

# 무선 센서네트워크를 이용한 자성체의 구역 추적 시스템

김영만<sup>0</sup> 김광훈  
 국민대학교 컴퓨터학부

ymkim@cclab.kookmin.ac.kr , devilord97@hotmail.com

## Sector Tracking System for Moving Object using Magnetic Sensor in Wireless Sensor Network

Young Man Kim<sup>0</sup> , Kwang Hun Kim  
 School of Computer Science, Kookmin University

### 요 약

인간과 컴퓨터와 사물이 유기적으로 연계되어 다양하고 편리한 새로운 서비스를 제공해 주는 유비쿼터스 컴퓨팅 환경에서, 외부 환경의 감지와 제어 기능을 수행하는 센서 네트워크 기술이 최근 활발히 연구가 되고 있다. 이러한 센서 네트워크 기술은 저전력 저가격의 무선 통신 기술, 초소형 마이크로 프로세서 기술, 자동구성이 가능한 ad-hoc 네트워크 기술, MEMS기술, 다양한 종류의 센서들과 이들의 표준화 노력, 그리고 임베디드 시스템 기술등의 발전으로 실현이 가능한 기술로서 평가되고 있다. 본 논문에서는 무선 센서 노드인 MICA2[1]와 자기장센서 HMC1002[2]를 이용하여 자기장의 변동을 감지하여 자성체의 위치를 파악하는 시스템의 설계 및 구현을 다룬다.

### 1. 서론

인간과 컴퓨터와 사물이 유기적으로 연계되어 다양하고 편리한 서비스를 제공해 주는 유비쿼터스 컴퓨팅 환경에서, 외부 환경의 감지와 제어 기능을 수행하는 센서 네트워크 기술이 최근 활발히 연구되고 있다. 센서 네트워크의 하드웨어와 소프트웨어 플랫폼의 개발과 응용분야의 발굴을 위하여 학계와 산업체에서 많은 연구가 이루어지고 있으며, 대표적인 센서 네트워크 연구 그룹으로는 센서노드용 하드웨어인 MICA와 운영체제인 TinyOS[3]를 개발한 버클리 대학과 상업용 응용 시스템에 필요한 연구를 수행하는 Intel[4]을 들 수 있다. 본 논문에서는 무선 센서 네트워크용 운영체제인 TinyOS와 소프트웨어 개발언어인 NesC 기반의 응용 프로그램 개발환경에서 실험 테이블 상에 일정한 간격으로 설치된 Mica2 무선 노드들에 부착된 자기장 센서 HMC1002를 이용하여, 테이블 위를 지나가는 무선 조종 소형 모형차의 하단에 부착된 자석이 생성하는 자기장의 변화를 센싱하여, 차의 위치를 추적하는 구역 추적 시스템의 설계 및 구현을 설명한다.

### 2. 연구 동향

무선 센서 네트워크의 초기 연구단계에는 주로 군사용이나 과학적 응용 시스템이 그 대상이었다. 예를 들어, 직접 접근이 위험한 적진의 군사 동향 감시를 위하여 사용된 무인 정찰 센서 네트워크 시스템이라든지, 장기 간의 밀접 관찰이 필요한 생태계 감시 시스템이 이에 해당한다. 현재 센서 네트워크는 상업적 응용으로 교통의 감시와 제어를 위한 최첨단 지능형 교통 시스템, 그리고 트럭에 타이어 압력 센서와 유압측정 센서 등을 부착한후 이를 센서 네트워크로 연결하여 트럭 감시와 제어에 사용되는 응용 등 다양한 분야에서 확장, 연구되고 있다. 본 논문에서 사용된 저전력 무선 센서네트워크 노드인 MICA2의 응용사례를 보면, 미국 플로리다주의 MacDill Air Force base 지역에서 사용된 "DARPA NEST" 프로그램[5]이 있다. 이것은 시간동기화, GRID라우팅, Reliable Communication 등의 여러 알고리즘을 사용하여, 적의 침입

탐지, 상세화(민간인, 군인, 전차), 추적을 수행하였다. 또한 국내의 한국정보통신대학 개인무선통신연구센터에서 테스트한 무인정찰 센서 네트워크 플랫폼이 있는데, 이는 UAV(Unmanned Aerial Vehicle)를 활용하여 정찰하고자 하는 목적지에 MICA 노드를 운반하여 정찰지역에 투하한 후, 정찰지를 비행하며 노드들이 측정한 데이터를 수집하는 형태의 응용 플랫폼이다.

### 3. 구역 추적 시스템 설계

자성체는 그 주변에 자기장을 형성한다. 이 자성체가 이동하게 되면 자기장의 모양에 변화가 생기며 파동을 생성한다. 자기장 센서인 HMC1002는 Mica2 노드에 장착되어 -6 ~ +6 gauss 범위의 자기력 선속밀도를 감지할 수 있다. 본 절에서는 MICA2 노드 6개를 15개의 구역으로 나눈 실험 테이블 위에 적절히 배치하여 자성체가 움직일 때 일어나는 자기장의 변화를 감지하도록 하여 자성체가 위치한 구역을 추적할 수 있는 시스템을 설계한다.

자성체 구역 추적 시스템은 자기장을 감지해서 RF 통신을 사용하여 메시지를 보내주는 센서네트워크부와, 무선 메시지를 수신한 후 시리얼 포트를 통해 PC에 전달해 주는 Base부, 그리고 시리얼 포트로 전달되어 온 메시지를 이용해 자성체의 위치를 계산하여 모니터 화면에 그래픽 영상으로 표현해 주는 클라이언트 프로그램부로 구성되어 있다.

#### 3.1 Magnetic Sensor HMC1002

HMC1002 자기장 센서는 x, y 축 두 방향의 자기력 선속 밀도를 감지할 수 있다. [표 1]은 자기장 센서가 자성체와 센서의 거리에 따라 감지한 센싱 값을 실험을 통해 얻은 결과이다. 본 논문에서는 센싱 값 자체를 활용하지는 않고 자성체의 위치 이동시에 일어나는 값의 변화 여부만을 사용하였다.

거리 c m	X	Y	거리 c m	X	Y
0	8802	192	26	4643	2927
2	8802	192	28	4601	2939
4	8802	1102	30	4570	2940
6	8802	1984	32	4550	2944
8	7103	2474	34	4535	2943
10	6273	2672	36	4520	2946
12	5709	2789	38	4506	2947
14	5364	2844	40	4506	2950
16	5122	2873	42	4495	2952
18	4967	2908	44	4487	2952
20	4845	2928	46	4479	2952
22	4765	2926	48	4479	2951
24	4692	2925	50	4489	2964

표 1. HMC1002의 자기장 센싱 결과

3.2 실험 테이블의 배치

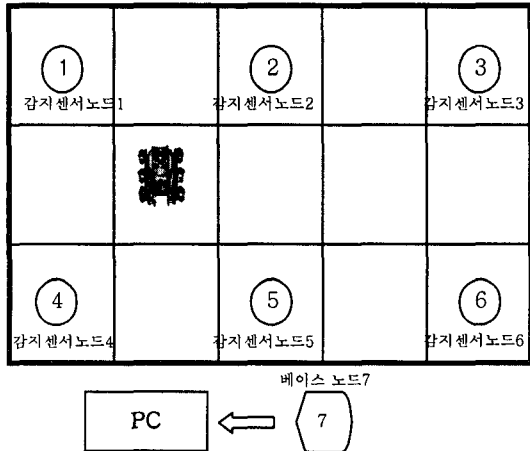


그림 1. 전체시스템 배치도

[그림 1]은 MICA2 노드 6개로 구성된 센서네트워크부와 노드 1개로 이루어진 Base부, 그리고 PC의 클라이언트부로 이루어진 전체 시스템 구성도를 보여준다. 그림에서 1~6까지 숫자가 적혀 있는 원은 각각의 HMC1002를 탑재한 센서를 뜻한다. 본 실험에 있어서 테이블의 실험 영역은 6개의 센서들에 의해 총 15개의 구역으로 나뉘어 지게 된다. 그리고 1,2,4,5 번 센서 사이에 위치한 소형자동차는 무선으로 조정되는 모형차로 하단에 작은 구형의 자석이 부착되어 있어서 자성체의 역할을 담당하게 된다.

3.3 센서 네트워크 소프트웨어 구조

각 센서노드는 무선 통신을 통해 시간 동기화를 하게 되고 일정 시간 간격으로 자기장을 감지해 이전에 감지했던 값과 차이가 있을 경우 자기장의 변동을 감지했음을 Base부에 무선으로 알려주게 된다. [그림 2]는 센서 네트워크부의 각 노드에 내장된 프로그램의 모듈 구성도이다.

3.4 Base 노드 소프트웨어 구조

Base부는 Mica2 노드 한개를 사용한 BS(Base Station) 로

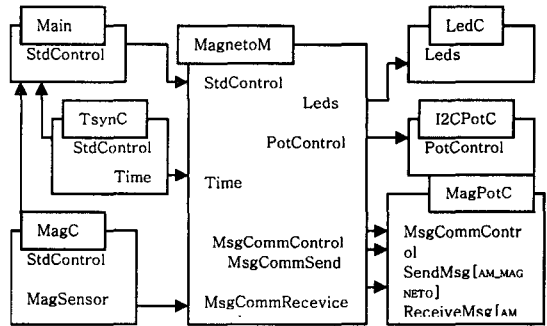


그림 2. 센서네트워크부 노드용 프로그램 모듈 구성도

이루어져 있으며, BS는 PC의 Serial Port와 케이블로 연결되어 있다. Base 부는 시간동기화 등의 작업 없이 대기하다가 감지센서가 RF 통신으로 보낸 메시지를 받게 되면 이를 패킷화 한 뒤, 케이블을 통해 PC에 전달한다. [그림 3]은 base 노드에 내장된 프로그램의 모듈 구성도를 나타낸다.

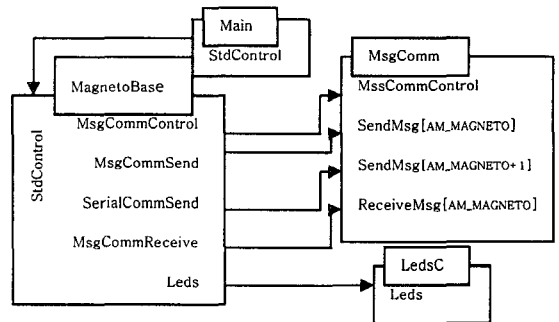


그림 3. Base 노드용 프로그램 모듈 구성도

3.5 PC 용 클라이언트 소프트웨어

클라이언트 프로그램은 시리얼 포트를 통해 Base부로부터 받은 패킷들을 해석하여 자성체의 현 위치를 파악 한 후 화면에 그래픽 영상(visualization)으로 실시간 추적 상황을 그려준다. [그림 4]는 클라이언트 프로그램의 실행화면을 보여주고 있는데, 15개의 사각형구간은 추적 영역들을 표시하고 6개의 원은 센싱 노드들을 나타낸다. 파란색은 자기장의 변화를

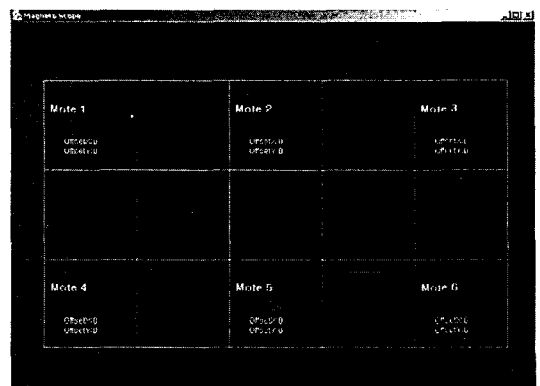


그림 4. Client Program 실행화면

감지하지 못한 노드이고, 붉은색은 현재 자기장의 변화를 감지한 노드를 나타낸다. 그림에서 5번과 6번 노드사이의 보라색 차량 그림은 현재 자성체가 위치해 있다고 계산된 구역을 나타낸다.

#### 4. 구역 추적 시스템 구현

센서네트워크부와 Base부에 사용된 프로그램은 TinyOS 및 NesC 응용 개발환경에서 구현하였으며, 클라이언트 프로그램은 PC의 윈도우환경에서 J2SE 기반의 자바언어로 구현하였다.

#### 4.1 센서 노드 소프트웨어

##### 4.1.1 시간 동기화

센서노드들 간의 시간동기화는 동일한 순간에 센싱한 값을 이용하여 자성체의 위치를 추적하기 위해 구현되었다. 시간동기화를 위한 기본적인 아이디어는 LocalTime, GlobalTime에 있다. LocalTime은 각 노드에서 독자적으로 사용하는 시간이고, 수정되지 않는 값이다. GlobalTime은 모든 노드가 공유하게 될 전체 네트워크 시간이다. GlobalTime과 LocalTime 사이에는 다음 공식이 성립한다.

$$\text{GlobalTime} = \text{LocalTime} + \text{Offset}$$

노드들은 일정 시간 간격으로(여기서는 15초) 현재 자신이 계산한 GlobalTime 값을 무선통신으로 다른 센싱 노드들 모두에게 알려 주며(broadcasting), 다른 노드들로부터 GlobalTime 값을 받은 노드는 현재 자신이 가지고 있는 GlobalTime 값과 비교, 값이 큰 쪽에 맞추어 Offset값을 조정해 준다.

##### 4.1.2 자기장의 감지와 송신

각 센싱 노드들은 GlobalTime 기준으로 일정 시간마다 자기장 센서값을 읽은 후 이전에 감지한 값과 비교하여 차이가 있을 경우, 메시지를 만들어 Base부에 무선통신으로 알려주게 된다. 이 때 이전에 감지한 값과 현재 감지한 값에 대해 어느 정도의 오차 값을 인정하여 평활화 필터를 사용하여 자기장 측정치의 noise를 제거 한다.

#### 4.2 Base 노드 소프트웨어

BaseStation 노드는 센서 네트워크부의 노드들이 보낸 메시지를 받아 패킷화한 뒤 케이블을 통해 PC로 전달해 주는 역할을 한다.

#### 4.3 Client Program

Client Program은 다음과 같이 크게 3개의 thread로 구성되어 있다.

##### 4.3.1 listenMsg Thread

이 thread는 무한 루프를 돌면서 시리얼 포트로부터 패킷이 들어 올 때까지 대기(wait) 하게 된다. 패킷이 수신 될 경우 해당 노드의 번호와 센싱 값을 추출해낸다. 다음으로 이 센싱 값의 오차를 줄이기 위해 최근 몇 회의 측정치들의 평균을 구해서 저장한다.

##### 4.3.2 checkSensingValue Thread

이 쓰레드는 일정 시간 간격으로 현재 저장되어 있는 센싱 값들을 이용하여 각 센싱 노드들이 위치한 구역에서 자기장의 변화가 있었는지를 파악하여 그 결과를 저장하게 된다. 여기서 변화라 함은 지금 파악되어 있는 센싱값과 바로 이전의 값 사이에 차이가 있다는 것을 뜻한다. 이때 noise를 줄이기 위해

상수 OSCFLAGWINDOW와 상수 CHECKLIMIT를 사용하는 데, 이는 최근 OSCFLAGWINDOW 횟수의 기간중에 CHECKLIMIT 값 이상의 횟수 만큼 변화가 있다면 해당 노드가 있는 구역은 진동이 있다고 파악하겠다는 것이다.

##### 4.3.3 scopeViewer Thread

이 쓰레드는 일정 시간(상수 SCOPETEMPO) 간격으로 각 센싱노드들이 위치한 곳의 자기장 변화 여부를 이용하여 자성체가 위치한 구역을 파악한 후에 그 결과를 화면에 그려주게 된다. 자성체의 위치를 구하는 절차는 다음과 같다. 각 노드가 위치한 구역의 자기장 변화 여부로 자성체가 위치한 구역을 1차적으로 판별 하는데, 이때 두 곳 이상에서 동시에 변화가 있다면 해당 구역들의 중간 지점을 자성체가 위치한 구역으로 파악한다. 이때 1차적으로 파악된 구역은 오차가 있을 확률이 높는데 이는 현재 자성체가 위치한 노드 인근에 위치한 다른 노드들이 파악한 잠깐의 변화로 인해 위치변동이 일어났다고 판단하는 경우가 많기 때문이다. 이러한 경우를 줄이기 위해 위치변동 filter를 적용 시킨다. 이것은 자성체의 기존 위치를 일정 시간동안 변동없이 유지시켜 주는 것으로 구간 이동이 연속적으로 일어나는 횟수가 일정한 레벨이상이 되기 전까지는 구간 이동을 인정하지 않고 현재 위치를 예전에 파악되었던 위치로 바꿔 버리는 filter이다. 또한 위치변동 filter 적용 후에도 자기장의 여진과 노이즈로 자성체가 한번에 한 구간 이상의 거리를 이동하는 결과를 보일 때 강제로 파악된 이동 거리를 한 구간으로 줄여버리는 one-step filter를 적용 시킨다. 마지막으로 자성체가 짧은 시간 동안 두 구역 사이를 왕복하는 backwarp 현상을 보일 때, 현재 위치를 고정시켜 주는 backwarp filter를 적용시킨다.

#### 5. 결론 및 향후 과제

본 논문에서는 자기장을 센싱하는 노드 6개와 측정값들을 수집하여 PC에 전달해주는 BS노드 1개를 사용하여, 자성체가 15개의 구간 중 어느 구간에 위치해 있는지를 파악하는 구역 추적시스템을 설계 및 구현하였다. 센싱네트워크부의 시간동기화와 1차적인 노이즈 제거, 클라이언트 프로그램에서의 3번의 filtering을 사용한 결과 상당히 신뢰성 있는 결과를 얻을 수 있었다. 향후 과제로는 센싱 노드들의 센싱값의 변화 여부만을 따지지 않고, 센싱값 자체를 활용하여 15개 보다 많은 구역을 나누거나, 아니면 구역이 아닌 보다 높은 정확도의 2차원 위치를 파악하는 시스템의 설계를 들 수 있다.

#### 참고문헌

- [1] MICA2, <http://www.xcross.com>.
- [2] HMC1002, [http://www.ssec.honeywell.com/magnetic/datasheets/sensor\\_catalog.pdf](http://www.ssec.honeywell.com/magnetic/datasheets/sensor_catalog.pdf).
- [3] TinyOS, <http://webs.cs.berkeley.edu>.
- [4] Intel, <http://www.intel.com/research/exploratory/wireless-sensors.htm#sensornetwork>.
- [5] DARPA NEST Project, <http://www.cse.ohiostate.edu/siefast/nest/nest-webpage/ALineInTheSand.html>.