[1.]Sinisa KUZMANOVIC, [2.]Zlatko BUNDALO, [3.]Dušanka BUNDALO, [4.]Boris NEDIC

APPLICATION OF DRIVE-BY-WIRE CONCEPT AND CAN PROTOCOL IN AUTOMOBILES

[1.] M:tel a.d., Kneza Lazara 4, 74000 Doboj, BOSNIA & HERZEGOVINA
[2.] University of Banja Luka, Faculty of Electrical Engineering, Patre 5, 78000 Banja Luka, BOSNIA & HERZEGOVINA
[3.] University of Banja Luka, Faculty of Philosophy, Petra Bojovica 1A, 78000 Banja Luka, BOSNIA & HERZEGOVINA
[4.] EIB Internationale a.d., Skendera Kulenovica 14, 78000 Banja Luka, BOSNIA & HERZEGOVINA

Abstract: Possibilities and ways of application of drive-by-wire concept and CAN protocol in automobiles are considered, described and proposed in the paper. Modern automobile is a system with many embedded microcontroller systems that perform and control needed operations in the automobile. For communication between that microcontroller systems it is used CAN protocol. Also, drive-by-wire concept for steering and speed control of automobile is very easy, promising and constantly developing. Basic characteristics and advantages of using drive-by-wire concept and CAN protocol in automobile are described first. Then, the way of developing and practical implementation of one drive-by-wire system in automobile using CAN protocol and microcontrollers is described. For the implementation were used two microcontroller controlled CAN nodes based on microcontroller development boards. One CAN node performs steering and speed control functions and generates appropriate control data. The data are transferred by CAN protocol to the second CAN node. The second CAN node receives and processes the data and performs functions for control of automobile speed and movement direction. Hardware and software implementations of the solution are described in more details.

Keywords: drive-by-wire concept, CAN protocol, microcontroller systems

INTRODUCTION

Since the first modern automobile was built and the automobile production industry became one of the most profitable branches of industry, the automobile did not basically change much. Due to the constant demands for environment protection, there were developed electric and hybrid automobiles. The biggest changes in automobiles are result of rapid development and use of electronics. Microcontrollers are used for the control of many systems in automobiles. It was achieved a good combination of electronic and computer technologies that are applied together in many automobile subsystems. There is large amount of complex electronic devices, over 70 electronic control units (ECUs), several hundred sensors, etc. that are incorporated into the modern automobile. During driving automobile, electronic systems in vehicle continuously monitor more than a hundred different parameters from sensors and adequately control the automobile [1-3]. Every few milliseconds are read data from numerous sensors distributed in the vehicle and then are controlled certain elements of automobile. The function of electronic elements is to increase the overall performance of the vehicle. Electronic components in automobile are interconnected electronic subsystems, with their own central processor unit, working and permanent memory, exchanging data using communication protocols specially developed for automobiles. Modern automobile is a system with many networked embedded microcontroller systems that monitor, perform and control needed operations in the automobile [1-3]. The most significant improvements that are such achieved are: increasing driving safety, reducing fuel consumption, reducing environmental pollution, increasing comfort, improving diagnostic functions [1-3].

For communication between microcontroller based systems in automobile it is used CAN (Control Area Network) protocol [1-4]. The CAN is the most widely used network and protocol in vehicles. The CAN using a twisted pair of copper wires became an ISO standard and is standard in Europe for data transmission in automotive applications, due to its low cost, robustness, and bounded communication delays [1-4].

Drive-by-wire concept for steering and speed control of automobile is very efficient and easy for use and constantly developing [1-6]. That concept integrates and implements the function of steering the direction of movement of car and function of control of speed of movement of car using electronic elements controlled by using simple stick (so called joystick) instead of using classic steering wheel.

Possibilities of application of drive-by-wire concept in automobiles, using CAN protocol, are considered, proposed and described in the paper. Basic characteristics and advantages of using drive-by-wire concept and CAN protocol in automobile are described. The way of developing and practical implementation of one drive-by-wire system in automobile using CAN protocol and microcontrollers are described. For the implementation were used two microcontroller CAN nodes based on microcontroller development boards. One CAN node performs steering and speed control functions using joystick and generates appropriate control data that are transferred by CAN protocol to the second CAN node. The second CAN node receives and processes the data and performs needed functions for control of automobile speed and movement direction. The proposed and implemented solution is mainly intended for electric automobiles. It can be used also for the classical automobiles. The solution also shows the possibility to use standard microcontrollers and standard open source microcontroller based platforms for the implementation of such systems.

DRIVE-BY-WIRE CONCEPT

The drive-by-wire concept in automobiles is similar and it was created on the basis of the so-called fly-by-wire system used in aviation and avionics systems [1-6]. It is a system that replaces the classical manual mechanical steering wheel and movement control system of automobile with an electronic interface and enables electronic control of vehicle movement. The movements of the vehicle steering mechanism are converted into electric signal transmitted through the wire to the vehicle control elements. Microcontrollers for control of the movement of vehicles, based on such obtained information, determine how to control and run actuators in all vehicle control parts. In practice there are existed and used such two systems and two terms, steer-by-wire and drive-by-wire, that are often separated. But, that systems are mainly integrated into one system known under the common name drive-by-wire (or X-by-wire) [1-6]. That way of the integrated approach was also used in the solution designed, implemented and described in this paper.

Drive-by-wire (or X-by-wire) is term used when mechanical and/or hydraulic systems in vehicle are replaced by electronic ones. Used electronic systems are with intelligent microcontroller devices, networks, microcontrollers supporting software components that implement filtering, control, diagnosis and other needed functions. Drive-by-wire technology replaces traditional mechanical and hydraulic actuators and intermediate components with electronic and electric components that are interconnected only by wires, used as informational and power lines.

Drive-by-wire technologies enhance comfort, enhance control and improve vehicle performances, since they introduce completely electronic control of actuators. The main purpose of drive-by-wire systems is to assist the driver in different situations in a more flexible way. Purpose is also to decrease production and maintenance cost for vehicle braking and steering systems. Vehicles equipped with drive-by-wire system have kept traditional mechanical technologies as a backup in case the electronic ones fail. Completely drive-by-wire system has to have at least the same level of safety as traditional mechanical systems, using redundancy, replication, functional determinism and fault tolerance concepts [1-6].

The control system of automobile is a set of elements that allows the driver to determine the path and the speed of vehicle movement [1-3]. That system provides precise control of the direction of movement of the vehicle front wheels and the speed of rotation of the vehicle driving wheels. It also reduces the amount of driver efforts needed to direct the front wheels and to change the speed of the driving wheels rotation.

The control is performed very easily using appropriate stick (so called joystick). By moving the joystick to the left or to the right changes the direction of movement of the vehicle. By moving the joystick forward or backward changes the speed of the vehicle movement. The joystick controls two potentiometers by what are changed the voltages that are applied to the control system. The values of those voltages define the speed of change of the direction and speed of the vehicle. Moving the stick to the left from the initial position causes the movement of the vehicle to the left side. Moving the stick to the right from the initial position causes the movement of the vehicle to the right side.

Also, moving the stick forward from the initial position causes increase of the speed of the vehicle movement. Moving the stick backwards from the initial position causes reduction of the speed of the vehicles movement. That principle is also used in the practical solution designed, implemented and described in this paper.

Influence of interferences is one of problems in drive-by-wire systems. Therefore, it was developed drive-by-wire technology that is resistant to the occurrence of malfunctions [1-6]. Thus, drive-by-wire systems of high integrity are implemented in a very economical way, achieving required levels of security, reliability and availability. It is used multiple redundancy in design of secure drive-by-wire systems. Thus, it is achieved full control of the driver over the vehicle in case of partial malfunction of the system.

CAN protocol

The CAN communication protocol is used for connection and communication of microcontroller systems in automobiles [1-4]. It was designed by company Bosch for communication between ECUs in vehicles and for decreasing length and number of dedicated wires for interconnection. In modern automobiles CAN is mainly used as network for real-time control in power train and chassis domains. It is also used as network for the electronics in the body domain of vehicle.

Before application of the CAN protocol electronic components in automobiles were interconnected directly using wires. This type of interconnection is complicated and complexity of the network increases with the increase of number of components. Such networks also significantly reduce reliability in communication between components and reliability of vehicle itself. Also, the way of wiring of automobile has effect on vehicle weight and on occupation of vehicle space.

Figure 1 shows the CAN bus and way of connection of electronic components (CAN nodes) to the CAN bus [1-4].

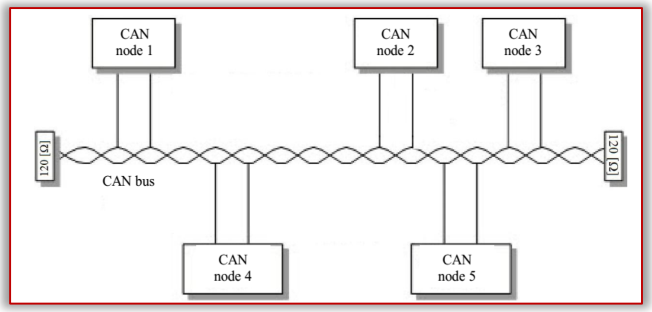


Figure 1. Connection of electronic components to CAN bus

The CAN bus has two wires called CAN_H and CAN_L that are terminated using 120 ohm resistors. This pair of wires is twisted what reduces electromagnetic interference. The CAN is serial communication protocol accepted by the Society of Automotive Engineers (SAE). It is called Automotive Serial Controller Area Network and was first developed for the automotive industry, for connecting electronic systems in cars. The CAN protocol was later applied in many areas, primarily in industrial plants, airline industry and medicine. CAN bus is primarily used in embedded systems and it is network established between microcontrollers. The CAN bus is two-wire network system suitable for fast applications that use short messages.

The CAN bus connects all the CAN nodes. Figure 2 shows a typical structure of the CAN node [1-4].

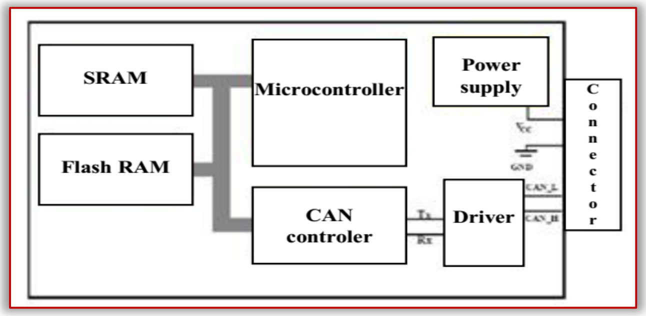


Figure 2. Structure of CAN node

The CAN node uses microcontroller that performs needed digital data processing. The shown CAN node (Figure 2) is implemented using specialized CAN controller realizing the connection of the node to the network and exchange of information with microcontroller. This type of CAN controller is so called stand-alone controller.

The CAN protocol provides communication speeds up to 1 Mbit/s, enabling real-time control and real-time applications. Transmission speed and distance between CAN nodes are inversely proportional, as is shown in Table 1 [1-4].

Table 1. Transmission speed and length of CAN bus

Transmission speed	Length of CAN bus
1 Mbit/s	25 m
800 kbit/s	50 m
500 kbit/s	100 m
250 kbit/s	250 m
125 kbit/s	500 m
50 kbit/s	1000 m
20 kbit/s	2500 m
10 kbit/s	5000 m

The CAN bus is widely used in power train communication of automobile, in the chassis domain (where is used high-speed HS-CAN at 500 kbit/s) and in the body domain (where is used low-speed LS-CAN at 125 kbit/s) [1-4].

Data is sent and received in the CAN system using message frames. The CAN is based on principle that every node in the system receives message from the bus. Hardware within each CAN node performs local filtering that allows only certain nodes to respond to the received message. The CAN system uses four types of messages:

- Data Frame,
- Remote Frame,
- Error Frame,
- Overload Frame.

The CAN system uses data frame to transmit data over network. The data frame consists of identification and different control information. It can contain maximum of eight bytes of data. CAN system uses two types of data frames, the basic frame and the extended frame. When CAN system was developed for large systems with dense traffic of messages, the basic frame format was insufficient. The number of messages created by the transmitter was greater than the number of possible ID codes that CAN system could assign to them to ensure that each message is unique. By adding and using larger 29-bit ID field, the CAN system can create about 512 million different messages and priorities. Figure 3. shows the basic format of the data frame [1-4].

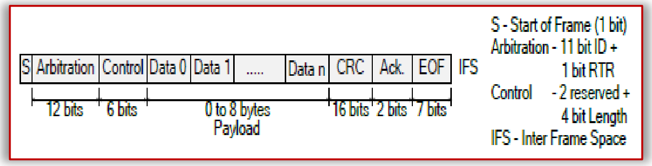


Figure 3. Basic format of CAN data frame

The data frame consists of the following sections [1-4]:

- Start of Frame - Single dominant bit (logical 0), which marks the start of Data Frame or Remote Frame. Node can only start transmission when it detects that the bus is idle. It is required that all nodes synchronise to the leading edge of the Start of Frame bit.
- Arbitration field - 11-bit node Identifier (which also defines its priority) and remote transmission request (RTR) bit (dominant for Data Frame and recessive for Remote Frame).
- Control field - 4-bit Data Length and 2 reserved bits.
- Data field - Up to 8 bytes of information (including the option for zero byte data).
- CRC field - 15-bit cyclic redundancy check (CRC) code followed by single recessive bit as the CRC delimiter.
- Acknowledgement field - 1-bit Ack slot followed by a 1-bit Ack delimiter. Any node that validates the transmitted data will override the recessive bit sent in the Ack Slot by the transmitter to indicate that the frame has been transmitted correctly. However, since CAN is a broadcast

network, the node which acknowledges the receipt of data may not be the intended recipient.

- End-of-Frame field (EOF) - Sequence of seven recessive bits followed by inter-frame space.

The CAN specification also defines special frame formats like the Error Frame and Overload Frame.

The extended format of the data frame is almost identical to the basic format of the data frame. The only difference between these two formats is in the bit of extension of identifier and the size and arrangement of the arbitration field. Both of these formats can coexist in the same CAN system. The basic format of the frame has always priority in relation to the extended format of the frame.

The CAN system is very effective in detection of errors. Each frame is simultaneously accepted or rejected by each node in the network. If the node detects an error, it transmits field of error to each node in the network and destroys the transferred frame [1-4].

The CAN protocol uses the following methods to detect errors:

- Check of bits,
- Check of frame,
- Cyclic redundancy check (CRC),
- Acknowledgement check,
- Check of filling rules.

The main risk in any system with serial bus is that one faulty node can turn off the entire network. To deal with this, CAN protocol is designed to automatically detect faulty nodes and disconnect them from the network. CAN uses two error counters for each node, one for errors in transmission and the other for errors in reception. When error in sending or receiving occurs, the corresponding counter is increased for an appropriate value. That value depends on the error type and which node has caused the error. For each successful sending or receiving, the corresponding counter is decreased for one. Typically, if the node recognizes that it is the source of the error then the counter increments by nine for the received errors and for eight for the transmitted errors. Otherwise, each node counter is increased by one. Based on these counters, each node can be in one of three states [1-4]:

    - Active Error,
    - Passive Error,
    - Bus Off.

The remainder of the network will continue to operate and if the node is set to bus off state. Sometimes errors occur due to external factors, for example, if the network is exposed to occasional electromagnetic interference. The CAN system is therefore designed that counters of errors in nodes can be decreased if data frames are successfully received and sent.

IMPLEMENTATION OF DRIVE-BY-WIRE CONCEPT IN AUTOMOBILE USING CAN protocol

One practical proposal, design and implementation of a simplified drive-by-wire system in automobile using CAN protocol is described here. Two microcontroller based CAN nodes were used and implemented. One CAD node reads the information about changes of the voltage at the A/D converter input, generated by the potentiometers of the joystick. This information is sent via the CAN bus to the second CAN node. The second CAN node processes and sends this information to the elements for control of direction and to the elements for control of speed of the car movement. This information is also displayed on the LCD display in the second CAN node. As the hardware basis for the design and implementation, the standard development boards Easy8051A (marked as Board 2) and Easy8051B (marked as Board 1) were used [7, 8]. The programs for both CAN nodes were also practically designed and implemented. Figure 4 shows simplified hardware block scheme of the implemented drive-by-wire system.

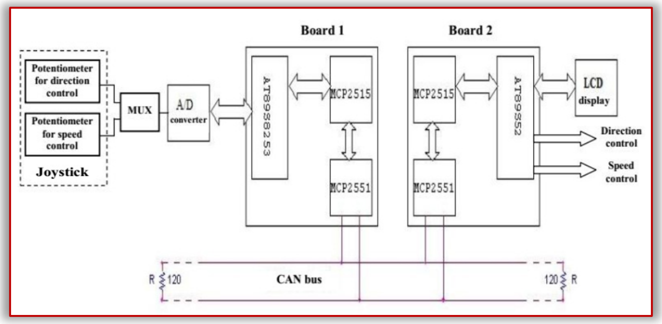


Figure 4. Simplified block scheme of implemented drive-by-wire system

The development environment model used to connect and accomplish the functionality of all the necessary components of the system is Easy8051 [7, 8]. The Easy8051 contains the following components:

- Power supply (8 V to 16 V AC/DC),
- ON/OFF switch,
- USB connector,
- USB 2.0 programmer,
- Socket for placing an 8051 microcontroller,
- Connectors of input/output ports of the microcontroller,
- Selection of pull-up/pull-down resistors,
- Keys for the input of the microcontroller input pins,
- Selection of the logical level that is applied on the keys,
- Reset button for microcontroller,
- LED for displaying the logic level on the pins,
- Using the first four switches of the SW2 group, the LEDs on the ports P0, P1, P2, P3 and P4 are connected or disconnected to the MCU, and the remaining four switches select the digits on the 7-segment display,
- Connector for alphanumeric LCD display,
- Adjustment of the contrast of the alphanumeric LCD display,
- Connector of graphic 128x64 LCD display,
- Adjustment of the contrast of the graphic 128x64 LCD display,
- Temperature sensor DS1820,

- Connector for RS-232 communication,
- Clock signal generator,
- 7-segment display,
- Switch group SW1 enables MOSI, MISO and SCK pin for SPI communication. It also enables A/D and D/A converter, RX/TX line for RS-232 communication and LCD/GLCD backlighting,
- 12-bit A/D converter,
- 12-bit D/A converter,
- Source of referent voltage of 4.096V.

The development environment is powered by a USB port. The program is downloaded via a USB programmer, and USB cable is used for connection to the personal computer. In order to load the program it is necessary to have the appropriate software installed on the computer (8051Flash Programmer and necessary drivers). Switches are used for establishing or terminating contacts with peripheral units. The switches on the Easy8051B board (Board 1 in Figure 4) in the SW1 group are set to enable the use of A/D converter and to enable SPI communication. On the Easy8051A board (Board 2 in Figure 4) switches from group SW1 are set so that SPI communication is enabled. Also, on the Easy8051B board using the J10 jumper was selected that the reference voltage has value of Vcc=5 V, and using the J9 jumpers, was select the CH0 output.

The communication between the Easy8051B board (Board 1), with the AT89S8253 microcontroller, and the Easy8051A board (Board 2), with the AT89S52 microcontroller, is performed using the CANSPI module [8]. On the Easy8051A board the frequency of the oscillator is 12MHz, and CANSPI module with a frequency of 16MHz is installed on it [9]. On the Easy8051B board the frequency of the oscillator is 24MHz, and a CANSPI module with a frequency of 8MHz is installed on it.

The voltage changes on the joystick potentiometers, caused by movement of the joystick, are read at the output CH0 of the A/D converter and transmitted via the CAN bus from the Easy8051B board (Board 1) to the Easy8051A board (Board 2). In the Board 2 the read values are sent to the elements for control of direction and to the elements for control of speed of vehicle movement and are displayed on the LCD display. It causes change in the direction and in the speed of the vehicle movement.

In the implementation it was used the MCP2515 stand-alone CAN controller with SPI Interface [10]. It is able to receive and send messages created in both standards, messages created in the basic and extended data frame and remote frame. The MCP2515 has two masks and six filters that are used to filter out undesirable messages and thus reduces the load of the microcontroller. The MCP2515 is connected to a microcontroller via an industry standard called Serial Peripheral Interface (SPI). The MCP2515 is a stand-alone CAN controller developed to simplify access of the applications to the CAN bus. This CAN module manages all functions of receiving and transmitting messages to and from the CAN bus. The message is transmitted when it is first entered in the buffer and the control register. The transmission is initiated using a control register bit by the SPI interface or by enabling transmission pins. States and errors can be checked by reading the appropriate registers. All messages detected on the CAN bus are checked for the presence of errors and then are compared with user-defined filters to see if they need to be moved to one of the two receiving buffers. Writing and reading of all registers is achieved by using standard SPI read and write commands.

The implementation also uses the MCP2551 high-speed CAN transceiver device that operates as an interface between the CAN protocol controller and the physical CAN bus [11]. It provides differential transmission and reception for CAN controller. It operates at speeds up to 1Mb/s. Typically each node in the CAN system must have a device to convert a digital signal created in the CAN controller into a signal suitable for transmission via the CAN bus (differential output). It also provides a buffer between the CAN controller and the voltage interference that can be created on the CAN bus from an external source.

In the development of this system it was used the software tool called mikroC PRO for 8051 [12]. Programming was performed in programming language C, after which the compiler translates the program into a hexadecimal code that was used for programming the microcontroller. To write the program into the microcontroller it was used 8051Flash Programmer software [11]. Using the microC PRO for the 8051 compiler and the 8051Flash Programmer software, two programs were developed and implemented, one for the AT89S52 microcontroller (Board 2) and the other for the AT89S8253 microcontroller (Board 1).

The algorithm and flow chart of the program for the Board 1 (the Easy8051B board) is shown in Figure 5.

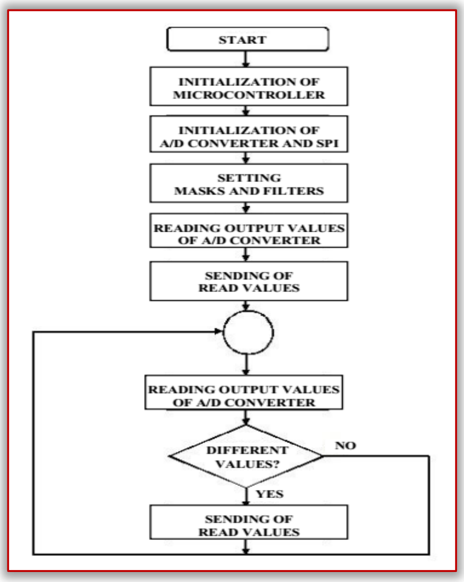
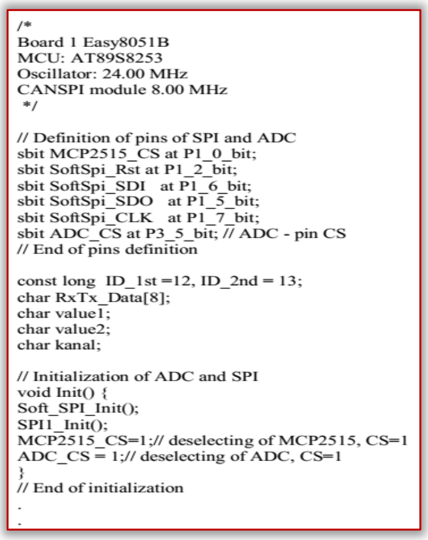


Figure 5. Flowchart of program for Board 1

As an illustration, the part of developed and implemented program for the Board 1 (the Easy8051B board) is shown in Figure 6.



```
/*
Board 1 Easy8051B
MCU: AT89S8253
Oscillator: 24.00 MHz
CANSPI module 8.00 MHz
 */

// Definition of pins of SPI and ADC
sbit MCP2515_CS at P1_0_bit;
sbit SoftSpi_Rst at P1_2_bit;
sbit SoftSpi_SDI   at P1_6_bit;
sbit SoftSpi_SDO   at P1_5_bit;
sbit SoftSpi_CLK   at P1_7_bit;
sbit ADC_CS at P3_5_bit; // ADC - pin CS
// End of pins definition

const long  ID_1st =12, ID_2nd = 13;
char RxTx_Data[8];
char value1;
char value2;
char kanal;

// Initialization of ADC and SPI
void Init() {
Soft_SPI_Init();
SPI1_Init();
MCP2515_CS=1;// deselecting of MCP2515, CS=1
ADC_CS = 1;// deselecting of ADC, CS=1
}
// End of initialization
.
.
```

Figure 6. Part of program for Board 1

The algorithm and flow chart of the program for the Board 2 (the Easy8051A board) is shown in Figure 7.

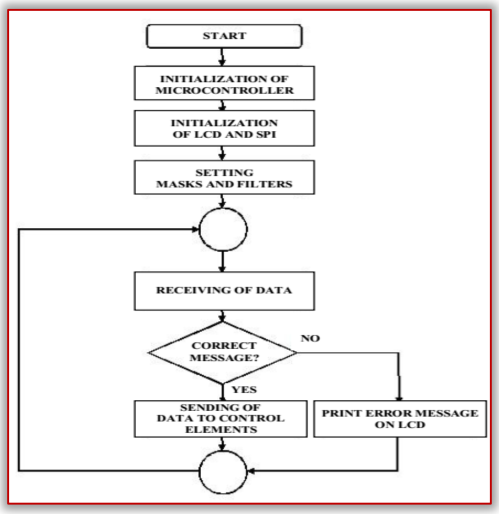
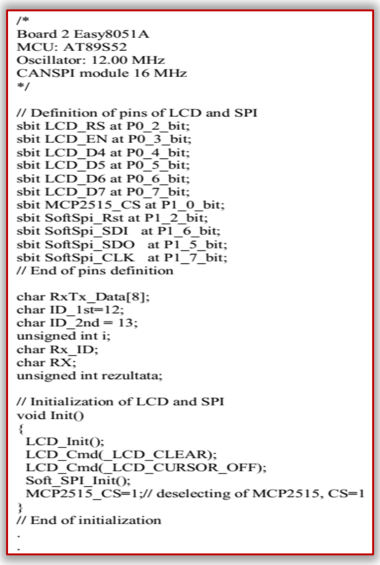


Figure 7. Flowchart of program for Board 2

The part of developed and implemented program for the Board 2 (the Easy8051A board) is shown in Figure 8.



```
/*
Board 2 Easy8051A
MCU: AT89S52
Oscillator: 12.00 MHz
CANSPI module 16 MHz
*/

// Definition of pins of LCD and SPI
sbit LCD_RS at P0_2_bit;
sbit LCD_EN at P0_3_bit;
sbit LCD_D4 at P0_4_bit;
sbit LCD_D5 at P0_5_bit;
sbit LCD_D6 at P0_6_bit;
sbit LCD_D7 at P0_7_bit;
sbit MCP2515_CS at P1_0_bit;
sbit SoftSpi_Rst at P1_2_bit;
sbit SoftSpi_SDI   at P1_6_bit;
sbit SoftSpi_SDO   at P1_5_bit;
sbit SoftSpi_CLK   at P1_7_bit;
// End of pins definition

char RxTx_Data[8];
char ID_1st=12;
char ID_2nd = 13;
unsigned int i;
char Rx_ID;
char RX;
unsigned int rezultata;

// Initialization of LCD and SPI
void Init()
{
 LCD_Init();
 LCD_Cmd(_LCD_CLEAR);
 LCD_Cmd(_LCD_CURSOR_OFF);
 Soft_SPI_Init();
 MCP2515_CS=1;// deselecting of MCP2515, CS=1
}
// End of initialization
.
.
```

Figure 8. Part of program for Board 2

In the development and implementation of the realized drive-by-wire system there were performed many tests of system operations. Figure 9 shows the voltage value that was read from the A/D converter and that was sent from the first CAN node (Board 1), received on the second control CAN node (Board 2) and displayed on the LCD display during testing of system operation.

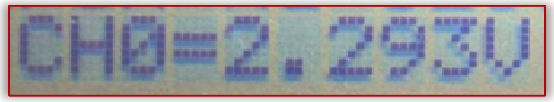


Figure 9. Voltage value read from A/D converter and displayed on LCD display

If the data information was not transferred via the CAN bus or if the data loss occurs, on the LCD display will be shown information ERROR!. It is shown in Figure 10. From the error state, the system exits when the data information is transferred correctly.

Figure 10. Information on LED display that transmission error occurred

CONCLUSIONS

Development and progress in the automotive industry is continual. The number of used embedded electronic components and digital microcomputers is constantly increasing, and new ways of networking are developed. This all increases the performance and safety of the automobile. An increasing number of existing automobile systems have been improved by the introduction of microprocessor control, and new systems are appearing that provide new possibilities. Without application of microcontrollers in cars it would not be possible to achieve all the requirements for performances and safety of the car. Also, it would not be possible to achieve of increasing and of more strict requirements for reducing environment pollution.

There are many possibilities and systems in the automobile where are used microcontrollers and CAN bus. Microcontrollers are used for monitoring and controlling of many systems in automobiles. Modern automobiles have many interconnected microcontroller systems that monitor, perform and control all needed operations. For networking and communication of microcontroller systems in automobile it is mostly used CAN protocol and CAN bus. The CAN bus uses twisted pair of copper wires and has very important advantages: low complexity, low cost, robustness and bounded communication delays. The CAN protocol has made a revolution in the automotive industry, enabling the increasing application of electronic systems in cars. Also, the price-quality ratio of the CAN is very good, what is very important factor for its massive usage.

Drive-by-wire control of movement direction and movement speed in automobiles is very easy and simple for use, and very efficient and inexpensive in implementation. That concept replaces mechanical and/or hydraulic systems in vehicle by electronic microcontroller based ones. The drive-by-wire system enables that the driving of the automobile be much easier and decreases production and maintenance cost for the vehicle.

The proposed and described practical solution shows basic good characteristics of the CAN system, such as simple and secure data transfer and simple realization of the bus. Messages are constantly filtered and checked for errors, so the number of wrongly transferred messages is reduced to a minimum. It is also shown the system reaction on the occurrence of an error caused by break of the bus connection or by disabling the transmission or reception of messages. In such a situation, the CAN system immediately notices the error and sends information to the central controlling system or to the user that a transmission error occurred and that appropriate corrective measures should be taken to correct error.

Practically developed, implemented and described drive-by-wire solution uses standard microcontrollers and CAN bus. The solution is simple and inexpensive. It shows all advantages and benefits of application and implementation of drive-by-wire concept, microcontrollers and CAN protocol in automobiles. All this was tested and verified through the practical realization of this simple drive-by-wire system.

The proposed and implemented system is mainly intended for control of electric automobiles. It can be also used for control of the classical vehicles. The implemented drive-by-wire solution also shows the possibilities and advantages of using inexpensive standard microcontrollers and standard open source microcontroller based platforms for the implementation of such systems for automobiles.

Note

This paper is based on the paper presented at INTERNATIONAL CONFERENCE ON APPLIED SCIENCES – ICAS 2018, organized by UNIVERSITY POLITEHNICA TIMISOARA, Faculty of Engineering Hunedoara (ROMANIA) and UNIVERSITY OF BANJA LUKA, Faculty of Mechanical Engineering (BOSNIA & HERZEGOVINA), in cooperation with the Academy of Romanian Scientists, Academy of Sciences Republic of Srpska, Academy of Technical Sciences of Romania – Timisoara Branch and General Association of Romanian Engineers – Hunedoara Branch, in Banja Luka, BOSNIA & HERZEGOVINA, 9 – 11 May 2018.

References

[1] Denton T 2012 Automobile electrical and electronic systems, Routledge, Taylor and Francis Group
[2] Navet N and Simonot-Lion F 2009 Automotive Embedded Systems Handbook, CRC Press, Taylor and Francis Group
[3] Shanker S 2016 Enhancing Automotive Embedded Systems with FPGAs, Nanyang Technological University, Singapore, Doctoral Thesis
[4] Di Natale M, Zeng H, Giusto P and Ghosal A 2012 Understanding and using the Controller Area Network communication protocol, Springer
[5] Bergmiller P 2015 Towards Functional Safety in Drive-by-Wire Vehicles, Springer
[6] Yih P Steer-By-Wire: Implications for vehicle handling and safety, http://www-cdr.stanford.edu/dynamic/bywire/defense_v5.pdf
[7] Easy8051A Users manual, MikroElektronika, http://www.mikroe.com/
[8] Easy8051B Users manual, MikroElektronika, http://www.mikroe.com/
[9] CANSPI Users manual, MikroElektronika, http://www.mikroe.com/
[10] MCP2515 Stand-Alone CAN Controller With SPI™ Interface, Microchip
[11] MCP2551 High-Speed CAN Transceiver, Microchip
[12] mikroC for 8051 User manual, MikroElektronika, http://www.mikroe.com/

ISSN: 2067-3809

Faculty of Engineering Hunedoara,
5, Revolutiei, 331128, Hunedoara, ROMANIA
http://acta.fih.upt.ro