

# 열차간 통신기반 열차자율주행제어시스템의 주행안전기능 설계 및 구현: Part I

채성윤\*, 이병훈\*, 최현영\*, 방준호\*

## Design and Implementation of Safety Functions of Communication-Based Autonomous Train Control System: Part I

Sung-Yoon Chae\*, Byung-Hun Lee\*, Hyeon-Yeong Choi\*, June-Ho Bang\*

### 요 약

본 논문에서는 열차간 통신기반 열차자율주행제어시스템의 ATP 차상장치 주행안전기능을 설계 및 구현한다. 열차자율주행제어시스템의 주행안전기능은 열차 간의 최소한의 안전 간격을 유지하기 위해서 열차의 위치를 결정하고, 이동권한 계산, 속도프로파일을 생성하고, 열차의 속도를 속도프로파일 내에서 감시하는 기능이다. 이를 위해 안전성 및 신뢰성있는 소프트웨어를 제작하기 위한 소프트웨어 개발툴인 SCADE를 활용한다. 본 논문에서는 열차자율주행제어시스템 주행안전기능의 요구사항에 따라 SCADE를 이용하여 주행안전기능을 모델링하고, 모델에 대한 테스트를 통해 기능을 검증한다. 검증된 모델을 소스코드로 생성하고, 테스트환경을 구축하여 정상 동작 결과물을 확인한다.

**키워드** : 열차자율주행제어시스템, 주행안전기능, ATP, SCADE, 시뮬레이션

**Key Words** : autonomous train control system, safety functions, onboard ATP, SCADE, simulation

### ABSTRACT

In this paper, we design and implement the safety functions of the T2T communication-based autonomous train control system. The safety functions of the autonomous train control system is to determine the position of the train, to create the speed profiles, and to monitor the speed of the train in the speed profile in order to maintain the minimum safety interval between trains. To this end, we use SCADE, a software development tool for producing safe and reliable software. We model the safety functions according to the system requirement specifications and perform model-specific tests of the safety functions implemented using SCADE. We build a test environment similar to the prototype of ATP onboard equipment and test the source code generated through SCADE.

### I. 서 론

본 논문에서는 열차간 통신기반 열차자율주행제어 시스템(Autonomous Train Control System: ATCS)의

주행안전기능을 SCADE(Safety-Critical Application Development Environment)를 이용하여 구현한다. 기존의 열차주행제어시스템인 CBTC(Communication based Train Control) 시스템<sup>[1,2]</sup>은 지상

※ 본 연구는 한국철도기술연구원 주요사업의 연구비 지원으로 수행되었습니다.

•° First and Corresponding Author : Korea Railway Research Institute, unichae@krii.re.kr, 정희원

\* Korea Railway Research Institute, {bhlee85, hchoi, jhbang}@krii.re.kr,

논문번호 : 201911-304-D-RU, Received November 19, 2019; Revised November 21, 2019; November 23, 2019

ATP(Automatic Train Protection)로부터 이동권한 및 속도프로파일을 수신하여 열차의 주행을 제어하는 반면, 열차자율주행시스템은 지상 시스템 없이 열차 스스로 이동권한, 경로, 속도를 결정하는 열차주행제어 시스템이다. 이를 통해 지상의 설비를 최소화하고, 열차간 통신을 통해 주행 정보를 교환하여 운영의 효율성을 높일 수 있다. 기존의 CBTC 방식의 열차주행제어시스템과 비교하여, 열차간 통신기반 열차자율주행제어시스템<sup>[3]</sup>은 선행열차의 위치 및 속도에 기반하여 간격을 제어한다. 선행열차의 위치에서 비상제동 거리를 고려하여 안전거리를 유지하기 때문에 기존의 시스템보다 열차 주행 간격을 줄일 수 있고, 이를 통해 효율적인 운영이 가능하다. 열차자율주행제어시스템의 ATP(Autonomous Train Protection) 차상장치의 주행안전기능은 열차의 위치를 결정하고, 속도프로파일을 계산하여 선행열차와의 안전거리를 침범하지 않게 열차의 속도를 제어하는 기능을 수행한다. 이를 위해서 각 열차의 ATP 차상장치는 열차의 안전한 이동을 위한 선로 및 선로전환기 자원을 요청하고, 자원관리자는 열차의 자원 요청을 처리하여 이미 점유된 자원이 중복되어 다른 열차에 의해 점유되는 것을 방지하는 기능을 수행한다.

열차자율주행제어시스템의 주행안전기능은 열차의 안전한 이동을 담당하는 소프트웨어 기능이기 때문에 설계 및 구현 시 생성된 모델 및 소스코드에 대한 높은 신뢰성을 보장하여야 한다. 본 논문에서는 SCADE 개발도구를 이용하여 주행안전기능을 모델링하고, 모델에 대한 테스트를 통해 기능을 검증한다. 최종적으로 검증이 끝난 모델은 SCADE-kcg를 이용하여 C언어 기반 소스코드로 생성한다.

본 논문의 구성은 다음과 같다. 2장은 관련연구를 기술하고, 3장에서는 열차자율주행제어시스템의 구조 및 컴포넌트 별 핵심 기능을 설명하고, ATP 차상장치의 주행안전기능을 설계 및 구현, 테스트한다. 4장은 결론이다.

## II. 관련연구

CBTC 시스템<sup>[1,2]</sup>은 무선통신기반 열차주행제어시스템으로 moving block 기반의 간격제어를 통해 열차운행의 효율성을 높인 시스템이다. 차상 ATP는 주기적인 통신을 통해 위치, 속도 등의 정보를 지상 ATP로 보고한다. 지상 ATP에서는 수집된 열차의 위치, 속도를 기반으로 각 열차의 이동권한, 속도프로파일을 생성하여 전달한다. 열차는 지상 ATP로부터 수신한

정보를 이용하여 안전한 운행을 보장받는다.

SCADE는 안전성 및 신뢰성이 보장된 제어 어플리케이션을 개발하기 위한 소프트웨어 개발도구로, ISO26262, EN50128 등 표준 기반 안전 요구사항을 만족하고, 주로 산업 제어설비, 비행 제어 소프트웨어, 국방 소프트웨어, 철도 소프트웨어 등에서 이용된다<sup>[4]</sup>. SCADE는 소프트웨어 사양명세, 설계, 모델 개발 및 소스코드 생성까지 소프트웨어 개발 전반적인 부분에 활용할 수 있고, SCADE를 통해 생성된 소스코드는 항공표준 DO-178B levelA 등의 안전 표준에 대한 인증을 받을 수 있다. 특히 모델 개발 단계에서 정형기법<sup>[5]</sup>에 기반한 모델 검증이 가능하고, 모델과 생성된 소스코드 간의 일치성을 보장하기 때문에 개발기간을 단축시킬 수 있다. 특히 SCADE-kcg는 EN50128<sup>[6]</sup> 표준이 명시한 철도신호시스템의 신뢰성, 안전성에 대해서 SIL(Safety Integrity Level)3/4 수준의 인증된 소스코드를 생성한다. 따라서, 모델 수준에서 검증이 완료된 경우, 이를 통해 생성된 소스코드에 대한 추가적인 검증 및 수정이 불필요하여 개발 기간을 단축시킬 수 있다.

## III. 본 론

### 3.1 시스템 개요

열차자율주행제어시스템은 그림 1과 같이 지상(wayside)과 차상(onboard) 시스템으로 구성된다. 지상시스템은 노선 전체의 열차운행 상황 모니터링을 위한 ATS 관제, 선로 및 선로전환기 자원의 점유를 기록하는 자원관리자, 선로전환기를 컨트롤하는 object controller 등의 서브시스템으로 구성된다<sup>[3]</sup>.

차상시스템은 열차의 이동권한, 속도 등을 관리하

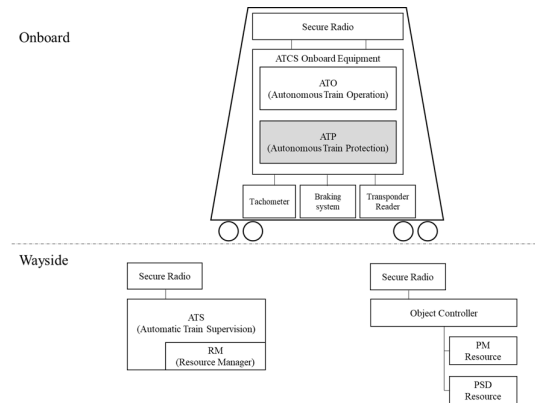


그림 1. 열차자율주행제어시스템 구조  
Fig. 1. ATCS architecture

는 ATP 차상장치, 열차의 운행 경로 설정, 운행 속도 조절 등의 기능을 담당하는 ATO 차상장치, 열차 및 지상시스템과의 통신을 위한 무선통신시스템, 타코미터, 제동시스템, 태그리더기 등의 서브시스템으로 구성된다. ATP 차상장치에서 열차의 위치, 속도, 이동권한을 기반으로 계산된 열차의 속도프로파일을 이용하여 ATO 차상장치는 다음 목적지까지의 운행 및 플랫폼 정차, 출입문 제어를 위한 허용 속도프로파일을 생성한다.

본 논문은 열차자율주행제어시스템의 서브시스템인 ATP 차상장치의 주행안전기능 설계 및 구현에 대해 기술한다. ATP 차상장치의 주행안전기능은 열차의 안전한 운행을 위해 ATP 차상장치에서 담당하는 핵심 기능 중 하나이다. ATP 차상장치는 현재 열차의 위치, 속도, 점유된 자원, 이동권한(Movement Authority: MA) 및 선행 열차의 위치, 자원 점유 현황, 속도 등을 고려하여 현재 열차의 상용제동/비상제동 속도프로파일을 생성한다<sup>7,8,9)</sup>. 생성된 속도프로파일을 기반으로 열차가 주행 중에 속도프로파일의 한계값을 초과할 경우 ATP 차상장치에서는 추진을 차단하고, 초과한 속도프로파일에 맞는 제동 명령을 제동시스템으로 전달한다. 이를 통해 열차의 속도는 항상 선행열차와 안전간격을 침범하지 않는 한계점 이내로 제어된다.

### 3.2 주행안전기능 설계

열차자율주행제어시스템 ATP 차상장치의 주행안전기능은 열차 위치결정, 제동 곡선 계산, 속도프로파일 생성, 추진 차단, 제동 체결 기능 등의 세부 기능모듈로 분류한다. 그림 2는 열차자율주행제어시스템 ATP 차상장치의 전체 기능 및 입출력 인터페이스에 대한 정의이다. 본 논문에서 설계 및 구현한 주행안전기능은 그림 3과 같이 세부 기능모듈로 구분할 수 있다. 열차위치결정기능은 열차의 차상 서브시스템인 태그리더기와 타코미터의 입력을 기반으로 열차의 현재 위치를 계산한다. 열차의 위치는 태그리더기에 의해 입력된 태그의 ID 및 타코미터로부터 계산된 이동거리로 표현된다. 또한, 단위 시간당 이동한 거리를 기반으로 열차의 현재 속도를 측정한다. 자원 및 이동권한 관리 모듈은 열차의 위치 및 속도를 이용하여 이동권한 및 제동 곡선을 계산한다. 이동권한은 열차가 안전하게 운행할 수 있게 허용된 최대 거리 및 속도이다. 이동권한의 한계점에서 열차는 주어진 속도 내로 운행할 수 있다. 따라서, 상용제동/비상제동 속도프로파일은 이동권한 한계점에서 열차의 현재 위치까지 열차 제동 감속도를 이용하여 계산한다. ATP 차상장치의 제동 명령은 열차의 속도가 속도프로파일의 한계값을 초과할 경우 상용제동/비상제동 속도프로파일

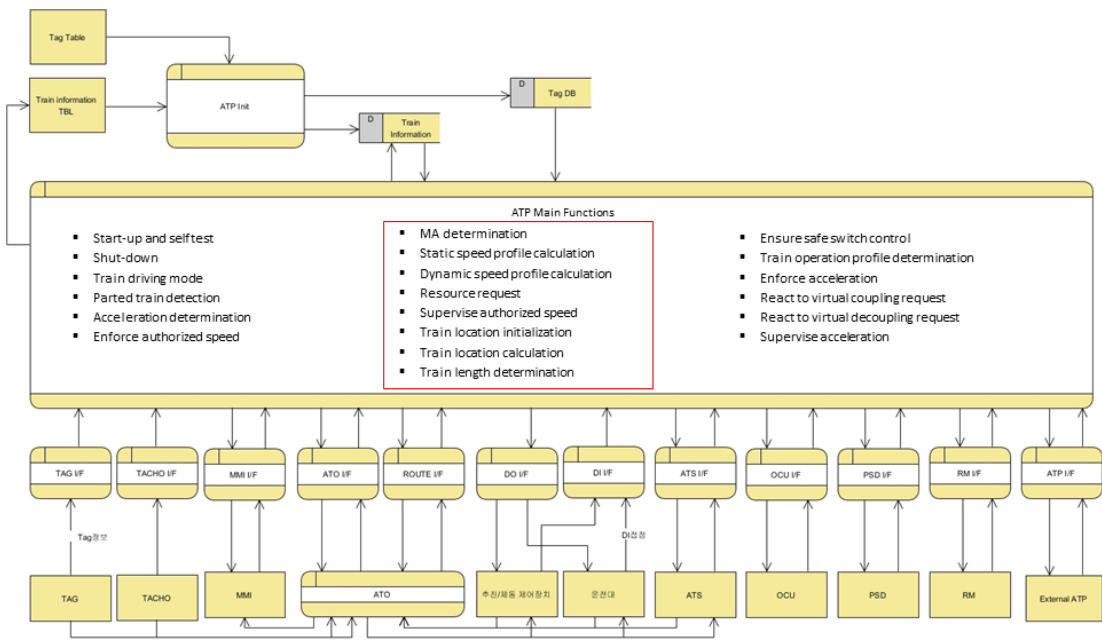


그림 2. 열차자율주행제어시스템 주행안전기능 입출력 인터페이스  
Fig. 2. ATCS safety functions and I/O interfaces

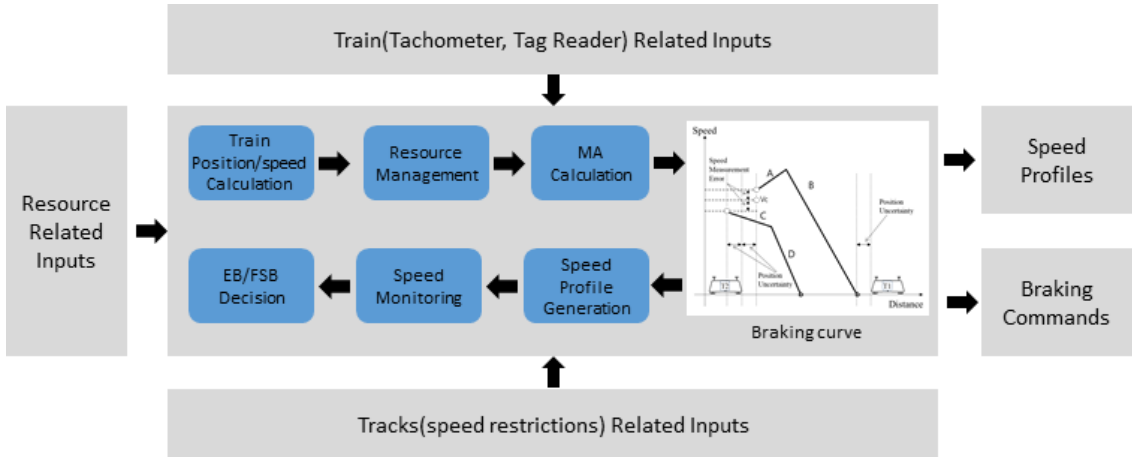


그림 3. 주행안전기능 세부 기능 및 동작과정  
Fig. 3. Safety sub-functions and functional procedure

에 해당하는 제동 명령을 제동시스템으로 전송한다. 본 논문에서는 열차 주행안전기능의 세부 기능모듈 중 열차위치결정, 자원 및 이동권한 관리 모듈에 대한 설계 및 구현을 기술한다.

열차 위치결정기능은 선로 상 열차의 위치를 판단하는 기능이다. 열차가 선로 상을 이동할 때 정확한 위치를 파악하지 못하면 비상 상황으로 인식하여 비상제동을 체결해야 한다. 열차의 위치는 선로 상에 일정한 간격으로 설치된 태그를 기준으로 파악할 수 있다. 그림 4와 같이 열차는 선로 상을 운행하면서 열차 하부에 설치된 태그리더기가 태그 위를 지나가면서 해당 태그의 ID를 수신한다. 또한, 열차가 운행하면서 회전하는 차축에 설치된 타코미터로부터 주행 거리를 수신한다. 태그리더기로부터 입력된 태그 ID와 차량 바퀴 회전수를 통해 계산된 거리로 열차의 현재 위치를 결정한다. 열차의 태그리더기로부터 입력된 태그

ID를 가장 최근에 입력된 태그 ID인 LRTI(Lasted Read Tag ID)로 저장하고, 타코미터로부터 수신한 누적이동거리를 단위시간당 이동거리(offset)로 변환하여 열차 전두부의 현재 위치를 (LRTI, offset) 형태로 저장한다. 열차 후미부의 위치는 전두부 위치에서 열차 길이를 제외하고 계산한다. 열차 위치결정기능은 태그리더기의 신호 수신 범위와 타코미터의 에러를 고려하여 보정하는 기능을 포함한다.

자원관리 기능은 ATP 차상장치에서 열차의 위치를 기반으로 열차의 안전한 운행을 위해 필요한 선로 및 선로전환기 자원에 대한 요청/점유 과정을 담당한다<sup>8)</sup>. 자원 요청/점유 과정은 그림 5와 같다. 열차는 점유된 자원 내에서 계산된 이동권한의 한계점까지 이동이 허용된다. 열차가 이동함에 따라 열차의 후미부 자원은 점유를 해제하고, 경로 상의 선로 및 선로전환기 자원에 대한 점유를 요청한다.

자원관리자는 요청된 자원에 대한 점유 및 소유자를 기록하는 DB로, 동일한 자원이 동시에 두개 이상의 열차에 의해 점유되는 것을 방지하는 기능을 수행한다<sup>7)</sup>. 자원 요청/점유 과정은 그림 5와 같이 진행된다. i) 선행열차의 이동한 거리만큼 점유된 자원 (a)의 해제 요청, ii) 자원해제 확인, iii) 열차간 위치 정보 공유, iv) 후행열차의 선로 및 선로전환기 자원의 점유 현황 확인 후 자원(b) 점유 요청, v) 자원 점유 확인. 자원 점유 절차가 완료되고, 점유된 선로전환기에 대해서 방향을 제어한다.

열차자율주행제어시스템에서 이동권한은 열차가 안전하게 이동할 수 있는 한계점 및 속도이다<sup>9)</sup>. 이동권한의 한계점은 자원관리자를 통해 점유한 선로 또

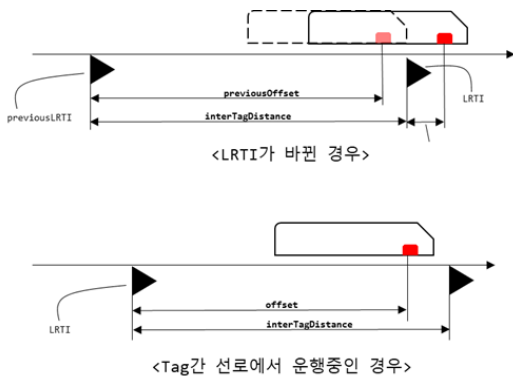


그림 4. 위치결정기능 동작과정  
Fig. 4. Train position determination procedure

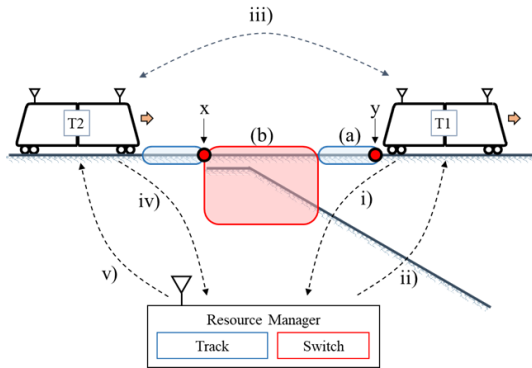


그림 5. 자원 요청/접수 과정  
Fig. 5. Resource request/response procedure

는 선로전환기 자원의 끝 지점으로 지정된다. ATP 차상장치는 점유된 자원 중에 선로전환기가 위치한다면 선로전환기의 방향을 열차의 경로와 일치하도록 제어한다. 제어가 완료되어 선로전환기의 방향이 열차의 경로와 일치하게 되면 이동권한의 한계점은 해당 선로전환기 자원을 포함하여 연장된다. 만약 점유된 자원 중 다수의 선로전환기가 포함되어 있다면 열차의 위치에서 가장 가깝고, 열차의 경로와 방향이 일치된 선로전환기부터 이동권한에 포함된다. 즉, 이동권한의 한계점은 열차의 전두부 위치로부터 연속되어 점유된 선로자원과, 열차의 경로와 일치된 방향의 선로전환기의 끝 지점으로 계산된다. 그림 6은 열차자율주행제어 시스템의 점유 자원, 이동권한 한계점, 안전마진에 대한 관계를 표시한다. 열차는 점유된 자원의 한계점 내에서 선행열차의 비상제동거리를 고려한 이동권한의 한계점 및 속도를 계산한다.

이동권한의 한계점이 계산되면, 이동권한 한계점까지의 MRSP(Most Restrictive Speed Profile)을 계산한다. MRSP는 열차의 임시제한속도, 열차의 최대속도, 선로의 임시제한속도, 선로의 영구제한속도를 기반으로 계산된 각 구간에서의 최소 속도값이다.

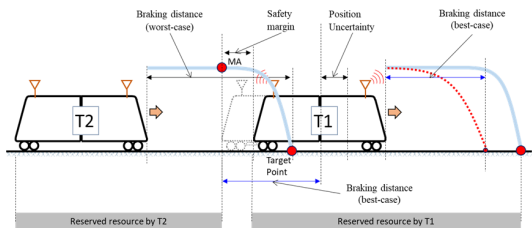


그림 6. 열차의 점유 자원 및 이동권한 계산 과정  
Fig. 6. Reserved resource and MA calculation procedure

### 3.3 주행안전기능 구현

SCADE는 안전성이 보장된 소프트웨어를 개발하는 틀로서 모델 구현시 동적 메모리 할당에 대한 제한이 있다. 따라서 본 연구에서도 이러한 제한사항을 고려하여 주행안전기능을 구현하였다. 열차자율주행시스템 ATP 차상장치 주행안전기능의 세부 모듈별로 SCADE 모델로 구현하였다. 구현된 열차 위치결정기능, 자원관리 및 이동권한 관리 기능 모델은 그림 7, 8과 같다.

그림 7과 같이 위치결정기능은 열차가 새로운 태그를 지나칠 경우 offset을 초기화하는 과정과 태그리더기로부터 새로운 입력이 없을 경우 offset을 갱신하는 기능으로 구현한다. 위치결정기능의 결과는 열차의 전두부 및 후미부의 태그 ID와 offset으로 출력된다.

그림 8과 같이 열차는 자원 및 이동권한의 한계점을 계산한다. 이동권한 한계점 계산과정은 다음과 같다.

- i) ATP 차상장치는 이동 경로를 기준으로 점유를 요청할 자원의 한계점을 다른 열차에 의해 점유되지 않고, 선로의 중단점과 이동권한 최대값을 고려하여
 
$$dRequestResource = \text{Min}(dReservedResourceByATP, dEndofLine, cMAX\_MA)$$
 로 계산한다.
- ii) ATP 차상장치는 계산된 값에 따라 필요한 자원을 자원관리자에게 요청한다. 자원관리자는 열차가 요청한 자원에 대한 점유 가능 여부를 열차로 송신한다. ATP 차상장치는 이를 수신하여 점유된 자원을 현재 열차의 전두부에서부터의 거리 (dReservedResource)로 계산한다.
- iii) ATP 차상장치는 dReservedResource 내의 자원

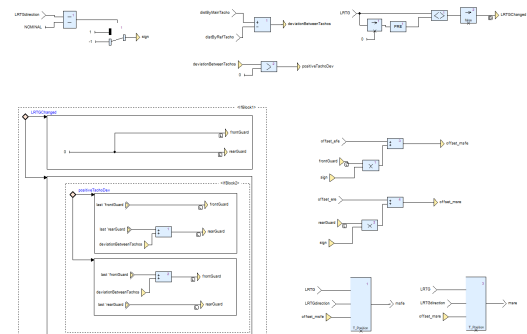


그림 7. 열차 위치결정기능 모델  
Fig. 7. Model of train position determination

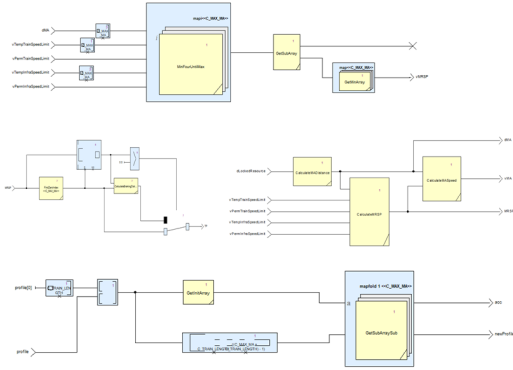


그림 8. 자원 및 이동권한 관리 기능  
Fig. 8. Model of resource and MA management

중 선로전환기가 있다면 선로전환기의 방향과 이동 경로의 방향을 비교한다. 선로전환기의 방향과 이동 경로의 방향이 다를 경우 열차의 점유 완료된 자원의 한계점(dLockedResource)을 선로전환기 전까지로 설정한다.

- iv) 선로전환기의 방향이 이동경로와 일치하지 않는 선로전환기를 제어한다.
- v) 선로전환기의 제어가 완료되었으면 dLockedResource를 dReservedResource로 설정한다.

위의 계산과정에 따라 자원의 한계점을 기반으로 이동권한의 한계점을 계산한다. 이동권한을 계산하기 위해서 cMAX\_MA 크기 만큼의 MRSP를 입력으로 받는다. MRSP는 4개의 정적 속도프로파일(영구 선로 제한속도, 임시 선로 제한속도, 영구 열차 제한속도, 임시 열차 제한속도)중 가장 작은 속도값으로 정해지며, 이때 열차의 길이를 포함시킨다. 제한속도가 낮은 값에서 높은 값으로 변하는 구간은 열차의 후미부가 낮은 제한속도 구간을 빠져나가기 전까지 열차의 전두부도 낮은 제한속도에 영향을 받는다. 따라서 제한속도가 낮은 값에서 높은 값으로 변하는 구간의 MRSP는 제한속도가 낮은 값이 열차의 길이만큼 연장되어 적용된다. 이를 구현하기 위해서 SCADE 모델에서는 정적 속도프로파일 중 최소값을 구한 속도프로파일(vSP)의 첫 번째 속도값인 vSP[0]을 열차의 길이(TRAIN\_LENGTH)만큼 연장하여 최초 입력값의 배열을 (vSP[0]\*TRAIN\_LENGTH, vSP) 로 확대한다. 이후 열차의 길이를 기준으로 단위거리(1m)마다 최소 속도값인  $\text{Min}(vSP[i], vSP[i+1], \dots, vSP[i+TRAIN\_LENGTH])$ 을 구하여 최종적으로

MRSP[i]에 적용한다.

이동권한의 한계점과 MRSP를 기반으로 이동권한 한계점에서의 속도를 계산한다. 이동권한 한계점에서의 속도를 계산하기 위해서 타겟포인트(target point)를 정의한다. 타겟포인트는 열차의 속도가 반드시 0이 되어야 하는 지점으로 다음과 같은 항목 중 최소값으로 정의된다. 최종적으로 이동권한은 타겟포인트를 기준으로 비상제동곡선으로 계산된 거리와 속도값인 (dMA, vMA)로 출력된다.

- i) 선행열차의 비상제동거리
- ii) MRSP[cMAX\_MA+1]에 대한 비상제동거리
- iii) 열차의 비상제동거리
- iv) 이동경로와 방향이 다른 선로전환기 시점까지 거리

위와 같이 위치결정기능, 자원 및 이동권한 관리 기능을 SCADE 모델로 구현하였다. 구현된 모델을 소스코드로 생성하기 전에 각 기능별 테스트를 진행한다.

### 3.4 주행안전기능 모델 테스트

열차자율주행제어시스템 ATP 차상장치의 주행안전기능은 SCADE의 모델로 구현하였다. 구현된 모델을 테스트하기 위해서 테스트케이스를 생성하고, 테스트케이스에 대한 결과에 따라 기능을 보완한다. 이렇게 테스트 완료된 모델은 SCADE-keg를 이용하여 C언어 기반의 소스코드로 생성된다. 생성한 소스코드를 ATP 차상장치 프로토타입에 탑재하기 전, 유사한 사양의 테스트환경을 구축하여 소스코드에 대한 테스트를 수행한다. 이 과정을 통해 ATP 차상장치의 기존 소스코드와 구현된 주행안전기능 모델의 소스코드를 통합하기 전 불필요한 디버깅과정을 생략할 수 있다.

test_operator	test_id	test_name	test_type	test_target	test_scope	test_data	test_result	test_status	test_date	test_time	test_location	test_executor	test_executor_id
TestOperator	1000000001	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000002	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000003	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000004	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000005	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000006	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000007	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000008	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000009	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한
TestOperator	1000000010	이동권한	기능	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한	이동권한

그림 9. 이동권한 요구사항 항목에 대한 테스트케이스 예시  
Fig. 9. Test-case example according to the system requirement specification of MA

테스트환경에서 생성된 소스코드에 대한 테스트가 완료되면, 향후 연구로 실제 ATP 차상장치 프로토타입에 생성된 소스코드 통합과정 및 검증을 수행한다.

열차자율주행제어시스템 차상 ATP 장치의 주행안전기능을 검증하기 위해서 테스트케이스를 생성한다. 생성된 테스트케이스는 각 기능별로 입력값과 예상 출력값을 포함한다. 입력값에 대한 출력값이 예상 출력값과 같을 경우 해당 테스트케이스는 통과한 것으로 표시한다. 그림 9는 이동권한 관리 기능에 대한 테스트케이스 예시이다. 테스트케이스는 각 기능에 대해서 요구사항의 각 항목이 SCADE 모델로 구현되었는지를 확인할 수 있게 작성한다.

테스트케이스에 따라 테스트 수행 시 가장 하위 레벨의 기능에서부터 최상위 레벨의 기능(ATP 차상장치 주행안전기능) 순으로 테스트한다. 테스트 수행 결과에 따라 기능에 대한 수정 및 보완을 구현한다. SCADE 모델에 대한 기능별 테스트가 완료되면 최상위 기능인 주행안전기능에 대한 시뮬레이션을 통해 SCADE 구현 모델에 대한 검증을 수행한다. SCADE 모델 테스트과정은 그림 10과 같다. 총 28개의 세부 기능 모델에 대해서 테스트를 수행하여 구현된 모델의 기능을 검증하였다.

모델에 대한 테스트가 완료되면 그림 11과 같이 생성된 소스코드를 타겟머신에 탑재하여 호환성을 테스트한다. 생성된 소스코드는 헤더파일과 각 모델 별 소스파일로 구성된다.

소스코드 생성에서 테스트까지 전체 과정 및 결과는 다음과 같다.

- i) SCADE를 이용하여 구현한 열차자율주행제어 시스템 ATP 차상장치 주행안전기능 모델에 대한 테스트케이스 생성
- ii) 모델 테스트 수행: 테스트케이스의 예상 결과값과 비교하여 모델 보완

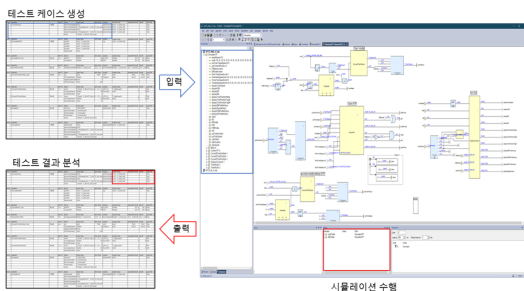


그림 10. 모델 테스트 과정  
Fig. 10. Model test procedure

```

1  *****
2  *** Generated KCG H/Ads files ***
3  *****
4  SimulateATP.h
5  DisplayConnector.h
6  ConvertF64toF32.h
7  Convert2Float32.h
8  ConvertF32toInt32.h
9  Convert2Int.h
10 ProfileHistoryMap.h
...
50 *****
51 *** Generated KCG C/Adb files ***
52 *****
53 SimulateATP.c
54 DisplayConnector.c
55 ConvertF64toF32.c
56 Convert2Float32.c
57 ConvertF32toInt32.c
58 Convert2Int.c
59 ProfileHistoryMap.c
60 Convert2TrainPosition.c
...
98 *****
99 *** Generated SC2C Files ***
100 *****
101 user_macros.h
102 kcg_imported_types.h
103
104 *****
105 *** Include Directories ***
106 *****
107 .
108 $(SCADE)
109 $(SCADE)\include
110 $(SCADE)\include\C
111 $(SCADE)\include\Ada
112 $(SCADE)\lib\Ada
113 ..\..\ATCS_lib_v1\lib\libmathext
    
```

그림 11. 생성된 소스코드 리스트  
Fig. 11. List for the generated source codes

- iii) 테스트 완료된 모델에 대해서 SCADE-kcg를 이용하여 소스코드로 생성
- iv) RTOS(Real-time Operation System) 기반 테스트환경 구축
- v) 생성된 소스코드를 테스트환경에서 빌드하여 실행파일 생성
- vi) 실행파일 테스트

본 논문에서는 소스코드 검증을 위해서 그림 12와 같이 열차자율주행제어시스템 ATP 차상장치 프로토타입과 유사한 RTOS 기반 테스트환경을 구축하였다. ATP 차상장치 프로토타입과 테스트환경에 대한 환경 비교는 표 1과 같다.

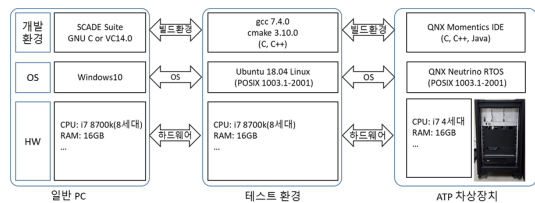


그림 12. 소스코드 테스트환경  
Fig. 12. Test environment for the source codes

표 1. 테스트환경  
Table 1. Test environment

	ATP Prototype spec.	test machine spec.
PC 사양	i7 4th CPU 32GB DDR3 DRAM	i7 7th CPU 32GB DDR3 DRAM
OS	QNX Nutrino RTOS	Ubuntu 18.04 linux
kernel	QNX Neutrino 7.0	linux-5.0.21 (RT patch 5.0.21)
build environment	QNX Momentics IDE (gcc)	cmake 3.10.0 gcc 7.4.0
language	C, C++	C, C++

열차자율주행제어시스템 ATP 차상장치 프로토타입에 대한 테스트환경을 구축하기 위해서 일반 리눅스 커널에 real-time patch를 적용하였다. 그림 13과 같이 커널 컴파일 시 fully preemption kernel로 설정하여 real-time을 지원하는 환경을 구축하였다. 향후, 프로세스 단위의 priority 제어를 통해서 ATP 차상장치 프로토타입과 같은 테스트환경을 구축할 예정이다.

최종적으로 구현한 열차자율주행제어시스템 ATP 차상장치의 주행안전기능에 대한 동작을 테스트환경에서 확인하였다. 실제 열차자율주행제어시스템 ATP 차상장치 프로토타입의 RTOS와 같은 환경에서 구현된 기능의 동작을 확인하기 위해서 주기를 100ms으로 고정하고, 그림 14와 같이 기능의 동작 여부를 확인하였다. 테스트결과 소스코드의 정상적인 실행과 정상 결과값을 확인하였다. 향후, 열차자율주행제어시스템 ATP 차상장치 프로토타입 기반으로 구현된 소스코드를 탑재하여 테스트를 수행할 계획이다.

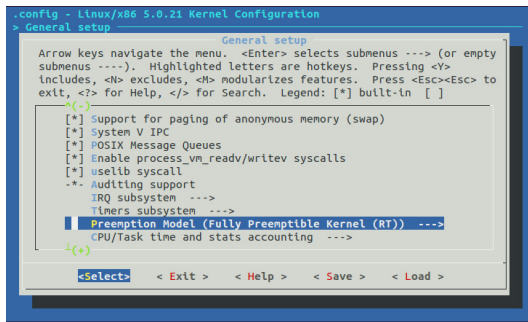


그림 13. 테스트환경 커널 설정화면  
Fig. 13. Screenshot of kernel configuration of test environment

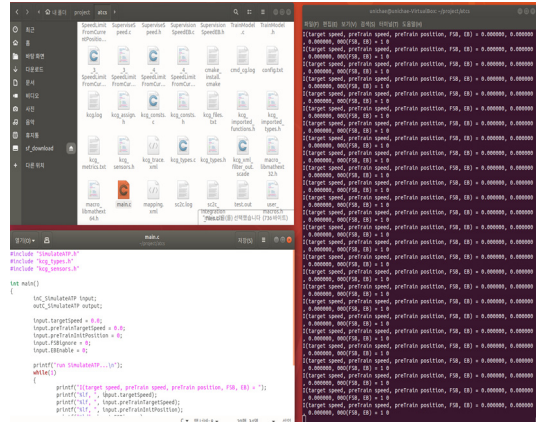


그림 14. 테스트 결과 화면  
Fig. 14. Screenshot of test result of the source codes

#### IV. 결론

본 논문에서는 열차자율주행제어시스템 ATP 차상장치의 주행안전기능 중 열차 위치결정기능과 자원 및 이동권한 관리 기능을 설계 및 구현하였다. 높은 안전성과 신뢰성이 요구되는 소프트웨어 설계 및 구현에 적합한 SCADE 개발도구를 활용하였다. 구현된 모델에 대한 검증은 각 기능별 테스트케이스를 생성하여 단위 테스트로 수행하고, 결과를 반영하여 모델을 보완하였다. SCADE-kcg를 통해 구현된 모델에서 소스코드를 생성하였고, 테스트환경을 구축하여 생성된 소스코드를 테스트하였다. 향후, 열차자율주행제어시스템 ATP 차상장치 프로토타입에 생성된 소스코드를 탑재하여 검증한다.

#### References

- [1] IEEE Std. 1474.1, "IEEE Standard for Communication-Based Train Control(CBTC) Performance and Functional Requirements," 2004.
- [2] IEEE Std. 1474.3, "IEEE Recommended Practice for Communication-Based Train Control(CBTC) System Design and Functional Allocations," 2008.
- [3] S. Oh, "Next Generation Communication-based Train Control System," *J. Railway*, vol. 21, no. 3, pp. 79-86, 2018.
- [4] F. X. Dormoy, "SCADE 6 - A Model Based Solution for Sfty Critical Software



Development,” in *Proc. Eur. Congress ETCS Embedded Real Time Softw.*, 2008.

- [5] E. M. Clarke and J. M. Wing, “Formal Methods: State of the art and Future Directions,” *ACM Comput. Surv.*, vol. 28, no. 4, pp. 626- 643, 1996.
- [6] Esterel Technologies, “*Efficient Development of Safe Railway Applications Software with EN50128 Objectives Using SCADE suite*,” 5th Ed., 2009.
- [7] S. Oh and Y. Cho, “Track Switching Algorithm for the T2T-based Autonomous Train Control System,” *J. KICS*, vol. 42, no. 11, pp. 2160-2169, 2017.
- [8] S. Oh and Y. Cho, “Shared Resource Allocation Scheme and Interval Control Algorithm for the T2T-based Autonomous Train Control System,” *J. KICS*, vol. 42, no. 9, pp. 1781-1791, 2017.
- [9] S. Oh, K. Kim, and H. Choi, “Train Interval Control and Train-centric Distributed Interlocking Algorithm for Autonomous Train Driving Control System,” *J. Korea Academia-Ind. Soc.*, vol. 17, no. 11, pp. 1-9, 2016.

**채 성 윤** (Sung-Yoon Chae)



2011년 2월 : 아주대학교 컴퓨터공학과 박사과정 수료  
2018년 12월~현재 : 한국철도기술연구원 주임기술원  
<관심분야> 컴퓨터공학, 소프트웨어공학  
[ORCID:0000-0001-9576-0114]

**이 병 훈** (Byung-Hun Lee)

한국철도기술연구원 선임연구원  
[ORCID:0000-0001-9886-8198]

**최 현 영** (Hyeon-Yeong Choi)

한국철도기술연구원 선임연구원  
[ORCID:0000-0001-8028-2957]

**방 준 호** (June-Ho Bang)

한국철도기술연구원 선임연구원  
[ORCID:0000-0001-5676-4358]