UML을 이용한 유비쿼터스 센서 네트워크용 임베디드 시스템 소프트웨어 개발에 관한 연구

최종원, 임동진
전자전기제어계측 공학과 한양대학교

A Study on the Development of Embedded System Software for Ubiquitous Sensor Network

JongWon Choi, DongJin Lim
Dept. of Electrical Engineering & Computer Science, Hanyang University, Korea
IEEE

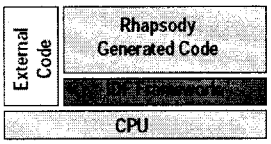
Abstract - UML(Unified Modeling Language) is the most frequently used modeling language in the process of analysis, design, implementation and etc. The main reason of using UML is not only to help users to work visually but also to draw better communication among developers. In addition, UML is one of the global standards and supports MDA. In this paper embedded software development method for USN using UML is discussed To show how the development method is used, IEEE 802.15.4 radio module are programmed using UML software tool.

1. 서 론

최근 홈 네트워크 및 유비쿼터스 네트워크에 대한 요구가 증대하면서, 가정 내의 각종 가전 기기들을 배선의 불편함 없이 연결시켜줄 수 있는 무선 네트워크 기술이 주목을 받고 있다.
이중 무선네트워크기술 중 하나인 Zigbee는 저전력 소모, 시스템구성의 저비용, 많은 네트워크 노드의 수용과 간단한 프로토콜 구조의 특징 때문 홈 오토메이션에 적합하여 최근에 각광을 받고 있고, 홈오토메이션 뿐만 아니라 센서 네트워크, 공장자동화 등에도 적용할 수 있어 Zigbee를 이용한 다양한 무선 네트워크 기능을 가진 Embedded system 들이 개발이 되고 있다.
하지만 일반적인 제품 개발과정에서 보여주듯, Zigbee를 이용한 Embedded system의 개발에서도 다양한 요구조건을 충족시키기 위해 개발과정이 보다 복잡해지고 거대해질 경우, 다양한 문제점들이 드러날 수가 있는데, 개발과정에 참여한 사람들 간의 불명확한 의사소통, 불량품의 증가, 신뢰성의 저하 등이 이러한 것들이다.
이러한 문제점들 때문에 최근에는 모델 기반 개발(model-based development) 기법이 도입 되고 있다. 이는 통상 문서화된 자료들로 서로의 의사를 확인했던 것에 반에 그래픽한 객체들을 이용하여 개발자들의 의사소통을 원활하게 한다는 것이다. 시뮬레이션을 통해 사전오류 검출도 가능하여 신뢰도를 높여 제품의 정확도 또한 높일 수가 있다.
본 논문에서 IEEE 802.15.4 기반의 무선 통신 모듈을 통해 데이터의 전송과정을 UML로 모델링하고 이를 통한 코드를 생성하여 실제 하드웨어 모듈에 탑재하려 한다. 그리하여 보다 쉽고 명확한 방법으로 무선 센서 네트워킹을 설계, 구현하고자 한다.

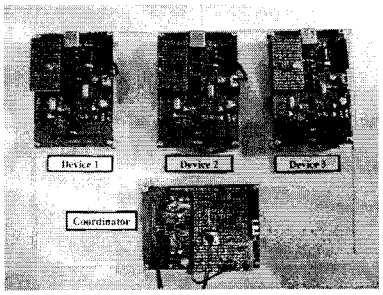
2. 시스템의 구성 및 구현결과

IEEE 802.15.4 모듈에 탑재될 실행 코드들은 지금까지 C 기반의 컴파일러를 통해 제작되었다. 그러나 C 기반의 방법은 개발자가 미처 알지 못하는 구조적 문제를 유발하기 쉬우며 또한 이에 대한 검증은 실제 하드웨어 탑재 및 기능 점검을 하기 전까지는 알지 못하는 경우도 많다. 본 실험은 UML 도구중 하나인 Rhapsody를 이용하여 프로그램을 직관적으로 작성하였다.
Rhapsody는 자체 컴파일러를 가지고 있지 않기 때문에 해당 플랫폼의 MCU에 맞는 코드를 생성하기 위해선 IDF라는 Framework이 필요하다. <그림 1>과 같이 IDF는 Rhapsody에서 만든 모델들을 해당 MCU에 맞게 코드변환 과정을 수행하며, 이로 인해 작성된 객체들은 하드웨어에 독립적인 구조를 가지고 있기 때문에 플랫폼의 MPU가 어떤 것이냐에 따라 코드 변환이 가능하다는 특징을 가지고 있다. 실험에 사용되는 IEEE 802.15.4모듈은 ATMEGA28 칩을 쓰고 있어, IDF가 IAR 컴파일러나 AVR 스튜디오의 컴파일과정을 대신 수행하게 된다. Rhapsody를 통해 최종적으로 만들어진 파일은 IEEE 802.15.4모듈에 탑재되어, 외부 인터럽트가 인가되었을 경우 Coordinator 역할을 하는 모듈의 Channel이 변경되어 Coordinator의 Channel와 같은 모듈의 LED를 제어 할 수 있게 하였다.



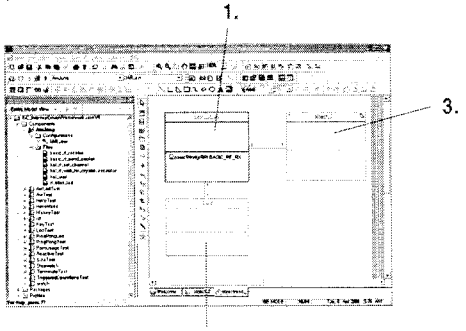
<그림 1> IDF와 Rhapsody의 관계

실험에서는 OLmicrowaves사의 IEEE 802.15.4 모듈들을 <그림 2>과 같이 하나의 Coordinator와 3개의 Device들로 구성을 하였다. IEEE 802.15.4 모듈은 128kbyte의 프로그램 메모리와 RAM을 갖는 저전력 ATMEGA128L을 통해 제어되며 통신은 Chipcon 사의 CC2420을 통해 2.4GHz 대역의 통신망을 사용한다. ATMEGA128에는 2개의 12비트 분해능을 갖는 ADC를 가지고 있으며 이를 통해 해당 LED의 밝기를 조절할 수 있다. Device들 간의 LED는 Coordinator의 Potentiometer로 밝기 조절을 하게 된다.



<그림 2> IEEE 802.15.4모듈의 구성

본 실험은 무선 통신을 하기위해 필요로 하는 많은 코드들 중 Coordinator 모듈에 탑재할 코드를 Rhapsody로 표현하였다. 표현된 코드는 IEEE 802.15.4에 규약 된 MAC layer의 MCPS-DATA.request와 무선통신을 위해 CC2420칩을 초기화하는 Class 그리고 시스템 전체를 작동시키기 위한 main loop이며, <그림 3>의 사각형들은 Rhapsody의 Object Diagram에 표현된 Object들로 Class의 instance를 의미한다.

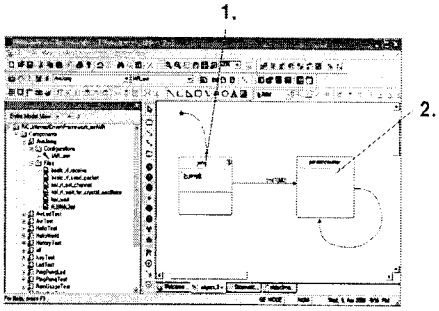


<그림 3> 비주얼한 객체로 표현된 코드

<그림 3>의 1은 basic_rf_init Object 이다. 무선 통신을 하기 위해 CC2420을 초기화하는 과정을 담당하며, RF Channel 설정, PanID, myAddr 등의 protocol configuration도 이곳에서 수행하게 된다.

<그림 3>의 2는 basic Object이며, IEEE 802.15.4에 규약 된 MAC layer의 MCPS-DATA.request중 PanID, myAddr가 이 Object 내에 정의되어있다.

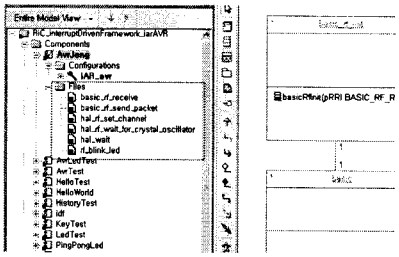
<그림 3>의 3는 main Object로 main loop의 전반적인 수행과정과 Potentiometer의 변화를 감지한다. Object내의 상태가 변화되었을 경우 이를 비주얼하게 Statechart로 모형화 할 수 있는데, 본 실험에서는 RF Channel, PanID, myAddr 의 값 설정과 Potentiometer의 변화를 Statechart로 표현하였으며 외부인터럽트의 경우는 C코드 형식으로 Files에 추가하였다.



<그림 4> main Object 내의 Statechart

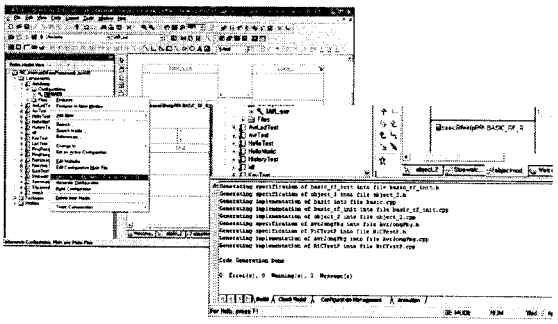
<그림 4>의 1은 main loop 내의 jong함수이다. RF operation을 하기위해 basic_rf_init Object의 Channel, PanID, myAddr의 값을 설정해 주는 역할을 하며, 2는 Potentiometer함수로 jong함수가 수행되고 난후 1초 후에 무한 루프로 빠져 Potentiometer의 변화를 지속적으로 감지하게 된다.
즉, <그림 3>의 2인 main Object가 실행이 되면 main loop내의 jong함수가 실행이 되어 Channel, PanID, myAddr의 값들이 <그림 3>의 1인 basic_rf_init Object에 셋팅이 되고, 무선통신을 하기위한 셋팅과정이 끝나면 Potentiometer함수에서 지속적으로 Potentiometer의 변화를 감지한다.
basic_rf_init Object 내의 PanID와 myAdd의 정의는 <그림 3>의 3인 basic Objec에 되어 있고, 해당 파라미터에 대한 번지수는 CC2420 내에 설정되어 있다.

Object Diagram에 표현하지 못한 나머지 코드들을 <그림 5>와 같이 C언어형식으로 작성하여 Files에 첨가 하였다.



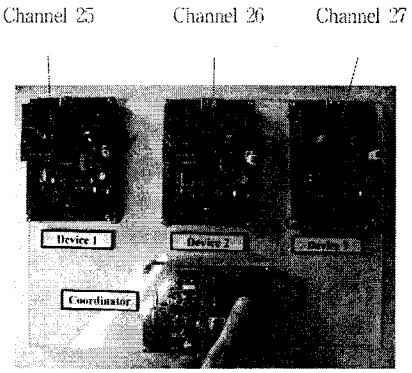
<그림 5> Files폴더에 추가된 코드들

완성된 Object들은 <그림 6>와 같이 IDF를 이용하여 IEEE 802.15.4 모듈에서 실행가능 하도록 Generate을 하며, Generate과정을 통해 생성된 Hex 파일은 IAR embedded workbench 로 IEEE 802.15.4 모듈에 다운로딩 한다.



<그림 6> IDF를 통해 모듈에 실행 가능한 파일 생성

<그림 7>은 Coordinator의 외부 인터럽트를 인가되었을 때, 해당 모듈의 LED가 제어되는 과정을 보여준다. 인터럽트를 가했을 경우 기존의 Channel이 25-> 26-> 27로 변경되어 Coordinator와 같은 Channel을 가진 Device의 LED가 제어되는 것을 확인 할 수 있다.



<그림 7> 외부인터럽트를 통한 Channel 변경

이 실험을 통해 비주얼한 모델을 이용하여 변환된 코드가 기존의 C코드와 같은 수행을 하는 것을 확인할 수가 있었다. Rhapsody의 이런 그래픽한 효과 때문에 개발과정이 보다 쉽게 이행 될 수 있으며, 실험에선 시행을 하지 못하였지만 시뮬레이션 기능도 있어 PC상에서 실제와 같은 동작을 통해, 제품을 만들어 보지 않고도 사전 오류검출을 해 낼 수가 있다.

3. 결 론

본 논문에서는 UML 도구인 Rhapsody를 이용하여 기존의 C로 작성되었던 코드들을 시스템 모델링해서 변환 과정을 거친 후, 모듈에 적용 하였다. Coordinator 역할을 하는 모듈의 외부인터럽트를 통해 Channel을 바꿔가며, Potentiometer로 Coordinator의 Channel과 일치하는 Device의 LED를 제어하는 과정을 수행하였다. 실험을 통해 모델화한 Object들을 통해 변환된 코드와 기존의 C코드가 같은 수행을 하는 것을 확인할 수가 있었으며, 시각적인 효과로 코드프로그래밍의 진행과정을 보다 쉽게 이해 할 수 가 있었다. 본 실험에선 C언어로 구성된 코드 중 일부 함수만을 Rhapsody로 구현하였기 때문에, 추후에 나머지 코드를 전부 변환하는 과정이 진행 되어야 할 것이며, Rhapsody의 장점 중 하나 인 시뮬레이션 과정을 Statechart나 Sequence diagram을 이용하여 보다 구체적 표현하여 PC상에서도 오류검출하는 과정이 가능하도록 한 구현이 필요할 것이다.

이 논문은 경기도에서 지원하는 경기도지역협력 연구센터 사업의 지원을 받아 연구되었음.

[참 고 문 헌]

[1] "ZigBee Specification Document", December 2006
[2] 이원준,이춘화 ,"저속 WPAN IEEE 802.15.4 센서 네트워크", June 2005
[3] Luciano Lavagno, Grant Martin, Bran Selic "UML for Real", January 2003
[4] Cris Kobryn, Grady Booch, Ivar Jacobson, and Jim Rumbaugh "UML DISTILLED", October 2005
[5] Jason T.Roff, "쉽게 배우는 UML과 객체지향 설계", July, 2003
[6] 박건우,"UML을 이용한 CANopen 프로토콜 개발에 관한 연구", February, 2008
[7] 이용우, 고영국, 박태희, 김준수 "UML과 객체지향 시스템 분석 설계", October 2002