

논문 2019-56-12-14

# 차량 내부 통신을 위한 고속 CAN-FD 컨트롤러

## (High-Speed CAN-FD Controller for In-Vehicle Network)

최도영\*, 윤영현\*, 오정환\*\*, 이승은\*

(Do Young Choi, Young Hyun Yoon, Jung Hwan Oh, and Seung Eun Lee<sup>©</sup>)

### 요약

Controller Area Network (CAN)은 차량 내부 네트워크로 가장 많이 사용되는 통신 프로토콜이다. CAN 프로토콜은 비용, 신뢰성 측면에서 차량 내부 통신으로 적합한 특징을 지니고 있다. 하지만 최근 들어 차량에 적재되는 Electronic Control Unit (ECU)이 증가하게 되면서, CAN bus의 부하를 줄일 수 있는 새로운 통신 프로토콜이 필요하게 되었다. Controller Area Network with Flexible Data rate (CAN-FD)는 CAN과 비교했을 때, 전송 속도가 빠르고 전송량이 크다는 장점을 갖고 있다. 이러한 장점은 CAN node들이 공유하는 CAN bus의 부담을 줄여 차량 내부 전자 장치의 동작 반응 시간을 단축시킨다. 또한, CAN과 CAN-FD는 서로 호환 가능하기 때문에 유지보수가 편리하다는 장점이 있다. 본 논문에서는 8 Mbit/s 통신 속도의 CAN-FD 컨트롤러를 제안한다. 제안한 CAN-FD 컨트롤러는 ISO-11898 국제 표준을 준수하여 설계되었으며, Field Programmable Gate Array (FPGA)를 통해 기능을 검증하였다. 또한 CAN 통신 hardware scope와 software를 사용하여 CAN-FD 컨트롤러의 출력 frame을 측정하고 국제 표준에 맞는 frame임을 검증하였다. 제안하는 CAN-FD 컨트롤러는 차량 내 ECU 간의 향상된 통신을 구축하는데 일조할 것으로 사료된다.

### Abstract

Controller Area Network (CAN) is the most commonly used communication protocol for In-Vehicle Network. CAN protocol has proper properties for In-Vehicle Network in terms of cost and reliability. As Electronic Control Units (ECUs) loaded on vehicles are increasing recently, new communication protocols are required to reduce the burden on the CAN bus. Compared to the CAN protocol, Controller Area Network with Flexible Data rate (CAN-FD) has higher transmission rate and payload. These advantages make response time of electronic devices shorter by reducing the loads of CAN bus. CAN-FD is also easy to maintain as CAN and CAN-FD are compatible with each other. In this paper, we propose CAN-FD Controller with 8 Mbit/s transmission rate. Our CAN-FD Controller is designed in accordance with ISO-11898 international standard and we verified the CAN-FD Controller by implementing on Field Programmable Gate Array (FPGA). Also we measured and verified whether the CAN frame is in accordance with the standard by using CAN hardware scope and software. We expect that the proposed high-speed CAN-FD Controller will contribute to advanced communication among multiple ECUs.

**Keywords** : Controller Area Network, CAN-FD, ECU, In-Vehicle Network, FPGA

## I. 서론

차량에 적재되는 전장의 기능이 점차 복잡해지면서 전장

\*정회원, 서울과학기술대학교 전자공학과  
(Department of Electronic Engineering, Seoul National University of Science and Technology)

\*\*정회원, 현대오트론(Hyundai Autron)

© Corresponding Author(E-mail : seung.lee@seoultech.ac.kr)

※ 이 연구는 서울과학기술대학교 교내연구비의 지원으로 수행되었습니다.

Received ; September 3, 2019 Revised ; September 17, 2019

Accepted ; October 31, 2019

의 입력 신호를 분석하고 계산하는 Electronic Control Unit (ECU)의 성능이 나날이 발전하고 있다. 최근 생산되는 차량의 ECU는 이전의 ECU에 비해 더 많은 연산을 수행하며 다량의 데이터를 송수신한다. 이러한 영향으로 인해 ECU 간의 데이터 전송량이 증가하게 되었고, 차량 내부 네트워크가 감당해야 할 데이터 전송량과 전송 속도 또한 증가하게 되었다<sup>[1]</sup>. 대부분의 차량 내부 통신 프로토콜로 사용된 Controller Area Network (CAN)은 다른 프로토콜에 비해 비용, 성능, 반응속도 및 신뢰성 측면에서 더 나은 이점이 있다<sup>[2-3]</sup>. 하지만 기술의 발전으로

데이터 처리 요구사항이 증가하게 되면서 CAN 통신 방식을 대체할 새로운 통신 프로토콜이 필요하게 되었다. 이에 따라 CAN의 신뢰성과 전송 속도를 보완하여 더 좋은 성능을 갖춘 Controller Area Network with Flexible Data rate (CAN-FD)가 새로운 차량 내부 통신 프로토콜로 대체되는 추세이다. CAN의 단점을 보완한 CAN-FD는 최대 8 Mbit/s의 데이터 전송 속도로 통신을 수행한다<sup>[4~5]</sup>. 또한 CAN-FD는 CAN에 비해 더 넓은 대역폭을 갖는 장점이 있다. CAN은 하나의 CAN frame을 1 Mbit/s 속도로 최대 8 byte까지 전송 가능하지만, CAN-FD는 8 Mbit/s 속도로 최대 64 byte까지 전송이 가능하다. 이는 차량의 특성상 많은 데이터가 송수신되는 CAN 버스의 부하를 줄일 수 있는 장점이 된다. 이러한 장점을 기반으로 ECU 간의 고속 통신 시스템 구축이 가능하다.

본 논문에서는 최대 8 Mbit/s의 속도로 통신할 수 있는 고속 CAN-FD 컨트롤러를 제안한다. 제안하는 CAN-FD 컨트롤러는 ISO-11898 국제 표준을 토대로 설계되었으며, Field Programmable Gate Array (FPGA)에 구현하여 동작 검증을 수행하였다. 또한 hardware scope로 CAN frame을 분석하여 ISO-11898 국제 표준 준수 여부를 검증하였다.

## II. CAN-FD protocol

### 1. CAN and CAN-FD

CAN은 차량 내부 통신으로 가장 많이 쓰이는 통신 프로토콜 중 하나이다. Bosch 사의 차량용 네트워크 프로젝트로 처음 개발되었으며, 1986년에 CAN 프로토콜이 공식적으로 공개되었다. 이후 통일된 CAN 프로토콜을 사용할 수 있도록 ISO-11898 국제 표준이 제정되었다. 이후 CAN 프로토콜은 In-Vehicle Network에 적합한 특징 덕분에 많은 차량에서 내부 통신 프로토콜로 사용되고 있다.

CAN 통신 프로토콜은 In-Vehicle Network로 적용하기에 적합한 여러 특징이 있다. 우선 CAN은 다른 차량 내부 통신 프로토콜에 비해 node 설치 비용 대비 통신 속도가 빠르다<sup>[6]</sup>. 또한 cyclic redundancy check (CRC), bit stuffing 등의 오류 방지 기능을 적용하고 외부 노이즈에 강한 2선 차동 신호를 사용하여 높은 신뢰성을 갖추고 있다<sup>[7]</sup>. 하지만 이러한 장점에도 불구하고, 차량에 탑재되는 ECU의 증가와 ECU 간의 데이터 전송량이 증가하게 되면서 CAN bus의 부하를 줄일 수 있는 새로운 프로토콜이 필요하게 되었다.

CAN-FD는 CAN의 단점을 보완한 통신 프로토콜이다. 통신 속도가 1 Mbit/s로 일정한 CAN과 다르게 CAN-FD는 data phase에서 최대 8 Mbit/s 통신이 가능하다. 또한 하나의 CAN frame이 보낼 수 있는 데이터 전송량도 최대 64 byte로 증가하게 되면서 더 많은 데이터를 짧은 시간 동안 송수신할 수 있게 되었다. 이러한 장점은 CAN node 간의 빠른 통신을 가능하게 하며, 탑승자의 안전과 관련된 ECU의 데이터가 차량의 주행에 신속히 적용될 수 있도록 한다. 또한 CAN-FD는 CAN 통신과 호환이 가능하므로 CAN-FD 통신 프로토콜을 새롭게 적용할 때 비용 부담이 적다는 장점이 있다.

### 2. CAN node

CAN-FD 프로토콜을 적용한 차량의 ECU는 CAN node를 매개로 CAN frame을 송수신한다. CAN node는 CAN-FD 컨트롤러와 CAN transceiver로 구성되며, CAN bus와 2선 차동 신호로 연결되어 ECU 간의 통신을 가능하도록 한다. CAN node 내의 CAN-FD 컨트롤러는 ECU로부터 받은 데이터를 기반으로 CAN frame을 생성하거나, CAN bus로부터 받은 CAN frame을 샘플링하고 분석하여 data field 구간을 ECU에 전달하는 기능을 수행한다. CAN transceiver는 디지털 기반 시스템인 CAN-FD 컨트롤러가 아날로그 신호를 기반으로 한 CAN bus를 매개로 통신할 수 있도록 한다. CAN-FD 컨트롤러가 CAN frame을 송신하는 경우, CAN transceiver는 디지털 신호를 2선 차동 신호로 변환하는 기능을 수행한다. 반대로 CAN bus로부터 데이터를 받는 경우에는 2선 차동 신호를 디지털 신호로 변환하여 CAN-FD 컨트롤러가 CAN frame을 정상적으로 수신할 수 있도록 한다.

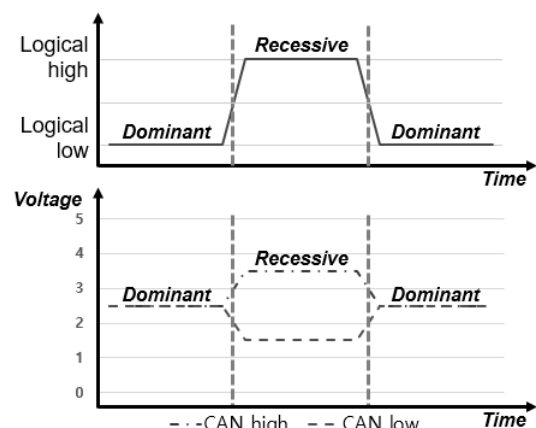


그림 1. CAN 2선 차동 신호의 dominant, recessive  
Fig. 1. Dominant, recessive of CAN-bus differential signal.

CAN bus는 CAN node들을 물리적으로 연결하며, CAN high와 CAN low로 구성되는 2선 차동 신호를 사용하여 신뢰성을 갖추고 있다. 그림 1은 CAN 통신에서 사용하는 2선 차동 신호를 나타낸다. 디지털 신호 값이 0일 때, CAN high와 CAN low 신호는 기준 전압인 2.5 V를 갖는다. 이는 CAN 통신에서 dominant이며 우성 bit를 의미한다. 반대로 디지털 신호 값이 1이면, CAN high와 CAN low 신호는 각각 3.5V, 1.5V를 갖는다. 해당 신호를 CAN 통신에서는 recessive로 규정하며, 이는 열성 bit를 의미한다. Dominant는 recessive보다 우선권을 가지며, CAN bus는 2선 차동 신호의 특성을 사용하여 CAN node에 우선순위를 부여한다. 각각의 CAN node는 고유 ID를 갖게 되는데, 통신 하고자 하는 다른 CAN node들과 MSB에서 LSB순으로 ID를 비교하여 우선순위가 높은 CAN node가 CAN bus를 먼저 사용할 수 있게 한다. CAN 통신 프로토콜의 우선순위 특성은 차량에서 중요한 기능을 수행하는 CAN node에게 우선순위가 더 높은 ID를 할당하여, CAN bus를 사용하고자 하는 CAN node가 많은 경우에도 차량의 정상적인 동작이 가능하게 한다.

### 3. CAN frame

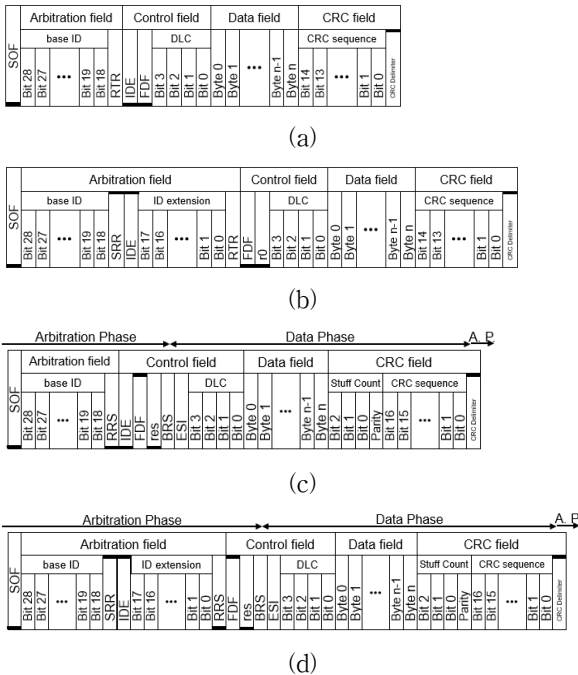


그림 2. CAN frame의 종류 (a) CBFF, (b) CEFF, (c) FBFF, (d) FEFF  
Fig. 2. Format of CAN frame (a) CBFF, (b) CEFF, (c) FBFF, (d) FEFF

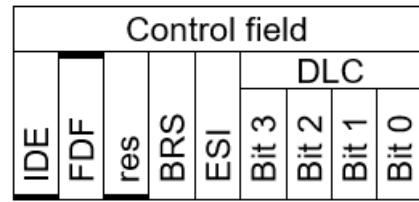


그림 3. CAN frame의 control field  
Fig. 3. Control field of CAN frame

CAN node 간의 통신을 위해서는 CAN 통신 규약에 맞는 CAN frame을 사용해야 하며, CAN frame의 형식은 ISO-11898에 명시되어 있다. 그림 2는 국제 표준에 명시된 CAN frame의 종류를 나타낸다. CAN frame은 Classical Base Frame Format (CBFF), Classical Extended Frame Format (CEFF), FD Base Frame Format (FBFF), FD Extended Frame Format (FEFF)가 있다. CAN frame 형식은 총 4가지 종류가 있으며, 통신 속도와 ID를 기준으로 분류된다. Classical과 FD는 data phase에서 고속 통신의 가능 여부를 기준으로 분류되고, Base와 Extended는 ID의 bit 수를 기준으로 구분된다. Classical frame은 전체 frame이 1 Mbit/s의 일정한 속도로 통신이 이뤄지며, FD frame은 data phase에서 8 Mbit/s로 통신이 수행된다. Base frame은 11 bit ID를 사용하고, extended frame은 29 bit 크기의 ID가 할당된다.

각각의 CAN frame은 4개의 field로 분류되며, 4개의 field는 CAN frame의 각기 다른 정보를 포함하고 있다. 그림 2에 보이는 바와 같이, CAN frame은 frame의 시작을 나타내는 sof와 arbitration field, control field, data field, CRC field로 구성된다. Arbitration field는 CAN node의 ID에 대한 정보를 포함하고 있다. Control field는 데이터의 크기나 속도를 결정하는 bit들로 구성된다. Data field는 CAN node가 송수신하는 데이터이며, CRC field는 CRC 연산 결과를 포함한다.

CAN frame을 생성하는 데 있어서 가장 중요한 field는 control field라고 할 수 있다. Control field는 CAN frame의 특성 정보를 포함하고 있다. 그림 3은 CAN frame의 일부인 control field를 나타낸 그림이다. Control field의 dlc는 data field의 크기를 결정하며, brs는 data phase의 전송 속도를 결정한다. Classical frame은 control field 내의 dlc 값에 따라 최소 0 byte에서 최대 8 byte 크기의 data field를 사용할 수 있다. 반면, FD frame은 0 byte에서 최대 64 byte까지 할당 가능하다. Control field 내의 brs는 data phase의 전송 속도를 결정한다.

표 1. CAN frame과 data field의 크기에 따른 CRC 연산 범위 및 크기

Table1. Range and payload of CRC calculation according to CAN frame and data field

Frames	Payload	Range of CRC	CRC calculation bit
Classical	0 ~ 8 Byte	sof ~ data field	15 Bit
FD	0 ~ 16 Byte	sof ~ stuff count	17 Bit
	20 ~ 64 Byte	sof ~ stuff count	21 Bit

brs가 0이면 data phase를 1 Mbit/s로 송수신하고, brs가 1이면 data phase를 8 Mbit/s로 통신한다.

CRC field는 CRC 연산과 관련된 정보들로 구성된다. CAN 통신 프로토콜은 송신 측 CAN node로부터 받은 CRC 연산 값과 수신 측 CAN node가 직접 계산한 CRC 연산 값을 비교하여 수신 받은 CAN frame에 이상이 없는지 확인한다. 이때 CRC 연산 범위와 bit 수는 표 1에 명시된 바와 같이, CAN frame의 종류와 data field의 크기에 따라 다르게 적용된다. Classical frame은 data field의 크기와 상관없이 항상 15 bit CRC 연산을 수행한다. 또한 CAN frame의 sof에서부터 data field 영역에 CRC 연산을 수행하여 연산 결과가 CRC sequence를 구성하게 된다. 반면에 FD frame은 CRC 연산 범위가 sof에서 stuff count까지이며, data field의 크기가 16 byte 이하일 경우에는 17 bit CRC 연산을 수행하고, 20 byte 이상일 경우에는 21 bit CRC 연산을 수행한다.

기능에 따라 4개의 field로 분류된 CAN frame은 전송 속도에 따라 2개의 phase로 분류된다. 통신 속도가 항상 일정한 classical frame은 phase의 구분이 없는 반면, 고속 통신이 가능한 FD frame은 arbitration phase와 data phase로 구분된다. Arbitration phase는 classical frame의 통신 속도와 같은 1 Mbit/s로 송수신이 이뤄지며, 범위는 CRC field에 속하는 CRC delimiter에서 brs까지다. 반면 data phase는 8 Mbit/s 고속 통신이 가능한 영역이며, 범위는 brs부터 CRC delimiter까지다.

4. Bit stuffing

전자 장치 간의 통신을 위해 사용되는 디지털 인코딩 방식은 None Return to Zero (NRZ), Return to Zero (RZ), Manchester 등 여러 방식이 있다. 그중 NRZ는 CAN, LIN 등 In-Vehicle Network에서 많이 사용되는 디지털 인코딩 방식이며, active low일 때 ‘0’으로 인식하고 active high일 때 ‘1’로 인식하는 인코딩 과정을 수행한다. 또한 NRZ는 Manchester 방식이나 RZ 방식과

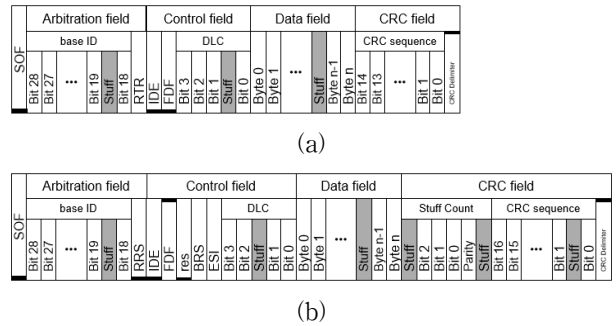


그림 4. Bits tuffing이 적용된 CAN frame (a) CBFF, (b) FBFF

Fig. 4. CAN frame with bit stuffing (a) CBFF, (b) FBFF

는 달리, bit 마다 edge를 만들거나 3 가지 이상의 전압 상태를 사용하지 않으므로 다른 방식들에 비해 구현하기 쉽다는 장점이 있다<sup>[8]</sup>. 하지만 디지털 데이터가 일정한 값을 유지하는 경우에는 송신 측과 수신 측의 클럭 동기화에 어려움이 생기는 단점이 있다. 이러한 단점을 보완하고자 NRZ 디지털 인코딩을 사용하는 통신 프로토콜 중 일부는 bit stuffing 방식을 사용한다. Bit stuffing은 기준 이상의 연속적인 디지털 신호가 검출되는 경우에 그 신호의 반대 bit 값을 추가함으로써, 수신 측의 동기화에 지장이 없도록 한다. NRZ 디지털 인코딩 방식을 사용하는 통신 프로토콜 중 하나인 CAN도 bit stuffing 방식을 사용한다. CAN 통신 프로토콜은 bit stuffing을 사용함으로써, NRZ 방식의 동기화 문제를 해소하고, 송수신된 CAN frame에 에러가 없는지 검출하는 과정을 진행한다.

그림 4는 bit stuffing이 적용된 CAN frame이다. CAN 프로토콜은 CAN frame의 종류에 따라 bit stuffing 방식에 차이가 있다. Classical frame의 경우, bit stuffing 적용 범위는 sof에서 data field 까지이며 5 bit의 연속적인 값이 나올 때만 적용된다. 반면 FD frame은 classical frame에서 진행된 bit stuffing 과정 외에도 CRC field에 추가적인 bit stuffing이 적용된다. FD frame의 bit stuffing 진행 과정은 classical frame과 마찬가지로 5 bit 마다 stuffing을 진행한다. 이후, 적용된 bit stuffing 전체 개수에 modulo 8 연산을 진행하여 얻은 3 bit 결과값과 결과값의 even parity 값으로 stuff count를 구성한다. CRC field에 속하는 stuff count와 CRC 연산 결과인 CRC sequence가 완성되면 FD frame의 전체 CRC field에 추가적인 bit stuffing을 적용한다. CRC field에 적용하는 bit stuffing은 classical frame에서 적용한 연속적인 값에 대한 bit stuffing이 아니며, 4 bit 마다 주기적으로 이전 값의 반대 bit 값을 CAN

frame에 추가하는 과정을 진행한다. Classical frame에 비해 더 많은 데이터를 전송하는 FD frame은 이러한 추가적인 bit stuffing을 적용함으로써, CAN frame의 stuff 에러를 더 정밀하게 감지하게 된다.

### III. CAN-FD 컨트롤러

#### 1. CAN-FD 컨트롤러 구조

CAN-FD 프로토콜을 적용한 In-Vehicle Network는 CAN-FD 컨트롤러를 매개로 통신을 수행한다. 그림 5는 CAN-FD 컨트롤러를 기반으로 한 In-Vehicle Network 구조이다. 제안하는 시스템은 ECU, CAN node, CAN bus로 구성되며, CAN node에 포함되는 CAN-FD 컨트롤러에 의해 전체 네트워크의 성능이 결정된다. CAN-FD 컨트롤러의 주된 기능은 ECU의 전송 데이터를 기반으로 CAN frame을 생성하여 CAN bus에 송신하거나, 수신 받은 CAN frame의 데이터 부분을 ECU에 전송하는 것이다. 그림 6은 설계한 CAN-FD 컨트롤러의 구조도를 나타낸다. CAN-FD 컨트롤러는 차량 내 전장간의 원활한 통신을 위해 최대 8 Mbit/s 통신이 가능하다. 또한 입출력 신호로는 can rx, can tx를 사용하며, CAN transceiver를 매개로 CAN frame을 CAN bus에 송수신한다. CAN-FD 컨트롤러는 receiver, framer, transmitter 모듈로 구성된다.

CAN-FD 컨트롤러 내의 receiver 모듈은 CAN bus로부터 CAN frame을 수신하는 기능을 수행한다. 또한 receiver 모듈은 CAN frame을 샘플링하고, 에러 여부를 확인하는 기능을 한다. Receiver 모듈은 CRC 연산 결과를 수신 받은 CRC 결과와 비교하여 CAN frame의 CRC 에러를 검증한다. 또한 6개의 연속적인 bit 값이 없는지 확인함으로써 stuff 에러의 유무를 확인한다. CAN frame

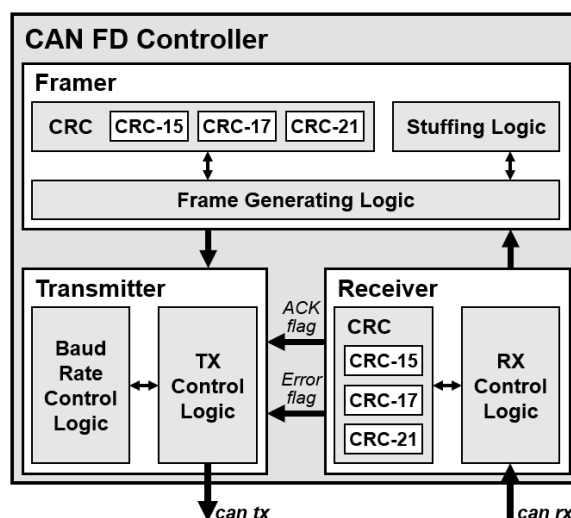


그림 6. CAN-FD 컨트롤러 구조도  
Fig. 6. Block diagram of CAN-FD Controller.

이 FD frame일 경우, 추가적으로 stuff count를 비교하여 stuffing bit 개수를 확인하는 절차를 진행한다. Receiver 모듈은 CRC 에러와 stuff 에러 외에도 bit 에러, form 에러, ack 에러를 확인한다. Bit 에러를 감지하기 위해, receiver 모듈은 디지털 신호가 2선 차동 신호와 다른 지 확인한다. Form 에러는 CAN frame이 ISO-11898 국제 표준 규약에 맞지 않을 때 발생하며, ack 에러는 수신 측 CAN node가 CAN frame을 수신하고 나서 ack 신호를 보내지 않을 때 감지된다. 수신 받은 CAN frame에 하나의 에러라도 발생하게 되면 error flag를 transmitter에 보낸다. 반대로 에러가 없다면, transmitter에 ACK flag를 보낸다. CAN frame을 정상적으로 수신했다는 의미의 ACK flag를 전송하면, receiver 모듈은 시리얼 통신을 사용하여 ECU에게 CAN frame의 data field를 전송한다.

Framer 모듈은 ECU의 전송 데이터를 CAN 통신 규약에 맞는 CAN frame으로 변환하는 기능을 수행한다. 하나의 CAN frame을 생성하기 위해서는 많은 과정이 수행된다. 우선, framer 모듈은 CAN node의 ID, CAN frame의 종류, ECU 데이터를 기반으로 arbitration field, control field, data field를 생성한다. 이후 3개의 field에 stuff 에러가 발생하지 않도록 5개의 연속적인 0 또는 1 값을 감지하고, 연속적인 값의 반대 값을 bit stuffing 한다. Bit stuffing 작업이 완료되면 stuffing bit의 개수를 modulo 8 연산하여 3 bit의 값으로 나타내고 해당 3 bit 값의 even parity를 생성하여 총 4 bit의 stuff count를 구성한다. 이후에 CAN frame의 종류와 data field의 길이에 해당하는 CRC 모듈을 사용하여 CRC 결과값을 계

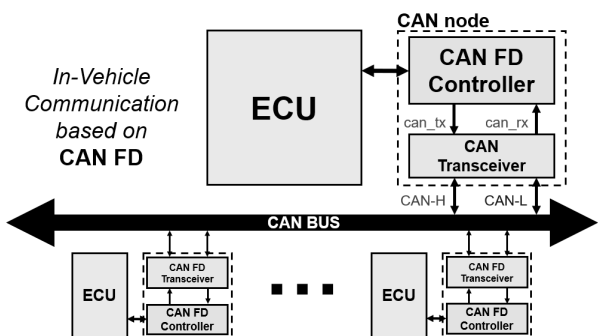


그림 5. CAN-FD 컨트롤러 기반의 In-Vehicle Network  
Fig. 5. In-Vehicle Network based on CAN-FD Controller

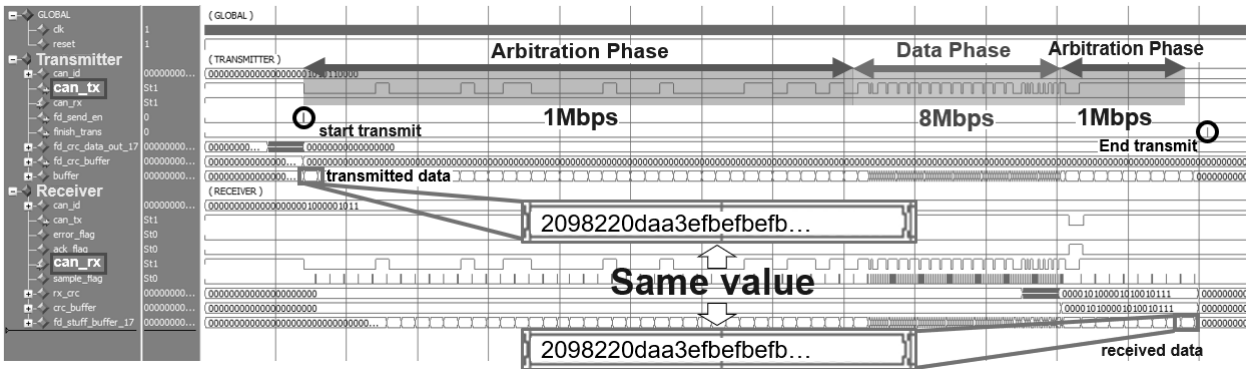


그림 7. CAN-FD 컨트롤러의 시뮬레이션 결과  
Fig. 7. Simulation result of CAN-FD Controller.

산한다. Classical frame은 CRC 15 모듈을 사용하여 15 bit의 CRC 연산 결과를 도출한다. 반면 FD frame은 data field가 16 byte 이하일 경우에는 CRC 17을 사용하고, 20 byte 이상일 경우에는 CRC 21을 사용하여 CRC sequence를 구성한다. CRC 모듈이 생성한 stuff count와 CRC sequence가 CRC field의 초기값을 구성하게 된다. 마지막으로 CRC field에 4 bit 마다 이전 bit의 반대 값을 stuffing 하는 과정을 진행하면 최종적인 CAN frame이 완성된다.

Transmitter 모듈은 framer 모듈이 생성한 CAN frame을 CAN bus로 송신하는 기능을 한다. CAN frame의 전송 속도는 CAN frame의 종류와 control field를 기준으로 다르게 적용된다. Classical frame은 전체 데이터 프레임의 통신 속도가 1 Mbit/s로 일정하므로 phase의 구분이 없다. 하지만 FD frame은 control field 정보에 따라 arbitration phase와 data phase의 전송 속도가 다르게 적용된다. Transmitter 모듈은 arbitration phase를 classical frame의 송신 속도와 같은 1 Mbit/s로 전송하는 반면, data phase에서는 brs 정보에 따라 전송 속도가 달라진다. 제안하는 CAN-FD 컨트롤러는 brs 값이 1이면 data phase를 8 Mbit/s로 전송하고, brs 값이 0이면 1 Mbit/s로 전송한다.

### III. 실험

#### 1. Simulation

제안한 고속 CAN-FD 컨트롤러를 Verilog HDL을 사용하여 구현하였으며 시뮬레이션을 통해 기능을 검증하였다. 그림 7은 CAN-FD 컨트롤러의 시뮬레이션 결과이다. 해당 시뮬레이션에서는 CAN-FD 컨트롤러의 송수신 기능을 검증하기 위해 두 개의 CAN-FD 컨트롤러를 생성하였다.



그림 8. CAN-FD 컨트롤러의 검증 환경  
Fig. 8. Verification environment of CAN-FD Controller

하나의 CAN-FD 컨트롤러는 송신 기능을 수행하며, 다른 하나의 CAN-FD 컨트롤러는 CAN frame을 수신하여 에러 여부를 검사한다. 송신 측 CAN-FD 컨트롤러는 CAN frame을 CAN bus에 전송하기 위해, framer 모듈이 생성한 CAN frame을 transmit data에 저장한다. 이후 송신 측은 transmitter 모듈의 동작에 따라, transmitted data에 저장된 CAN frame을 arbitration phase에서는 1 Mbit/s, data phase에서는 8 Mbit/s 속도로 버스에 전송한다. 수신 측 CAN-FD 컨트롤러는 receiver 모듈을 사용하여 송신 측이 전송한 CAN frame을 수신한다. Receiver 모듈은 arbitration phase와 data phase의 통신 속도에 맞게 sampling point를 적용하여 CAN frame을 수신한다. 수신한 CAN frame은 received data에 저장되며, 저장된 CAN frame에 CRC, stuff, bit, form, ack 에러가 없는지 확인한다. 해당 시뮬레이션에서는 에러가 발생하지 않았으므로, 수신 측 CAN-FD 컨트롤러는 CAN bus에 ACK 신호를 전송한다. 또한 송신 측의 transmit data와 수신 측의 received data가 같으므로, 시뮬레이션으로 검증한 CAN-FD 컨트롤러의 통신이 정상적으로 동작했음을 확인할 수 있다.

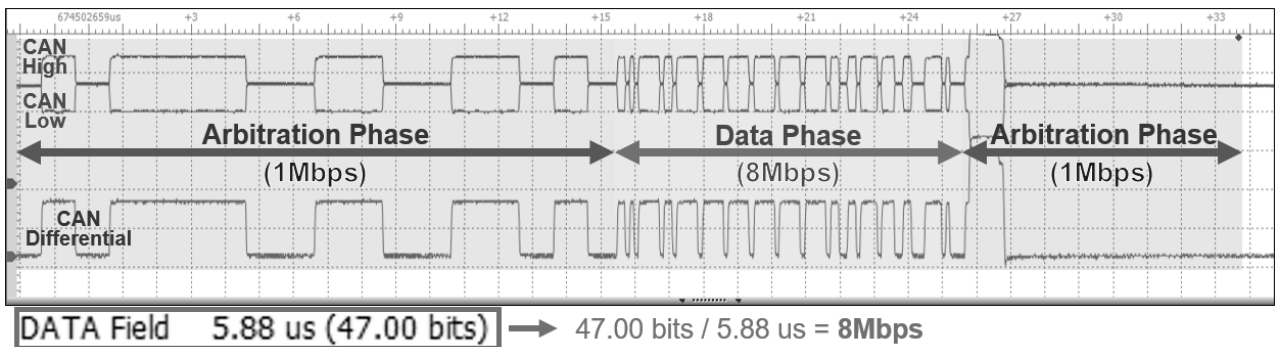


그림 9. Hardware scope 측정 결과  
Fig. 9. Result of hardware scope measurement.

## 2. Hardware implementation

시뮬레이션으로 검증한 CAN-FD 컨트롤러가 CAN 국제 표준인 ISO-11898 통신 규약을 준수하는지 확인하기 위해 그림 8과 같은 검증 환경을 구축하였다. 제안하는 CAN-FD 컨트롤러는 Field Programmable Gate Array (FPGA)에 구현하였으며, 동작 검증을 위해 FPGA와 CAN transceiver를 포함하는 prototype 보드를 제작하였다. 또한 CAN 통신 전용 hardware scope인 PicoScope 5000 Series를 사용하여 CAN transceiver의 2선 차동 신호를 분석하고, VN1630A를 사용하여 hardware scope와 PC의 인터페이스를 구축하였다. Hardware scope로 측정된 CAN frame은 CANoe 소프트웨어를 사용하여 분석했다. 그림 9는 CANoe 소프트웨어로 얻은 CAN frame의 측정 결과이다. CAN frame 측정을 통해 CAN transceiver의 2선 차동 신호인 CAN high와 CAN low를 확인할 수 있으며, 두 신호의 차인 CAN differential 신호가 CAN-FD 컨트롤러의 디지털 신호와 같은 값을 갖는지 확인하였다. 또한 CAN differential 신호를 분석하여 수신 받은 CAN frame의 에러 발생 여부를 확인하였다. 검증을 수행한 결과, prototype으로 구현한 CAN-FD 컨트롤러는 data phase에서 8 Mbit/s 속도로 통신을 수행하였고, ISO-11898 표준에 맞는 CAN frame을 생성하였다.

이후, 이전 연구에서 설계한 CAN 컨트롤러, 최대 2 Mbit/s 속도를 갖는 CAN-FD 컨트롤러의 전송속도를 측정하여 논문에서 제안하는 CAN-FD 컨트롤러와 성능을 비교하였다. 전송 frame은 FBFF 형식을 사용하였으며, data field의 전송량은 5 byte를 사용하였다. 또한, stuff bit를 고정하기 위해 모든 실험에 같은 데이터를 사용하였다. 전송속도 측정 결과, 하나의 frame을 전송하기 위해 CAN과 2 Mbit/s 속도의 CAN-FD 컨트롤러는 각각 80.3us, 56.8us의 전송 시간이 측정되었다. 반면

제안하는 CAN-FD 컨트롤러의 전송 시간은 39.18us로 측정 되었으며, 이는 CAN 컨트롤러에 비해 2.05배, 2Mbit/s 속도의 CAN-FD 컨트롤러에 비해 1.45배 빠른 전송 성능을 보인다. CAN-FD 컨트롤러의 전송속도 성능은 data field의 전송량이 증가할수록 더 높은 효율을 보이며, 이는 CAN bus의 부하를 줄여 차량 내 ECU간의 원활한 통신을 가능하게 한다.

## IV. 결론

본 논문에서는 차량 내부 통신 프로토콜을 위한 고속 CAN-FD 컨트롤러를 제안한다. 고속 CAN-FD 컨트롤러는 최대 8 Mbit/s 통신이 가능하며, FPGA에 구현하여 동작을 검증하였다. 또한 CAN 통신 전용 측정 장비와 소프트웨어를 사용하여 ISO-11898 국제 표준에 부합하는지 확인하였다. 제안한 CAN-FD 컨트롤러는 다량의 ECU 데이터를 고속으로 송수신할 수 있으며, CAN bus의 부하를 줄임으로써 차량 내부에 사용되는 전장파 ECU의 발전 속도에 부응할 수 있을 것으로 사료된다.

## REFERENCES

- [1] Jung Hwan Oh et al., "An FPGA-based Electronic Control Unit for Automotive System," 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, March 2019.
- [2] Jung Woo Shin et al., "CAN-FD controller for In-Vehicle System," 2016 International SoC Design Conference (ISOCC), Jeju, South Korea, December 2016.
- [3] Jung Woo Shin et al., "In-Vehicle CAN-FD Network for Smart Wearable Devices," 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, January 2017.

[4] Guo-sheng Feng et al., "CAN Bus Application in Automotive Network Control," 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha City, China, March 2010.

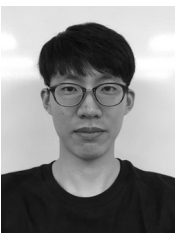
[5] Samuel et al., "A Practical Security Architecture for In-Vehicle CAN-FD," IEEE Transactions on Intelligent Transportation Systems, March 2015.

[6] C. Bomu and J. Jae Wook, "The CAN-FD network performance analysis using the CANoe," IEEE ISR, Seoul, South Korea, October 2013..

[7] ISO 11898, International Organization for Standardization, 2015.

[8] Singh et al., "Investigation of XOR operation in all-optical system with NRZ, RZ and Manchester modulation formats," International Conference on Communications, Devices and Intelligent Systems (CODIS), Kolkata, India, February 2013.

저 자 소 개



최 도 영(정회원)  
 2019년 명지대학교 전자공학과  
 학사 졸업.  
 2019년~현재 서울과학기술대학  
 교 전자공학과 석사 과정  
 재학.

<주관심분야: 컴퓨터 구조, SoC, NoC, 하드웨어 가속기>



윤 영 현(정회원)  
 2019년 서울과학기술대학교 전자  
 공학과 학사 졸업.  
 2019년~현재 서울과학기술대학  
 교 전자공학과 석사 과정  
 재학.

<주관심분야: 컴퓨터 구조, SoC, 마이크로프로세서 시스템, 머신러닝>



오 정 환(정회원)  
 2017년 서울과학기술대학교 전자  
 공학과 학사 졸업.  
 2019년 서울과학기술대학교 전자  
 공학과 석사 졸업.  
 2019년~현재 현대 오토론 연구원.

<주관심분야: 컴퓨터 구조, SoC, 하드웨어 멀티코어 스케줄러>



이 승 은(정회원)  
 1998년 한국과학기술원 전자공학  
 과 학사 졸업.  
 2000년 한국과학기술원 전자공학  
 과 석사 졸업.

2008년 University of California, Irvine 전자공학  
 과 박사 졸업.  
 2010년~현재 서울과학기술대학교 전자공학과 교수  
 <주관심분야: 컴퓨터 구조, MPSoC, 저전력 내고  
 장형 VLSI, 하드웨어 가속기>