CAN FD performance analysis for ECU re-programming using the CANoe

Conference Paper · June 2014

DOI: 10.1109/ISCE.2014.6884472

CAN FD Performance Analysis for ECU Re-programming Using the CANoe

Tien-Hai Nguyen
Department of Electrical and Computer Engineering and Faculty of Cybernetics
Sungkyunkwan University and Tula State University
Suwon, Republic of Korea and Tula, Russian Federation
nguyentienhai@gmail.com

Bo Mu Cheon, Jae Wook Jeon
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Republic of Korea
cheonbomu@naver.com, jwjeon@yurim.skku.ac.kr

***Abstract*—This paper presents performance analysis of Controller Area Network with Flexible Data-Rate (CAN FD), a new automotive communication protocol that was officially proposed by Robert Bosch GmbH in March 2012, with consideration of the size of data field of the CAN FD message. Using the CANoe simulation tool by Vector GmbH, we measured the performance of CAN FD for ECU (Electronic Control Unit) re-programming, and compare it to that of CAN. The experimental result showed that when flashing large amounts of data to ECUs, CAN FD has significantly better performance than CAN. In addition, for identical amounts of data, the longer the size of the data field, the shorter the software download time.**

***Keywords*— CAN FD; performance analysis; automotive network; CANoe; ECU re-programming**

I. INTRODUCTION

The Controller Area Network (CAN), which was officially released in 1986 by Bosch GmbH, is still the dominant communication protocol used in automotive systems today [1]. As time goes by, more and more applications require real-time data transfer with a bit rate higher than that supported by high-speed CAN (maximum of 1 Mbps). Therefore, it has recently been a trend to replace CAN with FlexRay in vehicle networks. FlexRay is very different to CAN in nature, because it is a deterministic time-trigger protocol, while CAN is event-triggered [2]. In order to change systems over from CAN to FlexRay, automotive engineers have to systematically change their way of thinking on network design, as well as their working habits; automotive OEMs also have to replace most of hardware devices and software packages. The costs of implementing these changes would be very high, so not many automotive OEMs have implemented FlexRay in their vehicle networks to date. Automotive OEMs would prefer a step by step improvement with maintenance of CAN features, rather than an outright revolution. Evolution is very important in the automotive industry [3].

In March 2012, Bosch officially announced CAN FD - an evolution protocol of CAN that inherited many of the CAN features, such as bus arbitration and message acknowledgment. Moreover, CAN FD supports sizes of the data field up to 64 bytes, and a data rate over CAN's 1 Mbps limit [4, 5, 6, 7, 8]. While CAN FD solves the limitations of CAN mentioned above, it is also cheaper than implementation of FlexRay [9]. CAN standard communication can also be performed with CAN FD controllers. The automotive industry hopes that CAN FD will be the next protocol to fill in the gap between CAN and FlexRay in the near future. International standardization of CAN FD is now in progress, and it will be included in the next revision of ISO 11898-1 standard [10]. While CAN FD transceiver chips are now available on the market, microcontrollers supporting CAN FD have not been available for sale. Many automotive OEMs interested in CAN FD are waiting for microcontrollers from chip makers which support CAN FD so that they can do real experiments to test it before application to in-vehicle networks.

At this moment, there is a lack of study on CAN FD. Authors in a previous paper [11] analyzed the network performance of CAN FD using the CANoe simulation tool, but they limited the size of the data field to 8 bytes, though CAN FD supports 12, 16, 20, 24, 48 and 64 bytes. In this paper, we will also use the CANoe to analyze the CAN FD network performance for re-programming ECUs with data field sizes over 8 bytes, and also compare it to CAN.

II. MAIN CHANGES IN CAN FD COMPARED TO CAN

CAN FD was designed based on CAN. The development of CAN FD had the requirement to increase the data rate, while leaving the original CAN physical layer unchanged. The reliability of CAN arbitration has been preserved in CAN FD, because the bit rate of the arbitration phase in CAN FD was not changed. Instead, it just increases the bit rate of the data phase after BRS (bit rate switch) bit. CAN FD also extends the size of the data field, supporting 12, 16, 20, 24, 32, 48, and 64 bytes [6]. Standard CAN uses cyclic redundancy check code CRC-15, while CAN FD, besides CRC-15, also uses CRC-17 for frames with up to 16 data bytes, and CRC-21 for frames with more than 16 data bytes [12]. A comparison of the data frame structure of standard (also known as base format) CAN and CAN FD is shown in Fig. 1.

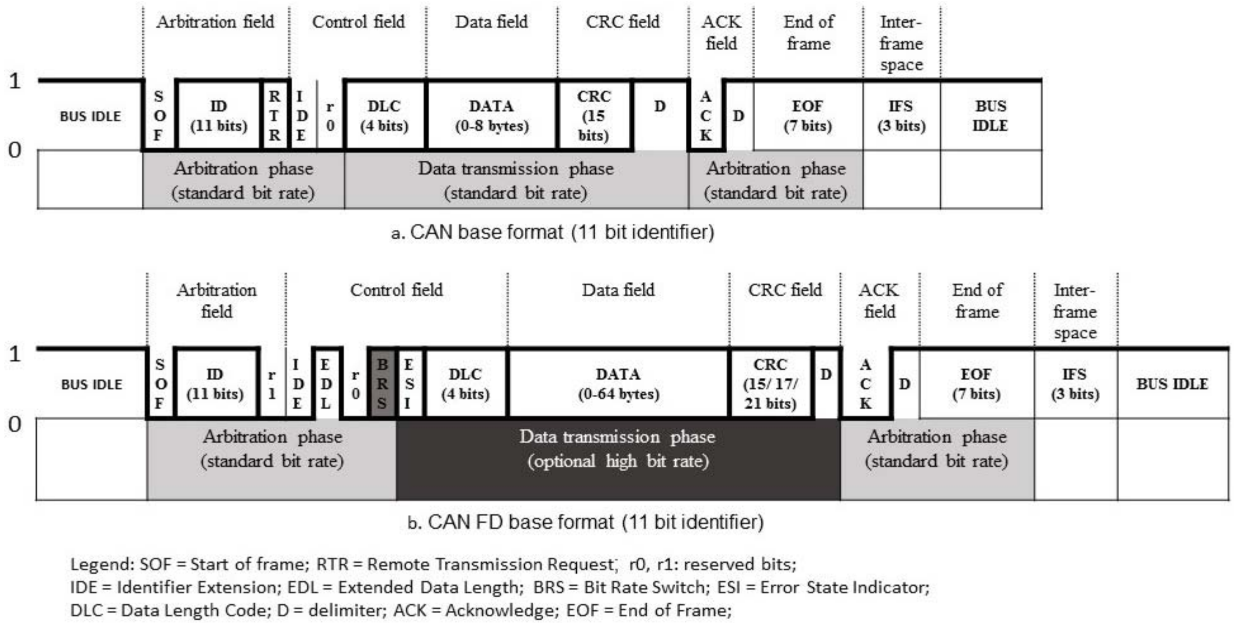

Fig. 1. CAN and CAN FD data frame format comparison

As shown in Fig. 1, the arbitration phase (until BRS bit) of CAN FD uses the standard CAN bit rate (up to 1 Mbps), while the data transmission phase (after BRS bit) uses the optional high bit rate (up to 15 Mbps) [5]. These two bit rates may have the same value. The frame structure of CAN FD with a 29 bit identifier was described in details in a CAN FD white paper of the Bosch [6].

III. ECU RE-PROGRAMMING WITH CAN FD

Use in the ECU re-programming process is one of the many functions of CAN FD [13]. It has three phases: first, erasing the flash memory; second, transmitting software to the ECU and programming the data segment; and finally, verification [14]. The software download time of the ECU re-programming has to meet the requirements of real time systems.

In this paper, we analyze the performance of CAN FD with various sized data fields. The transceiver ECU sent the same amount of data using CAN FD messages of the same size with 8, 12, 16, 20, 24, 32, 48, 64 data bytes, respectively, to the receiver ECU, and the transmission time (software download time) was measured. Because the least common multiple of 8, 12, 16, 20, 24, 32, 48 and 64 is 960, we sent a multiple of 960 (i.e., 960*n, n is positive integer) bytes to the receiver ECU using CAN FD, so that the numbers of transmitted messages were integers. Table I shows the relationship of the size of the data field (S) and the number of transmitted messages (M) for sending 960*n bytes to the ECU. For examples, in order to send 960*n bytes to ECU, 40*n CAN FD messages of 24 bytes in data, or 15*n CAN FD messages of 64 bytes in the data field were sent.

The transmission time t was calculated by formula (1) as follows:

$$t = t_1 - t_0, \tag{1}$$

where t_0 is the point of time when the transmitter ECU started to send the first message, t_0 value is integrated in the data field of the transmitted message; and t_1 is the point of time when the last message was received at the receiver ECU.

TABLE I. RELATION OF THE SIZE OF DATA FIELD (S) AND THE NUMBER OF TRANSMITTED MESSAGES (M)

S	8	12	16	20	24	32	48	64
M	120n	80n	60n	48n	40n	30n	20n	15n

*For sending 960n bytes

TABLE II. TOTAL SIZE OF CAN FD FRAME IN BITS WITHOUT STUFF BITS

	FD 8	FD 12	FD 16*	FD 20	FD 24	FD 32	FD 48	FD 64
Arbitr. phase size, [bits]	29	29	29	29	29	29	29	29
Data trans. phase size, [bits]	85	119	151	187	219	283	411	539
Total frame size, [bit]	114	148	180	216	248	312	440	568

*FDm: CAN FD with m bytes in data field without stuff bits

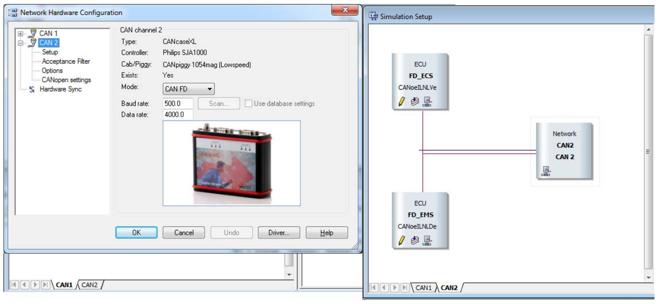

Fig. 2. CAN FD virtual network setup in the CANoe tool

A CAN frame with 8 bytes in data field has total size of 111 bits without stuff bits. The total frame size of CAN FD with m bytes in the data field (FDm) is shown in Table II, as follows.

IV. SIMULATION SETUP

The CAN FD virtual network was set up in the CANoe simulation tool [15] as shown in Fig. 2. In this experiment, we sent 960 KB to the receiver ECU using CAN FD with various sized data fields. Bit rates were set as follows: 500 Kbps for the arbitration phase, and 4000 Kbps for the data transmission phase.

V. EXPERIMENT RESULT AND DISCUSSION

The software download time T_d using CAN with 500 Kbps bit rate for 960 KB data (according to Table 1, n = 1024) was calculated as follows: $T_d = 120n*111/500000 = 27.2794$ (s), and was experimentally measured to take 29.83524 seconds.

The transmission time of one CAN FD message with n_1 bits in the arbitration phase, n_2 bits in the data field, bit rates f_1 [bits/s] in the arbitration phase and f_2 [bits/s] in data phase is calculated by formula (2) as follows:

$$T_{dFD} = n_1/f_1 + n_2/f_2. \quad (2)$$

We ignored the time for switching bit rates. Table III shows the transmission time of CAN FD messages with f_1 = 0.5 Mbps, f_2 = 4 Mbps.

As shown in Table 3, the transmission time when using CAN FD messages with a data field of 64 bytes was just 32% of the transmission time required for an 8-byte data field, and

TABLE III. THE ECU RE-PROGRAMMING TIME USING STANDARD CAN FD WITH 0.5 MBPS FOR ARBITRATION PHASE AND 4 MBPS FOR DATA PHASE

Data field size [byte]	Transmission time of one message [µs] (theoretical)	Number of messages	Total transmission time for 960 KB [s] (theory)	Total transmission time for 960 KB (measured)
8	79.25	122,880	9.73824	10.45708
12	87.75	81,920	7.18848	7.691064
16	95.75	61,440	5.88288	6.36624
20	104.75	49,152	5.148672	5.602192
24	112.75	40,960	4.61824	5.050254
32	128.75	30,720	3.9552	4.355914
48	160.75	20,480	3.29216	3.661612
64	192.75	15,360	2.96064	3.317384

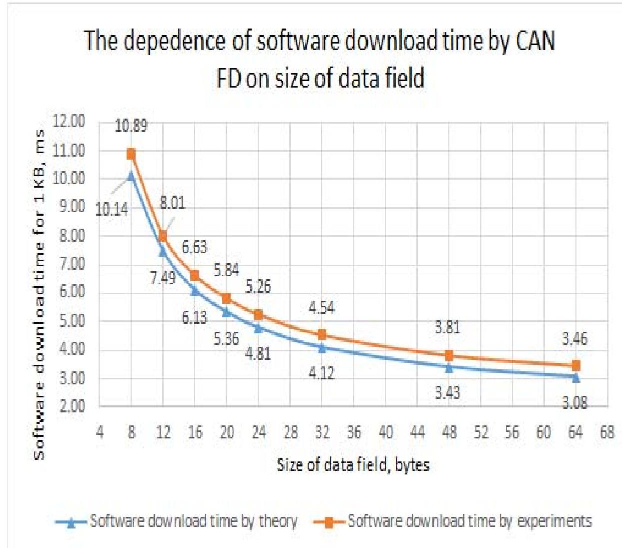

Fig. 3. The dependence of CAN FD software download time on the size of the data field

11% of the time required when using CAN with 8-byte data field. Dividing by 960, we can obtain the dependence of the software download time by CAN FD on the size of data field of the CAN FD message for 1 KB, as shown in Fig.3.

I. CONCLUSION

The experimental results showed that with CAN FD, the software download time significantly decreased as the size of the data field increased. Using CAN FD with data field of 64 bytes can reduce 68% software download time compared to CAN FD with 8-byte data field and 89% compared to CAN with 8-byte data field. This characteristics ensures that the software restoration process will work in real time conditions when faults occur, therefore improving the safety of the system.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2010-0020210).

REFERENCES

[1] "CAN history", CAN in Automation. Available online: http://www.can-cia.de/index.php?id=161

[2] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in Automotive Communication Systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204-1223, June 2005.

[3] P. Decker, "CAN gets even better", Vector Informatik GmbH, April 2013 [Translation of a German public in Hanser automotive, issue February 2013].

[4] "The CAN FD protocol overcomes CAN limits", CAN Newsletter Online, March 2012. Available online: http://www.can-newsletter.org/engineering/standardization/nr_stand_can-fd_2012-03-08/

[5] F. Hartwich, "CAN with flexible data-rate", CAN Newsletter, February 2012.

[6] Robert Bosch GmbH, "CAN with Flexible Data-Rate", white paper version 1.1, Germany, August 2011.

[7] Robert Bosch GmbH, "CAN with Flexible Data-Rate", specification version 1.0, Germany, April 2012.

[8] "First-hand information on CAN FD", CAN Newsletter Online, December 2012. Available online: http://www.can-newsletter.org/catalogue/do/printArticle/id/nr_stand_can-fd_detroit_121022/nr_stand_can-fd_detroit_121022.pdf

[9] T. Lindenkreuz, "CAN FD – CAN with Flexible Data-Rate", Vector Kongress, 2012.

[10] "Progress in CAN FD standardization", CAN Newsletter, October 2013. Available online: http://www.can-newsletter.org/engineering/standardization/nr_progress-in-can-fd-standardization_iso-cia_131004/

[11] B. Cheon, J. Jeon, "The CAN FD network performance analysis using the CANoe", 44th International Symposium on Robotics (ISR), 2013.

[12] D. Hickman, "Speed up your calibration with CAN FD", International CAN Conference (iCC), 2013.

[13] Y. Pradeep, "CAN-FD and Ethernet create fast reliable automotive data buses for the next decade", Automotive Compilation Vol. 10, Atmel, 2013.

[14] P. Decker, "High speed reprogramming and calibration with CAN FD: A case study", Vector Informatik GmbH, 14th iCC, Paris, France, 2013.

[15] CANoe 8.2, Vector Informatik GmbH. Available online: http://vector.com/vi_canoe_en.html