doi: 10.5937/str2302042L

The Analysis of Operation of CAN Communication Using Different Types of Sensors

Đorđe Lazarević [1)]
Jovan Ćirić [2)]

This paper presents the analysis of CAN communication protocol operation which actually is a communication between the nodes utilizing different types of sensors. The system consists of multiple nodes with microcontrollers of different architectures. One node is a control node and other nodes send data to it. The system is implemented and checked in the laboratory. For signal analysis,a digital oscilloscope and logic analyzer are used. Also, the obtained results and signal diagrams are presented as well.

Key words: **CAN, microcontroller, oscilloscope, logic analyzer.**

Introduction

CAN (Control Area Network) bus was developed in the late 1980s, by Bosch company and it became the international standard ISO11898. This technology enables the connection of cable equipment at several points and provides a significant saving of cables instead of a conventional direct-wired connection [1]. As a standard, it is designed to enable communication between the microcontroller and other components or parts of the system without a host computer. It is a message-based protocol designed originally for multiplexing conductors in automobiles and its main purpose is to save copper, but it can be used in other contexts. For each component or device, the data frame is sent sequentially, but in such a way that if more than one device sends data at the same time, the device with the highest priority can continue the operation, while the others are turned off. Frames are sent and received on all devices within the network. Application of the CAN protocol is most common in cars, i.e. in the automotive industry, while it is also used in industrial automation, elevators, medical instrumentation, etc. Modern cars have more than 70 control units for different subsystems. Typically, the test node is the motor control unit while the other nodes are transmission, airbags, servo, fuel flow control, ABS, and others [2]. This bus protocol is used for communication and exchanging information between ECU, which present one of the core components in the electronic system in vehicles. CAN protocol is one of the most widely used bus protocols and it is public as another bus protocol [3]. A twisted-pair cable and 120-ohm resistors are used in practice to implement the CAN bus. There are also nodes on the bus that contain a microcontroller unit (MCU) that controls all procedures and manages devices on the CAN bus. The conversion of data provided by the MCU into a CAN message is the responsible CAN controller [4]. Low cost, high utilization rate, good real-time performance, and high reliability are the main characteristics of the CAN controller [5].

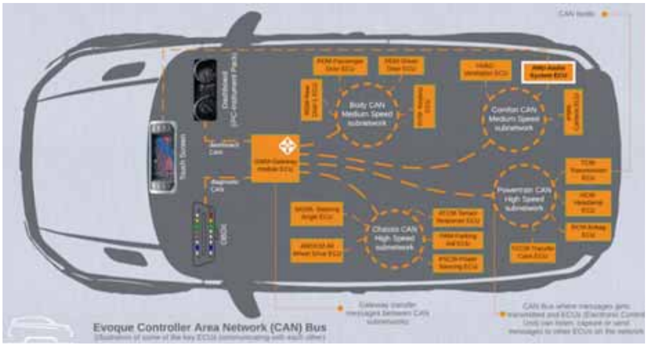

Figure 1. Car control units connected via CAN bus

CAN protocol

The overall structure of the CAN bus is based on the physical layer and data link layer and it follows the standard OSI model [6]. There are 3 versions of the CAN bus, classical CAN, CAN FD (Flexible Data Rate), and CAN XL (eXtra Large). The newest version of the CAN bus is the CAN XL and it is under development. CSMA/CD (Carrier Sense Multiple Access with Collision Detection) with the AMP technique (supporting Arbitration on Message Priority) is an arbitration method on which the CAN bus is based [7]. The maximum signal rate of the CAN bus is 1 Mbps and it is a multi-master message broadcast system [8].

CSMA means that each node on the bus must wait a prescribed idle time before attempting to send a message. CD+AMP means that collisions are resolved through intelligent bit definition and are based on the pre-programmed priority of each message in the message identifier field. The identifier with the highest priority always accesses the bus first. It is the last logical 1 in the identifier contained in the transmission that is at the highest priority. Each node on the bus participates in writing each bit according to the "as

1) University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, SERBIA
2) Toplica Academy of Vocational Studies – Department of Agricultural and Food Studies, Ćirila i Metodija 1, 18400 Prokuplje, SERBIA
Correspondence to: Đorđe Lazarević., e-mail: djordje.lazarevic@elfak.ni.ac.rs

written" principle, and the deciding node recognizes if there is a logical 1 bit on the bus. The standard with an 11-bit identifier enables speeds from 125 kb/s to 1 Mb/s. The standard was later supplemented with an extended 29-bit identifier. A standard 11-bit identifier provides 211 or 2048 different message identifiers while a 29-bit identifier provides 229 or 537000000 different identifiers. Fig.2 shows the CAN standard with an 11-bit identifier, while Fig.3 shows the extended CAN standard with a 29-bit identifier.

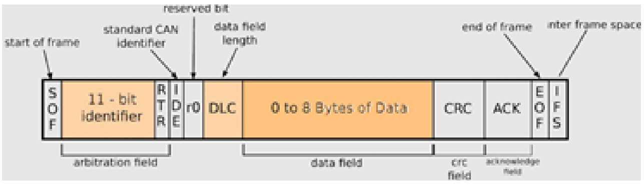

Figure **2.** CAN standard with an 11-bit identifier

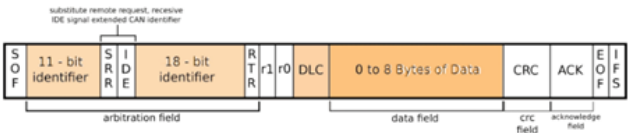

Figure 3. Extended CAN standard with a 29-bit identifier

A fundamental feature of CAN is the inverse logic state on the bus, of both the driver input and the receiver output. Under normal circumstances, logic 1 is associated with logic 1, and logic 0 is associated with logic 0, but this is not the case on the CAN bus.

The reason why CAN transceivers have the driver input and receiver output pins passively connected to a logic 1 level is that in the absence of any input, the component automatically resets the state of all input and output pins to the default value. Fig.4 shows the inverse logic of the CAN bus.

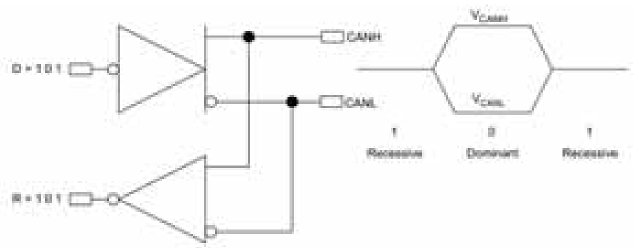

Fig.4. Inverse CAN bus logic

The prioritization of messages in the identifier is an advantage of CAN that makes it suitable for use in real-time control environments for real-time operation. The lower the binary message identifier number, the higher the priority. An all-zero identifier is the highest message priority in the network because it has been dominant on the bus for the longest time. In the event that two nodes send data at the same time, the dominant node will be the one with the last bit of the identifier equal to 0. The other nodes send 1 and they are recessive.

A node with an identifier with a trailing bit of 0 retains control of the CAN bus and completes its message. A dominant bit always overrides a recessive bit on the CAN bus. The sending node also monitors every bit of its transmission. This is why the driver's CANH and CANL output pins are connected internally to the receiver's inputs. The signal propagation delay in the internal loop from the driver input to the receiver output is typically used to qualitatively measure CAN transceivers. This propagation delay refers to the loop time (tLOOP).

The CAN arbitration process is managed automatically by the CAN controller. Because each node continuously monitors its own transmission, if node B's recessive bit is overridden by node C's dominant bit, B detects that the bus state does not match the bit that was sent. With that, node B stops transmitting while node C continues sending its message without interruption. Another attempt to send the message is made by node B once the bus is released by node C. This functionality is contained entirely in the CAN controller and fully available to the CAN user. Fig.5 shows arbitration on the CAN bus.

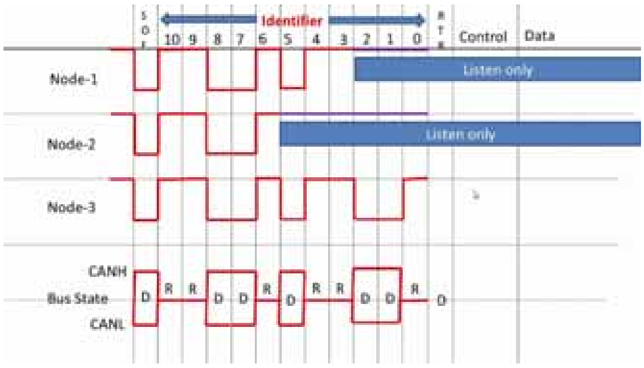

Figure 5. Arbitration on the CAN bus

The data link and physical signaling layers shown in Fig.6 are normally visible to the system operator and are included in many controllers that implement the CAN protocol. Fig.6 shows the method of connecting several nodes for communication via the CAN bus.

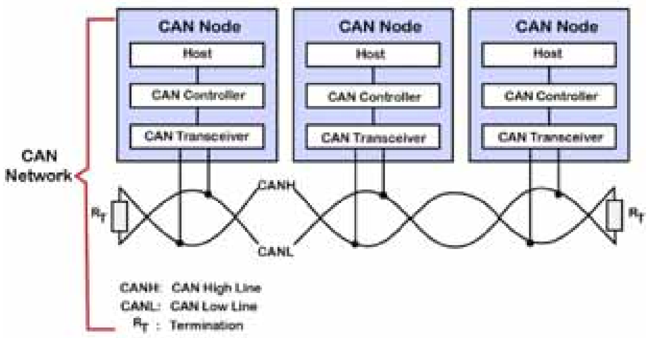

Figure 6. Connecting multiple nodes to the CAN data bus

The signals on the bus are differential, which is why the CAN bus has high immunity to interference and error tolerance. Balanced differential signals on both lines reduce noise superimposition and allow high signal speeds over twisted pair (twisted pair cable). Balancing with differential signals means that the current of each signal, that is both signal lines, is identical only in the opposite direction, and thus interference is automatically canceled, which is the key to low interference emission. Common signal rejection and high immunity to interference on the CAN bus can be improved by using balanced differential receivers and twisted pair cables.

The two signal lines on the bus, CANH and CANL in the quiescent recessive state are passively coupled to 2.5 V. The dominant state on the bus sets CANH~1V to 3.5 V, and sets CANL~1V to 1.5 V, making typically 2 V difference signals [9].

The difference between CANH and CANL signals as well as the signals themselves are shown in Fig.7.

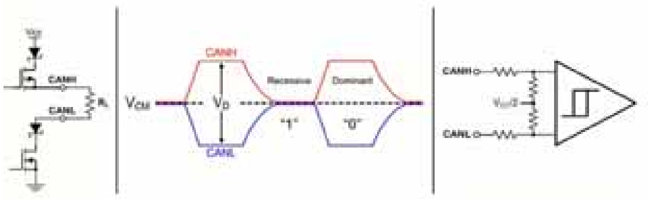

Figure 7. CANH and CANL line – recessive and dominant condition

Fig.8 shows the mutual communication of the nodes.

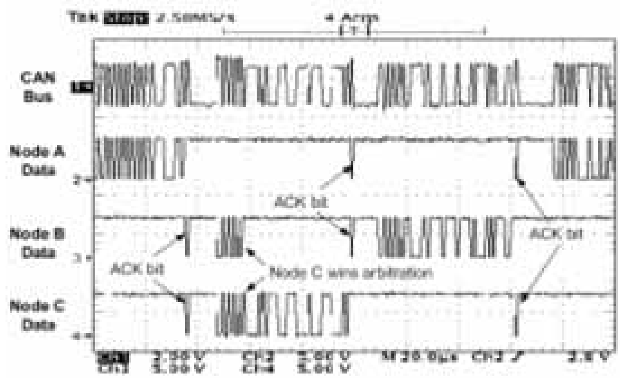

Figure 8. Mutual communication of nodes on the CAN bus

Practical implementation of the system

The electrical diagram of the CAN communication protocol system was designed in the Altium Designer. For all nodes the appropriate software for microcontrollers in different development environments is implemented, which provides full functionality of modules used for CAN communication between nodes. The system consists of 4 CAN nodes, where nodes 1, 2, and 3 measure and send data via the CAN protocol. Node 4 represents a monitoring node or test node that monitors events on the bus and displays received values on an alphanumeric LCD. In addition to the CAN interface, this device has a serial RS232 interface. This interface can be used for connecting the monitoring node with a PC where data from the sensor can be archived [10].

The microcontroller system representing node 1, in this case, measures the temperature and air humidity using the DHT11 [11] sensor, while the data from the sensor is received via the ATMega328P [12] microcontroller on the Arduino Nano development board and further processing and sending of data is finished via the MCP2515 CAN controller [13] and the TJA1050 transceiver [14].

Node 2 represents a system with 32-bit ARM microcontroller STM32F429ZIT6 [15] on the STM32F4 Discovery development board that measures ambient lighting using the BHT1750 [16] sensor and further sends data to the CAN bus via MCP2515 and TJA1050.

Node 3 consists of a 32-bit microcontroller ESP32-WROOM-32 [17] with Tensilica dual-core processor, TJA1050 transceiver, and BMP180 air pressure sensor. ESP32 microcontroller reads the value of pressure from BMP180 [18] sensor over the I2C protocol and sends the data through TJA1050 transceiver on the CAN BUS

The test node controller is an ATMega328P microcontroller board which is packed in a box with an LCD screen on the top of the package. For CAN communication a DB9 connector is used, which is also used for RS232 serial communication. This node is powered with an industrial power supply, while a DC-DC regulator is implemented on the board, which regulates the voltage to 5 V and enables the power supply of all parts of the system. All data received from the CAN bus is processed and all nodes connected to the CAN bus are monitored by the test node and values of the data are represented on an LCD screen.

The analysis of the CAN protocol operation, i.e. the signal from the Tx and Rx pins of the TJA1050 integrated circuit, was performed with a logic analyzer in the form of a miniature acquisition card. Physical signals on the CANH and CANL lines were analyzed using a digital oscilloscope.

The system for communication of the microcontroller with the peripherals using the CAN protocol consists of 4 nodes. Node 1 and node 4 are systems based on an 8-bit microcontroller ATMega328p, while node 2 and node 3 are based on 32-bit microcontrollers.

A block diagram of connecting nodes is shown in Fig.9.

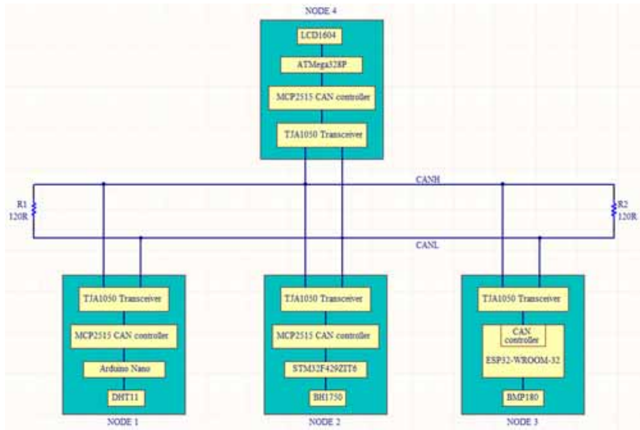

Figure 9. Block diagram of CAN nodes connection

Experimental setup and results

For implementing a system prototype that is used to analyze and display the operation of the CAN communication protocol the breadboards with the appropriate development boards with microcontrollers are used. To connect the prototype boards and to test the electronic system, the insulated twisted-pair cables were used. Recording and testing of the signals on the Tx and Rx communication lines on CAN transceivers was performed by using an 8-channel logic analyzer. Analyzing the analog voltage values is done by using a digital oscilloscope in the form of an acquisition card with probes connected directly to the outputs of the TJA1050 circuit on the CANH and CANL lines. The output voltages on the CANH and CANL lines are in the range of up to 5V, specifically 3.5V on the CANH output and 1.5V on the CANL output, for the dominant bit. These signals are differential and much more immune to interference than a typical serial transmission via the RS232 interface. For analyzing signals on the CAN bus using oscilloscope and logic analyzer a software installed on the PC is used. The CAN communication speed is set to the value of 500kb/s. The identifiers of the messages sent on the CAN bus are 0x037 from node 1, which sends the temperature and air humidity values. The identifier of the messages from node 2 is 0x0A0, which sends the ambient lighting value directly to the lux unit. Node 3 sends the data of the air pressure in pA and it has the message identifier 0x001 for ESP32. The test node for serving the receiving and presenting received data from other nodes is node 4 and it has a 0x030 identifier.

The priority of sending a message on the bus is determined after arbitration and the node with the identifier 0x001 always sends first because the node with 0x030 mainly serves as a receiver, although it will always have priority if it is necessary to send a control frame or message. Fig.10 shows the connection of the nodes in the laboratory.

Sending the messages on the CAN bus starts by pulling the Tx line to LOW, during which arbitration and determination of the dominant and recessive nodes on the network occur. Given that in this case we have three nodes that send data on the bus, the one with the lower binary identifier of the message always gets the dominant state. Fig.11 shows the message identifier 0x001, while Fig.12 shows the message with the identifier 0x037, i.e. the message sent by node 1.

Figure 10. Connection of the nodes in the laboratory

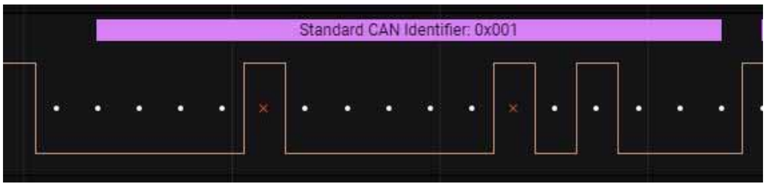

Figure 11. Message identifier 0x001

Figure 12. Message identifier 0x037

In arbitration, node 1 went into a dominant state and was the first to send a message on the bus, while nodes 2 and 3 became recessive. After the finished sending of the message from the node 1, a message is sent from the node 2. With the lowest value of identifier the node 3 sends the message on the data bus the last. Identical as with the node 1, sending starts with the SOF bit, i.e. the change of the logic state of the bus from HIGH to LOW.

Fig.13 presents data frames from the connected nodes on the CAN bus that sends data.

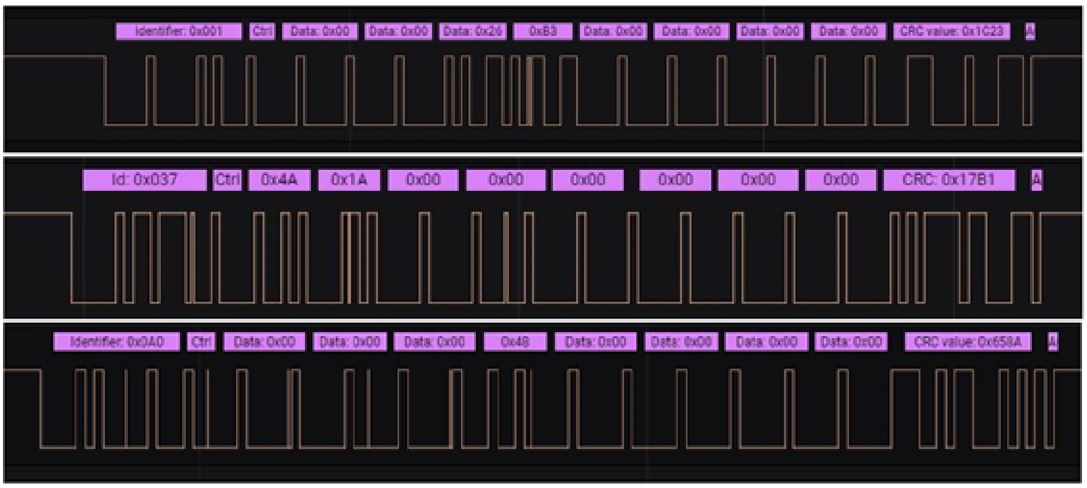

Figure 13. Data frames from the connected nodes on the CAN bus

The node with identifier 0x001 sends the data with values of atmospheric pressure, the node with identifier 0x037 sends the data with values of temperature and humidity and the node with identifier 0x0A0 sends the data with value of illuminance.

To confirm the operation of CAN communication, a test node with a LCD1604 display is used. It receives data and is used to display values from sensors sent by all nodes on the CAN bus. Information is received in accordance with the sending rules on the CAN bus, where data is sent depending on the value of the binary identifier. This testing node 4 confirmed that communication is successful and all data are properly sent to the CAN bus without losing any data. Fig.14 shows the received values from the sensors on the LCD screen on a testing node.

Figure 14. The values from the sensors on the testing node's LCD screen

A digital oscilloscope is used to present voltage values at the output of transceiver modules. These voltage signals have values of 3.5V on the H line and 1.5V on the L line. For the nodes the voltage regulator is used which has an output of 5V so the CAN bus line is working with this voltage level. The signal is differential and provides the elimination of noises that may be present in other forms of serial communication. Fig.15 shows differential signal on the CAN bus.

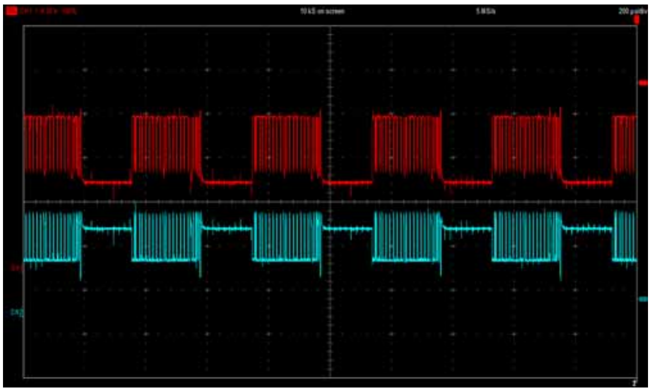

Figure 15. The differential signal on the CAN bus

Conclusion

Thanks to its robustness the application of the CAN protocol is very large. CAN communication is widely used in medical instrumentation and industrial electronics. Also, the CAN protocol can be used wherever a communication protocol is needed, which allows a reduction in the level of interference, robustness, high speed, and flexibility. This paper presents the operation of CAN communication and successful sending and receiving of data. It is possible to further improve the system and connect it to other types of wired interfaces or with wireless interfaces which is also the case in real systems.

Acknowledgment

This research was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References:

[1] HUAN,L., QINZHU,W.: *EMC improvement of CAN bus in automotive bus*, Journal of Physics: Conference Series, Vol. 1607, 2020, pp. 1-6

[2] LAZAREVIĆ,Đ., ŽIVKOVIĆ,M., KOCIĆ,Đ., ĆIRIĆ,J.: *CAN komunikacioni protokol*, IEEESTEC - 15th Student Projects Conference, 2022, pp. 13-17

[3] SHI,D., KOU,L., HUO,C., WU,T.: *A CAN Bus Security Testbed Framework for Automotive*

[4] *Cyber-Physical Systems*, Hindawi, Wireless Communications and Mobile Computing, Volume 2022, pp. 1-11.

[5] SPENCER,G., MATEUS,F., TORRES,P., DIONISIO,R., MARTINS,R.: *Design of CAN Bus Communication Interfaces for Forestry Machines*, MDPI, Computers, 2021, Vol.10, pp. 144-159.

[6] MENGAN,L., BING,H., LIYOU,X., YANYING,L.: *CAN bus network design of bifurcated power electric tractor*, Peer-to-Peer Networking and applications, 2020, Vol.14, pp. 2306-2315.

[7] LIU,C., WANG,X., TIAN,L., ZHANG,Z.: *Fuzzy Testing Method of CAN Bus of Charging Pile Based on Genetic Algorithm,* Security and Communication Networks, Volume 2022, pp. 1-11

[8] ALI,Z., ALI,Q.I.: *A CAN Bus-Based Efficient Design for a Simple Autonomous Vehicle (AV)*, 1st International Conference of Scientific and Innovative Studies, 2023, pp. 377-386.

[9] YONG,S., MA,Y., ZHAO,Y., QI,L.: *Analysis of the Influence of CAN Bus Structure on Communication Performance*, 5th EAI International Conference, IoTaaS, 2019, pp. 405-416.

[10] Introduction to the Controller Area Network. Dostupno na: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1592815338549&ref_url=https%253A%252.

[11] LAZAREVIĆ, Đ., ŽIVKOVIĆ, M., KOCIĆ, Đ., ĆIRIĆ, J.: *The Utilizing Hall Effect-Based Current Sensor ACS712 for TRUE RMS Current Measurement in Power Electronic Systems*, Scientific Technical Review, 2022, Vol.72, No.1, pp. 27-32.

[12] DHT11 datasheet. Available at: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.

[13] Microcontroller ATMega328P, datasheet. Available at: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.

[14] MCP 2515, Stand-Alone CAN Controller with SPI Interface. Available at: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf.

[15] TJA1050 datasheet. Available at: https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf.

[16] STM32F429ZIT6 microcontroller datasheet. Available at: https://www.st.com/resource/en/datasheet/dm00071990.pdf.

[17] BHT1750 datasheet. Available at: https://www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf.

[18] ESP32-WROOM-32 module, datasheet. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[19] BMP180 datasheet. Available at: https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf.

Received: 30.10.2023.
Accepted: 15.12.2023.

Analiza rada CAN komunikacije korišćenjem različitih vrsta senzora

U ovom radu je prikazan rad CAN komunikacionog protokola odnosno komunikacije između čvorova korišćenjem različitih tipova senzora. Sistem se sastoji od više čvorova sa mikrokontrolerima različitih arhitektura. Jedan čvor je kontrolni čvor dok drugi čvorovi šalju vrednosti očitavanja sa senzora. Sistem je implementiran i testiran u laboratoriji. Za analizu signala se koriste se osciloskop i logički analizator. Takođe su prikazani dobijeni rezultati i dijagrami signala na magistrali.

Ključne reči: **CAN, mikrokontroler, mrežni analizator.**