

모바일 센서 네트워크를 위한 에너지 효율적이고 경제적인 소형 이동 로봇의 개발

Energy-Effective Low-Cost Small Mobile Robot Implementation for Mobile Sensor Network

김 홍 준*, 김 병 국
(Hongjun Kim and Byung-Kook Kim)

Abstract : In this paper, we describe an implementation of small mobile robot that can be used at research and application of mobile sensor networking. This robot that will constitute the sensor network, as a platform of multi-robot system for each to be used as sensor node, has to satisfy restrictions in many aspects in order to perform sensing, communication protocol, and application algorithms. First, the platform must be designed with a robust structure and low power consumption since its maintenance after deployment is difficult. Second, it must have flexibility and modularity to be used effectively in any structure so that it can be used in various applications. Third, it must support the technique of wireless network for ubiquitous computing environment. At last, to let many nodes be scattered, it must be cost-effective and small. Considering the above restrictions of the mobile platform for sensor network, we designed and implemented robots control the current of actuator by using additional circuit for power efficiency. And we chose MSP430 as MCU, CC2420 as RF transceiver, and etc, that have the strength in the aspect of power. For flexibility and modularity, the platform has expansion ports. The results of experiments are described to show that this robot can act as sensor node by RF communication process with Zigbee standard protocol, execute the navigation process with simple obstacle avoidance and the moving action with RSSI(Received Signal Strength Indicator), operate at low-power, and be made with approx. \$100.

Keywords : mobile sensor network, small mobile robots

I. 서론

최근 들어 관심이 높아지고 활발히 연구되고 있는 센서네트워크 분야는 일반적으로 센싱 기능과 계산 처리, 그리고 무선통신 능력을 가지고 있는 센서 노드와 게이트웨이 또는 데이터 집중 국 역할을 하는 베이스 스테이션으로 구성된 무선 네트워크로 그림 1과 같이 구성된다.

이와 같은 시스템을 구성할 때 사용되는 로봇들은 다음과 같은 여러 가지 제한 조건을 충족시켜야 한다.

- 저전력: 센서 노드들은 주로 물리적인 접근이 불편하거나 용이하지 않는 곳에 설치되어 작동하게 되므로 battery를 교환하는 비용이 오히려 더 커질 수 있다. 최대한 에너지를 고려한 저전력 설계를 통해 수명을 연장해야 한다.

- 비용효율성: 일반적으로 많은 노드들로 센서 네트워크를 구성하므로 기능에는 문제가 없으면서 가격이 낮은, 즉 비용효율성을 고려하여 많은 개체를 만들어 활용하기에 적절 해야 한다.

- Robustness & Flexibility: 이미 많은 노드들로 구성되어 있는 네트워크에 새로운 노드들이 추가되는 경우가 많고, 예측 불가능한 상황에 따른 고장이나 방전 때문에 노드가 오동작하거나 이탈하는 등 예기치 않은 상황을 맞이하는 경우가 생길 수 있기 때문에 네트워크 토폴로지에 변화를 주거나 자체적인 이동을 통해 시스템을 재구성하는 등의 대처가 자유롭게 이루어 질 수 있어야 한다.

센서 네트워크를 위해 만들어진 로봇은 아니나 여러 가지 다양한 로봇 어플리케이션에 쓰이고 있는 이동 로봇으로 Pioneer Robot[1]이 있다. Mapping, 위치 추정, 정찰, 협동 작업 등 다양한 어플리케이션에 사용될 수 있는, 기능면에서 강력한 이동 로봇으로 거리 센서로 정확한 resolution의 Laser scanner를 전방에 설치하고 그 밖에 초음파 센서 및 카메라까지 있으며 이를 32bit RISC 기반 컨트롤러로 제어하고 있다 (그림 2). 또한 기계적으로도 robust한 구조를 이루고 있어 짐을 싣고 달릴 수도 있도록 설계되었다. 그러나 가격이 수천\$에 달하고 크기가 44cm x 38cm x 22cm로 커서 다 개체 로봇 시스템에서의 연구에는 사용되기 어렵다.

Pioneer외에도 작은 크기를 가지고 있는 Khepera[2]라는 로봇이 있다. 그 크기에 비해 다양한 어플리케이션에 사용할

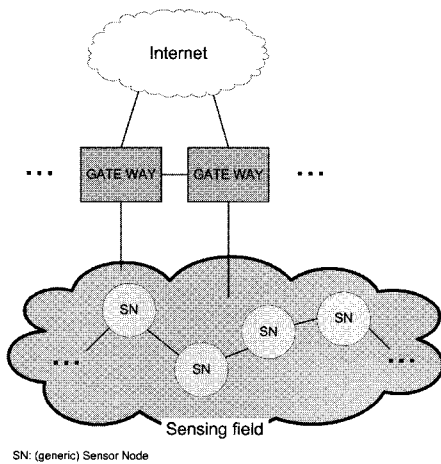


그림 1. 센서 네트워크.
Fig. 1. Sensor network.

* 책임저자(Corresponding Author)
논문접수 : 2007. 5. 17., 채택확정 : 2008. 1. 14.
김홍준, 김병국 : 한국과학기술원 전자전산학과 전기전자전공
(hjkim@rtcl.kaist.ac.kr/bkkim@ee.kaist.ac.kr)

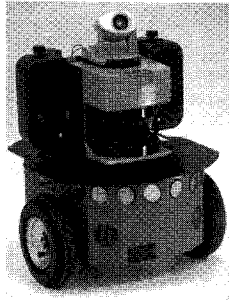


그림 2. Pioneer III.
Fig. 2. Pioneer III.

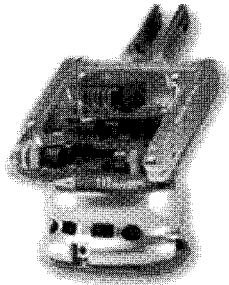


그림 3. Khepera II.
Fig. 3. Khepera II.

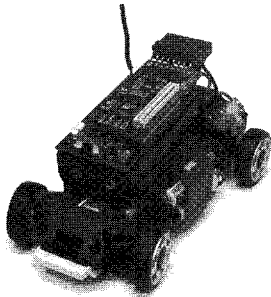


그림 4. CotsBots.
Fig. 4. CotsBots.

수 있도록 강력한 프로세서를 지니고 있다는 점뿐만 아니라 쉽게 살 수 있는 점에서 매우 매력적이지만, 가격이 약 \$2000나 된다(그림 3).

센서 네트워크에서의 이동로봇을 노드로 사용하면, 즉 노드들 혹은 여러 개의 노드들로 구성되어 있는 클러스터에서 헤드 역할을 하는 노드들에 이동성을 부여하게 되면 노드 배치와 네트워크 재구성 알고리즘의 개발을 통해서 센서 네트워크의 에너지 측면이나 유연성 측면 등에서 향상을 이끌어 낼 수 있다. 꼭 센서네트워크를 이루는 모든 노드에 이동성을 부여할 필요는 없으며, 이동형 노드와 고정형 노드들을 조합하여 에너지 효율의 향상을 이끌어 낼 수 있다.

Khepera 및 Pioneer와 같이 기존의 이동 로봇들이 센서 네트워크에서 사용될 것을 고려하여 전력 및 비용 등의 측면에서 효율적으로 설계된 로봇은 아니므로 네트워크를 구성하게 될 멀티 로봇 시스템에서의 연구에 비효율적이라는 것을 인지하고 여러 가지 제한 요건을 해결한 로봇들이 개발되었는데, CotsBots[3], Robomote[4], Micabot[5], Millibot[6]과 같은 로봇들이 이에 해당한다.

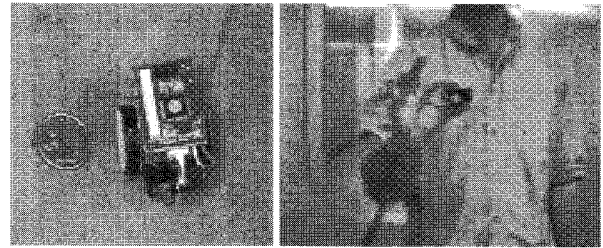


그림 5. Robomote.
Fig. 5. Robomote.

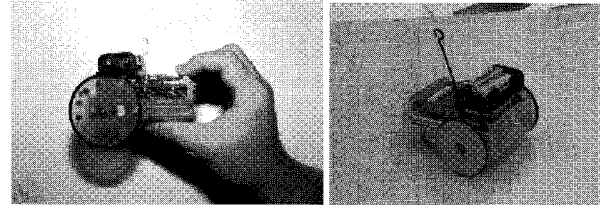


그림 6. Micabot.
Fig. 6. Micabot.

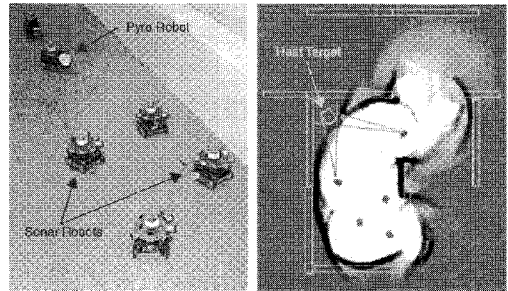


그림 7. Millibot.
Fig. 7. Millibot.

CotsBots[3](그림 4)는 버클리대학에서 개발한 작은 크기 (7cm x 7.5cm x 5cm-구동부 제외)의 이동 로봇 platform으로서 센싱 및 프로세싱, 통신을 담당하는 Mica Mote라는 보드와 모터 제어를 위한 프로세서 및 제어 회로가 꾸며진 모터 보드로 이루어져 있다. 구동부에는 미니 자동차를 사용하였고 센서 추가를 위한 부가적인 센싱보드도 부착 가능토록 되어 있다. \$200 미만으로 구현할 수 있으나 이는 센서를 뺀 가격이다.

Robomote[4](그림 5)는 \$150 미만으로 구현할 수 있도록 구현된 로봇으로 크기가 7cm x 4.5cm x 3.5cm으로 매우 작다. 작은 바퀴로 인해 속도는 7.5cm/s로 느리고 OOK (on-off keying) 방식을 이용한 무선 통신 또한 19.2Kbps로 느린 편이다.

Micabot[5](그림 6)은 Robomote보다는 크지만 비교적 작게 만들어진(8.6cm x 6.1cm x 5.4cm)로봇으로 RF Monolithic TR1000 트랜시버를 이용하여 115Kbps로 통신하고, 홀 이펙트 센서를 이용하여 직접 간이 엔코더를 만들었다. 그러나 비용이 \$350 정도 들어간다.

Millibot[6](그림 7)은 앞서 언급한 로봇들에 비해 센싱 보다는 주행 및 협동작업이 더 강조된 로봇으로 크기는 비슷하나 (7cm x 7cm x 7cm) 주행을 위한 센서 때문에 \$500 이상의 비용이 든다.

이와 같이 실제로 센서 네트워크를 구성 또는 연구하기 위한 플랫폼으로써 많은 로봇들이 개발되었으나 각각의 로봇들마다 개선의 여지가 있고 좀 더 비용효율적이면서도 전력 소모 및 기능면 등에서 성능이 향상된 로봇의 개발이 바람직하다. 본 논문에서는 한정된 장소에서 다개체(>50)로 실험할 수 있을 만큼 작은 크기(7cm x 7.7cm x 5cm)면서, 센서 노드의 역할을 충분히 이행할 뿐만 아니라 부가적인 센서 및 프로그램을 통해 여러 어플리케이션에 응용할 수 있도록 유연성이 있는 소형 이동 로봇의 설계구현을 소개한다. 무엇보다 모바일 센서네트워크에서 이용될 로봇에 있어서 가장 중요한 특성, 즉 전력과 비용 면에서 매우 효율적으로 설계하여 작동시간이 길고 (약 200일), 기본적인 장애물 회피도 가능한 이동 로봇을 재료비 기준 약 \$100정도로 구현할 수 있어 매우 비용효율적이다.

본 논문은 총 5장으로 구성 되어 있다. 제 2장에서는 크기 및 전력 소모, 가격 등, 제한된 조건을 충족하는 로봇 설계에 관한 내용들을 실었고, 제 3장에서는 설계를 바탕으로 실제로 구현된 로봇을 하드웨어와 소프트웨어 부분으로 나누어 소개한다. 제 4장에서는 최종 구현한 이동 로봇의 간단한 주행과 같은 실험과 어플리케이션에 대한 내용들을 포함한다. 마지막으로 제 5장에서는 연구에 대한 결론과 앞으로의 과제에 대해서 기술하였다.

II. 설계

이 장에서는 본 연구를 위해 실제로 설계제작된 이동 로봇의 하드웨어 설계에 관련된 부분과 소프트웨어 구조에 대한 내용을 기술한다.

1. 하드웨어

센서 노드의 하드웨어는 다양한 종류의 센서들과 ADC (Analog Digital Converter)로 구성된 센싱 유닛, 소프트웨어를 탑재하게 되는 프로세싱 유닛, 그리고 센서 데이터를 비롯한 무선 데이터 통신에 필요한 무선 트랜시버 등이 필수적이다. 그 밖에 이동성을 위해 구동 유닛이 필요하다. MCU를 프로그래밍하기 위한 USB programmer 외에 구동부를 위한 모터 드라이버, 무선 통신을 위한 무선 트랜시버가 MCU에 연결되어 로봇의 작업들을 수행하게 된다(그림 8).

그 중 프로세싱 유닛, 로봇의 두뇌 역할을 담당하게 될 MCU를 선정하는데 있어 기존의 MCU시장을 살펴보면, 마이크로칩사사의 PIC계열과 인텔의 유명한 8051이 공개된 회로와 어플리케이션이 많아 널리 사용되었다. 하지만 센서 네트워크에서는 에너지 효율성을 위해 대부분 8-bit계열의 낮은 클럭을 가지는 RISC CPU가 많이 사용된다. 센서 데이터 처리를 위한 ADC가 내장되어 있으면서 저전력의 8-bit 마이크로 컨트롤러를 기반으로 하여 많은 센서 노드들의 연구가 진행되었다. 최근에 개발된 MSP430[7]이라는 16-bit 마이크로 컨트롤러는 동작 때와 휴식 때의 소모 전류량이 여타 마이크로 컨트롤러에 비해 2배 이상 작을 뿐만 아니라 여러 가지 동작모드를 지원함으로써 필요한 기능에 따라 에너지를 절약할 수 있다. 또 16-channel ADC외에도 DAC가 2 channel이나 내장되어 있어 응용 범위가 넓다는 점에서 선택되었다(표 1).

한편, 다 개체 로봇간의 통신을 위해 필요한 무선 통신에 대한 표준에 대해서 살펴보면, Zigbee가 다른 표준들보다 통신속

도는 느리지만 전력 및 단순도 면에서 탁월한 장점을 지니기 때문에 센서 네트워크 구현에 가장 적합한 통신 기술이다.

제어와 통신부외에 실제로 주행하게 될 구동 부에 있어서 가장 중요한 모터는 간단한 엔코더를 제공하는 미니 DC 모터 RGO-0331[9]을 쓰고 엔코더 신호를 통한 dead reckoning으로 주행 동작을 수행하게 된다.

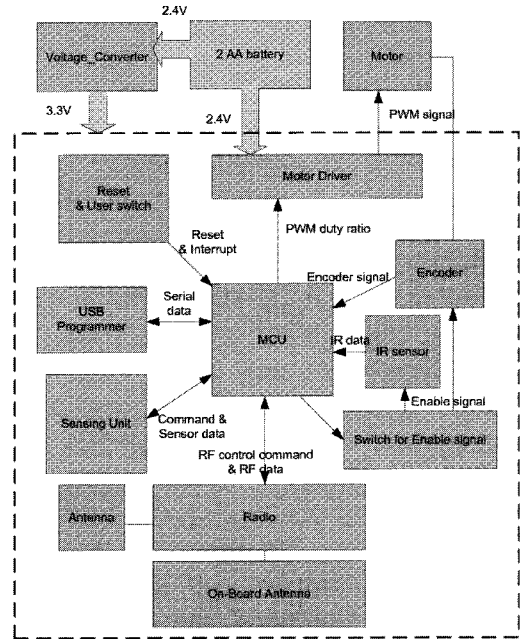


그림 8. 전체 블록 다이어그램과 데이터의 흐름.
Fig. 8. Overall block diagram and flow of data.

표 1. 센서네트워크용 MCU 비교.

Table 1. MCU for sensor networks.

Manufacture	Atmel	Texax Instruments	Motorola
Device	Atmega128L	MSP430F16x	HCS08
RAM(KB)	4	10	4
A/D converter (# of channel)	8	16	8
D/A converter (# of channel)	0	2	0
Active current (mA)	8	2	6.5
Sleep current (µA)	20	1	1

표 2. 무선 통신 표준 비교[8].

Table 2. Wireless communications.

Feature(s)	IEEE 802.11b	Bluetooth 802.15.1	Zigbee 802.15.4
Power Profile	Hours	1 Week	1 Year+
Complexity	Complex	Very Complex	Simple
Nodes/Master	32	7	64000
Range	100m	10m	70m
Data Rate	11Mbps	1Mbps	250Kbps

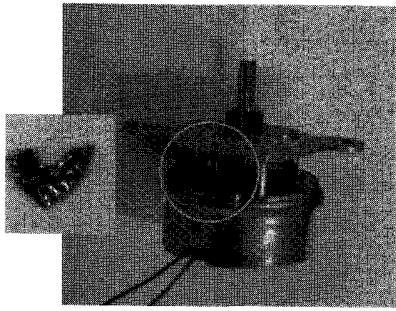


그림 9. 엔코더 설치.
Fig. 9. Installation of encoder module.

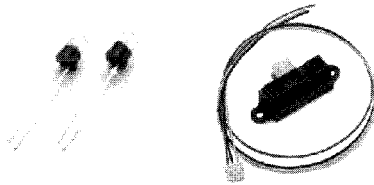


그림 10. IR diode(좌)와 IR distance sensor(우).
Fig. 10. IR diode(left) and IR distance sensor(right).

표 3. 거리 센서 비교 (정면의 백색 평면).
Table 3. Distance sensors (frontal white plane).

Device	IR sensor (SI-5315D - ST-5811)	IR distance sensor (GP2d120[11])
Cost	\$1	\$10
감지 거리	0.5cm~16cm	4cm~38cm

엔코더는 적외선 발광부와 수광부 소자를 이용한 회로로 꾸며져 있다. 이를 모터의 톱니바퀴 부분을 감지하도록 설치하였다(그림 9).

주행하는 도중에 장애물이 나타남을 감지하기 위해서는 쓰는 거리센서는 매우 다양한 종류가 있지만 로봇의 크기나 무게를 고려하고 필요한 감지거리를 20cm 정도로 볼 때, IR distance sensor와 IR다이오드가 적당하다(그림 10).

특히 비용 면에서 매우 합리적인 IR 다이오드와 Photo Transistor 조합(SI-5315[10]/ST-5811)을 이용한다(표 3).

2. 소프트웨어

앞서 언급한 바와 같이 센서 노드는 MCU를 내장한 일종의 소형 컴퓨터시스템으로, 센싱 데이터를 받아들여 처리할 수 있는 센싱 응용 처리 기능과 노드들 간의 통신, 그리고 구동 부를 위한 모터 제어 등의 여러 가지 기능을 수행해야 하기 때문에 운영체제를 필요로 하며, 이러한 운영체제들은 설계 시 여러 가지가 고려되어야 한다. 우선 자원이 제한적인 환경, 즉 메모리 및 computing power가 제한된 시스템에서 사용되기 때문에 작은 크기여야 할 뿐만 아니라 전력 소모가 적어야 한다. 또한, 센서 노드 간에 저전력 통신을 제공하면서도 프로세스와 메모리를 효율적으로 관리하도록 설계되어야 한다. 그리고, 센서 네트워크의 전체적인 에너지 효율을 높이기 위하여 센서 노드들 간의 시간 동기화가 이루어져야 한다. 또한, I/O장치를 하드웨어 레벨의 접근 방식에 의해 접근하여야 하기에 이러한 I/O의 하드웨어적인 여러 제약 사항

들을 고려해야 한다. 현장에 배치된 후로는 유지 보수가 어렵고 환경의 영향을 많이 받을 수 있기 때문에, 이러한 문제점을 해결하기 위하여 동적으로 환경에 대응할 수 있는 특성을 지녀야 한다. 그리고 실제적으로 다양한 응용분야에 쓰일 수 있는 센서 노드임을 감안하여 응용분야마다 다를 수 있는 하드웨어의 다양성 속에서 효과적으로 사용할 수 있도록 또, 빠르게 변화하는 기술 발전에 발맞추어 가기 위해 유연성과 모듈성 그리고 용이성을 갖추고 있어야 한다.

이들을 고려 하여 개발된 대표적인 운영체제로는 버클리대에서 Mote와 함께 개발하고 있는 TinyOS[12]를 비롯하여, 콜로라도 대학의 Mantis[13], 유럽의 EYES 프로젝트의 PEEROS[14](Preemptive EYES Real-time OS), Pumpkin사의 Salvo[15] 등이 있다. Salvo는 많은 프로세서와 많은(>500) API를 제공하지만 open source가 아니다. PEEROS는 유럽의 Twente Enshcede대학에서 개발된 하드웨어 독립방식 기법을 채택한 운영체제로 멀티 태스킹 및 선점 형 기반의 스케줄링 기법을 제공하고, 저전력 파워소비를 위한 저전력 모드를 지원하며, C기반의 프로그래밍 기법을 제공하지만 TinyOS보다 많은 resource가 필요하다. 또한, MANTIS는 연구가 활발하지 않아 업데이트가 느린 단점이 있다. TinyOS는 state machine 기반의 구조를 가지는 운영체제로, 각각의 상태는 컴포넌트에 해당되며 컴포넌트끼리의 연결로 일을 수행하게 된다. 그리고 센서 노드들의 일이 요구되지 않을 경우 저전력 모드인 sleep모드로 전환함으로써 효율적인 CPU의 사용을 이룰 수 있다. 무엇보다 여타 다른 OS에 비해 가장 활발한 토의 및 연구를 통해 빠른 업데이트가 이루어 지고 있고, open source라는 장점을 지니고 있다. 또한, 전력, 메모리, 프로세싱에 제한이 있는 환경에 적합하도록 event driven방식을 따르고 function call이 아닌 in-line code로 컴파일하며 sleep상태일 때는 겨우 수 mW의 전력만을 소비한다. 그리고 모듈성과 재사용 성이 뛰어나 빠르게 변화는 센서 네트워크 분야의 연구를 위한 운영체제로서 적합하다.

TinyOS는 nesC라는 동적 메모리를 할당하지 않는 정적인 언어를 통하여 센서 네트워크용 응용 프로그램을 작성할 수가 있다. nesC는 C와 매우 유사한 언어로 네트워크를 구성하는 임베디드 시스템을 대상으로 만들어졌고, 어플리케이션 개발자를 위한 새로운 디자인 환경을 제공한다. Event driven 방식의 수행과 유연한 병행 모델, 컴포넌트화된 어플리케이션의 디자인을 결합하여 구성된 언어로 nesC 컴파일러가 전체 프로그램을 분석하여 data-race를 인지하는 등의 방식으로 디버깅이 어려운 병행성에 도움을 주고 자원소모를 줄이는 function inlining을 지원한다. 많은 노드를 프로그래밍하여 작업을 해야 하는 센서네트워크 분야에 특히 강점을 보인다.

소프트웨어는 전류소모 효율을 극대화시켜야 하는 로봇의 특징을 고려하여 전류소모가 비교적 큰 주행에 이용되는 어플리케이션들과 전류소모가 적은 센싱 어플리케이션들을 두 가지 모드로 나누어 동작할 수 있게 하였다. 이를 통해 이동할 필요가 없을 때에는 고정 센서 노드처럼 센싱 및 통신 어플리케이션만 수행하고 움직일 필요가 있을 때만 전류소모가 큰 주행 어플리케이션을 수행하는 것이 가능하도록 구현하였다(그림 11).

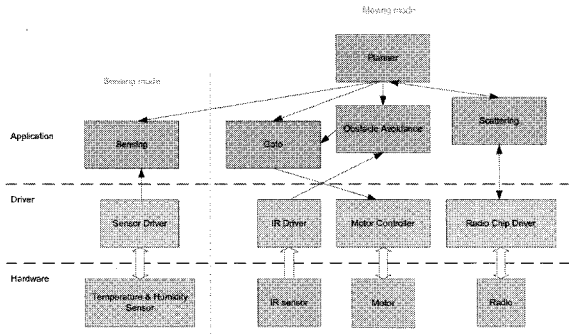


그림 11. 전체적인 소프트웨어 구조.
Fig. 11. Overall software structure.

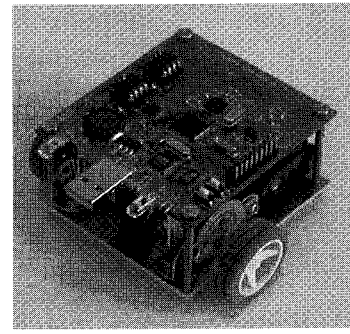


그림 12. 완성된 로봇의 모습.
Fig. 12. Appearance of the robot.

III. 구현

앞서 서론에서 말한 여러 가지 조건들을 고려하여 설계하고 구현한 로봇은 아래 그림 12와 같은 모습을 보인다.

1. 하드웨어 구현

하드웨어 부분은 로봇의 모터와 배터리, 그리고 모든 회로가 들어가는 컨트롤러 보드가 배치될 수 있도록 골격 및 판 역할을 하는 기계 부와 실질적으로 로봇의 주행을 관장하고 여러 기능을 수행하는 전자부로 나눌 수 있다.

우선 기계부는 로봇이 배터리로써 2개의 충전용 AA 건전지를 이용하여 전체 시스템을 구동하기 때문에 배터리와 모터, 컨트롤러 보드를 지탱할 베이스를 제작하였다. 이를 위해 일반적으로 가공소에서 많이 사용하는 두께 2mm의 알루미늄 판을 가로70mm, 세로77mm의 크기로 절단하고 Controller 보드와의 연결을 위한 구멍 및 모터 고정을 위한 구멍, 그리고 볼 캐스터를 위한 구멍을 위해 간단한 홀 가공을 수행하여 제작되었다.

베이스 위에 들어갈 전자부는 마이크로 컨트롤러가 들어가게 되는 보드에 MSP430뿐만 아니라 통신을 위한 RF 트랜시버(CC2420[16])와 안테나, 구동부 제어를 위한 모터 드라이버, 장애물 회피를 위한 IR 센서, 디버깅을 위한 LED, 전원 변환을 위한 Voltage converter, 전원공급 선택을 위한 아날로그 스위치, 안테나 등이 모두 하나의 보드에 구현되어 있다 (그림 13).

그리고 ADC입력 및 전원 확장 핀을 외부로 내어놓아 부가적인 센서를 통한 다양한 응용에 사용될 수 있도록 하였다. 그 밖에 프로그램 로딩을 위해 USB to UART chip(FT232 BM[17])이 들어가 있다.

충전용 AA배터리에는 1.2V의 출력 전압을 낸다고 적혀있다. 따라서 2개를 직렬로 사용하게 되면 출력 전압은 2.4V가 되나, 실제 공급하게 되는 전압은 충전 직후에는 2.7V까지 올라가지만 계속해서 사용하다 보면 2V미만으로 떨어지게 된다. 이렇게 공급전압이 시간에 따라 변화하는 상황이 발생하면 시스템의 구성요소 중에 하나인 analog switch chip의 내부 저항이 입력 전압에 따라 바뀌게 되어 전압감소량이 변화한다. 장애물 회피를 위해 설치한 IR diode의 출력 세기도 바뀌게 되어 이미 실행되어 작성된 출력전압 대 장애물 거리의 calibration값이 틀러지게 되고, 결국 장애물을 피할 수 없을 수도 있다. 그리고 시스템 전체를 관장하는 MCU, 즉 MSP430f1611 chip[7]의 동작 frequency가 입력 전압에 따라 달

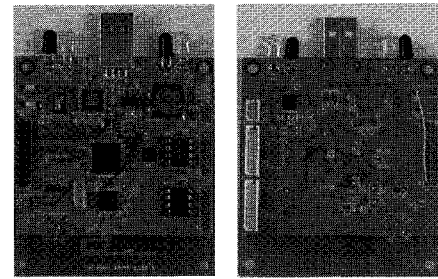


그림 13. 컨트롤러 보드 앞면(좌)과 뒷면(우).
Fig. 13. Front and rear view of controller board.

표 4. 확장용 포트.
Table 4. Expansion ports.

핀 번호	기능	핀 번호	기능
1	VCC(DVCC)	2	UART Receive (UART0RX)
3	Analog Input 0(ADC0)	4	UART Transmit (UART0TX)
5	Analog Input 1(ADC1)	6	I2C Clock(I2C_SCL)
7	Analog Input 2(ADC2)	8	I2C Data(I2C_SDA)
9	Analog Input 3(ADC3)	10	Analog Input 6(ADC6) DAC0
11	Analog Input 4(ADC4) DAC1/SVS in	12	Exclusive Digital I/O 0 (GIO0) Auxiliary Clock(ACLK)
13	Exclusive Digital I/O 1 (GIO1) Timer A Capture(TA1)	14	Exclusive Digital I/O 2 (GIO2) External DMA Trigger (DMAE0)
15	Exclusive Digital I/O 2 1 Wire	16	User Interrupt(User_INT)
17	Ground(GND)	18	Reset

라지는 특성이 수행성능의 차이로 나타나 프로세서의 main frequency를 이용하는 프로그램 수행 시 일정하지 못한 결과가 나타난다.

따라서 2개 AA 배터리의 출력 전압을 그대로 사용하지 않고 step-up DC/DC 컨버터 칩(TPS61031[18])의 출력 전압을 controller 보드의 공급전압으로 사용함으로써 배터리 출력 전

표 5. 비용.

Table 5. Cost.

Part	Cost
Mechanical Part	37,390원
(Battery, Motor, etc.)	
Parts	48,720원
(R, L, C, chips, etc.)	
PCB and assembly	5,580원
Total	91,690원(\$98.97)

압이 1.8V 미만으로 떨어지기 전까지는 항상 일정한 전압(3.3V)이 공급되어 로봇의 작업들이 일관성 있게 수행된다. 배터리를 통해서 공급된 전력으로 로봇을 움직이는데 있어서 로봇의 주행은 가장 많은 전력 소모가 일어나는 작업으로 주행하지 않을 시는 전원 공급이 이루어 지지 않도록 MCU에서 제어할 수 있어야 한다. LB1630[19]은 낮은 전압에서도 동작 가능한 모터 드라이버 chip으로 standby 모드에서의 전류소모가 30 μ A이하이므로 MCU에서 PWM신호 제어로 전류소모를 극대화 할 수 있다. 모터 드라이버 외에 엔코더, 주행시 장애물 회피를 위해 쓰는 IR 다이오드에 전원을 공급하여야 한다. IR 다이오드를 동작시킬 때 100mA의 전류가 소모되고 엔코더에는 약 20mA의 전류가 소모되므로 이를 MCU에서 차단 가능하도록 아날로그 스위치(MAX4751[20])를 사용하였다. MCU의 GPIO port로 이 스위치를 제어하여 엔코더 및 IR 센서 회로에 전원을 공급한다. 그리고, 다양한 어플리케이션에 쓰일 수 있다는 점을 고려하여 유연성 및 확장성을 갖추기 위해 확장용 포트를 설치하였다.

기능적인 면에서 이동형 센서 노드의 역할을 충분히 이행하고 다른 로봇들에 비해 부족한 면이 없으면서 비용절감 측면에서는 뛰어나도록 비용효율성을 고려하여 로봇을 설계하고 구현하였다. 하나의 로봇을 완성하는데 드는 비용은 표 5와 같다.

2. 소프트웨어 구현

간단한 이동명령(Goto)을 통해 목표 위치를 넘겨받고 이를 수행 가능하도록 표 6의 API들을 구현하였다.

그림 14는 API들을 이용하여 위치 이동 및 회전을 구현한 Goto API를 나타내었다.

goto_w_velocity에서 x와 y좌표 값이 주어지면 그 좌표 방향으로 turn API를 이용하여 회전 한다(1). 목표 좌표까지 go_forward API로 직진한다(2).

$$\text{turn}(\theta), \text{ where } \theta = \arctan(x/y) \tag{1}$$

$$\text{go_forward}(v, t), \text{ where } v \cdot t = \sqrt{x^2 + y^2} \tag{2}$$

주행 중에 장애물이 나타나면, IR 센서 데이터를 이용하여 장애물 회피 동작을 수행하게 된다. IR(Infrared Ray) 다이오드에서 쏜 적외선이 장애물에 반사되어 돌아왔을 때, 포토 트랜지스터에서 받는 세기를 측정하여 장애물과의 거리를 측정하고 이를 장애물 회피에 사용하도록 구현되어 있다. 적외선 세기로 거리를 판별하기 때문에 주위 환경에 매우 민감하여 포토 트랜지스터를 검은 색 튜브로 감싸 개선

표 6. 주행 API들.

Table 6. API for goto action.

API	Parameter	기능
go_forward	velocity: mm/s, time: 100ms	주어진 속도로 주어진 시간만큼 직진 주행한다.
go_backward	velocity: mm/s, time: 100ms	주어진 속도로 주어진 시간만큼 후진 주행한다.
go_arc	left_wheel_velocity, right_wheel_velocity, time	주어진 왼쪽 및 오른쪽 바퀴의 속도로 주어진 시간만큼 주행한다.
turn	angle: degree	주어진 각도만큼 회전 한다
stop	time: 100ms	주어진 시간만큼 정지 한다.
goto_w_velocity	velocity: mm/s, x: mm, y: mm	주어진 속도로 주어진 좌표(x, y)로 이동한다.

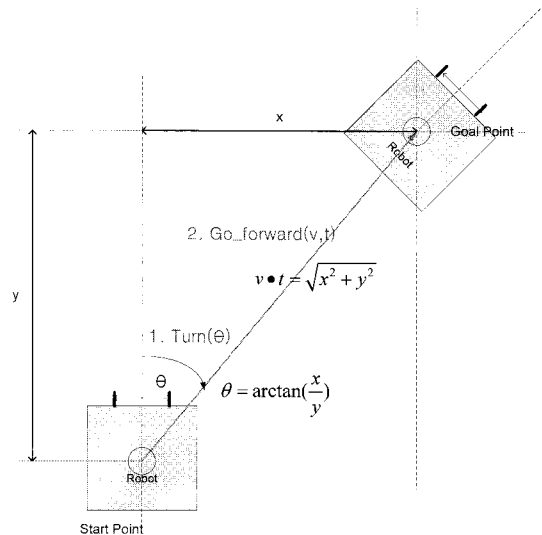


그림 14. Goto API.

Fig. 14. Goto API.

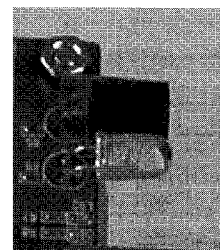


그림 15. 검은 색 튜브로 감싼 포토 트랜지스터.

Fig. 15. Black tube wrapped photo transistor.

하였다(그림15).

주행 중에 장애물이 발견되면, 즉 IR 센서를 통해 500mm보

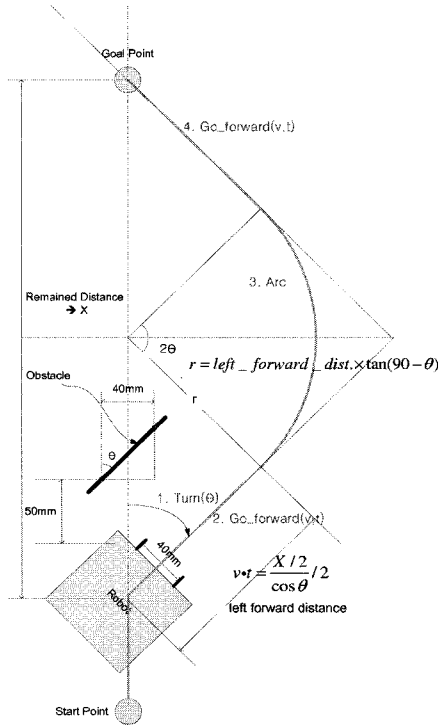


그림 16. 간단한 장애물 회피 주행.
Fig. 16. Goto API with simple obstacle avoidance.

다 가까운 거리에서 장애물이 인지되면 다음과 같은 단계로, 일단 정지한 후, 장애물 면의 각도만큼 회전 한 후(step 1), 목표지점까지 남은 거리를 고려하여 직진하고(step 2), 다시 목표 지점으로 향하기 위해 곡선 주행(step 3) 후 직진(step 4)으로 목표 지점에 도달한다(그림 16). 4단계를 수식으로 표현하면 다음과 같다.

Step 1: $\text{turn}(\theta)$,
where $\theta = \arctan(40[\text{mm}] / \text{difference between left and right sensor} [\text{mm}])$

Step 2: $\text{go_forward}(v, t)$, where $v \cdot t = \frac{X/2}{\cos \theta}$

Step 3: $\text{go_arc}(\text{left_velocity}, \text{right_velocity}, \text{arc_time})$,

여기서 $\text{radius} = \text{left_forward_distance} \cdot \tan(\frac{\pi}{2} - \theta)$,

$$\text{path}_{\text{right}} = (\text{radius} + 42.5[\text{mm}]) \cdot 2\theta,$$

$$\text{arc_time} = \frac{\text{path}_{\text{right}}}{\text{right_velocity}},$$

$$\text{path}_{\text{left}} = (\text{radius} - 42.5[\text{mm}]) \cdot 2\theta,$$

$$\text{left_velocity} = \frac{\text{path}_{\text{left}}}{\text{arc_time}}$$

Step 4: $\text{go_forward}(v, t)$, where $v \cdot t = \frac{X/2}{\cos \theta}$

위와 같은 장애물 회피 주행을 위해 구현된 프로그램에서 구현된 컴포넌트들 중 기본적인 주행에 쓰이던 모듈 외에 장애물 회피를 위한 모듈이 추가적으로 구현되었다.

주행 외에 로봇간의 통신을 위해 RF 트랜시버 칩으로 쓰이는 CC2420은 내부적으로 RSSI(Received Signal Strength Indicator)를 제공한다. RSSI는 RF의 Power를 나타내는 8 bits의

signed 2's complement의 값으로 이 때 실제 RF Power는 다음과 같은 식으로 나타난다.

$$P = \text{RSSI_VAL} + \text{RSSI_OFFSET} [\text{dBm}]$$

RSSI_OFFSET는 많은 테스트 결과 약 -45정도 된다. 따라서 RSSI값으로 -20이란 값을 읽으면 이는 RF 입력 Power가 약 -65 dBm이 된다는 뜻이다. 거리 비 RSSI값을 통계 내어 노드들 간의 일정 거리(예:1m)를 유지하도록 원하는 위치로 이동하는 알고리즘은 쉽게 구현 가능하나 거리 비 RSSI값이 정확하고 장애물에 의해 RF power가 줄지 않는 open된 공간 내에서만 실험한다는 가정 아래 실행할 수 있는 알고리즘으로 측정 실험 결과 RSSI값은 RF회로(예:RF 안테나)의 차이에 의해 아주 민감하게 차이가 발생하여 각 센서 노드마다 거리 대 RSSI 값이 달라지고 그 오차가 무시할 수 없을 뿐만 아니라 도중에 장애물이 발생하는 경우도 있을 수 있으므로 실제 공간에 투입될 노드에 들어갈 위치 추정 알고리즘으로 적절치 않았다. 센서 노드들을 배치할 때 거리보다 에너지소모 효율성이 더 중요하므로 RSSI값을 그대로 이용하면 항상 적절한 RF power가 도달하는 거리를 유지하여 RF 통신에 문제가 없도록 할 뿐만 아니라 좀 더 에너지 효율을 높일 수 있다. 따라서 heuristic한 알고리즘을 통해 간단한 어플리케이션을 구현하였다. 이 주행은 3단계로 나뉘게 되며 각 단계는 다음과 같다.

Step 1: 일정한 거리만큼 직진한다. (by go_forward)

Step 2: 각 노드들과의 RSSI변화량을 근거로 원하는 RSSI만큼 떨어진 위치방향으로 회전한다.

Step 3: 정해진 노드 간의 RSSI값에 해당하는 위치에 도달할 때까지 100ms마다 노드들 간의 RSSI값을 근거로 곡선 주행 혹은 직진 주행한다.

실제 구현된 코드에서는 RF 칩 관련 드라이버 모듈과 RSSI값을 활용하는 모듈이 추가적으로 구현되어 사용되었다.

이같이 RSSI를 이용하여 다음과 같은 어플리케이션들에 활용할 수 있다.

센서 노드들 간의 중복된 센싱 영역이 없다고 가정할 때, 하나의 노드가 고장 또는 작동 불량 상태가 되었을 때, 이를 해결하기 위해 모바일 센서노드를 통해 재배치한다(그림 22).

노드의 에너지가 줄거나 장애물이 발생하여 신호가 미약할 때 모바일 노드를 이동시켜 중요한 센싱영역을 보완한다.

일반적인 노드들 즉, leaf 노드는 고정 노드로 해당 cluster의 우두머리 역할을 하는 cluster head만 이동형 센서 노드로 구성할 때 cluster head노드의 이동으로 최적의 전력소모효율을 얻을 수 있다(그림 18, 19).

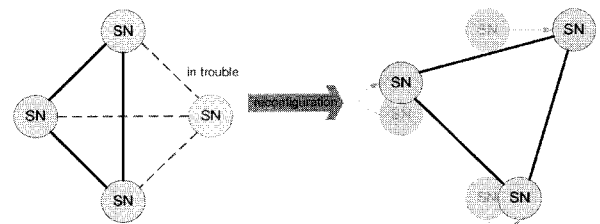


그림 17. 노드 재배치.
Fig. 17. Rearrangement of nodes.

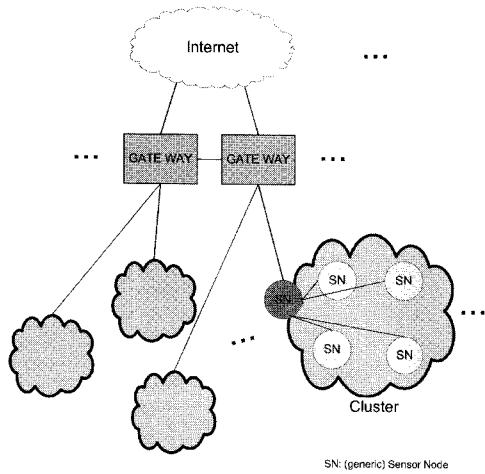


그림 18. Cluster head 이동 전의 노드 배치.
Fig. 18. Arrangement before movement of cluster head.

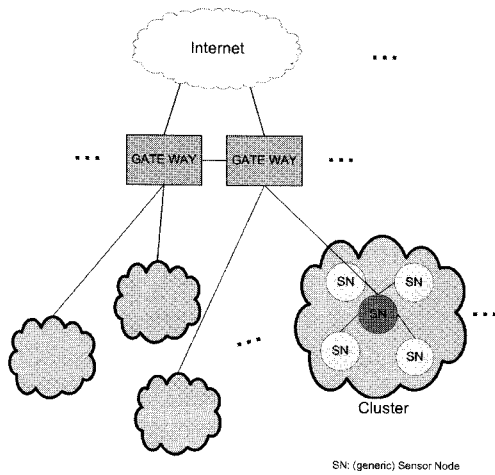


그림 19. Cluster head 이동 후의 노드 배치.
Fig. 19. Arrangement after movement of cluster head.

3. 에너지 효율성 측면을 고려한 구현 요소

주로 소모 전류를 줄이는 다음과 같은 방법으로 배터리의 교체 없이 최대한 긴 시간 동안 정상 동작 할 수 있도록 구현하였다.

- Voltage Converter: 1.8V에서 2.7V까지의 배터리 전원을 일정한 3.3V의 전원으로 바꾸어 공급함으로써 MCU의 clock을 일정하게 할 뿐만 아니라 정상적으로 동작 할 수 있는 시간을 늘려준다.
- Analog Switch: 모터 비구동시 총 120mA정도 소모되는 엔코더와 IR 센서 모듈에 들어가는 전원을 차단한다.
- Motor Driver: 모터 비구동시 모터 드라이버로 사용하는 LB1630을 standby 모드로 동작시켜 30μA이하로 소모되도록 한다.
- MSP430: active current가 2mA로 매우 적고, sleep current 또한 1μA로 전력소모 면에서 강력한 장점을 가지는 MCU를 선택하였다.

4. 최종 구현된 로봇의 사양

최종적으로 구현된 모바일 로봇 플랫폼의 사양을 기존에 연구되었던 플랫폼들과 비교하여 보면 표 7과 같다.

표 7. 다른 플랫폼과의 비교.

Table 7. Comparison with pre-developed robots.

Platform	CotsBots	Robomote	Micabot	Robot
Size(mm)	70 x 75 x 50	70 x 45 x 35	86 x 61 x 54	70 x 77 x 50
Cost	≈\$200	≈\$150	≈\$350	≈\$100
Radio	40 Kbps	19.2 Kbps	115 Kbps	250 Kbps
Max. active time	1 hour	3.5 hours	2.45 hours	≥5 hours
Max. sleep time	N/A	131 days	N/A	≥200 days
DAC	X	X	X	O

IV. 실험 및 결과

이 장에서는 본 논문의 주제에서 나와있듯이 얼마나 에너지효율적이고 경제적인 이동 로봇인지를 실험하여 결과로 실었다. 경제성은 로봇 구현에 소요되는 비용으로, 3장의 표 5로부터 알 수 있고, 에너지효율성은 여러 가지 어플리케이션을 실행시킨 후 각각의 전류 량을 제어 에너지 소모량이 다른 여타 로봇에 비해 적응을 보인다. 또한, 센서 노드의 역할 외에 이동성을 갖춘 로봇이라는 점에서 기본적인 주행 명령과 간단한 장애물 회피 수행 및 RSSI를 이용한 이동 어플리케이션 수행결과도 보인다.

1. 에너지 소모량 측정 및 비교

모터를 구동시킬 때는 측정되는 전류는 모터 드라이버에 들어가는 전류 외에도 인코더 및 IR 센서 부에 들어가는 전류도 포함된 양이다. 모터에 100% duty ratio의 PWM으로 구동시키면서 RF 송신 프로세스도 같이 실행되고 있을 때는 최대 7.79시간 동작 가능하고 sleep상태일 때는 최대 260.4 일까지 배터리 교체 없이 동작 가능하지만 실제로는 배터리의 효율이 떨어지므로 이보다 적은 시간 동안 동작 가능하다.

표 8. 전류량 및 작동 시간.

Table 8. Power consumption and operating time.

Mode	Power Consumption (in mW)	Days (2500mAh 2 AA기준)	
Sleep	1.03(0.4 mA)	260.4 days	
Stop	None	14.0 days	
	Radio Receive	105.52(31.2 mA)	3.3 days
	Radio Transmit	106.81(31.7 mA)	3.3 days
Move (2 motor -100% duty, 2IR,2ADC, encoder.)	w/o Radio process	8.43 hours	
	Radio Transmit	7.79 hours	
	Radio Receive	7.87 hours	

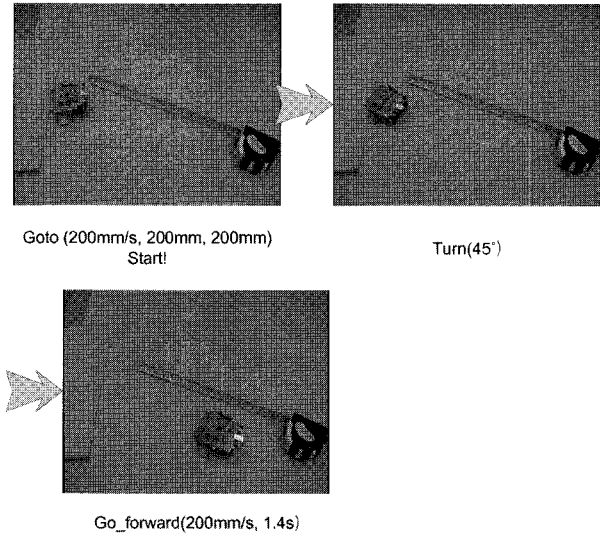


그림 20. Goto(200mm/s, 200mm, 200mm)의 실제 실행 모습.
Fig. 20. Experiment of goto API.

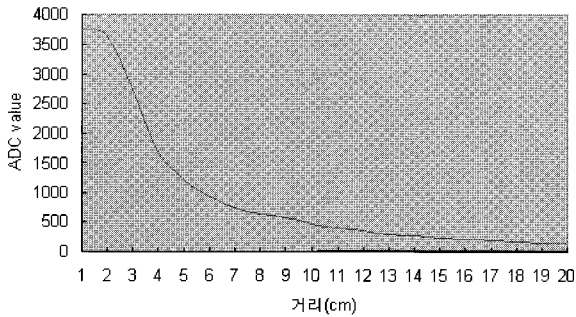


그림 21. 거리 vs. IR 세기의 ADC 값.
Fig. 21. Distance vs. ADC value of IR intensity.

2. 주행

간단한 주행 테스트를 위해 Goto(200mm/s, 200mm, 200mm)의 명령을 주었을 때 실행동작은 그림 20과 같다. 열 차례의 테스트 결과, 평균 이동 거리는 264mm이었고 표준편차 15.2mm이었다.

3. 간단한 장애물 회피

간단한 장애물 회피를 위해 IR센서모듈을 이용한다. IR diode에서 보낸 적외선이 반사되어 포토 트랜지스터로 들어온 양을 MCU의 ADC로 연결하여 값을 읽게 된다(그림 21).

12bits ADC를 이용하므로 3754의 값이면 포토 트랜지스터에서 얻은 출력전압의 값은

$$\frac{3754}{4096} \times (3.3(\text{high reference}) - 0(\text{low reference})) = 3.02V$$

가 된다. 간단한 장애물 회피 실험을 위해 Goto(200mm/s, 0mm, 1000mm)의 명령을 주고 주행 중에 장애물이 놓여있었다. 그림 16과 같이 실행되게 되고 실제 실행 모습은 그림 22와 같다.

4. RSSI신호를 이용한 어플리케이션

그림 23은 거리 비 RSSI값을 장애물이 없는 상태에서 고정 노드들로부터 테스트한 결과이다. 정지된 노드와의 RSSI측정임에도 불구하고 RSSI값의 변화가 크고 장애물에 민감하

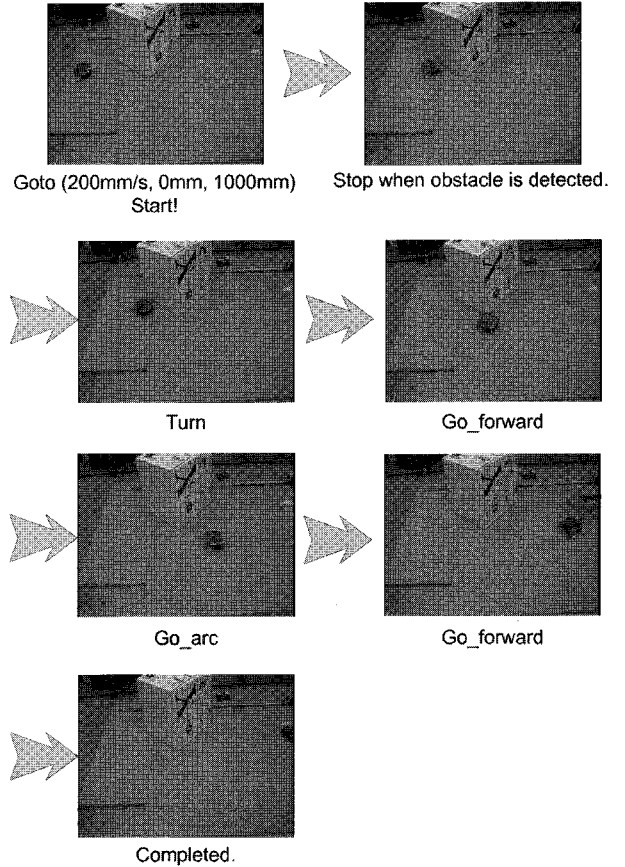


그림 22. 장애물 회피 주行的 실제 모습.
Fig. 22. Experiment of obstacle avoidance.

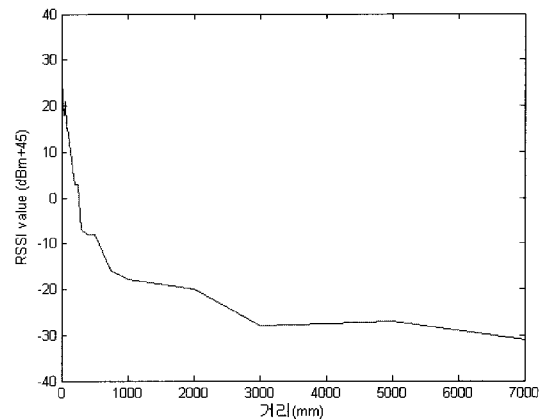


그림 23. 거리 vs. RSSI 값.
Fig. 23. Distance vs. RSSI value.

여 RSSI값만으로 거리 데이터로 이용하기에는 무리가 따른다는 것을 알 수 있다.

다음은 RSSI값을 거리로 변환하지 않고 RF power의 지표로만 삼아 이동 실험을 한 것으로 로봇 외의 추가적으로 센서 노드를 두어 테스트하였다.

2개의 노드와 각각 -25dBm의 세기를 가지는 곳으로 이동하는 어플리케이션을 구현하였다(그림 24). 두 개의 고정 노드(작은 노드)로부터 -25dBm의 RF power만큼 떨어진 곳으로 이동하는 어플리케이션으로 조금 전진하여 현재의 방향을

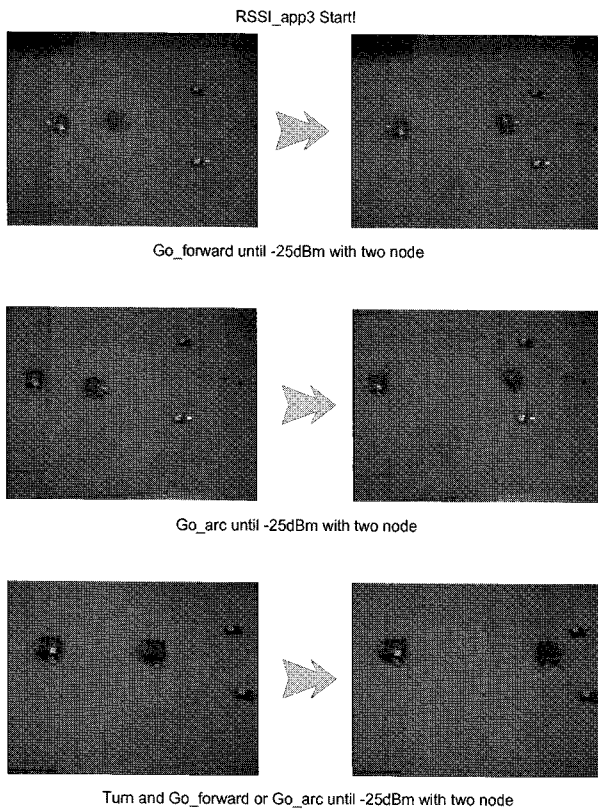


그림 24. 두 노드와 -25 dBm의 세기 만큼 떨어진 곳으로 이동하기(RSSI 어플리케이션).

Fig. 24. Experiment of RSSI applicaitoin.

인식한 다음 회전 후 직진이나 곡선운동으로 두 노드로부터 목표 RF power가 될 때까지 이동한다.

V. 결론

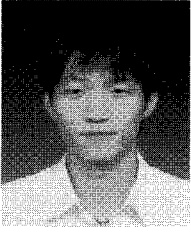
센서 네트워크 기술은 산업 전반에 걸쳐 활용되어 우리에게 유비쿼터스 시대를 열게 해줄 기술로 아직은 주로 무인 감시 및 정찰 등 군사적 목적에 이용될 목적으로 많이 개발되었으나, 사람이 접근하기 힘들거나 위험한 방사능 지역이나 독극물 유출 지역에서 지속적인 관찰이 필요할 때나 화재 감지 등에 활용될 수 있다. 이 같은 활용을 위해서 센서네트워크용 모바일 플랫폼은 여러 가지 면에서 제약조건을 지니게 된다. 한 번 배치되면 유지 보수가 어렵기 때문에 강한 구조에 저전력으로 설계되어야 한다. 광범위한 응용 분야에 사용되기 때문에 어떠한 구조에도 효과적으로 사용될 수 있도록 유연성과 모듈성을 갖추고 있어야 한다. 또한, 유비쿼터스 컴퓨팅 환경에 맞게 무선 네트워크 기술을 지원해야 한다. 많은 노드가 산재되기 위해 가격이 저렴하고 크기가 작아야 한다.

본 논문에서는 이동성 기능을 지닌 센싱노드를 활용하여 센서 네트워크에서 노드의 분산 알고리즘 연구 및 노드의 추가, 고장 시의 네트워크 재구성 연구 및 응용에 이용할 수 있기 위한 플랫폼으로써 통신 속도 및 기능성, 유연성 등의 여러 제약 조건을 만족시키는 비용효율적인 모바일 로봇을 구현하고 기본적인 주행과 RF통신 관련 어플리케이션의 타당성을 실험으로 보였다.

추후 연구과제로는 50개 정도의 다 개체의 로봇으로 센서 네트워크 알고리즘의 연구에 직접적으로 활용해보는 것이다. 실제로 노드들의 작업 수행을 위해 필수적인 알고리즘들(예: 라우팅 기능, USN, 통신 스택 등)을 추가했을 시에 예상되는 전력 소모까지 고려하여 이동 로봇의 전체적인 전력 소모를 최적화 시킬 수 있는 심도 있는 알고리즘의 개발이다. 전력소모는 이동 거리 외에도 통신 빈도나 시간, 거리 등에 대한 함수로 다양한 조건에서의 실험이 필요할 것이다. 그리고 현재 구현되어 있는 단순한 주행 API들과 RF관련 API들을 좀 더 확장하고 보완하여 현재 로봇의 기능성을 향상시키는 것이다. 그 한 예로 위치 추정을 위해 사용하기엔 부족한 RSSI신호 대신 초음파 센서를 장착하여 위치 추정을 하고 이에 따른 주행이 가능한 API들을 구현하는 일을 들 수 있다.

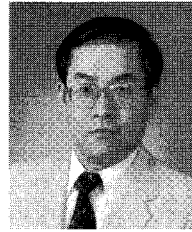
참고문헌

- [1] ActivMedia Robotics, *Pioneer 2 General Purpose Robot*, <http://www.activrobots.com/ROBOTS/p2dx.html>, 2008.
- [2] K-Team, *K-Team: Khepera II Minirobot*, <http://www.k-team.com/kteam>, 2008.
- [3] Pister KSJ Bergbreiter, S, "CotsBots: An off-the-shelf platform for distributed robotics," in *Intelligent Robots and Systems, IEEE/RSJ*, 2003.
- [4] Rahimi MH Sukhatme GS Sibley, GT, "Robomote: A tiny mobile robot platform for large-Scale Ad-hoc sensor networks," in *IEEE Robotics and Automation*, May 2002.
- [5] Goodwine B Montestruque LA McMickell, MB, "MICAbot: A robotic platform for large-scale distributed robotics," in *ICRA 2003*, 2003.
- [6] Paredis CJJ Khosla PK Navarro-Serment LE, Grabowski R, "Millibots," in *IEEE Robotics and Automation Magazine* v.9, n.4, Dec. 2002, pp. 31-40.
- [7] *MSP430, Ultra-Low Power Microcontrollers*, <http://www.ti.com/>, 2008.
- [8] Gutierrez JA Howitt I, "IEEE 802.15.4 low rate-wireless personal area network coexistence issues," in *2003 IEEE Wireless Communication and Networking* v.3, 2003, pp. 1481-1486.
- [9] *RGO-0331, micro geared motor*, <http://www.smallmotor.co.kr>, 2006.
- [10] *SI-5315D, Infrared Emitting Diode*, <http://www.auk.co.kr>, 2008.
- [11] *GP2D120, Sharp Distance Measuring Sensor*, <http://www.sharpsma.com/product>, 2008.
- [12] *TinyOS*, <http://tinysos.net>, 2008.
- [13] *Mantis*, <http://mantis.cs.colorado.edu>, 2008.
- [14] *PEEROS*, <http://www.eyes.eu.org>, 2008.
- [15] *Salvo*, <http://www.pumpkininc.com>, 2008.
- [16] *CC2420, IEEE 802.15.4 compliant and ZigBeetm ready RF Transceiver*, <http://www.ti.com>, 2008.
- [17] *FT232BM, USB UART I.C.*, <http://www.fidichip.com>, 2008.
- [18] *TPS61031, Efficient Synchronous Boost Converter*, <http://www.ti.com>, 2008.
- [19] *LB1630, Low-Saturation Bidirectional Motor Driver for Low-Voltage Applications*, <http://www.semiconductor-sanyo.com>, 2008.
- [20] *MAX4751, Low-Voltage, Single-Supply Quad SPST Analog Switches*, <http://www.maxim-ic.com>, 2008.



김 홍 준

1981년 6월 29일생. 2004년 한국과학기술원 전자·전산학과 졸업. 2007년 동대학원 전기전자전공 석사. 2007년~현재 동 대학원 전기전자전공 박사과정 재학 중. 관심분야는 이동 로봇, 센서 네트워크, 최적 제어.



김 병 국

1952년 10월 5일생. 1975년 서울대학교 전자공학과 졸업. 1975년 한국과학기술원 전기 및 전자 공학과 석사. 1981년 동 대학원 박사. 1981년~1986년 우진계기(주) 연구실장. 1982년~1984년 University of Michigan 방문연구. 1986년~현재 한국과학기술원 전기 및 전자공학과 교수. 관심분야는 실시간 시스템, 로봇틱스, 임베디드 제어.