

굴절 차량을 위한 전차륜 조향 시스템 전자 제어 장치 진단 프로그램 개발

이효걸*, 정기현*, 최경희**, 박태원***, 문경호****, 김상중*****

*아주대학교 전자공학과

**아주대학교 정보통신전문대학원

***아주대학교 기계공학과

****한국철도기술연구원

*****계명문화대학 컴퓨터학부

e-mail:hyogeollee@gmail.com

Development of the All Wheel Steering ECU Diagnostic Program for an Articulated Vehicle

Hyo-geol Lee*, Ki-hyun Chung*, Kyung-hee Choi**, Tae-won Park***, Kyung-ho Moon****, Sang-jung Kim*****

*Division of Electronics Engineering, Ajou University

**Graduate School of Information and Communication, Ajou University

***Division of Mechanical Engineering, Ajou University

****Korea Railroad Research Institute

*****Computer Faculty, Keimyung College

요 약

저상 굴절 차량에는 전차륜 시스템이 요구되며, AWS ECU는 전차륜 시스템의 핵심적인 역할을 하는 전자 제어 장치이다. 실제 차량 주행을 위해서는 차량에 따른 ECU의 설정 값 변경이 필요하며, 현재 ECU의 동작 상태를 점검할 수 있는 기능이 요구된다. 이러한 기능을 수행하기 위하여 ECU에 서비스 루틴을 추가하고, 진단 프로그램을 개발하여 성능을 평가하였다.

1. 서론

저상 굴절 차량은 차량의 길이가 길기 때문에, 앞 차륜에 의한 전통적인 조향 방식을 사용할 경우 차량의 회전반경이 커져 국내 도로 여건에 맞지 않게 되며, 차량이 정거장에 진입 시 정밀 정차가 어려워지는 문제점이 발생한다. 따라서 1축 차륜과 속도 등의 상태를 바탕으로 3축 차륜을 조향하는 전차륜 시스템이 필요하며, AWS ECU는 전차륜 시스템의 핵심적인 역할을 하는 전자 제어 장치이다[1].

실제 차량 주행을 위해서는 현재 차량의 특성에 따라 ECU의 설정 값을 변경할 수 있는 기능이 필수적이다. 정상적인 주행을 위해서 정확한 wheel alignment이 이루어져야 하며, 조향 알고리즘에 사용되는 파라미터, 예러 판단 기준 등도 차량 특성에 따라서 바뀌어야만 한다.

또한 현재 ECU의 동작 상태를 점검할 수 있는 기능이 필요하다. 주행 중의 ECU가 인식한 축의 각도, 계산된 요구 조향각, 유압 시스템 압력, 속도 등의 정보나 ECU에서 발생된 예러 정보를 확인할 수 있어야 한다.

이에 따라 차량에 따른 ECU의 설정 값을 변경하고, ECU의 동작 상태를 점검할 수 있도록 ECU에 서비스 루틴을 추가하고, 진단 프로그램을 개발하였다.

2. 진단 프로그램의 요구사항

진단 프로그램은 크게 ECU의 설정 값을 변경하는 기능, startup을 통하여 wheel alignment를 하는 기능으로 나눌 수 있다. 각 기능에서 진단 프로그램의 요구사항을 간략히 정리하면 표 1과 같다.

<표 1> 진단 프로그램 요구사항

ECU의 값을 읽어 올 수 있다.	Display 값을 읽어 올 수 있다.
	디지털 입출력 값을 읽어 올 수 있다.
	아날로그 입력 값을 읽어 올 수 있다.
	알고리즘 파라미터 값을 읽어 올 수 있다.
	예러 리스트를 읽어 올 수 있다.
	Fail/Safe 파라미터를 읽어 올 수 있다.
	ECU Identification 값을 읽어 올 수 있다.
	CAN 파라미터를 읽어 올 수 있다.
ECU의 파라미터를 변경할 수 있다.	Encoder 값을 읽어 올 수 있다.
	값을 주기적으로 읽어 올 수 있다.
	알고리즘 파라미터를 변경할 수 있다.
	Fail/Safe 파라미터를 변경할 수 있다.
Startup 과정을 수행할 수 있다.	외부 RTC를 변경할 수 있다.
	CAN 파라미터를 변경할 수 있다.
	Encoder-Potmeter 값을 변경할 수 있다.
Startup 과정을 수행할 수 있다.	Front Axle에 대한 Startup 과정을 수행할 수 있다.

	Rear Axle에 대한 Startup 과정을 수행할 수 있다.
	Startup 설정 값을 ECU에 전송할 수 있다.
ECU의 외부 저장 메모리를 초기화 할 수 있다.	시스템 설정을 초기화 할 수 있다.
	에러 리스트를 초기화 할 수 있다.
ECU의 값을 기록할 수 있다.	주기적으로 읽은 Display 값을 파일로 저장할 수 있다.
ECU의 설정 값을 기록할 수 있다.	ECU의 모든 설정 값을 파일로 저장할 수 있다.
ECU의 설정 값을 복구할 수 있다.	ECU의 모든 설정 값을 파일로부터 복구할 수 있다.
사용하기 편한 User Interface를 가진다.	Graphic User Interface를 가진다.
	진단 프로그램 관련 설정을 쉽게 바꿀 수 있다.

3. 진단 프로그램 프로토콜

3.1 KWP2000

진단 프로그램에 사용된 프로토콜은 차량 내부 ECU 진단을 위해 사용되는 프로토콜인 KWP2000 기반으로 한 프로토콜이다. 진단 프로그램에서는 KWP2000에서 physical layer와 data link layer를 변경하여 사용하였다.

진단 프로그램을 지원하기 위해 ECU에 구현된 프로토콜은 표 2와 같다.

<표 2> 진단 프로그램 프로토콜

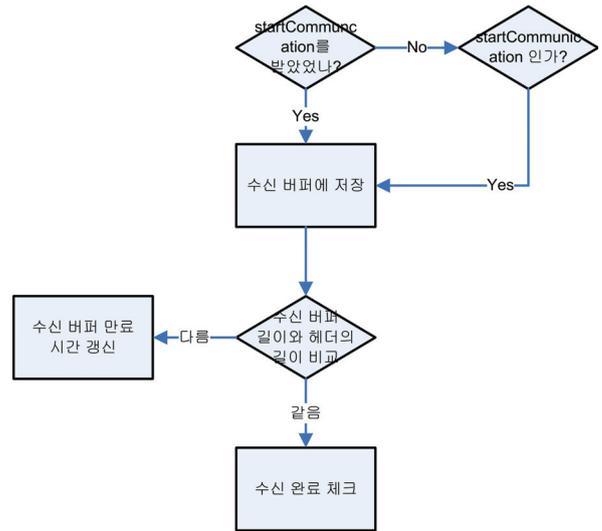
Physical	RS-232 Serial line
	115.2 kbps
Data Link Layer Service	StartCommunication Service
	StopCommunication Service
Diagnostic Service	startDiagnosticSession Service
	readEcuIdentification Service
	readDataByLocalIdentifier Service
	writeDataByLocalIdentifier Service
	testerPresent Service

3.2 진단 프로그램 프로토콜 서비스 구현

진단 프로그램의 request는 ECU의 인터럽트 핸들러에서 처리되며, request message가 일정 시간 내에 모두 수신되지 않을 경우에 해당 request를 포기하도록 구현되었다.

(a) 인터럽트 핸들러

ECU의 기능이 동작하는 도중에도 진단 프로그램으로부터 request를 수신하기 위해서는 ECU의 UART 인터럽트 핸들러에 request를 수신하는 루틴을 작성하여야 한다.



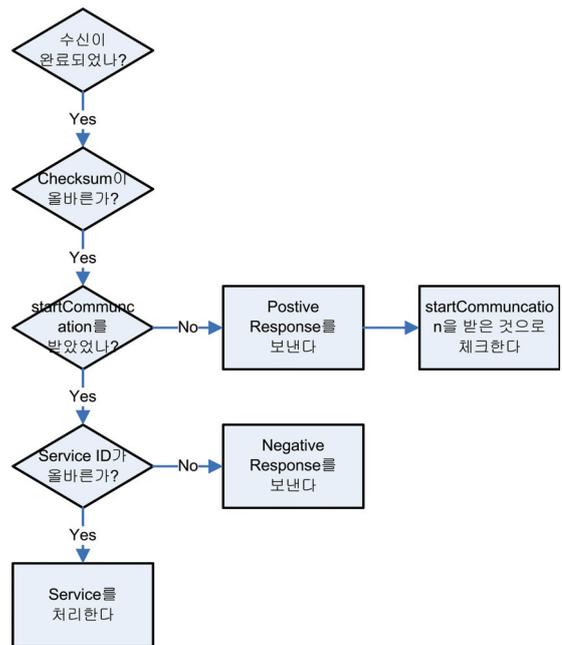
(그림 1) ECU 인터럽트 핸들러

인터럽트 핸들러의 행동은 이전에 startCommunication 요청을 받았는지 여부에 따라 달라지며, 그림 1은 인터럽트 핸들러의 개략적인 flow를 보여준다.

(b) 타이머

타이머에서는 startCommunication 요청을 받아 통신이 시작되었고, 수신 버퍼가 비어있지 않은데 수신이 완료되지 않았다면, 수신 버퍼의 만료 시간을 감소시킨다.

수신 버퍼의 만료 시간이 0이 되면 수신 버퍼를 초기화 시켜서, 현재 수신 내용을 포기한다.

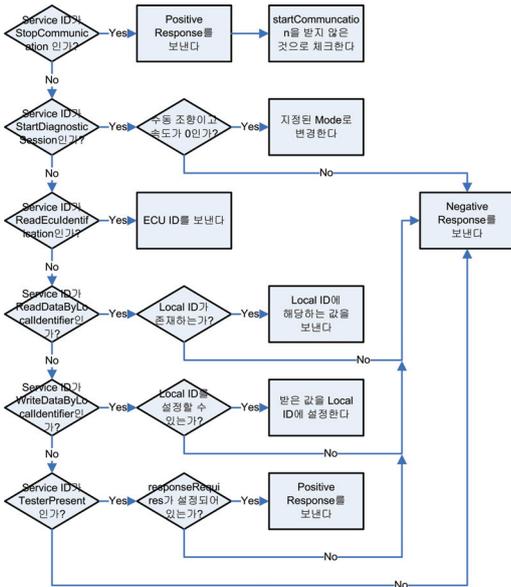


(그림 2) 서비스 루틴 - Checksum, ID 검사

(c) 서비스 루틴

서비스 루틴은 ECU의 매 동작 loop마다 수행된다. Tester의 request를 인터럽트 핸들러에서 처리하므로 서비스 처리 루틴에 진입한 시점에서 수신이 완료되지 않는 경우가 있을 수 있다.

서비스 루틴의 행동은 이전에 startCommunication 요청을 받았는지 여부에 따라 달라지며, 그림 2, 3은 서비스 루틴의 개략적인 flow를 보여준다.



(그림 3) 서비스 루틴 - Service 처리

4. 진단 프로그램 개발

4.1 개발 환경

진단 프로그램은 Python과 wxPython으로 개발되었다. Python의 경우 재 컴파일 없이 여러 Platform에서 사용 가능하고, 개발 속도가 빠르기 때문에 진단 프로그램을 개발하는데 사용되었다. wxPython은 wxWindow의 Python binding으로 cross platform GUI application을 개발하는데 용이하다.

진단 프로그램은 Python과 wxPython을 기반으로 개발되었기 때문에 Windows 이외에 UNIX 계열 OS 등 여러 환경에서 동작이 가능하다.

진단 프로그램 개발에 사용된 환경은 표 3과 같다.

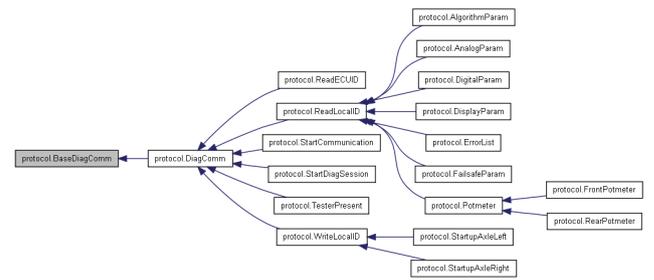
<표 3> 진단 프로그램 개발 환경

- | |
|--|
| <ol style="list-style-type: none"> 1. Python 2.6.4 2. wxPython 2.8.10.1 3. pyserial 2.5-rc2 |
|--|

4.2 진단 프로그램 구성

진단 프로그램은 크게 프로그램 실제 logic 부분과 화면에 보여지는 display 부분으로 나눌 수 있다. 실제 Logic 부에는 통신을 담당하는 protocol 부분과 이외 동작을 담당하는 model 부분이 있다.

Protocol 부분은 실제 ECU와 통신하는 protocol에 관련된 구현이다. Protocol 패키지의 주요 클래스인 BaseDiagComm 클래스의 상속 다이어그램은 그림 4와 같다.



(그림 4) BaseDiagComm 클래스 상속 다이어그램

Model 부분은 display가 보낸 message에 대한 logic 처리를 하는 부분이다. 사용자는 display 부분을 사용하며 event를 보내고 event 발생 시에 display는 model 부분이 logic을 처리하도록 message를 보낸다. Model 부분은 logic을 처리 후 응답 message를 다시 display로 보낸다. Logic 부분과 display 부분은 Publish/Subscribe 패턴을 사용하는 wx.lib.pubsub를 이용하여 통신을 하도록 구현되었다. Model 부분에서는 주로 protocol 패키지를 사용하여 ECU와 통신 후에 응답을 display로 보내게 된다.

4.3 Startup 과정

Startup은 ECU 초기 장착 시 차량의 wheel alignment를 수행하는 과정으로 정상적인 주행을 위해서 반드시 필요하다. 그림 5는 startup 과정의 개략적인 flow를 보여준다.

4.4 설정 값 검증

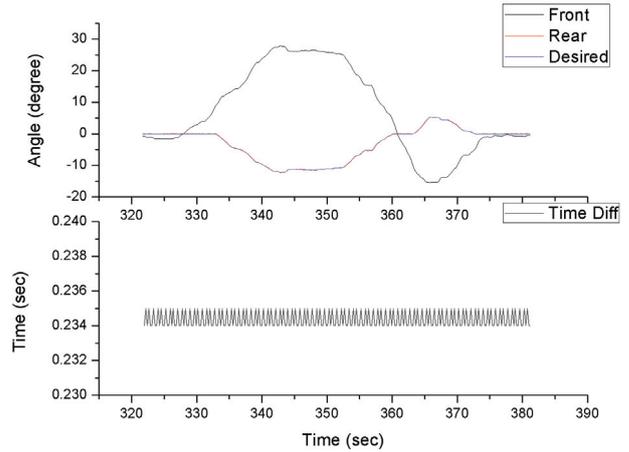
진단 프로그램에서 설정한 값을 ECU에 전송하기 전에 그 값이 적합한지 검사하는 과정이 필요하다. ECU에 전송하여 ECU 구동에 영향을 주는 값들은 알고리즘 파라미터 값들과 Startup 값들인데, 모두 ECU 구동에 크게 영향을 주는 부분이라 값의 유효성 검사는 반드시 필요하다.

값의 물리적인 한계를 기준으로 조건을 설정하였으며,

표 4와 같다.

<표 4> 설정 값 유효성 판단 조건

1. Startup 값은 0.2 ~ 4.8 V 값을 가진다.
2. Startup 각 항목의 track A, track B 합은 5 V 정도이다.
3. Startup 항목의 track A 값을 기준으로 left < center < right 의 관계이다.
4. Startup 항목의 track B 값을 기준으로 left > center > right 의 관계이다.
5. 알고리즘 Parameter Delay에서 Start time < End time 의 관계이다.
6. 알고리즘 Parameter에서 Hydro request valve upper > Hydro request valve bottom 의 관계이다.
7. 알고리즘 Parameter에서 Hydro request valve upper < 4.1 V 의 관계이다.
8. 알고리즘 Parameter에서 Hydro request valve bottom > 0.0 V 의 관계이다.



(그림 6) 진단 프로그램으로 수집한 주행 데이터

그림 6은 시험 주행 시 진단 프로그램을 통하여 수집한 ECU의 주행 데이터이며, 각 측정 시간차가 235 ms로 유지되는 것을 확인할 수 있다.

6. 결론

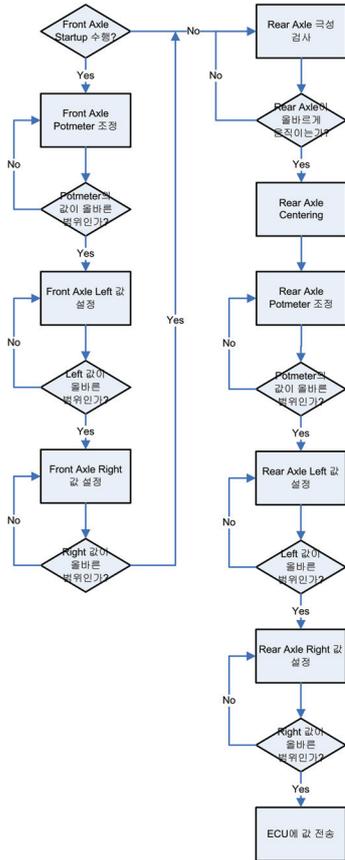
저상 굴절 차량은 전차륜 시스템이 필요하며, AWS ECU는 전차륜 시스템의 핵심적인 역할을 하는 전자 제어 장치이다.

AWS ECU는 차량의 특성에 따라 ECU의 설정 값을 변경하고, ECU의 상태를 점검할 수 있는 기능이 필요하며, 이에 따라 진단 프로그램을 개발하였다.

실제 시험 저상 굴절 차량에서 개발한 진단 프로그램을 사용하여 AWS ECU를 설정하고, 주행 데이터를 수집할 수 있었다.

참고문헌

- [1] 김기정, 정기현, 최경희, 이수호, 박태원, 문경호, "굴절 차량을 위한 전차륜 조향 시스템 전자제어 장치 개발", 한국철도학회 2008년도 추계학술대회논문집, 2008. 11
- [2] ISO/DIS 14230-1, "Road Vehicles-Diagnostic Systems : Keyword Protocol 2000 - Part 1: Physical Layer", 1996
- [3] ISO/DIS 14230-2, "Road Vehicles-Diagnostic Systems : Keyword Protocol 2000 - Part 2: Data Link Layer", 1996
- [4] ISO/DIS 14230-3, "Road Vehicles-Diagnostic Systems : Keyword Protocol 2000 - Part 3: Implementation", 1996
- [5] wxPython API Reference
- [6] Noel Rappin, Robin Dunn, "wxPython in Action", Manning, 2006



(그림 5) Startup 과정

5. 시험 평가

시험 저상 굴절 차량에 개발한 ECU를 장착하고, 개발한 진단 프로그램을 통하여 startup 과정을 수행하였다. Startup을 마친 시험 차량으로 바른 직진 주행을 할 수 있었다.