

Verilog Implementation of Collision Avoidance System based on FlexRay Protocol

S. Athira Nair*, T. V. Pooja, Seethalakshmi V. Nair, Veena U. Pillai and S. Rajalekshmi

Department of Electronics and Communication, Amrita Vishwa Vidyapeetham, Kollam - 690525, Kerala, India;
athirasn1994@gmail.com, poojamananoj27@gmail.com, seethalakshmiavnair1994@gmail.com,
veena.u11@gmail.com, rajalekshmis95@gmail.com.

Abstract

Background/Objective: The objective of this paper is to provide an insight into the FlexRay protocol, as well as build a simple collision avoidance system based on it. FlexRay is a next generation intra-vehicular communication protocol that provides fault-tolerant, flexible, high data rate and deterministic communication between the different electronic units in an automobile, mainly used for advanced automobile control and safety-critical applications. **Methods/Statistical Analysis:** Here, a simple collision avoidance system for an automobile is built using Verilog HDL, where data is transmitted according to the FlexRay protocol. Separate Verilog modules are designed for an ultrasonic sensor and a motor. Data is transmitted in the FlexRay frame format. Simulation of results are tested in ModelSim tool. **Findings:** It is observed that under the FlexRay protocol, data is transmitted at a much greater speed. Also, there are better fault tolerance and better error detection mechanisms, all of which are quintessential for a safety-critical application like collision avoidance. It also ensures safe communication between the devices inside an automobile. **Applications/Improvements:** More advanced automotive applications like automatic steering, temperature regulation, etc., can be implemented in a similar manner based on the protocol. The protocol is mainly used for safety-critical applications owing to its huge cost of implementation.

Keywords: Cyclic Redundancy Check, FlexRay, Frame Format, Generator Polynomial, Intra-Vehicular Communication

1. Introduction

With the ever increasing need for huge bunches of data to be transmitted with high reliability between the different electronic components in an automobile, the FlexRay protocol has been developed to meet the present day demand for a highly reliable, fault tolerant, flexible, high data rate communication protocol. The FlexRay protocol, with a high data rate of 10 Mbps, with favourable error detection techniques, and featuring both time and event triggered transmissions, is expected to become the future standard for intra-vehicular communication. Its scalable fault tolerance is achieved by featuring a dual channel communication. It displays a much better performance compared to its predecessors, Controller Area Network (CAN) and Local Interconnect Network (LIN). In the FlexRay topology, the nodes may be connected in active star, serial bus or hybrid configuration. Each node is a communication

interface of a specific electronic device within the automobile. The paper mainly discusses how the transmission of data happens according to the protocol, and the FlexRay frame format.

Sections 2, 3 and 4 of this paper give a general overview of the FlexRay communication protocol and its features. The FlexRay frame format and communication cycles are discussed here. The 5th Section deals about a simple collision avoidance system that can be designed, where data is being transmitted according to the FlexRay protocol. The simulation results in Model Sim are given in the sixth section. The 7th Section presents the conclusion and future scope.

2. FlexRay Frame Format

In the FlexRay protocol, data is transmitted serially, and in the form of frames. Each slot of the static segment or

*Author for correspondence

the dynamic segment has a FlexRay frame. The FlexRay frame consists of three sections - the Header, the Payload and the Trailer¹:

- Header

The header segment is 40 bits long and it begins with the five start bits, each of which defines certain basic features of the frame. It is followed by the 11 bit frame ID, which identifies the frames, the 7 bit payload length that gives the number of bits transmitted in a frame. The 11 bits of Header Cyclic Redundancy Check (CRC) are used to detect errors during the transmission of Header. The 6 bit Cycle Count gives the value of a counter that increments every time a new communication cycle starts.

- Payload

The payload section contains the actual data that is transmitted by the frame. Its length can be up to 254 bytes. It is much more than that in CAN.

- Trailer

The Trailer segment is 24 bits long, and consists of three 8 bit CRCs, which are used to check errors during the transmission of Header and Payload together. An illustration of the FlexRay frame² is given in Figure 1.

3. FlexRay Communication Cycle

The communication in FlexRay is between different nodes, the nodes representing the different electronic components of an automobile. The FlexRay communication cycle has a typical length of 1 to 5 milliseconds. It consists of the Static segment, the Dynamic segment, the Symbol Window and the Network Idle Time³.

- Static segment - This time slot is for deterministic data that arrives at a fixed time.
- Dynamic segment - This slot is for the non-deterministic, event triggered data.

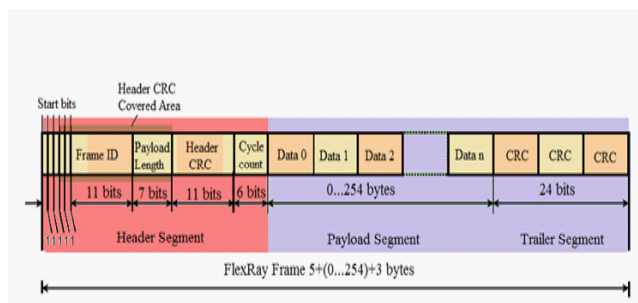


Figure 1. Frame format¹.

- Symbol Window - It is mainly for the maintenance of the network. It is of a known length.
- Network Idle Time - This is the time slot where synchronization between the clocks of the different nodes is achieved. No other action is performed during this time. An illustration of the FlexRay communication cycle is given in Figure 2.

4. FlexRay Communication Controller

The FlexRay Communication Controller consists of:

- Controller - Host Interface: It is the interface between the host and the controller. It controls the command and message interfaces, and also controls the buffers.
- Protocol Operation Control: It reacts to commands of the host and also changes the modes of operation.
- Clock Synchronization: It deals with synchronization between different nodes. The smallest synchronized unit of time is called Macrotick.
- Media Access Control: Deals with transmission. It assembles the header of message.
- Frame and Symbol Processing: It deals with reception. It also checks for errors.
- Coding and Decoding Channel: It encodes the data for transmission, adds CRC bits to the data, decodes the received data, and also checks for errors in the received frames.

5. Collision Avoidance System

The design of a simple collision avoidance system for an automobile, where data is being transmitted in the

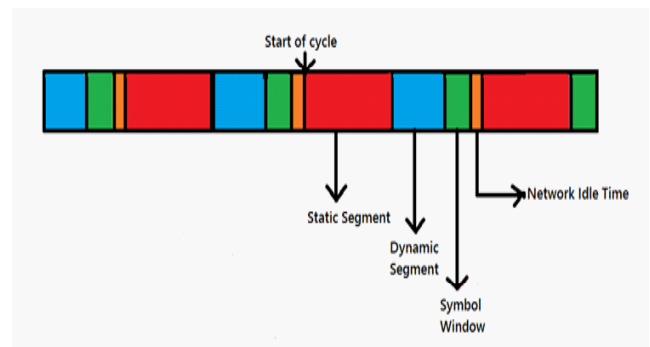


Figure 2. FlexRay communication cycle.

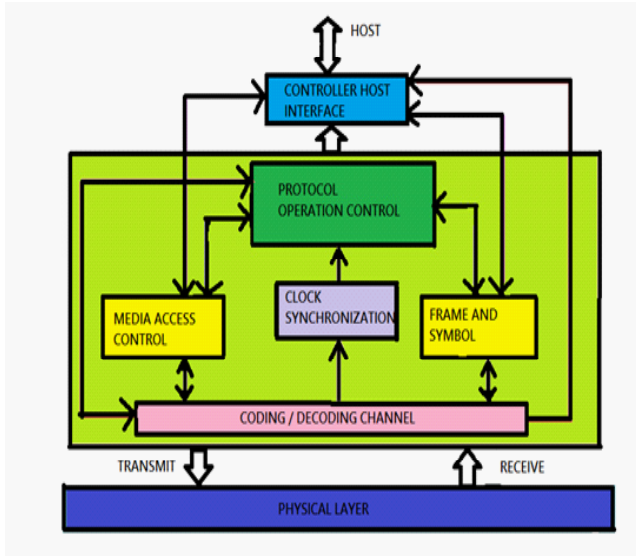


Figure 3. FlexRay communication controller.

FlexRay frame format, is achieved using Verilog HDL. The simulation of the results is obtained in ModelSim. First, we have the Verilog module for an ultrasonic sensor. The ultrasonic sensor has to measure the distance of the obstacle from the vehicle. The echo signal is input to the ultrasonic sensor, while the trigger signal comes as output of the sensor. The echo signal is, actually the signal reflected off from the obstacle. The trigger is a short pulse that stays on for a very short duration of time. Then, it stays at zero or low for a specific duration of time. It is during this time that the echo signal from the obstacle is awaited. Taking into account the time taken for the error signal to be received at the sensor, and the speed of radio waves in space, the distance of the obstacle, from the automobile can be calculated.

Distance of the obstacle from the sensor = Distance travelled by signal in 1 clock cycle \times Number of clock cycles. (1)

Where,

Distance travelled by signal in 1 clock cycle = Speed of radio waves \times Duration of 1 clock cycle. (2)

Once the distance of the obstacle is obtained, it is time to send appropriate signals to the receiver. For convenience, we pick certain limits, like, if the distance is less than 1m, the motor needs to be signaled to stop immediately. If the distance is between, say, 1 m and 5 m, the motor is signaled to reduce its speed. In a similar manner, the rotation of the motor is controlled according to the distance from the obstacle. We set up a counter and the

number of clock cycles up to which it needs to count can be obtained from Equation (1). For each distance, we set as limits, we can get the number of clock cycles for the counter by substituting for distance in Equation (1).

5.1 Cyclic Redundancy Check

Cyclic Redundancy Check is used to detect any errors during transmission. In the header section of the FlexRay frame, we have 11 bits as header CRC bits. In the trailer section, we have three 8 bit CRCs for header and payload together. Separate CRC polynomials are used for each⁴.

For the 11 bit header CRC, the generator polynomial is: $x^{11}+x^9+x^8+x^7+x^2+1$.

For the 24 bit trailer CRC, the generator polynomial is: $x^{24}+x^{22}+x^{20}+x^{19}+x^{18}+x^{16}+x^{14}+x^{13}+x^{11}+x^{10}+x^8+x^7+x^6+x^3+x+1$.

With the data bits that are to be transmitted, the corresponding CRC bits are also attached. These bits are obtained based on the remainder during division of the data bits by the corresponding generator polynomial. The resulting bits obtained are transmitted.

5.2 Finite State Machine

The transmitted data is then passed through a Finite State Machine module. It is here where the Header and Trailer segments are added to the data. The Header segment, with the start bits, Frame ID, payload length, and cycle count, as well as the trailer segment with the CRC bits are added to the data. The Finite State Machine used here has 18 states. Once the data comes out of the Finite State Machine, it achieves the FlexRay frame format.

The frame is then transmitted serially, one bit at a time, to the receiver section.

5.3 Reception of Data

At the receiver end, the bits are received serially, and the payload segment is separated from the frame. The payload data is now used to control the motor module. A counter is set up to control the speed of rotation of the motor. The number of clock cycles for the counter is calculated using equations mentioned before. The motor either stops immediately, i.e., brings its rpm to zero, or reduces its speed accordingly, based on the data from the sensor, i.e., based on the distance of the obstacle from the automobile. On reception, the CRC calculation is repeated, and in case the check values do not match, it can be confirmed that the data received is corrupted. Only if

it is implemented only in expensive luxury cars, mainly for safety-critical applications. Further developments in the protocol is expected to enhance its features, and also turn it more cost-effective. FlexRay meets the performance requirements of the next generation of automobiles. FlexRay networking standard provides new dimensions to the future of automobile electronics, and is an area where more researches and advancements are important.

8. Acknowledgements

We are greatly obliged to Dr. Sundararaman Gopalan, Head of Department of Electronics and Communication Engineering, Amrita School of Engineering, Kollam, for the encouragement he has provided us. We are grateful to Mr. Senthil Murugan, Assistant Professor, Department of Electronics and Communication Engineering, for his valuable guidance throughout. We are deeply indebted to our very supportive councilor, Ms. Poorna SS, Assistant Professor, Department of Electronics and Communication Engineering, for her constructive criticisms and guidance. We also thank our families and friends for their unconditional support.

9. References

1. Gaidhane AN, Khorgade MP. FPGA Implementation Of Serial Peripheral Interface Of Flexray Controller. 13th International Conference on Modelling and Simulation. Cambridge: U K Sim; 2011. p. 128–32.
2. Hande JY, Khanapurkar M, Bajaj P. Approach for VHDL and FPGA implementation of flex-ray controller. 2nd International Conference on Emerging Trends in Engineering and Technology; Nagpur. 2009. p. 397–401.
3. Zhao A. Reliable in-vehicle flexray network scheduler design. Delft: Delft University of Technology; 2011.
4. Paul A, Raju D, Pradeep N, Parvathy G, Abraham SM. Simple automotive application using flexray protocol. Cochin: Cochin University of Science and Technology; 2014.
5. Han K-H, Bae W-S. Proposal and verification of function-based security protocol for vehicular communication system. *Indian Journal of Science and Technology*. 2015 Oct; 8(27). DOI: 10.17485/ijst/2015/v8i27/81018.
6. Murty ASR, Pavani K, Kalyani TD. Automobile SOS system using MEM sensor. *Indian Journal of Science and Technology*. 2016 May; 9(17). DOI: 10.17485/ijst/2016/v9i17/93027.