iCC 1995

2nd international CAN Conference

in London (United Kingdom)

Sponsored by

Motorola Semiconductor
National Semiconductor
Philips Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

R. Dietz
Robert Bosch GmbH, Dep. FV/FLM
Postfach 106050, 70049 Stuttgart

F. Hartwich, P. Friedrichsohn
Robert Bosch GmbH, Dep. K8/EIS
Postfach 1342, 72703 Reutlingen

An evaluation chip for smart sensors with an integrated CAN interface

Abstract

The smart sensor concept offers the benefit of a digital error compensation of non-ideal sensor characteristics and temperature influences. For this purpose, a dedicated correction processor was implemented which uses individual calibration data organized as a two-dimensional characteristic diagram to calculate the true measurement value. Sensor values up to 14 bit are supported. The characteristic diagram can be set up in an end-of-line calibration process using a serial EPROM data interface.

The calculated sensor value is transmitted via a digital, bus-capable sensor interface which offers the possibility of interference-free signal transmission and supports future multi-sensor structures. In the presented implementation, this task is performed by a CAN protocol controller with a reduced message memory, thus leading to a minimized chip area.

In the design process, the CAN protocol's reference environment, consisting of a protocol controller model, a testbench, waveforms and a simulator kernel was used to verify the implementation of the CAN interface. By describing the other parts of the chip including test patterns in the same high level language, the whole chip could be verified on functional level before breaking down the design to the schematic entry of the selected IC-CAD. Testvectors automatically generated by the functional model ensured the correctness of this transformation. The chip named CC400 was fabricated in a 1.2 µm CMOS process with EPROM option.

1. Introduction

In conventional sensor systems, the mechanical sensor element and the electronic evaluation circuit are generally operating at separate locations. Smart sensors, however, integrate the mechanical and electronic components into a single system. With the electronic circuit operating on site at the same temperature as the sensor element itself, a common error compensation for systematic errors and temperature influences is possible. As the calibration process can be handled very easily in the electronic circuit, a charge-specific or even a specimen-specific calibration is possible.

Fig. 1 shows the structure of a smart sensor application: in addition to the main sensor element which transforms the process quantity z into an electrical signal x, a temperature sensor generates a second signal y depending only on the sensor temperature T. This sensor element can be realized by a cheap resistor as no linear characteristic is required. The third part is the smart sensor evaluation circuit which consists of four main blocks: The *measurement unit* has two channels in order to evaluate x and y. The data stored in the *calibration memory* consist of parameters describing the non-ideal characteristics x(z,T) and y(T). This information is used by the *correction processor* which evaluates the measured values x and y and calculates the true value for the process quantity z. The result of this calculation finally is transmitted via the *sensor interface*.

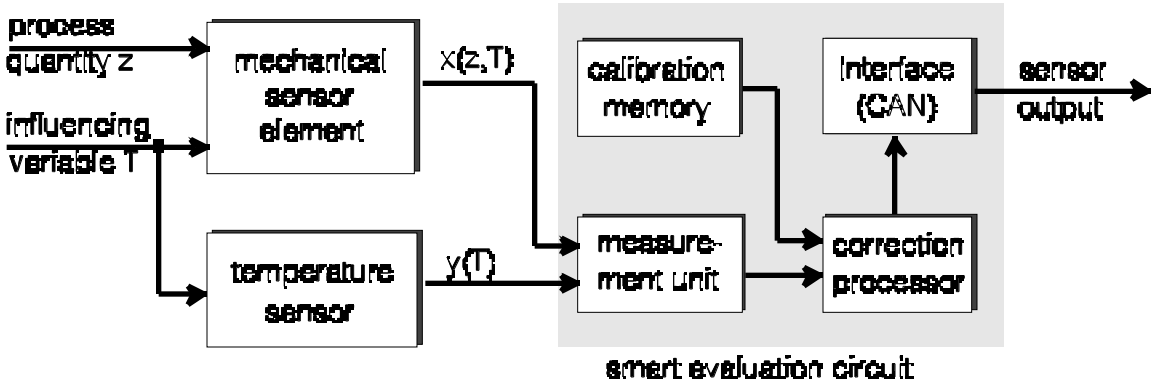

Fig. 1: Smart Sensor Concept

As the correction processor operates with digital values, the measurement unit must perform an analog to digital conversion. The digital signal processing has several advantages compared to an analog processing. For instance, the interference-insensitive digital signal transmission is more reliable and, if the digital sensor interface is bus-capable, wiring in multi-sensor applications is reduced essentially. In addition, new hardware concepts with decentralized intelligence, e. g. for diagnosis purposes, become possible.

For the sensor interface, the well known CAN protocol was chosen. It was originally developed for the communication between control units in the motor vehicle, but has also been used successfully in industrial field bus applications. A CAN controller module with basic message handling functions is well suited for sensor bus connection. The module, requiring limited chip area, can be integrated together with the correction processor.

In many sensor applications, especially in automotive applications with a high volume of produced items, the cost factor is essential for the acceptance of a sensor concept. As the chip costs correspond directly with the chip size, a main target for the development of the smart sensor evaluation circuit was to minimize chip area. Not only the CAN controller, but likewise the correction processor and the measurement unit had to be developed considering this main objective. These three parts of the evaluation circuit are briefly described in the next

sections. For the calibration memory, a standard EPROM with a serial programming interface was used, which is not described in detail.

2. Measurement unit

Both channels of the measurement unit are able to evaluate sensors which convert the sensor input to a variable resistor, capacitor or inductivity. The sensor element must be connected with a non-variable element in order to form a LR- or a RC-pair. This pair is driven by a rectangular voltage as shown in Fig. 2. The output voltage of the LR- or RC-pair is fed back to the CC400-chip by another pin and is internally connected with two comparators. If the sensor output reaches the voltage levels U_H and U_L (defined by two other pins), the driving voltage is switched on or off, respectively. The time τ_P for a full period can be calculated by the equation

$$\tau_P = \tau \left(\ln \frac{U_0 - U_L}{U_0 - U_H} + \ln \frac{U_H}{U_L}\right), \qquad (1)$$

where U_0 denotes the amplitude of the driving voltage and τ the time constant given by L/R or R C. If the reference voltages are chosen to $U_L = U_0 / 3$ and $U_H = 2\,U_0 / 3$, equation (1) can be simplified to

$$\tau_P = 2\,\tau \ln 2. \qquad (2)$$

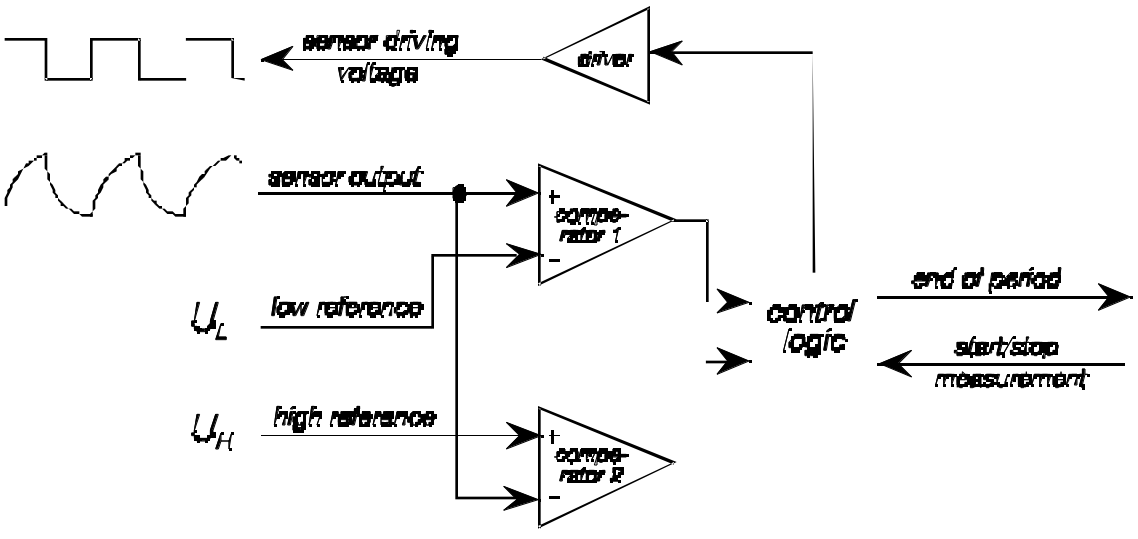

Fig. 2: Principle of measurement

The number of periods is programmable within given limits. The whole time for all periods is measured by a counter running with a count frequency derived from the system clock. The frequency divider is also programmable.

The measurements for both channels are done sequentially so that the same measurement counter can be used for both channels. The maximum resolution of this time measurement is 14 bit. As the reference voltages, the non-variable external elements and the programmable number of periods can be chosen independently for each channel, the measuring system can be adapted to a wide range of different sensor types.

3. Correction processor

The data stored in the calibration memory are parameters which characterize a two-dimensional function which we call sensor correction function. This function gives the true measurement value z depending on the mesured values for x and y. The correction processor has to perform the calculation of the function value z(x,y) for any given values for x and y.

The complexity of this calculation depends on the way how the correction function is represented by the stored parameters. Different methods have been examined and evaluated as described in /1/. The so called 4-point-approximation method allows a highly accurate representation of the correction function and leads to a simple calculation task for the correction processor. Thus a dedicated processor requiring less hardware can be used.

With this method, the two dimensional error correction function is represented by function values at equidistant support points. All other function values are calculated by the interpolation scheme shown in Fig. 3: The function values at the 4 surrounding support points define a bilinear area which approximates the correction function.

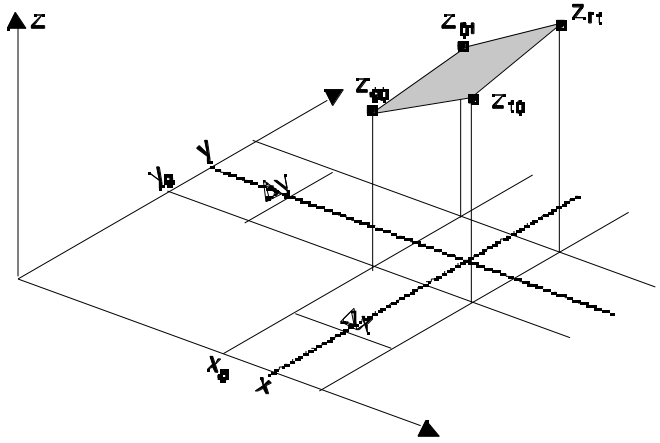

Fig. 3: Interpolation method

This leads to the following term which must be calculated by the correction processor:

$$z(x,y) = z_{00} + \frac{(z_{10} - z_{00}) \cdot (x - x_0)}{\Delta x} + \frac{(z_{01} - z_{00}) \cdot (y - y_0)}{\Delta y} + \frac{(z_{00} + z_{11} - z_{10} - z_{01}) \cdot (x - x_0) \cdot (y - y_0)}{\Delta x \cdot \Delta y} \quad (3)$$

As the distances Δx and Δy between the support points can be chosen as powers of 2, the divisions in (3) can be executed by simple shift operations. The substractions and multiplications can also be reduced to additions /2/, thus the most complex part of arithmetic unit of the dedicated correction processor is a 15-bit adder which can handle values up to 14 bit. The most significant bit is used as sign bit. The number of bits for the function values z and for the measured values x and y can be adjusted within certain limits by the configuration data also stored in the calibration memory. The chip configuration process allows to set a lot of operating conditions which must be adapted to a given sensor application.

The structure of the simple operation unit is shown in Fig. 4. Altogether, the instruction set consists of about 40 simple instructions requiring exactly one clock cycle. The microprogram, consisting of about 140 instructions is stored in the ROM matrix of the controlling unit. Depending on the configuration data, some code sequences must be repeated several times, but in all cases, the desired calcalation is completed after 232 program steps. Given a clock frequency of 10 MHz, the calculation will last less than 23,2 µs. The implementation in the chosen CMOS process requires 2,25 mm^2 for the operating unit and 0,8 mm^2 for the controlling unit including the ROM matrix.

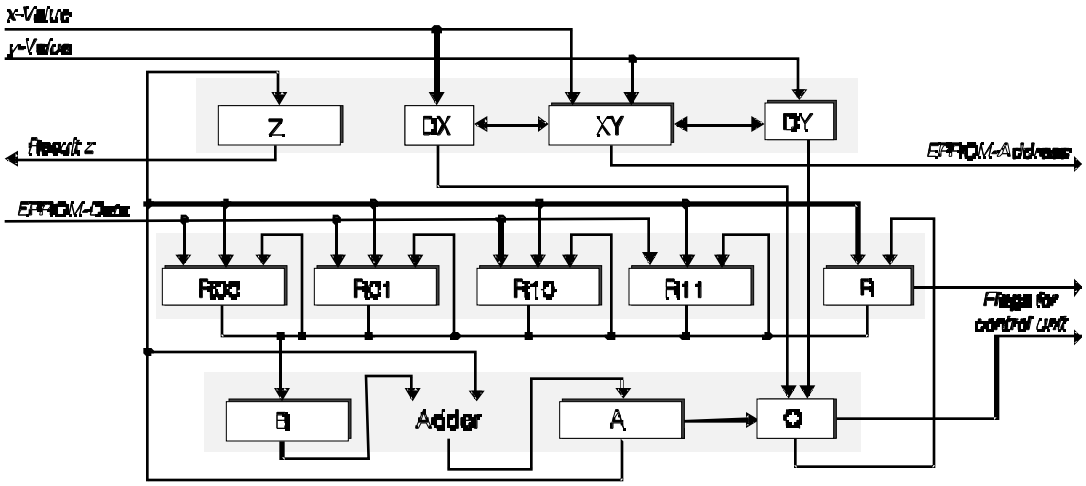

Fig. 4: Operation unit of correction processor

4. CAN Interface

The CAN module is an on board supplement to the sensor controller. It communicates using the CAN protocol 2.0 part A (tolerating part B) at a data rate of up to 125 KBit/s. For the connection to the CAN bus, a standard CAN bus driver is required.

Different from usual CAN controllers, this CAN module is not designed to support the message transfer of a CPU, but is limited to the handling of one single message object, thus representing the minimum implementation of a CAN interface as shown in Fig. 5.

The necessary configuration of the CAN module (defining the CAN bit time, the message object's identifier, and the handling of remote frames) is done in the startup phase after the

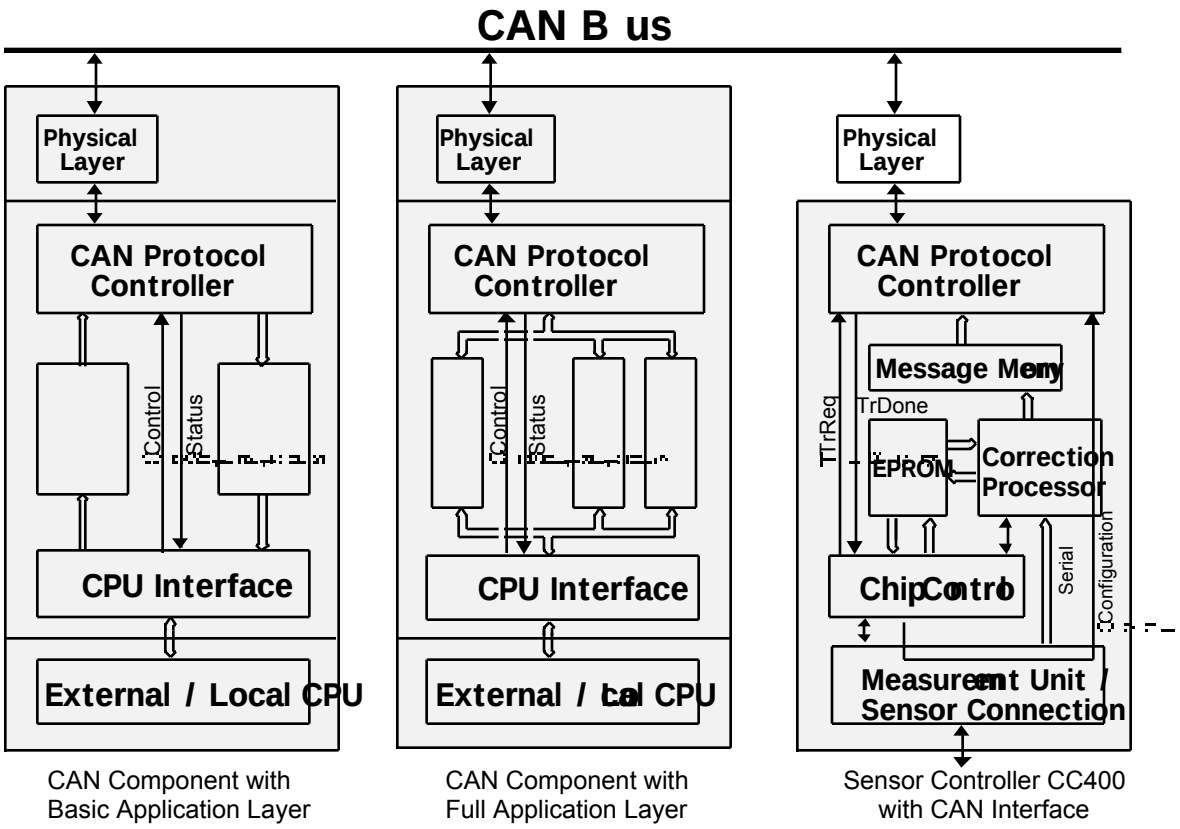

Fig. 5: Structure of different CAN controllers

power-on reset. For this purpose, the chip control unit which is responsible for the interaction of all other function blocks reads data from the on board EPROM and transfers the data to the CAN module using a serial bus. At the end of the configuration, the CAN module goes on-bus and starts the CAN protocol handling.

After the configuration, the communication with the CAN module is limited to the sensor data and to the signals TrReq (transmission request) and TrDone (transmission done) as shown in rightmost diagram of Fig. 5. The chip control unit sets TrReq each time new sensor data become valid; the CAN module sets TrDone each time a data frame is successfully transmitted. The actual transmission of the message containing the sensor data starts either at TrReq or, if so configured, triggered by matching remote frames.

The data length of the sensor data message is two bytes, representing the 14 bit corrected sensor output and two state bits. The sensor data bits in the message memory are updated each time a new conversion is done. The state bits give other CAN nodes in the network the possibility to monitor the internal state of the CAN module (levels of the receive and transmit error counters).

5. Chip design

For the design of CAN protocol controllers, Bosch has developed a reference environment, consisting of a protocol controller model, a testbench, a set of waveforms and a simulator kernel to verify all CAN protocol functions of the new CAN implementation as described in /3/. The verification is done by the comparision of the function of the implementation's model with the function of a reference model in a simulated CAN network. This CAN network consists of several instances of the reference model acting as receivers and transmitters, thus providing the implementation's model with CAN input signals. Additionally, the waveforms have the possibility to directely set the CAN bus to recessive or dominant values, to check the implementation's reaction to CAN bus errors. This reference is distributed to all CAN licensees, assuring the compatibility of all CAN implementations.

In this design, the CAN protocol's reference environment was used not only for the development of the CAN module, but for the verification of the whole chip. The testbench and the set of waveforms have been expanded with new waveforms and with functions to stimulate the sensor inputs. This is shown in Fig. 6. In this way, a coherent high level language description and simulation of the design was possible. The verification of all functions could be done in the simulated CAN network. In a typical configuration, the waveform sets the sensor to a certain value and waits for the sensor controller to convert the new input and for the CAN module to transmit the new data. A second reference CAN controller in the network, receiving the transmitted data frame, provides the waveform with the response data to be checked.

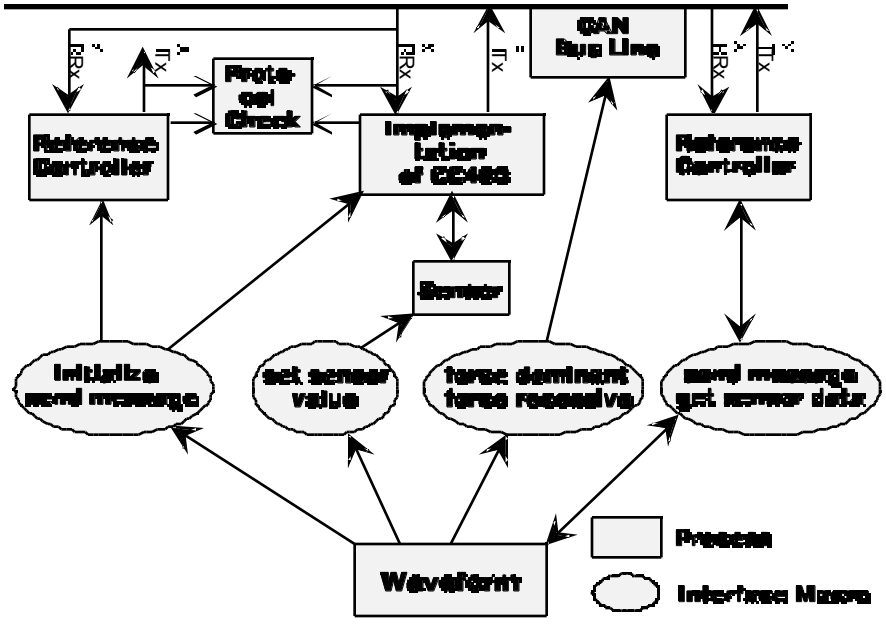

Fig. 6: Testbench of CC400

During the simulation of waveform and implementation, a protocol of the implementation's input and output signals is written into a file. This way, the verified high level language model provides test vectors for the verification of first the transistor netlist and then the silicon samples.

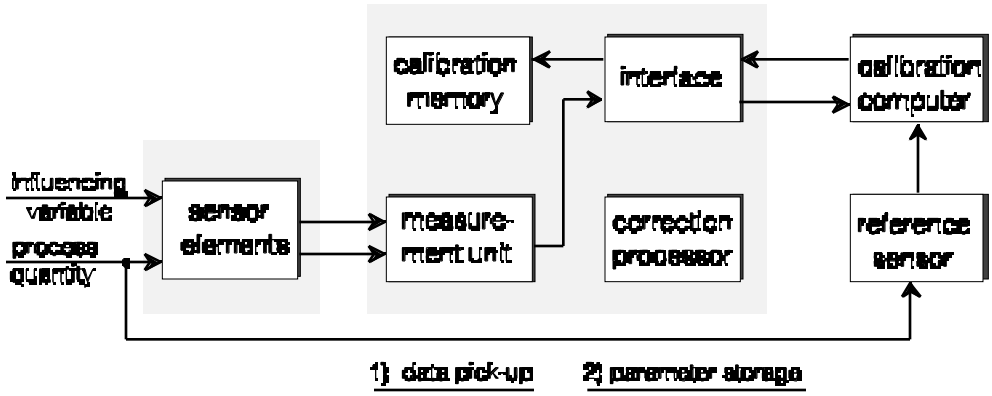

Fig. 7: Calibration process

6. Chip calibration

In general, a smart sensor calibration process consists of two phases as shown in Fig. 7: In the data pick-up phase, only the measurement unit and the sensor interface are used. The measured data of both channels are transmitted as "raw" data to an external calibration computer. Given the reference information for a set of operation points, the calibration computer can calculate the sensor specific correction data. In the second phase, the calculated parameters must be stored in the calibration memory of the sensor.

In order to start with the pick-up phase, the CC400 can be switched to a special calibration mode. The measured values for x and y are transmitted uncorrected via the CAN interface. This calibration can be done without pre-programming the EPROM as another mode allows the configuration of the whole chip via a serial interface. In the programming mode, this serial interface is also used to transmit the programming data to the EPROM of the CC400.

The handling of the different operating modes of the CC400 is supported by a comfortable application system. The application system communicates with the CC400 using a 82527 CAN controller. The user-interface of the application software runs on a standard PC which is connected to the application system by a serial interface line.

7. Conclusions

The CC400 implements both an accurate sensor evaluation and a comfortable interface realized by a CAN protocol controller. A main advantage of the smart sensor concept realized by the CC400 is the possibility of a common error compensation of sensor and evaluation circuit. The programmable on-chip-EPROM allows an end-of-line sensor calibration and an adaption to a wide range of different sensor types and sensor characteristics. The sensor calibration process is supported by a PC-based application system.

8. References

/1/ R. Dietz, E. Zabler, F. Heintz; "An Efficient Error Correction Method for Smart Sensor Applications in the Motor Vehicle"; SAE International Congress and Exhibition, Detroit, USA (1993), Technical paper no. 930357, printed in Sensors and Actuators SP-948 (1993), pp. 105-110.

/2/ R. Dietz, E. Zabler, F. Heintz; "Prozessor für die Fehlerkorrektur intelligenter Sensoren in einem zweidimensionalen Kennfeld", tm - Technisches Messen 59 (1992), Heft 9, Seite 353 - 360.

/3/ F. Hartwich, E. Esch, M. Strugala; "Modelling of the CAN Protocol in C and VHDL, a way to maintain protocol consistency in IC implementations", Proceedings of the 1st International CAN Conference (1994), pp. 5-17 - 5-25.