Axel Wolf

Connecting the C166 architecture to CAN (II)

Thanks to their high performance, controller area networks (CANs) are being used on a growing scale for real-time applications in industrial and automotive electronics. The following descriptions of connecting a C165 to CAN, of development tools for the Siemens C166 family and a CAN demonstration model designed for trade fairs and workshops illustrate the versatility and benefits of networked microcontrollers. Part I of this article appeared in *Components* XXIX (1994) No. 5, pp. 42 to 44.

Processor features take pride of place in the C165. It represents a C167 with pared-down peripherals but undiminished computer power. The C165 can therefore be used to implement high-end embedded control applications in telecommunications and data processing (e.g. mass storage units and laser printers) at even lower cost. In contrast with the C167, the C165 contains no A/D converter and no PWM or capture/compare units. However, the two serial interfaces, the complex timers and the chip-select signals are still available. The C165 also retains the 2K-byte internal RAM and fast interrupt system (24 sources).

Connecting the C165 to CAN

The SAE 81C90/91 is an obvious choice for using the C165 in a CAN, just as it is for the SAB 80C166. In contrast with the first example shown in Fig. 1, however, the SAE 81C91 (without 8-bit I/O ports) was selected in the circuit shown in **Fig. 3**. It is connected to the C165 and its synchronous serial channel (SSC) via the synchronous serial interface (SSI), which is selected by applying a high level to the MS pin. In this example, the SSC should be configured as follows: single master/full duplex mode, master mode, MSB first, transfer data width = 8 bits, clock idle level = low, first edge of clock = data shift, second edge = data latch. The diversity of these setting options impressively demonstrates the flexibility of the C165/C167's SSC.

Every access to the SAE 81C91 starts by activating the module (CS = 0). No external address decoder logic is needed, as this function is performed by a port pin (e.g. P4.6). Pin P4.5 of the C165 then selects a read (W# = 1) or a write (W# = 0) operation via the W# input of the CAN controller. The address of the first data byte is subsequently written to the CAN controller by connecting the master transfer/slave receive (MTSR) pin to the data input (DI) pin. Depending on the operation, one or more additional data bytes can now be written to the SAE 81C91, or one or more data bytes can be read from the CAN module via the line between the master receive/slave transmit (MRST) pin and the data output (DO) pin. Finally, the SAE 81C91 is deactivated again (CS# = 1). The two controllers are synchronized via the SCLK-CLK connection.

The level applied to the timing (TIM) pin of the CAN controller decides whether data is output at the rising edge (TIM = 0, see Fig. 3) or at the falling edge (TIM = 1) of the CLK signal at the DO pin. Pin P4.7 of the C165 controls the power-down pin of the CAN transceiver, where this is present. On the CAN controller side, an inactive level is applied to pins WR#, RD#, and ALE, and to pins AD5 to AD7, which are not required when the serial connection between the C165 and the SAE 81C91 is used.

Development tools for the C166 family and CAN

The tried and tested development tools for the C166 architecture can be used to generate programs and verify the system.

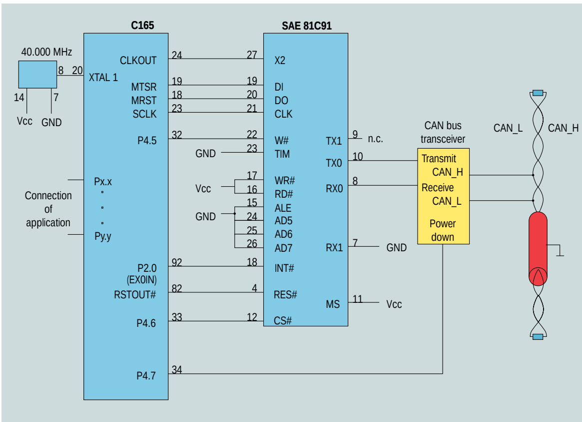

Fig. 3 Connection of a C165 to CAN via the SAE 81C91 (serial)

They contain not only C compiler support but also, of course, an emulation facility, even for modules with an integrated CAN module as in the C167CR. Software drivers are available as CAN-specific aids for simple implementation of CAN communication under control of the respective microcontroller. Interfaces for integration of CAN have already been implemented at operating system and HLL debugger level. Beyond this, evaluation boards are available so that users can assess the high performance of the C166 CAN solutions and start developing their own applications.

CAN demonstration model

The circuits shown in Figs. 1 and 2 (published in Part I) were developed by the author in connection with a thesis for a telecommunications degree at a college in Stuttgart. The aim was to develop a programmable demonstration model for use at trade fairs and workshops which would illustrate networking of Siemens microcontrollers with the aid of CAN (**Fig. 4**). This application made use of three microcontrollers: the C167CR, SAE 81C90 and SAB 88C166-5M (a flash EPROM derivative of the SAB 80C166). The hardware used in the model comprises three CAN bus user boards. Communication was represented visually with the aid of an automotive module containing lighting equipment, a small electric motor with a rotating disk and an LCD display, and acoustically with a loudspeaker.

Hardware

The hardware of the CAN demonstration model comprises the three bus nodes EVA167+NODE_167, NODE_166_1 and NODE_166_2 as well as the demonstration vehicle.

The bus node EVA167+NODE_167 (bus node 1) consists of the EVA167 evaluation board made by Ertec and supplementary board NODE_167. The Ertec evaluation board is a low-cost entry-level development tool for projects based on the C167 controller. A complete processor module is mounted on the board. Its standard version contains the following major components as well as the C167 controller: two RAM modules for application programs, a 40 MHz quartz oscillator and a 9-contact subminiature-D female connector matching standard PC port. All controller pins are run to a 160-contact male connector. The C167, which does not contain a CAN module, was replaced by the C167CR for this application, and the RAMs were replaced by EPROMs containing the software for this bus node. The supplementary board NODE_167 contains eight bounce-free function keys with integrated control LEDs, the CAN transceiver PCA 82C250 (from Philips), the LCD display, a 9-contact subminiature-D female connector and 9-contact subminiature-D male connector which plugs into the CAN bus. A potentiometer is included to allow "speed simulation". A 160-contact female connector provides the connection between the two boards EVA167 and NODE_167.

The NODE_166_1 board acts as the second bus node. It contains the SAB 88C166-5M controller used as a control processor, in whose flash EPROM the node's control program is stored. The CAN controller SAE 81C90 enables the SAB 88C166-5M to communicate on the CAN bus. Coupling to the bus is again provided by the CAN transceiver PCA 82C250. A quartz oscillator supplies the two controllers with the 20 MHz clock. A simple circuit allows the SAB 88C166-5M to be reset. The chip-select signal for the CAN controller is generated by the 74HC138 decoder.

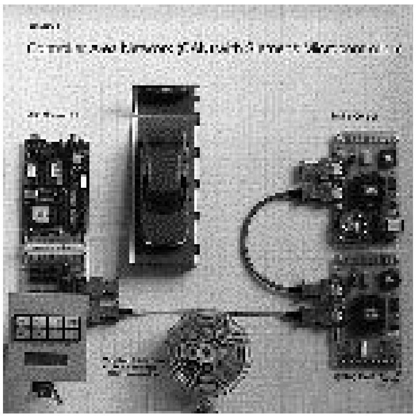

Fig. 4 This CAN demonstration model representing an automotive electronics applications illustrates the benefits of networked microcontrollers. It was specially developed for mobile presentations at trade fairs and workshops

Up to this point, the NODE_166_1 (bus node 2) and NODE_166_2 (bus node 3) boards are completely identical. They differ only in the components and circuits required for the various functions of the two representative subsystems controlled in the demonstration model. Bus node 2 controls the lighting of the demonstration vehicle, and contains eight transistors acting as drivers for this purpose. In contrast, bus node 3 is designed to simulate the engine management module of the demonstration vehicle; it contains an electric motor driving a rotating disk, and a function key. This bus node board also controls a loudspeaker.

The bus nodes are interconnected via CAN cables 1 and 2. Connectors with bus terminating resistors are located at bus nodes 1 and 3.

Software of the CAN demonstration model

The complete software for the CAN demonstration model was written in the high-level programming language C. The function keys can be used to send various instructions (e.g. "dip headlights") as messages from bus node 1 to the two other nodes via the CAN bus. These messages then trigger the appropriate operations, which are indicated on the LCD display and by the control LED on the relevant key. The potentiometer connected at bus node 1 allows the speed of the engine or rotating disk at bus node 3 to be altered; this option requires a continuous transmission of the relevant information via the bus.

One of the function keys allows the user to change the baud rate of the CAN. The function key at bus node 3 is designed to simulate a mode malfunction in the subsystem controlled, which is then reported to node 1 via the bus and indicated on the LCD display.

The ST10F166 flash programming board from SGS-Thomson was used to write the software for bus nodes 2 and 3 to the flash EPROM of the SAB 88C166-5M. After the final tests, the software for bus node 1 was stored in two EPROMs, which were then exchanged for the RAMs on the EVA167.

The CAN demonstration model was presented at the 1994 Hanover Fair, at the Japan Electronics Show and at *electronica '94*. It is currently being used at the Siemens Microelectronics Training Center as an instructional system for CAN courses. More detailed information on this application and the components presented here is available on request. □

Note on Part I, Fig. 1: The specification of the SAE 81C90/91 has been changed. CLKOUT should be connected to pin 10 (not 11), i.e. at X2 (not X1).