

A Smart CAN-Bus Model for In-Vehicle Message Passing Channel

Prabira Kumar Sethy¹, Diptimayee Jena², Nalini Kanta Barpanda³, Gyana Ranjan Panigrahi⁴

Department of Electronics, Sambalpur University, Burla, India^{1,2,3}

Department of IMSc ETC, B.J.B (A) College, Bhubaneswar-751014

Email: prabirsethy.05@gmail.com¹, diptijenacs18@gmail.com²,

nkbarpanda@suniv.ac.in³, gyan7420@gmail.com

Abstract – In the process of pre-deployment decision-making, an emulation-assisted system model can quickly become unrealistic for evaluating complex message channels. Over the years, many strategies have been employed for transport support and management. However, recent attention over intelligent SMART based (self-monitoring, analysis and reporting techniques) indicate a specific change in the transport system in academic and industry-coordinated systems. To improve overall performance, various Bus standards for cars are introduced, one of them as Controller Area Network (CAN). Creating SMART-based control of CAN-Bus is a daunting problem, which is being seen in recent needs, in which there are more real-time limitations with the introduction of electric launcher or driver support. However, Vehicles will take on SMART-based computing and communication platforms whose inclusion further enhance sensory capabilities. On this work, we have presented a way to control the smart-based CAN-Bus for in-vehicle automation. It enables new multipurpose systems with increasing transport safety and efficiency measures. This paper surveyed sophisticated methods, resolutions and knowledge in a broad range for the in-vehicle communication systems.

Keywords- SMART, CAN, CAN-bus Communication, Vehicular Networks, Intelligent Transportation Systems

1. INTRODUCTION

It was only by Bosch corporation of German, for which controller area network (CAN) is technologically advanced for in-vehicle communication network system, where the electronic control unit (ECU) was interconnected without the peer system [1]. All automotive productions have faced major development challenges of the 21st century. Vehicle systems have an automotive electrical structure that selects a large number of computer processing tasks [2] [3]. CAN provides real-time control by sending and receiving messages in any pair of nodes. It can be strong against flaws and with less complexity and cost [3] [4]. With this capability, the CAN Bus is widely used for many applications such as railroads, aircrafts, etc. Today, many applications are built in vehicles like GPS, camera, sensor and black box. In future, if the car is driven without a driver, the number of implanted systems in the car will be increased and it will be more materialistic. Since the CAN data rate typically reaches 1 Mbps, the processing power is limited to low frequency communication applications. The flexible data rate of the controller network (CAN-FD) can support up to 10 Mbps data rates. However, since it is a standard overclocking system, the CAN-FD system does not provide background synchronization [4][5]. For example, we have recently proposed a high-end bandwidth CAN system, which improves data throughput dramatically while maintaining background synchronization. As a result, Automakers are taking advantage of the latest technology so that their vehicles can stand in the

competitive market [5]. By accepting the demands of increasing operation in spontaneous search, engineers bring intelligent computing systems, which are different types of CAN-based SMART systems. Recently, car manufacturers have supported SMART technology. SMART aided drives a set of tested features that can be improved over time. There may be many ECU SMART-based CAN-Bus controls for different sub-systems in a modern vehicle such as airbags, anti-lock braking, engine control, audio system, wheel control, windows, doors, mirror correction etc. for which the message is generating & communicating among dependent crucial in-vehicle subsystems which is essentially a life saving measure [4][5][6]. The conformist electronic control system may improve car slackening and mobility. Still, there is a need to address issues such as body cables, space restrictions and reliability issues. To solve these problems, the car network engineering continues to be formed. On automotive vehicular networks, protocols must meet basics, including significant improvising in in-vehicle channel flow control, anti-strike body protection and decrease of error value, delayed efficiency, error diagnosis, flexibility parameters and intelligent control levels [7]. Different subsystems require exchange of performance and fixed position in specific defined time. The SMART CAN-Bus message passing channel provides a simple but effective communication protocol between programs, controls (machineries), activators, sensors and other nodes in real time manner without any significant down flow

[8][9]. Accessibility and standardization of CAN-Bus technology has opened up tremendous opportunities for researchers and will work as a platform for new equipment management and the delivery of agricultural product discipline. Power is just starting to see the sights.

2. RELATED WORKS

As presented by Farsi, M. et al., (1999) the CAN use popularity due to ease of routine, ensures extremely high efficiency & reliability. It is easy to use as a serial URT, and now the cost of CAN controllers is still decreasing because it has reached more applications in areas such as medicine, automation industry & in housekeeping equipment. As reported by Papadimitratos, P. et al., (2009) the highly automated vehicles (HAVE-IT) for intelligent transportation predict the availability of in-vehicle CAN-based communication. The project develops a co-pilot that optimizes the categories of tasks between the driver and the co-driving system with a void and failsafe electronic SMART architecture system architecture for collaborative data synthesis in the future. As stated by Moniruzzaman, M. et al., (2016) since ECU numbers increase in cars, CAN-Bus has been introduced to meet the high-quality data transfer requirements between the various components. Real-time, Reliability and Flexibility - these features make CAN-Bus an icon network communication technology applicable to car network communication. As given by Vegni, A. M. et al., (2013) the main technology used in the vehicle network for smart technology. Different technologies can be utilized to provide communication, control and security capabilities, as well as entertainment agreements and apps are well supported. described the end-to-end channel modeling of CAN communication using a transmission matrix and verified it through experiments.

3. SMART CAN BUS DESIGN

CAN offers very low cost, high reliability and excellent realtime performance. This is a vehicle auto-diagnosis model template designed to allow ECUs and nodes to communicate with each other without using the host. CAN allows engineers to use bandwidth to improve the new functionality provided by the multinational's for steadfastness, performance, and distributed software features, to improve the current Electrical and Electronic (EE) functions and to import new data with the help of self-diagnosis tool. This is mainly done by sensors that send unusual data packets.

A. SMART CAN-Bus Communication

SMART CAN is a multi-master serial universal receive & transmit bus model for connecting ECU in real time basis. A real design shows how the electronic components are scattered in the control architecture for the vehicular network system. Some

of these independent subsystems & others need uninterrupted communication between them. The block represents ECUs and thick line networks. Generally, there are three ECUs divisions: energy train and chassis, entertainment and body electronics. Many network subsystems are used to connect. It can be around 60 ECUs in the maximum configuration at the same time for the in-vehicle CAN system.

B. Architype Model

The SMART-based CAN-Bus communication has been illustrated in figure 1. It exhibits message communication among the host nodes which are sent to SMART controller for further auto-diagnosis. It is an analyzer program, is used for fault investigation. A CAN community is generally attached to sensors, activators and other controllers

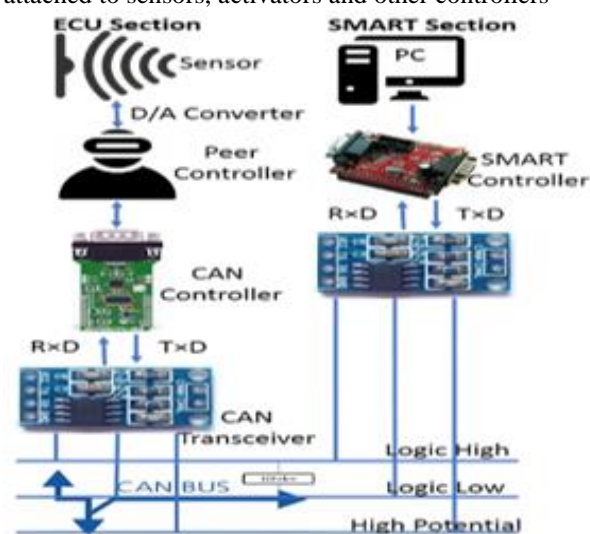


Fig. 1. SMART-based CAN-Bus communication.

Not necessarily devices are directly connected to the channels but by a central chip and a CAN controller. CAN nodes can include host chips, controllers and transceivers. A separate CAN-Bus line is interconnected.

C. SMART CAN-Bus flowchart

This section describes the process of emulation for SMARTbased CAN-Bus communication using frame feedback flow chart system which can be self-discovered through network node during runtime. SMART bus vehicles can be the primary communication between the number of ECUs, so the self-diagnosis and its responses to the in-vehicle system have been fully discovered. Frame response time delivery provides sufficient information about the system behavior in real time basis.

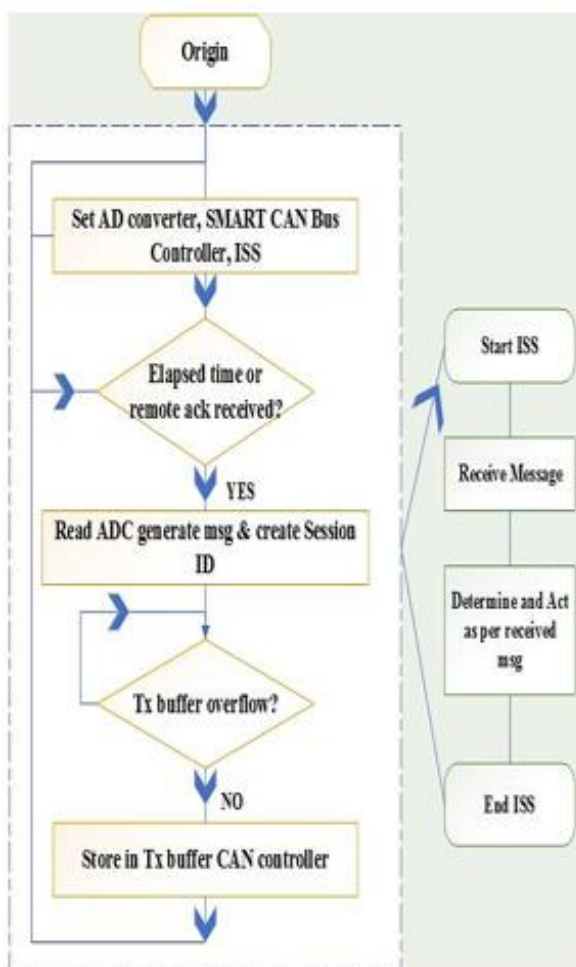


Fig. 2. Flowchart for SMART-based CAN-Bus communication. [Where, ISS is for interrupt service subroutine.]

D. SMART CAN-Bus controller organization

This section shows the typical organization for the SMARTbased CAN-Bus controller for an in-vehicle message passing channel in figure 3.

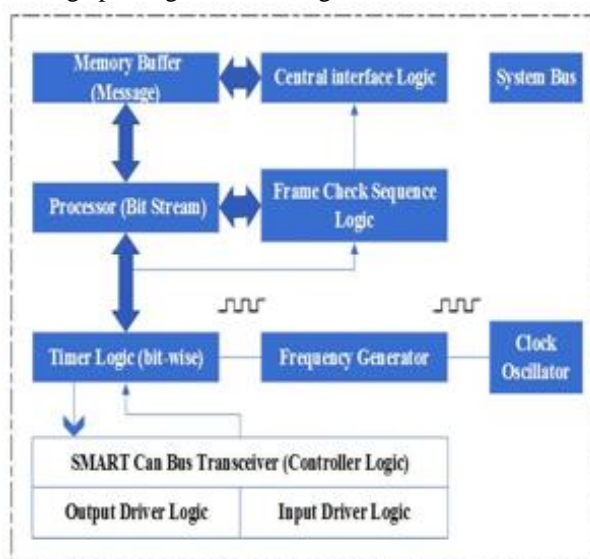


Fig. 3. SMART CAN-Bus controller organization.

- *Central Interface Logic (CIL)* performs commands from peer nodes and checks information transfer on serial universal receive & transmit (URT) buses. Universal Status and Control Registers and Contact Object Bits are mainly used by CPU interface logic.
- *The processor (bit stream)* It checks the data stream between parallel buffer and serial bus lines. After checking entire protocol, it distinguishes between frame types and notices frame errors.
- *Frame check sequence logic (FCS)* helps to identify the streams of bits by detecting error messages from the processor and instead sends information about the errors to the bit stream logger and it sends to the central logic SMART interface unit.
- *The timer logic* determines the time interval of the bits and smart CAN-Bus for bitwise edge synchronization. It helps to render the comparison of channel lines by comparing a differential input. Synchronizes an unalignment at the beginning of the frame and reconnects the frames during compression. Provide programmable time slots to compensate for any transit delays and phase changes.
- *SMART transceiver logic* comprises of bit stuffing, programmable output driver, cyclic redundancy check and data shift registers logic. It combines the various components of a logical transceiver. Message input, arbitration, messaging and error messages are actually executed by this unit.
- *Buffer message memory* stores individual CAN data for transmission or reception. The CPU communicates only with this area in order to transmit and receive messages. The bus interface logic manages the bus traffic.
- *Frequency Generator* is simply used to derive a suitable clock frequency for the CAN controller based on the frequency of an external clock oscillator.

E. Task distribution using SMART CAN-Bus

At the time of the mapping for in-vehicle message passing channel, shown in Figure 4, work is distributed into the ECU and the signals are packed into messages.

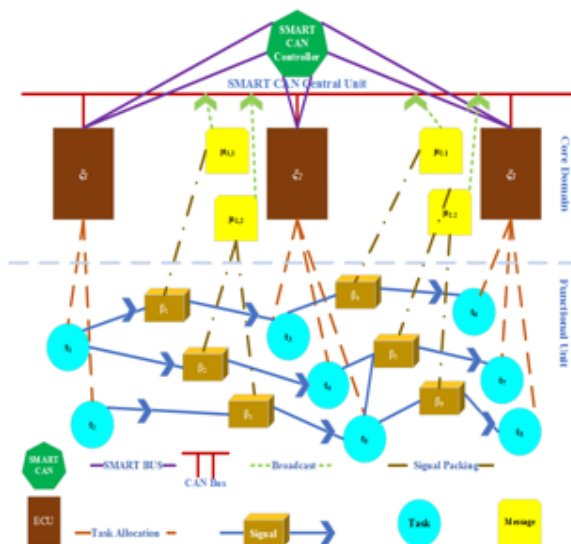


Fig. 4. SMART based in-vehicle CAN-Bus system model.

Static cum dynamic priority works and messages are allocated. The goal is to look for design of signal packaging, priority allocation, and a set of SMART assignment.

F. SMART CAN-Bus controller Initialization

With the aid of the internal counter registers and the inbox registers the SMART CAN controller can easily be initialized. After system on, when it is about to initialize, the SMART registers in the monolithic chip will clear the read and write system buffer, sets the frequency register, code cum message sending receiving at hand registers & interrupt controller. Then it is in normal state and ready to initialize the SMART CAN-Bus controller for in-vehicle system. The initialization code is as follows

```
Device: 'Virtual 1'
Device Channel Index: 2
Device Serial Number: 0
```

To Create a SMART CAN channel on which to transmit is txCh, we have to register each message on the channel at a specified periodic rate as given below

```
transmitPeriodic(txCh, msgTx100, 'On', 0.500);
transmitPeriodic(txCh, msgTx200, 'On', 0.250);
transmitPeriodic(txCh, msgTx400, 'On', 0.125);
transmitPeriodic(txCh, msgTx600, 'On', 0.050);
transmitPeriodic(txCh, msgTx800, 'On', 0.025);
```

After the successful emulation of SMART CAN txCh it is necessary to check among the CAN-Bus about the packet arrival time, time to live (TTL), rate of packet recurrence, jitteriness, throughput rate and access time of command and response (C/R) generation by running ECU nodes and its ID for several seconds by incrementing the message data regularly to ensure the in-vehicle system validation as tested and given by code below.

```
for ii = 1:50
```

```
Increment the message data bytes.
msgTx200.Data = msgTx200.Data + 1;
msgTx400.Data = msgTx400.Data + 1;
msgTx600.Data = msgTx600.Data + 1;
msgTx800.Data = msgTx800.Data + 1;
Wait for a time period.
```

```
pause(0.100);
end
```

Stop the CAN channel.

```
stop(txCh);
end
```

4. DESIGN VERIFICATION

After the successful execution of code as SMART CAN-Bus identifier with timestamp, the executed result has taken

TIME	ID	EXTENDED	NAME	DATA	LENGTH	SIGNALS	ERROR	REMOTE
0.007296 sec	100	FALSE	"	[1:0uint8]	0	[0:0stnct]	FALSE	FALSE
0.0073045 sec	200	FALSE	"	[1:2uint8]	2	[0:0stnct]	FALSE	FALSE
0.007311 sec	400	FALSE	"	[1:4uint8]	4	[0:0stnct]	FALSE	FALSE
0.0073162 sec	600	FALSE	"	[1:6uint8]	6	[0:0stnct]	FALSE	FALSE
0.0073213 sec	800	FALSE	"	[1:8uint8]	8	[0:0stnct]	FALSE	FALSE
0.038459 sec	800	FALSE	"	[1:8uint8]	8	[0:0stnct]	FALSE	FALSE
0.028773 sec	600	FALSE	"	[1:6uint8]	6	[0:0stnct]	FALSE	FALSE
0.028785 sec	800	FALSE	"	[1:8uint8]	8	[0:0stnct]	FALSE	FALSE
0.06008 sec	800	FALSE	"	[1:8uint8]	8	[0:0stnct]	FALSE	FALSE
0.091249 sec	400	FALSE	"	[1:4uint8]	4	[0:0stnct]	FALSE	FALSE
0.091264 sec	600	FALSE	"	[1:6uint8]	6	[0:0stnct]	FALSE	FALSE

Table 1. (11 × 08) CAN identifier timestamp table.

in the form of 11 × 08 time table as given in the table 1 with its corresponding simulated plot given in the

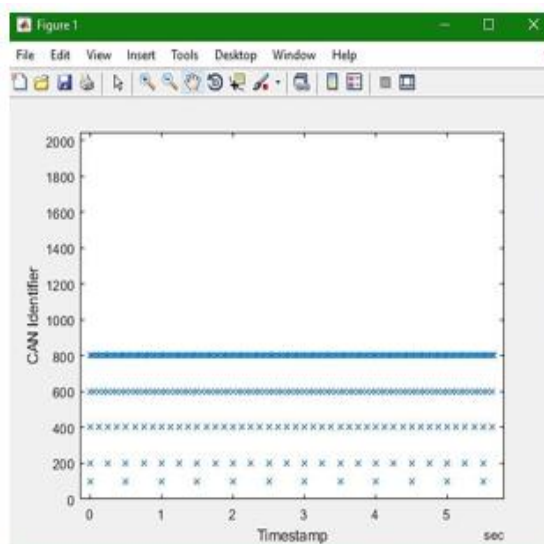


Fig. 5. CAN identifier Vs. Timestamp plot.

After designing SMART CAN-Bus to send messages inside the car, including software and hardware, we need to verify our system's permanency, consistency and data compatibility. After receiving the system, select the battery combination, afterward, 11 ECU groups can be tested with the CAN identifier. Record the information received from the bus machine through

the SMART CAN-Bus data buffer in every 5 seconds. And the tab records the collection results as follows in Table 1. Verified that a large number of real-time packet information sanded from the in-vehicle CAN node is accurate, stable, and no lost number, frame dropping, the number of strings of communication such as False. Stability of system functionality, reliability and data consistency requirements are complied with. It has been verified that a large number of real-time packet information stored on the car's CAN node are accurate, fixed, and no packets are been dropped and false. Compliance with the stability of system functionality, dependability and compatibility requirements. Since it was recently adopted for a new car, the work could be useful for the new in-vehicle SMART based CAN-Bus network development.

5. CONCLUSIONS

In real-time and versatility, this model can apply a necessary SMART-based CAN-Bus uninterrupted message communication technology which is indispensable and applicable to all variant of automotive in-vehicle network communication systems successfully. The main decisions are the efficient use of CAN-Bus utilization smartly and the distribution of identifiers among various ECUs without fail. The results of the analysis are verified through a MATLAB simulation magnificently. It has some drawbacks, some of these issues include the worst-case message delay and priority isolation in some cases. Our analysis performance measures its impact, using self-monitoring analysis and reporting techniques in the form of CAN-Bus model for in-vehicle message passing channel successfully.

REFERENCES

- [1] Tindell, K., Hanssmon, H., & Wellings, A. J. (1994, December). Analysing Real-Time Communications: Controller Area Network (CAN). In RTSS (pp. 259-263).
- [2] Sevillano, J. L., Pascual, A., Jiménez, G., & Civit-Balcells, A. (1998). Analysis of channel utilization for controller area networks. *Computer Communications*, 21(16), 1446-1451.
- [3] Misbahuddin, S., & Al-Holou, N. (2003, July). Efficient data communication techniques for controller area network (CAN) protocol. In ACS/IEEE International Conference on Computer Systems and Applications, 2003. Book of Abstracts. (p. 22). IEEE.
- [4] Lee, K. C., Kim, M. H., Lee, S., & Lee, H. H. (2004). IEEE-1451-based smart module for in-vehicle networking systems of intelligent vehicles. *IEEE Transactions on Industrial Electronics*, 51(6), 1150-1158.
- [5] Johansson, K. H., Törngren, M., & Nielsen, L. (2005). Vehicle applications of controller area network. In Handbook of networked and embedded control systems (pp. 741-765). Birkhäuser Boston.
- [6] Papadimitratos, P., Fortelle, A. L., Evensen, K., Brignolo, R., & Cosenza, S. (2009). Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *IEEE communications magazine*, 47(11), 84-95.
- [7] Shinde, A. S., & Dharmadhikari, V. B. (2012). Controller area network for vehicle automation.
- [8] Vegni, A. M., Biagi, M., & Cusani, R. (2013). Smart vehicles, technologies and main applications in vehicular ad hoc networks. In Vehicular technologies-deployment and applications. IntechOpen.
- [9] Lin, C. W., Zhu, Q., & Sangiovanni-Vincentelli, A. (2015). Security-aware modeling and efficient mapping for CAN-based real-time distributed automotive systems. *IEEE Embedded Systems Letters*, 7(1), 11-14.