

# PUMA Robot에서의 RCCL (Robot Control C Library)의 구현

배 본 호                      이 진 수

포항공과대학 전자전기 공학과

## Implementation of RCCL on PUMA

B.H. BAE                      J.S. LEE

Dept. of Electronic and Elecrtical Engineering, POSTECH

### Abstract

RCCL(Robot Control C Library) is general purpose robot control language.

It is programmed with C language and composed of C library. So it is well portable and supports sensor integration control and high level force control algorithms.

We implemented RCCL on PUMA. We developed servo controller of DDC(Direct Digital Control). We used intel 8097BH one chip micro controller as CPU. One digital servo board controls three motors.

Host computer is IBM PC 386DX-33 with RCCL.

### 1. 서론

초기의 로봇이 간단한 하드웨어에 의해 제어되었던 반면 근래의 추세는 소형이나 중형의 컴퓨터를 이용하여 고기능의 제어언어로서 로봇을 제어하는 방향이다.

그리하여 그동안 다양한 로봇들과 함께 여러가지 제어언어가 개발되었다. 그 예로 PUMA로봇용의 VAL이나 Stanford로봇용의 AL 등이 있겠다. 그러나 이러한 제어언어들은 로봇자체의 특성을 효과적으로 구현하는 데에 많은 노력이 기울어져 일반 컴퓨터 언어의 관점에서 보면 부족한 점이 많다. VAL의 경우 부동소수점, 문자 스트링을 사용할

수 없고 서브 루틴에도 인자를 전달할 수가 없다는 등의 것이다. 즉 각 로봇에 특화되어 있어서 다양한 로봇에 대한 호환성이 부족하다.

그래서 최근의 이러한 로봇 제어 언어로 일반 컴퓨터 언어를 사용하는 것이 바람직하다는 의견이 대두되고 있다. 실제 이를 뒷받침하는 증거로 로봇제어에 사용되고 있는 프로그램을 살펴보면 작은 일부만이 로봇종속적(robot specific)인 문장이고 그 대부분이 사용자 인터페이스, 데이터 베이스, 그리고 수치제어 등과 같은 것이라는 연구 결과가 발표되었다.

이러한 관점에서 로봇종속적인 부분을 서브루틴 형태로 개발하여 라이브러리 형태로 두고 프로그래머가 필요한 기능의 루틴을 서브루틴 호출의 형식으로 로봇 라이브러리 안에 저장되어 있는 함수를 불러 사용하는 방식이 제안되었다. 본 논문에서 사용하는 RCCL(Robot control C Library)도 이러한 서브루틴 호출 방식을 사용하고 있다. RCCL 라이브러리 안에는 각종 로봇종속적인 문장들이 서브루틴 호출 방식으로 저장되어 있다. 그리고 사용자는 힘제어(Force Control)와 같은 새로운 제어 방식을 C 언어로 작성하여 적용할 수가 있다.

원래의 RCCL은 micro vax상에서 개발 운용되었고 디지털 서보를 가진 PUMA로봇의 제어에 이용되고 있었다.

본 연구에서는 이를 IBM PC 386에 이식시키고 또 아날로그 서보를 가진 구형 PUMA 로봇의 서보 콘트롤러를 디지털 서보로 재제작하여 RCCL 제어 언어의 개발과 발전을 도모

하고 새로운 제어 알고리즘을 실제 PUMA로봇에 실험할 수 있는 환경을 개발하고자 한다.

## 2. 전체 개요

아래 그림 1은 전체 구성도를 나타낸다.

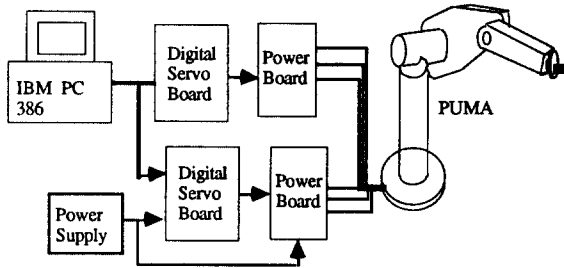


그림 1 전체 구성도

전체 로봇 제어언어는 RCCL로서 IBM PC 386DX-33MHz상에서 돌아가고 있으며 병렬 입출력 보드를 통해 하부 서보 보드와 매 주기마다 통신을 주고 받는다.

디지털 서보 보드는 CPU로서 intel의 MCS96계열의 8097BH 또는 조금 더 빠른 80C196을 호환성이 있게 선택할 수 있다.

디지털 보드는 상부 컴퓨터의 모드에 따라 두가지 다른 반응을 보인다.

우선 제어 알고리즘을 RCCL에서 돌리는 모드에서는 받은 값을 그대로 DAC를 통해 출력하고, 일반적인 위치제어 모드에서는 상부의 설정치를 받아 이를 더 빠른 주기로 보간(interpolation)를 하고 PID제어 알고리즘을 소프트웨어적으로 행하고 이를 출력한다.

전력증폭기는 로봇에서 쓰이고 있는 모터의 특성에 가장 적합하게 설계된 기존의 것을 그대로 썼다.

본 연구에서는 로봇과 전력증폭기를 제외한 전체 컨트롤러를 새로 제작하였고 2개의 디지털 서보 보드로 6축의 모터를 제어한다.

## 3. 디지털 서보 보드

디지털 서보 보드를 중심으로 한 개략도는 그림 2와 같다.

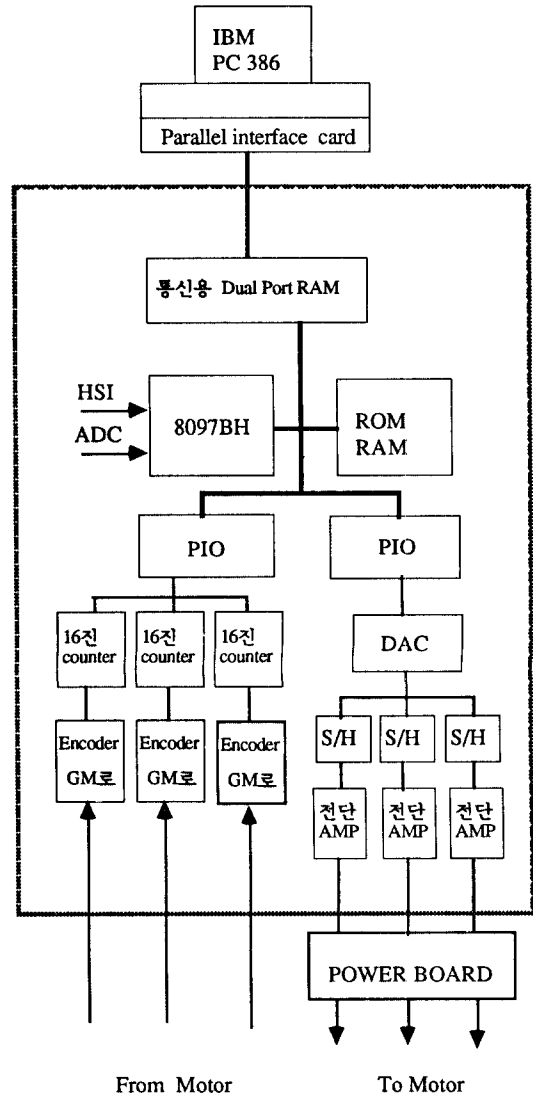


그림 2 디지털 서보 보드 개략도

한 장의 디지털 서보 보드는 3축의 모터를 구동하고 전체 6축의 모터를 구동하기 위해서는 2개의 디지털 서보 보드가 이용된다.

그 중앙 연산 장치로는 intel 사의 8097BH또는 80C196을 사용할 수 있게 제작했다

8087BH는 16BIT의 단일칩 마이크로 컨트롤러로서 내부에 256byte의 register RAM, 카운터, 타이머, RS-232C통신장치,

PWM 장치, HSI, AD 변환기 등을 포함하고 있는데 레지스터가 충분하고 레지스터 대 레지스터(register to register)구조를 취하고 있어 속도가 빠르므로 어셈블러로 프로그램을 짜기도 쉽다. 또한 C 언어 컴파일러가 지원되어 서보 콘트롤러의 중앙연산장치로는 매우 적합하다.

8097BH는 외부적으로 8BIT 또는 16BIT의 메모리와 버스를 지원해준다.

본 연구에서는 속도를 고려하여 메모리(ROM, RAM)를 16BIT으로 액세스(access)하고 입출력으로는 8BIT과 16BIT을 동적으로(dynamically)혼합해 쓰고 있다. 이는 우리가 쉽게 구할 수 있고 흔히 쓰는 입출력 칩들이 주로 8BIT 버스를 요구하기 때문이다.

DAC출력은 하나의 칩을 쓰고 12BIT의 값을 변환시키기로 이를 샘플/홀더 회로를 써서 3개의 출력을 보낸다.

PUMA로봇은 로터리 엔코더와 포텐션미터를 이용하여 위치를 표시하고 있다. 이를 위해 엔코더 신호처리 회로와 16BIT카운터를 제작하였고 또한 포텐션미터의 값은 8087BH에 내장된 ADC로 읽어서 이를 소프트웨어적으로 처리하여 위치를 알게 한다. 또한 로봇암의 위치를 초기화하는 과정에서 엔코더의 Z상 신호(엔코더가 한바퀴에 한번씩 내는 펄스)를 잡을 필요가 있는데 이는 HSI(High Speed Input)인터럽트로 잡아낸다.

상부 컴퓨터와의 통신은 AMD의 1Kbyte 쌍방향 RAM(Dual port RAM)을 써서 충분히 빠른 속도로 이루어진다. 원래 일반적인 위치제어를 위해서는 단순한 병렬 입출력장치를 써서 통신을 해도 충분하지만 제어 알고리즘을 상부 RCCL에서 돌리는 경우의 빠른 통신을 위해서 쌍방향 메모리를 설치하였다. 통신의 동기화를 위하여 쌍방향 메모리는 세마포어를 지원해주고 있다.

### 3. 제어 소프트웨어

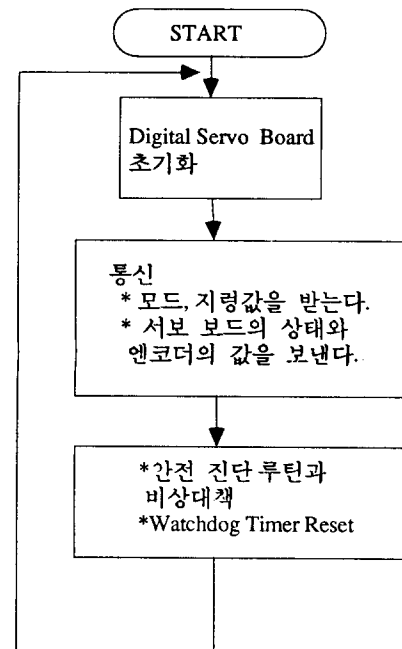
디지털 서보 보드의 제어 소프트웨어는 IC96 컴파일러를 이용하여 C 언어로 만들었다. 실제 연산속도가 어셈블러보

다 조금 뒤지나 CPU의 속도가 충분히 빠르고 디버깅의 이점 때문에 C 언어를 선택했다.

소프트웨어의 주요기능으로는 초기화, 상부 컴퓨터와의 통신(동작 모드, 지령치를 받고 엔코더의 값과 상태를 상부로 전달), 일반적인 위치제어 모드에서의 보간(interpolation)과 PID루틴, 비상시에 대비한 각종 안전 진단 등이 있겠다.

특히 보간은 계산시간을 줄이기 위해 각 값에 대해 미리 계산하여 테이블로 만들어서 쓰고 있다.

그 대략의 흐름도는 그림 3과 같다.



| Software interrupt Routine                  |                         |
|---|-------------------------|
| 일반적인 위치 제어 모드                               | RCCL에서 제어 알고리즘을 실행하는 모드 |
| *매 0.875ms마다 지령치 보간<br>*PID루틴의 실행<br>*DAC출력 | *지령치를 그대로 출력            |

그림 3 흐름도

만일 제어 소프트웨어를 상부 RCCL에서 돌리는 경우는 위치제어뿐 아니라 다양한 힘제어나 최신의 제어 알고리즘을 C언어로 짜서 실행할 수가 있다.

#### 4. 시스템 동기화(Synchronization)

본 연구에서는 RCCL을 IBM PC 386에 이식시키고 IBM PC의 DOS가 지원해주는 인터럽터를 이용한다. 즉 디지털 서보 보드의 8097BH의 내장 타이머를 이용하여 상부 IBM PC의 IRQ(hardware interrupt request)신호를 걸어 RCCL의 궤적 발생 루틴(Trajectory generator routine)과 디지털 서보 보드의 제어 알고리즘 사이에 동기화가 이루어진다. 일반적인 위치제어모드에서는 궤적 발생 루틴이 28mS마다 한번씩 궤적을 발생시켜 하부 디지털 서보 보드에 지령값을 내려 주고 엔코더의 값과 상태를 받는다. 디지털 서보 보드는 이 값을 32번 보간(interpolation)하고 PID 루틴을 실행하여 DAC로 출력하고 아날로그 회로와 전력 증폭기를 거쳐 모터에 전달한다. 이를 위해서 디지털 서보 보드는 8097BH에 내장된 타이머를 이용하여 매 28mS마다 IBM PC에 인터럽터를 요구한다.

또 RCCL에서 제어 루틴을 실행하는 모드에서는 그 제어루틴의 계산량에 따라 허용하는 최대한의 속도를 정하고 그 주기마다 디지털 서보 보드는 인터럽터를 걸어 지령치를 받고 엔코더 값과 상태값을 전달해주고 지령치를 그대로 출력한다.

#### 5. 통신

디지털 서보 보드와 상부 IBM PC사이의 통신은 AMD2130 1Kbyte 쌍방향 메모리(Dual Port RAM)을 써서 이루어진다. 디지털 서보 보드는 8097BH의 데이터와 주소(address) 버스가 직접 쌍방향 메모리의 데이터와 주소에 연결이 되어 있고 제어신호(RD\,WR\등)도 연결이 되어 있어 직접 쓰고

읽는 것이 가능하다. IBM PC는 입출력 버스의 8BIT 데이터 버스는 쌍방향 메모리의 데이터 버스에 직접 연결되어 있고 제어 신호 또한 직접 연결이 되어 있다. 그러나 IBM PC의 주소 버스는 병렬 입출력용 칩인 8255를 통해서 연결된다. 이는 IBM PC의 메모리 지도(memory map)에서 1Kbyte를 쉽게 떼어내기가 힘들기 때문이다. 따라서 IBM PC에서는 8255에 주소를 쓴 다음 읽고 써야 한다.

쌍방간의 핸드셰이킹(hand shaking)은 세마포어 기능을 이용하고 있다. 그 개략도는 그림 6과 같다.

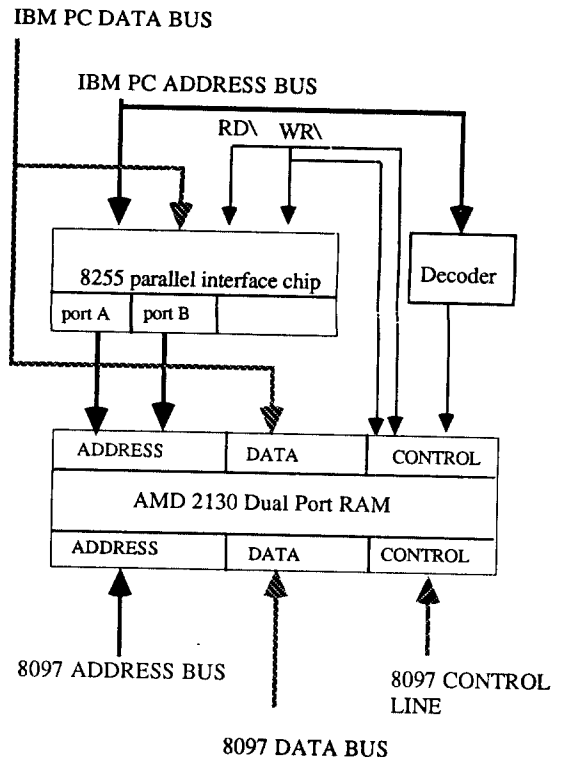


그림 6 쌍방향 메모리 개략도

#### 6. 전력 증폭기(Power Board)

PUMA의 전력 증폭기는 선형 증폭기(Linear Amp)형이다. 에너지 효율이 낮고 열의 발생이 큰 것이 단점이지만 용량이 크지 않을 경우는 빠른 동작을 보장하고 제어 최로가 간단하다.

특히 PUMA의 전력증폭기와 전단증폭기는 입력전압에 비례하는 전류를 발생시키는 구조이므로 토크(torque)제어에 적합하다.

PUMA의 6축은 각각 3개씩 2종류의 모터가 쓰이고 있고 전력 증폭기 역시 2개가 있어 각각 3개씩을 구동한다. 본 연구에서는 전력 증폭기는 원래의 것을 그대로 쓰고 있고 있다.

## 6. 시스템 구성(System Integration)

RCCL은 범용 로봇제어 라이브러리가기 때문에 특별히 PUMA로봇의 제어를 위해서는 PUMA 로봇 종속적인 데이터 테이블 코드를 필요로 한다. 실제 프로그램에서는 PUMA 로봇에 대한 라이브러리가 있어 이를 RCCL에 덧붙여서 사용한다. 이 라이브러리 안에는 PUMA 매니퓰레이터의 기구학적인 정보들과 다이내믹 특성이 저장되어 있고 포워드 키네마틱 루틴, 인버스 키네마틱 루틴, 자코비안 루틴이 포함되어 있다.

또한 디지털 서보 보드와 통신을 하기 위하여 IBM PC 상에 인터페이스 루틴이 있어야 한다. 인터페이스 루틴에서는 매 주기마다 디지털 서보 보드에서의 인터럽트 리퀘스트에 응답하여 궤적 발생 루틴에서 생성되는 지령값들을 쌍방향 메모리에 써주고 하부의 엔코더값과 상태값을 읽고 세마포어를 조작한다. 만약에 상태값에 에러가 발생하면 사용자에게 알려주고 에러 처리 루틴으로 점프한다.

## 7. 결론

본 연구에서는 범용 로봇 제어 언어인 RCCL을 PUMA로봇에 구현하기 위하여 RCCL을 IBM PC 386에 이식하고 이에 맞게 디지털 서보 보드를 제작하였다.

디지털 보드의 프로그램 역시 C언어로 짜여져 있어 누구나 쉽게 이해할 수 있고 유연하게 이용할 수가 있다.

이를 이용하여 RCCL의 다양한 기능을 살리고 그 기능을 더 풍부하게 개선시킬 수가 있고 최신의 제어 기법을 RCCL 상에서 C언어로 짜서 실제 PUMA에 실험을 할 수 있는 환경

이 개발된 것이다.

그러나 RCCL이 범용 제어 언어로 로봇 종속적인 라이브러리만 바꾸면 모든 로봇에 적용할 수 있는데 반해 여기서 개발된 디지털 서보 보드는 PUMA에 종속적이라는 것이 아쉬운 점이다.

따라서 앞으로는 디지털 서보 보드와 전력 증폭기를 개량하여 모든 로봇의 다양한 모터(직류전동기, AC전동기, BLDC 전동기)에 대해서도 소프트웨어적인 전환을 통해 하드웨어적인 변환이 없이 사용가능한 유연한 컨트롤러(Flexible Servo Controller)를 개발해야 할 것이다.

## 8. 참고 문헌

- (1) 이상열, "PUMA-760 로봇의 손목 서보 제어 장치에 관한 연구", 한국 과학 기술원 석사 논문, 1986
- (2) Frank de Schepper, "A study on flexible and integrated servo controllers for actuators in mechatronics systems", Toyohashi University of Technology, 1989
- (3) Nikitas A. Alexandriris, "Microprocessor System Design Concept", Computer Science Press.
- (4) "IC96 User's Guide for DOS System", Intel.
- (5) Franklin, Powel, Workman, "Digital Control of Dynamic System"
- (6) Spong, "Robot Dynamics and Control", Wiley.
- (7) Malcolm, "Robotics an introduction", PWS-KENT
- (8) "DC motors, Speed control, Servo systems", Pergamon Press

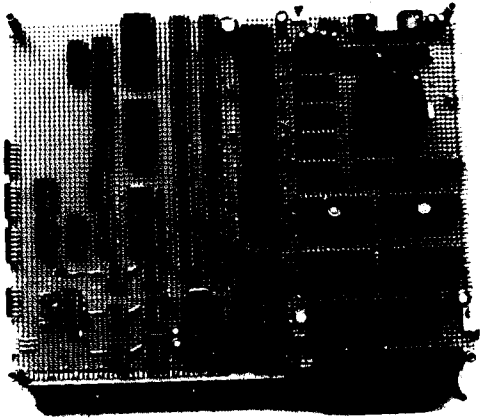


사진 1 디지털 서보 보드