

# 임베디드 시스템 소프트웨어 테스트 방법

이 상 수\*, 신 석 규\*, 김 현 수\*\*

## Software Test for Embedded Systems

Sang-Soo Lee\*, Seok-Kyoo Shin\*, Hyun-Soo Kim\*\*

### Abstract

There are many stages of the development for the embedded systems' hardware and software as well, which results in many test stages of it with respect to corresponding development stages. The software validation of embedded systems is taken into consideration for the testers to make sure that the systems work correctly after the deployment. Among test stages, especially control logic level testing and system validation testing are considered as the most important test, because the robustness of embedded systems' software can be validated by those two level of tests. In this paper, we would like to introduce the idea of how software test system for embedded systems can be established and of what necessary components are needed.

Keyword : Embedded Systems, Software Validation Test

---

\* 한국정보통신기술협회, \*\*국민대

# 1. 서론

일반적으로 임베디드 시스템은 PC기반 시스템과는 하드웨어 및 소프트웨어가 다르므로 테스트 방식 또한 다르다.

PC기반 시스템은 주요 입력 및 출력 장치가 표준화 되어 있으며 표준화 된 입/출력 장치를 사용하는 단일화된 테스트 방식을 사용할 수 있다. 그러나 임베디드 시스템은 사용목적에 따라 다양한 형태의 하드웨어를 사용하며 이에 따른 입/출력 장치가 다양하여 단일화된 테스트 방식을 적용할 수 없다. 그러므로 다양한 시스템에 맞는 다양한 소프트웨어 테스트 방식을 사용해야 한다[1].

임베디드 소프트웨어의 범위는 매우 넓다. 냉장고에서부터 휴대폰, DMB, 우주왕복선을 제어하기 위한 용도에까지 임베디드 소프트웨어가 널리 사용되고 있다. 다양한 분야에서 임베디드 소프트웨어가 사용되므로 하나의 단일화된 형태의 테스트 시스템을 제안하기는 어렵다. 본 논문에서는 한 예를 통해 독자가 설계하고자 하는 임베디드 테스트 시스템에 대한 아이디어를 제공하는데 목적이 있다.

본론에서 다양한 임베디드 시스템 테스트 기법을 개발 단계별로 보여주고 Control Logic 및 시스템 통합을 통한 테스트 방법을 살펴봄에 결론에서는 본 논문에서 제시한 임베디드 시스템 테스트 방법에 대해 정리한다.

## 2. 임베디드 시스템 품질확보의 중요성

현대 사회는 컴퓨터 시스템 기술과 네트워크 기술의 발전으로 유비쿼터스 사회로 빠르게 이전되어 가고 있다. RFID를 SCM(Supply Chain Management)에 도입하여 능동적인 물류관리시스템으로 진화하고 있으며 IPv6 기술 및 BcN을 통해 단일 통신망과 관련 서비스에 대해 심도 있게 논의되고 있다. 유비쿼터스 사회에서는 PC기반의 컴퓨

팅 환경뿐만 아니라 다양한 기기들이 로컬 망에서 글로벌 망으로 연결되는 형태로 발전될 것이다. 이러한 형태의 시스템에서는 로컬 시스템의 품질이 유비쿼터스 시스템의 전체 품질을 결정하는 중요한 요소가 될 것이다. 미래에는 이처럼 로컬 망의 한 네트워크를 구성하는 임베디드 시스템의 중요성이 날로 높아져 갈 것이며 이 시스템의 품질 및 신뢰성이 전체 네트워크에 미치는 영향이 클 것이다.

일반적으로 임베디드 시스템은 한번 개발 후 장착되면 유지보수가 어렵다. 예를 들어 우주 탐사선에 장착된 임베디드 시스템과 소프트웨어는 오류가 발생할 경우 즉시 조치하기가 어렵다. 따라서 임베디드 소프트웨어는 시스템에 장착되기 전에 철저한 테스트를 거쳐 적절한 수준의 품질이 확보되어야 한다. 임베디드 소프트웨어의 품질확보는 곧 그 시스템의 성공여부와 결부되기 때문이다.

성공적인 임베디드 시스템 테스트 설계를 위해서는 시스템의 용도와 사용 환경에 맞는 수준의 품질을 보장하기 위한 노력이 선행되어야 한다.

## 3. 개발단계별 테스트 기법

임베디드 시스템의 개발단계에 따른 테스트 방법은 소프트웨어 공학에서 제시하는 V 모델을 그대로 따른다. 따라서 개발 단계별로 실시하는 테스트

Level	Check Points	Tools
System	<ul style="list-style-type: none"> <li>Integration</li> <li>Reliability, Robustness</li> <li>Exception Handling in the system level</li> </ul>	Chamber, Automated Tools (LabView)
Control Logic	<ul style="list-style-type: none"> <li>Customer Specification &amp; Requirement</li> <li>Functionality</li> </ul>	Stub (Speed 2k, CAN)
Transport	<ul style="list-style-type: none"> <li>Protocols</li> <li>IEEE Standard, De-facto Standard, Specific Protocol</li> <li>Performance</li> <li>Load &amp; Stress, Scalability, Optimization, Benchmark</li> <li>Compatibility Test</li> <li>Inter-operability, Competibility, Reliability</li> <li>Resource Allocation</li> </ul>	Dedicated Tools for each Protocols (CanOE)
O/S	<ul style="list-style-type: none"> <li>Device Driver Reliability Test</li> <li>Functionality, Competibility</li> <li>Resources Allocation</li> </ul>	Tool Chains, Cross compiler, Oscilloscope
Hardware	<ul style="list-style-type: none"> <li>PCB/SOC Level Test bed</li> <li>Timing &amp; Clock debugging</li> <li>Hardware Competibility</li> <li>Reliability &amp; Performance</li> </ul>	Oscilloscope, Digital Analyzer, JTAG, in-circuit emulator

그림 1. 임베디드 시스템 개발단계별 테스트 방법

트는 각 개발단계와 연계되어 진행된다. 그림 1은 임베디드 시스템의 각 개발 단계별 테스트 기법을 보이고 있다. 본 논문에서는 PC기반 응용프로그램의 블랙박스 테스트에 해당하는 Control Logic 및 System 통합 레벨의 테스트에 대해 중점적으로 설명한다.

임베디드 시스템의 개발단계는 하드웨어단계, O/S단계, Transport단계, Control Logic단계 그리고 System통합단계로 구분될 수 있다. 하드웨어와 O/S개발 단계에서는 오실로스코프, JTAG 및 컴과 일러를 이용하여 하드웨어 또는 하드웨어 동작에 관한 테스트를 할 수 있다[2]. 이 단계에서는 정적 및 동적 분석을 활용한 화이트박스 테스트 기법의 소프트웨어 테스트 방식과 로직 게이트레벨의 칩 타이밍 분석의 하드웨어 테스트 방법을 병행하여 사용한다.

IEEE 표준 및 산업표준에 적합한 프로토콜을 시스템에 적용할 경우 프로토콜 분석에 적합한 툴을 사용하여 블랙박스 테스트를 수행할 수 있다.

Control Logic레벨은 블랙박스 테스트를 이용한 테스트를 수행할 수 있는 개발 단계이며 PC 기반의 일반 응용 소프트웨어의 테스트 방식을 그대로 적용할 수 있다. 단, 입/출력이 PC기반과 같이 표준화 되어 있지 않고 PC기반과 다르므로 테스트 시스템 구성이 차이가 있을 수 있다.

시스템 통합 레벨에 대한 테스트는 임베디드 시스템에서 매우 중요한 요소이며 PC기반의 테스트에는 존재하지 않는 부분이다. 임베디드 시스템은 그 특성상 단일 시스템에서 정상적으로 동작하더라도 시스템 통합 후에도 정상적으로 동작한다는 보장을 할 수 없다. 따라서 통합시스템 기반의 테스트가 필수적으로 요구된다.

임베디드 시스템의 테스트는 개발단계에 따라 하드웨어 뿐 만아니라 소프트웨어에 대한 지식이 상당부분 요구된다. 그러므로 본 논문에서는 PC기반의 소프트웨어 테스트 기법을 그대로 활용할 수 있도록 하기 위해 임베디드 시스템의 다양한 테스트 단계 중 Control Logic과 시스템 통합 부분에

대한 테스트 기법에 한해서만 설명하고자 한다.

#### 4. 임베디드 테스트 시스템의 구성

임베디드 소프트웨어는 엔진, 구조물 등과 같은 기계 시스템을 직접 컨트롤하거나 다른 시스템의 동작을 제어하는 형태의 소프트웨어가 많다. 이런 경우 한번 장착되면 사용자에게 잘 노출되지 않으

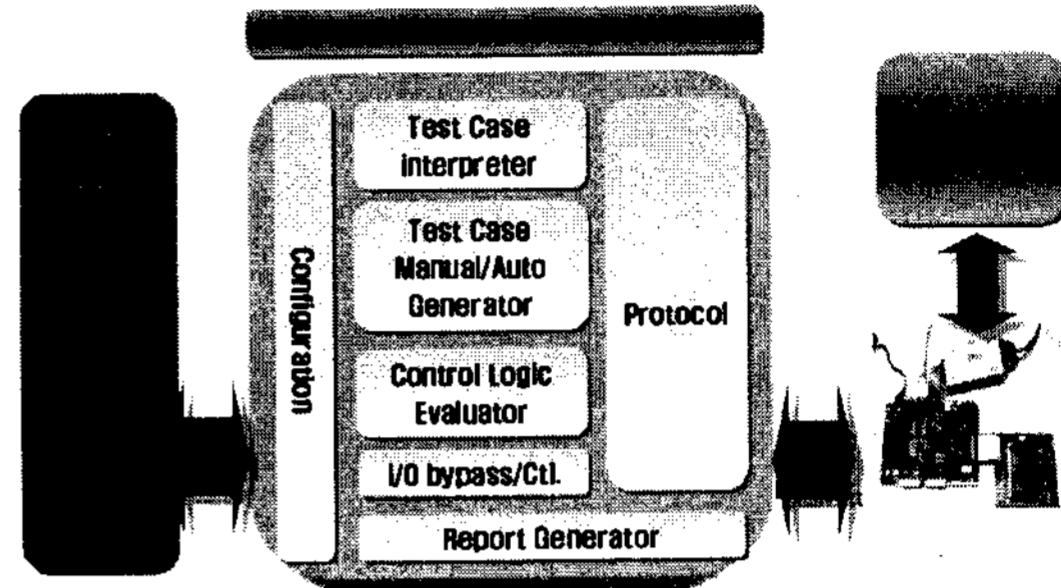


그림 2. 수동/자동 테스트 시스템 SW구성도

므로 기능의 정상동작에 대한 신뢰성은 테스트항목으로 매우 중요하게 고려되어야 한다. 따라서 컨트롤 시스템(임베디드 시스템)의 Control Logic에 대한 입/출력의 확인은 매우 중요하다. 그러나 일반적으로 입력 및 출력은 디지털 뿐 아니라, 아날로그 입력의 형태를 가지고 있으며 Control Logic의 입력 파라미터가 수~수십 개이므로 가능한 테스트 케이스의 개수는 수백만~수천만 개에 이른다.

모든 테스트 케이스에 대한 테스트를 수동으로 수행할 수 없으므로, 테스트 케이스를 줄이는 기법과 병행해서 자동화 테스트 시스템을 개발하여 사용하는 것은 매우 유용하다.

임베디드 시스템의 출력을 볼 수 있는 출력장치가 제공되지 않으므로 시리얼 또는 USB를 사용한 통신을 통해 출력 값을 확인 할 수 있다. 이러한 테스트 시스템에서는 잘 설계된 프로토콜이 테스트 시 시스템 제어 및 모니터를 위해 필수적이다. 자동화 테스트 시스템을 설계할 경우에는 그림 2에서와 같이 Control Logic의 정확성 여부를 판단

하는 제어 판단기(Control Logic Evaluator)와 자동으로 생성한 테스트 케이스를 읽어온 후 해석하는 테스트 케이스 해석기(Test Case Interpreter), 시험 결과를 리포트로 작성하는 틀인 리포트 생성기(Report Generator)의 주요 모듈을 추가로 구성할 수 있다. 또한 Control Logic의 동작 상태를 쉽게 확인하기 위해서는 틀 들을 사용하여 모니터링 화면을 구성할 수도 있다.

그림 3에서와 같이 Control Logic을 테스트하기 위해서는 위에서 설명한 자동 테스트 시스템을 사용하여 각종 입력(Virtual Input)을 조정한 후 출력(Simulator Monitors)를 확인하여 제어로직의 정상 작동 여부를 확인하는 방식으로 구성한다[3].

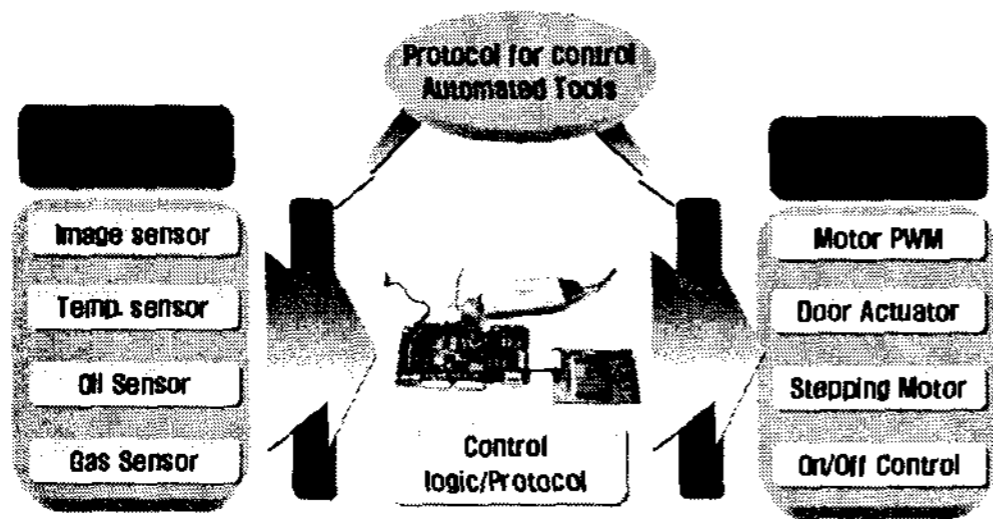


그림 3. Control Logic에 대한 테스트

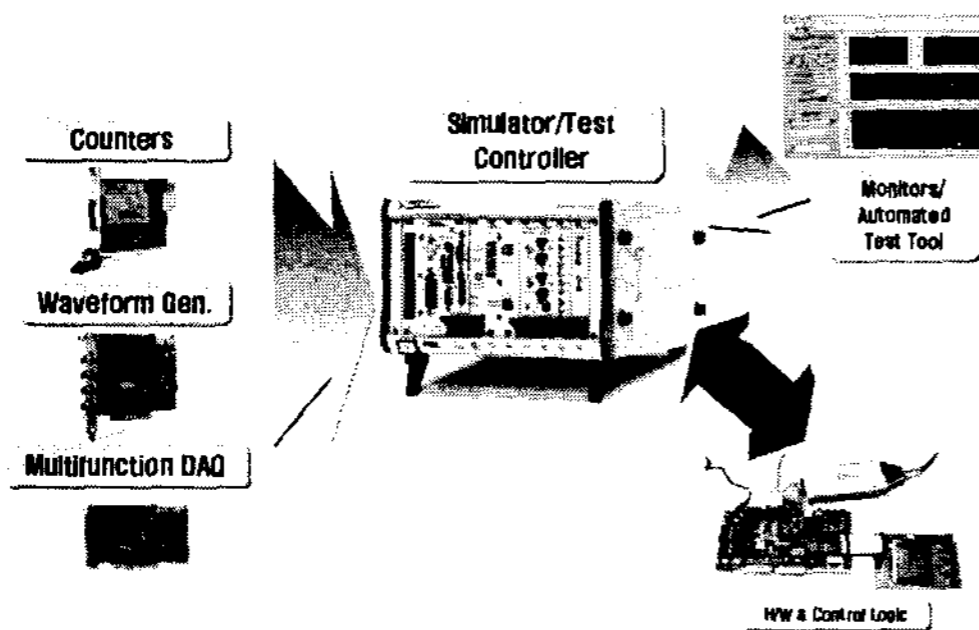


그림 4. 임베디드 테스트 하드웨어 구성 예

지금까지 Control Logic레벨에 대한 테스트 방법을 설명하였다. 그러나 앞서 언급 바와 같이 임베디드 시스템의 블랙박스 테스트에서 중요한 부분은 시스템 통합에 대한 테스트이다. 시스템 통합 테스트(System Validation Test)를 수행하기 위해서는 다른 시스템과의 통합(System

Integration)이 필요하다. 그러나 일반적으로 임베디드 시스템의 통합 시스템을 구성하여 테스트하기는 많은 제약이 따른다. 따라서 입력 및 출력의 값을 생성할 수 있는 하드웨어를 구성하여 시험을 수행하는 방식을 사용한다. 예를 들어 디지털 및 아날로그 입력을 만들기 위해 그림 4와 같이 Multi Function I/O 및 Waveform Generator를 이용하거나 출력을 생성 및 모니터링 하기 위해 Multi function I/O를 사용하여 시스템 통합 테스트 환경을 구성한다.

## 5. 결론

임베디드 시스템 소프트웨어에 대한 테스트는 하드웨어 및 소프트웨어 개발 단계별로 다양하다. 임베디드 시스템의 특정 상 Control Logic 기능의 정상동작에 대한 신뢰성 확보가 시스템의 품질을 좌우 하므로 이에 대한 테스트인 Control Logic 및 System 통합 테스트가 매우 중요함을 보였다.

Control Logic의 특성상 수많은 테스트 케이스를 수동으로 테스트 할 수 없으므로, 자동화된 테스트 도구의 사용이 필요하며 이를 구성하기 위한 소프트웨어 모듈을 설명하였으며 시스템 통합 테스트를 위한 하드웨어 시스템 구성도 설명하였다. 본 논문에서는 임베디드 시스템 소프트웨어 품질 향상을 위해 타 시스템에 쉽게 적용할 수 있는 임베디드 시스템 하드웨어 및 소프트웨어 구성을 소개하였다.

## 참 고 문 헌

- [1] Gray Pollice, *Testing Embedded Software*
- [2] Madiseti, Vijay K., Akgul, Tankut *Debugging Embedded Systems*, 2006.
- [3] Broekman, Bart & Notenboom, Edwin, *Testing Embedded Software*