XX IMEKO World Congress
Metrology for Green Growth
September 9–14, 2012, Busan, Republic of Korea

METHODS FOR MEASUREMENT OF FLEXRAY NODE BASIC TIMING PARAMETERS

Jan Sobotka, Jiří Novák

Czech Technical University, Faculty of Electrical Engineering, Technická 2, 166 27 Prague 6, Czech Republic
jan.sobotka@fel.cvut.cz, jnovak@fel.cvut.cz

Abstract: For FlexRay fault free communication it is necessary to ensure correct parameterization of each network node. This paper describes methods for evaluation of basic internal timing parameters of a single FlexRay node. Methods reflect common situation in automotive industry, where each electronic control unit comes from different manufacturer. No direct access to ECUs actual parameters is usually available, thus the independent methods to validate specific parameters are needed.

Keywords: FlexRay; parameters; synchronization; microtick

1. INTRODUCTION

Automotive distributed systems based on FlexRay [1] communication standard, which is described in section below, are very complex (in terms of number of configuration parameters). Protocol robustness [2] can be threatened by wrong parameterization of individual FlexRay node. Intent of the paper is to introduce methods for measurement of the parameters related to the FlexRay node synchronization mechanism. The parameters and their ranges permitted by specification [1] are summarized in table 1.

Table 1 List of measured parameters

Parameter	**Range**
pdMicrotick (µT)	12.5 ns, 25 ns, 50 ns
pClusterDriftDamping	0 - 10 µT
pOffsetCorrectionOut	15 - 16082 µT
pRateCorrectionOut	3 - 3846 µT

2. FLEXRAY COMMUNICATION

In this section the FlexRay standard is briefly described, especially the synchronization mechanism related to presented work. The standard was established by FlexRay consortium in 1999. FlexRay is primarily intended as a communication standard for automotive x-by-wire applications, but today it is mostly used for active chassis control systems or drivetrain control in higher class vehicles [3]. Structure of a node can be divided into 3 parts corresponding to the three protocol layers of ISO/OSI communication model.

- Physical layer is implemented by a FlexRay transceiver. There are up to two FlexRay transceivers in the node, as FlexRay supports two channels (A and B) structure.
- Data link layer is represented with the communication controller implementing FlexRay communication protocol described by the standard.
- Application layer protocols are implemented in host. The host provides two main functions. It configures the communication controller with required setting associated with communication parameters (physical and data link layer parameters) first. Second, the application software is started.

Physical layer uses differential signalling. Nodes are usually connected by twisted pair cable. Passive linear network, passive star, active star (and their combination) network topologies are supported. Maximum communication speed defined by the standard is 10 Mbps.

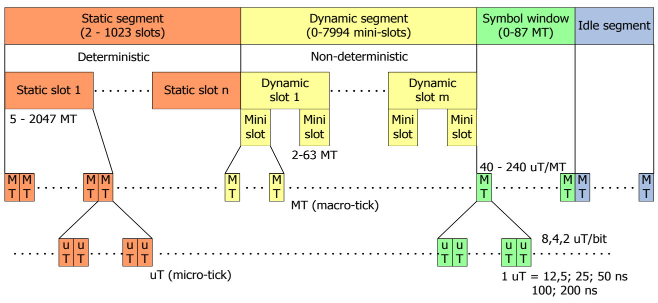

Fig. 1 Structure of a FlexRay communication cycle

Data link layer communication runs in communication cycles, each consisting of up to four segments. The mandatory static segment comprises a fixed number of equally sized TDMA static slots. The segment is designed for communication where predictable frame delay is required [4]. In each static slot a frame (with a unique identifier) can be transmitted by a node. Once the static slot

This research is supported by the Czech Science Foundation under the project SYNERGY - Mobile Sensoric Systems and Network 102/09/H081.

is assigned to a specific node, only this node is allowed to transmit frame within this slot. The optional dynamic segment employs the flexible TDMA approach. It is intended for the non-critical, sporadic event-driven messages with varying length. The length of a dynamic slot is flexibly adapted to the length of data in a particular data frame. In addition, the length of dynamic segment is constant. The optional symbol window segment is designed for transmission of so-called symbols - arbitrary sequences of bits that do not respect the frame format. The mandatory idle segment is used to isolate two successive communication cycles (see Fig. 1).

Data among nodes are exchanged by means of data link frames. The frame consists of three parts - the header field, the payload field, and the CRC field (see Fig. 2).

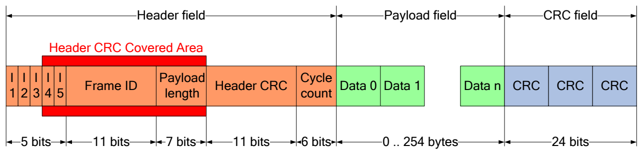

Fig. 2 FlexRay frame format

3. FLEXRAY SYNCHRONIZATION MECHANISM

The communication timing is based on two time units:

- **Micro-tick (μT)**
  Micro-tick is the smallest indivisible time unit in the FlexRay protocol. The unit is directly derived from the time base in the node (e.g. local oscillator). For example, if frequency of clock signal is 80 MHz, 1 μT corresponds to 12.5 ns.
- **Macro-tick (MT)**
  Macro-tick is a longer time unit consisting of integer number of μT. One MT is a global time unit in network. The duration of communication cycle as well as all its parts is defined in integer number of MTs. The nominal values must be the same in each node of the network. On the other hand, the μT is a local time unit (derived from local oscillator) and can be of different length in particular nodes. Consequently, the number of μT falling into single MT is different within nodes.

Time synchronization is based on measurement of incoming frames arrival time. FlexRay network consists of two types of node. First type nodes send synchronization messages while second type nodes only receive synchronization messages. Synchronization message is a standard message with sync frame attribute (a bit in header field, see Fig. 2) set. In any network there must be at least two synchronization nodes (sending synchronization messages). Each node measures deviations between the actual and expected times of synchronization message arrivals in every communication cycle. Afterwards a deviation from global time domain is determined using Fault-Tolerant-Midpoint algorithm (FTM) [5]. This value is further used for calculation of local time correction. FlexRay synchronization mechanism (SYM) uses two types of correction – offset and rate corrections (OC and RC). Offset correction is equal to value calculated by FTM algorithm and applied at the end of each odd communication cycle. Rate correction is calculated as difference between deviations in two following communication cycles (even and odd). The value is added to last rate correction value and applied in each communication cycle, but updated only in even communication cycles.

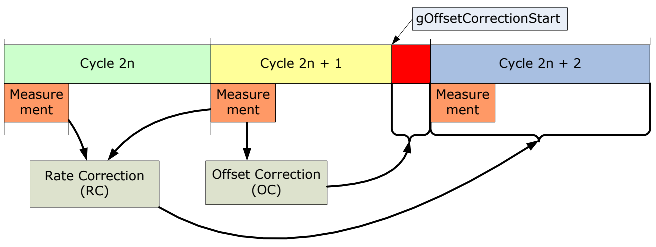

Fig. 3 Principle of synchronization mechanism in FlexRay

Behaviour of FlexRay SYM was modelled by analytic model presented in [6]. Implementation of derived model was used for development of presented methods.

Definition of related terms:
pOffsetCorrectionOut **[μT]** Maximum permissible offset correction
pRateCorrectionOut **[μT]** Maximum permissible rate correction
pClusterDriftDamping **[μT]** Damping factor is used for the rate correction. The computed RC parameter is always changed by *pClusterDriftDamping* factor towards zero. If absolute value of computed RC parameter is lower than *pClusterDriftDamping*, resulting RC value is zero.

4. MEASUREMENT ISSUE IDENTIFICATION

Proposed measurement methods are based on following presumptions:

FlexRay network:
- Consist only of tester and tested node. More complex network topology is possible, but due to Fault-Tolerant-Midpoint algorithm FTM tester node needs to send more synchronization messages (up to four).

ECU under test (EUT):
- Sync or Non-sync node with minimal one assigned static slot (unit periodically sends frames in defined time).
- No direct access to actual SYM parameters of the node

Tester node (TN):
- Two logical controllers (sending synchronization frames in two static slots)
- Able to send tightly shifted synchronization frames
- Precise measurement of incoming frames arrival time

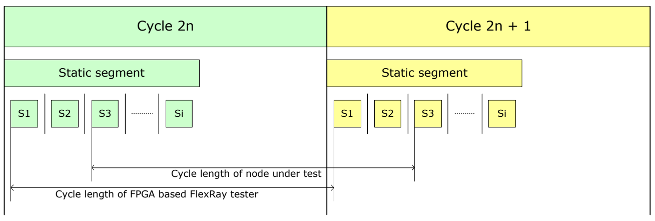
Cycle 2n
Cycle 2n + 1
Static segment
Static segment
S1
S2
S3
Si
S1
S2
S3
Si
Cycle length of node under test
Cycle length of FPGA based FlexRay tester

Fig. 4 Principle of cycle length measurement

5. MESUREMENT METHODS DESCRIPTION

Evaluation of tested ECU behaviour is possible only by bus traffic timing observation. On the other hand, it is possible to affect its synchronization mechanism behaviour by modification of synchronization message transmission times (sent by the tester node).

***pdMicrotick* time evaluation**
According to the FlexRay protocol specification possible nominal values are 12.5 ns, 25 ns and 50 ns. First idea could be to try to lightly push the SYM to make correction exactly 1 µT. It is all right, but the simplest solution is to use a FlexRay network natural behavior. Situation is described by following facts. FlexRay communication is running - tester and tested ECU are synchronized together. Due to Fault-Tolerant-Midpoint algorithm tested EUT adapts to tester node timing (tester sends two synchronization frames). A difference between oscillator frequencies f_{TN} and f_{EUT} in particular communication cycle *i* is expressed by EUT rate and offset corrections (RC_i and OC_i). If we accept, that in real case it is practically impossible, that frequency difference can be expressed as integer number of µT, the offset correction by 1 µT could be expected in each n-th odd communication cycle. Evaluation of *pdMicrotick* is possible by measurement of EUT communication cycle duration in sufficient number of communication cycles. Value of *pdMicrotick* is equal to minimal distance in histogram of communication cycle lengths.

Offset correction (OC)
For presented methods actual value of offset correction parameter sometimes is needed to know. This value can be extracted from length of two following communication cycles (presumption is short term oscillator stability – constant rate correction) – published in [7].

$$OC_{EUT}(2n+1) = cl_{EUT}(2n+2) - cl_{EUT}(2n+1) \qquad (1)$$

pClusterDriftDamping
Let’s the difference between cycle lengths of nodes TN and EUT is equal or greater than $pClusterDriftDamping_{EUT}$ times $pdMicrotick_{EUT}$. Value of offset correction in each odd communication cycle is equal to or greater than 2 × *pClusterDriftDamping*. Value of *pClusterDriftDamping* is equal to minimal measured OC divided by 2.

pRateCorrectionOut
Communication cycle (CC) length with the rate correction can be measured by measurement duration of each even communication cycle (offset correction takes place in odd communication cycles only). Sending synchronization frames with increasing advance or delay causes that RC reaches its limit (even communication cycle duration stops shortening or extending). Pushing the rate correction in opposite direction gives the second limit of communication cycle length.

$$pRateCorrectionOut = \frac{\max CClength - \min CClength}{2 \times pdMicrotick} \quad (2)$$

pOffsetCorrectionOut
For this parameter evaluation step change (tester add an offset) of synchronization frames arrival time is used. Shift in an odd cycle caused, that EUTs SYM immediately change OC value according to measured deviations. In ideal case, when OC before push of SYM was zero microticks, OC in following even cycle will be equal to added offset in microticks plus particular amount of RC. RC is affected by step change too and it is changed by same value like OC minus *pClusterDriftDamping*.
Three possible scenarios can cause, that EUT protocol operation control automata reaches halt state. First case is that calculated OC is greater than *pOffsetCorrectionOut*. Second case is that calculated RC is greater than *pRateCorrectionOut* (*pRateCorrectionOut* is smaller than *pOffsetCorrectionOut*). Last case happens when synchronization frame does not come within assigned static slot boundaries. Consequence of the last situation is that no valid synchronization frames are received and EUT goes to halt state.
Last paragraph gives limitation of the measurement method. Actual value of *pOffsetCorrectionOut* is testable only for suitable parameterization of FlexRay cluster (it is not guaranteed by permitted ranges of parameters by FlexRay specification). In the following section an example of EUT reaction to synchronization frames shift is described.

6. SIMULATION RESULTS

Designed methods were tested by simulation using the derived analytic model of FlexRay SYM [6]. Basic configuration parameters are in table 2. ECU under test nominal clock frequency is 80 MHz. Tester node affects the EUT synchronization mechanism by changing its frequency (duration of microtick). Modification of communication cycle length by adding or removing microticks is also possible.

Table 2 Node parameters

Nominal Clock frequency	80.00 MHz
gdStaticSlot	50
gMacroPerCycle	5000
pMicroPerCycle	200000
gOffsetCorrectionStart	4920
pRateCorrectionOut	600
pOffsetCorrectionOut	1201

First simulation results are depicted in figures 5 and 6. The simulation is intended to evaluate method for *pClusterDriftDamping* parameter measurement. Tester statically changes frequency with step of 400 Hz (cycle is lengthened by 1 μT). The results fully correspond with presumption that frequency difference smaller than *pClusterDriftDamping* have to be adjusted by offset correction.

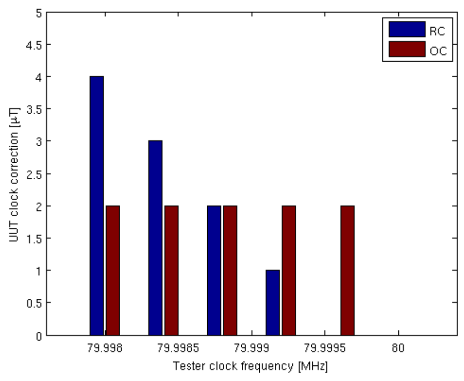

Fig. 5 EUT *pClusterDriftDamping* = 1 μT

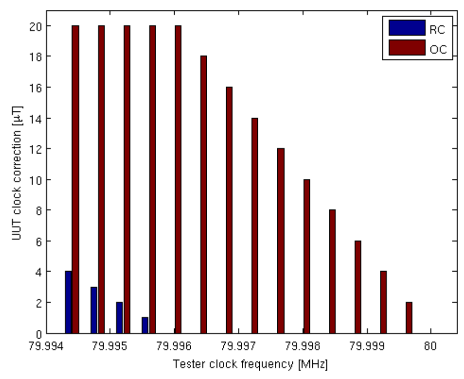

Fig. 6 EUT *pClusterDriftDamping* = 10 μT

Figure 7 shows results of simulation intended to evaluate method for the RC limit (*pRateCorrectionOut*) measurement. Frequency step is 80 kHz, which corresponds to 200 μT per cycle. Observations confirm our expectation - when the limit of rate correction is reached the even cycles lengthening stops. This situation can be detected by precise cycle length measurement.

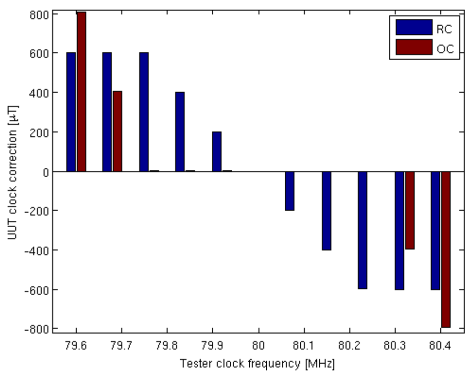

Fig. 7 Rate correction limit measurement method evaluation

Last simulation analyses SYM reaction to step (offset) change of synchronization frames transmit time. Unfortunately EUT does not react only by OC correction, but RC is affected too. Figure 8 shows the situation when synchronization frames were shifted by 100 μT in odd cycle no. 1. Reaction of EUT starts in following even cycle no. 2. Values in graphs are deviations from proper action point from TN point of view. Applied offset change is corrected by EUT's rate and offset correction in cycles no. 2 – 5. From sixth cycle the deviation is constant and equal to applied offset change.

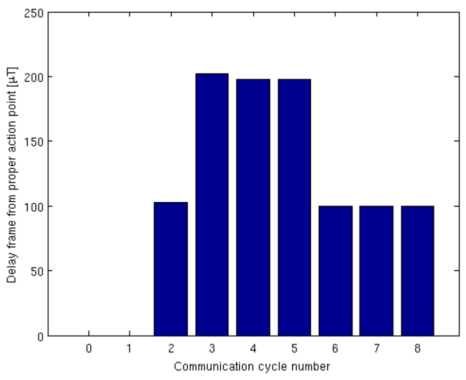

Fig. 8 Offset correction limit measurement method evaluation

7. EXPERIMENTAL EVALUATION

Presented measurement methods were implemented and evaluated with real FlexRay controllers. Illustration photo of experiment setup is in Fig. 9.

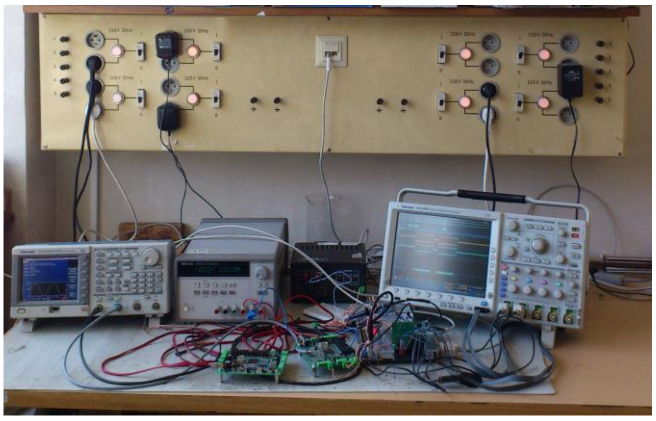

Fig. 9 Measurement place

Measurement methods were implemented in special system on chip developed in CTU Prague for FlexRay bus testing. This system is realized in Altera FPGA Cyclone II EP2C20F484C7N. The system works as our TN mentioned in section 4. The TN includes our own implementation of FlexRay controller with special testing capabilities in VHDL which is controlled by Altera's soft core microprocessor NIOS II. Tektronix AFG 3102 generator is used as precise 80 MHz clock source for the tester. Tester is able to timestamp different bus events with resolution of 12.5 ns. Detailed description of the FlexRay controller developed in CTU in Prague will be published later.

Two microprocessors with different integrated FlexRay controllers available on the market were used as EUT. First of them is Freescale microprocessor MC9S12XF. Freescale Semiconductor uses its own implementation of FlexRay controller. Second is Texas Instruments TMS570LS31 with Bosch E-Ray IP Module FlexRay controller. Both controllers are implemented according to the specification V2.1. Tester and EUTs are using the same bus drivers NXP Semiconductors TJA1080.

Implementation details

Presented methods are based on even and odd cycle length measurement. For this purpose TN is able to timestamp incoming frames by 64 bit timestamp directly derived from 80 MHz clock signal. Next timestamps are processed by NIOS II microprocessor and stored in external 512 KB SRAM. User interface with TN is provided by RS232 and terminal program running on PC.

Measurements details

For all measurements *pdMicrotick* was 25 ns long according to table 3. More experiments were performed with microprocessor TMS570LS31, as the module offers more comfortable configuration using graphical user interface without need to rebuild the firmware. Finally, we were limited by time constraints too.

pdMicrotick

According to presumption mentioned in section 5, offset correction change by 1 μT occurs in each n-th odd communication cycle. Limitation of this method is resolution of tester 12.5 ns, which is not enough for the shortest possible *pdMicrotick* with same length 12.5 ns (in case of single measurement when averaging is not possible).

Table 3 *pdMicrotick* measurement

MCU	Set [ns]	Measured [ns]
MC9S12XF	25	25
TMS570LS31	25	25

pClusterDriftDamping

Measurement method gradually increases/decreases CC length by 1 μT. Direction is determined by OC at start of measurement. If OC value is positive algorithm tries to increase CC, in opposite case the CC length is sequentially shortened. Measurement ends when OC stops changing or maximal *pClusterDriftDamping* value 10 μT is reached.

Table 4 *pClusterDriftDamping* measurement

MCU	Set [μT]	Measured [μT]
MC9S12XF	1	1
TMS570LS31	1	1
TMS570LS31	3	3
TMS570LS31	4	4
TMS570LS31	7	7
TMS570LS31	10	10

pRateCorrectionOut

Measurement principle was described in section 5. Correction of small frequency difference by 1 μT each n-th odd CC is filtered by averaging. Table 5 shows minimal and maximal measured CC lengths. *pRateCorrectionOut* is calculated from CC lengths according to formula 2. Actual and measured values are the same in all cases.

Table 5 *pRateCorrectionOut* measurement

MCU	CC min [ns]	CC max [ns]	*pRateCorrectionOut* [μT]
MC9S12XF	4984975	5014975	600
TMS570LS31	4992125	5007125	300
TMS570LS31	4984625	5014625	600
TMS570LS31	4977125	5022125	900

8. CONCLUSION

Methods for evaluation of selected parameters of the FlexRay node were presented. Evaluation of these methods was done by simulation using analytic model of synchronization mechanism. The methods were implemented using our FPGA based FlexRay node tester with exception of method for *pOffsetCorrectionOut* measurement. Experimental results figured out that first three methods are able to measure parameters for at least two types of FlexRay controller available on the market. All four methods are suitable for independent evaluation of FlexRay node timing parameters which are critical for fault free communication.

9. REFERENCES

[1] FlexRay Consortium, FlexRay Protocol Specification V3.0.1 , 2010. [Online]. Available: http://www.flexray.com.

[2] Sedaghat, Y. & Miremadi, S. G. (2010), 'Classification of Activated Faults in the FlexRay-Based Networks', *JOURNAL OF ELECTRONIC TESTING-THEORY AND APPLICATIONS* **26**(5), 535-547.

[3] Navet, N.; Song, Y.; Simonot-Lion, F. & Wilwert, C. (2005), 'Trends in automotive communication systems', *PROCEEDINGS OF THE IEEE* **93**(6), 1204-1223.

[4] Zeng, H.; Di Natale, M.; Ghosal, A. & Sangiovanni-Vincentelli, A. (2011), 'Schedule Optimization of Time-Triggered Systems Communicating Over the FlexRay Static Segment', *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* **7**(1), 1-17.

[5] Obermaisser, R., ed. (2011), *Time-Triggered Communication*, CRC Press.

[6] Sobotka, J., Novak, J. and Malinsky, J. "Analytic Model of FlexRay Synchronization Mechanism" 'IDAACS'2011 - Proceedings of the 6th IEEE International Cnonference on Intelligent Data Acquisition and Advanced Computing Systems', IEEE, THEY, CTU, Prague, 2011, pp. 969--974.

[7] Armengaud, E. and Steininger, A. "Remote Measurement of Local Oscillator Drifts in FlexRay Networks" 'DATE: 2009 DESIGN, AUTOMATION & TEST IN EUROPE CONFERENCE & EXHIBITION, VOLS 1-3', IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA, Design, Automation and Test in Europe Conference and Exhibition, Nice, FRANCE, APR 20-24, 2009, 2009, pp. 1082-1087