

논문 2009-04-23

AUTOSAR XML을 이용한 테스트 자동화 시스템 개발

(Automated Testing System Using AUTOSAR XML)

금대현, 이성훈, 박광민, 조정훈*

(Daehyun Kum, Seonghun Lee, Gwangmin Park, Jeonghun Cho)

Abstract : Recently a standard software platform for automotive, AUTOSAR, has been developed to manage growing software complexity and improve software reuseability. However reuse of testing system and test data are difficult because they are dependant on implementation language and testing phases. In this paper, we suggest a automated testing approach for AUTOSAR software component using a standardized testing language, TTCN-3. AUTOSAR defines the AUTOSAR XML Schema for the data exchange format so that it is possible to automatically convert AUTOSAR model into TTCN-3 testing model. Therefore our approach is to present generation techniques for the TTCN-3 testing system from a AUTOSAR XML description. With the proposed testing techniques we can reduce time and effort to build the testing system and reuse testing environment.

Keywords : AUTOSAR, TTCN-3, Software component, Software testing, Automated testing

1. 서론

최근 차량의 지능화로 말미암아 차량 임베디드 소프트웨어는 점차 복잡해지고 네트워크화되고 있으며, 그 중요성 또한 증가하고 있다. 세계 자동차 업계는 소프트웨어의 신뢰성 및 재사용성 향상을 위하여 AUTOSAR[1] 표준 소프트웨어 플랫폼을 제정하였지만, 소프트웨어의 표준화만으로는 테스트에 소요되는 많은 시간과 노력을 줄이기에는 한계가 있다. 또한, 지난 몇 년 동안 자동차 제조사와 부품업체의 많은 노력으로 품질 향상에 상당한 진전이 있었지만, 급속히 늘어나는 소프트웨어의 복잡성을 해결하기에는 역부족이었다.

소프트웨어의 요구사항이 복잡할수록 테스트에 소요되는 시간이 늘어나고, 오류 발생 확률은 높아진다. 그러나 아직도 차량 임베디드 소프트웨어 테

스팅의 상당 부분이 수작업으로 이루어지며, 표준화되지 않은 자체 테스트 환경과 도구를 사용하고 있다. 이러한 문제점들을 해결하기 위해서 테스트 표준을 도입하여 테스트 데이터와 테스트 환경의 재사용성을 향상시키고, 테스트를 자동화하여 신뢰성 향상 및 개발 기간을 단축할 필요성이 있다. 최근 차량 임베디드 소프트웨어의 테스트를 위한 다양한 연구가 진행되고 있지만, 표준화 및 자동화 측면에서 부족한 면이 있다[2-3].

AUTOSAR는 설계 도구의 적합성 테스트를 위하여 테스트 표준 언어인 TTCN-3(Testing and Test Control Notation Version 3)을 채택하고 있으며, AUTOSAR 설계 도구의 적합성을 위한 TTCN-3 테스트 시스템을 규정하고 있다.[4-6] 그러나 적합성 테스트와 AUTOSAR 응용 시스템의 테스트 환경은 서로 다르며 응용 시스템에 적합한 표준화된 테스트 자동화 방법에 대한 연구가 필요하다. 그리고 현재까지 진행되고 있는 연구는 단순히 AUTOSAR 소프트웨어 컴포넌트를 TTCN-3로 맵핑하는 기술에 제한되고 있으며[7], 테스트 자동화를 위한 TTCN-3 테스트 및 타겟 시스템의 구조, 테스트 자동화 방법에 대한 연구가 필요하다.

본 논문에서는 TTCN-3 테스트 표준을 적용한 AUTOSAR 응용 소프트웨어 컴포넌트의 테스트 자

* 교신저자(Corresponding Author)

논문접수 : 2009. 10. 19.,

수정·채택확정 : 2009. 11. 18.

금대현, 이성훈, 박광민 : 대구경북과학기술원

조정훈 : 경북대학교 전자전기컴퓨터학부

※ 본 연구는 교육과학기술부에서 지원하는 기관 고유사업비로 수행하였음.

동화 방법 및 테스트 시스템의 구조를 제안하고자 한다. 표준화된 테스트 환경을 적용함으로써 정보 교환 및 의사소통이 명확해 지고, 또한 테스트 환경 및 데이터의 재사용성을 향상시킬 수 있다. 그리고 시스템의 개발 초기 단계에 하드웨어 및 네트워크의 도움 없이 설계 모델로부터 테스트 모델을 자동 생성함으로써 개발 기간 단축 및 신뢰성을 향상시킬 수 있다.

II. AUTOSAR 및 TTCN-3 표준 소개

1. AUTOSAR 표준 [8-9]

차량 소프트웨어에 대한 공개 표준을 제정하는 것을 목표로 2003년 AUTOSAR 컨소시엄이 만들어졌으며, 현재까지 표준에 대한 공개가 꾸준히 이어지고 있다. 현재 주요 자동차 업체들은 AUTOSAR 적용 차량의 개발을 준비하고 있다.

AUTOSAR 소프트웨어 아키텍처는 재사용성 향상을 위하여 소프트웨어 컴포넌트와 모듈의 기능 그리고 서로 간의 인터페이스를 표준화하고 있다. 그리고 개발 방법에 대한 표준도 정의하고 있으며, 표준화된 AUTOSAR XML 스키마를 정의하여 설계 명세의 호환성 및 재사용성을 보장하고 있다. 본 논문에서는 AUTOSAR XML 설계 명세를 이용하여 테스트 시스템을 자동으로 생성하는 기술을 제안하고 있다. 본 학회의 논문의 여백 규정은 다음 표안의 내용으로 정의된다.

2. TTCN-3 표준 [10-11]

TTCN-3은 통신, 자동차 등 다양한 시스템의 블랙박스 테스트에 적용되고 있는 테스트 명세를 위한 표준이다. TTCN-3은 2000년에 최초로 발표되었으며, 현재 다양한 산업분야에서 응용되고 있으며, 특히 최근에는 MOST 및 AUTOSAR에서 적합성 테스트 표준으로 채택되었다. 테스트 대상 시스템의 구현에 독립적으로 테스트 케이스 및 테스트 시스템을 설계할 수 있도록 고안되었으며, 테스트 시스템 설계를 위한 TRI(TTCN-3 Runtime Interface), TCI(TTCN-3 Control Interface) 인터페이스를 제공한다. 본 논문에서는 TTCN-3을 적용하여 AUTOSAR 테스트 시스템의 재사용성 및 호환성을 향상시키고자 한다.

III. 테스트 시스템

1. 테스트 자동화 방법

본 테스트 시스템은 개발 초기 단계에 하드웨어 및 네트워크 통신의 도움 없이 AUTOSAR 응용 소프트웨어 컴포넌트를 테스트 할 수 있게 하고, 테스트 환경의 표준화 및 자동화하는 데 목적이 있다.

TTCN-3 테스트 표준을 적용함으로써 테스트 케이스 및 테스트 실행 환경의 공유가 쉬우며 타겟의 구현에 독립적으로 재사용 가능하다. 그리고 AUTOSAR의 설계 모델로부터 TTCN-3 테스터와 타겟 시스템을 자동 생성함으로써 설계 변경 또는 오류 수정으로 말미암아 반복적으로 수행되는 테스트에 소요되는 노력과 시간을 최소화할 수 있다.

AUTOSAR는 표준화된 XML 스키마를 이용하기 때문에 소프트웨어 설계 정보의 활용이 쉽다. 이러한 XML 정보를 사용하여 테스트 시스템을 빠르고 정확하게 생성할 수 있다. 그림 1과같이 소프트웨어 컴포넌트, ECU 그리고 시스템 설계 명세로부터 테스트 시스템의 환경설정을 위한 명세를 추출하고, SUT(System Under Test)를 생성한다. 그리고 테스트 케이스를 위한 XML 스키마를 정의하고 설계함으로써, 테스트 케이스를 TTCN-3 테스트 데이터로 변환하여 즉시 실행 가능하도록 자동화하였다.

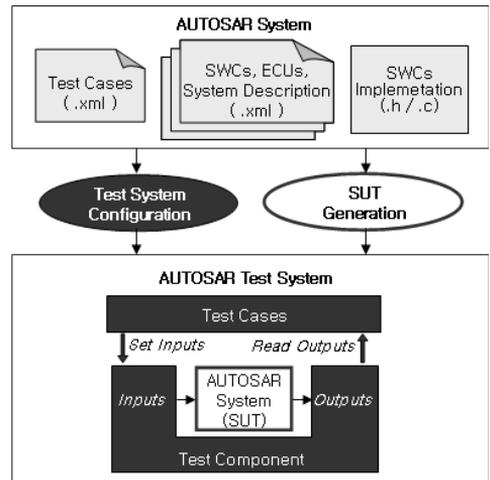


그림 1. AUTOSAR 시스템의 테스트 자동화 방법
Fig. 1. Automated testing technique for AUTOSAR systems

2. 테스트 시스템 구조

AUTOSAR 응용 소프트웨어 컴포넌트를 위한 TTCN-3 테스트 자동화 시스템은 테스터와 타겟

시스템으로 나누어진다. 두 시스템은 UDP/IP, TCP/IP, CAN 등과 같은 네트워크 통신을 이용하여 서로 정보를 교환하며, 통신을 사용함으로써 테스트와 타겟 시스템은 서로 독립성을 유지할 수 있다. 그림 2는 테스트 시스템의 구조를 보여준다.

AUTOSAR 시스템을 위한 TTCN-3 테스트의 구조는 간단히 표현하면 그림 2의 좌측과 같다. 테스트 컴포넌트는 테스트 입력을 송신하고 SUT에서 실행된 결과 값을 수신하여 결과를 판단한다. 인코더는 TTCN-3 테스트 데이터 타입을 통신 프로토콜에 적합한 형식 변환하기 위한 모듈이며, 디코더는 반대로 수신한 메시지를 TTCN-3 데이터 타입으로 변환한다.

AUTOSAR 타겟 시스템은 SUT, SUT 인터페이스, 타겟 코덱, 통신 인터페이스, 타겟 스케줄러 그리고 타겟 환경 모듈로 이루어지며 그림 2의 우측과 같다. SUT 인터페이스 모듈은 AUTOSAR 시스템의 입출력과 타겟 시스템의 입출력 신호를 서로 연결해주는 역할을 한다. 타겟 코덱 모듈은 테스트로부터 수신된 통신 메시지를 테스트 신호로 변환하고, 반대로 테스트로 송신할 테스트 출력 신호를 통신 메시지로 변환하는 역할을 담당한다. 통신 인터페이스는 테스트와 통신 메시지를 송수신하기 위한 모듈이다. 타겟 스케줄러는 타겟 시스템의 스케줄링을 담당한다.

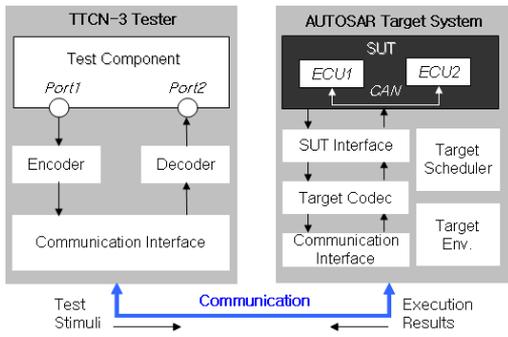


그림 2. AUTOSAR 시스템의 테스트 시스템
Fig. 2. Testing system for AUTOSAR systems

IV. 테스트 자동화 시스템

1. TTCN-3 테스터

TTCN-3 테스트 컴포넌트는 크게 테스트 컴포넌트 파일, AUTOSAR XML 명세로부터 추출되어 생성되는 환경 설정(Configuration) 파일 그리고 테

스트 입력, 예상 출력, 테스트 시간 등의 테스트 데이터 파일로 구성된다.

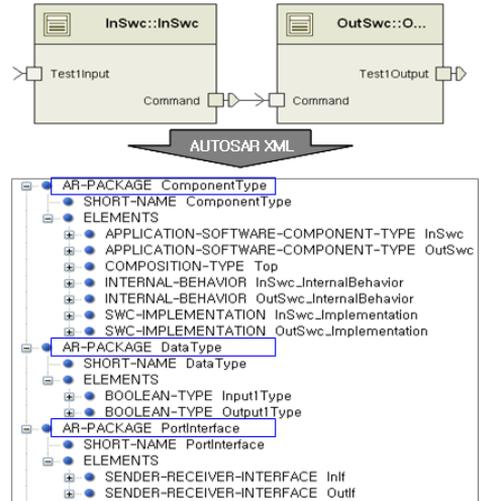


그림 3. AUTOSAR XML
Fig. 3. AUTOSAR XML

```

/** Generated data types */
type integer Input1Type (0..3);
type integer Output1Type (0..2);

/** Test input signal ID */
const integer Input1Signal := 1;

/** Test output signal ID */
const integer Output1Signal := 2

/** Test signal size */
const integer nTI := 1; // TI : Test Input
const integer nER := 1; // ER : Expected Result
type TestSignal TI[nTI];
type TestSignal ER[nER];

/** Test Template */
template TestSignal setInput1Signal (integer Value) := {
    signalID := Input1Signal,
    signalLength := 1,
    signalValue := Value
}
template TestSignal getOutput1Signal (integer Value) := {
    signalID := Output1Signal,
    signalLength := 1,
    signalValue := Value
}
    
```

그림 4. TTCN-3 테스터 생성
Fig. 4. Generation of the TTCN-3 tester

TTCN-3 테스터 환경 설정 파일은 AUTOSAR 소프트웨어 컴포넌트의 설계 모델에 따라서 바뀌는 파일이며 AUTOSAR XML[12] 설계 명세로부터 자동 생성된다.

```

/**** Test Data ****/
group TestData_01 {
  const TI TL_01 := { setInput1Signal (0) };
  const ER ER_01 := { getOutput1Signal (0) };
}
group TestData_02 {
  const TI TL_02 := { setInput1Signal (1) };
  const ER ER_02 := { getOutput1Signal (1) };
}

/**** Test execution ****/
control {
  execute ( TC (TL_01,ER_01,nTL_01,nER_01), 30.0);
  execute ( TC (TL_02,ER_02,nTL_02,nER_02), 30.0);
}
    
```

그림 5. TTCN-3 테스트 데이터 및 테스트 실행
 Fig. 5. TTCN-3 test data and test execution

```

/**** AUTOSAR data types ****/
type integer uint8 (0..255);
type integer uint16 (0..65535);

/**** Test signal format ****/
type record TestSignal {
  integer signalID,
  integer signalLength,
  integer signalValue
}

/**** Test port ****/
type port pt message {
  inout TestSignal :
}

/**** Test component ****/
type component System {
  port pt ptSystem;
}
type component TestComponent {
  port pt ptTC;
}

/**** Test function ****/
function funcSendTestInputs (TI TIs, integer nTIs)
runs on TestComponent {
  var integer i;
  for(i:=0 ; i<nTIs ; i:=i+1){
    pt.send(TIs[i]);
  }
}

function funcEvaluateResult(ER ERs, integernERs)
runs on TestComponent {
  var integer i;
  timerTC.start;
  for(i:=0 ; i<nERs-n ; i:=i+1) {
    alt {
      [ ] alt_evaluateResult(ERs, nERs) { }
      [ ] alt_timeGuard (timerTC) { }
    }
  }
}

/**** Test case ****/
testcase TC (TI TIs, ER ERs, integernTIs, integernERs)
runs on TestComponent system System{
  map(mtc:ptTC, system:ptSystem);
  funcSendTestInputs(TIs, nTIs);
  funcEvaluateResult(ERs, nERs);
  all component.done;
  unmap(mtc:ptTC, system:ptSystem);
}
    
```

그림 6. TTCN-3 테스트 컴포넌트
 Fig. 6. TTCN-3 test component

2. AUTOSAR 타겟 시스템

AUTOSAR 응용 소프트웨어 컴포넌트는 일반적으로 하드웨어의 개발보다 선행되어서 개발된다. 그러므로 개발 초기 단계에 응용 소프트웨어 컴포넌트의 빠른 검증을 위해서는 일반 PC환경에서 테스트 할 수 시뮬레이션 환경이 필요하다.

AUTOSAR 타겟 시스템은 응용 소프트웨어 컴포넌트의 테스트 자동화를 위한 PC 시뮬레이션 환경을 제공하고 TTCN-3 테스터와 테스트 데이터를 송수신 할 수 있는 인터페이스를 제공한다.

소프트웨어 컴포넌트를 PC에서 테스트하기 위해서는 AUTOSAR RTE, OS, COM, IO 모듈 및 CAN 네트워크의 시뮬레이션이 필요하다. PC 시뮬레이션 환경은 AUTOSAR XML 설계 명세로부터 생성한다 [12-14].

TTCN-3 테스터와 테스트 데이터 교환을 위해서 SUT 인터페이스, 타겟 코덱 그리고 통신 인터페이스모듈이 필요하다. SUT 인터페이스는 TTCN-3 테스터로부터 수신된 테스트 입력을 버퍼에 저장하고, 저장된 값은 AUTOSAR 타겟 시스템에서 읽게 된다. 그리고 수행된 결과 값은 테스트 결과 버퍼에 저장하여 TTCN-3 테스터로 송신하게 된다. 그림 7은 타겟 시스템의 각 모듈에 대한 개념을 설명하고 있다.

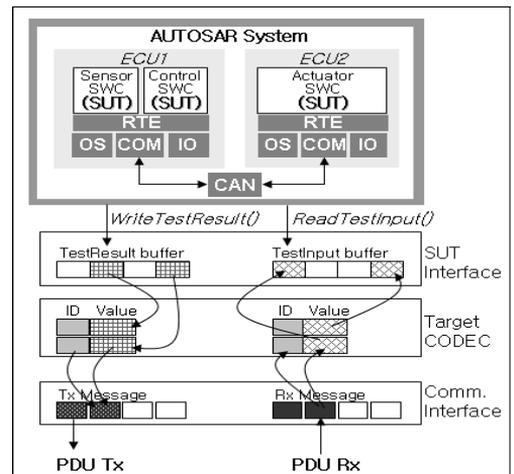


그림 7. AUTOSAR 타겟 시스템
 Fig. 7. AUTOSAR target system

그림 2의 타겟 스케줄러는 각 ECU, CAN 네트워크, 테스트 입력 수신, 테스트 출력 송신 등에 관련된 스케줄링을 담당한다. 각 ECU 내부 태스크의 스케줄링은 해당 ECU의 OS 모듈에서 담당하지만,

ECU 및 네트워크 사이의 스케줄링은 타겟 스케줄러에서 담당한다. 스케줄링은 순차적으로 ECU들과 네트워크 모듈을 실행시키고, 테스트 출력 값에 변화가 발생하면 송신 요청을 한다. 테스트 입력 수신 태스크는 가장 높은 우선순위를 가지며 수신 즉시 테스트 입력 버퍼에 저장하게 된다. 테스트 입력 및 출력 버퍼에는 각각 플래그를 두어서 버퍼에 값이 쓰이면 플래그를 설정하고 값을 읽은 후에는 플래그를 해제한다.

```

/** Test Signal Size */
#define TESTINPUTSIZE 1
#define TESTRESULTSIZE 1

/** Test Signal ID */
#define SID_Input1Signal 1
#define SID_Output1Signal 2

/** Test Signal Mapping */
void MapSIGNALID( ) {
    BUF_TL_SIGNAL[0] = SID_Input1Signal ;
    BUF_TO_SIGNAL[0] = SID_Output1Signal;
}

/** TargetScheduler Configuration */
#define nENTITY 3
#define ENTITY1 Schedule_ECU1
#define ENTITY2 Schedule_ECU2
#define ENTITY3 Schedule_CAN
    
```

그림 8. AUTOSAR 타겟 시스템의 생성

Fig. 8. Generation of the AUTOSAR target system

타겟 시스템의 환경 설정은 SUT 인터페이스와 타겟 스케줄러에서 이루어진다. SUT 인터페이스의 환경설정은 테스트 입출력 신호의 ID 설정하는 부분과 AUTOSAR 타겟 입출력 신호를 SUT 인터페이스의 버퍼에 맵핑하는 부분으로 나누어진다. 타겟 스케줄러의 환경 설정은 ECU와 네트워크 실행에 관련된 부분이다. 그림 8은 타겟 시스템의 환경 설정에 관련된 예를 보여준다. 환경 설정에 관련된 정보도 AUTOSAR XML 설계 명세로부터 생성한다.

V. Case Study - 차량 실내등 시스템

1. 차량 실내등 시스템

본 논문에서 제시한 TTCN-3을 이용한 AUTOSAR 응용 소프트웨어 컴포넌트 테스트 자동화 시스템을 차량 실내등 시스템에 적용하였다. 실내등 시스템의 소프트웨어 컴포넌트 구조는 그림 9와 같이 6개의 소프트웨어 컴포넌트로 구성되고, 2개의 ECU와 CAN 통신으로 구성된다. 실내등 시스템

의 입력 신호는 7개의 스위치 입력과 8개의 CAN 신호이며, 출력은 5개의 램프로 구성되어 있다.

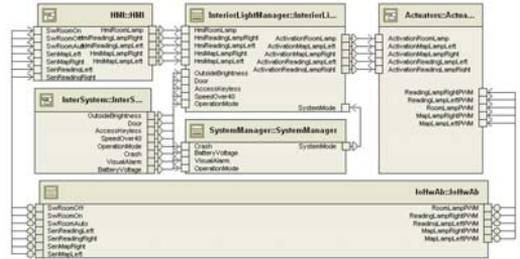


그림 9. 실내등 시스템

Fig. 9. Interior lighting system

테스팅 시스템은 Java 기반의 TTCN-3 설계 도구를 사용하였으며, 타겟 시스템은 Microsoft Visual Studio의 C언어 개발 도구를 사용하였다. 두 시스템 사이의 통신은 UDP/IP 방식을 사용하였으며, 하나의 PC환경에서 테스트 할 수 있도록 설계하였다.

2. 차량 실내등 테스트 시스템의 생성

AUTOSAR 소프트웨어 컴포넌트를 위한 테스트 시스템의 환경 설정 파일은 AUTOSAR XML 명세로부터 생성하였다. TTCN-3 테스터의 환경 설정은 그림 4와 같이 15개의 테스트 입력 및 5개의 결과 신호에 아이디를 부여하고 각 신호에 대해서 TTCN-3 템플릿을 생성하였다. 타겟 시스템의 환경 설정은 그림 8과 같이 테스트 입력 및 출력 아이디를 TTCN-3 테스터와 같이 하고, 버퍼와 테스트 신호를 맵핑시킨다. 그리고 타겟 시스템의 스케줄링을 위하여 ECU 2개와 네트워크 1개에 대한 정의를 하였다.

소프트웨어 컴포넌트의 PC 실행 환경을 위하여 AUTOSAR XML 설계 명세로부터 PC에서 실행 가능한 RTE, COM, OS, IO 모듈을 각 ECU에 대해서 생성하고, 네트워크 통신을 위한 CAN 시뮬레이션 모듈을 생성하였다.

3. 테스트 케이스의 실행 및 결과

차량 실내등 시스템에 대한 기능 테스트 케이스를 실행하였다. 그림 10은 룸 램프에 대한 테스트 시나리오의 일부를 보여주고 있으며, 그림 11은 테스트 시나리오로부터 생성된 TTCN-3 테스트 데이터와 실행을 보여주고 있다. 그리고 그림 12는 테스트

트 결과를 보여주고 있다.

RoomLamp									
Test_ID	Req_ID	Name	Pre Condition	Time	Test Input Name	value	Time	Expected Result Name	Value
TID_001	UFR_001	RoomLampOn	Initialization	0	SwRoomOn	1	100	RoomLampPWM	100
TID_002	UFR_002	RoomLampOff	RoomLampOn	0	SwRoomOff	1	100	RoomLampPWM	0
TID_003	UFR_003	RoomLampAuto	RoomLampOff	0	SwRoomAuto	1	100	RoomLampPWM	100
				0	OutsideBright	1			

그림 10. 실내등 시스템의 테스트 시나리오

Fig. 10. Test scenario of the interior light system

```

/** Test data */
group TID_001 {
  const integer nTL_001 := 1;
  const integer nER_001 := 1;
  const TI TL_001 := {
    setSwRoomOn(1), -, -, -, -, -, -, -, -, -, -};
  const ER ER_001 := {
    getRoomLampPWM(100), -, -, -, -};
}
group TID_002 {
  const integer nTL_002 := 1;
  const integer nER_002 := 1;
  const TI TL_002 := {
    setSwRoomOff(1), -, -, -, -, -, -, -, -, -, -};
  const ER ER_002 := {
    getRoomLampPWM(0), -, -, -, -};
}
group TID_003 {
  const integer nTL_003 := 1;
  const integer nER_003 := 1;
  const TI TL_003 := {
    setSwRoomAuto(1),
    setOutsideBrightness(1), -, -, -, -, -, -, -, -, -, -};
  const ER ER_003 := {
    getRoomLampPWM(100), -, -, -, -};
}

/** Test execution */
control {
  execute TC (TL_001, ER_001, nTL_001, nER_001), 30.0);
  execute TC (TL_002, ER_002, nTL_002, nER_002), 30.0);
  execute TC (TL_003, ER_003, nTL_003, nER_003), 30.0);
}
    
```

그림 11. 실내등 시스템의 테스트 케이스 및 실행
Fig. 11. Test cases and execution of the interior lighting system

테스트 시스템의 환경 설정 및 소프트웨어 컴포넌트 실행 환경은 AUTOSAR XML 설계 명세로부터 생성 가능하므로 테스트 시스템 설계에 소요되는 시간을 최소화할 수 있었으며, 정형화된 테스트 시나리오로부터 TTCN-3 테스트 데이터로 즉시 변환하고 실행하여 그 결과를 확인할 수 있었다.

일반적으로 설계 모델을 완성 후 테스트를 위한 환경을 설계하고 구현하게 되며, 테스트 모델의 검증 과정도 필요하다. 더구나 설계 모델의 변경이 발생할 경우 테스트 모델을 다시 만들고 검증하는 경우도 발생하게 된다. 기존의 방법을 사용해서 차량 실내등 시스템의 테스트 환경을 설계하고 검증하는데 며칠이 소요되었지만, 본 논문에서 제시한 자동화 방법을 적용하게 되면 불과 몇 분 만에 테스트 모델을 생성해서 테스트 결과를 확인할 수 있었다.

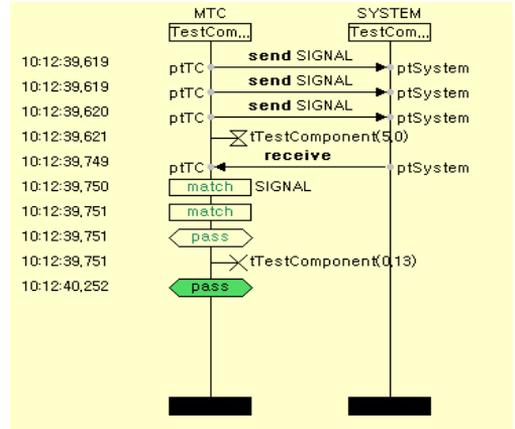


그림 12. 실내등 시스템의 테스트 결과

Fig. 12. Test result of the interior light system

VI. 결론

최근 소프트웨어의 복잡성 때문에 테스트에 소요되는 시간과 비용이 증가하고 있으며, 테스트 방법의 표준화와 자동화로 차량 임베디드 시스템의 경쟁력을 높일 필요성이 있다.

본 연구에서는 AUTOSAR 응용 소프트웨어 컴포넌트를 위한 TTCN-3 기반 표준 테스트 시스템 및 테스트 자동화 방법을 제시하였으며 차량 실내등 시스템의 테스트에 적용해 보았다. 본 테스트 시스템을 차량 실내등 시스템에 적용한 결과, 생성된 테스트 자동화 환경을 이용하여 즉시 테스트를 수행하고 그 결과를 확인할 수 있었다.

테스팅 표준화로 테스트 케이스 및 테스트 환경의 재사용이 용이하며, AUTOSAR XML 설계 명세로부터 테스트 환경을 자동 생성할 수 있는 장점이 있다. 본 연구에서 제안한 방법을 사용함으로써 테스트 비용 및 시간을 절감하고 신뢰성 향상에 기여할 수 있을 것이다.

참고문헌

- [1] <http://www.autosar.org>.
- [2] D. Kum, J. Son, J. Son and M. Kim, "Automotive embedded system software development and validation with AUTOSAR and model-based approach", Journal of Control, Automation, and System Engineering, Vol.13, No.12, pp. 1179-1185, 2007.
- [3] J. Youn, J. Ma, M. Sunwoo and W. Lee, "Model based design and validation of control systems using real-time operating system", Transaction of KSAE, Vol.16, No.2, pp. 8-17, 2008.
- [4] AUTOSAR BSW & RTE Conformance Test Specification Part 1: Background, AUTOSAR Specification R3.1.
- [5] A. Gilberg, "AUTOSAR conformance testing using TTCN-3", TTCN-3 User Conference, 2009.
- [6] J. Grossmann, D. Serbanescu and I. Schieferdecker, "Testing embedded real-time systems with TTCN-3", Proceedings of 2009 International Conference on Software Testing Verification and Validation, pp. 81-90, 2009
- [7] J. Großmann, I. Schieferdecker, "Mapping AUTOSAR interfaces to TTCN-3", TTCN-3 User Conference, 2009.
- [8] W. Yoo, J. Park, J. Yoo and S. Hong, "AUTOSAR-based Software architecture for automotive systems", Proceedings of KSAE Symposium, pp. 60-65, 2006.
- [9] D. Kum, S. Lee and W. Jung, "Scheduling analysis in the AUTOSAR software architecture", Proceedings of KSAE 2008 Annual Conference, pp. 2174-2179, 2008.
- [10] J. Grabowski, D. Hogrefe, G. Réthy, I. Schieferdecker, A. Wiles and C. Willcock, "An Introduction to the testing and test control notation (TTCN-3)", Computer Networks Vol.42, pp. 375-403, 2003.
- [11] S. Schulz and T. Vassiliou-Gioles, "Implementation of TTCN-3 test systems using the TRI", Proceedings of the IFIP 14th International Conference on Testing Communicating Systems XIV, pp. 425-442, 2002.
- [12] SW-C and System Modeling Guide, AUTOSAR Specification R3.1.
- [13] Specification of ECU Configuration, AUTOSAR Specification R3.1.
- [14] Technical Overview, AUTOSAR Specification R3.1.

저 자 소 개

금대현

2001년 계명대학교
자동차공학과 학사.
2003년 계명대학교
자동차공학과 석사.
현재, 경북대학교 전자전
기컴퓨터학과 박사과정.

2003~2005 LG전자 연구원.
2005~현재 대구경북과학기술원 연구원.
관심분야: 임베디드소프트웨어, 테스트자동화.
Email: kumdh@dgist.ac.kr

이성훈

1996 경북대학교
전자공학과 학사.
1998 경북대학교
전자공학과 석사.
2007 경북대학교
전자공학과박사.

1999~2002 대우 정밀 기술연구소.
2002~2005 국방과학연구소 .
2005~현재 대구경북과학기술원 선임연구원.
관심분야: 차량 임베디드 시스템.
Email: shunlee@dgist.ac.kr

박광민

2005년 한국항공대
항공기계학과 학사.
2007년 광주과학기술원
기계전자과 석사.
현재, 대구경북과학기술원
연구원.

관심분야: 임베디드 소프트웨어, SW테스트.
Email: ggangmin@dgist.ac.kr

조정훈

1996년 KAIST
전기및전자공학과 학사.
1998년 KAIST
전기및전자공학과 석사.
2003년 KAIST
전자전산학과 박사.
2005년 하이닉스반도체

응용설계팀 선임연구원.
현재, 경북대학교 전자전기컴퓨터학부 조교수.
관심분야: SoC, 임베디드 시스템 최적화, 컴파일러, 운영체제.
Email: jcho@ee.knu.ac.kr