논문 2013-08-09

FlexRay 네트워크 시스템을 위한 FIBEX 자동 생성 알고리즘에 관한 연구

(A Study on FIBEX Automatic Generation Algorithm for FlexRay Network System)

박 지 호, 이 석*, 이 경 창
(Ji-Ho Park, Suk Lee*, Kyung-Chang Lee)

Abstract : As vehicles become more intelligent for safety and convenience of drivers, in-vehicle networking systems such as controller are network (CAN) have been widely used due to increasing number of electronic control unit (ECU). Recently, FlexRay was developed to replace CAN protocol in chassis networking systems, to remedy the shortage of transmission capacity and unsatisfactory real-time transmission delay of conventional CAN. However, it is difficult for vehicle network designers to calculate platform configuration registers (PCR) and determine a base cycle or slot length of FlexRay. To assist vehicle network designers for designing FlexRay cluster, this paper presents automatic field bus exchange format (FIBEX) generation algorithm from CANdb information, which is de-facto standard database format for CAN. To design this program, structures of FIBEX, CANdb and relationship among PCR variables are analyzed.

Keywords : In-vehicle network, FlexRay, Field bus exchange format (FIBEX), CANdb, Platform configuration register, Automatic generation algorithm.

Ⅰ. 서론

최근 들어, 자동차의 안정성과 신뢰성에 대한 운전자들의 욕구가 증대됨에 따라, 고속 전송 속도와 예측 가능한(predictable) 전송 특성을 가진 프로토콜이 필요하게 되었다 [1-3]. 일반적으로, Controller Area Network (CAN)은 전송 속도가 낮고 전송 지연이 비확정적(non-deterministic)이기 때문에, 제동 시스템이나 조향 시스템과 같은 실시간 특성을 요구하는 섀시 네트워크(chassis network)용 프로토콜로 적합하지 않다고 알려져 있다 [4].

* Corresponding Author (slee@pusan.ac.kr)
Received: 22 Oct. 2012, Revised: 11 Dec. 2012, Accepted: 07 Jan. 2013.
J.H. Park, S. Lee: Pusan National University
K.C. Lee: Pukyong National University
※ 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

ISSN : 1975-5066

이러한 문제를 해결하기 위하여, 섀시 제어용 네트워크로 FlexRay가 개발되었으며, 자동차 완성차 업계를 중심으로 그 적용이 확대되어 가고 있는 추세이다 [5-10]. 그러나 FlexRay는 프로토콜 자체의 복잡성 때문에 CAN에 비하여 구현이 어렵다는 문제점을 가지고 있다. CAN에서는 수십년간의 개발과 적용 과정에 따라 다양한 개발 방법론들이 제시되어 있으며, 각 ECU에서 생성되는 센서나 모터 신호의 원활한 전송을 위하여 완성차업계에 의하여 CAN 메시지 표준이 잘 정립되어 있다. 특히, CAN은 수많은 차량의 섀시 네트워크 시스템에 적용되어 안정적으로 동작되고 있다 [11-12].

섀시 네트워크 시스템에서 CAN 네트워크를 FlexRay 네트워크로 변경하기 위해서는 시스템 설계 단계에서 네트워크의 구성과 동작에 관한 스케줄링이 필요하다. FlexRay 프로토콜은 네트워크에서 TDMA 방식이 가지는 장점을 효과적으로 사용하기 위해서 면밀한 네트워크 클러스터의 설계와 높은 수준의 ECU간 동기화가 요구된다. 그러나 프

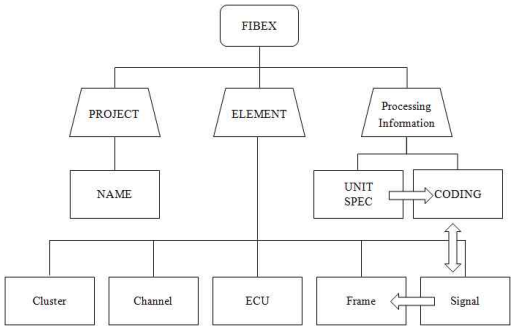

그림 1 . FIBEX 구조
Fig. 1 Structure of FIBEX

로토콜 규격을 분석하여 FlexRay 클러스터에 관한 수많은 네트워크 구성 변수(configuration register)를 계산하는 것은 기존의 CAN 네트워크 설계자에게 매우 어려운 일이며, FlexRay 프로토콜의 적용을 느리게 만드는 주요 원인이 되고 있다 [13].

CAN은 네트워크의 모든 설계 정보를 포함하고 있는 사실상의 표준(de-facto standard)인 CANdb라는 데이터베이스를 가지고 있으며, CANdb에는 메시지의 ID, 주기, 전송 노드, 수신 노드 등이 정의되어 있다. 반면, FlexRay 에는 FIBEX라는 데이터베이스를 가지고 있으며, 내부에 데이터 전송을 위한 플랫폼 구성 변수(platform configuration register)가 정의되어 있다. FlexRay의 플랫폼 구성 변수는 총 53개로서 수식이 상당히 복잡하고 서로 복잡하게 얽혀 있다 [13].

만약, CAN의 데이터베이스로부터 FlexRay의 데이터베이스로 이동(migration)시켜 줄 수 있다면 FlexRay 네트워크의 개발이 매우 용이해 질 것이다 [14]. 그러나 설계자가 메시지의 길이와 주기 등에 따라 플랫폼 구성 변수를 직접 설정하여야 하기 때문에, CAN 데이터베이스에서 FlexRay 데이터베이스로의 이동은 상당한 시간이 요구되는 매우 어려운 작업이다.

본 논문에서는 CAN 데이터베이스를 FlexRay 데이터베이스로 이동시킬 때, 설계의 편의성을 향상시키고 복잡도를 줄이기 위한 방법으로서 CANdb를 이용하여 FIBEX를 자동으로 생성할 수 있는 알고리즘에 대하여 제안한다. 이를 위하여, CANdb와 FIBEX를 분석한 후 CANdb와 FIBEX 간의 상호 연결 관계를 규명한다. 마지막으로, CANdb로부터 FIBEX를 자동 생성하는 프로그램을 개발한다.

II. FIBEX 및 CANdb 분석

1. FIBEX의 구조

ASAM(association for standardization of automation and measuring systems) 컨소시엄에 의해 개발된 FIBEX(field bus exchange format)는 데이터 및 정보를 쉽게 교환하기 위해서 자동차 산업에 도입되었으며, FlexRay의 네트워크 설계 정보에 대해 정의하고 있다. 차량용 FIBEX는 자동차의 네트워크를 나타내는 데 사용되는 XML(extensible markup language) 기반의 표준 형식으로서, 통신을 이용한 데이터 교환을 하고자 할 때 가장 적합한 기술로 인정받고 있다. 또한, 다양한 네트워크 프로토콜을 지원하는 확장성을 가지고 있다 [15].

FIBEX는 정보를 이해하기 쉽게 만들고, 접근을 용이하게 하기 위하여 객체 모델(object model)이라는 구조를 사용하고 있을 뿐만 아니라, 모든 정보는 다양한 객체로 나누어져 있다. 특히, 그 객체들 자체는 계층 구조(hierarchy)로 그룹화되어 분류된다. 일반적으로, 이러한 구조를 트리(tree) 구조라고 하며, FIBEX에 포함되어 있는 항목들을 엘리먼트(element)라고 한다.

그림 1은 FlexRay FIBEX의 구조를 나타내고 있다. 그림에서, FIBEX의 최상위에는 XML의 선언(declaration)에 관한 정보가 정의되어 있다. 여기에는 XML 버전, 파서(parser)에서 XML 정보를 분석하기 위한 인코딩 방식, 서로 다른 데이터 구분을 위한 네임 스페이스(name space)가 존재한다.

프로젝트(project) 요소에는 설계자가 FIBEX를 설계할 때 사용되는 프로젝트 이름이 정의되어 있다. 다음으로, 엘리먼트(element) 요소에는 클러스터(cluster), 채널, ECU, 프레임, 시그널에 관한 정보가 정의되어 있다. 클러스터 요소에는 53개의 클러스터 구성 변수(통신주기, 정적 구간, 동적 구간, 네트워크 유휴시간 등)들의 정보가 정의되어 있다. 채널 요소에는 채널 A, B를 통해 전송될 슬롯 정보에 대해 정의되어 있으며, 슬롯 ID에 따라 데이터가 전송될 절대 시간(absolute scheduling timing), 사이클 반복(cycle repetition), 주기 시간(cyclic timing)에 관한 정보가 정의되어 있다. ECU 요소에는 정의된 ECU의 이름과 노드 변수(키 슬롯, wake up 패턴, 각 채널의 지연 보상 등)에 관한 정보가 정의되어 있다. ECU마다 각각의 정보가 다르기 때문에 이와 같은 정보가 포함하게 된다. 프레임 요소에는 정의된 프레임의 이름, 페이로드 길이, 프레임

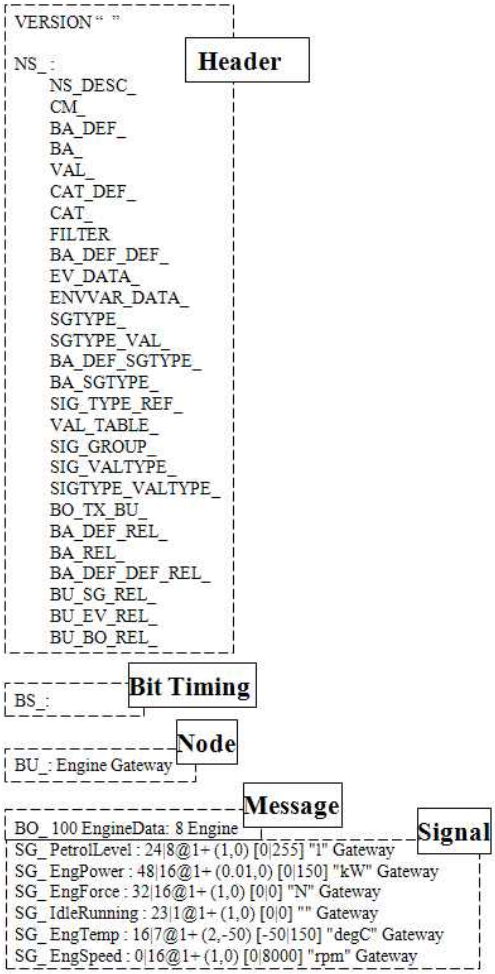

그림 2. CANdb의 구조
Fig. 2 Structure of CANdb

타입(frame type), 주기에 관한 정보가 포함되어 있다. 또한, 프레임에는 그 프레임에서 의미를 주고받을 수 있도록 하기 위해 사용되는 시그널들의 ID와 비트 위치(bit position), 저장 방식(byte order)에 관한 정보가 포함되어 있다. 하나의 시그널이 프레임 전체를 다 사용할 수도 있는 반면, 다수의 시그널이 하나의 프레임을 사용할 수도 있다. 시그널 요소에는 프레임 하부에 정의된 시그널에 관한 정보 즉, 각각의 시그널의 ID, 초기값(default value), 이름에 관한 정보가 포함되어 있다. 프레임 요소에 정의 되어 있는 시그널을 사용할 때 시그널 요소에 정의되어 있는 시그널의 ID를 이용한다.

마지막으로, 처리 정보(processing information)

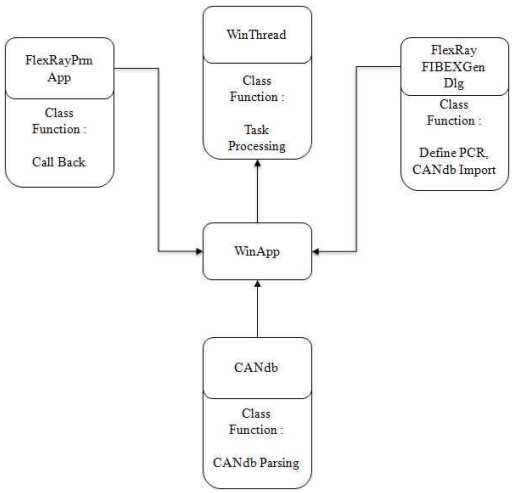

그림 3. FIBEX 자동 생성 알고리즘을 위한 클래스 디자인
Fig. 3 Class design of FIBEX automatic generation algorithm

요소에는 단위 정보(unit spec) 요소와 코딩(coding) 요소가 정의되어 있다. 단위 정보 요소에 정의된 ID는 시그널 코딩을 위해 코딩 요소에서 호출을 하게 된다. 또한 코딩 요소에는 시그널 인코딩 방법, 시그널 길이(length of bits), 시그널 최대값(upper limit), 최소값(lower limit)이 정의되어 있다.

이상과 같이 FIBEX 생성을 위해서는 클러스터 스케줄링에 관한 정보와 송수신 노드, 프레임에 대한 정보가 필요하다. 하지만, 프레임 내부의 시그널에 대한 정보는 FIBEX 구성에 있어서 필수적인 것은 아니다.

2. CANdb의 구조

CAN 네트워크에서는 CANdb를 통해 설정 정보들을 제공 받을 수 있다. CANdb는 차량업계 표준인 DBC 포맷으로 구성되어 있으며, FIBEX와 같이 차량용 ECU 간의 데이터 및 정보 공유를 위해 개발되었다 [16]. CANdb는 물리적 노드없이 시뮬레이션을 통해 모니터링과 분석을 할 수가 있다는 장점이 있다.

CANdb의 기본 동작을 위해서는 비트 타이밍(bit timing), 노드, 메시지들이 정의되어야 한다. 그림 2는 CANdb의 구조를 나타낸다. 그림에서, 헤더에는 버전에 관한 정보와 심볼(symbol)에 관한 정

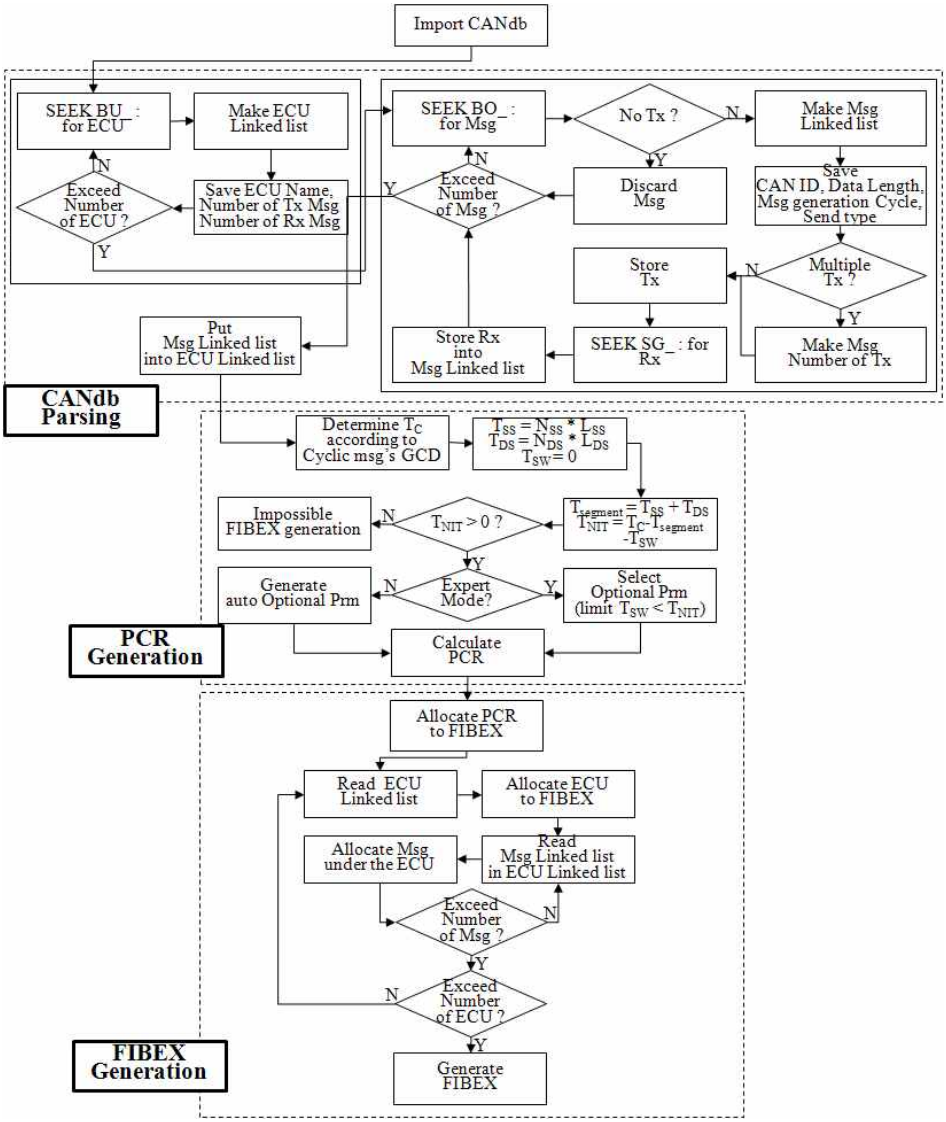

그림 4. FIBEX 자동 생성 알고리즘의 플로우차트
Fig. 4 Flowchart of FIBEX automatic generation algorithm

보가 정의되어 있다. 버전은 공백으로 두거나, CANdb 에디터(editor)를 이용하여 변경할 수 있다. 둘째, 비트 타이밍(bit timing)에는 CAN에서 사용될 대역폭과 BTR(bit timing register)이 정의되어 있다. 셋째, 노드(node)에는 통신에 참여하고 있는 노드의 이름이 정의되어 있다. 넷째, 메시지(message)에는 클러스터에 포함되어 있는 모든 메시지의 ID, 이름, 크기, 전송할 시그널에 대한 정보가 정의되어 있다. 마지막으로, 시그널(signal)에는 전송해야 되는 정보의 이름, 시작 비트, 크기, 저장 방법, 수신부 등이 정의되어 있다.

Ⅲ. CANdb 기반 FIBEX 자동 생성 알고리즘

그림 3은 FIBEX 자동 생성 알고리즘을 위한 클래스(class)별 설계도를 나타낸다. 그림에서, FlexRayPrmApp와 FlexRayFIBEXGenDlg, CANdb 클래스가 FIBEX 자동 생성 알고리즘의 핵심적인 역

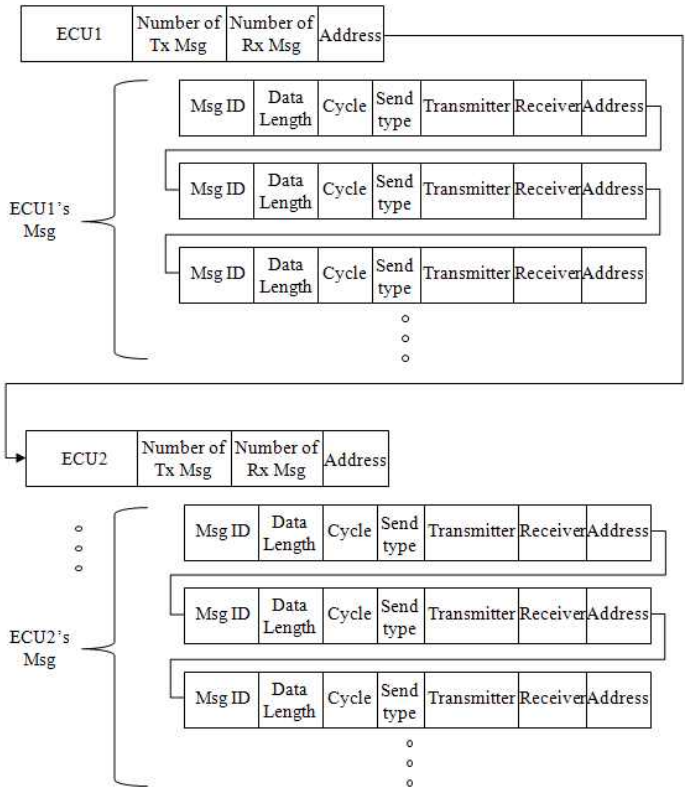

그림 5. 연결 구조를 이용한 ECU로의 메시지 할당 방법
Fig. 5 Allocation method of message to ECU using linked list

할을 수행한다. 먼저, FlexRayFIBEXGenDlg는 53개의 플랫폼 구성 변수와 프로그램 외부에 저장되어 있는 CANdb 불러오기(import), CANdb와 플랫폼 구성변수를 이용한 FIBEX 생성 등이 정의되어 있는 클래스이다. 여기서 정의된 플랫폼 구성 변수는 FlexRayPrmApp에서 사용되는데, 이 클래스는 플랫폼 구성 변수에 관한 내용을 함수(function)로 가지고 있어 필요 시 함수를 호출하여 사용하는 역할을 한다. CANdb를 불러오게 되면 CANdb의 송수신 노드, 메시지, 시그널, 주기, 크기 등을 고려하여 FIBEX를 생성하기 위해 CANdb의 데이터를 파싱(parsing)하게 되는데, 이는 CANdb 클래스에서 동작하도록 설계하였다.

WinThread 클래스는 세 가지 태스크를 동시에 수행하는 역할을 한다. 첫 번째 태스크는 CANdb에 의해 결정되는 5개의 FlexRay 기본 구성 변수(Base cycle (T_C), Number of Static Slot (N_{SS}), Length of Static Slot (L_{SS}), Number of Dynamic Slot (N_{DS}), Length of Dynamic Slot (L_{DS}))를 생성하는 역할을 한다. 먼저, CANdb에 정의되어 있는 메시지들의 주기를 이용하여 통신 주기(T_C)를 결정하고, 주기적 메시지와 비주기적 메시지를 판단하여 정적 슬롯의 개수(N_{SS})와 동적 슬롯의 개수(N_{DS})를 정하게 된다. 또한, 각각의 메시지의 크기를 고려하여 메시지 중 가장 큰 메시지의 크기를 통해 슬롯의 크기(L_{SS}, L_{DS})를 결정하며, 생성된 변수와 CANdb 정보를 이용하여 FIBEX를 생성하는 역할을 수행한다.

두 번째 태스크는 결정된 5가지의 구성 변수를 통해 나머지 48가지의 구성 변수를 업데이트하는 역할을 수행하며, 이는 FlexRay 규격 2.1 버전에 의거하여 계산된다. 마지막으로, 세 번째 태스크는 계산된 플랫폼 구성 변수와 CANdb 정보를 이용하여 FIBEX를 생성하는 역할을 수행한다.

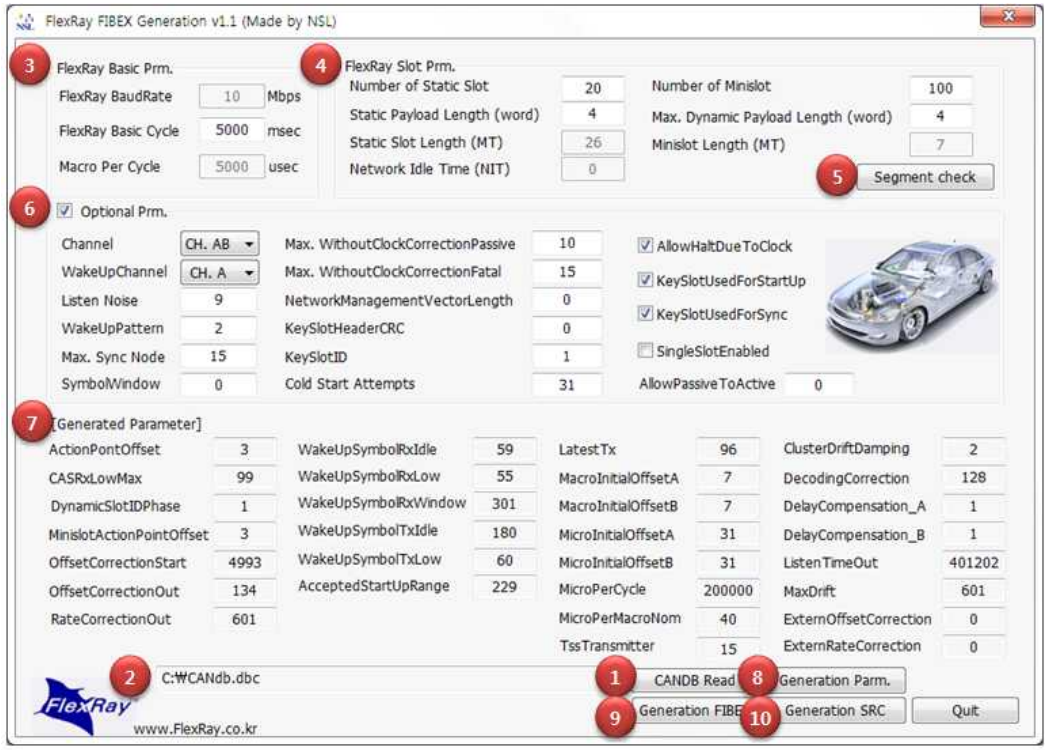

그림 6. FIBEX 자동 생성 프로그램
Fig. 6 FIBEX automatic generation program

본 논문에서 제안한 FIBEX 자동 생성 알고리즘은 그림 4와 같이 CANdb 파싱(CAN parsing) 기능, 플랫폼 구성 변수 생성(PCR generation) 기능, FIBEX 생성(FIBEX generation) 기능을 가지고 있다. CANdb 파싱 기능은 CAN 네트워크 설계 정보를 알 수 있는 CANdb를 불러온 이후에 진행되는데, 내부적으로 크게 두 가지 부분으로 나눌 수가 있다. 첫 번째는 ECU에 관한 정보를 수집하는 것이고, 두 번째는 메시지에 관한 정보를 수집하는 것이다. ECU에 대한 정보를 얻기 위해 CANdb에서 정의하고 있는 ECU 헤더 'BU_:'를 검색한다. CANdb에서 ECU에 대한 정보가 검색되면 ECU의 정보를 저장하기 위해 ECU 연결 목록(ECU linked list)을 생성하고, 내부에 정의되어 있는 ECU 이름(ECU name)과 전송 메시지 개수(number of tx msg), 수신 메시지 개수(number of rx msg)를 연결 목록에 저장하게 된다. 이러한 루틴을 CANdb에 정의되어 있는 ECU의 개수만큼 반복 수행하여 데이터를 저장하게 된다.

ECU에 관한 정보 수집이 끝나게 되면, 메시지에 대한 정보를 얻기 위해 동작하는 기능이 실행된다. 메시지의 헤더 BO_:를 검색하여 메시지에 대한 정보를 찾게 된다. 송신 ECU가 정의되어 있지 않는 메시지 같은 경우에는 FlexRay로 이동 시키면서 사용하는 슬롯의 개수를 줄이기 위해 저장하지 않도록 설계하였다. 그렇지 않고, 송신부가 정의되어 있는 메시지들은 메시지 연결 구조(Msg linked list)를 생성하여 CAN ID, 데이터 길이(data length), 메시지 생성 시간(msg generation cycle), 메시지의 주기적(cyclic)/비주기적(spontaneous)을 구분하여 전송 형태(send type)를 저장하게 된다. 하나의 메시지가 여러 ECU에서 전송되어야 한다면, ECU의 개수만큼 메시지를 생성하고 수신 ECU에 따라 메시지를 각각 저장하게 된다. 전송에 관한 부분이 끝나게 되면 시그널에 정의되어 있는 메시지가 수신될 ECU를 알기 위해 'SG_:'를 검색하며, 메시지가 수신될 ECU에 관한 정보를 메시지 연결구조에 저장하게 된다. 이와 같은 루틴을 CANdb에 정의되어 있는 메시지 수만큼 반복 수행하게 된다.

ECU와 메시지에 관한 정보를 연결 구조(linked list)에 모두 저장하게 되면, 그림 5와 같이 ECU 내부에 저장되어 있는 전송 메시지 정보를 이용하여 그에 맞는 메시지를 ECU 연결 구조 하부로 저장하게 된다. 이와 같은 과정을 통해 ECU에서 필요로 하는 정보를 모두 할당하여 메모리상에 가상 ECU를 만들고 하부에 메시지에 관한 정보를 저장하게 되면서, CANdb 파싱 기능은 종료된다.

CANdb의 파싱 기능이 종료되면, 플랫폼 구성 변수 생성 기능이 실행된다. 메시지 연결 구조에 저장된 모든 주기적 메시지의 생성 주기의 최대공약수(greatest common divisor, GCD)를 구해 FlexRay의 통신 주기를 결정한다. 이후 정적 구간의 시간을 결정하기 위해 주기적 메시지 개수와 데이터 길이의 최대값을 이용하여 계산하고, 동적 구간의 시간을 구하기 위해 비주기적 메시지 개수와 데이터 길이의 최대 값을 이용한다. 그리고 심볼 윈도우(TSW)가 반드시 필요한 구성 요소가 아니기 때문에 초기값을 '0'으로 설정하였다. 정적 구간 시간과 동적 구간 시간의 합인 세그먼트 시간($T_{segment}$) 과 통신주기, 심볼 윈도우를 이용하여 네트워크 유휴시간(T_{NIT})을 계산한다. 네트워크 유휴시간이 음의 값을 가지게 되면 FIBEX를 생성할 수 없다고 팝업(pop-up)시켜 사용자에게 알린다. FlexRay 프레임을 구성하기 위한 4가지의 구성요소(T_{SS}, T_{DS}, T_{SW}, T_{NIT})의 길이의 합이 사용자가 입력한 통신 주기(T_C)와 같으면, 선택적으로 변수를 설정할 수 있는 단계(expert mode)로 넘어가거나, 나머지 48가지 변수를 자동으로 생성할 수 있다. 선택적 변수 구성을 사용하게 되면, 웨이크 업 패턴(wake up pattern), 채널 선택, 심볼 윈도우 등을 사용자가 직접 설정할 수 있다. 만약 사용자가 심볼 윈도우를 선택한다면, 네트워크 유휴시간을 초과하지 않는 범위 내에서 선택을 해야 하며, 이를 반영하여 48개의 플랫폼 구성 변수를 생성한다.

마지막 기능은 CANdb 파싱을 통해 저장된 정보와 플랫폼 구성 변수 정보를 이용하여 FIBEX를 생성하는 기능이다. 가장 먼저 플랫폼 구성 변수의 태그(tag)를 만들고, 플랫폼 구성 변수의 값을 FIBEX에 할당하여, FIBEX 내에 FlexRay 네트워크의 클러스터 정보를 생성하게 된다. 이후 ECU의 정보를 FIBEX로 이동시켜야 하는데, ECU의 정보가 저장되어 있는 ECU 연결 구조를 읽어 FIBEX에 ECU에 대한 정보를 할당한다. ECU의 정보를 할당하였으면, ECU 연결 구조 내부에 저장되어 있는 메시지 연결 구조를 읽어와 ECU 하부에 포함되는 메시지에 대한 정보를 할당하게 된다. 이때, 메시지 생성주기의 GCD로 결정된 통신 주기와 동일하게 메시지 생성 주기를 변화시키게 된다. 이는 생성주기에 따라 데이터 프레임이 전송되거나 널 프레임이 전송되기 때문에 GCD의 생성주기를 가지면서 매 통신주기마다 전송되어도 문제가 되지 않기 때문이다. 정적 구간에 메시지를 할당할 때 저장된 메시지 중에 주기적 메시지만 고려하여 ID의 순서로 슬롯의 번호를 할당하여 배치하게 된다. 동적 구간에 메시지를 할당할 때도 정적 구간의 메시지 할당과 동일하게 비주기적 메시지만 고려하여 ID 순서로 미니 슬롯의 슬롯번호를 할당하게 된다. 하나의 ECU에 있어야 할 주기적 메시지와 비주기적 메시지가 모두 할당되면, 다른 ECU에 대한 정보와 메시지에 대한 정보를 할당하는 순서로 모든 CANdb에 정의 되어있는 모든 ECU가 할당 될 때까지 반복하게 된다. 이와 같은 반복 수행으로 인해 클러스터 정보와 ECU 정보, ECU 하부의 메시지 정보가 모두 할당되면 FIBEX 파일이 생성된다.

IV. CANdb 기반 FIBEX 자동 생성 프로그램의 구현

본 절에서는 CANdb를 기반으로 FIBEX를 자동 생성하는 프로그램을 설명한다. FIBEX 자동 생성 프로그램은 사용자의 편의성과 가독성을 위해 Microsoft사의 PC 프로그램 개발 소프트웨어인 Visual Studio 2010의 MFC(Microsoft function class)를 사용하여 GUI 형태로 개발하였다. 또한, 읽어 들인 DBC 파일이나 생성된 PCR을 사용자가 확인할 수 있도록 구성하였다.

그림 6은 본 논문에서 설계한 FIBEX 자동 생성 프로그램의 화면을 나타내었다. 그림에서 보는 것과 같이 총 9개의 부분으로 구성을 하였다. 먼저, ①을 누르게 되면 프로그램 외부의 저장매체에서 CANdb 파일을 선택하여 불러올 수가 있으며, 위치는 ②에 표시가 되어 어떤 파일을 읽어왔는지 확인할 수 있도록 제작하였다. ③에서는 FlexRay 네트워크의 통신 속도와 통신 주기를 나타내었다. 통신 속도는 FlexRay에서 가장 빠른 10Mbps로 설계하였고, 통신 주기는 CANdb의 메시지들의 GCD를 이용하여 나타내었다.

또한, 계산의 편의상 1MT를 1us로 설정하기 위해 'Macro per Cycle' 값을 통신 주기와 동일하게

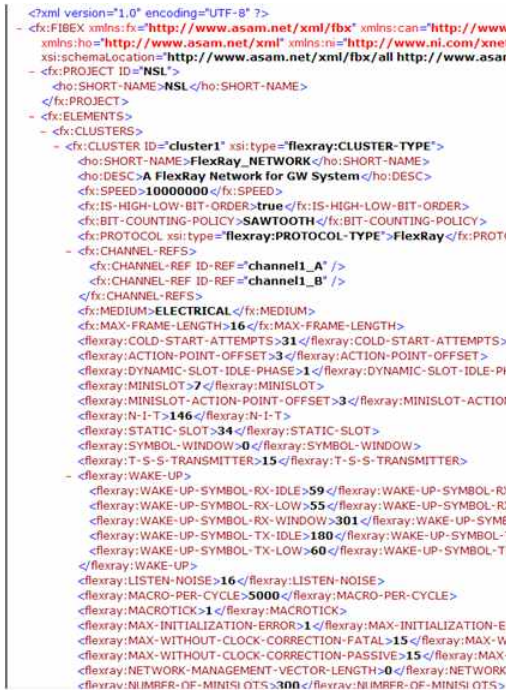

그림 7. FIBEX 자동 생성 프로그램의 FIBEX 자동 생성 결과

Fig. 7 Result of FIBEX automatic generation

설정하였다. CANdb 내부에 정의된 주기적 메시지 개수 및 크기, 비주기적 메시지 개수 및 크기를 FlexRay로 이동시켜 ④와 같이 정적 슬롯 개수 및 크기, 동적 슬롯 개수 및 크기를 나타내고, 그 값을 이용하여 슬롯 하나마다 가지는 시간의 크기를 나타내었다. 이후 ⑤를 누르게 되면 통신 주기와 정적 구간, 동적 구간을 고려하여 네트워크 유휴시간을 계산하게 된다. 여기서 정적 구간, 동적 구간, 네트워크 유휴시간을 합한 값이 통신 주기를 넘게 되면 사용자에게 FIBEX를 생성할 수 없다고 알린다. 이와 같은 과정이 다 끝나면 ⑥에서 선택적 구성 변수를 설정하는 단계로 넘어가게 된다. 여기서는 설계자의 필요에 따라 통신 채널, 최대 동기화 노드 개수, 웨이크 업 패턴 등을 설정할 수 있다. ⑧의 버튼을 누르게 되면 ③, ④, ⑥을 이용하여 나머지 구성 변수들이 자동 생성되며, 이는 ⑦에서 확인 할 수가 있다.

⑨을 누르게 되면 ③, ④, ⑥, ⑦을 통해 생성된 PCR 정보와 CANdb에 저장 되어있는 ECU, 메시지 정보를 이용하여 그림 7과 같이 본 프로그램의 최종 목표인 FIBEX를 자동 생성하게 된다. ⑩은

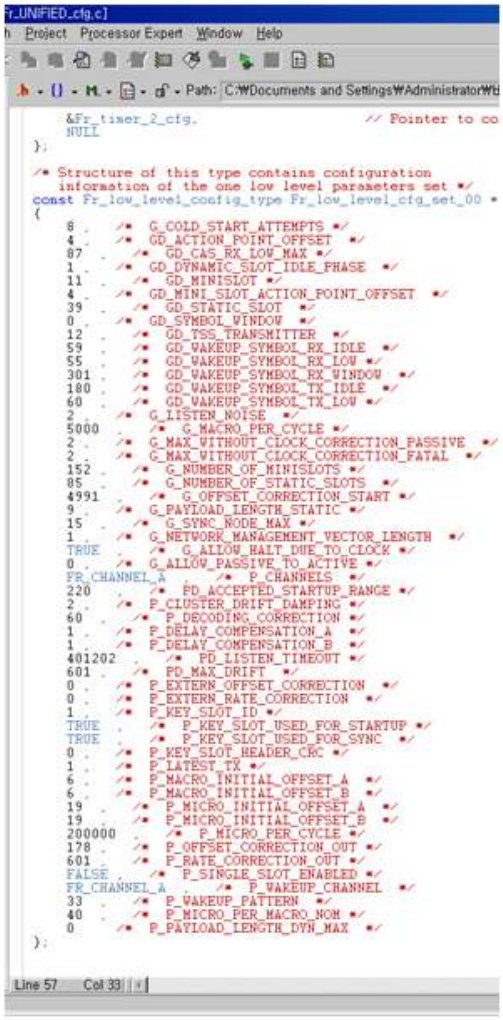

그림 8. FIBEX 자동 생성 프로그램의 PCR 자동 생성 결과

Fig. 8 Result of PCR automatic generation

생성된 PCR을 펌웨어(firmware)에 입력하기 위해 그림 8과 같은 소스코드 형태로 출력하게 된다.

V. 결 론

본 논문에서는 섀시 시스템에서 사용되는 CANdb를 이용하여 FIBEX를 자동 생성하는 알고리즘을 개발하였다. 기존의 CAN 프로토콜은 차량용 네트워크로 오랜 시간 동안 사용되어 왔기 때문에 설계자들에게 상대적으로 용이하며 많은 노하우가 제공되고 있다. 하지만 최근에 개발된 FlexRay

의 경우 아직 개발 단계에 있고 널리 보급되지 않았기에 CAN 프로토콜 설계자가 FlexRay 프로토콜을 이해하고 설계하기란 상당한 어려움이 존재한다. 특히, CAN 프로토콜에 정의되어 있는 데이터베이스들을 FlexRay 데이터베이스로 이동시키기에는 상당한 시간과 설계자의 많은 노력이 요구된다. 또한, FlexRay는 time-triggered 방식을 사용하기 때문에 사전에 플랫폼 구성 변수를 미리 계산하여 결정해줘야 하는데, 이는 상당히 복잡한 계산식으로 이루어져 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 기존에 설계된 CANdb를 이용하여 FlexRay의 데이터 및 정보를 교환하기 위한 FIBEX를 자동 생성하는 프로그램을 개발함으로써 다음과 같은 결과를 얻을 수 있었다.

첫째, CANdb에 의해 생성된 FlexRay 플랫폼 구성 변수는 CANdb 메시지의 생성 주기에 최대공약수를 이용하여 구하였으며, 주기적 데이터는 정적 구간으로, 비주기적 데이터는 동적 구간으로 배치하여 T_{SS}, T_{DS}, T_{NIT}를 구할 수 있었고, 이를 기반으로 나머지 플랫폼 구성 변수를 자동으로 생성할 수 있었다.

둘째, CANdb에서 정해진 ECU와 메시지를 FlexRay로 이동시키고, 자동 생성된 플랫폼 구성 변수를 FIBEX 구조에 맞게 배치함으로써, CANdb를 기반으로 하는 FIBEX를 생성할 수 있었다.

본 논문에서 개발된 FIBEX 자동 생성 프로그램은 FlexRay 네트워크의 성능 개선의 목적보다는 기존의 차량용 네트워크 설계자의 기술적 진입을 용이하게 하기 위한 목적으로 개발되었다. 따라서, 실차 CANdb를 이용하여 개발된 자동 생성 알고리즘이 효과적인지를 검증해야 할 필요가 있다. 또한, FIBEX 자동 생성 프로그램의 성능을 검증하기 위하여 FlexRay 네트워크 실험 모델을 제작하여 자동 생성된 FIBEX가 정상적으로 동작하는 지에 대한 검증이 필요하다. 마지막으로, 개발된 알고리즘이 보다 실용적이기 위해서는 데이터의 길이, 갯수, 송수신 ECU 간의 관계, 메시지 생성 주기 등을 고려한 플랫폼 구성 변수의 최적화와 FlexRay 네트워크의 이용률 최적화에 대한 연구가 필요하다.

참고문헌

[1] N. Navet, Y. Song, F. Simonot-lion, C. Wilwert, “Trends in Automotive Communication Systems,” Proceeding of the IEEE, Vol. 93, No. 6, pp.1204-1224, 2005.

[2] H.K. Kim, H.Y. Jung, J.H. Park, "Vehicle Detection for Adaptive Head-Lamp Control of Night Vision System," Journal of IEMEK, Vol. 6, No. 1, pp.8-15, 2011 (in Korean).

[3] S. Jung, T.C. Hsia, "Intelligent technique application for autonomous lateral position control of an unmanned 4 wheel steered snow plow robotic vehicle," Journal of IEMEK, Vol. 6, No. 3, pp. 132-138, 2011 (in Korean).

[4] K.H. Park, "Global Chassis Control Technologies for Intelligent Vehicles," Journal of KSPE, Vol. 23, No. 9, pp.15-22, 2006 (in Korean).

[5] K. Jang, I. Park, J. Han, K. Lee, M. Sunwoo, "Design framework for FlexRay network parameter optimization," International Journal of Automotive Technology, Vol. 12, No. 4, pp.589-597, 2011.

[6] I. Park, M. Sunwoo, "FlexRay network parameter optimization method for automotive applications," IEEE Transactions on Industrial Electronics, Vol. 58, No. 4, pp.1449-1459, 2011.

[7] H. Zeng, M. Natale, A. Ghosal, "Schedule optimizationof time-triggered systems communication over the FlexRay static segment," IEEE Transactions on Industrial Informatics, Vol. 7, No. 1, pp.1-17, 2011.

[8] E.G. Schmidt, K. Schmidt "Message scheduling for the FlexRay protocol: the dynamic segment," IEEE Transactions on Vehicular Technology, Vol. 58, No. 5, pp.2160-2169, 2009.

[9] E. Armengaud, A. Steininger, M. Horauer, "Towards a systematic test for embedded automotive communication systems," IEEE Transactions on Industrial Infomatics, Vol. 4, No. 3, pp.146-155, 2008.

[10] F. Sethna, E. Stipidis, F.H. Ali, "What lessons can controller area networks learn from FlexRay," Proceedings on IEEE Conference of Vehicle Power and Propulsion, pp.1-4, 2006.

[11] G. Leen, D. Hefferman, A. Dunne, "Digital networks in the automotive vehicle," Computer & Control Engineering Journal,

Vol. 10, No. 6, pp.257-266, 1999.
[12] R. Cummings, "Easing the transition of system designs from CAN to FlexRay," SAE Technical Paper 2008-01-0804, 2008.
[13] J.S. Yang, J.H. Park, S. Lee, K.C. Lee, G.H. Choi, "Study on Automatic Generation of Platform Configuration Register in FlexRay Protocol," Journal of IEMEK, Vol. 7, No. 1, pp.41-52, 2012 (in Korean).
[14] P. Mishra, "Distributed control system development for FlexRay-based systems," Proceedings on SAE World Congress & Exhibition, Vol. 114. No. 2005-01-1279, pp.240-250, 2005.
[15] http://www.vector.com/vi_fibex_en.html
[16] Vector, DBC File Format Documentation, 2007.

저 자 소 개

박 지 호 (Ji-Ho Park)

2011년 부경대 제어계측공학과 학사.
2013년 부산대 기계공학부 석사.
현재, LG전자 연구원.
관심분야: 차량용 네트워크.
Email: joe@pusan.ac.kr

이 석 (Suk Lee)

1984년 서울대 기계공학과 학사.
1985년 펜실바니아주립대 석사.
1990년 펜실바니아주립대 박사.
현재, 부산대학교 기계공학부 교수.
관심분야: 산업용 네트워크, 차량용 네트워크, 센서 네트워크.
Email: slee@pusan.ac.kr

이 경 창 (Kyung-Chang Lee)

1996년 부산대 생산기계공학과 학사.
1998년 부산대 생산기계공학과 석사.
2003년 부산대 지능기계공학과 박사.
현재, 부경대학교 제어계측공학과 교수.
관심분야: Cyber-Physical System, 차량용 네트워크, 로봇용 네트워크.
Email: gclee@pknu.ac.kr