# SFICAN BASED INFRASTRUCTURE FOR AUTOMOTIVE APPLICATION

P.Sheela, J.Gobinath

**Abstract--** Star based infrastructures have been proposed for automotive application. In the existing system physical fault-injection is carried out using sfiCAN protocol in which CAN-Compliment hub allows injecting fault independently in transceiver bits. It recreates fault scenarios beyond the capability of the injectors of the CAN. Here only Software implementation is carried out with the lack of time resolution. This conflict is rectified in proposed system. The proposed system designs a CAN topology, which is converted to star formation in order to increase the efficiency of the system which is implemented in motor control and energy consumption application. Automated systems are governed and controlled by server unit. It saves the times resolution and implementation of hardware is carried out efficiently.

**Index terms**- Fault-injection, sfiCAN

## 1 INTRODUCTION

The Control area network (CAN) is a serial bus protocol with low cost, robust technology like Ethernet. CAN is a broadcast bus which has priority to access the medium which makes it as de-facto standard for data transmission in automotive applications [1]. It is widely used field in buses in distributed control systems, and still today such as CAN with a flexible data rate (CAN-FD) and ISO 11898-6. In fact it is withstanding old and new markets, and new higher end applications are upcoming [2]. In automotive field Can is most widely used network technology and new CAN based application are expected in upcoming market [3]. Usage of Can is more controversial in critical automotive applications such as AUTOSTAR is established which is one of the fundamental technologies [5]. Interest in integrating CAN with newer felid buses such as FlexRay [6] which are used in high speed backbone with increased bandwidth [7], [8].

This interest in CAN is due to low cost and as per current economic situation investing on a newer technology is more expensive even though some practical difficulty have been noticed in the development of distributed systems with newer field buses such as FlexRay [9]. Proliferation of CAN network can be observed in other sectors. In aerospace application CAN network have been integrated with other networks as reflected in the development of ARNIC 825 standard [11]. This standard had made CAN suitable for flight safety-critical systems in future aircrafts, where this technology is intend to be used as the primary or secondary network. CAN can be used in other safety related protocols such as DeviceNet Safety, SafetyBUS p and forthcoming CANopen Safety EN 50325-5 [5] and [12]-[20].

A CAN system comprises of a set of hubs that exchange messages emulating the CAN convention to work helpfully. Hubs of a CAN system are constituted of three fundamental components: a processor, regularly a microcontroller, which executes the application programming; a CAN controller, which actualizes the greater part of the CAN convention; and a CAN handset, which basically adjusts the transmission and gathering signs to the correspondence medium.

An expanding number of CAN systems require their processors to have a synchronized clock. Notwithstanding, as the CAN convention does exclude such administration, this must be given by executing a synchronization calculation at the application layer. Those synchronization calculations are commonly executed in the product executed by the processor [4], [10], and [20] however this sort of implementation has a few impediments. In the first place, it requires imperative changes in the product of the processor, which have an innate expense and many-sided quality. Second, since the interface between the processor and the CAN controller has not been institutionalized yet, this product executed administration of clock synchronization is not totally free of the CAN controller.

At that point, if the CAN controller was supplanted, the product ought to be significantly changed once more. Lastly, as the synchronization data is transformed by programming in the processor and not at the equipment level, and then there are critical latencies that hinder to accomplish a high exactness in the clock synchronization. Rest of the paper is designed as follows Section I includes

- *P.Sheela is currently pursuing master's degree in embedded system and technologies from Rajiv Gandhi College of Engineering, Chennai. Email id- sheela.alwar30@gmail.com*
- *J.Gobinath M.E, (Assistant professor), Head of the department of Computer science engineering, Rajiv Gandhi College of Engineering, Chennai. Email id-mail2gobi@gmail.com*

introduction of sfiCAN, section II consist of design of sfiCAN, section III includes syatem implementation and finally section IV consist of conclusion and future work.

## 2. DESIGN OF SFICAN

sfiCAN is made out of a situated of parts that agreeably work to do a flaw infusion test, i.e., a trial amid which the conduct of a given target framework is investigated when it is compelled to manage blunders incited by flaws. The focal component of sfiCAN is a centre to which the hubs of the framework and a PC-based administration station are joined. This centre executes a coupling diagram focused around that of Concentrate. From one viewpoint, the centre gives a CAN arrange that permits conveyance of the diverse parts of sfiCAN as a set of system configurable segments (NCCS), i.e., segments whose operation can be designed and facilitated remotely through the system.

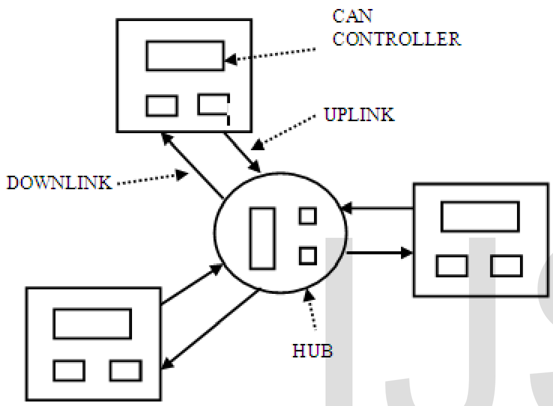


**FIGURE 1 SFICAN BLOCK DIAGRAM**

The NCCS are placed inside the center point and on the framework's hubs. Then again, the center point's coupling blueprint permits to execute propelled fault injection also logging peculiarities inside the center itself. A content document called the flaw infusion detail, which is put away in the PC, contains the portrayal of the flaws to be infused in an investigation. The PC designs the NCCS, in understanding with this determination, triggers the execution of the test, and once it is done, recovers the data logged by the NCCS.

Next, all these ideas are clarified in more detail.

### 2.1 SFICAN BASICS

Figure1 shows how every hub is associated with the center point of SfiCAN by method for a devoted connection contained a differentiated uplink and downlink. The hub's CAN controller interfaces with two COTS CAN handsets as in Concentrate [41], in which one handset is utilized to transmit through the uplink and another to get from the downlink. Figure 2, which demonstrate the inner structure of the centre, shows how every hub commitment, i.e., Bi, is gotten through the comparing uplink. At the point when no shortcoming is being infused, every commitment spreads from its uplink multiplexor, i.e., dmuxi, to the coupler module, which couples every one of them by method for an AND entryway. At that point, the resultant coupled sign, i.e., B0, passes through every downlink multiplexor, i.e., dmux, and is show again to the hubs by means of the downlinks. Since this coupling is carried out in a small amount of the bit time, the edge watched at B0, i.e., the resultant edge, is the same as in a CAN transport, in this manner making the center straightforward for the hubs. Then again, interestingly to a transport, a star topology permits the center to recognize the sign that every hub generally transmits and gets. In this manner, as restricted to transport based issue injectors, sfiCAN can infuse channel flaws that influence a few hubs however not others and can focus the reaction of every individual hub to the infused shortcomings. The center likewise has a devoted port to which the PC-based administration station is joined utilizing a COTS CAN controller board. The relating connection is not differentiated into an uplink also downlink since no flaws are infused into this connection. This additional association permits the PC to remotely arrange and coordinate the distinctive NCCS. The NCCS are the components that really infuse blames and log data concerning these deficiencies and their outcomes on the framework. The diverse sorts of NCCS are the CFI, the center point lumberjack (HL), and the hub lumberjacks (Nls). The CFI and the HL are equipment modules found inside the centre point as shown in figure 3.

The previous is in charge of infusing shortcomings at the sign transmitted or got by every hub; though the HL logs data about each one casing telecast amid the fault injection test. For each one casing, the HL logs its source port, identifier, information field, and, if the casing is influenced by an mistake, additionally the area of that blunder (edge field and bit number). Concerning NL, it is a bit of programming connected to each hub's application that logs data about the inside state of the hub amid an examination. The NL accumulates the casings that the application effectively transmits or gets and the estimation of the transmission/gathering slip counters (TEC/REC) [1] of the CAN controller.

This data makes it conceivable to figure out which outlines the hub rejected and the condition of the controller itself, i.e., slip dynamic, blunder detached, or transport off [1]. In charge of arranging and organizing the NCCS is a product called shortcoming infusion administration station (FIMS), which executes on the PC. The FIMS contains the fault injection detail document and uses an alleged NCC convention on top of CAN (see Section III-D) to arrange every NCC likewise. It additionally utilizes this convention to constrain the NCCS to begin the investigation in the meantime and, once the trial is done, to gather the information observed by every NL and the HL.

## 3. SYSTEM IMPLEMENTATIONS

SfiCAN is designed like a star formation in which many nodes are connected to single hub. In this system

Includes two loads connected to single hub. First load is designed as the motor speed control and second load is energy consumption application. Hardware requirements for these loads are microcontroller (PIC16F877A), relay, max 235, LCD, bulbs, fuse, And Personal computers etc

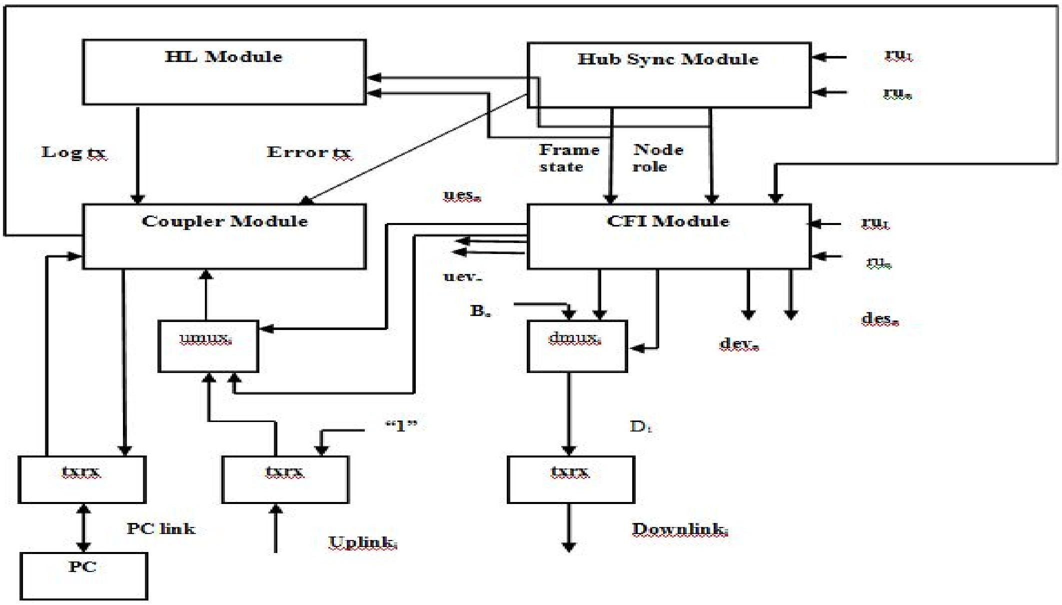


FIGURE 2 DESIGN OF HUB



In first load motor speed control has been designed in which an load is connected with microcontroller. The system is implemented like as shown in figure 3. An 230V supply is given to the step down transformer which convert this 230V to 12V AC supply. Obtained 12 V is given to bridge rectifier which converts 12 V AC to DC supply. Now this 12V is given to the voltage regulator which converts to 5V supply. Obtained 5V is given as supply to PIC , LCD and relay. Initially load runs in 16000 rpm slowly it increases its speed up to 17000 rpm when it reaches 16600 rpm. Cooling agent get on. This cooling agent will decreases the surrounding heat produced by the load. The surrounding heat decreases with the supply of coolant speed of load decreases slowly to 16200 rpm.

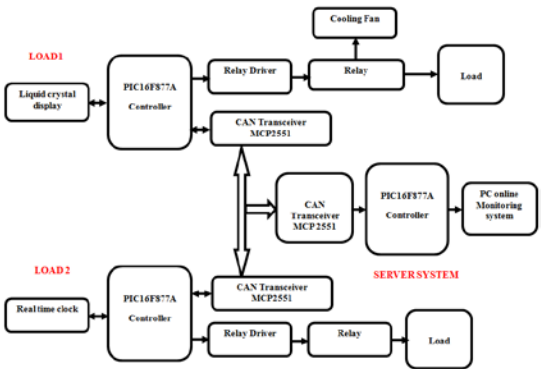


FIGURE 3 BLOCK DIAGRAM

Second load is designed based on the energy consumption. Load is taken here as four bulbs initially complete supply has been provided to the system. When the complete supply is not required only particular bulbs will glow. If any fault is identified then an alternate bulbs will glow.

A centralized hub is an online PC monitoring system. It acts as a server unit which controls and governs the whole system. The whole process of governing has been developed in LAB view. The communication between these nodes is carried out by CAN transceiver.

## 4. CONCLUSION AND FUTURE WORK

This Paper has presented the implementation and design of sfiCAN. An server system is designed as a governing system for two loads. Both the system works at a time. A fault has been introduced in the second load.

Figure 4 shows the implementation of first load in which motor speed control has been carried out successfully. Lab view software is used to display the results of two nodes and related graph have been plotted for the for power consumption in second load. The respective values of Speed in RPM have been displayed in Labview.

In future more faults can be injected and monitoring features can be injected and monitoring features can be added inside in other device e.g the injector with hub can be extended to inject application-specific fault such as babbling-idiot faults.

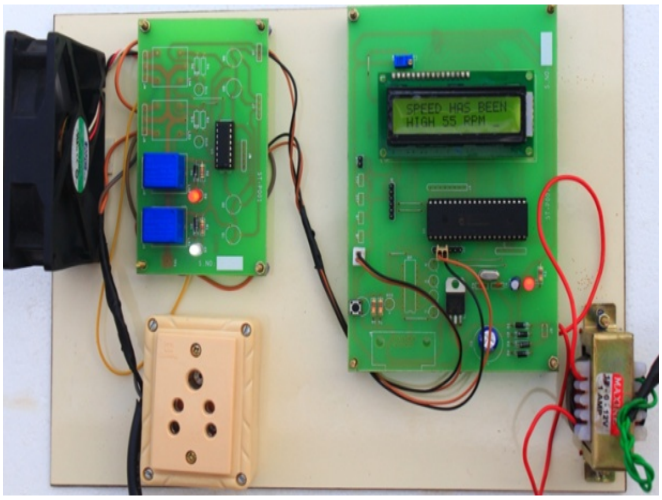


**FIGURE 4 MOTOR SPEED CONTROL**

.

Figure 5 show the graph plotted for energy consumption in normal stage and fault detection stage. And corresponding graph have been shown in Labview.

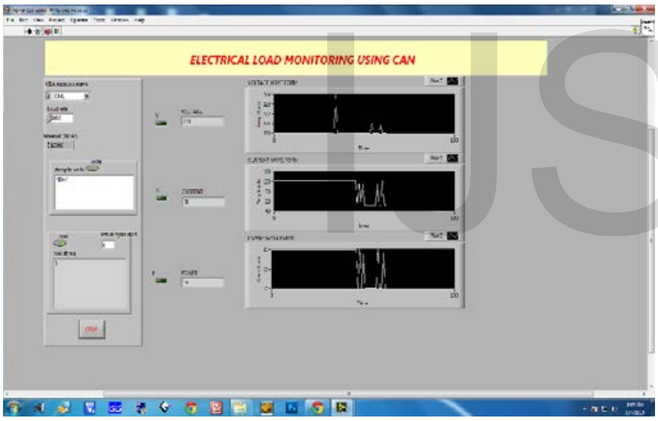


**FIGURE 5 LOAD MONITORING**

Both nodes will run simultaneously and their respective values have been displayed. Fault has been identified when using green LED glows. Then for that an alternative load is provided as alternate. This prevent the operation from dead condition. Other type of fault can be introduced in order to use this network efficiently This sfiCAN concept can be implemented in various lathe application and various automated application in future.

## REFERENCES

[1] *Controller Area Network (CAN) - Part 1: Data Link Layer and Physical Signalling*, ISO Std. ISO11898-1, 2003.

[2] H.Zeltwanger, "Controller Area Network- Introduce 25 year ago," *CAN Newsletter*, no. 1, pp.18-20, Mar. 2011.

[3] S. Kim, E. Lee, M. Choi, H. Jeong, and S. Seo, "Design optimization of vehicle control networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3002–3016, Sep. 2011.

[4] M.Gergeleit and H.Streich, Implementing a Distributed High-resolution Real-time Clock using the CAN-bus, Proceedings of the 1st International CAN Conference, Mainz, Germany, 1994.

[5] P. Lanigan, P.Narasimhan, and T. Fuhrman, "Experiences with a CANoe based fault injection framework for AUTOSAR," in *Proc. IEEE/FIP Int.Conf. DSN, 2010, pp.* 569-574.

[6] R.A. Gupta and M.-Y. Chow, "Networked Control System: Overview and research trends," *IEEE Trans. Ind. Electron*, vol. 57 , no.7, pp, 2527-2535, jul.2010.

[7]F. Baronti, E. Petri, S. Saponara, L. Fanucci, R. Roncella, R. Saletti, P. Abramo, and R. Serventi,"Design and verification of hardware building blocks for high-speed and fault-tolerant in-vehicle networks," *IEEE Trans. Ind. Electron.*, vol. 58, no.3, pp. 792-801, Mar. 2011.

[8] H. Kimm and H-S. Ham, "Integrated fault tolerant system for automotive bus networks," in *Proc. 2th Int.Conf. Comput. Eng. Appl.*, 2010, pp. 486-490.

[9] "Control of sir pollution from new motor vehicle and new motor vehicle engines; modification of federal on-board diagnostics regulation for: light-duty vehicle, light-duty trucks, medium duty passenger vehicles, complete heavy-duty vehicles and engines intended for use in heavy duty vehicles weighing 14,000 pounds GVWR or less," *uniteds Staes Fed. Reg.*, vol. 70 no. 243, pp. 75 403-75 411, 2005.

[10] L.Rodrigues, M. Guimaraes and J. Rufino, Fault-tolerant Clock Synchronization in CAN, Proceedings of the 19th IEEE Real-time Systems Simposium, Madrid, Spain, 1998.

[11] J. Munoz-Castaner, R. Asorey-Cacheda, F.Gil-Castineira, F Gonzalez-Castano, and P.Rodriguez-Hernandez, " A Review of Aeronautical Electronics and Its Parallelism with Automotive Electronics, " *IEEE Trans. Ind. Electron.*, vol. 58, no 7, pp. 3090-3100, Jul. 2011.

[12] T. Nolte, M. Nolin, and H. Hansson, "Real-time server-based communication with CAN," *IEEE Trans. Ind. Informat.*, vol. 1, no. 3, pp. 192-201, Aug 2005.

[13] B. Gaujal and N. Navet, "Fault confinement mechanisms on CAN: Analysisand improvements," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 1103–1113, May 2005. [14] J. Rufino, C. Almeida, P. Verissimo, and G. Arroz, "Enforcing Dependability and Timeliness in Controller Area Networks," in *Proc. 32nd Annu. IEEE IECON*, Nov. 2006, pp. 3755–3760.

[15] J. Ferreira, L. Almeida, J. Fonseca, P. Pedreiras, E. Martins, G. Rodriguez-Navas, J. Rigo, and J. Proenza, "Combining operational flexibility and dependability in FTT-CAN," *IEEE Trans. Ind. Informat.*, vol. 2, no. 2, pp. 95–102, May 2006.

[16] K. Schmidt and E. G. Schmidt, "Systematic message schedule construction for time-triggered CAN," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3431–3441, Nov. 2007.

[17] G. Buja, J. R. Pimentel, and A. Zuccollo, "Overcoming babbling-idiot failures in CAN networks: A simple and effective bus guardian solution for the FlexCAN architecture," *IEEE Trans. Ind. Informat.*, vol. 3, no. 3, pp. 225–233, Aug. 2007.

[18] M. Short and M. J. Pont, "Fault-tolerant time-triggered communication using CAN," *IEEE Trans. Ind. Informat.*, vol. 3, no. 2, pp. 131–142, May 2007.

[19] B. Hall, M. Paulitsch, K. Driscoll, and H. Sivencrona, "ESCAPE CAN limitations," *SAE Trans. J. Passenger Cars—Electron. Elect. Syst*, vol. 116, pp. 422–429, 2008.

[20] J. Pimentel, J. Proenza, L. Almeida, G. Rodriguez-Navas, M. Barranco, and J. Ferreira, "Dependable automotive CANs," in *Automotive Embedded Systems Handbook*, N. Navet and F. Simonot-Lion, Eds. Boca Raton, FL, USA: CRC, 2008, ch. 6, pp. 1–56.