TOSSIM 시뮬레이터에서 센서 값을 설정하는 TinyViz 플러그인 설계

*김성훈°, *박양수, **이명준
울산대학교 컴퓨터정보통신공학부
*{heinz°, yspark}@mail.ulsan.ac.kr
**mjlee@ulsan.ac.kr

Design of a TinyViz Plugin setting values for the TOSSIM simulator

Seonghune Kim°, Yangsoo Park, Myungjoon Lee
School of Computer Engineering & Information Technology, University of Ulsan

요 약

USN은 각종 센서를 이용하여 무선으로 정보를 수집할 수 있도록 구성한 네트워크를 말한다. 센서네트워크 노드를 위하여 설계된 운영체제인 TinyOS에서 제공하는 TOSSIM 시뮬레이터를 이용하면 응용 프로그램을 개발하면서 시뮬레이션을 수행하고, 데이터를 분석하여 문제점을 미리 발견하고 이를 보완할 수 있다. TOSSIM에서는 그래픽 유저 인터페이스를 제공하는 TinyViz와 TOSSIM 스크립트 언어인 Tython을 이용하여 다양한 시뮬레이션이 가능하다.

본 논문에서는 능동적인 시뮬레이션 모델을 제공하기 위하여 Tython 스크립트 언어를 이용하여 TinyViz 플러그인으로서 PeriodicADCPlugin을 설계하였다. PeriodicADCPlugin은 그래픽 유저 인터페이스 상에서 모트들에게 다양한 센서 값을 여러 가지 방법으로 간편하게 입력하는 기능을 제공한다.

1. 서 론

USN(Ubiquitous Sensor Network)은 각종 센서를 이용하여 무선으로 정보를 수집할 수 있도록 구성한 네트워크를 말한다. 필요한 모든 사물에 센서네트워크 노드를 설치하고, 이를 통하여 사물의 인식정보를 기본으로 온도, 습도, 조도 등 주변의 환경정보를 탐지하고, 이 정보를 실시간으로 네트워크를 통하여 정보를 관리할 수 있다. 사람의 접근이 불가능한 취약지구에 수백개의 센서네트워크 노드를 설치하여 사람이 감시하는 것과 마찬가지의 역할을 한다. 매우 작은 크기의 독립된 무선 센서들을 건물, 도로, 의복, 인체 등 물리적 공간에 배치하여 주위의 온도, 빛, 가속도, 자기장 등의 정보를 무선으로 감지, 관리할 수 있다.

TinyOS[1, 2]는 USN을 구축하기 위한 무선 센서네트워크를 위하여 설계된 이벤트기반의 운영체제이며 재사용 가능한 컴포넌트 기반의 운영체제이고 상태 머신 기반의 구조를 가지고 있다. TinyOS는 센서 노드들의 저전력 소모를 위해서 재빨리 작업을 처리하고 휴지 상태로 들어가게 됨으로써 불필요한 전력 소모를 줄인다.

TOSSIM[3, 4]은 TinyOS에서 제공되는 센서네트워크 시뮬레이터이다. TOSSIM을 이용하면 실제 센서네트워크 환경을 구축하기 이전에 센서 모트들을 시뮬레이션 할 수 있으며, 시뮬레이터한 데이터를 분석하여 문제점을 미리 발견하고, 향후 어떤 방향으로 개선되고 개발되어야 하는지에 대하여 프로그램의 동작을 확인하고 응용할 수 있다. TinyViz는 정적이며 단순한 TOSSIM 시뮬레이터를 보완하여 그래픽 유저 인터페이스를 제공하고, 다양한 플러그인들과 TOSSIM 스크립트 언어인 Tython을 이용하여 시뮬레이션의 시각화와 디버깅등을 제공한다.

본 논문에서는 TinyViz 플러그인의 구조와 ADCPlugin을 분석하고, Tython[5] 스크립트 언어를 이용하여 그래픽 유저 인터페이스 상에서 복잡한 시뮬레이션을 보다 손쉽게 하기 위하여 TinyViz 플러그인 PeriodicADCPlugin의 개발을 설계하였다. 센서네트워크 환경을 구축할 때, PeriodicADCPlugin을 이용하여 시뮬레이션을 수행하면, 사용자는 그래픽 유저 인터페이스 상에서 조도나 온도 정보와 같은 환경정보를 손쉽게 설정할 수 있으며, 또한 실세계의 변화하는 다양한 센서 정보들을 스크립트를 이용하여 실제 모트를 테스트 하는 것과 동일한 시뮬레이션을 수행할 수 있다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 관련 연구로써 무선 센서 네트워크를 위하여 설계된 TinyOS, TOSSIM 시뮬레이터, TinViz와 Tython에 대하여 살펴보고, 3장에서는 TinyViz 플러그인의 구조와 PeriodicADCPlugin을 위한 ADCPlugin을 분석하고, 4장에서는 PeriodicADCPlugin의 설계를 살펴보도록 한다. 그리고 마지막 5장에서는 결론 및 향후여구 과제에 대하여 살펴본다.

2. 관련 연구

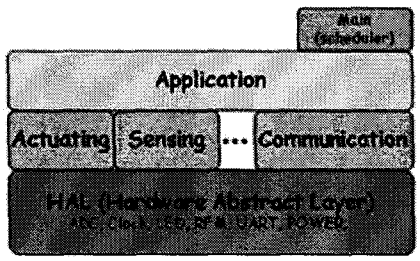
센서네트워크 운영체제인 TinyOS는 센서 네트워크의 응용 개발을 지원하며, 제한된 센서 노드 자원을 관리하기 위하여 설계된 컴포넌트 기반, 이벤트 구동방식의 운영체제이다. TinyOS는 무선 센서노드의 일반적인 특징인

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성지원사업의 연구 결과로 수행되었음.

최소한의 하드웨어, 작은 메모리, 낮은 CPU 성능 그리고 한정된 에너지를 고려하여 최적화된 운영체제 환경을 제공하고 있다.

2.1 TinyOS

TinyOS는 미국 버클리대학에서 개발된 무선 센서네트워크를 위한 오픈-소스 운영체제이다. TinyOS는 센서네트워크에서 요구되는 제한된 메모리에 맞게 code 크기를 최소화하여 구현되어 있으며 제한된 자원을 가진 작은 크기의 센서 노드에서 효율적인 자원의 사용과 프로세싱 동시성을 지원해 준다. TinyOS는 재사용 가능한 소프트웨어 컴포넌트 기반의 운영체제이며, 응용 프로그램은 하드웨어 컴포넌트의 입/출력을 연결하듯 소프트웨어 컴포넌트의 입/출력 인터페이스를 연결함으로써 작동된다. 그리고 TinyOS는 상태 머신 기반의 구조를 가지는 운영체제로서 전체 시스템은 여러 상태머신들로 구성되어 있으며, 각각의 컴포넌트는 해당 상태를 나타내고 있다. 제한된 자원만을 가진 노드들로 구성된 센서네트워크의 가장 큰 문제점인 저전력 소모를 위해서 TinyOS는 CPU가 사용되지 않을 동안 휴지 상태로 들어가게 함으로써 불필요한 전력 소모를 줄인다. (그림 1)은 TinyOS의 구조를 보여주고 있다. Mote에 다운로드 되는 하나의 애플리케이션에는 TinyOS 커널이 그 애플리케이션의 스케줄러 역할과 하드웨어 초기화 역할을 수행하게 된다.



(그림 1) TinyOS의 구조

2.2 TOSSIM

본 절에서는 TOSSIM의 특징에 대하여 살펴보고, 그래픽 유저 인터페이스 환경을 지원하여 주는 TinyViz와, 스크립트를 이용하여 TOSSIM 시뮬레이션을 할 수 있는 Tython 스크립트 언어에 대하여 살펴본다.

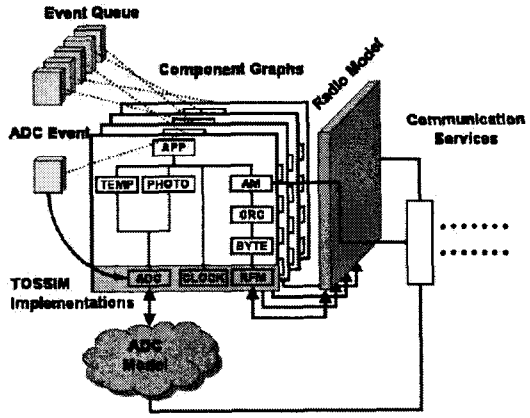
2.2.1 TOSSIM 시뮬레이터

TOSSIM은 무선 센서네트워크를 구성하기 위해 개발되어진 TinyOS를 각각의 센서에 심어 센서의 동작을 확인하기 위한 시뮬레이터이다. TOSSIM을 이용하여 시뮬레이터한 데이터를 분석하고 Packet Loss Rate, Packet CRC failure rate, Scalability 등이 어떻게 변하는지 분석한 후 향후 프로토콜이 어떤 방향으로 개선되고 개발되어야 하는지, 시뮬레이터에 추가되어야 할 사항은 어떤 것들이 있는지에 대하여 알아나가면서 프로그램의 동작을 확인하고 응용할 수 있다. (표 1)은 TOSSIM을 이용하여 시뮬레이터를 할 때 요구사항을 간략하게 정리하고 있다.

Scalability	시뮬레이터에는 많은 수의 센서노드들로 이루어진 대규모 네트워크의 다양한 구성을 지원할 수 있어야 한다.
Completeness	시뮬레이터는 가능한 한 정확하게 시스템 상호작용들을 반영할 수 있어야 한다. 네트워크 프로토콜의 검증 및 응용 프로그램 실행에 따른 시스템 동작의 검증도 가능해야 한다.
Fidelity	시뮬레이터는 센서노드 내부 및 노드들간의 세부 동작까지 반영할 수 있어야 한다. 그리하여, 개발자가 예상하지 못한 상호작용도 검출할 수 있어야 한다.
Bridging	시뮬레이터는 개발자가 실제 하드웨어에서 동작할 코드를 테스트하고 검증할 수 있게 해줌으로써, 알고리즘과 구현 사이의 갭을 줄이는 데 도움을 줄 수 있어야 한다.

(표 1) TOSSIM 시뮬레이터의 요구 사항

(그림 2)에서 보는 바와 같이 TOSSIM의 구조는 Frame, Events, Models, Components, Services로 구성되어 있다.



(그림 4) TOSSIM의 구조

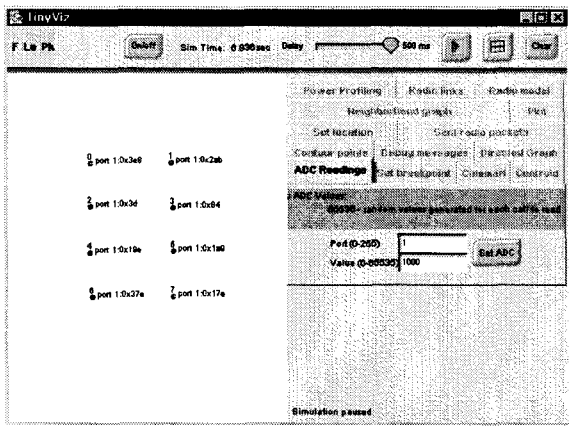
2.2.2 TinyViz

TinyViz는 TinyOS 시뮬레이터인 TOSSIM의 디버깅, 시각화등 상호작용을 위하여 신장성 있는 그래픽 유저 인터페이스 환경을 제공한다. TinyViz에 제공되는 여러 가지 플러그인들을 이용하면 시뮬레이션에서 디버그 메시지, 무선 메시지 등과 같은 이벤트를 관찰할 수 있으며, 시뮬레이션 파라미터 설정과 시험되는 모트가 읽을 센서 값 설정 등과 같은 시뮬레이션 작동을 화면에 표현할 수 있다. TinyViz 플러그인들은 다음과 같다.

• Debug messages - 시뮬레이션에 의해 발생한 모든 디버그 메시지를 윈도우에 보여준다.

• Set breakpoint - 어떤 조건을 만날 때 시뮬레이션이 멈추도록 할 수 있다.
• ADC readings - 각 모트 다음에 각각의 ADC 채널의 가장 최근의 값을 보여준다.
• Sent radio packets - Debug message plugin 처럼 윈도우에 보낸 모든 무선 패킷을 보여준다.
• Radio links - 그림으로 무선 메시지 활동을 화면에 보여준다. 모트가 메시지를 브로드캐스트할 때 파란 원이 그 주변에 그려진다.
• Set location - 각 모트의 가상 위치를 만든다.
• Radio model - 무선 접속가능성의 다양한 모델과 그들의 위치에 따라 두 mote 사이의 비트 에러 비율을 정한다.

(그림 3)은 TOSSIM 시뮬레이션에서 TinyViz의 동작 화면을 보여주고 있다. TivViz의 상단은 플러그인을 선택할수 있는 메뉴와 시뮬레이션의 시간을 조절하는 바, 그리고 시뮬레이션의 정지와 재시작을 하는 버튼등이 구성되어져 있다. TinyViz의 왼쪽창은 시뮬레이션 모트들이 동작하고 있으며, 오른쪽 창은 동작하고 있는 플러그인들들의 탭과 옵션을 설정할 수 있도록 그래픽 유저 인터페이스가 구성되어 있다.



(그림 5) TinyViz의 동작 화면

2.2.3 Tython

TOSSIM을 이용하여 알고리즘과 통제된 애플리케이션과 제시할 수 있는 환경에서 시뮬레이션 가능하지만 정적이며 단순한 모델만을 제시할 수 있기 때문에 복잡한 실세계를 모델 하기엔 힘이 든다. 해결책으로 그래픽 유저 인터페이스를 지원하며 모트를 이동가능 하도록 하고 여러 가지 옵션을 셋팅 할 수 있는 TinyViz가 만들어졌다. 하지만 TinyViz는 이미 생성된 플러그인을 사용하여 제한된 시뮬레이션을 지원하고 있다. Tython 스크립트 언어를 이용하여 시뮬레이션 환경과 모트들을 제어 할 수 있으며 TinyViz와 상호작용을 한다. Tython은 TinyOS의 TOSSIM 시뮬레이터의 내용을 비주얼하게 보여주는 TinyViz에서 사용되는 스크립트 언어이다. Tython은 스크립트 언어인 Python과 Java를 합쳐놓은 언어이기 때문에 Python 클래스와 Java 클래스를 모두 사용할 수 있다.

(표 2)는 몇가지 기본적인 Tython 명령어를 요약해 놓은것이다. 스크립트 실행쉘에서 명령어를 이용하여 모트들을 제어할 수 있으며 환경 모델들을 설정할 수 있다.

Name	Effect
sim.pause()	Pause TOSSIM
sim.resume()	Resume TOSSIM
motes[i].turnOn()	Turn mote i on
motes[i].turnOff()	Turn mote i off
motes[i].moveTo(x, y)	Move mote i to *x,y*
motes[i].move(x,y)	Move mote i *x,y* from its current position
comm.setSimRate(rate)	Set the simulator rate to *rate* (This is identical to TOSSIM's -I option)
radio.setLossRate(senderID, receiverID, prob)	Set the radio loss rate between two motes
comm.sendRadioMessage(mote, time, message)	Deliver message to mote over Radio
comm.sendUARTMessage(mote, time, message)	Deliver message to mote over UART

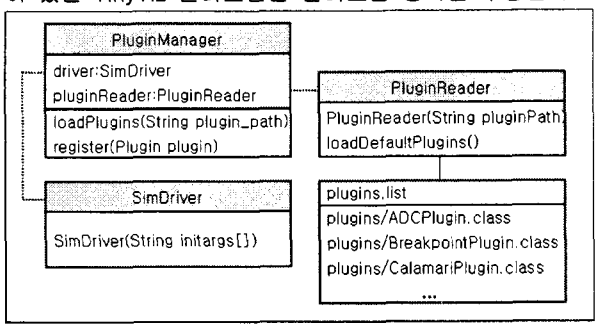
(표 2) Tython 명령어

3. TinyViz 플러그인 분석

본절에서는 TinyViz 플러그인 PeriodicADCPlugin을 설계하기 위하여 기본적인 TinyViz 플러그인의 구조와 선택된 모트에 센서 값을 설정할 수 있는 ADCPlugin을 살펴본다.

3.1 TinyViz의 플러그인 구조

TinyViz는 TOSSIM 시뮬레이터를 이용하여 시뮬레이션을 할 때 그래픽 유저 인터페이스 환경을 지원하여 준다. (그림 4)는 TinyViz의 플러그인을 불러오는 클래스들의 클래스 다이어그램을 간략하게 보여주고 있다. PluginManager는 시뮬레이션을 위하여 SimDriver를 로드하고, PluginReader을 이용하여 plugins.list에 등록 되어 있는 TinyViz 플러그인을 불러오는 동작을 수행한다.

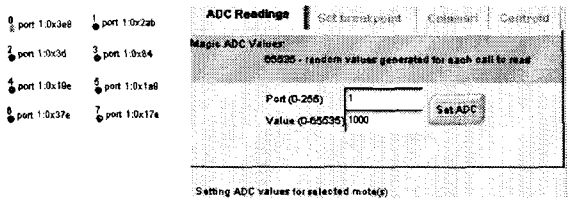


(그림 6) PluginManager의 클래스 다이어그램

3.2 ADCPlugin의 분석

ADCPlugin은 TinyViz 그래픽 유저 인터페이스 환경에

서 시뮬레이션을 위하여 모트에 값을 설정할 수 있는 기능을 제공하고 있다. (그림 5)에서 보여주는 바와 같이 0번 모트를 선택하고 플러그인 화면에서 포트번호와 센서 값을 설정하면 0번 모트에 입력된 센서 값이 설정되게 된다.



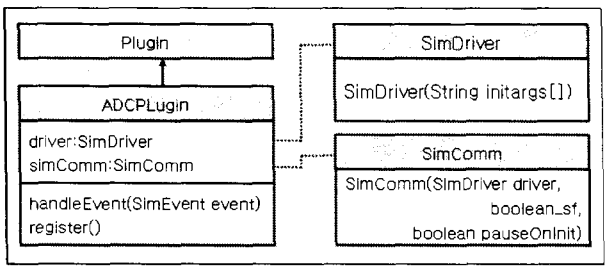
(그림 7)ADCPlugin을 이용한 센서 값 설정

(그림 6)은 ADCPlugin에서 사용하는 Port 번호에 입력되는 값을 보여주고 있다. ADCPlugin의 Port 입력 텍스트박스에 1이 입력되면 조도 값으로 설정이 되고, 2가 입력되면 온도 값으로 설정이 된다.

```
TOS_ADC_PHOTO_PORT = 1,
TOS_ADC_TEMP_PORT = 2,
TOS_ADC_MIC_PORT = 3,
TOS_ADC_ACCEL_X_PORT = 4,
TOS_ADC_ACCEL_Y_PORT = 5,
TOS_ADC_MAG_X_PORT = 6,
TOS_ADC_VOLTAGE_PORT = 7
TOS_ADC_MAG_Y_PORT = 8,
```

(그림 8) ADCPlugin의 Port 값

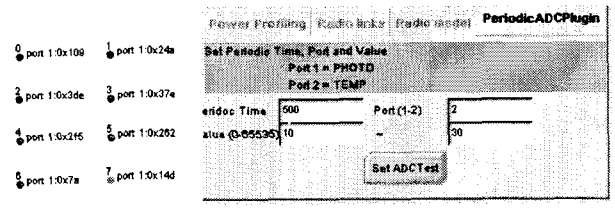
(그림 7)은 ADCPlugin의 클래스 다이어그램을 보여주고 있다. ADCPlugin 클래스는 Plugin 클래스를 상속받고, SimDriver 클래스와 SimComm 클래스를 이용하여 TinyViz에서 선택된 포트에 센서 값을 입력할 수 있다.



(그림 9) ADCPlugin의 클래스 다이어그램

4. TinyViz 플러그인 PeriodicADCPlugin을 설계

본장에서는 TinyViz 플러그인 ADCPlugin을 확장한 PeriodicADCPlugin의 설계를 살펴본다. PeriodicADCPlugin은 시뮬레이션을 할 때 센서 값을 모트에 주기적으로 입력할 수 있도록 지원하여 주는 TinyViz 플러그인이다. PeriodicADCPlugin은 시뮬레이션을 위한 환경 정보를 범위를 지정하고 범위내의 센서 값을 생성하거나, 파일이나 스크립트로 미리 제작하여 그래픽 유저 인터페이스상에서 간편하게 센서 값을 불러와서 시뮬레이션을 지원하여 준다. (그림 8)은 PeriodicADCPlugin의 동작 모습을 보여주고 있다. Peridoc Time을 500으로 설정하고, 온도 센서 값을 설정하는 Port 번호 2를 설정하고, 온도 값의 범위를 10에서 30으로 설정하면 500ms 마도 10도에서 30도 사이의 온도 센서 값을 자동으로 발생하여 시뮬레이션을 수행하게 된다. 정수로 입력한 센서 값은 Deciphering TinyOS Serial Packet[6]에 구현되어 있는 Calculating Temperature in Engineering Units를 이용하여 모트에 16진수로 입력되게 된다.



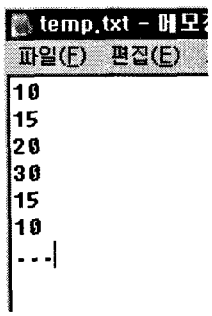
(그림 10) PeriodicADCPlugin의 실행 화면

PeridocADCPlugin은 (그림 9)의 주기적인 Action을 수행하는 Tython Periodic 명령 스크립트를 이용하여 설계하였다.

```
from simutil import *
def uart_beacon(mote, period, msg):
  def sendMessage(event):
    comm.sendUARTMessage(mote, 0, msg)
  return Periodic(period, sendMessage)
```

(그림 11) Tython Periodic command

시뮬레이션을 할 때 주기적으로 센서 값을 설정하기 위하여 위에서 설명한 범위를 설정하고 랜덤으로 센서 값을 설정하는 방법 외에 파일에 센서 값을 입력해두고, 입력된 값을 차례대로 불러와서 모트에 설정하는 방법과 센서 값을 스크립트로 설정하는 방법을 설계하였다. (그림 10)과 같이 미리 센서 값을 입력한 ex)temp.txt 파일을 불러와서 주기적으로 입력된 값을 모트에 설정할 수 있다.



(그림 12) temp.txt

낮동안의 온도변화를 일정한 주기를 두고 미리 입력해두고, PeriodicADCPlugin을 이용하여 입력된 파일의 값을 불러오면 실제 센서 모트를 이용하여 온도 값을 센싱하는 것과 같은 시뮬레이션을 실행할 수 있다.

5. 결 론

본 논문에서는 USN과 센서네트워크 운영체제인 TinyOS, 센서네트워크 시뮬레이터인 TOSSIM과 그래픽 유저 인터페이스 환경을 지원해주는 TinyViz, Tython 스크립트 언어에 대하여 살펴보고, 주기적인 센서 값을 입력할수 있는 PeriodicADCPlugin의 설계에 대하여 기술하였다. 기존의 시뮬레이터 환경에서는 정적인 센서 값을 입력하여 시뮬레이션을 하거나 스크립트를 통하여 시뮬레이션이 가능 하였지만, PeriodicADCPlugin을 이용하면 그래픽 유저 인터페이스 환경에서 주기적으로 센서 값을 입력하여 보다 능동적이고 간편하게 시뮬레이션을 할 수 있게 된다. 하루의 조도와 온도변화 등을 미리 스크립트나 파일로 작성해두고 이를 이용하여 실제 상황과 비슷한 환경을 시뮬레이션 할 수 있기 때문에 센서네트워크 응용 프로그램의 개발에 많은 도움을 줄 수 있게 된다.

향 후 연구과제로는 설계된 PeriodicADCPlugin의 개발을 완료하고, 더욱 다양한 센서 값을 그래픽 유저 인터페이스 상에서 편리하게 입력할 수 있는 TinyViz 플러그인을 개발할 예정이다.

6. 참고문헌

[1] "http://www.tinyos.net", TinyOS

[2] Philip Alexander Levis, ("TinyOS: An Open Operating System for Wireless Sensor Networks (Invited Seminar)," mdm, p. 63, 7th International Conference on Mobile Data Management (MDM'06))

[3] Welsh, M. ; Culler, D. ("TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications" SenSys '03, pp.126-137)

[4] Victor Shnayder , Mark Hempstead , Bor-rong Chen , Geoff Werner Allen , Matt Welsh ("Simulating the power consumption of large-scale sensor network applications", SenSys '04 pp.310-317)

[5] Michael Demmer and Phil Levis " http://www.tinyos.net/tinyos-1.x/doc/tython/tython.html", Tython

[6] Jeff Thorn, (Director of Product Development : "Deciphering TinyOS Serial Packets" , Octave Tech Brief #5-01, March 10, 2005)