An Introduction to FlexRay as an Industrial Network

Robert Shaw, Brendan Jackman
Automotive Control Group,
Waterford Institute of Technology,
Waterford,
Ireland.
E-mail: rshaw@wit.ie, bjackman@wit.ie
Website: http://www.wit.ie/automotive

Abstract
The FlexRay Protocol was developed by the FlexRay consortium which was started when BMW and DaimlerChrysler worked together to create a new network scheme that would suit their current and future needs. They were soon joined by other companies such as Bosch and Phillips [1]. FlexRay is set to become widely used in the automotive industry where it will replace or support existing networking schemes such as CAN. It has already been implemented by BMW in the 2006 X5 [2]. This paper will introduce the key features of the FlexRay protocol and where relevant it will be compared it to CAN and the existing Fieldbus technologies such as PROFIBUS and Foundation Fieldbus.

1. Introduction

The core concept of the FlexRay protocol is a time triggered approach to network communications. This is a different approach to some earlier successful networking schemes. For instance CAN (controller area network) was first developed for use in the automotive industry but was found to be useful in other areas such as industrial control applications [3]. The CAN networking scheme uses a priority driven bus arbitration system. This means that a message with a higher priority message ID will be given access to the network if a lower priority message is also looking for access to the bus. The resulting message transmission delays can lead to problems for safety systems and because of this a TDMA (time division multiple access) method was chosen for the FlexRay protocol [4].

The TDMA scheme used by FlexRay has some similarities to Fieldbus communication systems. With the Foundation Fieldbus a repeating communication cycle determines which node may transmit using a master-slave networking scheme. This ensures that only one node may have access to the physical bus at any one time [5][6][7].

2. FlexRay in Automation

FlexRay is an option for upgrading existing network systems using CAN in the automotive industry as well as other industrial control applications. It could also be used for new applications in industrial automation, where safety and reliability in a work environment is of utmost importance, due to its deterministic approach to communication of the messages. This is helped by the use of a two channel topology where each channel is able to work independently, but the two channels can also be used to communicate the same information and as such has built in redundancy.

The FlexRay protocol has been designed to carry information at a rate of 10Mbits/s over each of its two channels while CAN has a data rate of 1Mbit/s [3][8]. This means that an equivalent data rate of 20Mbits/s can be achieved which is twenty times that of a CAN based system. The high bit rate of FlexRay systems makes it suitable as the basis of a network backbone even where CAN is already in use.

A Fieldbus network has a data rate of around 31.5Kbits/s. When the data rate is compared to that of FlexRay there emerges a view that FlexRay, if used with an existing CAN system, could take on a similar role of the HSE communication role used in Foundation Fieldbus [5][6][7].

3. Network Topology

The FlexRay protocol defines a two channel network, channel A and channel B. A node can be attached to one or both of these channels. If a node is attached to a single channel it does not matter if it is channel A or channel B.

The FlexRay protocol allows for various bus topologies. These can be a point to point connection, passive star, linear passive bus, active star network, cascaded active stars, hybrid topologies and dual channel topologies. The FlexRay protocol will support hybrid topologies as long as the limits of each topology which makes up the hybrid topology (i.e. the star and bus topologies) are not exceeded [8][9]. Fig 1 [8] shows the possible connections in a dual channel configuration. Other possibilities for the network configuration are to have each channel connected in a different way as shown in Fig 2 [8].

978-1-4244-1666-0/08/$25.00 ©2008 IEEE

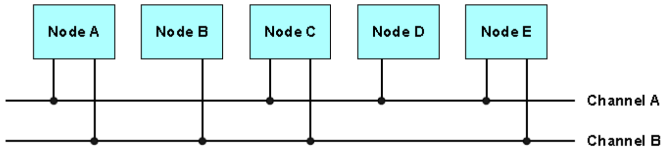

Fig 1. Dual channel configurations

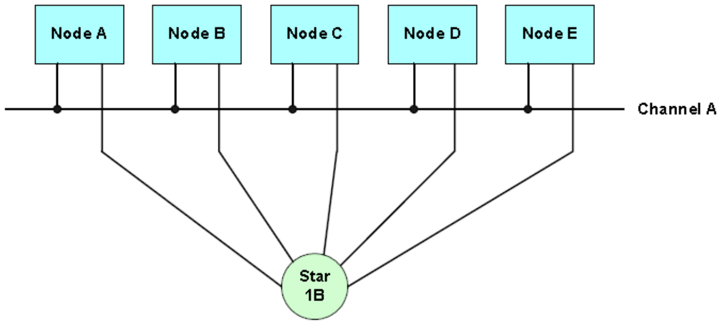

Fig 2. Dual channel implemented as different network types

In Fig 2 channel A is implemented as a bus while channel B is a star topology. The possible combinations of FlexRay topologies make it a very adaptive and flexible system which can be designed to suit various applications.

4. FlexRay Hardware

Each node has a communication controller, a host, a power supply unit and two bus drivers, one for each channel. Fig 3 [8] shows the logical connections of each element.

The host handles the applications of the system while the FlexRay protocol is handled by the communications controller. The bus driver is used to read and write data to the physical medium over which the data is transmitted. In sleep mode it also has the ability to start the wakeup procedure if it detects a wakeup symbol. The communications controller will mainly handle the framing of data and the checking of received data to ensure it was uncorrupted before passing it to the host. The host and communications controller share information such as control information and payload data from the host, while the communication controller relays status information and data received. The host interface to the bus driver allows it to change the operation of the bus driver as well as read status and error flags.

5. Media Access Control

CAN uses a serial bus priority driven networking scheme but allows for a time triggered communications using a higher protocol such as TTCAN [10]. This means that in a basic CAN system if any two nodes wish to transmit data at the same time, the message with the higher priority can transmit while the other message must wait. In contrast the FlexRay protocol uses a TDMA approach and also allows for a node to send frames in a dynamic way. To do this the protocol defines a recurring cycle called the communications cycle. This cycle has the same format and is of the same time length each time it occurs and in the case of FlexRay is divided into four sections: the static segment, dynamic segment, the symbol window and the network idle time. Fig 4 [8] shows a breakdown of the communication into various sections. Each section is then also broken down into its different slots.

The communications protocol of Fieldbus technologies operates in a very similar way to that of FlexRay. However in Fieldbus technologies there is one master node that controls the scheduling of the transmission. In FlexRay each node has its own view of the global time and will only transmit in allocated slots; this means that the network is not dependant on any one node to maintain the communication schedule. The Fieldbus configuration means that the network is liable to fail if the master node fails. In Fieldbus technologies however there is redundancy designed into the devices. This reduces the probability of an entire network failure [5]. In section 6 the method that each FlexRay node uses to determine the current time is outlined.

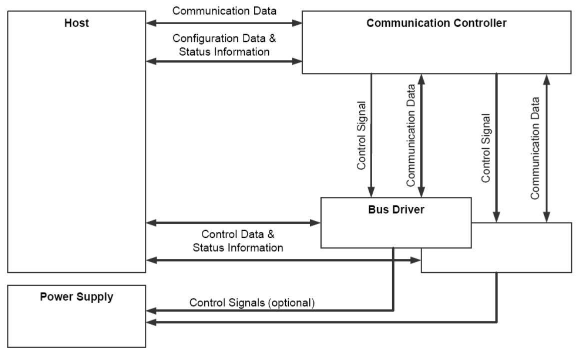

Fig 3. Logical connections

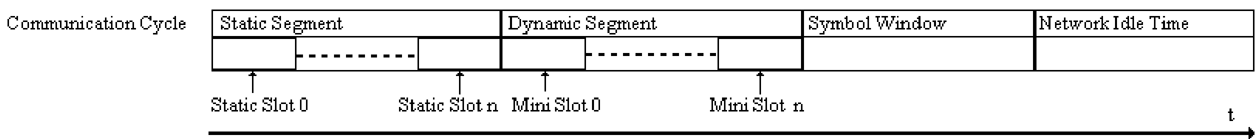

Fig 4. Communication cycle

5.1 TDMA

In a TDMA system the communication cycle is broken down into smaller time segments referred to as slots. The duration of the slots in the static segment are the same. The slots are assigned to a given communication node so that in every communication cycle only that node can transmit at that time. It should be noted that FlexRay does provide a cycle multiplexing system so that information can be sent out every odd cycle for example. This allows another message to be sent in that slot during the even communications cycles and again this message would also be set to that slot in that multiple of the communication cycles. Also a node may get more than one slot per segment depending on the setup of the system and the need to send different messages.

This approach to message arbitration leads to a very deterministic networking scheme making it very suitable for monitoring and safety systems applications.

5.2 Static Segment

The static segment is broken down into smaller sections called static slots. Every static slot is of the same duration. During transmission each slot is assigned to a specific message and only that message can transmit during that slot time.

5.3 Dynamic Segment

The Dynamic segment is an optional section of the communication cycle. It is broken down into smaller sections known as minislots. If a node wishes to communicate it must wait until its minislot comes around. If no transmission occurs after a given period the minislot counter is incremented and the node with the next message/frame id may begin transmission of data. The data to be sent will only be sent if there is enough time left in the dynamic segment. In this way the dynamic segment is priority driven with the message with the lowest ID having the highest priority, just like CAN.

5.4 Symbol Window

A symbol is used to signal a need to wake up a cluster amongst other things. This depends on the symbol sent and the status of the controller at the time. Within the symbol window a single symbol may be sent. If there is more than one symbol to be sent then a higher level protocol must determine which symbol gets priority as the FlexRay protocol provides no arbitration for the symbol window.

5.5 Network Idle Time

The network idle time is used to calculate clock adjustments and correct the node's view of the global time. It also performs communication specific tasks and uses up the remaining time of the communication cycle.

6. Timing

In FlexRay it is important for every node to share the same view of time. This is due to the fact that messages are sent at specific times and so if there is no global view of time then errors can occur.

To achieve a global view of time each node derives, from the oscillator, a value known as a microtick. This value will vary from node to node. The next level of time is called a

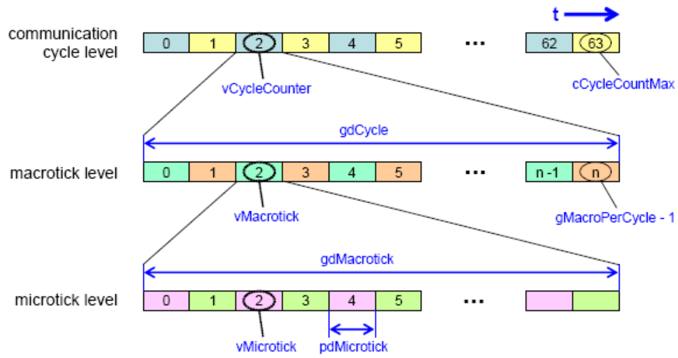

Fig 5. Timing hierarchy

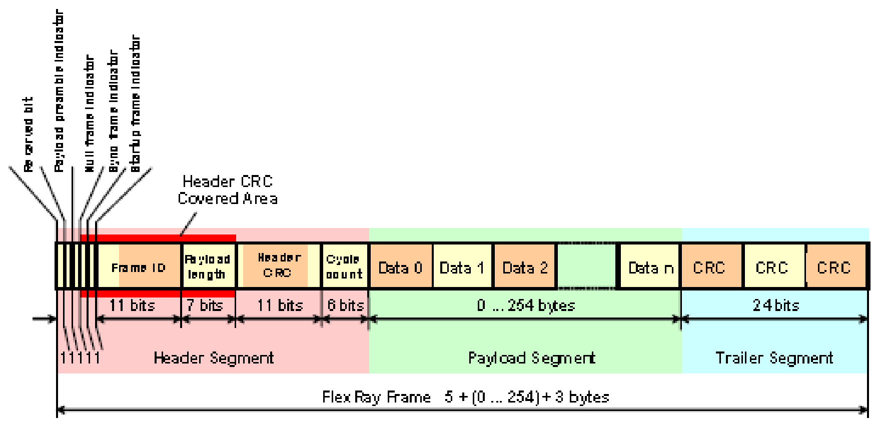

Fig 6. FlexRay frame format

macrotick and is made up of a given number of microticks. This number will vary from node to node so that the duration of a macrotick is the same length throughout the network. Each node will then see the communication cycle in terms of macroticks and this should be the same for all nodes in the network. Fig 5 shows the breakdown of the timing in a FlexRay system.

The nodes in a network will still tend to drift away from one another in terms of their view of the global time. As such the FlexRay protocol defines a clock synchronisation method to keep the networks timing the same within a given tolerance.

7. Frame Format

The frame of a FlexRay message is broken down into 3 sections: the header, payload and trailer section as seen is Fig 6. When compared to the CAN frame format, both standard format and extended format, it can be seen that the frame format of FlexRay is much larger. This is partly due to the extra error checking.

The header section contains status information such as status bits indicating if the fame is a null frame, i.e. contains no payload data, or if the frame should be used for clock synchronisation. There are also bits to indicate the length of the payload transmitted and cyclic redundancy check (CRC) bits so the receiver can determine if the header was received correctly.

The payload contains the data to be transmitted over the network. The payload may also be used to transmit more frame information as an option and this would be indicated in the header of the frame. The payload length can vary from 0 to 254 bytes. When compared to the 0 to 8 bytes in a CAN frame this is a significant improvement.

The trailer section contains a 24 bit CRC that is calculated over the payload and header sections. This is used by the receiving node again to determine if the frame was received without any errors.

8. FlexRay vs. CAN vs. Fieldbus Summary

The Following table, Table I, compares the feature of FlexRay, CAN and Fieldbus technologies. The use of FlexRay systems in automation could be a useful tool. However when designing a communication system the following trade-offs between FlexRay and other communication protocols should be considered; there would be no point in implementing a FlexRay system if CAN would be sufficient. Likewise if a high data rate is required then FlexRay may be a more suitable scheme to use.

Table I
FlexRay vs. CAN vs. Fieldbus

	FlexRay	CAN	Fieldbus
Communication Method	TDMA but allows for a dynamic communication as standard	Dynamic arbitration on the bus, allowing for a time triggered approach using a higher protocol	Master -Slave configuration where a master determines who can transmit data
Data Rate	10Mbits/s on two channels giving a combined total of 20Mbits/s	1Mbit/s	Values in the kbits/s range between slaves but increasing to up to 100MBits/s for applications such as the HSE connection of Foundation Fieldbus
Number of transmission bytes	0-254	0-8	1-244 data bytes with between 9 and 11 control bytes for PROFIBUS-PA
Number of communication channels	2 – second channel can be used as a redundant channel	1	1
Error Checking	11 bit header CRC & 24 bit frame CRC	15 bit CRC	1 byte frame check sequence in PROFIBUS-PA
Determinisim	TDMA leads to deterministic behaviour	No, unless a higher protocol is used	Master -Slave configuration leads to a predictable pattern
Complexity	High	Low	High
Areas of Use	New protocol means less widely used but high levels of research for future applications	CAN widely used in automation and automotive applications	Widely used for automation purposes

9. Automation Examples

I. Automation Example: Environment Control

An example of where a FlexRay system could be used is in environmental control of buildings. This could have an important role in manufacturing companies in the future due to the trend of reducing the reliance on fossil fuels to help reduce pollution along with stricter emission laws. This is clear from the Kyoto agreement where countries from around the world are looking to reduce their emissions to 5% above that of their 1990 levels by 2012 [11]. This is compounded by the need of companies to reduce manufacturing costs and with the increasing price of fossil fuels as these also have an effect on their production costs, a more efficient environmental control system could help reduce overall costs. Reference [12] is a CAN based controller designed to attempt to maximise the efficiency of an air conditioning system and thus to minimise the cost of running such a system.

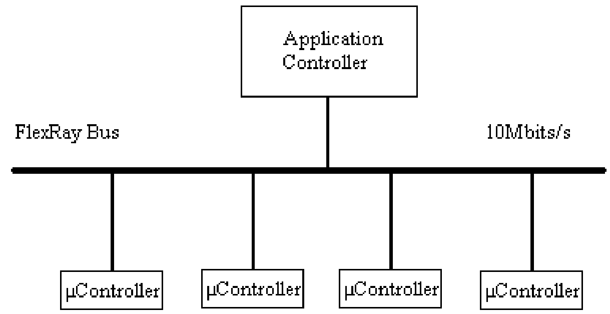

Fig 7. Block diagram for cooler example

This idea can be extended to be used by a FlexRay system. The idea behind this would be a main controller connected to various sensors and actuators over a FlexRay bus. This is illustrated in Figure 7 with the main controller having the ability to receive data from sensors and to send control signals down to the actuators. The diagram illustrates the sensors and actuators as just the microcontroller that would be connected to them.

Due to the deterministic nature of FlexRay this type of system could be set up to monitor or activate the various nodes to increase the efficiency throughout the plant. The sensors placed throughout the plant, both internally and externally, can be used together to determine the best course of action based on internal and external environmental conditions [12]. The high data rate would also mean that more nodes could be serviced in a given time and with bigger possible data payloads information from a given area, such as air temperature and humidity, could be combined into one packet for further efficiency. This could also mean that more sensors could be serviced by a single controller.

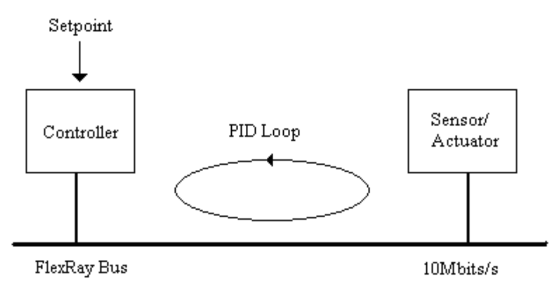

Fig 8. Remote PID loop control

II. Automation Example: Remote PID Control

Example 2 takes the idea of a controller further by using FlexRay to implement a PID controller. Again the high data rates will mean that information will arrive at the controller relatively quickly. The controller will again be able to monitor the incoming data and calculate suitable control signals using a PID algorithm. Again the deterministic nature of FlexRay lends itself to this type of application due to PID needing a regular time base for the calculations and FlexRay only sending data out at predetermined points, e.g. every 3 milliseconds. Figure 8 is a basic block diagram of a FlexRay based PID system. The FlexRay specifications don't specify a particular connection type but usually it is a shielded or unshielded twisted pair with a maximum distance of 24m between nodes [13]. However there is research on-going into the use of fibre optic cabling. This would make it very suitable in some manufacturing applications where noise is an issue or where the environment may become hazardous in the presence of electricity.

10. Conclusion

The FlexRay protocol developed by the FlexRay consortium has already found applications in the automotive industry and looks set to become the network scheme favoured especially in x-by-wire applications and other safety critical systems. There is on-going research into the migration from CAN based systems to FlexRay based systems and as such the protocol could find itself being used in many areas outside the automotive industry. With its deterministic time-triggered approach and the high data rates achievable it is also suitable for safety and control applications. This paper has briefly introduced the FlexRay protocol. The protocol has further defined areas of the network scheme such as frame and symbol coding/decoding as well as start-up of and integration of a node into a network that are detailed in the FlexRay protocol specification.

Acknowledgements

We would like to thank fellow research group members Gareth Leppla and Richard Murphy for their constructive comments and suggestions while writing this paper.

References

[1] FlexRay Consortium (2007) about FlexRay [online], available: http://www.flexray.com/index.php?sid=254188fe2bd59eb7108227f0adea90f5&pid=80&lang=de [accessed 19 Oct 2007].

[2] BMW Manufacturing Co. (2006) THE NEW BMW X5 Perfect Blend of Driving Dynamics, Functionality and Exclusivity [press release], 8 August, available: http://www.bmwusfactory.com/media_center/releases/release.asp?intReleaseNum=209&strYear=2006 [accessed 2 October 2007].

[3] Schofield, M. (2006) Controller Area Network [online], available: http://www.mjschofield.com/index.htm [accessed 30 Oct 2007].

[4] EPN online (2005) Drive-By-Wire: Electronics Vs Mechanical Engineering [online], available: http://www.epn-online.com/page/16409/drive-by-wire--electronics-vs-mechanical-engineering.html [accessed 31 Oct 2007].

[5] PROFIBUS (2007) PROFIBUS PA Technology and Application, System Description, Karlsruhe: PROFIBUS Nutzerorganisation e.V.

[6] Samson (2000) Foundation Fieldbus Technical Information, Part 4 Communication, Frankfurt: Samson AG.

[7] Samson (1999) PROFIBUS-PA Technical Information, Part 4 Communication, Frankfurt: Samson AG.

[8] FlexRay Consortium (2005) FlexRay Communication System Protocol Specification, Version 2.1 Revision A, Stuttgart: FlexRay Consortium GbR.

[9] FlexRay Consortium (2006) FlexRay Communications System Electrical Physical Layer Specification, Version 2.1 Revision B, Stuttgart: FlexRay Consortium GbR.

[10] CiA (2007) Controller Area Network (CAN) [online], available: http://www.can-cia.org/ [accessed 30 Oct 2007].

[11] Airtricity® Holdings Limited (2007) Kyoto Protocol [online], available: http://www.airtricity.com/ireland/environment/kyoto_protocol/ [accessed 19 Novemeber 2007].

[12] Tang, K.H., Sambamoorthy, M., Lin, P.K. and Kandiah, K. (1996) Chiller plant Control and Efficiency Optimisation, CAN Newsletter, September 2006, 8-16.

[13] FlexRay Consortium (2006) FlexRay Communications System Electrical Physical Layer Application Notes, Version 2.1 Revision B, Stuttgart: FlexRay Consortium GbR.