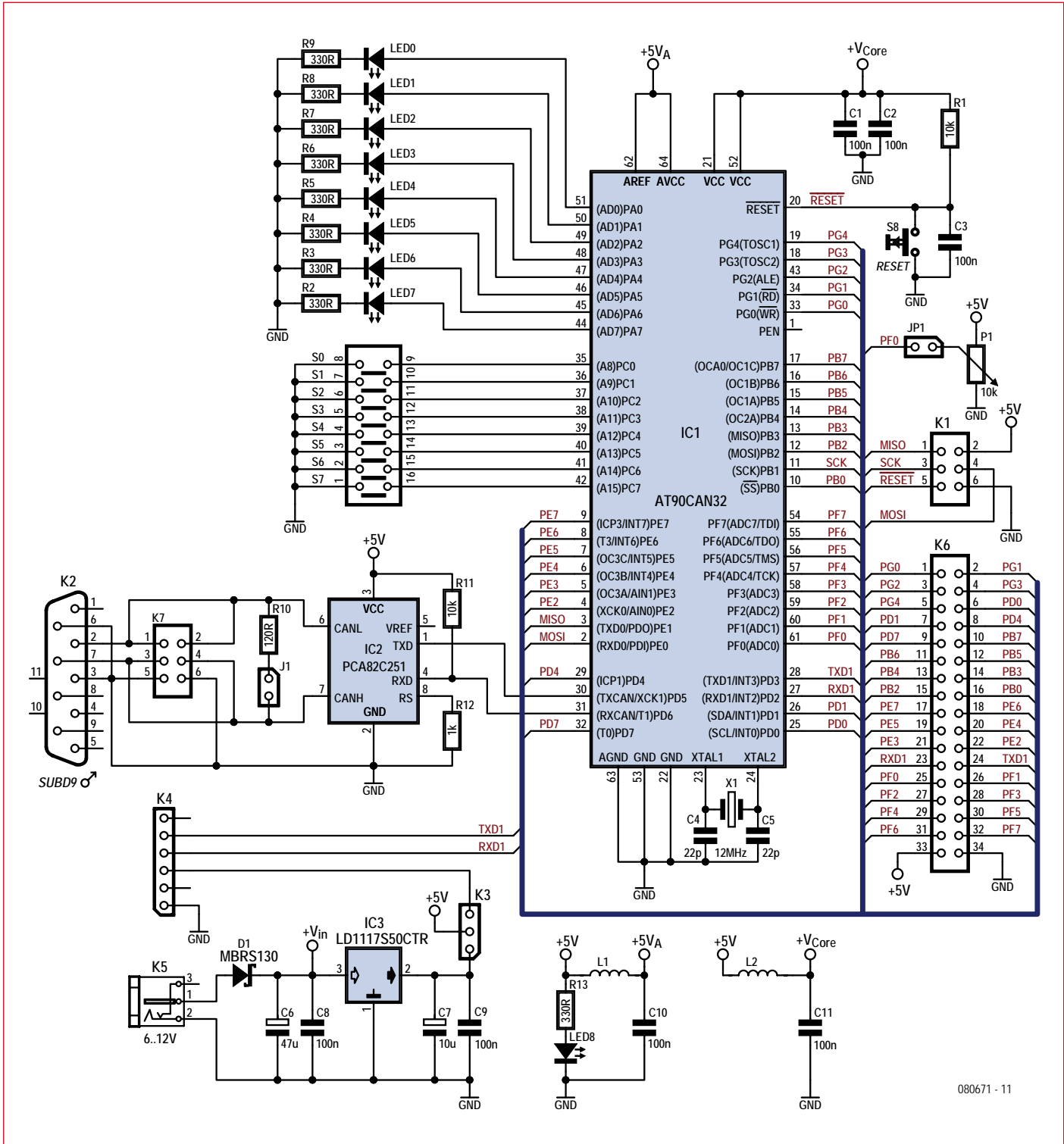


Figure 1. The schematic for automotive CANtroller module.

Schematic

The schematic for this module is a relatively simple design (Figure 1). The heart of the circuit is formed by the AT90CAN32 (IC1). The Reset pin is connected to an RC network (R1/C3) which provides a reset when the power supply voltage is turned on. S8

provides the option of manually resetting the circuit. A crystal of 12 MHz is used for generating the clock frequency. To use this crystal the configuration fuses of the microcontroller have to be programmed with the correct settings. Using AVR Studio, the SUT_CKSEL bits can be configured for an external crystal with 8 MHz mini-

um frequency.

To enable the CAN controller to communicate with a real CAN bus a CAN transceiver is required. This can be found on the schematic in the form of a PCA82C251 (IC2). This IC is quite well known by now and conforms to the ISO11898-24V standard. This transceiver can therefore be used with both

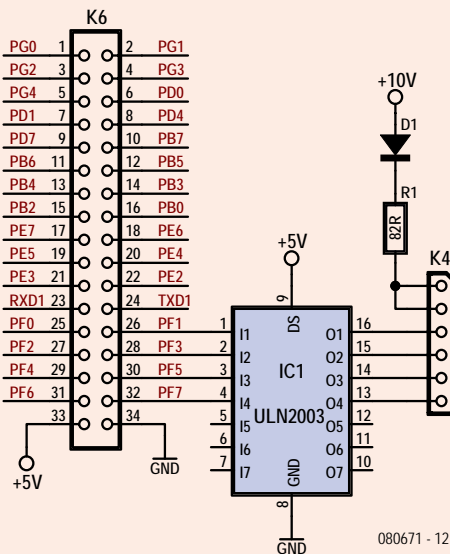
Stepper motor control

The example below shows how a bipolar stepper motor can be connected to the control module. Bipolar stepper motors contain a number of windings which need to be driven according to a certain pattern. In our example this is the following continuously repeating pattern : 0101 1001 1010 0110.

In the initialisation routine of this example, Ports A and F are configured as outputs. The stack pointer is initialised before any subroutines are used. Port C is entirely configured as inputs, because this is where the switches are connected. The switches are not used in this example, however.

In the main program the four steps, one at a time, are continuously written to Port F, with a small pause between each one. The stepper motor will rotate as a result. When step 4 is completed the software will begin again with step 1. The controller will repeat this pattern over and over again.

The stepper motor is connected to the microcontroller using the familiar ULN2003A (see schematic). This IC contains a number of Darlington transistors which can deliver sufficient current to get a small stepper motor to turn.



The example program is available as a free download from the Elektor website filed under number 080671-11.zip.

```

/*Program name: TESTPORTF.ASM
Program for de AT90CAN32 Elektor-Timloto board
Port F output
Port A output and drives LEDs
This program uses AVR Studio 4
The program runs from flash memory
*/

```

```

.DEVICE AT90CAN32
.INCLUDE "can32def.inc";definition of ports
                        are in a separate file

RJMP RESET ;jump to starting address

/*INITIALISATION*/

RESET: LDI R16,$FF ;set all pins of Ports A and
                F to outputs

OUT DDRF,R16
OUT DDRA,R16

LDI R16,high(RAMEND)
OUT SPH,R16
LDI R16,low(RAMEND)
OUT SPL,R16 ;stack pointer initialisation is
                necessary for subroutine

                ;is not yet used here

LDI R16,$FF ;activate the pull-up
                resistors
OUT PORTC,R16 ;by writing ones the the
                output port
LDI R17,$00 ;set all pins of Port C to
                inputs
OUT DDRC,R17 ;not really necessary
                (default value)

NOP

/* MAIN PROGRAM*/

BEGIN: LDI R17,0b00100010 ;0101 step 1
OUT PORTF,R17
OUT PORTA,R17
RCALL WAIT1

LDI R17,0b10000010 ;1001 step 2
OUT PORTF,R17
OUT PORTA,R17
RCALL WAIT1

LDI R17,0b10001000 ;1010 step 3
OUT PORTF,R17
OUT PORTA,R17
RCALL WAIT1

LDI R17,0b00101000 ;0110 step 4
OUT PORTF,R17
OUT PORTA,R17
RCALL WAIT1

RJMP BEGIN

/*DELAY SUBROUTINE*/

WAIT1: LDI R20,0x0F ;0F (01 for debugger)
WAIT: LDI R18,0xFF ;0xFF (01 for debugger)
AGAIN: LDI R19,0xFF ;0xFF (01 for debugger)
LOOP: SUBI R19,0x01
BRNE LOOP
SUBI R18,0x01
BRNE AGAIN
SUBI R20,0x01
BRNE WAIT
RET ;return to main program

```

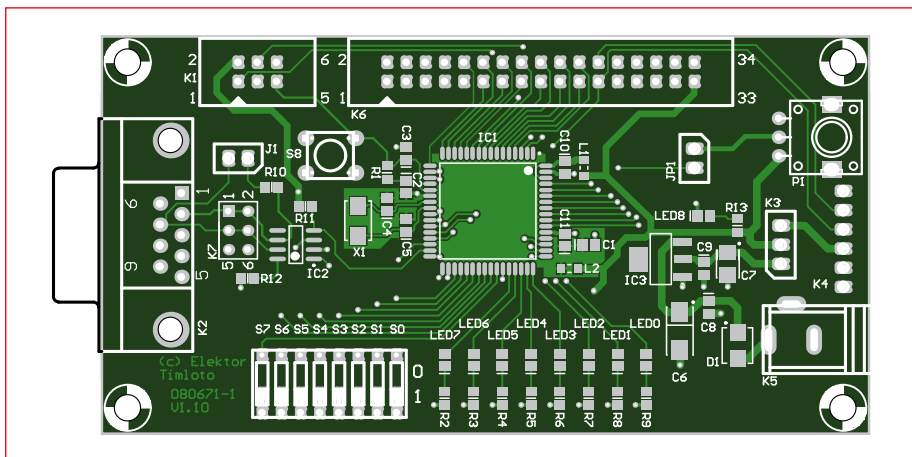
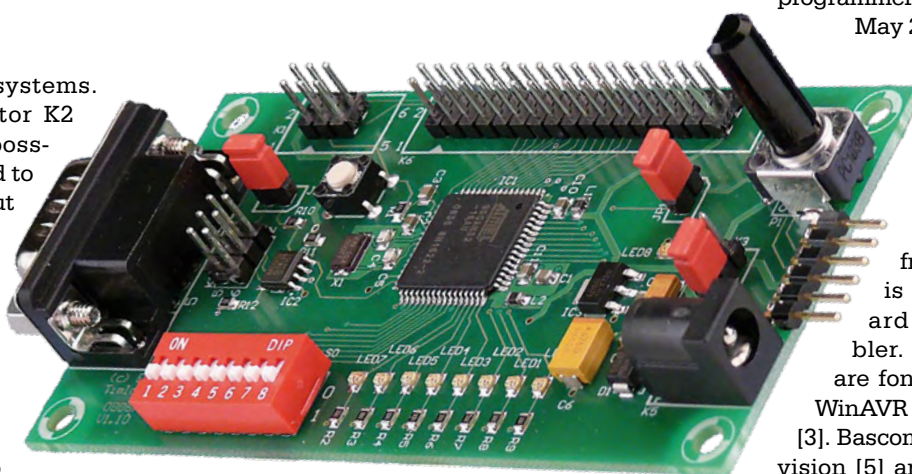


Figure 2. Component layout for the PCB that was designed for this circuit.

12-V as well as 24-V systems. Using sub-D9 connector K2 or pin header K7 it is possible to connect the board to the CAN-bus. The pinout of this connector corresponds with that of the USB-CAN adapter which was published in our October 2008 issue.

Considering the educational character of this project, the board is also provided with eight switches (S0 to S7) and eight LEDs (LED0 to LED7) which can be used when doing programming exercises. There is also a potentiometer (P1) on the board. You can, for example, let the microcontroller read the position of the potentiometer and depending on the measured value turn on a number of LEDs. You need to fit jumper JP1 to connect the potentiometer to the microcontroller.



K4 is the familiar header for the USB-TTL cable which we have used in several earlier Elektor projects (080213-71, see Elektor Shop).

The power supply for the circuit is built around a classic design using a LD1117S50. This is a linear low-drop 5-V regulator which requires very few external components. With K3 you can choose whether the circuit is powered

from the USB-connection or from voltage regulator IC3.

The PCB layout for the circuit is shown in **Figure 2**. We won't discuss the details of assembling the PCB. Experienced electronics enthusiasts are certainly capable of assembling this board by hand, but most users would probably order the ready-made board from Elektor instead.

Programming

In order to program the microcontroller you need to have a programmer. You can do this by, for example, connecting the Elektor USB AVRprog to K1. This programmer was featured in the

May 2008 issue. This programmer is still available from the Elektor Shop (no. 080083-71).

For the programming software you can use AVR Studio from Atmel [2]. This is available as standard with an assembler. Those of you who are fond of C can use the WinAVR open-source toolset [3]. Bascom AVR [4] and Codevision [5] are a couple of commercial alternatives. These have evaluation versions available that you can download.

The Automotive CANtroller module is available from the Elektor Shop and has the catalogue number **080671-91**. All SMD parts are already fitted on the board. Only the through-hole parts and the connectors still need to be soldered.

Finally, a comment about the power supply for the module. As you will have noticed already, this can be powered from either the USB connection or from a mains adapter. Make sure you have the correct setting for the jumper on connector K3.

(080671-1)

COMPONENT LIST

Resistors

R10 = 120 Ω (SMD0805)
R2-R9, R13 = 330 Ω (SMD0805)
R12 = 1kΩ (SMD0805)
R1, R11 = 10kΩ (SMD0805)
P1 = 10kΩ potentiometer (RK09K11310KB)

Capacitors

C1, C2, C3, C8-C11 = 100nF (SMD0805)
C4, C5 = 22pF (SMD0805)
C6 = 47μF 20V (CASE D)
C7 = 10μF 16V (CASE B)

Semiconductors

D1 = MBR5130 (SMB)
IC1 = AT90CAN32-16AU (TQFP-64)

IC2 = PCA82C251/N4 (SO8)
IC3 = LD1117S50CTR (SOT223)
LED1-LED9 = SMD LED (SMD0805)
X1 = 12MHz quartz crystal

Miscellaneous

JP1, JP2 = 2-way SIL pinheader + jumper
K1, K7 = 6-way DIL pinheader
K3 = 3-way SIL pinheader + jumper
K6 = 34-way DIL pinheader
K4 = right angled 6-way SIL pinheader
K5 = DC adapter connector
K2 = right angled 9-way sub-D plug (male), PCB mount
S0-S7 = one 8-way DIP switch
S8 = pushbutton
Kit of parts, contains SMD-pretuffed board and all through-hole components. Elektor Shop # **080671-91**.

Internet Links

- [1] www.atmel.com/dyn/resources/prod_documents/doc7682.pdf
- [2] www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725
- [3] <http://winavr.sourceforge.net/>
- [4] www.mcselec.com
- [5] www.hpinfofotech.ro/html/cvavr.htm