

논문 2010-4-11

로봇환경플랫폼을 위한 직비 기반의 고성능 센서노드 구현

Implementation of a ZigBee-based High Performance Sensor Node for the Robot Environment Platform

김기민*, 이경중*, 김재오*, 문찬우**, 안현식***

Kee-Min Kim*, Kyung-Jung Lee*, Jae-O Kim*, Chanwoo Moon**, Hyun-Sik Ahn***

요약 환경 플랫폼에서 로봇이 주변 환경과 효과적으로 결합되어 주어진 임무를 수행하기 위해서는 로봇의 위치화(Localization) 및 내비게이션(Navigation) 기능이 필수적이다. 본 논문에서는 우선, 로봇 플랫폼용 고성능 센서 노드로 이루어진 무선 네트워크를 구성하여 다양한 센서 데이터의 고속 연산 및 전송이 가능하도록 하고 특히, 로봇의 위치화 기능을 구현한다. 저비용, 저전력 특성의 직비(ZigBee) 기반 센서 노드를 32 비트급 마이크로컨트롤러 중심으로 구성함으로써 다양한 센서 데이터의 실시간 처리 및 로봇 위치화 성능을 향상시킨다. Atmel사의 RUM(Router Under MAC)을 이용하여 32 비트급 직비 스택을 개발하고 이를 활용함으로써 고속의 센서 데이터 처리 및 위치화 기능이 가능한 고성능 센서노드를 구현하고 실제 실험을 통하여 성능을 확인한다.

Abstract In this paper, a ZigBee-based wireless sensor network is configured for the robot to effectively communicate with the environment platform where sensor nodes are implemented using high performance microcontrollers. The localization and the navigation functions are also required to the robot which performs the given task using various types of sensor information. A new type of ZigBee stack is developed using the RUM(Router Under MAC) of the Atmel Corp. and it is applied to a 32-bit ARM core microcontroller for the high performance sensor data manipulation and transmission. It is verified by experiments that the wireless sensor network consisting of developed high performance sensor nodes can be effectively used for the robot environment platform.

Key Words : ZigBee, Localization, RUM, Robot

1. 서론

로봇 환경 플랫폼은 로봇이 보다 지능적인 서비스를 수행할 수 있도록 로봇 주위의 환경이 로봇을 보조하고 지원하여 인간의 편리성을 극대화하는 방향으로 재정립하여 구축한 환경을 의미한다. 이러한 로봇 환경 플랫폼에서 동작하는 로봇은 요구되는 많은 정보를 로봇자체에

탑재되어 있는 내부의 센서 뿐만 아니라 로봇 외부의 환경에 설치된 다양한 종류의 센서로부터 정보를 획득할 수 있다. 또한 환경 내 설치되거나 기기에 내장된 구동기능 등을 활용함으로써 로봇은 주어진 목표 서비스를 더욱 효과적으로 진행할 수 있다^[1].

이러한 로봇 환경 플랫폼에서는 센서를 통해 외부 환경 정보를 로봇에게 전송하거나 이 외부 환경에 위치한 특정기와 상호 정보를 주고 받을 때 로봇은 무선 통신 네트워크가 필수적이 된다. 무선 통신 네트워크 기술에는 Bluetooth, HomeRF, 직비, UWB, Z-wave 등이 있다^[2-4]. 그 중 직비 기술은 저가, 저전력 구현을 가능하게

*준회원, 국민대학교 대학원 전자공학과

**정회원, 국민대학교 전자공학부

***정회원, 국민대학교 전자공학부(교신저자)

접수일자 2010.6.14, 수정일자 2010.7.16

게재확정일자 2010.8.13

하여 가정, 빌딩 및 공장 자동화 등을 목표로 개발된 근거리 저속 무선통신을 위한 기술로서 최대100m 이내의 거리에서 적용 가능하며, 한 개의 Piconet 당 255개의 무선기기들을 연결할 수 있다. 또한, 1mW의 낮은 출력에도 10m 거리 이내에서 무선통신이 가능하여 센서 네트워크로서 가장 가능성이 높은 기술이다. IEEE802.15.4에서 PHY와 MAC에 대한 표준을 정의하고 있으며 직비 Alliance에서 MAC 이상의 계층과 응용을 위한 프로필을 정의하고 있다.^[5]

이러한 직비 통신을 위해서는 직비 프로토콜을 운용하는 소프트웨어인 직비 스택 소프트웨어가 필요하다. 직비 통신에 사용되고 있는 여러 종류의 직비 스택 소프트웨어 중에서 펌웨어 영역에 적용이 용이하고, 메모리 사용을 줄이기 위해 독립적으로 MAC Data Layer만 구성함으로써 통신 프로토콜이 간결한 RUM(Router Under MAC)이 제안되었다.^[6]

기존의 직비 기반 센서 노드는 대부분 8-bit 마이크로컨트롤러와 RF 칩을 결합시켜 구현되며 이로 인해 각 노드에서 복잡한 신호처리가 요구되는 경우 처리 시간이 길어지게 된다. 이것은 직비 통신의 전송속도 및 통신 신뢰성에까지 영향을 주게 되므로, 각 노드 내 마이크로컨트롤러를 32비트급으로 대체한 고성능 센서노드 개발이 필요하게 되었다. 그러나 현재까지 32비트급 마이크로컨트롤러용 직비 스택이 적용된 예는 거의 없어 이에 대한 개발 및 구현이 시급하다.

또한, 센서네트워크 내에서의 위치화는 무선통신 네트워크 분야에서 매우 중요한 부분 중에 하나로 인식되고 있고, 실제 구현과 실험이 꾸준히 진행되고 있으며, 대부분의 연구는 위치추정 정확도와 같은 성능적인 측면에 초점을 맞추고 있다.^[7-9] 이러한 연구들 중 GPS와 같은 기존의 위치 추적 서비스는 실내 환경에 부적합한 특성을 보이므로 실내 환경에 적합한 위치 추적 시스템인 ZigBee기반의 위치 추적 시스템을 개발하려고 연구 중에 있다.^[10] 따라서 센서 노드의 신호처리 능력을 높이면서 동시에 위치화 알고리즘과 제어 알고리즘을 수행할 수 있도록 하기 위하여 위치화 알고리즘인 WCL(Weighted Centroid Localization)을 사용하여 실제 구현을 위한 API들을 작성하고자 한다.

본 논문에서는 우선 기존의 8-bit 마이크로컨트롤러 Atmega128 전용으로 제안되어 있는 RUM API(Application Programming Interface)를 32-bit 마이크로

컨트롤러인 Cortex-M3에서 동작하도록 API 함수들과 레지스터 등을 변환하는 작업을 통하여 직비 스택 소프트웨어를 구현한다. 또한 로봇 환경 플랫폼 내에서의 각각의 노드들은 빠른 센서 데이터 처리 및 알고리즘 수행을 위하여 32-bit 마이크로컨트롤러와 직비 모듈로 구성하고, 이와 같이 구성된 환경에서 직비 모듈만으로 위치화가 가능한 것을 실험을 통하여 보인다.

II. RUM 기반 통신 스택 개발

1. RUM 구성 및 특징

RUM(Router Under MAC)은 Atmel사에서 제공하는 802.15.4 MAC Layer에 호환되는 통신 스택이다. RUM의 장점으로는 작고 단순하고 Multi-Hop을 지원하며 Self-Forming 네트워크를 지원 한다. 또한 RUM은 오픈 프로토콜이므로 Atmel 하드웨어를 사용하면 무료로 사용이 가능하다.

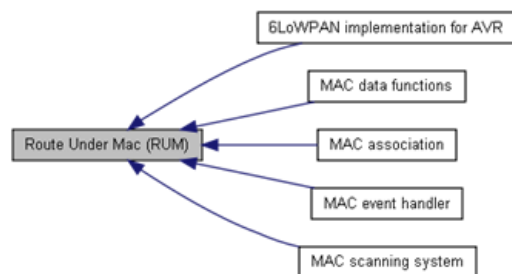


그림 1. Router Under Mac (RUM) 모듈 구성
Fig. 1. Configuration of the Router Under Mac (RUM) Module.

그림 1은 RUM의 구조에 대한 그림이다. RUM의 구조는 크게 5개의 모듈로 MAC association, MAC data functions, MAC event handler, MAC scanning system 와 옵션인 6LoWPAN으로 구성되어 있다. MAC data functions는 데이터의 송수신을 담당하고 있다. 노드 간의 데이터의 전송을 감지하면 MAC data functions는 데이터를 저장하고 시스템에 알려준다. MAC even handler는 타이머모듈을 사용하여 데이터의 이벤트를 감시하거나 새로운 이벤트를 발생하는데 사용된다. macTask() 함수는 메인 루프 또는 OS의 프로세스 상에 선언되어 계속적으로 이벤트를 감시한다.

MAC scanning system의 역할은 채널의 상태를 스캔하여 최적 채널을 Coordinator에 알려주고 또한 각 End 노드나 Router에서 Beacon 요청이나 Coordinator의 Beacon 프레임 전송을 담당한다. MAC association은 Coordinator노드와 Router/End 노드 간의 간을 연동하는 함수들을 포함하고 있다. MAC scanning system에서 찾아진 최적의 채널 등의 정보를 활용하여 실제적으로 네트워크를 구성하기 위한 기능을 제공한다.

RUM에서는 노드가 초기화되면 MAC scanning system 모듈의 함수가 작동하여 최적 채널 정보와 Beacon을 노드 간 교환하며 MAC association 모듈에서 네트워크를 구성한 다음 MAC data functions에서 데이터 흐름을 다룬다. 마지막 모듈은 6LoWPAN이며 이 모듈은 옵션이고 본 논문의 스택에서는 사용하지 않는다. RUM은 기본적인 무선 모듈의 전원 파워 등을 제어하는 프로그램 뿐 만 아니라 Coordinator를 위한 전 채널을 검색하여 최적의 채널을 검색하는 에너지 스캔과 옵션인 IPV6 패킷을 이용하는 6LoWPAN을 지원한다.

본 논문에서는 위와 같이 프로그램의 크기가 작아서 부담이 적고, 다루기 편하며 Multi-Hop을 지원하기 때문에 RUM Stack을 선택하였다. RUM은 기본적으로 8-bit 인 ATMEGA 1281 마이크로 컨트롤러 전용으로 제작되어 있으며 일부만 ARM을 지원한다. 따라서 HAL(Hardware Abstraction Layer)을 새로 제작해야 하며, 추가된 API와 변환 과정은 다음 장에서 설명하도록 한다.

2. 32-bit 마이크로컨트롤러용 RUM API 제안

기존의 RUM의 계층 구성은 그림 2(a)와 같다. 기존의 RUM 계층을 수정하여 그림 2(b)와 같이 제안한다. 모든 스택들을 ARM에 포함시켜서 센서 및 액츄에이터 제어 기와의 직비 통신 기술을 구현한다. ARM은 HAL을 통하여 통신 모듈을 제어한다. 기존에 제공된 RUM에서도 기본적인 ARM HAL을 제공하지만 기능이 AVR RUM API보다 제한적이며 Coordinator의 API만 제공하므로 수정이 필요하다. 따라서 HAL을 재설계 하고 각 모듈에서 ARM에 적용되는 기능을 추가하여 Router와 End노드를 지원하도록 한다.

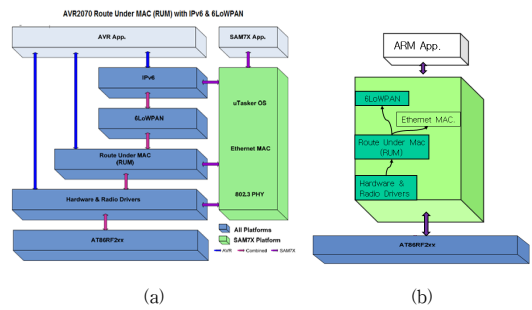


그림 2. RUM API 계층 구성도 (a) 기존 API, (b) 제안하는 API
Fig. 2. RUM API layout (a) Existing API, Proposal API.

32-bit 마이크로컨트롤러용 HAL은 이미 주어진 ARM HAL소스를 실험에 사용할 Cortex에 구현되도록 변환되었다. 라디오(Radio) 드라이버는 RUM API에서 제공하므로 추가적으로 변환 작업은 필요하지 않았다. HAL의 주된 기능은 SPI통신을 통해 모듈과 통신을 하여 모듈 내부의 레지스터 값을 변경하거나 값을 읽어오는 것이다. HAL은 총 3개의 소스로 구성되어 있는데, 그 중 SPI 통신에 관련된 부분은 HAL_arm.c에 정의 되어 있으며 이 파일은 8개의 함수로 구성되어 있다. 그러므로 HAL은 SPI 통신 관련 함수 중 가장 중요하며, 그림 3은 라디오의 레지스터에 값을 적는 함수를 제작한 예이다.

```
void hal_register_write(u8 address, u8 value)
{
    // Add the Register Write command to the address.
    address = HAL_TRX_CMD_RW | (HAL_TRX_CMD_RADDRM & address);

    // Pull SEL line.
    SPI_FLASH_CS_LOW();
    // Transmission is started by writing the transmit data.
    SPI_FLASH_SendByte( address ); // write
    // Transmission is started by writing the transmit data.
    SPI_FLASH_SendByte( value );
    // Release SEL line.
    SPI_FLASH_CS_HIGH();
}
```

그림 3. hal_register_write 함수
Fig. 3. hal_register_write function.

HAL의 나머지 부분은 타이머에 관련된 파일이며 이 두 개의 파일은 HAL_timer.c 와 HAL_timer_event.c 로 이루어지며 MAC event handler 모듈과 연동된다. HAL timer는 마이크로컨트롤러의 타이머 드라이버이며 event 는 타이머에 관련된 이벤트 발생을 담당한다. 또한 API 구조를 변경하여 기존의 RUM에서 사용하지 않는 함수들을 정리하였다.

III. 위치화 가능 노드 설계 및 구현

위치화 알고리즘에서 가장 중요한 점은 위치가 알려지지 않은 노드의 위치를 산출하는 것이다. 임의의 노드의 위치를 알아내는 유용한 방법으로 들어오는 Received Signal Strength(RSS)를 이용하여 거리를 산출하는 방법이 있다. 이상적인 RSS는 전송 장치의 전송 파워(P_{TX})를 직접적으로 반영하고, 수신 장치의 수신파워(P_{RX})에 영향을 준다고 가정한다. Friis의 자유공간 전송방정식(free space transmission equation)에 따르면, 검출된 신호 세기는 전송자의 거리의 제곱에 관계되어 있다^[11].

$$P_{RX} = P_{TX} \cdot G_{TX} \cdot G_{RX} \left(\frac{\lambda}{4\pi d} \right)^2. \quad (1)$$

P_{TX} = Transmission Power of Sender

P_{RX} = Remaining Power of wave at receiver

G_{TX} = Gain of transmitter

G_{RX} = Gain of receiver

λ = WAVE length

d = Distance between sender and receiver

임베디드 장치에서 RSS는 RSSI(Received Signal Indicator)로 변환해서 생각할 수 있다. RSSI는 레퍼런스 파워 P_{ref} 대비 수신 파워로써 정의되어 있다.

$$RSSI = 10 \cdot \log \frac{P_{RX}}{P_{REF}} [RSSI] = dBm \quad (2)$$

하지만 실제적으로 거리 측정에 RSSI를 사용하는 방법은 문제를 가지고 있다. 그 이유는 아래와 같이 여러가지 요인이 RSSI의 값에 영향을 끼치고 또한 RSSI 값이 급격히 줄어들도록 작용하기도 한다.

- 금속 물체에 반사
- 전자기장에 의한 감소
- 모서리에서의 회절
- 주변 전파의 간섭 (WIFI)등

그러므로 실제 간섭에 의해 발생하는 오차를 줄이기 위해서 LQ(Link Quality)를 이용하여 보정을 하였다. IEEE 802.15.4에 따르면 LQI는 수신되는 패킷의 퀄리티로 신호의 특성을 표현한다. 그러므로 LQI를 사용하면 신호 대 잡음 비를 계산할 수 있고 이것을 RSSI를 조합

하여 거리를 측정하면 RSSI에서 발생된 간섭에 의한 영향을 최소화 할 수 있다.아래의 그림 4는 RF230 모듈에서의 패킷 에러율 대비 LQI의 관계를 보여준다^[12].

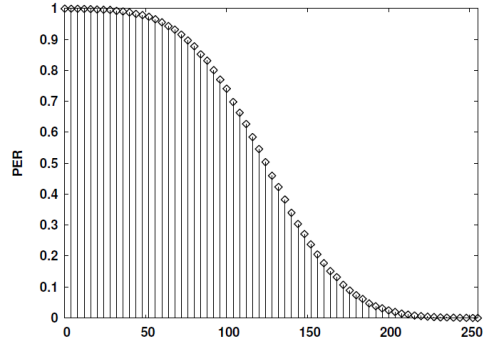


그림 4. 패킷 에러율 대비 LQI 관계 그래프
Fig. 4. Conditional Packet Error Rate Versus LQI.

본 논문에서는 WCL 알고리즘을 적용한다. 앞에서 구한 거리 비론과 임의의 노드 P와의 거리를 d 라고 하고 실험적으로 얻은 테스트 환경에 적합한 인자 g 를 이용하여 무게 함수(weight function) w_{if} 를 구한다. 여기서 w_{if} 는 거리와 센서 노드의 수신기의 특성에 관련된 함수이다. 모든 어플리케이션의 시나리오는 환경이 바뀌기 때문에 다른 무게를 요구한다. 각 비론에 대해 구해진 무게 함수를 이용하여 노드 P를 구하면 아래의 식과 같다.

$$P_i''(x, y) = \frac{\sum_{j=1}^n (w_{if} \cdot B_j(x, y))}{\sum_{j=1}^n w_{if}}, \quad w_{if} = \frac{1}{(d_{if})^g}. \quad (3)$$

d_{ij} - beacon B_j 와 센서노드 P_i 간의 거리, g - degree

IV. 실험 결과

1. 실험 시스템 구성

실험은 ARM의 최신 Core인 Cortex 제품군 중 Cortex-M3를 선택한다. Cortex-M3는 기존의 ARM7TDMI를 대체하기 위해 만들어진 코어로 기존의 제품보다 저전력이며 동작 속도도 동일 클럭 기준으로 30~40%의 성능이 빠르다^[13]. 본 논문에서는 Cortex-M3

를 메인 코어로 사용하는 칩 중 STMicroelectronics 사의 STM32F103을 선정한다.

환경 정보를 보내주는 센서 노드는 칩 사이즈가 되도록 작고 ARM7계열의 칩보다 저 전력인 STM32F103 칩을 선정한다. RUM을 사용하기 위해서는 Atmel사의 제품을 사용해야 하기 때문에 Atmel사의 직비 모듈 중에서 수신 감도가 3dBm으로 상대적으로 강한 AT86RF230을 사용하여 통신 모듈로 구성한다.

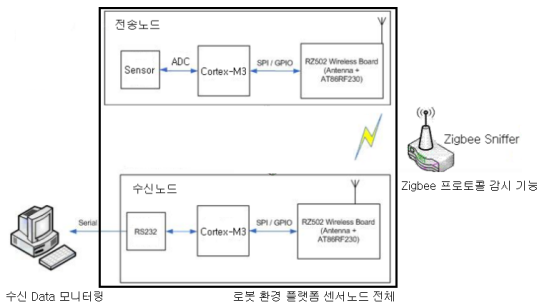


그림 5. 실험환경 구성도
Fig. 5. Configuration of Experimental Systems.

AT86RF230과 STM32F103은 SPI를 사용하여 연결하고 SPI클럭은 AT86RF230이 최대 8Mbps까지 지원하나 통신의 안정성을 위하여 4.5Mbps로 설정 한다. 이와 같이 구성된 실험환경은 그림 5와 같다.

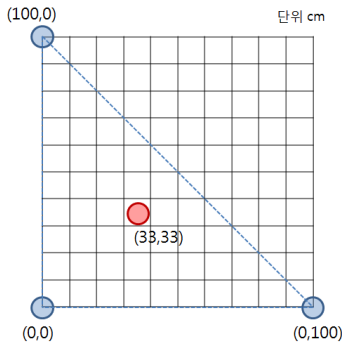


그림 6. 위치화 성능 분석을 위한 로봇 및 비콘 노드 위치
Fig. 6. Position of a robot(moving node) and beacon nodes for analyzing localization performance.

또한, 위치화를 확인하기 위하여 거리 1m 간격으로 beacon을 직각 이등변 삼각형으로 3개 배치하였다. 지정된 위치에 센서노드를 배치하여 임의의 이동로봇이 환경

내로 들어온 것으로 가정하였다. 테스트의 편의를 위하여 임의의 좌표는 직각 이등변 삼각형내의 (33, 33)좌표에 배치하였다. 그림 6은 beacon의 위치와 센서노드의 위치를 도식화 한 것이다.

2. 실험과정

32bit Cortex-M3기반의 ZigBee통신은 RUM 스택을 거쳐 전송노드의 AT86RF230 직비 모듈로 SPI통신을 통해 전송하고, 직비 프로토콜을 이용한 무선네트워크를 통하여 수신노드의 AT86RF230 직비 모듈로 데이터가 수신되며 다시 SPI통신을 통하여 수신노드의 Cortex-M3에 데이터를 저장하는 과정을 거친다. 이와 같이 통신되는 과정 중 노드간의 신호 세기를 측정 한 RSSI값을 데이터의 편차를 줄이기 위하여 마이크로컨트롤러에서 8번 측정 한 후 합산하여 평균값을 지속적으로 산출한다. 산출되어진 RSSI의 평균값을 이용하여 앞서 말한 무계함수 알고리즘을 통해 연산된 좌표 값을 위치화 하고자 하는 노드 내 Cortex-M3의 RS-232 시리얼 통신을 통해서 로컬 컴퓨터로 전달함으로써 연산된 좌표를 모니터링 할 수 있도록 한다. 통신과정 및 결과를 확인하기 위하여 Atmel사의 직비 통신 분석기인 직비 Sniffer를 활용해서 통신 Stack의 단계별로 확인한다.

위치 정보를 기록하기 편하기 위하여 하이퍼 터미널을 이용하여 거리 정보를 측정할 수 있도록 시리얼 기반의 프로그램을 제작하여 실제 위치 정보와 측정된 결과를 비교 하였다.

3. 실험결과

전송노드에서 전송한 데이터가 수신노드에 정상적으로 수신되는 것을 직비 Sniffer의 모니터링 프로그램을 통하여 확인할 수 있다. 네트워크상에서 2대의 모듈은 우선 Beacon을 교환한 후 Association 과정을 진행하고 있음을 그림 7을 통해 확인할 수 있다.

Seq No	Channel	Time	Time Delta	Src PAN	Src Sec	Dest PAN	Dest Sec	Src...	Packet Type
5	12	12:36:50.472	+0:00:00.003	0x5567	0x0000	0x5567	0x0000	0x00	Command: Beacon Request
6	12	12:36:50.890	+0:00:00.008	0x5567	0x0000	0x2211ff5ed50bba	0x3567	0x00	Beacon: Src: 15, Src: 15, Pk: 1, SF: 0
7	12	12:36:51.087	+0:00:00.004	0x2211ff5ed50bba	0x3567	0x0000	0x00	Command: Association Request	
8	12	12:36:51.888	+0:00:00.002	0x0000	0x0000	0x2211ff...	0x00	Command: Association Response	
9	12	12:36:52.194	+0:00:00.008	0x0000	0x0000	0x5567	0x00	Association Response	
10	12	12:36:51.896	+0:00:00.001	0x0000	0x0000	0x5567	0x00	Association Response	
11	12	12:36:54.170	+0:00:00.273	0x0000	0x0000	0x5567	0x00	Data	
12	12	12:36:54.171	+0:00:00.001	0x0000	0x0000	0x5567	0x00	Association Response	

그림 7. 직비 Sniffer (모니터링 프로그램)
Fig. 7. ZigBee Sniffer (Monitoring Program)

또한 시리얼 통신으로 데이터를 출력하는 모니터링

프로그램을 개발하여 위치화 알고리즘의 동작을 검증하였다. 임의의 노드는 각 비콘 노드에서 전송받는 비콘 신호를 통하여 비콘 노드까지의 거리를 산출 하였다. 임의의 노드는 비콘 노드까지의 거리 정보를 통하여 위치화를 진행 하였다. 아래 표 1는 위의 모니터링 프로그램에 의해 수신된 측정된 데이터 중 33개의 데이터를 수집하여 실제 좌표와 측정된 좌표의 거리 차를 연산하여 실제 위치와의 오차를 산출한 표이다.

표 1. 실험 데이터 및 실험 오차
Table. 1. Experimental Data and Error.

X축 측정	Y축 측정	실제위치와 측정된 위치의 거리차이
33	33	0
33	33	0
33	33	0
33	33	0
33	33	0
33	33	0
30	39	6.708203932
22	54	23.70653918
22	54	23.70653918
33	46	13
36	27	6.708203932
29	40	8.062257748
29	40	8.062257748
32	44	11.04536102
33	33	0
29	29	5.656854249
29	29	5.656854249
33	33	0
33	33	0
33	33	0
33	33	0
36	27	6.708203932
44	23	14.86606875
46	33	13
38	38	7.071067812
40	29	8.062257748
39	39	8.485281374
36	36	4.242640687
33	33	0
33	33	0
36	26	7.615773106
29	40	8.062257748
36	26	7.615773106
평균 거리		6.001284712
오차		8.487098232
비고		약 8.5%의 오차를 나타냄

실험과정에서 확인한 위치좌표는 그림 8의 모니터링 프로그램에서 확인 할 수 있다. 그림 8에서 px, py는 위치를 알고자 하는 노드의 x, y 좌표 값이며 단위는 cm이다.

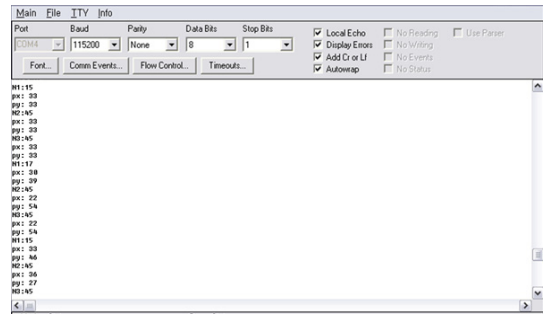


그림 8. PC에서 확인한 위치 좌표 값
Fig. 8. Confirmed Coordinate values on a PC

V. 결론

본 논문에서는 로봇 환경 플랫폼의 무선 네트워크 구성에 효과적으로 적용할 수 있는 고성능 센서노드를 개발하여 센서 데이터 처리능력 및 연산속도를 향상시켰고, 위치 추적 알고리즘을 적용하였다. 32-bit급 고성능 센서 노드에는 RUM API 변환 작업을 통하여 생성된 직비 스택이 구현되어 정상 동작함을 확인하였으며, 센서노드에 동시에 구현된 WCL 알고리즘에 의해 약 8.5%의 적은 오차로 로봇 위치 추적이 용이하게 수행됨을 확인하였다.

참고 문헌

- [1] 이경중, 김기민, “로봇 환경 플랫폼 구성을 위한 ZigBee기반 센서노드 구현”, 2010 정보 및 제어 학술대회, pp.141~142, 2010
- [2] 장동현, 현종웅, 김래근, “홈 네트워크 국내외 동향 및 발전 전망”, 한국정보처리학회지, 11권 3호, pp.3~11, 2004
- [3] 조성선, 이재환, “블루투스 기술 및 시장 동향”, 2001
- [4] 김창환, “UWB 무선통신 기술 동향”, 주간기술동향, 통권 1345호, 2008
- [5] ETRI, “홈 네트워크 기술 및 시장동향”, 2005
- [6] Atmel, “AVR2070:Toute under MAC(RUM) with IPv6 and 6LowPAN”
- [7] D. Bouvet, G. Garcia, “Improving the accuracy of dynamic localizationsystems using RTK GPS by identifying the GPS latency”, Proceedings of the

- 2000 IEEE, International conference on Robotics & Automation, 2000
- [8] Z. Xiang, S. Song., J. Chen, H. Wang, J. Huang, and X. Gao, "A wireless lan-based indoor positioning technology," IBM Journal on Research and Development, pp.617~626, September 2004
- [9] M. Wallbaum and S. Diepolder, "Bebchmarking Wirless Lan Location Systems," The 2nd IEEE International Workshop on Mobile Commerce and Service(WMCS'05), pp. 42~51, July 2005
- [10] J. Hightower and G. Borriello, "Location systems for ubiquitous computing", IEEE Computer, Vol.34, No.8, pp.57-66, August 2001
- [11] J. Blumenthal, R. Grossmann and F. Golatowski, "Weighted Centroid Localization in ZigBee-based Sensor Networks", Intelligent Signal Processing, 2007. WISP 2007, pp. 1~6, IEEE International Symposium, 3-5 October 2007
- [12] Atmel, "AT86RF230 PRELIMINARY", 2007
- [13] Shyam Sadasivan, "An Introduction to the ARM Cortex-M3 Processor", 2006

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업 (NIPA-2010-C1090-1021-0005)과 국민대학교 교내 연구비 지원으로 수행되었음.

저자 소개

김 기 민(준회원)



- 2009년 국민대학교 전자공학부 학사 졸업
- 2009년~현재 국민대학교 대학원 전자공학부 석사과정
- <주관심분야 : USN 기술, 로보틱스 응용>

이 경 중(준회원)



- 2010년 국민대학교 전자공학부 학사 졸업
- 2010년~현재 국민대학교 대학원 전자공학부 석사과정
- <주관심분야 : USN 기술, 로보틱스 응용>

김 재 오(준회원)



- 1994년 국민대학교 전자공학부 학사 졸업
- 1996년 국민대학교 전자공학부 석사 졸업
- 2008년~현재 국민대학교 대학원 전자공학부 박사과정

<주관심분야 : USN 기술, 로보틱스 응용>

문 찬 우(정회원)



- 2001년 서울대학교 전기컴퓨터 공학부 박사
- 2002년 ~ 2006년 전자부품연구원 선임, 책임 연구원
- 2006년 ~ 현재 국민대학교 전자공학부 교수
- <주관심분야 : 로보틱스 응용>

안 현 식(정회원)



- 1992년 서울대학교 제어계측 공학과 박사
- 2002년~현재 대한전기학회 정보 및 제어부문 이사
- 1993년~현재 국민대학교 전자공학부 교수

<주관심분야 : 지능형 로보틱스 및 차량 전자 제어>