Advance in Electronic and Electric Engineering.
ISSN 2231-1297, Volume 3, Number 2 (2013), pp. 145-154

http://www.ripublication.com/aeee.htm

# Design and Implementation of Flex Ray Communication Protocol in Automotives

**Dibya Prakash**

*Electronics and Communication Engg Dept, Sreenidhi Institute of Science and Technology, Ghatkesar, R.R Dist, Hyderabad, INDIA.*

**Abstract**

Today for a security critical and safety application in automotives, a protocol is required which can transfer the data with a higher speed than existing protocol. As concepts like x-by-wire(x used for drive, steer, brake, etc.,)where steering arms and brake pipes are replaced by electric wire and integrated circuits, a robust and fail-safe communication is needed, and Flex Ray is fulfilling these needs. Failure of these critical systems will cause accident and personal injuries, so a robust and fail-safe communication is needed to avoid these conditions. CAN is a existing protocol in automotives, which is a multi-master broadcast serial bus communication protocol for connecting electronic control units(ECUs).Flex Ray is a newly introduced, communication protocol for automotive control system which is developed to fulfil the increasing demand in automotives for higher safety and comfort. Fault-tolerant property of Flex Ray is mainly highlighted in this protocol so that it can be robust enough to apply for safety critical applications. Flex Ray can overcome the disadvantage of existing CAN protocol. It can be used in automotive industries as well as other industries, where safety and reliability is must, because of its deterministic approach to communication of the message. This is done by the help of two channel topology, where each channel can work independently and can also be used to communicate same information and as such has built in redundancy. In this protocol if one channel fails then other will be available for work, so one channel is always available for backup in safety critical applications. This paper will demonstrate the Flex Ray protocol in automobile and will also explain the design and implementation of electric window to show the communication in Flex Ray protocol.

**Keywords**: Code Warrior; Electric Window; Free Master; MC9S12XDP512; Philips TJA1080.

## 1. Introduction

Flex Ray is a next generation automotive control and communication protocol planed for safety – critical application such as steering-by-wire (SBW) system, where mechanical linkages between the steer wheel and the road wheel will be replaced by electronic connections. Such replacements have the advantage of saving cost, energy and improvement in productivity as well as maintainability. In 1998 BMW and Daimler-Chrysler analysed the available automotive networks (e.g., CAN, TTP, MOST and Byte light) and found that none of those technologies are able to fulfil the future needs of next generation automotive systems, especially when the automotive industry will take the next step towards x-by-wire. As response to this, the Flex Ray Consortium was formed to develop a new protocol. Today basically all car manufacturing company have joined this Consortium and in 2004, the protocol specification made public.

Flex Ray, as the next generation protocol, provides high bandwidth as well as scalable fault tolerance. Fault-tolerant mechanisms (FTMs) have been built into Flex Ray while the this protocol was developed. In SBW system, a large number of signals to be transmitted in very strict latency time requirement and time deviations require not only a fast communication system, but also time-triggered one. For this kind of system Flex Ray is most favourable, because of its time –triggered communication method with additional advantages of deterministic, fault-tolerant and high data rates. In these days almost all the leading automotive, semiconductor and electronic systems manufacturing companies has become the member of the Flex Ray Consortium, who contribute their product solutions and applications for further development in Flex Ray. Electronics was initially introduced in commercially available automobiles in late 1950s.Today number of devices and sensors are increases vastly in automotives. As number of devices and sensors increased, wiring increased vastly, number of wiring can cause the problem like increase in vehicle weight and causing more possible failure points which will affect the reliability of the system. Also complex wiring will take the large volume of vehicle. To overcome these problems following technologies were introduced, they are divided in two groups: (1) current wired and (2) multimedia

Flex Ray is a scalable static and dynamic message transmission (deterministic and flexible) protocol, which has a high net data rate of 10Mbit/sec; gross data rate approximately 20Mbit/sec. It is scalable fault-tolerance (single and dual channel).Flex Ray has two channels (A and B) which can be configured separately. This allows data to be transmitted on one channel without the other being used, to save bandwidth. For safety critical application data will be sending simultaneously to both the channel, if one channel fails other will be available for desire work. It shows that it has one extra channel to work as a backup at the time of any failure. Data is transmitted on bus in

both manner time and event driven. Each message is divided in to two sections one is static segment and other is dynamic segment.

## 2. Frame Format

Frame format of a Flex Ray is divided in to three sections; the header, payload and trailer section. In compare to CAN format, Flex Ray has a longer frame format because of extra error checking. Frame format of Flex Ray is shown below in fig;

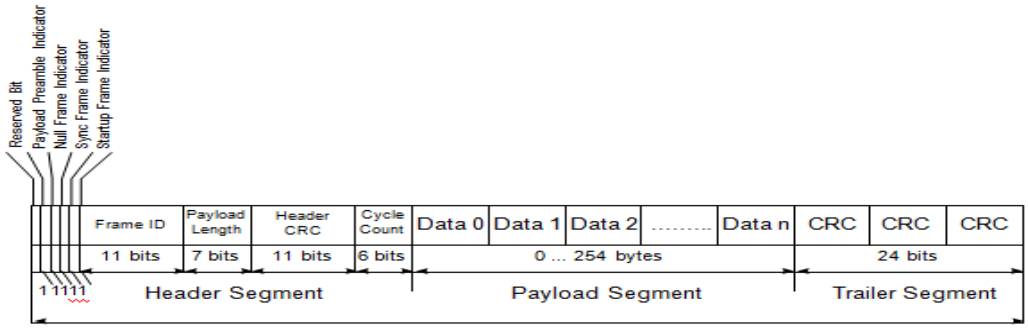


**Figure1**: Frame format of Flex Ray.

In frame format of Flex Ray header, payload and trailer segment is of 35bits, (0-254) byte and 24 bits respectively. The first five bits (shown in the left most side) define the basic features of the frame.

**Header section –**

Frame ID is of 11 bits define the slot position in the static segment and in dynamic segment it defines the priority of the frame, a lower identifier shows higher priority. Payload Length (7 bits) is defined as the data length. Header CRC (11bits) is a Cyclic Redundancy Check, so the receiver can determine if the header was received correctly. Cycle

Count (6 bits) defined locally in the node and this is the serial number of the frame.

**Payload section-**

Payload contains data to be transmitted over the network and it may vary from 0 to 254 byte. In compare to CAN where it is of 0 to 8 byte, it is a significant improvement.

**Trailer section-**

This section is of 24 bit CRC which is used by the receiving node to ensure that the frame is received without error.

## 3. Implementation of Flex Ray

This paper involved the implementation of a gateway between a CAN and Flex Ray network, thus a processor with the capability to handle these protocols needed to be

selected. While a number of manufacturers offered processors which had the capability to run a Flex Ray and CAN network. The processors to be used needed to meet the requirements of the project specification, and also to be able to operate in the harsh automotive environment.

In this paper Softec Microsystems - SK-S12XDP512-A Development Board containing Freescale HCS12X Processor has chosen. The MC9S12XDP512 has a built in CAN module, this means a separate IC is needed to convert the necessary information to the Flex Ray protocol. Separate IC which is used for this purpose is free scale MFR 4310 communication controller. This controller will communicate with Philips TJA1080, to transmit / receive messages on the Flex Ray bus. The MC9S12XD family will keep cost and power consumption low, It is base around an enhanced S12 core , so it will deliver almost 5 times the performance of a 25-MHz S12 while retaining a high degree of pin and code compatibility with the S12. The MC9S12XD family introduces XGATE module to boost the performance, using enhanced DMA functionality this module offloads the CPU by providing high speed data processing between peripheral modules, RAM, Flash EEPROM and I/O ports. The MC9S12XD family is composed of standard on-chip peripherals including up to 512 Kilobytes of Flash EEPROM, 32 Kilobytes of RAM, 4 Kilobytes of EEPROM, 6 asynchronous serial communications interfaces(SCI), three serial peripheral interfaces (SPI), an 8-channel IC/OC enhanced capture timer, an 8-channel,10-bit ADC, a 16-channel, 10-bit ADC, an 8-channelPulse-width modulator (PWM), five CAN 2.0 A,B Software compatible modules (MSCAN12), two Inter-IC bus blocks, and a periodic interrupt timer.

The MFR4310 stand-alone Flex Ray Controller eases the integration of Flex Ray Communication into applications using MCUs without an embedded Flex Ray controller. The MFR4310 controller is optimized for Communicating directly with free scale's S12X Family of 16-bit MCUs, and the MPC500 and MPC5500 families of 32-bit MCUs.

Initially a test program is created to verify the operation of the Flex Ray hardware before it is further developed to be used in a gateway application. The function of this first Flex Ray software application was to periodically transmit a value from Node 2 onto the Flex Ray bus using Slot 1 in the static segment, receive this message on Node 1 where it is incremented, and return the edited value back onto the bus using Slot 4 in the static segment. From there it is received again by Node 2.The Flex Ray network was configured with Macrotcick length (1_S),Communication cycle length(5000MT),Static segment(3000MT),Number of static slot(60),Static slot length(50MT),Dynamic segment(880MT),Maximum number of Minislots(22), Minislots length(40MT).

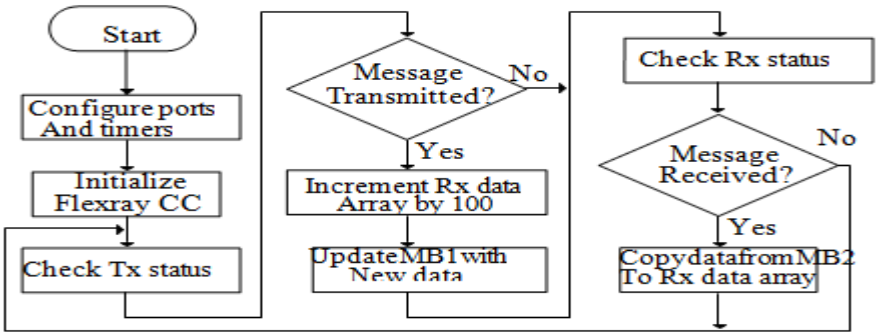


**Figure 2**: Flowchart of Flex Ray Node 1 Software Operation.

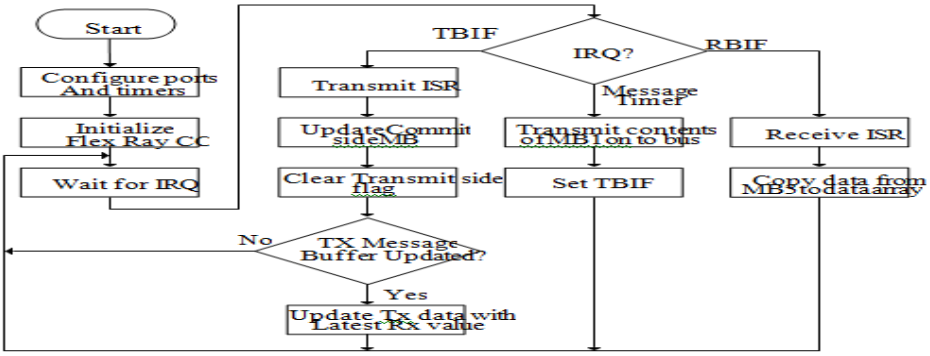


**Figure 3**: Flowchart of Flex Ray Node 2 Software Operation.

## 4. Implementation Testing

A software GUI tool called FreeMaster is used to monitor activity in flex ray bus. FreeMaster is a user-friendly real-time debug monitor and data visualization tool that you can use for any application development and information management. FreeMaster supports completely non-intrusive monitoring of variables on a running system. As well, FreeMaster supports additional capabilities and targets with an on-target driver for transmitting data from the target to the host computer. From the host PC running FreeMaster user can also control the embedded application running on the target. User can modify variables while the target application is running, as well as stop and start execution. FreeMaster is a truly open-ended tool.

To set an application, it is required to place a driver in the embedded code which allows the GUI access to the registers that are to be monitored. The information is send by the MC9S12XDP512 via a serial communication interface link to an RS232 transceiver, and then transmitted to the Serial Port of the PC. The GUI is also need to configure like that so it can take data from registers within MC9S12XDP512.once this is completed one can manipulate the received data on the screen at the form of graph, message or number. For this test application it was required to read the values

transmitted and received on the Flex Ray bus by Node 2. It was expected to see a difference of 100 between the data received and the data transmitted; this would show that the data has in fact been incremented as it passed through Node 1. The output observed on the screen was of two saw tooth waveforms, these were caused by the variables incrementing from 0 up to 65535 and once they reached this maximum value the data rolls over to zero and begins again. When zoomed out from the signal it is quite difficult to see the difference between the two signal traces, but when zoomed in as in Figure 4.11 we can see, as expected, that the received value is the transmitted value but offset by 100. This can also be seen in the numerical representation of the data on the bottom of the screen. Graphical based output after implementing the Flex Ray protocol by use of two nodes is shown below, here one node is developed as ECU node and other is as electric window. We can replace electric window node application with any other application of automotives as well as we can add any number of application nodes to the Flex Ray bus.

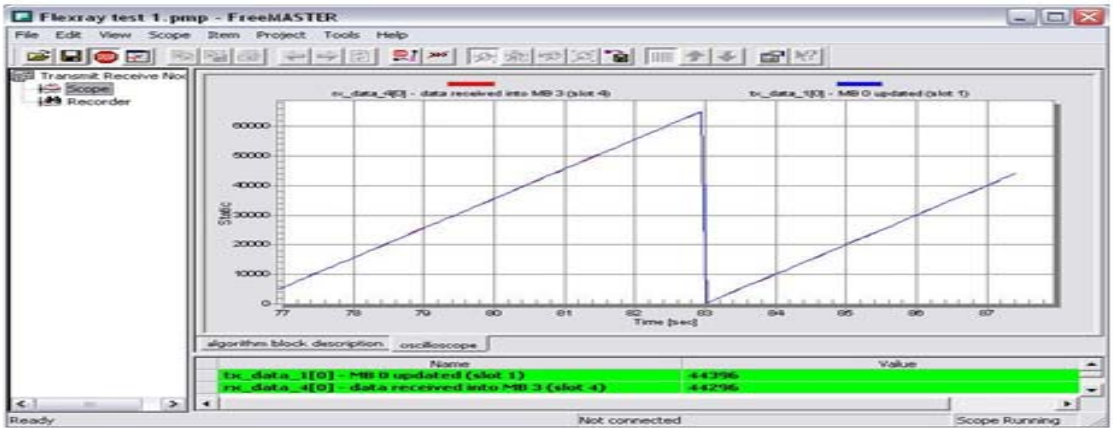


**Figure 4**: Free MASTER Screen Output.

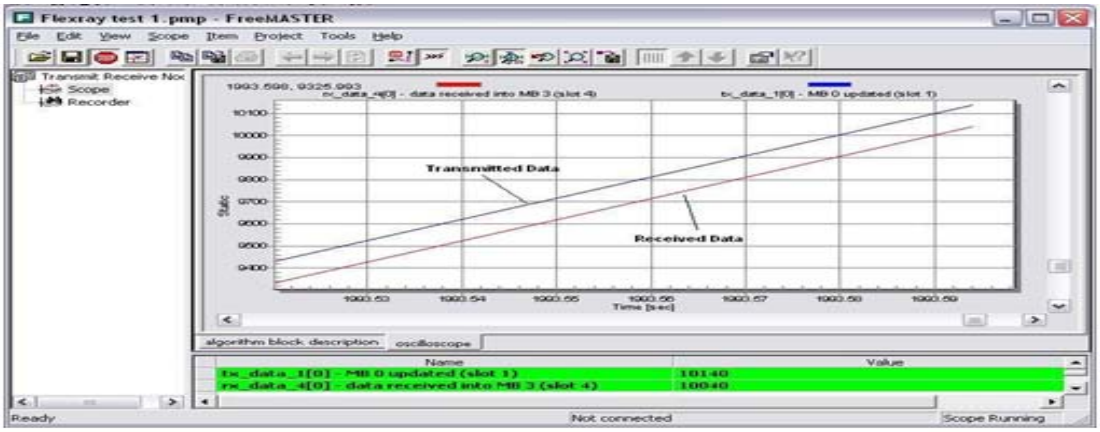


**Figure 5**: Free MASTER Screen Output (time scaled).

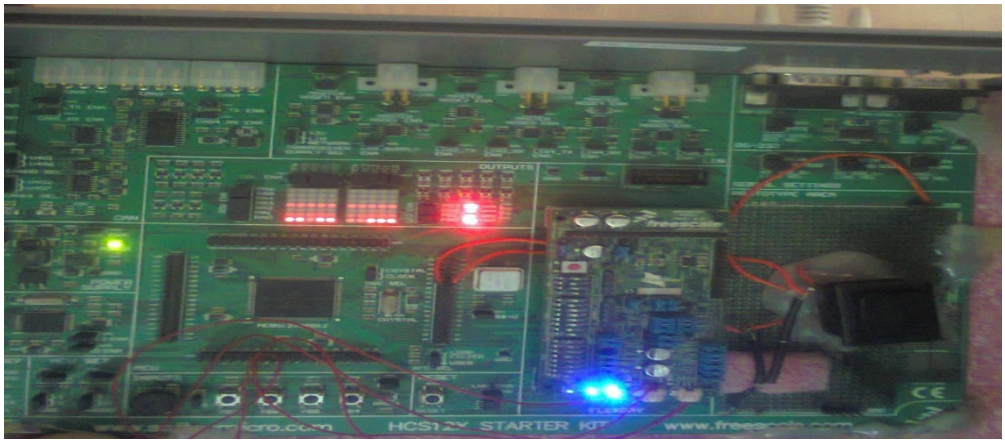


**Figure 6**: MC9S12XDP512 kit configured as ECU node.

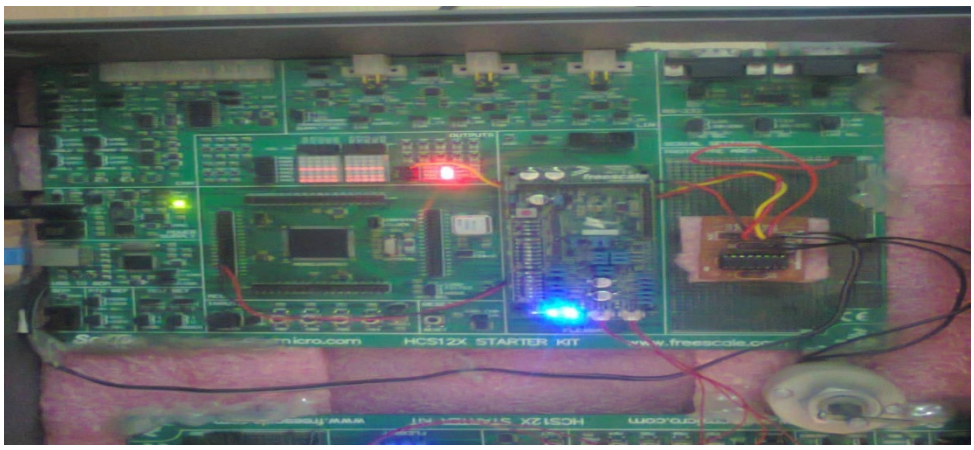


**Figure 7**: MC9S12XDP512 kit configured as electric window node.

**Figure 8**: Interfacing ECU and electric window node.

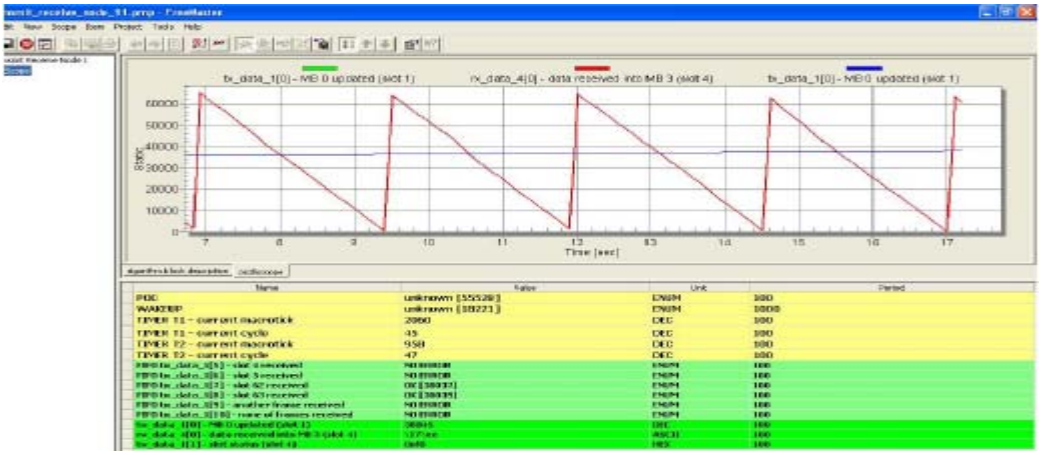

**Figure 9**: Data received at electric window node without switch press.

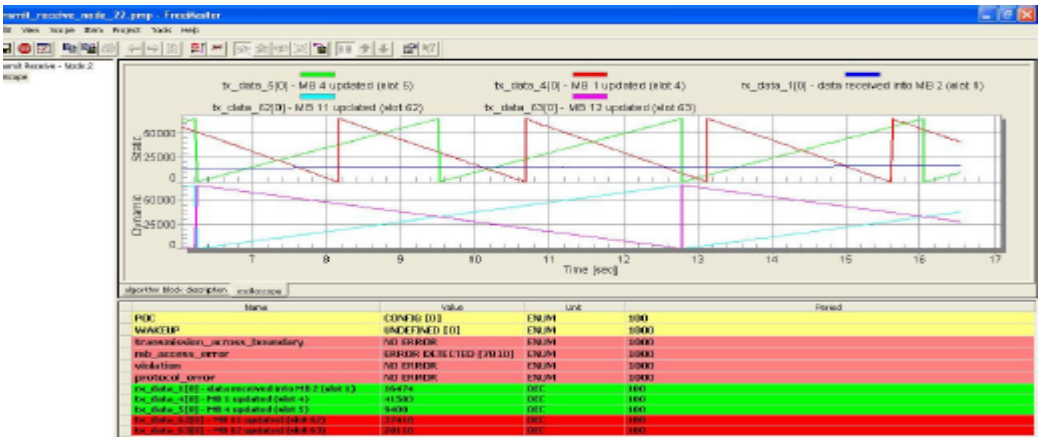

**Figure 10**: Data received at ECU node without switch press.

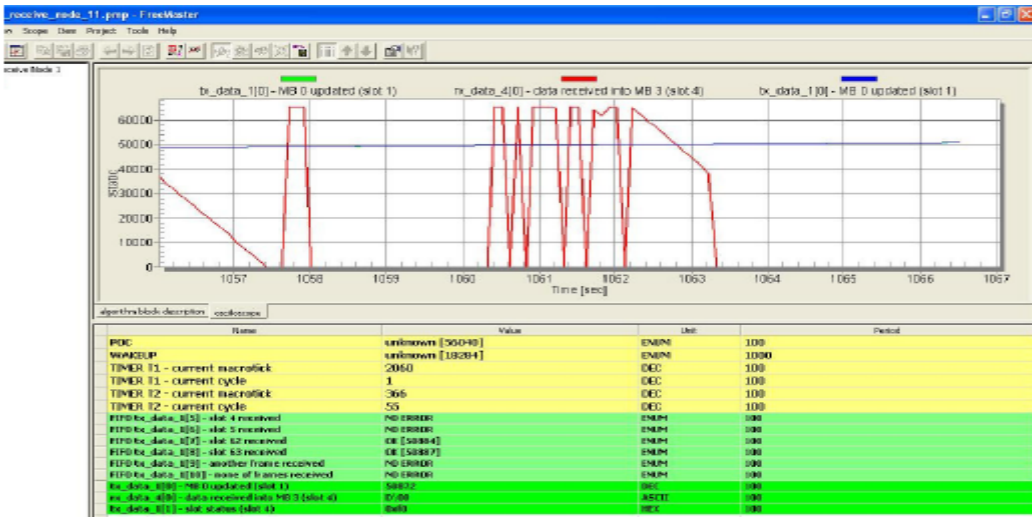

**Figure 11**: Data received at electric window node with switch press (UP).

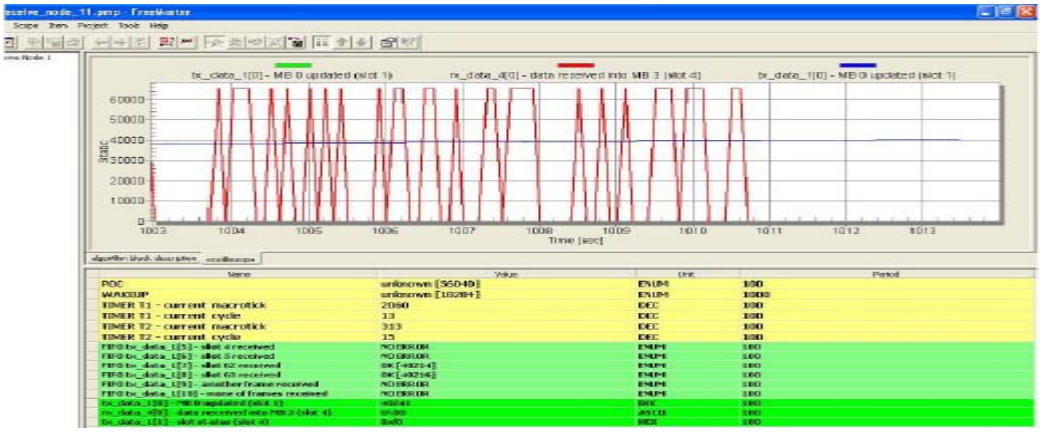

**Figure 12**: Data received at electric window node with switch press (DOWN).

## 5. Conclusions

Due to the increase in electronics requirements in modern vehicles, the CAN protocol is starting to reach its operational limits. Flex Ray is set to become the standard for communication for more advanced networking applications such as drive by wire. As CAN will still be used for body electronics, dash panel and engine management networking etc., inter-protocol gateways will be required to allow communication between these protocols. The aim of this paper was to investigate an inter-protocol gateway for Communication between a Flex Ray networks. The paper described a framework for the implementation of such a system, with the aim of reducing CPU loading by the gateway, and maximizing edibility of implementation. The real application is also explained in this paper which is described as the interfacing between two nodes namely ECU and electric window. Electric window can be replaced by any number of node applications.

This paper is using HSC12X processor for the implementation of the system and FREE Master GUI is developed in such a way that it will show the waveform to represent the data flow values and also same is displayed on the GUI. Figure.11 and figure.12 represents the waveform of data received on electric window when UP switch and down switch pressed respectively.

## 6. Acknowledgements

On the completion of my paper, like to thank FREESCALE SEMICONDUCTOR for availability of data sheet and device user guide on their site.

## References

[1] Flex Ray Communications System – Protocol Specification, v2.1 Revision A, Flex Ray Consortium, Dec. 2005.

[2] Flex Ray Consortium, http:// www.flexray.com.
[3] MC9S12XDP512 Data Sheet, FREESCALE SEMICONDUCTOR, 2010
[4] MC9S12XDP512 DEVICE USER GUIDE, FREESCALE SEMICONDUCTOR, 2010
[5] IEEE Paper on Automotive Communications - Past, Current and Future by Thomas Noltey, Hans Hanssony and Lucia Lo Belloz.