

개방형 모션 제어기의 GUI 설계

이경훈*, 김윤기**, 전재욱***

* 성균관대학교 전기전자 및 컴퓨터공학부(Tel: +82-331-290-7237 : Fax: +82-331-290-7237 : E-mail: cafri@micro.skku.ac.kr)

** 성균관대학교 전기전자 및 컴퓨터공학부(Tel: +82-331-290-7237 : Fax: +82-331-290-7237 : E-mail: kingcar777@yahoo.com)

*** 성균관대학교 전기전자 및 컴퓨터공학부(Tel: +82-331-290-7129 : Fax: +82-331-290-7231 : E-mail: jwjeon@yurim.skku.ac.kr)

The GUI Design of Open Architecture Motion Controller

Kyoung Hoon Lee*, YoonGi Kim**, Jae Wook Jeon***

* Dept. of Electrical & Computer Engr., Univ. of SungKyunkwan, Korea

** Dept. of Electrical & Computer Engr., Univ. of SungKyunkwan, Korea

*** Dept. of Electrical & Computer Engr., Univ. of SungKyunkwan, Korea

Abstract - Lots of automation equipments and robots have been developed and been used. But most control systems have been developed as dedicated systems rather than general-purpose systems. Thus, users cannot modify or improve their software including GUI.

In this paper, an open architecture GUI is proposed in order to make it easy to modify some functions of GUI or to add some functions of GUI. The format of all menu is similar as in previous systems, but some menu can be deleted and be added. Also some menu can be defined by users.

안에서 모든 작업을 할 수 있도록 구성하였다. 그리고 개발 도구는 마이크로소프트사의 비주얼 C++를 사용하였다. 윈도우즈 프로그래밍을 지원하는 컴파일러는 비주얼 베이직, 델파이 등 여러 종류가 있지만 현재 C/C++가 가장 보편화 되어있고 개발에 필요한 리소스 툴이나 유틸리티를 통합적으로 제공하는 것은 비주얼 C++뿐이기 때문이다. 프로그램을 계층적으로 살펴보면 다음 그림과 같다.

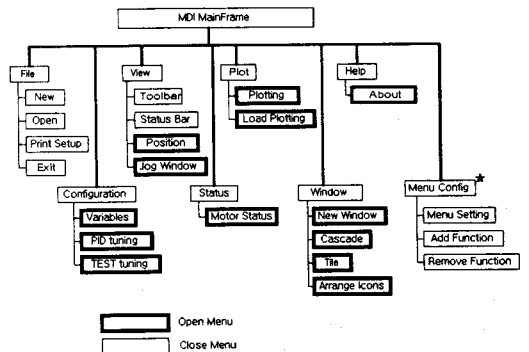


그림 1 프로그램의 계층도

1. 서 론

산업이 고도로 발달하여 많은 자동화 설비나 로봇들이 사용되어 왔고, 그에 따라 이러한 장비들을 위한 수많은 고성능의 제어 시스템들이 개발되고 발달되어 왔다. 하지만 대부분의 제어 시스템들이 특정 환경에 맞춰져 있거나, 다목적의 모션(motion) 제어기라 해도 GUI와 기타 소프트웨어들은 제조사의 독점적인 기술에 의해 구성되어 사용자는 제한적인 형식의 소프트웨어에 의존해야만 했다. 이는 사용자의 다양한 요구사항을 반영할 수 없다는 단점을 가진다. 따라서 개방형(Open Architecture) 소프트웨어 개발 환경을 필요로 하게 된다. 개방형의 개념을 도입한 GUI 환경의 경우 사용자에게 의한 기능의 변경과 추가가 용이해지고, 여러 가지 메뉴 중 사용자의 편의에 따라 일부 메뉴를 바꾸거나 다른 위치로 옮기고 필요한 경우 메뉴를 추가할 수도 있는 기능을 가진다.

따라서 본 논문에서는 개방형 제어 시스템 중 기존의 모션 제어 시스템의 GUI와는 차별화 된 다음과 같은 개방성을 가지는 GUI를 구현하고자 한다. 메뉴의 전체 형식은 동일하지만 일부 메뉴의 위치와 각 항목의 기능을 사용자가 정의할 수 있고 사용자가 원하는 기능만을 선택할 수 있어서 보다 유연성 있는 GUI 환경을 제시한다.

2. 프로그램의 구조

2.1 MDI(Multiple Documents Interface)

본 논문에서의 GUI프로그램에서는 주 프로그램의 상위프레임을 MDI로 설계하여 사용자가 하나의 윈도우

개방형 제어기에 있어서 사용자는 위의 계층도에서 표시된 Open 영역의 기능들을 필요에 따라 추가 또는 삭제할 필요성을 가진다. 이러한 메뉴들의 추가 또는 삭제는 Popup 메뉴 "Menu Config" 영역의 "Menu Setting" 부분을 선택하여 사용자의 요구대로 메뉴를 프로그램에 반영시킬 수 있다. Menu Setting 영역을 선택하게 되면 모달 프레임에 다이얼로그 윈도우가 나타나게 되는데 이 다이얼로그 윈도우는 프로그램의 전체 계층도를 나타내어 사용자가 쉽게 메뉴의 내용을 편집할 수 있게 하였다.

2.2 Menu Setting

"Menu Config"에서는 사용자가 메뉴를 필요에 맞게 편집할 수 있는 기능을 두었다. "Menu Setting"기능을 선택하였을 때의 나타나는 모달 다이얼로그 윈도우는 다음 그림과 같다.

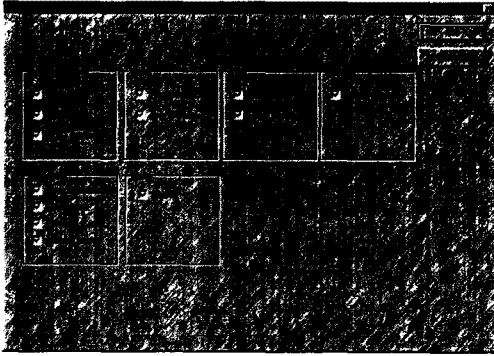


그림 2 Menu Setting Dialog

사용자는 간단하게 체크박스를 컨트롤함으로써 메뉴를 추가 및 삭제 할 수 있다. 본 연구에서는 메뉴의 삭제와 추가를 위해 CMenu Class의 DeleteMenu()와 InsertMenu() 함수를 사용하였다. 먼저 메뉴의 삭제를 위해 메인 프레임의 메뉴바의 포인터를 받아 함수를 실행 한다. 메뉴를 삭제 할 때의 프로그램은 다음 표에 다음 표에 나와 있는 것처럼 이루어진다.

```
CMenu* pSystemMenu = Get.Menu();
pSystemMenu->DeleteMenu(ID_NAME, MF_STRING);
```

표 1 메뉴 삭제

반면 메뉴를 다시 생성시키기 위해서는 생성되는 메뉴가 메인 메뉴의 Sub 메뉴이기 때문에 Sub 메뉴에 대한 포인터를 다시 한번 받아 들여야 한다. 메뉴를 다시 생성하고자 할 때의 프로그램은 다음 표와 같다.

```
CMenu* menu = Get.Menu();
CMenu* pSubMenu = menu->GetSubMenu(Popup number);
pSubMenu->InsertMenu(
    position, MF_STRING, ID_NAME, "Menu Bitmap");
```

표 2 메뉴 생성

다음의 그림에서는 메뉴를 사용자가 편집하여 TEST tuning의 메뉴를 삭제한 모습을 나타내고 있다.

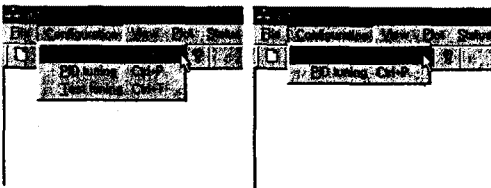


그림 3 메뉴의 추가 및 삭제

이와 같이 그림 1에서 표시된 Open 메뉴들은 Menu Setting을 통하여 삭제되거나 다시 생성 될 수 있는 기능을 가진다.

3. 개방형 모션 제어기

3.1 Control Function

개방형 모션 제어기는 각각의 동작을 정의하는데 있어

특정의 Library가 쓰이게 된다. 다시 말해 사용자가 PC상에서 GUI를 통하여 하나의 명령을 수행시키면 명령에 매핑 되는 Function에 특정 Library를 Call하게 되는 것이다. 기존 방식에 있어서 사용자는 이때 공급자로부터 제공받은 로봇 구동용 기능을 그대로 사용했어야 했다. 그러나 개방형 모션 제어기에서는 이러한 단점을 보완하고자 GUI에서 Library를 편집하여 사용자가 원하는 기능을 추가시킬 수 있다.

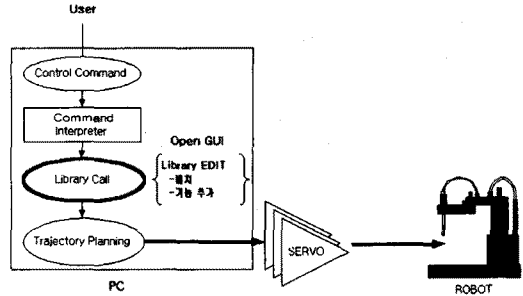


그림 4 개방형 모션 제어기의 동작 개요

Function을 추가하고 Library를 편집하는 과정은 다음과 같이 이루어진다.

3.1.1 Function 추가

함수의 추가는 앞에서 쓰인 CMenu Class 함수의 AppendMenu()를 사용한다. 함수의 생성 프로그램은 다음의 표와 같다.

```
CMenu* pSystemMenu = Get.Menu();
pSystemMenu->AppendMenu(
    MF_STRING, ID_FUNC_1, "menu Bitmap");
```

표 4 Function 추가

본 GUI에서는 사용자가 추가 할 수 있는 함수의 개수를 5개로 제한하고 있다. 이미 만들어진 NULL 함수 5개를 하나씩 추가하여 편집 할 수 있다.

먼저 "Menu Config"->"Add Function"을 선택하여 사용자가 원하는 함수 이름을 입력할 수 있는 다이얼로그가 나타나게 된다. Function Name 다이얼로그 윈도우에 원하는 이름의 함수를 입력하면 다음 그림과 같이 함수가 생성된다.



그림 5 생성된 My_Function1

3.1.2 Function EDIT

생성된 함수를 선택하게 되면 사용자가 함수를 정의할 수 있도록 입력 창이 생성이 된다. 이 창에서는 이미 사용자가 편집한 기능들을 불러올 수 있고 또한 다시 생성하여 저장시킬 수 있는 기능을 갖는다. 이와 같이 사용자가 함수를 설계하기 위해서는 제어기의 Library 자체가 사용자에게 의해서 쉽게 정의되기 위한 구조를 제공해야 한다. 그리고 사용자 함수를 이용하여 제어기의 작업

을 기술한 대로 로봇이 동작하도록 중간 역할을 해주는 Interpreter가 별도로 존재해야 한다.

일반적으로 인터프리터는 사용자가 입력한 문자를 문법적으로 하나의 텍스트씩 비교해 가면서 기존의 함수명과 일치하는지를 조사하고 그 함수 명이 합당하면 그 함수의 Parameter 값을 읽어들이어 동작을 수행하게 된다.

4. 사용자 환경

본 GUI는 각 파트별 기능들을 두어 사용자가 각 기능 창에서 실행 할 수 있는 메뉴들을 따로 구별해 두었다. 또한 각 창에 대한 메뉴들을 따로 Setting이 가능하도록 하여 사용자가 각 환경에 대한 기능을 실행하는데 있어 편리함을 제공하였다. 기존의 다른 프로그램에서 지원하는 메뉴 그룹의 추가기능 및 삭제 기능을 개선하여 기능들의 각각을 편집할 수 있으므로 사용자는 자신에게 가장 알맞은 환경을 갖추어 프로그램을 사용할 수 있을 것이며 다른 추가적인 기능을 생성하여 제어기의 사용 효율을 높일 수 있을 것이다

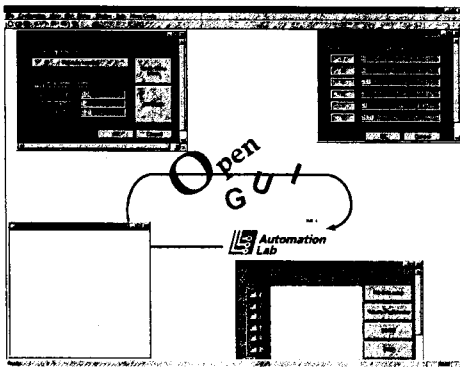


그림 6 Open GUI 실행화면

각 부분별의 기능들은 각각 독립적인 실행을 보장하고 각각의 영역에서 내려진 명령들이 다른 영역의 명령에 의해 누락되는 것을 방지하기 위하여 본 GUI에서는 Scheduler를 제공하여 명령을 관리한다. 여기서 Scheduler란 일종의 FIFO(First In First Out) 형태의 버퍼로써 다음 명령이 진행되기 위해서는 제어기 측에서의 다음 명령 진행에 대한 요구가 있어야 할 것이다.

5. 결 론

본 논문에서는 기존의 정형화 된 GUI의 여러 단점을 보완할 수 있는 방법 중의 하나인 개방성을 제시하고 이를 구현하였다. 불필요한 기능을 제외시키고 각 메뉴와 기능을 사용자의 편의에 따라 추가하거나 변형시킬 수 있도록 유연성을 제공하여 사용자의 편의에 중점을 두었다. 이로써 모든 메뉴와 기능을 익혀야 하는 부담을 덜 수 있고, 사용하지 않은 메뉴를 제거함으로써 프로그램 실행 속도를 높일 수 있으며, 간소화 된 형태로 사용할 수 있다. 앞으로 메뉴간의 항목 이동을 가능토록 하여 사용자가 자주 사용하는 기능만을 하나의 메뉴로 만들 수 있는 기능을 추가하고 더욱 다양한 기능을 구현할 수 있도록 하고자 한다. 본 논문과 같은 GUI 뿐만 아닌 라이브러리나 기타의 응용 프로그램에도 개방성을 부여한다면 기존의 프로그램들과 차별화를 할 수 있을 것이다.

[참 고 문 헌]

- [1] IEEE/NEMI Low-Cost Open Architecture Controller Specification. IEEE/NEMI PR 1553-1998
- [2] Multi-Motion Controller V4.0 사용자 Manual, 삼성전자
- [3] PMAC User's Manual, Delta Tau Data Systems, Inc.
- [4] PMAC Software Reference, Ver 1.16. Delta Tau Data Systems, Inc.
- [5] Introduction to Computer Graphics. Foley, Van Dam, Feiner, Hughes, Phillips
- [6] D. I. Kim, J. W. Jeon, and S. Kim, "Software acceleration/deceleration methods for industrial robots and CNC machine tools," *Mechatronics*, Vol. 4, No. 1, pp. 37-53, 1994
- [7] D. S. Khalsa, "High Performance Motion Control Trajectory Commands Based on The Convolution Integral and Digital Filtering," *Proceedings of Intelligent Motion*, pp. 54-61, Oct. 1990.
- [8] R. Nozawa, et al. "Acceleration/Deceleration Circuit", United States Patent, Patent Number 4,554,497, Nov. 19, 1985.
- [9] Koren Y., *Computer Control of Manufacturing Systems*, McGraw-Hill Inc. 1988.
- [10] H. Inaba et al., "Method and Apparatus for Controlling the Acceleration and Deceleration of a Movable Element without Abrupt Changes in Motion", United States Patent, Patent Number 4,555,758, Nov. 26, 1985.
- [11] J. W. Jeon, "An Efficient Acceleration for Fast Motion of Industrial Robots," *Proceedings of 1995 IEEE 21st IECON*, Orlando, FL, pp.1336-1341, Nov.6 - 10, 1995.
- [12] J. W. Jeon, "Method for Controlling the Traveling Path of A Robot during Acceleration and Deceleration", United States patent, Patent Number: 5,373,439, Dec. 13, 1994.
- [13] 전제욱, 김윤기, 하영렬, "FPGA를 이용한 로봇과 CNC 공작기계용 가감속 회로 개발", 대한전기학회, 1999년 1월 28일, 48A-3-18
- [14] 서일홍 외6, "개방 구조형 로봇 제어 기술 개발", 한양대학교 공학기술 연구소, 1998
- [15] servo drive 및 power controller 사용설명서, 삼성전자, 1993.