



한국정보처리학회

1권 제1호
I. 11 No. 1



2009

한국 소프트웨어공학 학술대회 논문집

Proceedings of 2009 Korea Conference on
Software Engineering

- 일시 : 2009년 2월 9일(월)~11일(수)
- 장소 : 한화 휘닉스 파크(강원도 평창)

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학 소사이어티
 한국정보처리학회 소프트웨어공학 연구회
 한국전자통신연구원, 한국소프트웨어진흥원
 POSTECH 정보통신연구소, (주)비트컴퓨터
 KAIST 소프트웨어 프로세스개선센터

후원 : 국방기술품질원, (주)엔쓰리소프트, STA컨설팅(주)
 인피언컨설팅(주), (주)모아소프트

2. IT융합기술의 미래 트렌드	83
손승원 소장 (한국전자통신연구원 융합기술연구부문)	
3. TSP for Vitalizing Software Organization with Coaching	84
Yoshihiro Akiyama 교수 (Kyushu Institute)	

2009. 2. 10 (화)

B1: SW 설계 및 아키텍처 II

1. 소형 다관절 로봇 운용 소프트웨어 경량화 구조 설계	85
손현승, 김우열, 서채연, 김동호, 류동국, 이종섭, 김영철(홍익대)	
2. 비즈니스 어플리케이션을 위한 분석모델 자동 생성 패턴	89
박수진, 박수용(서강대)	
3. 비용을 고려한 아키텍처 평가를 지원하는 소프트웨어 아키텍처 비용 추정 기법	97
조시호, 낭종호, 이준하, 박수용(서강대)	
4. 기존 시스템 기반의 소프트웨어 제품라인 공학 기법:케이블 셋톱박스 소프트웨어 사례 ..	99
최현식, 조호진, 조운호, 이혜선, 강교철 (포항공대), 이상은, 이혁재, 송주영 (KIPA)	

B2: 소프트웨어 응용 II

1. 오픈소스 프로젝트를 위한 가상의 컴파일러머신 모델제안	101
허형(상명대)	
2. 모바일 UI 명세를 이용한 모바일 응용 S/W 시뮬레이션	107
정재원, 최윤석, 양진석(포항공대)	
3. AUTOCL: OCL을 이용한 AUTOSAR 구문 검증 도구	115
임형주, 박사천, 권기현(경기대)	
4. 내장형 S/W 개발 시 명령어패턴을 이용한 모델 기반의 에너지 소모 예측 기법	122
임형인,한아림,전상욱(KAIST), 배두환(KAIST), 홍장익(충북대)	

B3: 임베디드 소프트웨어

1. 병렬처리 하드웨어 가상화에 관한연구	130
이희덕(삼성탈레스)	
2. 디지털 컨버전스의 전개가 통신/방송 산업에의 영향에 관한 연구	138
김문구, 박종현, 백종현(ETRI)	
3. 모바일 서비스 API 설계 방법 구축과 적용사례	147
김수진, 박찬진, 은나래(LG전자)	
4. 실시간 통합 지식관리 체계 구축에 관한 연구	158
전승수, 정경진(한국과학기술기획평가원), 이병수(인천대)	

박사학위 논문발표

1. 기능점수 도출 기법을 이용한 요구사항 기반 비용분석 및 관리 방법 160
최순황(삼성전자)
2. A Web Content Adaptation System based on Personalization Technologies 162
이승화(성균관대)
3. 서비스 기반 아키텍처에서 서비스의 재사용성 평가를 위한 프레임워크 164
최시원(숭실대)

232

C1: SW 설계 및 아키텍처 III

1. 서비스 기반 적응형 홈 네트워크 시스템을 위한 통합 모델링 166
한종대, 심재근(서울대), 이병정(시립대), 이형원,(강릉대) 우치수(서울대)
2. 2+1 View 메타모델 기반 PIM/PSM 컴포넌트 모델링 방법 174
송치양(경북대),조은숙(서일대)
3. 소프트웨어 공학 기반의 지식기반 네트워크 서비스 온톨로지 모델링 방법 184
김수경, 이상훈, 최호진(ICU)
4. 소프트웨어 신뢰도 분석을 위한 패트리넷 기반 모델링 방법론 192
이준훈,윤현상,이은석(성균관대)

236

C2: 정형 기법

1. 상황인식 오류 예방을 위한 가전제품 사용자 인터페이스의 정형분석 200
최윤자, 장훈, 이주형(경북대), 유희천, 김문성, 윤성혜(포항공대)
2. 소프트웨어 모델 검증을 이용한 비즈니스 프로세스의 정형 검증 208
이태훈, 권기현(경기대)
3. OCL 단순화를 통한 질의 처리 개선 214
박호진, 홍승안, 권기현(경기대)
4. Statechart를 이용한 CC 3.1기반 EAL6 ADV_TDS 개발 문서 작성 222
정연오, 이혁, 김현석, 최진영(고려대)

240

C3: 품질 및 테스트 I

1. 테스트 프로세스 개선을 통한 테스트 품질 향상 사례 230
한종희, 김창배, 김태호, 이정현, 배성한(파이오링크), 권원일(STA)
2. 국내 소프트웨어 품질비용 사례에 관한 연구 241
이상은, 이혁재, 배문식, 이에복(KIPA)
3. 소프트웨어 품질비용 리포지토리 구축 연구 248
양해술,이하용(호서대), 이상은,이혁재,김도형(KIPA)

D1: SW 설계 및 아키텍처 IV

1. 임베디드 소프트웨어 모델 요소의 에너지 프로파일링을 통한 기반 라이브러리 구축 및 활용 263
김두환, 홍장익(충북대)

소형 다관절로봇 운용 소프트웨어 경량화 구조 설계

손현승, 김우열, 서채연, 김동호, 류동국, 이종섭, 김영철

홍익대학교 컴퓨터정보통신공학과 소프트웨어공학 연구실
 {son, john, jyun, kdh, ryu, ijs, bob}@selab.hongik.ac.kr

요약: 소형 다관절로봇은 펌웨어로 시스템을 개발한다. 그러나 펌웨어는 작업량과 하드웨어 제어루틴이 많아지면 개발이 어렵다. 이런 펌웨어의 한계 때문에 RTOS(Real Time Operating System)를 사용하지만 소형 다관절로봇에는 바로 적용하기 어렵다. 하드웨어 용량이 낮아 탑재가 어렵고 제어를 위해 최적화 되어 있지 않기 때문이다. 본 논문에서는 이런 문제를 해결하기 위해 소형 다관절로봇 운용을 위한 경량화 구조를 설계한다. 제안한 설계는 기존 RTOS의 태스크 상태를 초경량화 하여 소형 다관절로봇에 맞게 최적화 한다. 적용사례로 제안한 소프트웨어를 6족로봇에 탑재하여 그 동작을 확인한다. 결과적으로 제안한 소프트웨어로 소형 다관절로봇을 효율적으로 운용 가능하였다.

핵심어: 다관절 로봇, RTOS(Real Time Operating System), 로봇 제어, RTOS 경량화, 경량화 구조 설계

1. 서론

기존의 소형 다관절로봇[1, 2]은 펌웨어로 시스템을 개발한다. 그러나 펌웨어는 간단한 하드웨어 제어용이기 때문에 복잡한 작업을 수행하는 소프트웨어 개발이 어렵다. 이런 펌웨어의 한계 때문에 RTOS를 사용한다[3].

RTOS에는 여러 가지 장점이 있다. 첫 번째로 하드웨어에 종속되지 않고 프로그램 호환성에 있다. 그러면 하드웨어 선정이나 교체에 따른 부담을 줄일 수 있다. 두 번째로 태스크 단위별로 프로그램을 제작하므로 재사용성이 높아진다. 재사용성이 높아지면 개발시간이 단축되고 검증된 코드를 사용으로 시험시간이 줄어든다. 세 번째로 다중의 프로세서에도 프

로세서간 통신이 통일성을 갖고 간편하게 통신 구현이 가능하다. 마지막으로 프로그램의 오류를 사전에 예방, 정정해주어 시스템이 다운되는 것을 방지하여 안정성을 높인다. 이러한 장점 때문에 로봇에 RTOS를 사용한다[4].

본 논문에서는 기존 RTOS를 소형 다관절로봇인 6족로봇의 운용을 위해 경량화한다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구로서 기존 임베디드 시스템 개발 방법에 대하여 설명한다. 3 장에서는 제안한 운용 소프트웨어 설계에 대하여 언급한다. 4 장에서는 적용사례로 소형 다관절로봇인 6족로봇에 제안한 소프트웨어를 실험한다. 마지막 장에서는 결론 및 향후 연구를 언급한다.

2. 관련 연구

기존의 다관절로봇 개발을 위한 방법은 크게 3가지로 나눌 수가 있다. 펌웨어[5]를 이용한 방법, Non-RTOS[6] 이용한 방법, RTOS[7]를 이용한 방법이 있다. 보통 저가형 소형 로봇에는 펌웨어가 사용되고 고가형 로봇에는 운영체제가 사용된다.

펌웨어 기반 개발 방법은 병렬성을 지원하지 않고, 적은 규모에서 사용되며, 하드웨어 의존도가 높아 제한된 로봇에만 적용되는 단점이 있다. 또한 복잡성이 높은 반면에 수행속도는 빠르다.

Non-RTOS 기반 방법과 RTOS 기반 방법은 운영체제가 요구되며, 병렬성이 지원되고, 스레드 단위별로 작업 분할이 가능하여 중대 규모의 소프트웨어의 개발이 가능하다. 하드웨어 의존도가 낮아 여러 시스템에 탑재 가능하며 복잡성이 낮은 장점이 있지만 펌웨어보다 느린 단점이 있다.

† 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업(2008~2009)으로 수행된 연구결과임

3. 운용 소프트웨어 경량화 설계

일반적인 운영체제는 태스크들의 수행을 위해서 그림 1 과 같이 상태를 가진다. 이 상태는 태스크가 현재 어떤 상태인지를 기술하여 스케줄링이나 시스템운영에 사용된다. 보통의 운영체제는 6 개의 Suspend, Wait, Dormant, Ready, Running, Interrupted 상태로 구분된다. Dormant 는 태스크가 실행되기 전의 상태를 말한다. Ready 는 메모리에 로드 되어 실행 준비가 끝난 상태이다. Running 는 태스크가 현재 수행된 상태 이다. Interrupted 는 인터럽트가 실행되어 잠시 대기하는 상태이다. Wait 는 Delay 함수나 스케줄러에 의해 잠시 대기 하는 상태이다. Suspend 는 메모리에 로드 되지 않으며 중지된 상태를 말한다.

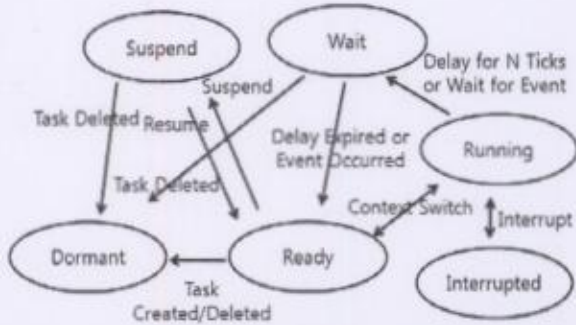


그림 1 일반적인 운영체제의 상태 다이어그램

제안한 운용 소프트웨어는 6 개의 상태를 사용하지 않고 경량화를 위해서 TASK_KILL, TASK_WAIT, TASK_READY, TASK_RUN 의 4 가지 상태만 사용한다.

TASK_KILL 은 태스크가 수행되기 전 초기화된 상태이다. 제안한 운용 소프트웨어는 태스크를 삭제하는 기능이 없으므로 TASK_KILL 상태는 초기화만 되어있는 모든 태스크를 말한다.

TASK_READY 는 태스크를 바로 수행할 수 있도록 준비된 상태이다.

TASK_RUN 은 현재 실행하고 있는 태스크를 말한다.

TASK_WAIT 는 태스크가 잠시 대기하는 상태이다. 제안한 RTOS 는 그림 2 와 같이 TASK_READY -> TASK_RUN -> TASK_WAIT 를 반복하는 간단한 구조를 가진다. 운용 소프트웨어의 상태를 4 가지로 축소하였기 때문에 인터럽트 처리에 대한 태스크 상태가 존재하지 않는다. 적용할 대상이 타이머 인터럽트만 사용하기 때문에 문제가 발생되지 않는다.

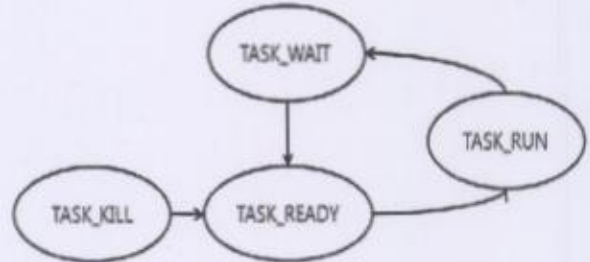


그림 2 제안한 운용 소프트웨어의 태스크 상태 전이

4. 적용사례

다관절로봇은 산악지형, 사막, 수중과 같은 다양한 환경 속에서 임무를 수행하게 된다. 험난한 지역을 이동하기 위해서는 다관절로봇 형태적합 하지만 로봇의 다리가 많고 다리를 동시에 움직여야 되기 때문에 기존의 방법만으로는 개발의 어려움이 있다. 본 논문에서 제안한 운용 소프트웨어를 다관절로봇인 6 족에 적용한다[8,9].

4.1 하드웨어 사양

6 족형 자율탐색로봇은 모터제어부에 Atmega128 를 센서 제어부에 Atmega8535 를 사용한 2 개의 마이크로컨트롤러를 가지고 있다. 그리고 4 축 방향으로 초음파 센서가 장착되어 사방의 물체를 감지 할 수 있도록 되어 있다. 단방향 자이로센서 2 개가 장착되어 x 축, y 축 방향의 기울기를 탐지 할 수 있다. 또한 다리당 3 개의 관절이 사용되어 총 18 개의 서보모터를 사용한다. 실시간제어를 위해 블루투스 사용 된다.

표 1 자율 탐색 로봇 하드웨어 구성

구분	자율 탐색 로봇	
Microcontroller	Main	Atmega 128 16MHz
	Sub	Atmega8535 16MHz
RAM	Main	4KByte
	Sub	512Byte
EEPROM	Main	128KByte
	Sub	8KByte
OS	none	
Sensor	4	
Gyro Sensor	2	
Communication	Bluetooth	
Motors	18	
Languages	C	

6 족로봇의 특징은 프로세서를 2 개 나누어 센서부와 수행부로 나누어져 있는 것이다. 센서부는 상위 모듈이라고 하고 수행부는 바디 모듈이라고 부른다. 상위 모듈과 바디 모듈은 UART 를 통해서 프로세서 간 통신을 수행한다. 상위 모듈은 Atmega8535 를 사용하며 4 개의 초음파 센서와 2 개의 자이로 센서로 구성된다. PC0~3 은 초음파센서를 제어하기 위한 포트 이고 입력 출력을 사용한다. 그리고 PA0, PA1 은 자이로 센서를 제어하기 위한 포트이다. 상위 모듈은 외부로부터 받은 센서 데이터를 바디부분에 전달하는 역할을 한다. 바디부분에 전달할 때는 직렬 통신을 이용한다. 자율탐색로봇의 구성은 표 1 과 같다.

4.2 응용 프로그램 개발

제안한 응용 소프트웨어로 6 족로봇을 개발하기 위해서는 그림 3 과 같이 3 개의 태스크로 분할한다. 센서처리 태스크는 상위모듈의 초음파와 자이로를 통해 거리 값과 기울기 값을 수신 받아 처리한다. 로봇 모션 처리 태스크는 전진, 후진, 좌턴, 우턴과 같은 동작을 수행하기 위한 태스크이다. 로봇 방향 처리 태스크는 상위모듈로 받은 데이터를 이용하여 로봇의 행동을 결정하는 태스크이다. 이렇게 제안한 응용 소프트웨어로 개발하면 태스크 단위로 만들 수 있어 새로운 하드웨어 및 기능 추가시 기존코드의 수정 없이 태스크 추가로 동작할 수 있다. 각 각의 태스크는 동시에 수행되기 때문에 기존 태스크에 영향을 받지 않는 독립적인 단위가 된다. 또한 태스크와 API (Application Programming Interface)를 통해 공통의 표준개발 방법을 제공하여 여러 사람이 동시에 개발이 가능하다.

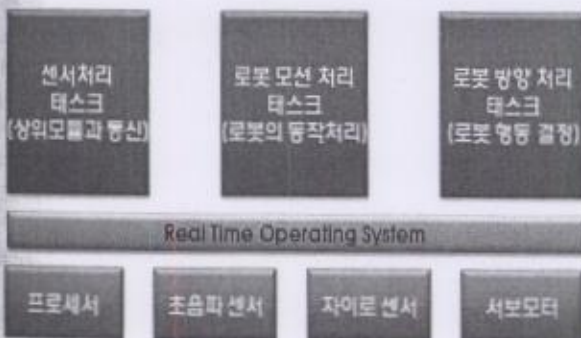


그림 3 제안한 응용 소프트웨어를 이용한 로봇제어

초음파센서를 이용하여 물체를 회피하기 방법은 먼저 우선 모듈에 물체가 감지되면 뒤로 후진하였다가 좌우 물체를 확인 후, 좌측에 물체가 있으면 우측으로, 우측에 물체가 있으면 좌측으로 회전하여 물체 피해간다.

4.3 응용 프로그램 수행

6 족로봇에 프로그램을 만들고 포팅하여 응용 프로그램을 수행한다. 본 논문의 6 족로봇의 예제는 장애물 회피와 경사면 오르기를 한다. 6 족로봇의 장애물 회피는 그림 4 와 같이 동작한다. 6 족로봇의 동작은 초음파센서로 앞에 물체를 발견하면 뒤로 후진한 후 좌우를 검사한 후, 좌측 물체가 없으면 좌측으로 우측에 물체가 없으면 우측으로 피해간다. 이러한 방법으로 장애물을 회피한다.



그림 4 6 족로봇의 장애물 회피

5. 결론 및 향후 연구과제

기존의 소형 다관절로봇은 펌웨어를 사용하여 시스템을 개발해 왔다. 하지만 작업량과 하드웨어 제어 루틴이 많아지면서 소프트웨어의 복잡성 증가로 펌웨어로 개발이 어렵다. 또한 펌웨어는 동시처리, 인터럽트 지연 문제, 타이밍 제어의 어려움이 있다. 이런 문제와 더불어 소형 다관절로봇에 RTOS 를 적용하므로 크기와 관절 제어 문제 해결이 필요하다.

개발한 응용 소프트웨어는 기존 RTOS 의 문제를 해결하기 위해 크기 최소화 및 관절 제어를 최적화 하였다. 크기 최소화를 위해 태스크 상태를 경량화 하였다.

또한 6 족로봇에 제안한 응용 소프트웨어를 적용해 보았다. 그 결과 6 족로봇에 적합한 크기의 응용 소프트웨어를 탑재하였고 관절 제어를 최적화할 수 있었다.

향후 연구과제로 다중 프로세서 지원과 프로세서 간의 상호운동을 위한 운영체제를 개발 중이다. 또한 모듈화 기능 지원하고 더 많은 하드웨어 기기를 지원하도록 소프트웨어의 확장이 필요하다.

참고문헌

- [1] 김동우, 손현승, 김우열, 김영철, "자율 탐색 로봇 설계를 위한 M&S 적용", 국방과학연구소, Vol. 12, pp. 168-171, 2008.
- [2] 김재수, 손현승, 김우열, 김영철, "A Study on M&S Environment for Designing The Autonomous Reconnaissance Ground Robot", 한국군사과학학회, Vol. 11, No. 6, pp. 127-134 2008. 12.
- [3] 김용규, 이명호, "자동차 조향 장치의 동향과 전망 -기계 중심에서 Software 중심으로-", 정보처리학회지, 제 15 권, 제 5 호, pp. 40-47, 2008.
- [4] 김동진, 유승환, 신윤덕, 장승익, 기창두, "엔터테인먼트용 조류형 2 족 보행 로봇의 설계 및 구현", 한국정밀공학회지, 제 22 권 제 3 호, pp. 38-45, 2005.
- [5] 엄우용, 이종호, "로봇 플랫폼을 위한 이족보행로봇 개발", 대한전자공학회, 제 43 권, 제 4 호, pp. 136-143, 2006.
- [6] L.Wang, B. Yao, Y. Yang, and Z. Zhu, "A survey of embedded operating system," Techical Report, University of California, San Diego, USA, 2001.
- [7] 진태석, "임베디드 시스템과 Real-Time OS", 정보통신연구진흥원, 주간기술동향 통권 1345 호, 2008.
- [8] 손현승, 김우열, 김영철, "Implementation of Technique for Movement Control of Multi-Joint Robot", 한국정보처리학회, Vol. 15, No. 2, pp.593-596, 08.11.14
- [9] 김재수, 손현승, 김우열, 김영철, "A Study on Education Softwarefor Controlling of Multi-Joint Robot", 한국정보교육학회, Vol. 12, No. 4, pp.469-476, 08.12.