

프로그래밍언어 학습 동기유발을 위한 교육용

Java 로봇의 설계 및 구현

백정현^o

^o우송정보대학 컴퓨터정보계열

e-mail: jhbaek@wsi.ac.kr

The Design And Implementation of Educational Java Robot for Learning Motivation of Programmng Language

Jeong-Hyun Baek^o

^oDivision of Computer Information, Woosong College

● 요약 ●

본 연구에서는 Atmel사의 AVR 마이크로프로세서에 적합하게 개발된 Java 바이트코드 인터프리터인 NanoVM을 자체 개발한 마이크로로봇에 이식하여 Java 언어 전용 로봇을 구현하였다. 따라서 마이크로프로세서의 구조와 회로를 모르는 학생들도 로봇을 프로그래밍하면서 Java 언어를 효율적으로 학습할 수 있다.

더욱이 최근 학생들의 프로그래밍언어 학습 능력이 떨어지면서 컴퓨터 관련학과의 프로그래밍언어 교육이 많은 어려움을 겪고 있다. 따라서 학생들의 프로그래밍언어 학습 동기를 부여하고 창의 공학적 프로그래밍언어 교육프로그램의 도입이 필요한 시점에서 본 연구에서 개발한 Java 로봇은 많은 기여를 할 것으로 기대된다.

키워드: 자바가상머신(Java Virtual Machine), 자바 로봇(Java Robot), 임베디드 자바(Embedded Java), 마이크로로봇(Micro Robot)

I. 서론

국내외적으로 많은 교육용 로봇이 개발되었으나 주로 초등학교의 창의적 학습지원 도구나 대학의 전기 및 전자공학과 학생들의 마이크로프로세서 학습용으로 개발되었다.

이러한 목적으로 개발된 로봇 프로그래밍 환경은 컴퓨터학과의 프로그래밍언어 교육용 도구로서 로봇을 활용하기에는 부적합한 면이 많았다.

따라서 Java나 C언어와 같은 컴퓨터 프로그래밍언어 학습용 로봇으로 응용 가능한 제품은 레고사의 LEGO 마인드스톰 NXT가 유일한 실정이다.

본 연구에서는 마이크로프로세서의 구조와 회로를 모르는 학생들도 로봇을 프로그래밍하면서 Java 언어를 효율적으로 학습할 수 있도록 Java 언어로 프로그래밍이 가능한 마이크로로봇을 개발한다.

Java 언어는 SUN 마이크로시스템즈사에 의하여 임베디드 시스템 프로그래밍언어로 개발되었다. 그러나 본래의 개발 의도와는 달리 인터넷 환경에 사용할 수 있도록 최적화 되면서 웹 프로그램 개발언어로 각광 받게 되었다.

또한 Java언어는 C언어에 기반을 두고 객체지향 프로그래밍 언어개념이 추가된 C++ 언어에서 유래되었기 때문에 프로그래밍언어의 개념을 학습하기에 매우 적합한 언어로 여겨진다.

Java 인터프리터가 탑재된 자바로봇은 범용 컴퓨터의 모니터중

심의 프로그래밍 실습에 비하여 로봇의 행위를 시각과 청각은 물론 시각에 의하여 감지하고 검증할 수 있기 때문에 프로그래밍 학습 효과가 뛰어나다.

따라서 학생들의 프로그래밍언어 학습 동기를 부여하고 창의 공학적 프로그래밍언어 교육프로그램의 도입이 필요한 시점에서 본 연구에서 개발한 Java 로봇은 많은 기여를 할 것으로 기대된다.

II. NanoVM

2.1 NanoVM 소개

NanoVM은 자바 가상머신인 JVM(Java Virtual Machine)을 최소화 시킨 버전이며, 저가의 8비트 마이크로컨트롤러에서 Java 프로그램을 실행시켜 주도록 구현 되었다. NanoVM의 배포판은 타겟 시스템에 설치되는 이미지와 자바 클래스파일을 변환하여 적재하는 도구인 NanoVMTool로 구성된다[1].

NanoVM은 메모리 관리도구와 같이 플랫폼 독립적인 구성요소와 네이티브 메소드들로 구성된다. 네이티브 메소드는 임베디드 시스템에서 하드웨어를 제어하기 위하여 사용되는 C언어 루틴들이며, 이들은 완전히 플랫폼 종속적이다. 네이티브 메소드는 다양한 on-chip 모듈들과 주변장치들에 간단히 접근하는 것을 허용하며, 자바 사용자 프로그램이라고 부른다. 네이티브 메소드의 사용은 자바 프로그램이 자바가상머신 환경에서 제공되는 하드웨어 기

능을 사용하는 것을 허용하며, 또한 몇몇 네이티브 메소드들은 자바의 표준 라이브러리 기능들을 구현할 때 사용되며, 네이티브 메소드를 사용함으로써 다음과 같은 장점을 얻을 수 있다[2].

- 자바언어 보다 C언어로 저수준의 하드웨어제어 기능을 코딩할 수 있다.
- 이미 다양한 on-chip 주변장치들을 제어하는 C 루틴들이 존재한다.
- 기금 상업용 혹은 오픈소스 C 라이브러리들을 사용할 수 있다.
- 부동소수점 연산과 같은 복잡한 저수준의 기능들은 네이티브 메소드로 구현할 때보다 빠르다.

2.2 NanoVM의 구성

그림1에 나타낸 것과 같이 NanoVM 가상머신은 NanoVM 가상기계의 코드사이즈는 8KB 이내로 구현할 수 있다. 자바 바이트코드 해석기를 위한 RAM의 소모량도 1KB 이내로 인턴프리터 스택과 Heap 영역을 할당할 수 있기 때문에 Atmel사의 AVR 시리즈에서 가장 작은 Atmega8 CPU로 구현이 가능함을 보여준다.

그러나 사용자의 응용프로그램 저장을 위한 별도의 비휘발성 메모리가 요구되며, 본 논문에서는 Atmega8 CPU에 내장된 512 byte의 EEPROM을 사용하였다. 따라서 사용자 프로그램의 크기는 바이트코드 512 바이트 이내로 작성해야 하며, 대략적으로 간단한 마이크로마우스나 라인트레이서 정도를 구현할 수 있는 크기이다.

따라서 대용량의 사용자 프로그램을 실행하기 위하여 시리얼 EEPROM을 CPU 외부에 장착하거나 Atmega128과 같이 대용량의 플래시 메모리를 가지고 있는 CPU를 사용하여 플래시 메모리의 일부를 사용자 응용프로그램 영역으로 사용하는 것이 필요하다.

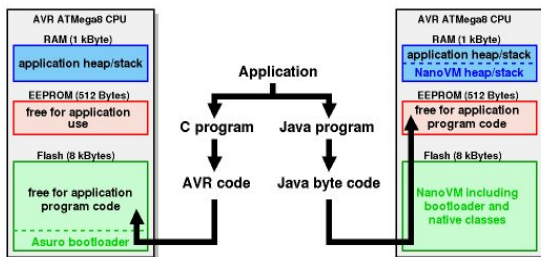


그림 1. NanoVM 자바가상머신의 구조
Fig. 1. Architecture of NanoVM Java Virtual Machine

III. 로봇 설계 및 제작

3.1 마이크로로봇 설계

본 논문에서 구현한 AVR 마이크로로봇 Robot-Tutor II는 휴대하기 편리하고, 아트멜사(www.atmel.co.kr)의 표준 개발환경인 AVR 스튜디오를 사용하여 프로그래밍할 수 있도록 개발하였다[3].

특히, 시리얼 ISP 인터페이스를 내장하여 RS-232 포트를 통하여 AVR 프로그램의 적재와 시리얼 통신 기능도 수행한다. 공개

소프트웨어인 PonyProg2000 사용하여 별도의 ISP장비나 소프트웨어 없이 사용자의 응용 프로그램을 AVR의 플래시 메모리로 다운로드하여 실행시킬 수 있다. 그림2는 자바로봇 프레임인 Robot-Tutor II의 ISP 및 시리얼통신 회로를 보여준다.

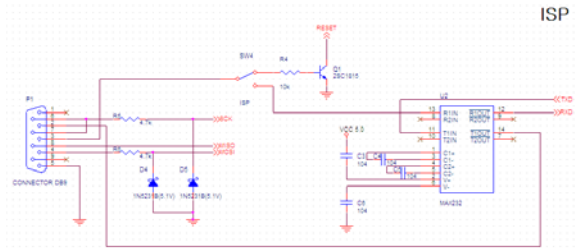


그림 2 Robot-Tutor II의 ISP 및 RS232 회로
Fig. 2. ISP and Rs232 Circuit of Robot-Tutor II

또한, 다양한 로봇의 기능들을 테스트할 수 있도록 풍부한 응용 소자들을 장착하고, 입출력 포트를 확장하여 다양한 응용 작품을 만들 수 있도록 하였다. Robo-Tutor II 마이크로로봇의 주요 사양은 다음과 같다.

- CPU - Atmega8(Flash 8KB, SRAM 1KB, EEPROM 512Byte, 28핀 PDIP 패키지)
- 클럭 - ATmega8의 내부 클럭 사용(8MHZ 동작)
- 디스플레이 - FND(4 digit), LED 10개(출력 8bit, RS232 통신 표시 2개), 캐릭터 LCD 커넥터 부착
- 디지털 입력 - 4개의 스위치(인터럽트 0,1과 타이머 0, ICP), TV 적외선 수신센서(KSM603LM)
- 아날로그 입력 - LM35DZ 온도센서, 적외선 송수신 센서 3조(마이크로 마우스용)
- 디지털출력 - 부저, 모터드라이브(BA6208) 2개
- 확장 커넥터 - 10핀 박스헤더 커넥터 장착, 디지털 I/O확장 및 라인트레이서용 센서보드 장착용
- 통신인터페이스 - Max232 인터페이스 장착(RS232 드라이버 2채널 지원)

그림 3은 본 논문에서 제작한 Java로봇 프레임인 Robot-Tutor II의 디스플레이와 PWM 모터 구동회로를 나타내었다.

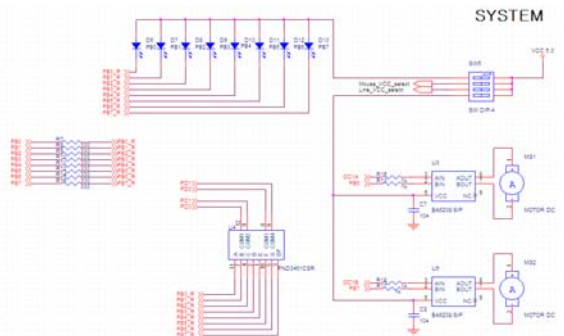


그림 3. Robot-Tutor II의 모터 및 디스플레이 회로
Fig. 3. Motor and Display Circuit of Robot-Tutor II

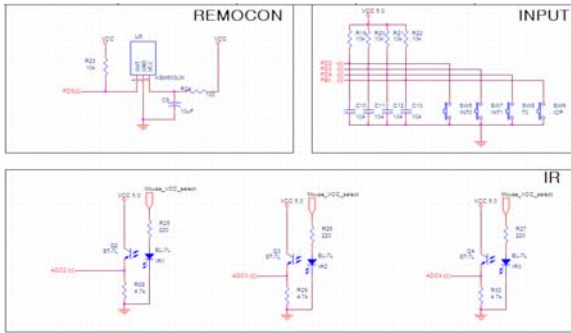


그림 4. 자바 로봇의 입력 및 적외선 센서 회로
Fig. 4. Input and IR Sensor Circuit of Java Robot

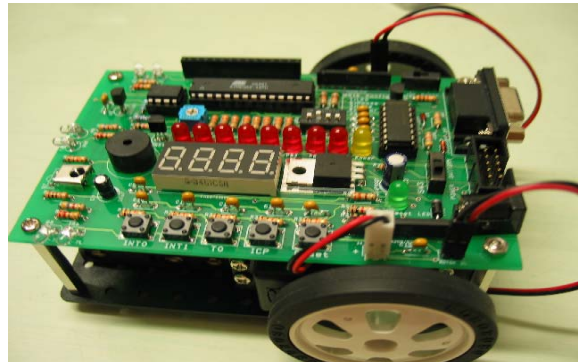


그림 6. 완성된 Java 로봇의 모습
Fig. 6. View of Complete Java Robot

그림4는 Robot-Tutor II의 TV 적외선 리모콘 수신회로와 거리 측정을 위한 3쌍의 적외선 송수신 센서 회로도를 나타내었다. 적외선 리모콘 수신회로는 TV 리모콘을 이용하여 로봇을 제어할 수 있고, 3쌍의 장애물 감지용 적외선 센서는 청소로봇이나 마이크로마우스와 같이 장애물을 감지하여 회피하는 로봇을 제작할 수 있다.

3.2 마이크로로봇 제작

본 논문에서 설계한 Java 교육용 마이크로로봇인 Robot-Tutor II는 OrCAD를 이용하여 회로를 설계하고 PCB 도면을 생성하였다. PCB는 2층의 에폭시 기판으로 제작하였으며 로봇 프레임과 최종 조립한 로봇의 모습은 그림5, 그림6과 같다. 특히, 모든 부품을 DIP 패키지형태로 설계 및 제작하였으므로 납땜인두와 실납과 같은 간단한 도구만 갖추면 학생들이 쉽게 조립하고 제작할 수 있도록 배려하였다.

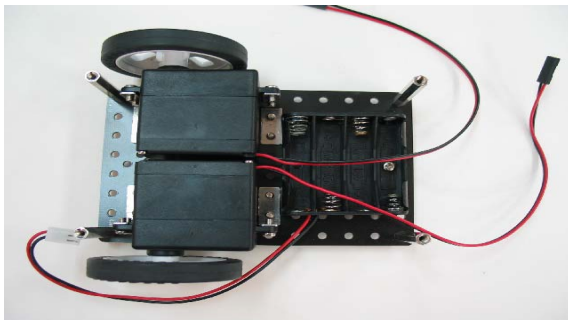


그림 5. Java 로봇의 프레임 구조
Fig. 5. Frame Architecture of Java Robot

IV. 실험 및 평가

4.1 Java 프로그램 작성

NanoVM에서 지원하는 자바의 클래스와 메소드를 이용하여 NanoVM 가상머신을 위한 자바 응용 프로그램을 작성할 수 있다. 또한 본 논문에서는 AVR로 제작된 마이크로로봇을 제어하기 위한 클래스와 메소드들을 추가하였다.

다음의 그림 7은 NanoVM을 탑재한 Java 로봇의 라인트레이서 로봇 프로그램의 일부이며, Java 개발도구인 JDK에 포함된 Javac 컴파일러 명령으로 컴파일한 후 생성된 자바의 바이트코드를 NanoVM이 탑재된 로봇에서 실행한다.

```
import nanovm.avr.*;
class LineRobot
{
    public static void main(String[] args)
    { System.out.println("Starting Line Robot!!");
      int intRightSensor=0;int intCentorSensor=0; //초기화
      int intLeftSensor=0; AVR.portB.setOutput(3);
      AVR.portB.setOutput(4); AVR.portB.setOutput(5);
      AVR.portB.setBit(3); AVR.portB.setBit(4);
      .....
      Timer.setPrescaler(Timer.DIV8);
      AVR.pwm0.setPrescaler(Pwm.DIV1024);
      .....
      while(true) { // 라인트레이서 기능 수행
        AVR.portB.clrBit(3); AVR.portB.setBit(4);
        AVR.portB.setBit(5); Timer.wait(20);
        if(Ac.getValue()==1) intRightSensor = 0;
        else intRightSensor = 1
        .....
        if(intLeftSensor == 0 && intCentorSensor == 1 &&
           intRightSensor == 0 ) //중앙 센서 감지
        { AVR.pwm0.setRatio(200); //왼쪽 모터
          AVR.pwm1.setRatio(200); //오른쪽 모터
        } else
        .....
      }
    }
}
```

그림 7. 라인트레이서 로봇 프로그램의 구조
Fig. 7. Architecture of Line Tracer Robot Program

4.2 Java 프로그램 적재 및 실행

본 논문에서 개발한 시리얼 통신전용 자바 바이트코드 다운로드 Visual Studio 2005에서 C# 언어를 이용하여 제작하였으며, NanoVM 자바로봇 전용으로 개발된 운영 소프트웨어로서 바이트 코드를 16진수로 표현하고 C언어의 배열 초기화 형식으로 작성된 텍스트 파일을 전송한다.

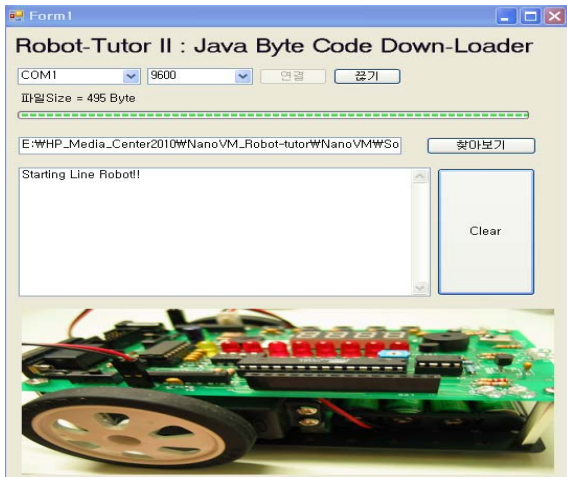


그림 8. Robot-Tutor II 자바 Byte 코드 다운로드
Fig. 8. Byte Code Down-Loader of Robot-Tutor II

또한, 자바 Byte코드 다운로드되는 자바 바이트코드를 로봇으로 전송하는 기능과 시리얼통신을 통한 콘솔 역할을 수행한다. 그림 9는 교육용 자바 로봇에 라인트레이서 프로그램을 탑재하여 실험 및 평가하는 모습을 보여준다.

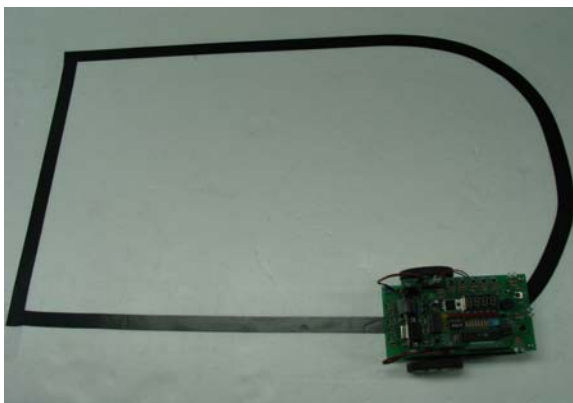


그림 9. Java 로봇의 라인트레이서 시험 및 평가
Fig. 9. Line Tracer Test and Evaluation of Java Robot

V. 결 론

학생들의 프로그래밍언어 학습 능력이 떨어지면서 컴퓨터 관련 학과의 프로그래밍언어 교육에 많은 어려움을 겪고 있다. 따라서 학생들의 프로그래밍언어 학습 동기를 부여하고 창의 공학적 프로그래밍언어 교육프로그램의 도입이 필요한 시점에서 본 연구에서 개발한 Java 로봇이 많은 기여를 할 것으로 기대된다.

그러나, AVR 시리 중 가장 규모가 작은 CPU를 사용하여 개발하였기 때문에 사용자 프로그램의 크기에 제약이 있고, 중대규모의 다양한 학습 프로그램의 작성을 위하여 Atmega128이나 외부 EEPROM을 장착하여 성능을 향상시켜야 하는 과제도 남아있다.

참고문헌

- [1] Harbaum, T., "The NanoVM-Java for the AVR", <http://www.harbaum.org/till/nanovm>. (2005). Holzner, S., Eclipse Cookbook. O'Reill., (2004).
- [2] Min-Soo Jung, "A study on high performance Java virtual machine for smart card", Journal of the Korean Data Information Science Society, 2009, 20(1), 125.137.
- [3] Lee-Sub Lee, Seong-Hoon Kim, "Java based Platform for Educational Robots on AVR", 지능정보연구 제15권 제3호 2009년 9월(pp.17~29).
- [4] SeMan Oh, YangSun Lee, KwangMan Ko, "Design and Implementation of a Virtual Machine for Embedded Systems", Journal of Korea Multimedia Society Vol. 8. No. 9, September 2005(pp. 1282~1291).