

# 퍼스널 로봇 미들웨어를 위한 CAN (Controller Area Network) Protocol

## Controller Area Network (CAN) Protocol for Personal Robot Middleware

박 태 규\*, Li Vitaly\*\*, 박 홍 성\*\*\*  
(Tai Kyu Park, Li Vitaly and Hong Seong Park)

**Abstract** - Personal robot consist of various modules that have independent functions. Because personal robot has requirement that support various construction for user's tendency. Therefore, Middleware mechanism that support not only personal robot's main functional element but also embedded modules functional elements is required. Each module have various heterogeneous network interfaces and variable services and variables. Therefore, Middleware must support these various network interfaces. This paper, pointed in Controller Area Network(CAN) interface that usually used in embedded system for control. For connect various heterogeneous network interfaces (Ethernet, RS232 etc..), it is necessary to modify basic CAN frame format. And also make some kind of BUS topology for CAN network.

**Key Words** : Personal robot, Middleware, CAN, heterogeneous network.

### 1. Introduction

#### 1.1 Personal Robot

초창기의 로봇 기술은 산업 현장에서 인간이 하던 일을 제한적으로 대신하는 도구로 활용하기 위한 대안으로부터 시작하였다. 이는 반복적이고 단순한 작업을 로봇에게 맡김으로써 인간에게는 편리성을, 산업 일선에게는 생산성의 증가를 갖고 왔으며, 로봇은 점차 사람의 능력으로는 하기 힘든 정교한 작업을 전문적으로 수행하는 또 하나의 산업 노동자원으로 진화해 오고 있다. 산업 현장에서 주로 사용되던 로봇은 점차 인간의 다양한 실생활에서도 사용될 수 있도록 발전하였고, 이로 인해 퍼스널 로봇이 만들어지게 이른다.

산업 현장에서 사용되는 로봇이 고정적인 위치에서 반복적이고 정교한 작업을 하는 것과는 다르게, 퍼스널 로봇은 가정 내에서 다양한 기능을 수행하기 위해 이동성 및 편리성, 그리고 다양한 가내 시스템과 연동될 수 있는 네트워크 기능을 지원해야 한다. 퍼스널 로봇의 경우, 이동성 지원을 위함 유선보다는 무선 네트워크를 사용하는 것이 일반적이며, 가정 내의 다양한 디지털 기기들과 데이터를 주고받으며 관리할 수 있어야 한다. 더불어 홈 게이트웨이를 이용하여 인터넷과 연결하여 집 외부에서도 로봇이나 가전제품을 제어할 수 있어야 한다.

#### 저자 소개

- \* 非會員 : 江原大學校 通信 멀티미디어 工學科
- \*\* 非會員 : 江原大學校 通信 멀티미디어 工學科
- \*\*\*非會員 : 江原大學校 電氣電子情報通信 工學部

기존의 많은 퍼스널 로봇들이 이런 기능을 하기 위해서 개발되었으나, 개발자들이 로봇을 일체형으로 제공하였기 때문에 자연히 기능이 제한될 수밖에 없었다. 로봇이 제공하는 네트워크 인터페이스가 한정적이어서 다양한 인터페이스를 갖는 디지털 가전 기기와 연동이 되지 않았으며, 관리할 수 있는 가내 시스템도 제한되었다.[1]

이러한 문제들을 해결하기 위해서 퍼스널 로봇은 각각 독립성을 갖는 다양한 모듈들을 조립하여 사용할 수 있어야 한다. 사용자는 가격이나 용도에 따라서 모듈을 부착하여 퍼스널 로봇을 사용하게 될 것이며, 이로 인해 일체형 로봇이 지니던 기능적 한계를 극복할 수 있을 것이다.

때문에 퍼스널 로봇을 구성하는 다양한 모듈들이 사용하는 네트워크 인터페이스(Ethernet, IEEE1394, Bluetooth, WLAN, CAN 등)를 지원하기 위하여 퍼스널 로봇 내부에 자체적으로 네트워크 인터페이스를 선택하고 메시지를 전달할 수 있는 미들웨어가 필요하다.[2]

본 논문은 CAN (Controller Area Network) 인터페이스를 사용하는 모듈이 퍼스널 로봇에 안정적으로 연결되고, 미들웨어와의 원활한 연동을 위하여 기본 CAN 프로토콜을 수정할 것을 제안한다. 2절에서는 퍼스널 로봇 미들웨어와 CAN에 대하여 간략히 소개하고, 3절에서는 수정한 CAN 프로토콜에 대하여 소개하며, 4절에서 결론과 앞으로 할 일에 대하여 다룬다.

### 2. 퍼스널 로봇 미들웨어

#### 2.1. 미들웨어

퍼스널 로봇은 하나 이상의 개방형 인터페이스에 기반하여 퍼스널 로봇을 개발해야 한다. 개방형 인터페이스 기반 퍼스널 로봇은 어떤 인터페이스의 모듈이 장착되더라도 새로 장착된 모듈이 퍼스널 로봇에서 정상적인 동작을 할 수 있도록 한다.

이에 따라 제안한 미들웨어의 구조는 그림 1과 같으며 스트리밍 계층 (Streaming Layer), 네트워크 적응 계층 (Network Adaptation Layer), 인터페이스 계층(Network Interface Layer)로 구성되어 있다. 스트리밍 계층에서는 미들웨어의 서비스를 사용하는 어플리케이션을 관리하고 원격의 어플리케이션과 메시지를 주고받기 위한 트랜잭션과 데이터의 재표현을 한다. 네트워크 적응 계층은 퍼스널 로봇 내부의 모듈 간 데이터 전송을 하기 위한 메커니즘을 제공한다. 즉, 개방형 이종 인터페이스를 지원하기 위한 모듈 어드레싱, 이종 네트워크에 따른 패킷타이제이션, 이종 네트워크 간 메시지 라우팅, 실시간 및 비실시간 전송 서비스 등의 기능을 제공한다. 네트워크 인터페이스 계층은 각 인터페이스 들에 대한 하드웨어 초기화 및 연결 초기화를 하고, 네트워크 적응 계층에서 데이터 전송 요청이 오면, 이를 인터페이스마다 다르게 제공되는 MTU(Maximum Transmission Unit)에 맞춰 데이터를 단편화(Fragmentation)하고 이를 재구성(Reassemble) 한다. 그리고 네트워크 적응 계층과 마찬가지로 실시간, 비실시간 전송 서비스를 처리하는데, 이 서비스를 적응계층과 중복해서 하는 이유는 앞서 언급한 단편화 기능 EOans이다. 단편화 한 데이터를 보내는 도중 제어 데이터나 긴급 데이터가 발생할 경우에 대비할 수 있어야 하기 때문이다.

## MIDDLE WARE

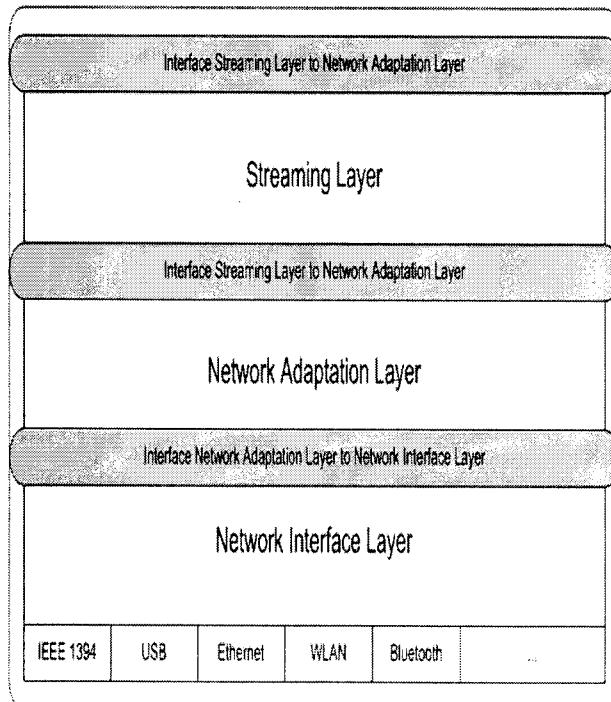


그림 1. 미들웨어 구조

## 2.2. CAN (Controller Area Network)

CAN은 독일 BOSCH사에서 늘어나는 자동차 내의 많은 ECU(Electronic Control Unit) 간의 정보 공유를 위하여 개발되었다. 이는 자동차와 같은 노이즈가 많은 환경에 적합하도록 꼬임 쌍선 (Twisted Pair) 케이블을 전송 매체로 사용한다. CAN BUS의 특징은 Multi-master, Multi-slave 방식을 지원한다는 것이다. 다시 말해 하나의 CAN 노드가 보낸 데이터는 필요에 따라 여러 노드가 공유할 수 있다. 그리고 공유 직렬 버스 형식 (Shared Serial Bus Type)으로 외부 전자파나 노이즈에 강인한 특성을 갖는다. 이는 외부 전자파나 노이즈가 영향을 끼치더라도 시리얼 라인의 위상 차이에는 영향을 미치지 않는다는 것에 기인한다. 이론적으로는 2032개의 서로 다른 노드를 연결하여 네트워크를 구성할 수 있으나, 현존하는 CAN 트랜시버의 한계로 약 100개 까지의 노드를 연결하여 사용할 수 있다. 그리고 통신 속도는 실시간 제어도 가능한 1Mbps (ISO11898 규격)의 고속 통신을 제공한다.[3]

CAN 2.0A Specification에 따른 CAN 프레임 포맷은 그림 2 와 같으며, 크게 세 가지 필드로 구분할 수 있다.

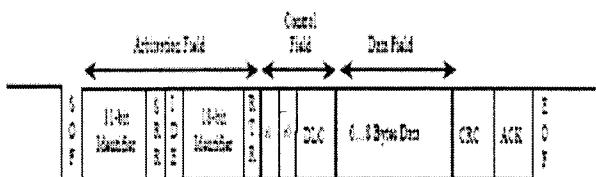


그림 2. CAN 프레임 포맷.

- Arbitration Field : CAN 메시지마다 고유하게 부여되는 메시지 식별자에 관한 정보를 담고 있다. 이 식별자는 메시지를 구분하는 것과 동시에 메시지의 우선순위도 나타낸다. CAN 2.0A의 경우엔 11bit, CAN 2.0B의 경우엔 29bit가 제공된다.
- Control Field : Data Length Code 같은 메시지에 대한 제어 정보를 담고 있다. 표준에서는 오직 DLC만 제공하고 r1, r0 같이 Reserve 된 필드들은 CAN 컨트롤러에 따라 다양하게 응용된다.
- Data Field : 실제 전달될 데이터를 담고 있다. CAN 노드가 한 번에 보낼 수 있는 최대 크기의 데이터는 8 byte이다. CAN 노드는 데이터를 보내기 전에 우선 CAN 버스 라인을 사용 중인지 여부를 파악한다. 동시에 메시지 간 충돌을 검출하는데 이는 Arbitration Field를 참고하여 버스에 동시에 전송되려 하는 데이터의 우선순위를 판별하고 높은 우선순위를 갖는 프레임은 전송하고 낮은 우선순위를 갖는 프레임은 전송을 위해 대기한다. CAN 네트워크상에 연결된 모든 노드는 CAN 버스 상에 있는 메시지를 수신한 뒤, 메시지 ID를 통하여 자신이 필요로 하는 데이터인지를 판단하고 수신하거나 버린다.

## 3. 수정한 CAN 프로토콜

CAN 네트워크 인터페이스의 경우 MTU가 8byte로 제한되기 때문에 용량이 큰 데이터를 보내는 것에는 적합하지 않다고 할 수 있다. 그리고 Multi-master, Multi-slave 를 지

향하기 때문에 경우에 따라서 노드의 구분을 할 수 없을 수도 있다. 퍼스널 로봇 미들웨어에 적용하기 위해 이 두 단점을 해결해야 하며, 기본 CAN 프로토콜의 프레임 포맷 중 Arbitration Field를 그림 3과 같이 변경하였다.

Session ID	Source ID	Destination ID
3 bit	4 bit	4 bit

그림 3. CAN Arbitration Field

본래 기본 CAN 프로토콜의 Message Identifier 필드인 Arbitration Field는 11bit (CAN 2.0a), 혹은 29bit(CAN2.0b)가 사용된다. 실제 일선에서 사용되는 대부분의 CAN 컨트롤러들은 양쪽 표준 모두를 지원하고 있기 때문에 두 가지 모두를 사용하는 것도 무방하다. 하지만 퍼스널 로봇에 장착되는 모듈의 수는 제한적일 수밖에 없기 때문에 좀 더 구성하기가 간단하고 범용성에 이점이 있는 11bit Arbitration을 사용하기로 하였다.

그런데 CAN 네트워크에서 노드는 기본적으로 수신 Identifier를 갖고 있어서 자신의 수신 Identifier와 일치하는 메시지를 받도록 되어 있다. 그렇지만 시중에 유통되고 있는 대부분의 CAN 컨트롤러는 메시지 Identifier와 상관없이 모든 메시지를 받을 수 있는 동작 모드를 갖고 있기 때문에 위와 같은 구성이 가능하게 되는 것이다.

Session ID는 각 세션의 구분 및 우선순위를 나타내고, 송신, 발신자 ID는 인터페이스 계층에서 자체적으로 제공하는 상수 ID로 각 노드를 구분할 수 있도록 한다. 그리고 메시지 식별자 (Message Identifier)라는 기본 CAN Protocol에서 제공하는 기능 중 우선순위에 대한 부분을 Session ID (Priority)로 제공하였다.

그리고 CAN 네트워크의 특성상 Message Identifier (Arbitration Field)에 따라서 우선순위가 결정되기 때문에, Session ID가 낮을수록 높은 우선순위를 갖게 되지만, 노드의 ID 역시 우선순위에 영향을 끼치기 때문에 네트워크를 구성함에 있어서 빈번하게 사용되는 모듈의 노드 ID를 낮게 배정하는 것으로 CAN 네트워크를 구성해야 한다.

또한 대용량의 데이터를 보낼 수 있게 하기 위해서 데이터 패킷을 그림 4와 같이 두 가지로 분류하였다.

Determination Packet	Data Length	Data Payload
	4 byte	4 byte

Data Packet	Data Payload
	8 byte

그림 4. CAN 데이터 패킷의 종류

기본 프로토콜에서 제공하는 8 byte의 MTU로는 서비스가 제한적일 수밖에 없기 때문에 큰 용량의 데이터를 위해 단편화 (Fragmentation) 을 지원해 줘야 한다. 이를 위해 Determination packet, Data packet 두 가지로 나누었고, 이

분할의 근거는 많은 용량의 데이터를 송신하고 수신할 수 있게 하려면, 수신 노드에서 데이터의 크기를 알아야 하기 때문이다. 즉, 먼저 Determination Packet이라 명명한 첫 번째 패킷을 보내어 수신측에서 수신할 데이터 패킷의 크기를 알 수 있게 하고 송신측은 그 크기만큼을 보내기만 하면 되는 것이다.

#### 4. 결론 및 앞으로 할 일

본 논문에서는 퍼스널 로봇 미들웨어에 적용하기 위한 CAN 프로토콜을 제안하였다. 기본 CAN 프로토콜을 완전히 변경한 것이 아니라 일부 형식만 변경했기 때문에 이 프로토콜을 사용함으로써 실제로 사용 중인 CAN 네트워크 인터페이스를 그대로 사용하면서도 퍼스널 로봇 미들웨어가 요구하는 사항을 충족시킬 수 있다.

실제 구현을 해 보았을 때, 이종 네트워크 인터페이스 간 동작이 원활히 이루어지는 것은 확인 하였으나, 동작 지연 시간이 만족스럽지 못하기에 본 논문에서는 성능 평가 부분을 논외로 하였다. 따라서 앞으로 할 일은 지연 시간을 효과적으로 개선을 해야 한다.

그리고 CAN 네트워크 자체가 Multi-Master, Multi-Slave를 표방하고 있다 보니 퍼스널 로봇 미들웨어의 동작 중에 CAN 인터페이스를 사용하는 노드의 연결이 끊어졌을 경우에 대한 대처 방안이 없다. 이는 무선랜에서 하는 것처럼 주기적으로 alive 시그널을 보내어 노드의 연결 여부를 확인하는 방법이 있을 수 있는데, 이에 대한 설계는 아직 이루어지지 않았으며 추가시킬 예정이다.

또한 현 설계나 구현은 객체 지향 중심이 아니었다. 이는 퍼스널 로봇이 지향하는 [다양한 모듈의 부착]이라는 기능을 제공할 수는 없다. 왜냐하면, 하나의 모듈 혹은 모듈 내 인터페이스가 추가될 때 마다 새로 컴파일하고 적용시켜야 하는 불편함을 내재하고 있기 때문이다. 퍼스널 로봇에 연결되는 모든 모듈을 객체로 취급하게 되면, 어플리케이션 입장에서는 모듈 자체의 인터페이스만 알고 있으면 모듈을 추가하는데 별도의 작업이 필요 없기 때문이다.

#### 참 고 문 헌

- [1] K. Taylor and B. Dalton, "Issues in Internet Tele-robotics," in *International Conference of Field and Service Robotics*, Dec. 1997.
- [2] Roberto Brega, Felix H. Wullschleger, "A Personal Robot for Mechatronic Application Programs" in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 8-11, JUL, 2001
- [3] IXXAT Automation GmbH, "CAN Application Layer", <http://www.ixxat.de>
- [4] Embedded System Korea, "Controller Area Network", <http://www.eskorea.net>
- [5] Synchworks, "TMS320VC33 DSP Controllers", <http://www.tms320.co.kr>