FlexRay Bus Communication: A Comprehensive Overview and Analysis

Neaz Mahmud
Electronics Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
Neaz.mahmud@stud.hshl.de

2nd Thanas Dushku
Electronics Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
thanas.dushku@stud.hshl.de

3rd Younsuk Choi
Electronics Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
younsuk.choi@stud.hshl.de

***Abstract*—**
***Index Terms*—**

I. Introduction

Modern automotive systems implement complex distributed system technology, and communication protocols play a vital role in networking of different electronic components and systems within the automotive system in order to enable the exchange of data. Nowadays, communication protocols have grown more important. Due to the growing complexity of modern cars, the number of ECUs embedded in modern cars can be nearly 100 and it is expected to increase more in the future.1 Also, it is predicted that the global market for Automotive ECU would grow at a 4.7 Compound Annual Growth Rate. Moreover, due to the ever increasing volume of data required to be transmitted, communication protocol for higher bandwidth is required. One of the recent communication protocols designed to meet todays and future requirements is FlexRay and various aspects of FlexRay, including an application example of this protocol, is discussed in this paper.

FlexRay is a relatively recent proposal for the automotive industry that the FlexRay Consortium, which includes automotive and equipment manufacturers, proposed from 2000 until 2010. After an exhaustive technical analysis of existing networks like CAN, TTCAN, TCN, TTP/C, and Byteflight, it was determined that they did not fully satisfy all of the technical and application requirements in the automotive industry. To be more specific, the objective was to create a communication system that can manage applications at four different levels—high throughput for digital transmission, implementing ”X-by-Wire” type solutions, providing redundancy, and handling all future electronic functions in motor vehicles. FlexRay allows for both synchronous and asynchronous transmission, numerous synchronous part sending slots for a single node, and node behaviour on single, dual, or mixed channels. To recapitulate, the features required for a FlexRay system were to meet communication, topology, security, and application needs and fast error detection and signalling, predictable data transmission, and high throughput are necessary for communication. FlexRay must be able to manage redundant communication for security purposes, prevent collisions for bus access, and keep communication going for as long as possible.

The application must have complete control over all security and network availability choices made, as well as system communication. Different operating modes, including a specific degraded mode, must be offered by FlexRay. Communication with high pass bands, deterministic communication with high pass bands, and redundant, deterministic communication with high pass bands are three groups of applications that are not addressed by CAN or other existing protocols. These requirements allow us to anticipate the deployment of new network hierarchies in embedded FlexRay applications.

II. **How FlexRay works**

A. Architecture

A FlexRay communication system is also known as a FlexRay cluster, a cluster is a collection of electronic control units (ECUs) or nodes connected by a single FlexRay bus. The nodes of a FlexRay cluster communicate with one another via the FlexRay protocol, which enables them to plan their activities and share information.

The nodes of a FlexRay cluster contain components such as engine control units, transmission control units, brake control units, and other equipment. Each node in the vehicle is in charge of a certain duty, and by exchanging information, they may cooperate to carry out a variety of activities.

FlexRay employs a time-triggered and event-triggered communication strategy to make sure that communication between nodes in a cluster is dependable and effective. This enables the transmission of information between nodes at specified intervals and in reaction to particular occurrences, such as when a sensor detects an obstruction in the route of a moving vehicle.

FlexRay clusters give different parts of a vehicle a method to interact and cooperate to carry out challenging tasks. FlexRay contributes to the safe, effective, and dependable operation of contemporary vehicles by providing fast, predictable, and fault-tolerant communication between nodes. [4]

B. Network Topology

FlexRay allows communications that are implemented using topologies of different categories in order to give flexibility in vehicle design as FlexRay communication is not constrained to any particular physical architecture. Passive and active stars, numerous stars with the potential for sub-buses, single channel, dual channels, bus, bus with stubs, and combinations of all of them. Point-to-point connections are just as practical as active star topologies, passive star topologies, and line topologies.

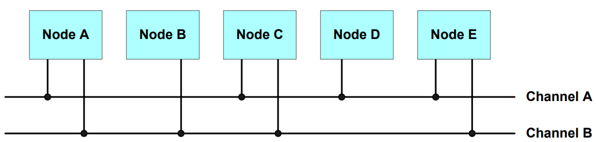

Fig. 1. Passive bus, dual channel [2]

In Fig. 1, a passive dual channel configuration is presented, where the nodes are connected in different ways, either to both channels or to only one, depending on the use case of the node.

FlexRay offers a redundant configuration for the communication channel to reduce the possibility of failure. A data rate of up to 10 Mbit/s may be used for each of the two communication channels. However, it is also possible to use this redundant channel to boost the data rate to 20 Mbit/s. Each communication using FlexRay has the option to select either higher transmission rate or fault tolerance.

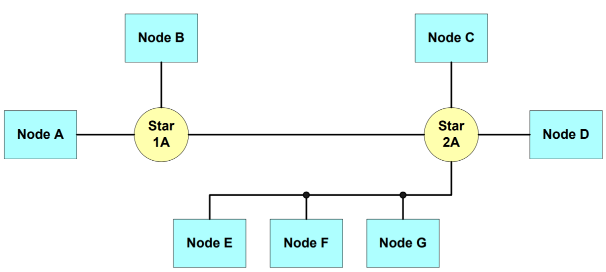

Fig. 2. Active star with passive bus, hybrid topology [2]

There are many other hybrid topologies that might exist; Fig.2 illustrates one form of hybrid topology. In this illustration, a few nodes (nodes A, B, C, and D) are linked to an active star via point-to-point connections. A bus topology is used to connect nodes E, F, and G to one another. Nodes E, F, and G may communicate with one another thanks to the bus' connection to an active star.

C. Frame format

A frame in FlexRay is a unit of data that is sent from one FlexRay bus node to another. Frames, which can include a range of various forms of data, such as sensor readings, control commands, or status updates, are used to transmit information from one node to another.

In order to make sure that communication between nodes is dependable and predictable, each frame in a FlexRay communication cycle is given a specified time slot in which it is broadcast and received. Based on their priority and the scheduling needs of the various system components, the frames are broadcast in a precise order.

FlexRay divides each frame into a hierarchy of static and dynamic pieces that are sent out at various points during the communication cycle. While messages with a more flexible transmission schedule are transmitted using the dynamic segment, messages with a fixed transmission schedule are transmitted using the static segment.

Additionally, FlexRay frames are made to be very flexible, supporting a variety of data types, transmission rates, and priority levels. FlexRay may be employed in a wide range of applications because to its versatility, including industrial automation, safety-critical automotive systems, and more.

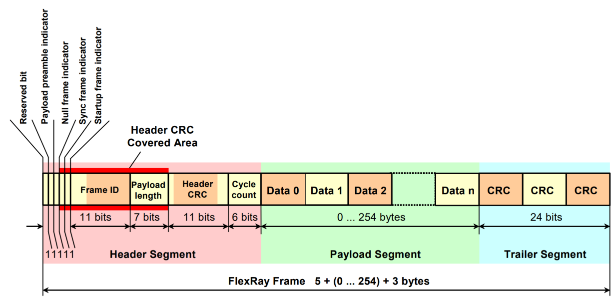

Fig. 3. Frame format [2]

The FlexRay header segment consists of 5 bytes. These bytes contain the reserved bit, the payload preamble indicator, the null frame indicator, the sync frame indicator, the startup frame indicator, the frame ID, the payload length, the header Cyclic Redundancy Code (CRC), and the cycle count. [2]

D. Timing and Synchronization

Every node in a distributed communication system has an own clock. Even though all internal time bases are initially synchronized, over a short period of time the internal time bases of the individual nodes drift due to temperature variations, voltage variations, and manufacture tolerances of the timing source (an oscillator).

The core principle of a time-triggered system is that each node in the cluster approximates the same global view of time, which serves as the foundation for the timing of each node's communications. "Approximately the same" in this context refers to the notion that any two nodes' views of the global time must be within a given tolerance range. The accuracy is the largest value of this difference. By carefully analyzing the time of sync frames provided by other nodes, the FlexRay protocol utilizes a distributed clock synchronization technique in which each node independently synchronizes itself to the cluster. The method is fault-tolerant. [2]

III. Example Application (Adaptive Cruise Control

A. Adaptive Cruise Control

Adaptive Cruise Control (ACC) is an important component of modern Advanced Driver Assistance Systems (ADAS) in vehicles. It is designed to maintain a safe distance between the vehicle and the vehicle in front by automatically adjusting the speed of the vehicle. To enable this functionality, ACC relies on accurate and timely communication between different components of the system. In this section, we will discuss in detail how FlexRay communication is used in ACC.

FlexRay is a communication protocol that was developed specifically for high-speed communication in modern automotive systems.FlexRay provides an expedient platform for the design of automotive control systems due to its high bandwidth and deterministic temporal behavior [6]. In ACC, FlexRay is used to enable communication between different components of the system, including the ACC sensor, ACC controller, and the vehicle's Electronic Control Unit (ECU). The ACC

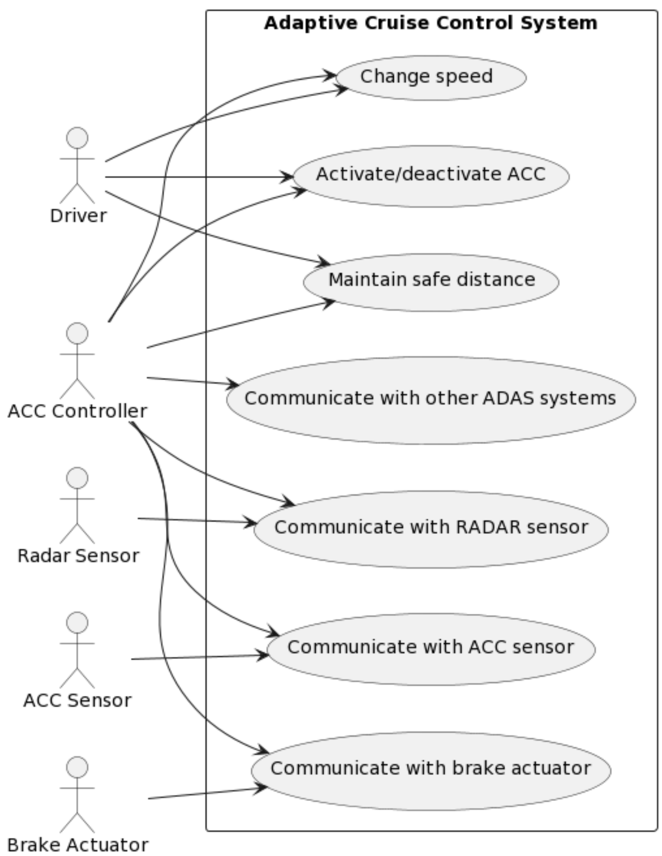

Fig. 4. Use Case

sensor is responsible for detecting the distance and speed of the vehicle in front, and transmitting this information to the ACC controller. The ACC controller uses this information to determine the appropriate speed to maintain a safe distance from the vehicle in front, and sends commands to the vehicle's ECU to adjust the speed accordingly. To enable this communication, the ACC sensor and the ACC controller are connected via a FlexRay bus [6]. The FlexRay bus allows the ACC sensor and

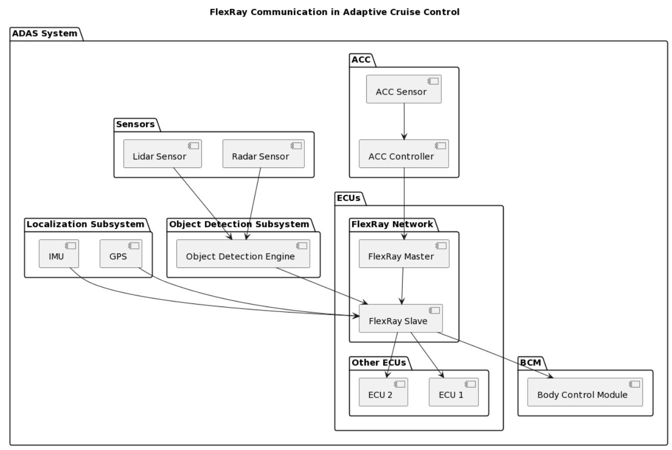

Fig. 5. Block Diagram

the ACC controller to exchange information in a timely and deterministic manner. This is critical for ensuring that the ACC system can respond quickly to changes in the environment and adjust the speed of the vehicle accordingly. Additionally, in order to handle significantly larger bandwidths,which is important for ensuring that the ACC system can accurately detect and respond to different types of obstacles; automotive manufacturers now strive for Ethernet-based systems. However, although Ethernet is cheap, well-known and comes along with many security solutions, it is a generic protocol that does not inherently incorporate the safety features of existing automotive protocols, such as FlexRay. [5] In addition to enabling communication between the ACC sensor and the ACC controller, FlexRay is also used to enable communication between the ACC controller and the vehicle's ECU. The ACC controller sends commands to the ECU to adjust the speed of the vehicle, and receives feedback from the ECU regarding the current speed of the vehicle. This feedback is used to ensure that the ACC system is maintaining the appropriate speed and distance from the vehicle in front. Overall, the use of FlexRay communication in ACC is critical for enabling the accurate and timely exchange of information between different components of the system. It allows the ACC system to respond quickly to changes in the environment and adjust the speed of the vehicle accordingly, which is important for ensuring the safety of the driver and other road users.

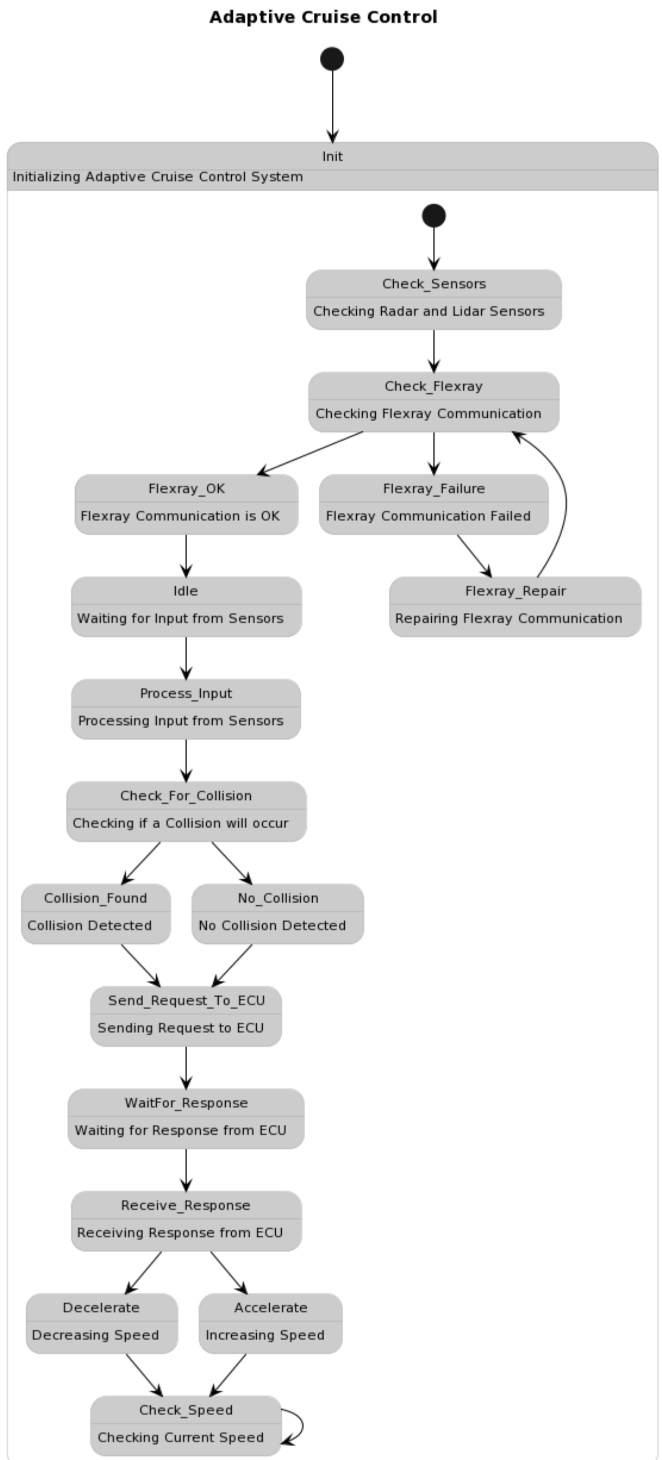

Fig. 6. State Machine

B. Requirements for Communication in the Example Application

When considering the communication requirements for ACC, it is important to consider both functional and non-functional requirements. Functional requirements include the need for reliable and efficient communication between the various sensors and control units involved in the system, as well as the ability to handle large amounts of data in real-time. Non-functional requirements include safety, security, and robustness of the communication system.

One of the primary **functional** requirements for ACC communication is real-time performance. The system must be able to receive and process data from various sensors and actuators in real-time to ensure safe and effective operation. This requires a communication protocol that is capable of high-speed data transfer and low-latency communication.FlexRay is a time-triggered protocol that is specifically designed for real-time applications, offering deterministic and reliable communication through multiple bus access schemes to support both priority-based volume data transfer and deterministic data transfer, thereby addressing the requirements of determinism, bandwidth and reliability. [7]

Another important functional requirement for ACC communication is fault tolerance. The system must be able to detect and handle communication errors or failures in a timely and effective manner to ensure safe operation. FlexRay architecture provides a scalable fault tolerance in terms of dual channel redundancy as well as bus guardian adoption. The bus guardian operates independently from the communication controller, and if a failure occurs in the latter (e.g., an attempt to transmit data in a time slot reserved to another node), it denies the bus access by disabling the bus driver. This way, the communication channel is protected against faulty nodes on the bus. [1]

In addition to these functional requirements, there are also a number of **non-functional** requirements that must be considered when designing the communication system for ACC. Safety is of paramount importance, and the communication protocol must be designed to ensure that the system can operate safely in all conditions. This includes the ability to detect and respond to potential hazards such as obstacles or other vehicles on the road. FlexRay is dedicated to safety and time-critical environments appropriate timing analysis techniques are required to support the safe system design process [3]

Another important non-functional requirement is security. The communication protocol must be designed to prevent unauthorized access or tampering with the system, which could potentially compromise its safety or performance. FlexRay provides a number of security features such as message authentication and encryption, which can help to protect the system against these kinds of threats

Finally, the communication system must be robust and reliable, able to operate effectively in a wide range of environmental conditions and in the presence of various types of interference. FlexRay's robustness is due in part to its use of differential signaling and advanced error detection and correction mechanisms.

IV. Pros and Cons of FlexRay Bus Communication

V. ES Design Methodology

**Definition of ES Design Methodology used in the study **Explanation of state charts and sequence charts for the example application **Steps taken to develop the model use cases using ES Design Methodology

VI. Conclusion

VII. References

References

[1] Federico Baronti, Esa Petri, Sergio Saponara, Luca Fanucci, Roberto Roncella, Roberto Saletti, Paolo D'Abramo, and Riccardo Serventi. Design and verification of hardware building blocks for high-speed and fault-tolerant in-vehicle networks. *IEEE Transactions on Industrial Electronics*, 58(3):793, 2009.

[2] FlexRay Consortium et al. Flexray communications system protocol specification. *Version*, 2(1):198–207, 2005.

[3] Moritz Neukirchner, Mircea Negrean, Rolf Ernst, and Torsten T Bone. Response-time analysis of the flexray dynamic segment under consideration of slot-multiplexing. In *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, page 21. IEEE, 2012.

[4] Dominique Paret. *FlexRay and its applications: real time multiplexed network*. Wiley Online Library, 2012.

[5] Dominik Püllen, Nikolaos Athanasios Anagnostopoulos, Tolga Arul, and Stefan Katzenbeisser. Security and safety co-engineering of the flexray bus in vehicular networks. In *Proceedings of the international conference on omni-layer intelligent systems*, pages 31–37, 2019.

[6] Reinhard Schneider, Dip Goswami, Sohaib Zafar, Martin Lukasiewycz, and Samarjit Chakraborty. Constraint-driven synthesis and tool-support for flexray-based automotive control systems. In *Proceedings of the seventh IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, page 139, 2011.

[7] Shanker Shreejith, Suhaib A Fahmy, and Martin Lukasiewycz. Reconfigurable computing in next-generation automotive networks. *IEEE embedded systems letters*, 5(1):12, 2013.