

BULETINUL INSTITUTULUI POLITEHNIC DIN IAȘI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iași
Tomul LVII (LXI), Fasc. 5, 2011
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

A NEW CONTROLLER AREA NETWORK SIMULATOR FOR A RAIN SENSING SYSTEM USING LABVIEW

BY

MONICA SĂLCIANU*, CRISTIAN FOȘALĂU and ANAMARIA HARITON

“Gheorghe Asachi” Technical University of Iași,
Faculty of Electrical Engineering, Energetics
and Applied Informatics

Received, June 14, 2011

Accepted for publication: August 26, 2011

Abstract. The Controller Area Network (CAN) has been used for automotive applications as a method to enable robust serial communication. The paper is focused on the simulation of the communication between a rain sensor and CAN bus, from a messages view point. The communication between sensor and CAN bus is monitored based on messages exchange between sensor and Electronic Control Units (ECU). The simulation monitors the messages vehiculated on CAN bus and translates, for the user (driver or designer), the information sent by sensor and by ECU.

Key words: CAN bus; rain sensor; sensor simulation; CAN communication; optical detection.

1. Introduction

The electronic systems in the motor vehicles and also the size of encapsulated software components have had a considerable growth in the last two decades. In addition, these developments have led to the use of small components with fewer mechanical and hydraulic parts, thereby reducing size and cost and increasing performance. Increasing the number of encapsulated

* Corresponding author: *e-mail*: helgiu@yahoo.com

electronic components and their functions within the network, reflects the increasing complexity of vehicles. The performance requirements and cost are higher and the production time is narrowing. Designing these systems is becoming increasingly important and difficult, requiring new mechanisms for hardware architecture design and software components in the automotive industry.

Efficient and high-speed communication network in the automotive industry is crucial in order to provide for better performance. To meet these requirements there are various communication protocols. Currently, the most widely used communication protocol is the CAN protocol. The most sensors have communication interface with the CAN bus. An automotive application contains one or more electronic control units (ECUs). An automotive control system contains over 70 ECUs requiring a distribution of more than 2,500 signals. Therefore a well established communication protocol is needed (Makowitz & Temple, 2006).

The requirements for a network communication in the automotive industry come with applications that have to bear. Fig. 1 presents the components of a generic node, an example of connecting a Smart Sensor (controlled by the local microcontroller) to a CAN bus. In this node, the signal delivered by a sensor is applied to a signal conditioning circuit and then to an A/D converter. Depending on the node, the microcontroller can immediately send information on the bus. This information can be used by other nodes or can expect to receive a value above a set limit and then send the data on the bus. The CAN Transceiver converts the logic signals from the microcontroller in signal levels used by the physical CAN bus. A typical smart sensor is made using digital and analog components, by which data is captured, processed, analysed and transmitted to other nodes in the system. The designers usually create generic hardware components that can be configured for different applications (ISO 11519-3, 1994).

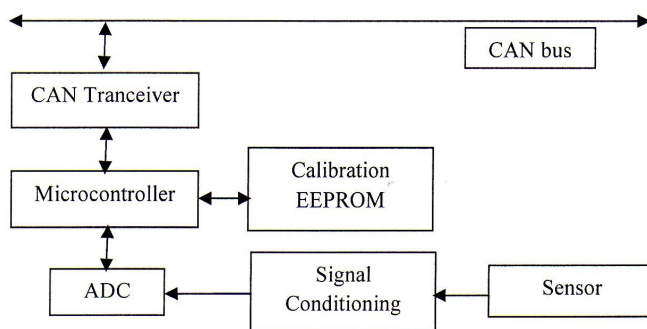


Fig. 1 – Interfacing the sensor with the CAN bus.

Currently, there are a variety of simulators for each sensor used in the automotive industry. These simulators allow measurement of signals sent by

sensors but also allow their simulation. For example the ADD71 simulator produced by KD Auto Scanner Factory Co., Ltd., is widely used in the automotive industry nowadays. It has the following specifications:

- a) measures car sensor signals;
- b) allows simulation of the majority of in-vehicle sensors;
- c) verifies the basic operations of the ECU;
- d) eliminates unnecessary replacement of sensors;
- e) simulates actual operating conditions of each sensor without disconnecting them;
- f) allows the verification response of ECU functionality, using the control of the signals delivered by the sensor.

ADD71 simulates the actual working conditions without eliminating any sensor inside the vehicle. The systems for which the simulator is used are: ABS (Antilock Braking System), Crank, coolers, Oxygen, MAP (Manifold Absolute Pressure), MAF (Mass Air Flow), MAT (Mass Air Temperature), VSS (Vehicle Speed Sensor), etc.(Instruction manual).

STS-600 is another type of simulator manufactured by KD Auto diagnostic Co., Ltd. STS-600 is a testing device with POST (Power on Self Test), being used to test voltages, resistances and frequency of the signals delivered by sensors and to simulate the output signals (voltage, frequency).

The presented simulator is based on messages exchange between a sensor and the control unit. When a new sensor must be introduced as a node on the CAN network, some problems may occur at the integration phase: identifier length, identifier uniqueness, messages periodicity. These problems can be detected from the beginning by using the presented simulator. It does not monitor the electrical signal level, but the messages sent by each active node on the bus. The messages correctly interpreted and an efficient error handler may highlight the importance of the error detection design of the node.

2. Sensor – CAN Communication Simulation

The paper deals with the simulation of a rain sensor connected to the CAN in-vehicle network. This type of sensor is found in 80% of currently produced cars. The rain sensor used in our experiment is produced by TRW. Without diverting the driver attention, this fully automatic sensor keeps the windshield clean, allowing the driver to concentrate on the road. The sensing element consists of a humidity sensor based on an optical principle, which detects and determines the intensity of rain or snow. Its role in the system is to drive the decision for wiper activation and speed. Using the technology offered by TRW, this sensor provides to the driver an extra safety and comfort in various weather conditions. Fig. 2 shows the schematic functional block of the rain sensor type 7812 produced by TRW. The optical detection unit is based on the reflection of infrared beam. After a suitable analog and digital processing,

the signal passes through a process control algorithm, after which the CAN transceiver will send data on the bus.

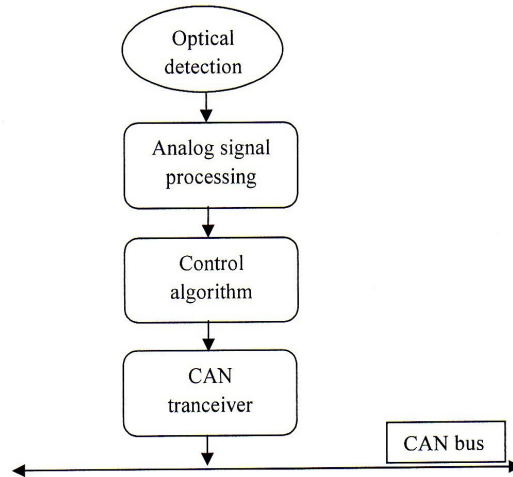


Fig. 2 – Rain sensor schematic functional block.

2.1. Sensor Operation

An infrared beam is reflected on the outer surface of the windscreen by an array of infrared sensors. When the moisture reaches the windscreen, the beam is broken out by the humidity barrier. The beam intensity is modulated by the moisture consistence, thus delivering the signal which depends on the rain intensity. Furthermore, the sensor communicates with the wiper control unit, which acts upon the wiper speed. Fig. 3 explains how the sensor works (Hazelden, 2010).

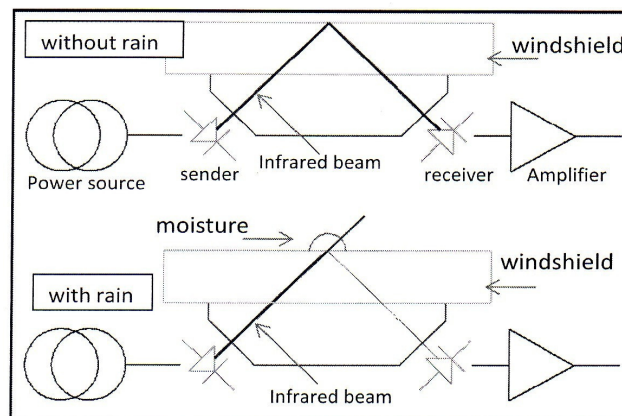


Fig. 3 – Schematic representation of the sensor system operation.

Several features of this sensor are the followings: vehicle specific requirements can be added without impacting onto the core technology, adaptability to any type of glass with no additional calibration and an advanced software control algorithm provided by the manufacturer.

The LabView programming environment is used because it gives the possibility to create an optimal interface for the network designer. Another advantage of LabView is the easy way in which the designer can modify the simulation, by adding new functions. The simulation was performed at the message level, monitoring identifiers, message content and message cycles.

2.2. Sending the Message to the CAN

The message sent by the sensor has an identifier that is initially introduced in the database of the ECU. Each message identifier which appears on the CAN bus is sought in the database. After it is recognized, the ECU responds in accordance with the specifications. The CAN message is sent with the periodicity indicated in the database. If the identifier is not recognized, the ECU doesn't respond.

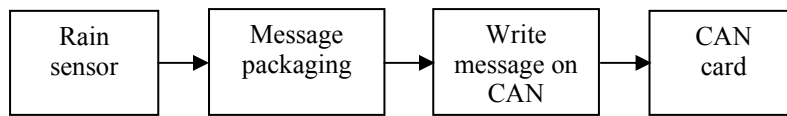


Fig. 4 – Steps for sending messages on CAN.

According to Fig. 4, after the value provided by the sensor composes the message, it is packed using a packaging protocol (in this case TP2.0) and then sent through the CAN hardware device, named CAN Card. This tool is used to monitor the CAN bus. It also provides the ability to view all the messages on the connected CAN bus.

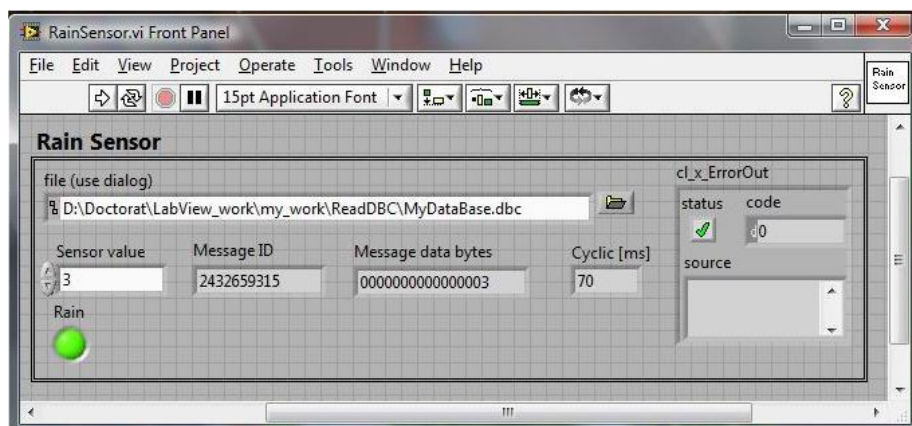


Fig. 5 – The interface used by the designer to simulate the messages sent by the rain sensor on the CAN bus.

Fig. 5 shows the interface for sending the message provided by the sensor on CAN. The message contains the amount of rain sensor ready encoded in digital value. The control unit which processes the data sent by the sensor is programmed so that, based on the input signals, it controls the activation and speed of the wipers. The electronic control unit includes a microcontroller, a memory and the CAN interface. It receives the messages from the sensor and sends the control command to other units or sensors.

Fig. 5 represents a front panel built in the LabView graphical programming environment, in which the entire simulation has been developed. On this panel, the first numerical control contains the value sent by the sensor. The values delivered by the sensor are comprised in the range of 0 to 10, where 0 means the absence of humidity and 10 is the maximum level. A value of 0 indicates “dry windshield”, for which the warning LED is off. 1, 2 or 3 values indicate a low intensity rain, 4, 5, 6 and 7 values indicate a moderate rain, whereas heavy rain is indicated by values 8, 9 and 10. The LED, named “Rain”, is turned on, only if the value of the sensor is different of 0. The “Message ID” contains the message identifier sent on the bus, while “Message data bytes” contains the value sent by the sensor. The “Cyclic” indicates that the sent message is a periodic one, the value being retrieved from the messages database. If the sensor sends a value other than 0, the LED flashes indicating a warning. On the panel the path of the database file is also pointed out.

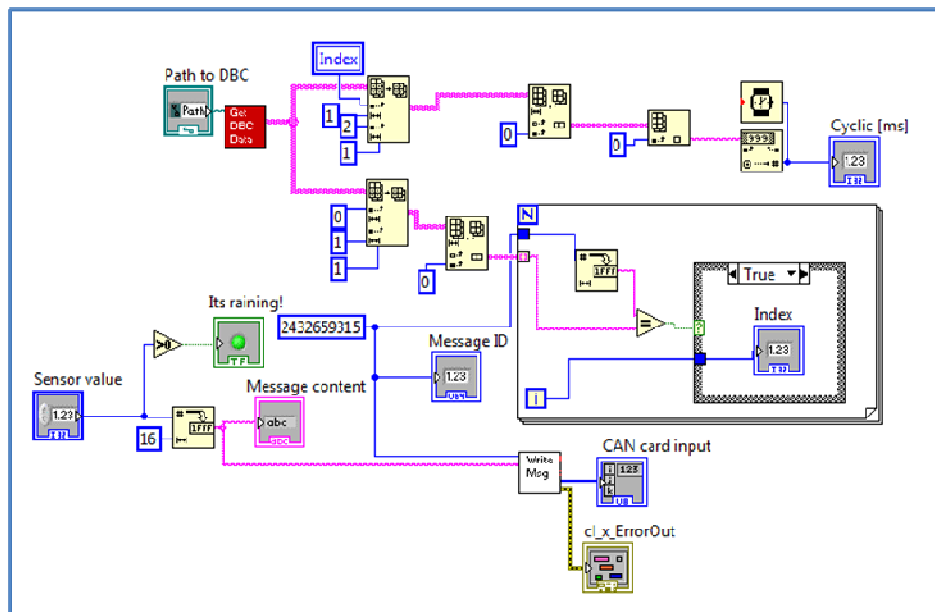


Fig. 6 – The block diagram of the simulator.

The identifier length is of 5 Bytes and the message length is of 8 Bytes. In this case, byte 0 contains the value indicated by the sensor. The periodicity of

the message to be sent is of 70 ms. The message is packed using the TP2.0 protocol and then sent on the CAN bus.

In the Fig. 6 is presented the block diagram of the simulator. It contains a sub-VI which simulates the creation and the transmission of the CAN message, named “CAN Msg” and a sub-VI used for getting the information from Data Base Container (DBC), named “Get DBC Data”.

2.3. Receiving Message by ECU

The received message is interpreted by ECU, having as main goal to announce the user about the external conditions, either by electronic display or by lighting the warning LED. The steps needed to receive the message by ECU are shown in Fig. 7. Depending on the contents of the message, ECU can take some decisions such as starting wipers, automatic closing windows or closing the hatch.

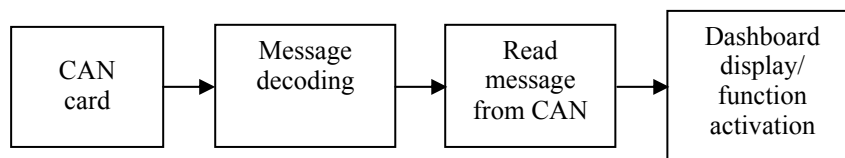


Fig. 7 – Steps needed for receiving the message.

Fig. 8 shows the content of the message received by ECU. The ECU decodes the content of the received message, in order to send an info message to the user. The information can be displayed on the dashboard or can be a voice message. Entire CAN communication is saved in a text file. In this way all the

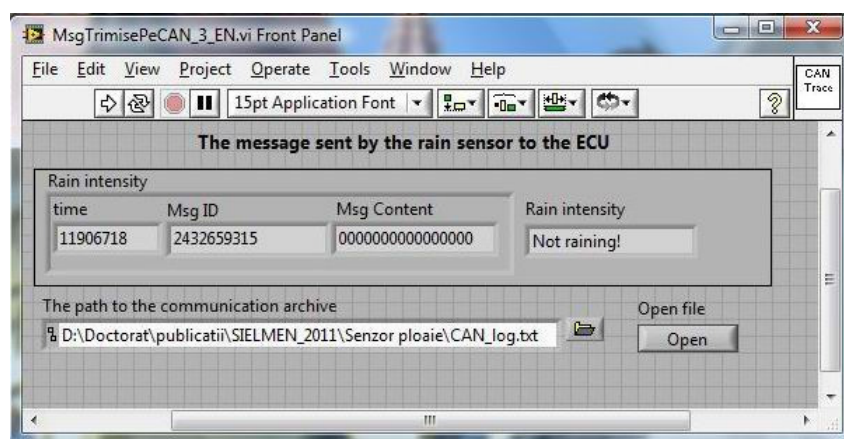


Fig. 8 – Front panel containing details about the message received by ECU.

messages sent on the bus can be analysed in terms of transmission time. The path to this file is shown by the control “The Path to the Communication

Archive”. The time indicator shows the moment when the control unit receives the value sent by the sensor, this time being measured in ms. The “Msg ID” displays the received message identifier and the indicator “Msg Content” displays the content of the received message. Based on its content, the ECU interprets and displays the result to the user.

3. Results and Discussion

By analysing the file containing information about the messages sent on the bus, one can draw conclusions regarding the correctness of communication on CAN. Fig. 9 presents such a file (CAN_log.txt) in which data are recorded in three columns. The first column contains the time when the message is sent, the second column represents the identifier of the message, and the third column represents the content of the message transmitted to bus.

TIME	Message ID	Message content
1540853	2432659315	0000000000000000
1540926	2432659315	0000000000000000
1540999	2432659315	0000000000000000
1541072	2432659315	0000000000000000
1541145	2432659315	0000000000000000
1541218	2432659315	0000000000000000
1541292	2432659315	0000000000000000
1541365	2432659315	0000000000000000
1541438	2432659315	0000000000000000
1541511	2432659315	0000000000000000
1541584	2432659315	0000000000000000
1541657	2432659315	0000000000000000
1541730	2432659315	0000000000000001
1541803	2432659315	0000000000000001
1541877	2432659315	0000000000000001
1541950	2432659315	0000000000000001
1542023	2432659315	0000000000000001
1542096	2432659315	0000000000000001
1542169	2432659315	0000000000000001
1542242	2432659315	0000000000000002
1542315	2432659315	0000000000000002
1542388	2432659315	0000000000000002
1542461	2432659315	0000000000000002
1542534	2432659315	0000000000000002
1542609	2432659315	0000000000000002
1542682	2432659315	0000000000000002
1542755	2432659315	0000000000000002
1542828	2432659315	0000000000000003
1542901	2432659315	0000000000000003
1542976	2432659315	0000000000000003
1543049	2432659315	0000000000000003
1543122	2432659315	0000000000000003
1543195	2432659315	0000000000000004
1543268	2432659315	0000000000000004
1543342	2432659315	0000000000000004
1543415	2432659315	0000000000000004
1543489	2432659315	0000000000000004

Fig. 9 – The file containing the archive of CAN communication: *a* – for 804 ms, the content of the message received by ECU from the rain sensor is 0 (not raining); *b* – for a period of 439 ms, the sensor value is 1 (light rain); *c* – during the 513 ms, the amount received from the sensor is changed to the value 2 (rain intensity increases); *d* – for 294 ms, ECU received message containing a value of 3; *e* – for 699 ms, the value sent by the sensor is 4 (rain intensity is moderate).

The file is generated in reception block. Each message received by the ECU is saved together with a time of reception and its identifier. By analysing the first columns it can be observed that the “Intensity Rain” message is not sent exactly with a periodicity of 70 ms, but with 73 ms. This is a disadvantage of the simulated communication because the simulation development environment (in this case LabView) introduces supplementary unwanted delay times. In real communication, however, messages are sent exactly with the periodicity established in the database, accepting deviations of only ± 1 ms. In this case, the second column contains the same identifier as when it sends a single message for 3,005 ms. As stated above, the third column contains data from the sent message. According to the file presented in Fig. 8, in a period of 804 ms the sensor value is 0, so no rain, then for 439 ms the sensor value is 1, meaning light rain. For 513 ms, the sensor value is 2, for 294 ms the sensor indicates the value 3, while for the remaining 699 ms the value is 4, when the rain began to be moderate.

The time between the rain states is so shorter because the sensor is simulated and the values are changed by designer in a very short time. In the real conditions, when the rain sensor send the values, the time between the rain states is greater (tens of seconds) because the sensor process the humidity level and then it sends the values in a few seconds.

4. Conclusions

A software package designed to simulate the signals flowing through in-vehicle CAN buses is described. The simulator operation is exemplified with the communication between a rain sensor and the ECU, highlighting the way in which the message is coded at sending point and then how it is decoded at receiving and recorded in a history file. A problem to be solved in future is the time delay introduced by the simulator programming environment, which induces discrepancies between the ECU database and the simulated signals. A solution for this problem would be the use of a Real Time Operating System which guarantees the accuracy of time loops within an error of $1 \mu\text{s}$.

Other possibilities of faulty communication on the bus are: periodicity disruption of the messages on the bus, skipping one or more sending messages, updating incorrect or delayed update of the content of the message and even incorrect integration of new introduced sensor. This simulation can be used with educational purposes and may be part of a complex application containing communication with more sensors and more control units.

REFERENCES

- Albert A., Hugel R., *Heuristic Scheduling Concepts for TTCAN Networks*. 10th Internat. CAN in Autom. Conf. (ICC), Catania, Italy, 2005.
- Hazelden R., *Optical Sensors in Automotive Applications, Solving Sensing Problems with Photonics*. Photonex 2010, November 3, 2010.

- Makowitz R., Temple C., *FlexRay – A Communication Network for Automotive Control Systems*. Factory Commun. Syst., 2006 IEEE Internat. Workshop on Factory Commun. Syst., Torino, Italy, 2006, 207-212.
- * * *Automotive Multimeter ADD81/ADD91/ADD71*, Beijing ADD-TECH Co., Ltd. Instruction Manual.
 - * * <http://kdscanner.en.ec21.com>.
 - * * <http://www.cnreliable.com/Auto-Testing-Series/Auto-Testing-Series-00356A.html>.
 - * * <http://www.trw.com/>.
 - * * http://www.trw.com/sub_system/rain_sensor.
 - * * *Road Vehicles-Low Speed Serial Data Communication. Part 3: Vehicle Area Network (VAN)*, ISO 11519-3, 1994.

UN SIMULATOR NOU PE CAN PENTRU UN SISTEM DE DETECȚIE A PLOII UTILIZÂND LABVIEW

(Rezumat)

Rețeaua Controller Area Network (CAN) a fost utilizată, pentru aplicații în domeniul automobilelor, ca o metodă de a obține o comunicație serială rezistentă. Lucrarea se axează pe simularea unui senzor de ploaie și interconectarea sa cu bus-ul de CAN, analizată din punct de vedere al mesajelor vehiculate pe bus. Comunicația dintre senzor și CAN este monitorizată pe baza mesajelor vehiculate între senzor și unitățile de control electronice (ECU). Simularea monitorizează mesajele de pe bus și traduce, pentru utilizator (conducătorul auto sau proiectant), informațiile trimise de senzor și de ECU.