FlexRay Communication Protocol

1st Oguz Kaan Tuna
Hochschule Hamm-Lippstadt
Electronic Engineering
4th Semester
oguz-kaan.tuna@stud.hshl.de

2nd Kristina Bala
Hochschule Hamm-Lippstadt
Electronic Engineering
4th Semester
kristina.bala@stud.hshl.de

***Abstract*—In 1999, some companies realized that the future requirements cannot be realized with current existing protocols. Therefore they , BMW, Daimler Chrysler, Motorola and Philips, started to cooperate and founded the FlexRay Consortium in 2000 [1]. FlexRay Protocol is a deterministic and fault tolerant communication protocol designed to deliver high data rates and it is intended to be used in automative industry, especially for controling the automative ECU's (Electronic Control Unit). It operates on a time cycle, split in static and dynamic segments, for event-triggered and time-triggered communications [2].**

***Index Terms*—**

I. Introduction

The increasing interaction between the electronic components, which takes more and more the place of mechanical elements, and the electronic systems, demands a complex automative system. Therefore present day vehicles are becoming not just cars but highly complicated distributed real-time systems. Naturally such a system would need a communication stream between its modules, in order to have a stable structure and organization in fields such as safety, reliability or comfort [3].

CAN (Controller Area Network), mainly recognized as the standard communication technology in the automative field, is becoming slowly unable to overcome the challenges of advancing requirements in previously mentioned areas. One of the main reason for this outcome is that CAN is only able to deliver a maximum data rate of 1Mbit/s, alongside as a result of its lack of redundent structure and mechanism, CAN cannot meet the needed requirements for fault tolerancy [4].

Because of the mentioned reasons, BMW and Daimler Chrysler decided to work together in the beginning of this millennia for developing a new communication technology, which should be able to solve these complications, called FlexRay. In the last two decades FlexRay Consortium grew in numbers for its key partners, Premium Associate Members and Associate Members [5]. This paper will highlight the concept of FlexRay, alogside the advantage and disadvantages of the use of FlexRay. In parallel to named topics, the history of FlexRay and comparisons of FlexRay with other communication technologies will also be portrayed.

II. History of FlexRay

The key companies in automative field realized that the systems they used wont meet the requirements of the future. Work on FlexRay began on the basis of byte flight, which was used by BMW in passive safety systems such as airbags [6]. The new bus should also be able to be used with active safety systems and therefore had to be further developed to satisfy the challenges regarding the determinism and fault tolerance accompanied with higher data rates. Therefore the leading companies in this field, BMW and Daimler Chrysler (later joined by Motorola and Philips), started to cooperate in 2000 with the foundation of FlexRay Consortium. After the consortium was established several other big companies joined the consortium as core members. These were Bosh, General Motors and Volkswagen. Besides the core members, there were also Premium Associates (are allowed to vote in consortium votes and to use the technology) and Normal Associates (could use the technology and have acces to information but cannot make any decisions). The core members would use their specific skills for the development. For instance the car manufacturers, BMW, Daimler Chrysler, General Motors and Volkswagen have defined the requirements for FlexRay. While Bosch used its experience from the developments of CAN and TTCAN to the FlexRay protocol. And Philips developed the physical layer specification and implemented early layer silicon solutions [7]. The first vehicle in which the control units were networked via the FlexRay system was in 2006, the BMW X5. After several version of FlexRay were released, the consortium was disbanded in 2010.

III. Charachteristics of FlexRay

FlexRay is a deterministic, fault-tolerant and high-speed type of bus system which delivers the error tolerance and time-determinism performance requirements for x-by-wire applications (i.e. drive-by-wire, steer-by-wire, brake-by-wire, etc.). The main purpose of the design of this communication bus is to keep costs down while still being able to deliver best performance in a uneven environment. To connect nodes together FlexRay uses unshielded twisted pair cabling and supports single- and dual-channel configurations, each of them consisting of one or two pairs of wires. In order to reduce the effects of external noise on the network without expensive shielding, differential signaling is used on each pair of wires. Usually FlexRay nodes also have power and ground wires available to power transceivers and microprocessors.

The one feature of this bus system that makes it fault-tolerant and increases its bandwidth is the dual-channel configuration. As applications increase in complexity and safety requirements, differently from most first generation FlexRay networks, which only utilize one channel for cost purpose, in the future networks will use dual channel configuration.

Termination at the end of the network is something the FlexRay bus requires and it is provided by a resistor connected between the pair of signal wires. The termination should be balanced because too much or too little can break the network and only the end nodes on a multi-drop bus need it. Depending on the specific application, the cabling impedance can vary from 80 and 110 ohms, so the end nodes should be terminated in such way to match this range. Termination is a very important part of the network, as it is one of the most common factors that causes frustration when connecting nodes to a test set up.

IV. Network Topology

When it comes to designing the vehicle's communication network, selecting the right topology plays a very important role as it can help reduce cost, optimize performance and increase reliability for a specific design. FlexRay and likewise networks such as: CAN, LIN; have a different topology or network layout that distinguishes them from other traditional networks like e.g. ethernet. This topology supports multi-drop passive connections as well as active star connections for more complex networks.

A. Multi-drop Bus Topology

Multi-drop bus is one of the topologies that FlexRay is most commonly used, which consists of a single network cable run that connects multiple ECUs together. It is a popular topology used by a lot of OEMs. Each ECU has a branch connected to the core, while the ends of the network are terminated with resistors in order to prevent problems with signal reflections. Compared to CAN, FlexRay operates at higher frequencies, up to 10Mbit/s which can cause signal integrity problems, therefore termination and lay out are very important in order to prevent these problems. The multi-drop format is usually applied when there is a similar type of layout and that is beneficial to simplify installation and reduce wiring.

B. Active Star Topology

The star configuration relies on a central active node and individual links that connect to that. This kind of layout is mostly used when there is a need to run the FlexRay network over longer distances or to make it more reliable by segmenting it in case when a portion of the network fails. The links are not dependent on each other, which means that when one of them is cut, the others are not affected and they continue to work normally. Using the star configuration reduces the amount of exposed wire and can therefore help reducing environmental noise.

C. Hybrid Topology

If both bus and star topologies are combined, another form of topology is created which is called a hybrid one. In this case, besides the core cable run, there are also star connections implemented when needed in the vehicle. This combination can be very powerful because of the advantages of both topologies. Therefore, future FlexRay networks are most likely going to rely on the hybrid topology.

V. Communication Protocol

The FlexRay protocol is a unique time-triggered protocol consisting of three main modules that make the data transfer on the network bus possible: Host Controller; Communication Controller; Transceiver and Bus Guardian. The Host Controller module runs the application software by sending and receiving data to or from the communication controller, which is used as an interface to the bus data and it takes care of transmission and reception of message processing as well as clock synchronization. The last one operates as a separate module itself and it is responsible for the generation of micro-tic (the smallest synchronized time unit), clock synchronization and clock synchronization startup. The FlexRay Transceiver module is responsible for converting from microcontroller voltage level to FlexRay protocol bus voltage level and vice versa

This type of architecture enables deterministic data to arrivee in a predictable time frame (down to the microsecond) as well as CAN-like dynamic event-driven data to handle a large variety of frames. This is accomplished with pre-set communication cycle that provides a pre-defined space for static and dynamic data.

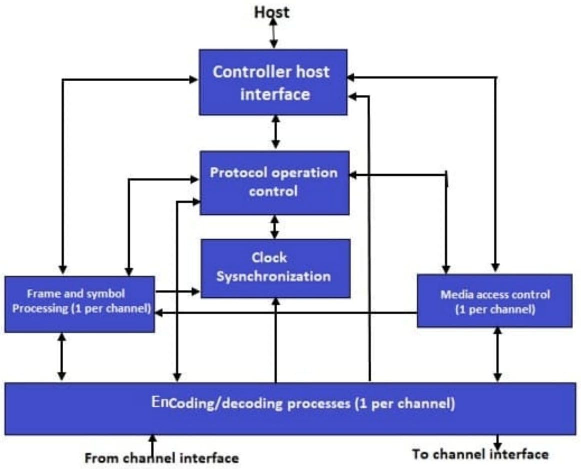
Host
Controller host interface
Protocol operation control
Clock Sysnchronization
Frame and symbol Processing (1 per channel)
Media access control (1 per channel)
EnCoding/decoding processes (1 per channel)
From channel interface
To channel interface

Fig. 1. The Communication Protocol of FlexRay

VI. Communucation Cycle

FlexRay uses a periodical and schedule based data communication. The communication cycle is consisted of at least two segments. These are the Static segment, which is used for deterministic communications and the Network Idle Time (NIT) segment. NIT is applied in order to be able to synchronize local clocks and there is no data tranfering allowed in this segment.

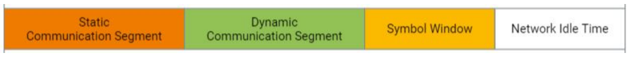

Fig. 2. The Communication Cycle of FlexRay

Additionally the communication can be expanded with including the dynamic time segment as well as a symbol window, Fig.2. The dynamic segment is applied for the event-driven messages and in case it is used , it follows the static segment. The symbol windows is used for sending symbols. Depengind on the symbols, which have been transmitted, a cluster can be woken up or a bus guardian can be tested.

A. Static Segment

The main goal of the static segment is to ensure the real-time data transmission in the system. The static segment uses the Time Division Multiple Acces (TDMA) method to reach this goal. This is an approach, in which the static segment is divided into a number of slots, called static slots. These static slots have the same duration and are appointed to a FlexRay node and when the information exchange proceeds, the FlexRay node has the right to acces the communication bus for transmitting static messages, according to the schedule, more specificly the local counters.Every local counter is increased when the next static slot has begun its own process and each counter value equals to a distinct FlexRay node and static message. Since the sending of the static data occurs periodicly, this method can be called deterministic.

These static slots are divided into 4 parts. These are the action point offset,the message,the channel idle delimeter and the channel idle, Fig.3. The message begins after the action point is reached. It is then followed by the channel idle delimeter. Afterwards there is a pause (channel idle), which corresponds to the beginning of the new slots action point offset. A good example of a static segment use is in airbag deployment, which needs the distinct real-time setting.

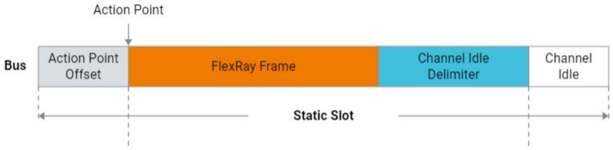

Fig. 3. A Static Slot

B. Dynamic Segment

Dynamic segment is used for delivering event driven messages, meaning it supports asynchronous processes. For asynchronous messages,instead of TDMA the FTDMA(Flexible Time Division Multiple Acces) method is used. Just alike the static segment, the dynamic segment is also broken down into smaller parts, called mini slots. This segment begins the process, after all the FlexRay nodes have incremented their counters. Those counter values equal to the individual dynamic message and node. In case the node has no demand for the dynamic message, which has the same value as the counter, the nodes add the length of a mini slot to their counters. Meaning, if a node wants to transfer data, it has to wait until its mini slot arrives. Unless there is no transmission needed, the mini slot counter is incremented and the next message in the line starts with the transmission. The data will be transmitted in case there is enough time left in the segment. Therefore dynamnic segment uses a prioritization approach, in which the message with the lowest ID has the highest priority. A fitting example for this approach is the wiper in the car.

C. Symbol Window

As previously mentioned, the symbol windows is used for transmitting symbols.Symbols are time-determined low pulses and they have little information content. They indicate an activity within a cluster. The three symbols, which are available in FlexRay, are:

- Collision Avoidance Symbol (CAS)

It is used by the first control unit to inform the node that the cluster has started.

- Media Test Symbol (MTS)

The MTS is sent to test the bus guardian

- Wakeup Symbol

The low phase, followed by an idle phas to wake up the other nodes.

D. Network Idle Time

The Network Idle Time is a phase at the end of a communication cycle in which there is no communication in the cluster. The NIT is used for clock correction, internal controller calculations and data processing.

Comparison with Other Protocols

In order to meet the design challengs, which were created because of the increasing number of electronic comnponents and therefore also control units, some other bus systems were also created. The most important bus systems used in cars alongside the FlexRay are CAN, LIN and MOST.

Since there are various operating sections with different requirements in a car, all the above mentioned bus systems used in certain fields to fulfill the the demanded requirements. Depending on the safety regulations, data rates or costs, the appropriate system is selected and used. For instance, the CAN bus is not the proper system to be selected if the intention is to use it for fast data transmission such as for multimedia applications. Additionally CAN bus is also not suited for low data rates where only a small part of the system is involved, such as heating system.

- MOST (Media Oriented Systems Transport)

As FlexRay , Most bus was realized by BMW and Daimler-Chrysler but for the purpose of being used in multimedia applications, for example the radio or navigation systems. It features a very high data rate,which makes it a very useful component for real-time audio and video utilities. Similarly the MOST bus is able to be arraged in different topoligies,specifically ring, star or chain topoligy.

- LIN (Local Interconnect Network)

Compared to FlexRay, which is used for more complicated functions, LIN however is created for more limited applications, where using the CAN bus would be too expensive. LIN was also created by a consortium with the key members, such as BWM, DaimlerChrysler, Audi and many more. The main application area of LIN are mostly the connection of intelligent actuators and sensors. These include small motors, various sensors such as rain or tempeture or the heating control. Since there is no need for high data rates or fault managemet in this areas, LIN has a much more lower data rate then the other mentioned bus systems. As a result, LIN bus system is very inexpensive compared to FlexRay.

- CAN (Controller Area Network)

The CAN bus was introduced by Bosch in 1983. It was the first standardized protocol for car networks and is still the leading standard today. In contrast to CAN, FlexRay is up to 20 times more faster, but the main difference lies in the fact that FlexRay also allows other topoligies, for example star topoligies.

Acknowledgment

VII. First References

[1](Reichart, 2005)

[2](Vaz et al., 2020).

[3](Steinbach, Korf, & Schmidt, 2010).

[4](Makowitz & Temple, 2006).

[5](Enosh & George, 2014).

[6](Kopetz, 2001)

[7] Flexray Consortium goes for speed, Lenny Case Dec 2006

[8](Rausch, 2007)

[9](Xu, Jang, Kim, Chung, & Lee, 2008)

[10](Shaw & Jackman, 2008)

References

Enosh, S. M., & George, N. S. (2014). An efficient implementation of protocol operation control unit of flexray communication controller. In *2014 first international conference on computational systems and communications (iccsc)* (pp. 256–259).

Kopetz, H. (2001). A comparison of ttp/c and flexray. *Inst. for Computer Eng., Vienna*, *38*.

Makowitz, R., & Temple, C. (2006). Flexray-a communication network for automotive control systems. In *2006 ieee international workshop on factory communication systems* (pp. 207–212).

Rausch, M. (2007). Flexray. *Grundlagen, Funktionsweise, Anwendung. Hanser Fachbuch*.

Reichart, G. (2005). Flexray enabler for future automotive system architectures. *Genf, 3rd March*.

Shaw, R., & Jackman, B. (2008). An introduction to flexray as an industrial network. In *2008 ieee international symposium on industrial electronics* (pp. 1849–1854).

Steinbach, T., Korf, F., & Schmidt, T. C. (2010). Comparing time-triggered ethernet with flexray: An evaluation of competing approaches to real-time for in-vehicle networks. In *2010 ieee international workshop on factory communication systems proceedings* (pp. 199–202).

Vaz, R. M., Hodel, K. N., Santos, M. M., Arruda, B. A., Netto, M. L., & Justo, J. F. (2020). An efficient formulation for optimization of flexray frame scheduling. *Vehicular Communications*, *24*, 100234.

Xu, Y.-N., Jang, I., Kim, Y., Chung, J., & Lee, S.-C. (2008). Implementation of flexray protocol with an automotive application. In *2008 international soc design conference* (Vol. 2, pp. II–25).