

---

# 모델링 기법을 이용한 차량용 IPC 소프트웨어구조 설계 및 구현

송봉기\* · 유윤식\*\*

A Design and Implementation of Software Architecture for IPC in Vehicles  
Using Modeling Methodology

Bong-Gi Song\* · Yun-Sik Yu\*\*

## 요 약

자동차용 IPC는 운전자에게 차량의 속도, RPM, 주행 거리 등의 운행 정보를 제공하는 장치이다. 기존의 전통적인 IPC의 경우 주로 기계식으로 구현되고 있으나 현재 사용자의 편의성 증대를 위한 요구 증가에 따라 IPC에도 그래픽 요소와 효율적인 정보제공을 위한 사용자친화적인 형태로의 변화가 필요하다. 이를 위해서는 모델링 기법을 이용하여 소프트웨어의 유연성과 강건성을 갖는 IPC의 소프트웨어 구조가 필요하다. 이에 본 논문에서는 MVC 모델과 UML을 이용하여 IPC 소프트웨어 구조를 설계하고 구현한다. 뷰를 분리하여 개발함으로써 다양한 형태의 정보 화면을 개발할 수 있고 또한 UML의 상태 다이어그램을 통해 뷰의 전환 등을 간단한 상태의 변화로 처리하여 개발이 용이하다. 제안하는 IPC 시스템은 개발 기간의 단축을 통한 비용 절감 및 MVC 모델 구조에 따른 유지보수의 용이성을 기대할 수 있다.

## ABSTRACT

An IPC(instrument panel Cluster) is a useful device that provides driving information to the driver. The information includes the vehicle speed, RPM, mileage, etc. The traditional IPC has been mostly implemented with mechanical technique. According to increment of needs for the convenience of IPC by user, the IPC must provide graphical interface and efficient driving information. Also the user-friendly IPC is needed by drivers. Thus flexible and robust software structure and development methods are required in order to develop IPC. In this paper, we propose software architecture and design method for the IPC using modeling method. We use MVC model and UML to model software architecture because they have flexible and robust characteristics. We can develop the various forms of information screen by separating views from model by using state diagram and class diagram in UML. Through this, the cost saving and ease of maintenance can be expected. The development time and cost can be reduced by using proposed method.

## 키워드

IPC(Instrument Panel Cluster), Modeling, Software Architecture, MVC(Model-View-Controller) Model  
차량용 계기판, 모델링 기법, 소프트웨어 구조, 모델-뷰-컨트롤러 모델

---

\* 부산IT융합부품연구소 팀장(alphamir@deu.ac.kr)

\*\* 교신저자 : 부산IT융합부품연구소 소장(ysyu@deu.ac.kr)

접수일자 : 2012 07. 02

심사(수정)일자 : 2012 11. 21

게재확정일자 : 2012 12. 10

## 1. 서론

자동차에서 운전자와 가장 밀접한 인터페이스인 IPC(Instrument Panel Cluster)는 전통적으로 기계식 게이지를 이용하여 구현되어왔다. 최근 IT기술과 자동차 기술의 융합과 사용자 친화적인 인터페이스에 대한 요구가 증가함에 따라 IPC에서도 그래픽 요소의 적용, 전장장치로써의 변화가 요구되고 있다[1][2][3]. 이를 위하여 IPC에 LCD가 적용되고 있으며, 게이지와 LCD의 혼용 혹은 LCD 단독으로 구현된 IPC 등이 개발되고 상용화되고 있다. 어떤 형태이던 간에 LCD를 적용한 IPC의 개발을 위해서는 LCD에 표시되는 트립 컴퓨터 정보 혹은 차량 정보를 처리하기 위한 임베디드 소프트웨어가 필요하며 이 소프트웨어의 효과적인 개발 방법이 요구된다[4].

본 논문에서는 IPC의 소프트웨어 구조를 설계하고 구현하는 효과적인 방법으로 UML(Unified Modeling Language) 기반의 모델링 기법을 이용한다. IPC의 소프트웨어 구조를 MVC(Model-View-Controller) 모델에 입각하여 설계하고 이를 UML로 표현하여 구현한다. MVC 모델에 따라 모델에서 차량용 정보 및 트립 컴퓨터의 정보를 처리하고, 뷰에서 IPC의 인터페이스 부분을 제공한다. 또한 컨트롤러에서 뷰와 모델 사이의 동작 제어 및 외부와의 통신을 담당한다. 이를 통해 유연성 있고 강건한 소프트웨어 구조를 설계한다. 또한, UML의 클래스 다이어그램, 상태 다이어그램을 적용하여 설계의 용이성을 제공하고 스킨 등 효과적인 사용자 인터페이스의 제공 방식을 설계한다. 본 논문은 2장에서 기존의 연구들에 대해 설명하고, 3장에서 모델링 기법의 소프트웨어 구조 설계 방법을 제안한다. 4장에서 실제 구현 예를 제시하고 5장에서 결론을 맺는다.

## II. 기존의 연구들

### 2.1. 차량용 IPC 시스템 개발

차량에서 차량의 정보 및 운행 정보를 표현하기 위한 장치로 사용되는 IPC 시스템은 직접적으로 운전자와 대면하는 HMI(Human Machine Interface)의 일종으로 매우 중요한 장치이다. 이에 다양한 형태의 IPC

들이 자동차 제조사 및 협력업체들을 통해서 개발되고 있고, 학술적으로 가시성을 지향하거나 편의성을 지향하는 형태의 여러 IPC 시스템에 대한 연구가 이루어지고 있다. 하지만 대부분 인지공학적 측면과 기계적인 형태의 IPC에 대한 연구가 주로 이루어졌다. 남택수[5] 등은 인지중심의 WDA(Work Domain Analysis) 기법을 적용한 IPC 개발 방법을 제안하였다. 이를 통해 사용자 친화적인 IPC의 개발방법을 제안하였으나 실제 구현 시 적용되는 소프트웨어 구조 및 구현방법에 대해서는 연구되지 않았다.

IPC의 구현에 대한 연구가 진병욱[6], 정상봉[7] 등에 의해 이루어졌으나 이 또한 기본적인 하드웨어 구조에 대한 연구만이 수행되고 기계식 시스템에 대한 연구만이 수행되었다. 디지털 시스템으로 구현되는 IPC의 소프트웨어 구조에 대한 연구는 이루어지지 않았다. ZHANG Wei[8] 등은 IPC를 평가하는 방법에 대해 연구를 하였다. 이 연구에서도 평가 방법만이 제시되었고 실제 설계 및 구현에 대한 방법은 연구되지 않았다.

이상진[4], 이광렬[9] 등에 의해서 차량 전장시스템 소프트웨어 개발에 대한 모델링 기법의 적용에 대한 연구가 수행되었다. 하지만 이 연구에서도 IPC에 대한 세부 특성에 대한 연구는 이루어지지 않았고 전장장치 소프트웨어 전반에 대한 추상적인 모델만이 제시되었다.

### 2.2. MVC 모델

MVC 모델은 응용프로그램을 3-계층 구조에 따라서 모델, 뷰, 컨트롤러의 3가지 구성요소로 응용프로그램을 분할한다. MVC 모델은 다음 그림과 같이 구성된다.

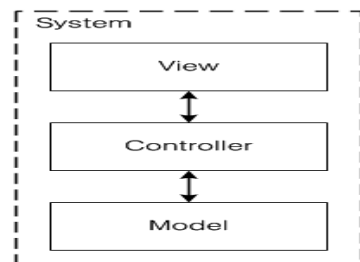


그림 1. MVC 모델  
Fig. 1 MVC model

모델은 데이터와 로직을 관리하는 비즈니스 규칙을 관리하고 데이터의 접근과 갱신을 담당한다. 뷰는 모델의 콘텐츠를 제시하고 실제 사용자 인터페이스를 제공한다. 그리고 컨트롤러는 뷰를 통해 사용자로부터 얻어진 정보를 모델에 전달하여 데이터와 연계하는 역할을 담당한다[10]. 단순한 구조와 각 영역간의 분산 개발이 가능한 특징으로 MVC 모델은 다양한 응용개발에 사용된다[11][12].

### III. IPC 소프트웨어 구조 설계

본 논문에서는 IPC 소프트웨어 개발을 위하여 MVC 모델과 UML 기반의 모델링 기법을 적용하는 소프트웨어 구조를 설계한다. 이를 통해 차량의 각종 정보들과 사용자 뷰를 분할하여 유연한 구조의 응용 개발이 가능하고, 협업 개발이 가능한 구조를 제안한다. 클러스터에 게이지와 함께 표시등을 표현하기 위해서 소프트웨어 구조는 다음 그림과 같이 구성된다.

HAL 계층을 두고 외부 ECU와의 통신 시 하드웨어 의존성을 없앤다. 통신의 물리계층은 RS-232C와 CAN 통신이 가능하나 상위의 미들웨어에서는 하위가 어떤 통신 방식인지를 고려하지 않도록 의존성을 낮추는 HAL 계층을 이용하였다. 이 때 RS-232C 통신의 패킷 프레임은 CAN의 Payload 부분에 캡슐화할 수 있도록 8바이트로 설계하여 호환성 및 계층성을 갖는다.

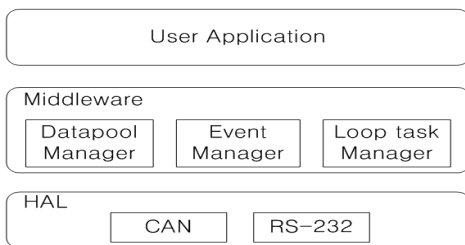


그림 2. 소프트웨어 구조  
Fig. 2 S/W architecture

미들웨어는 데이터 풀 매니저, 이벤트 매니저, 루프 매니저로 구성되어 상위단인 클러스터(Cluster) 어플리케이션과 통신하는 통로를 제공한다. 데이터 풀 매

니저는 HAL 계층을 통해 전달된 데이터를 클러스터 어플리케이션으로 전달한다. 이벤트 매니저는 외부 신호로 클러스터 어플리케이션에 이벤트를 발생시키는 역할을 수행한다. 즉, 이벤트 소스를 외부 신호로 대체하는 역할을 한다. 루프 매니저는 반복 수행되는 작업을 수행하기 위해서 사용된다. 주기적으로 표시되는 데이터와 이벤트의 처리를 담당한다.

HAL 계층에서 차량의 ECU로부터 데이터를 수신하기 위해서 LLD(Low level driver)를 이용한다. 이때 주로 이용되는 통신 프로토콜은 CAN과 RS-232C가 있다. 제안하는 시스템에서도 CAN과 RS-232C를 지원하기 위해서 통신 패킷 프레임을 정의해서 사용한다. RS-232C 통신을 위해 사용되는 데이터 패킷 프레임은 다음 그림 3과 같이 SOP, Code, Payload, Checksum, EOP로 구성된다.

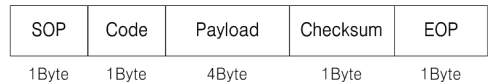


그림 3. 시리얼 데이터 패킷  
Fig. 3 Data packet (RS-232C)

RS-232C 통신용 데이터 패킷의 길이를 CAN 통신 프로토콜의 데이터 길이와 일치시킴으로써 상호 데이터 변환이 용이하게 하였다. 패킷의 구성 중 SOP(Start Of Packet)은 데이터 패킷의 시작을 나타낸다. 코드(Code)는 패킷의 기능을 인코딩하여 표현한다. 유효데이터(Payload)는 코드의 기능을 세부적으로 분류하는 역할을 한다. 코드 값에 따라 1바이트에서 4바이트까지 유효 데이터의 길이가 다르다. 즉, 기능성에 따라 1바이트 단위로 분할 가능한 4바이트 데이터이다. 에러 체크 코드인 체크섬(Checksum)은 일반적으로 사용되는 다음 식 1과 같이 나타낼 수 있다.

$$Checksum = -(Code + \sum_{i=0}^3 Payload_i) \quad (1)$$

데이터 전송 간에 발생한 오류 여부는 수신한 데이터와 체크섬의 합으로 알 수 있다. 예러는 다음 식 2로 알 수 있다.

$$Error = \begin{cases} 0, & \sum_{i=1}^6 Received\ data_i = 0 \\ 1, & \sum_{i=1}^6 Received\ data_i \neq 0 \end{cases} \quad (2)$$

마지막으로 EOP(End of Packet)는 데이터 패킷의 끝을 나타낸다.

RS-232C 데이터 패킷을 CAN 네트워크에 전송하기 위해서 CAN 프레임으로 캡슐화할 수 있다. 이 때 CAN 프레임에 캡슐화된 CAN 데이터 프레임의 구조는 다음 그림 4와 같다.

|      |                   |               |                       |           |           |      |
|------|-------------------|---------------|-----------------------|-----------|-----------|------|
| SOF  | Arbitration Field | Control Field | Data Packet (RS-232c) | CRC Field | ACK Field | EOF  |
| 1bit | 12bit             | 6bit          | 64bit(8Byte)          | 16bit     | 2bit      | 7bit |

그림 4. CAN 데이터 프레임  
Fig. 4 CAN data frame

RS-232C 통신의 데이터를 CAN 데이터 프레임에 캡슐화할 수 있는 구조를 이용하여 향후 RS-232C와 CAN 통신 사이의 게이트웨이 기능이 필요한 경우에 데이터의 변환을 용이하게 할 수 있다.

제안한 시스템에서 컨트롤러와 모델의 역할을 수행하는 미들웨어에서 사용되는 클래스를 UML의 클래스 다이어그램(Class Diagram)으로 설계하면 다음 그림 5와 같다.

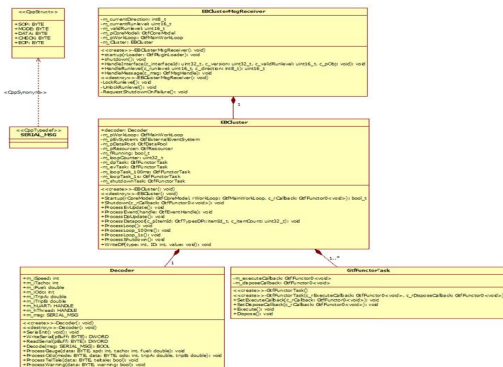


그림 5. IPC 시스템의 미들웨어 클래스 다이어그램  
Fig. 5 Middleware class diagram of IPC system

각 관리자의 기능을 수행하는 다수의 FactorTask

클래스와 클러스터 클래스 및 ContentMsgReceiver 클래스가 컨트롤러로 동작하고 HAL 및 LLD의 역할을 수행하는 Decoder가 모델의 역할을 수행한다. 컨트롤러와 모델에서 처리된 데이터가 ContentMsgReceiver클래스를 통해 뷰와 연결된다.

제안한 IPC 소프트웨어 구조의 전체 동작 알고리즘은 다음 그림 6의 순서도로 나타낼 수 있다. 먼저 하드웨어와 디바이스 드라이버 및 미들웨어 태스크들에 대한 초기화를 수행한다. 초기화가 완료된 후 HAL을 통한 데이터의 수신을 체크한다. 데이터가 수신된 경우는 데이터를 해석하고 이 과정에서 취득된 차량 데이터 및 뷰에서 요구되는 데이터로 데이터 풀을 갱신한다. 반복 루프에서 종료 시까지 반복 수행을 한다.

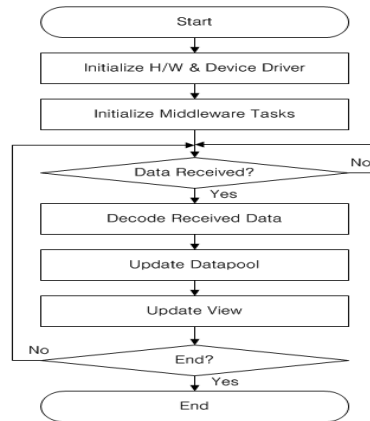


그림 6. IPC 소프트웨어 동작 순서도  
Fig. 6 Flowchart of IPC S/W

#### IV. IPC 소프트웨어 구현 및 테스트

본 논문에서 제안한 IPC 소프트웨어를 다음 표 1과 같은 구현 환경에서 구현하고 테스트하였다.

표 1. 구현 환경  
Table. 1 Implementation environment

|     |                                    |              |
|-----|------------------------------------|--------------|
| 구현  | 구현환경                               |              |
| OS  | Windows XP, Embedded Linux(ARM 기반) |              |
| MVC | View                               | EB Guide 5.3 |

|                |                              |
|----------------|------------------------------|
| Windows XP     | Visual C++ (DLL 형태)          |
| Embedded Linux | Eclipse CDT cross compile 환경 |

운영체제 환경은 PC환경과 실차에 적용될 수 있는 Embedded Linux를 이용하였다. 개발 도구로는 뷰의 개발을 위해 EB Guide 5.3을 이용하였고 모델 및 컨트롤러의 개발에는 Visual C++와 Eclipse CDT를 각 운영체제에 맞게 이용하였다. 개발 형태는 실행코드 형태의 뷰 프로그램에 모델과 컨트롤러를 라이브러리 형태로 개발하여 뷰와 모델, 컨트롤러를 분리하였다. 즉, 뷰 혹은 모델, 컨트롤러의 변경이 필요할 때 개별 부분만 수정 개발함으로써 높은 유연성을 가질 수 있다. 라이브러리는 PC의 경우 DLL 형태로 구현하여 메모리 요구량을 줄였고 Embedded Linux의 경우에는 shared library 형태로 구현하였다. Embedded Linux 환경에서의 개발을 위해 ARM 기반의 cross compiler를 이용하였다. 동작을 테스트한 환경은 다음 그림 7과 같다.

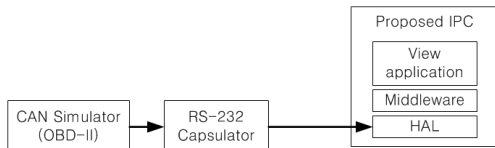


그림 7. 테스트 환경  
Fig. 7 Test environment

구현된 결과는 다음 그림 8과 같다.

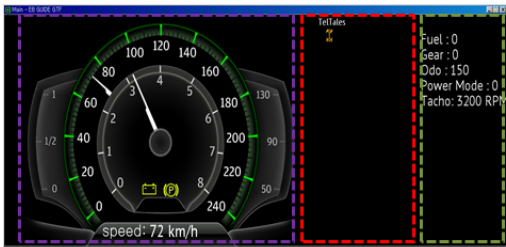


그림 8. 구현 화면  
Fig. 8 Implemented IPC

뷰 응용프로그램은 EB Guide를 이용하였고 미들웨어 및 HAL은 Visual C++로 구현하였다. CANoe 7.0과 CAN CaseXL을 이용하여 차량과 유사한 CAN DB를 이용하여 OBD-II 신호를 생성하고 이를 RS-232C 프로토콜로 캡슐화하였다. 이렇게 함으로써 CAN 컨트롤러가 없는 환경에서도 쉽게 데이터 처리가 가능하게 하였다. 생성된 데이터를 IPC 소프트웨어에서 HAL계층을 통해 수신한다. IPC의 HAL 계층에서 이를 수신하여 수신된 데이터는 미들웨어 계층에서 해석되어 데이터 풀에 저장되고 뷰에서 데이터 풀의 데이터를 참조하여 각종 게이지와 표시등의 점멸을 동작시킨다.

외부 시뮬레이터에서 속도 70 km/h, 3200RPM의 게이지가 각각 계기판 메인에 정상적으로 표시됨을 확인할 수 있었다. 표시등 표시 영역에 WDA 표시등을 점등한 동작에 대해서 정상 동작하였다. 이 때 누적거리계의 값도 정상적으로 수신되어 트립정보 표시 영역에 표시됨을 확인할 수 있다.

## V. 결론

본 논문에서는 3-계층 구조의 유연한 개발 모델인 MVC 모델을 이용하여 협업개발에 적합한 IPC 소프트웨어 구조를 제안하였다. 제안한 시스템을 UML을 이용하여 설계하고 PC 및 Embedded 환경에서 구현하였다. MVC 모델과 UML 모델링 기법을 적용하여 개발 기간의 단축과 재활용성이 높은 소프트웨어 구조를 개발하였다. 또한 개발 기간 단축과 유지보수 비용의 감소로 전체 개발 비용 절감효과를 기대할 수 있는 IPC 소프트웨어 개발방법을 제안하였다. 구현한 시스템을 실차에 적용되는 시뮬레이션 환경에서 IPC와 응답성 등을 평가하여 유효성을 확인하였고, 향후 IPC 개발에 유용하게 적용될 수 있을 것을 기대한다.

### 감사의 글

본 연구는 지식경제부(정보통신산업진흥원), 부산광역시 및 동의대학교의 지원을 받아 수행된 연구결과임. (B1100-1101-0010, IT특화연구소: "부산IT융합 부품연구소" 설립 및 운영)

참고 문헌

[1] 전황수, “자동차-IT 융합 기술 및 국내외 개발 동향”, 한국멀티미디어학회지, 14권, 1호, pp. 1-10, 2010.

[2] 이소연, “차내망 인터페이스”, TTA Journal No. 117, 2008.

[3] 윤성하, 손희배, 이영철, “차세대 자동차 통합스마트 모니터 시스템에 관한 연구”, 한국전자통신학회논문지, 6권, 3호, pp. 439-444, 2011.

[4] 이상진, 손장경, 이성훈, 김명진, 이선봉, “차량 전장시스템의 개방형 소프트웨어 설계를 위한 개발 방법에 관한 연구”, 대한전자공학회 학술대회 논문집, pp. 829-830, 2007.

[5] 남택수, 명노해, 홍승권, “자동차 계기판 개발을 위한 WDA(Work Domain Analysis) 적용”, 대한인간공학회지, 26권, 4호, pp. 127-133, 2007.

[6] 진병욱, 김봉열, 이문기, “자동차용 전자계기판 및 전용 LCD 집적회로의 설계”, 대한전자공학회 학술대회논문집, 8권, 2호, pp. 692-695, 1985.

[7] 정상봉, 정동명, 박장규, 김해운, 홍승홍, “자동차 계기판의 LCD 디지털 DISPLAY 및 음성 출력 시스템의 개발”, 대한전자공학회 학술대회 논문집, 12권, 1호, pp. 200-203, 1989.

[8] Zhang Wei, Zhang Xian-kui, Zhu Lei, Zhang Rui and Chen Zhi-Bin, “Study on the Evaluation System of Instrument 클러스터”, Computer-Aided Industrial Design and Conceptual Design, 2006. CAIDCD '06. 7th International Conference, pp.1~5, 2006.

[9] 이광열, 홍성준, 오승호, 이우택, “모델 기반 차량 전장 시스템 개발 프로세스에 관한 연구”, 한국자동차공학회, 2006년 추계학술대회 논문집 3권, pp. 1480-1485, 2006.

[10] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1994

[11] 이준희, “MVC 프레임워크 기반의 CMS 설계 및 구현”, 한국콘텐츠학회/한국통신학회 2003 추계 종합학술대회 논문집, 1권, 2호, pp. 157~160, 2003

[12] 정세훈, 권용욱, 심춘보, “자동차 번호판 인식 및 스마트폰을 활용한 객체지향 설계 기반의 효율적인 차량 관리 시스템”, 한국전자통신학회논문지, 7권, 5호, pp. 1153-1164, 2012.

저자 소개



**송봉기(Bong-Gi Song)**

1997년 부경대학교 전자공학과 졸업 (공학사)

2000년 부경대학교 대학원 정보시스템학과 졸업(공학석사)

2005년 부경대학교 대학원 정보시스템학과 졸업(공학박사)

2005년~2008년 경성대학교 컴퓨터과학과 초빙교수

2008~현재 동의대학교 부산IT융합부품연구소 연구개발부 팀장

※ 관심분야 : 임베디드시스템, 차량용 FW & SW



**유윤식(Yun-Sik Yu)**

1979년 부산대학교 물리학과 졸업 (이학석사)

1992년 부산대학교 물리학과 졸업 (이학박사)

1983~2010년 동의대학교 물리학과 교수

2008~현재 동의대학교 부산IT융합부품연구소장

2010~현재 IT특화연구소협의회 회장

2010~현재 동의대학교 방사선학과 교수

※ 관심분야 : FBG광센서, MEMS센서, 광통신