2007년도 대한전기학회 하계학술대회 논문집 2007. 7. 18 - 20

UML을 이용한 CANopen 프로토콜 개발에 관한 연구

박건우, 임동진
한양대학교 전기전자제어계측학과

A Study on the development of CANopen Protocol using UML

Gun-Woo Park, Dong-jin Lim
Dept. of Elec. Elec. Con. &Inst. Eng. Hanyang Univ.

Abstract - Development of software for microprocessors is one of the areas where UML can be used. There are many UML tools which is capable of generating source code for microprocessors. In this paper, a part of CANopen protocol is implemented using UML and the source code generated by a UML tool is tested.

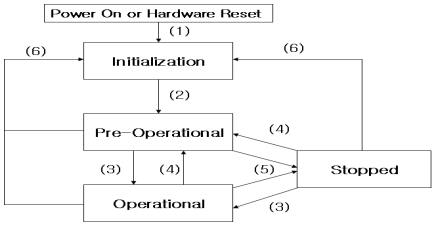
1. 서 론

UML(Unified Modeling Language)은 소프트웨어 개발 과정에서 많이 사용되는 모델링을 위한 언어로 1997년 버전 1.0을 시작으로 현재 버전 2.1까지 출시되었다. 다양한 관점을 표현하기 위해 Class Diagram, Object Diagram, Use-case Diagram, Statechart Diagram 등 9개의 다이어그램이 있다. 이렇게 많은 다이어그램을 제공하는 이유는 소프트웨어 개발 과정의 산출물들을 비주얼하게 제공하고 개발자들 간의 의사소통을 원활하게 할 수 있도록 하기 위함이다. 또한 UML 도구인 RATIONAL SOFT사의 Rational Rose 나 TELELOGIC사의 Rhapsody는 프로그램 모델을 설계하고 그 모델을 이용하여 프로그래밍 언어로 자동 코드 변환이 가능한 기능도 제공하고 있다. 본 논문에서는 Rhapsody를 이용 Object Diagram과 Statechart Diagram으로 CANopen 프로토콜의 일부를 모델링 하며, 나아가 그 모델에서 프로그래밍 언어로 자동 변환시켜 직접 시스템에 적용 가능함을 보여준다.

2. 본 론

2.1 CAN 과 CANopen

CAN(Control Area Network)는 차량제어를 위한 시리얼 통신 방식으로 1986년 BOSCH사에서 개발되었다. 1992년 처음으로 자동차(S-Series/BMW8)에 CAN 통신을 사용하였고 점점 그 적용 분야가 확대되고 있는 추세다. 11비트의 식별자와 최대 8바이트의 데이터를 전송할 수 있으며 1Mbit/s까지의 데이터 전송이 가능하고 ISO 11898로 국제표준으로 정해져 있다. ISO 11898은 OSI7계층 중 하위 2계층(물리 계층, 데이터 링크 계층)에 대해 정의한다. CANopen 은 CAN을 기반으로 하는 프로토콜로 CANopen 외에도 CAL, CAN Kingdom, DeviceNet, SDS, SAE J1939등이 있다. CANopen 디바이스는 모든 정보가 OD(Object Dictionary)에 저장되어 있으며, 이 OD에 접근하기 위해서는 SDO(Service Data Object)통신을 통해서 접근이 가능하다. 또한 실시간으로 변하는 CANopen 디바이스의 Digital I/O나 Analog I/O는 PDO(Process Data Object)통신을 통해 전송이 이루어진다. SDO와 PDO통신은 디바이스의 상태에 따라 제한이 있다. 그림1은 CANopen Slave Device의 상태 다이어그램을 나타낸 것으로 이 상태들의 전환은 Master Node의 NMT(Network Management) Service 메시지에 의해 가능하다. Pre-Operational 상태에서는 SDO 통신만 가능하나 Operational 상태에서는 SDO, PDO 통신 모두 가능하다. PDO메시지는 SDO메시지보다 우선한다.



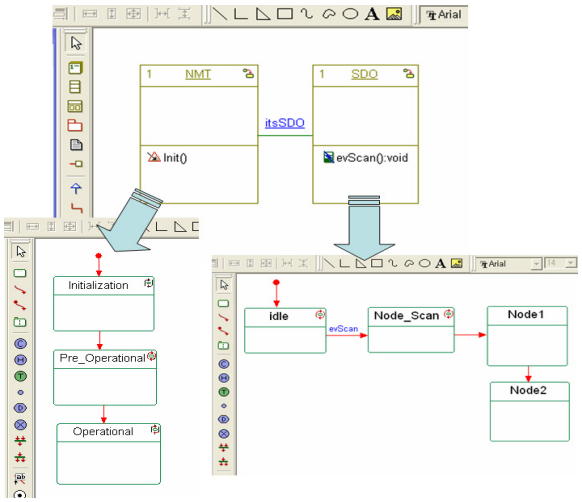
<그림 1> CANopen Slave Device의 State Diagram

2.2 소프트웨어 모델링

UML도구인 Rhapsody는 현재 7.0 까지 출시가 되었으며 TELELOGIC사의 제품으로 C언어와 C++언어를 지원한다. 또한 Rhapsody는 OS(Operating System)가 탑재되어있지 않은 8비트나 16비트 마이크로프로세서와 같은 작은 시스템에서 속도나 사이즈를 최적화한 코드를 생성하기 위해 IDF(Interrupt Driven Framework)패키지를 제공한다. 본 논문에서는 C167보드에 적합한 Tasking_c166을 사용하여 코드를 생성하게 된다. CANopen 프로토콜 구현을 위해 소프트웨어 모델링에 사용된 Diagram은 그림2와 같이 Object Diagram과 Statechart Diagram이다. Object Diagram은 클래스의 인스턴스(Instance)인 객체들을 표현하는 것으로 일반적인 대상을 실체화한 고유한 사물이나 개념으로 묘사되어진다. Statechart Diagram은 객체와 클래스에 대한 모형화에 이용되는데, 이는 행위가 유발될 때 각각의 객체가 상태를 어떻게 변화시키는지 기술한다. 또한 추상화(Abstraction)를 사용하여 복잡한 시스템을 쉽게 표현할 수 있으며 애니메이션 기능을 통하여 화면상에 상태의 변화에 따른 흐름을 보여주는 유용한 도구이다. 표1은 CANopen 프로토콜의 주요 Service들과 UML로 구현된 Service들의 목록이다. 그림2에서는 시스템을 두 부분으로 나누어 NMT Service 부분과 SDO 통신 부분을 Object Diagram으로 나타내었으며 세부 동작 상태를 Statechart Diagram을 이용하여 표현하였다. 세부동작 상태에서는 모든 노드들의 동작, 비동작을 관장하는 NMT Service메시지가 모든 노드들을 Operational 상태로 전환시킨 후 evScan 이벤트를 발생시켜 SDO 통신을 수행하도록 한다. evScan 이벤트가 발생하면 Master Node는 네트워크에 있는 모든 노드를 검색하며, 검색된 각 노드들의 동작 여부의 에러 체크를 하게 된다.

Service 종류	Mode	구현여부
NMT	Reset	O
	Pre-Operational	O
	Operational	O
	Stop	X
SDO	Read (Upload)	O
	Write (Download)	X
PDO1	Read (Upload)	X
	Write (Upload)	O
PDO2	Read (Upload)	X
	Write (Upload)	O
Error Control	Node Guarding	X
	Heartbeat	X

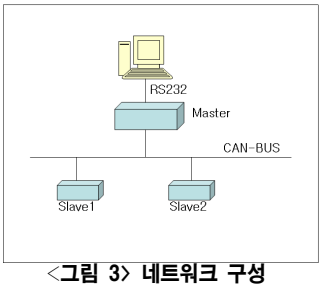
<표 1> CANopen 프로토콜의 Service들



<그림 2> Object Diagram과 Statechart Diagram

2.3 시스템 구성

실험에 사용된 시스템은 CAN 통신을 지원하는 노드들을 CAN-BUS로 네트워크를 구성하고 CANopen Protocol을 구현하도록 설계하였다. 그림 3와 같이 CAN 통신을 지원하는 SIEMENS사의 C167 보드를 Master Node로 설정하고 PHYTEK ELECTRONIC사의 MN-214-CA05V2 Slave Device[3] 모듈 2개를 이용하여 CAN-BUS로 네트워크를 구성하였다. 각각의 Slave Device는 모두 CiA Draft Standard 301[1] 버전3.0과 CiA Draft Standard 401[2]을 내장하고 있다. 모든 노드들의 전송속도는 1Mbit/s로 설정을 하였고 Rhapsody로 작성된 프로그램은 RS232 통신을 통해 Master Node로 다운로드할 수 있도록 하였다. 그림4는 Master Node로 사용된 C167 보드와 MN-214-CA05V2 Slave Device의 사진이다.



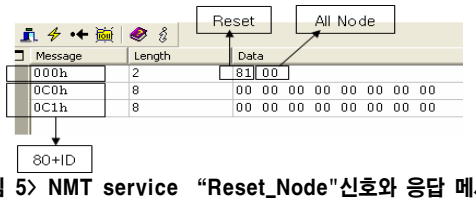
<그림 3> 네트워크 구성

<그림 4> C167(상)과 MN-214-CA05V2(하)

2.4. 실 험

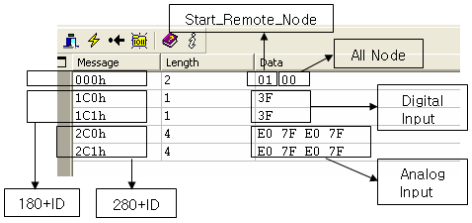
2.4.1 NMT Service 통신 실험

그림5는 Master Node 로부터 모든 노드들에게 Reset 메시지를 보내는 데이터와 Slave Node들의 응답 메시지이다. 특정 노드를 Reset 시키려면 All Node 바이트에 노드 ID를 입력하면 된다. Reset된 노드들의 응답 메시지의 형식은 80h+Device ID 와 8바이트의 00h이다.



<그림 5> NMT service "Reset_Node"신호와 응답 메시지

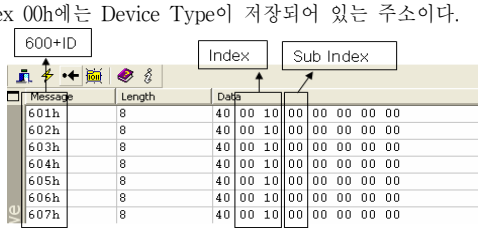
그림6은 Master Node로부터 보내는 "Start Remote_Node" 신호와 Slave Node들의 응답 메시지이다. 특정 노드를 Operational Mode로 만들려면 All Node 바이트에 노드ID를 입력한다. "Start_Remote_Node"신호를 받은 Slave Node는 Digital Input 과 Analog Input값을 반환한다.



<그림 6> "Start Remote_Node" 신호와 응답 메시지

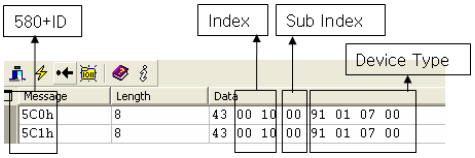
2.4.2 SDO(Service Data Object) 통신 실험

Slave Node는 최대 110개까지 지원된다. 그림7은 Master Node가 모든 Slave Node들을 스캔하기위해 네트워크에 보내는 SDO 메시지이다. 각각의 Slave Node 들은 고유한 Device ID를 가지고 있으므로 600h+Device ID, Object Dictionary의 Index 1000h 번지, Sub Index 00h에 접근하는 SDO 메시지를 순차적으로 보냄으로써 존재하는 모든 Slave Node를 찾아낼 수 있다. Object Dictionary의 Index 1000h번지, Sub Index 00h에는 Device Type이 저장되어 있는 주소이다.



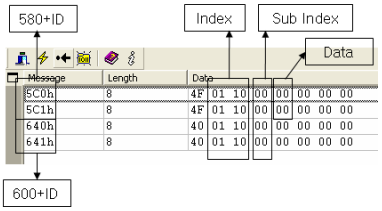
<그림 7> 노드들을 스캔하기 위한 SDO 메시지

그림8은 그림7에서 Master Node가 송신한 SDO 데이터에 대한 Slave Node들의 응답 메시지이다. 해당되는 Slave Node가 존재하면 응답 메시지를 반환하며 존재하지 않으면 반환되는 데이터는 없다. 반환시의 메시지 형태는 580h+Device ID와 Object Dictionary의 Index 1000h 번지, Sub Index 00h , 그리고 Device Type에 대한 데이터이다. MN-214-CA05V2 Slave Node 는 각각 Node ID 가 40h, 41h 이고 Device Type은 32비트 값으로 모두 00070191h이라는 것을 알 수 있다. 이 Device Type의 정보는 Device Number 는 191h 이고 Digtal Input(16th Bit = 1), Digital Output(17th Bit = 1), Analogue Input(18th Bit = 1)의 기능을 포함하고 있으며 Analogue Output(19th Bit = 0)기능은 포함하지 않음을 보여준다.



<그림 8> 스캔 과정에서 Slave Node의 SDO 응답 메시지

그림9는 Slave Node들의 에러체크를 하기위해 Master Node가 보내는 SDO 메시지와 Slave Node의 응답 메시지이다. Object Dictionary의 Index 1001h의 Sub Index 00h에는 Error Resgister를 저장하고 있으며 값이 00h이면 Device는 정상 동작중이며 00h가 아니면 에러가 존재함을 나타낸다. 그림6의 Slave Node 들의 Error Register의 값은 00h이므로 정상 동작 상태임을 알 수 있다.



<그림 9> 에러 체크 SDO 메시지와 Slave Node들의 응답 메시지

3. 결론 및 추후 과제

본 논문에서는 UML도구인 Rhapsody를 이용하여 Object Diagram과 Statechart Diagram으로 CANopen 프로토콜의 NMT, SDO, PDO 서비스의 일부를 모델링하고 프로그래밍 언어로 변환과정을 거친 후 시스템에 적용하였다. Device간에 전송되는 모든 데이터(그림5-그림9)들은 SYS-TEC ELECTRONIC 사의 USB-CAN module을 통하여 캡처를 하였으며 PEAK SYSTEM TECH사에서 제공하는 프리 소프트웨어인 CANopen Magic 프로그램의 실행 후 데이터와도 일치하였다. UML의 강력한 기능인 비주얼적인 측면을 보여주는 Sequence Diagram이나 Statechart Diagram의 애니메이션 기능은 작은 시스템에서는 적용할 수 없는 한계가 있었다. 추후 CANopen 프로토콜 중 본 논문에서 구현이 누락된 서비스들을 추가하여야 하며 데이터의 신뢰성을 얻기 위해 더 많은 노드들의 테스트가 요구되며 센서나 엑추에이터의 추가로 실시가 변화하는 데이터들의 검출이 필요할 것이다.

이 논문은 경기도에서 지원하는 경기도지역협력연구센터 사업의 지원을 받아 연구되었음.

[참 고 문 헌]

[1] "CANopen: a CAL based Communication profile for indus trial systems" Draft Standard DS-301 Revision 3.0, CAN in Automation Group, October 1996

[2] "CANopen device profile for digital I/O modules", Draft Standard DS-401 Revision 1.0 CAN in Automation Group, September 1996

[3] MN-214-CA05V2 CANopen Chip System Manual, March 2001

[4] K.Etschberger "Controller Area Network" July. 2001

[5] M.Farsi, K. Ratcliff and Manuel Barbosa "An introduction to CANopen" pp.161-168 Aug. 1999

[6] Jason T.Roff "쉽게 배우는 UML과 객체지향 설계" pp.223-245 July. 2003