

M2M(Mobile to Machine) 환경에서의 유비쿼터스 기반 공작기계 운용기술

김동훈, 송준엽 | 한국기계연구원

1. 개 요

본 연구에서는 개방형 CNC 공작기계를 대상으로 언제 어디서나 감시 및 제어가 가능한 환경을 구하고자 무선 환경하에서 PDA 폰만 있으면 공작기계의 가동상태 및 가공정보 등을 모니터링할 수 있으며, 원하는 장소에서 언제나 제어가 가능하도록 하는 것을 목적으로 한다.

현재 유비쿼터스를 추구하는 사회에서 공작기계와 같은 머신의 상태 및 가공내용을 관찰하고 명령을 언제 어디서나 내릴 수 있게 하기 위하여, 서버(공작기계 컨트롤러, 윈도우 환경)와 PDA 폰간의 통신을 가능하게 하여, Anywhere, Anytime 개념을 구현하였다.

이 구현의 방법으로 기계제어 PC에 TCP/IP 프로토콜을 이용한 서버모듈을 만들었으며, Embedded Visual C++를 이용하여 Arm 기반 PDA와의 모바일용 통신모듈을 구현하였다.

또한, 모바일용 통신모듈과 서버모듈과의 유·무선 통신 과정에서 전송되는 공작기계의 상태(가동여부, NC프로그램명, Alarm여부, X,Y,Z 축 위치 등) 모니터링 데이터와 공작기계 제어(가동, 일시정지, Emergency Stop, 예약가동, 예약정지 등) 데이터의 실제 값을 기계로부터 추출하고 기계에 적용하기 위하여 OPC(Object Linking and Embedding for Process Control)를 기반으로 실제 공작기계 컨트롤러와 인터페이스를 수행하였다. 따라서 언제, 어디서나 CNC 공작기계를 실시간 모니터링 및 제어하며, 알람발생시 PDA폰으로의 자동 통지가 가능한 양방향 통신도 가능한 시스템을 구현하였다.

2. 연구동향

현대 사회는 유비쿼터스 등의 말들이 유행하는 것처럼, 언제 어디서나, 사용자가 하고자 하는 목적을 신속하고 정확하게 이룰 수 있는 산업·생활 환경을 만들고자 하는 요구가 있다. 이러한 요구는 산업계에서 가장 많이 사용되고 있는 생산시스템 및 공작기계분야에서도 요구되고 있다. 특히, 공작기계의 CNC가 개방형 구조를 가지면서 공작기계를 대상으로 한 많은 연구가 이루어지고 있다. 이러한 예로써, 원격 점검 및 제어 등의 연구가 이루어지고 있으며^[1-3], 원격 제어 및 점검 등은 Wireless시대에 들어오면서 무선 환경하에서의 원격 제어 및 감시의 필요가

대두되고 있다^[3-6].

그러나 종래의 공작기계 분야에서의 제어·감시는 기계 자체적인 경우, 그리고 기계와 주변기기와의 경우가 주를 이루어 왔다^[5-6]. 최근에는 컴퓨터와 네트워크가 발전됨에 따라 공작기계 분야에서도 원격지 제어·감시 및 진단에 관한 연구가 많이 진행되고 있다. 하지만 대부분의 경우가 유선 랜 환경에서의 고전적인 Client/Server방식의 Local Domain이 주를 이루며, 아주 최근에 와서야 일부에서 인터넷을 이용한 웹기반 제어·감시가 가능하게 되었다^[4]. 그 이유는 전자상거래 등의 분야와 달리, 아직은 공작기계 및 생산시스템 분야에서는 IT 기술의 접목이 어려운 열악한 현장 환경과 최근에서야 기계기술과 IT기술의 융합 기술에 대한 적용·시도하는 단계에 있기 때문이다. 그리고 이 또한 유선상에서만 가능하며, 지금 이슈가 되고 있는 유비쿼터스 환경의 무선 또는 모바일 환경에서는 거의 이루어지지 않고 있다^[4-5].

따라서 본 연구에서는, 무선 환경하에서 공작기계의 원격 제어 및 감시의 필요가 대두됨에 따라 이러한 요구에 대응할 수 있는 솔루션을 개발하는 것이 연구의 목적이며, 무선 환경 하에서 공작기계를 원격제어 및 점검할 수 있는 방법을 찾고, 이를 언제 어디서든 가능하도록 구현함으로써, 산업계에서 요구하고 있는 유비쿼터스를 기계분야에 접목시킬 수 있는 근간을 마련하고자 한다.

3. 접근방법 및 개요

본 연구의 개요는 현재 유비쿼터스를 추구하는 사회에서, machine(machine-tool: 공작기계)의 상태 및 가공내용을 관찰하고 명령을 언제 어디서나 내릴 수 있게 하기 위하여, 서버(공작기계 컨트롤러, 윈도우 환경)와 PDA 폰 간의 통신을 가능하게 하여, Anywhere, Anytime 개념을 구현한다.

이 구현의 방법으로 기계제어 PC에 TCP/IP 프로토콜을 이용한 서버모듈을 만들었으며, Embedded Visual C++를 이용하여 Arm 기반 PDA와의 모바일용 통신모듈을 구현하였다.

또한, 모바일용 통신모듈과 서버모듈과의 유·무선 통신 과정에서 전송되는 공작기계의 상태(가동여부, NC프로그램 명, Alarm여부, X,Y,Z 축 위치 등) 모니터링 데이터와 공작기계 제어(가동, 일시정지, Emergency Stop, 예약가동, 예약정지 등) 데이터의 실제 값을 기계로부터 추출하고 기계에 적용하기 위하여 OPC(Object Linking and Embedding for Process Control)를 기반으로 실제 공작기계 오퍼레이팅 장치와 인터페이스를 수행하였다.

무선환경 하에서의 기계의 원격 점검 및 제어를 목적으로 언제 어디서든 가능하도록 하기 위하여, 다음과 같은 방법으로 구현 하려고 시도 하였다.

- ◎ 단말기는 PDA 휴대폰을 이용한다.
- ◎ 단말기의 프로그램은 WinCE.Net환경에서 사용될 수 있도록 Embedded Visual C++ 를 이용하여 프로그램 한다.
- ◎ 서버 프로그램은 PDA의 단말 프로그램과 통신하여, 기계의 상태를 보여주고, 기계를 제어할 수 있는 명령을 보낸다.
- ◎ 기계의 실제 모니터링 및 제어를 위하여, 서버용 모듈과 CNC 컨트롤러가 OPC(OLE for Process Control)방식으로 필요한 정보가 인터페이스 된다.

4. 시스템 구성

본 연구의 구성은 아래와 같다.

크게 클라이언트용 모바일 콘텐츠 모듈과 서버 콘텐츠 모듈로 나누어 진다, 서버 콘텐츠 모듈은 상기에서 설명한 바와 같이 Visual C++ 6.0을 이용하여 일반 윈도우 상에서 실행되는 에이전트이다.

클라이언트용 모바일 콘텐츠 모듈은 WINCE ,NET 4.2에서 수행이 가능하도록 Embedded Visual C++ 4.0을 이용하여 완성된 에이전트이며, PDA 폰(예: 삼성 Nexio S155)기기에서 동작하도록 설계 되었다. 서버와 클라이언트의 콘텐츠 모듈간 통신은 무선 전화망과 무선 랜을 통하여 상호간 통신이 이루어 진다.

서버는 기계로부터 기계의 상태, 알람발생 여부, NC 프로그램 명, 각 축의 구동위치 등 모니터링 데이터의 획득과, USB 카메라로부터 영상 획득, 그리고, 통신을 위한 제반 사항을 수행한다. 클라이언트는 서버로부터 데이터를 수신하여 PDA 폰 화면상에 나타내며, 기계를 가동, 정지, 예약가동, 예약정지 등 원격 컨트롤 하기 위한 컨트롤 신호를 서버로 보내는 역할을 한다.

먼저 두 콘텐츠(서버/클라이언트)의 통신을 도식적으로 나타내면, 아래 그림과 같다.

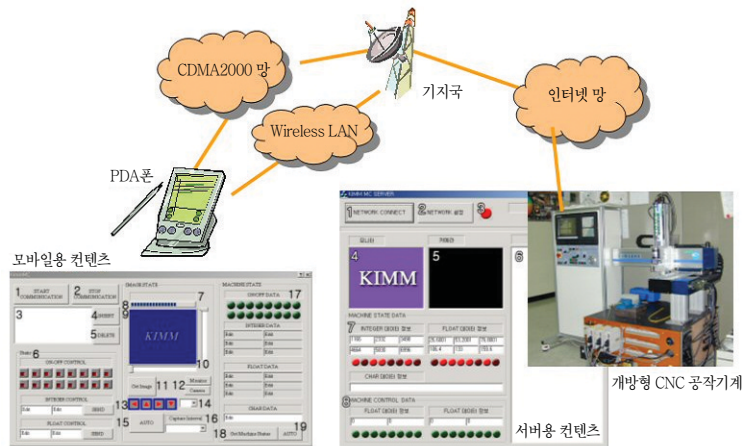


그림 1. 시스템 개략도 및 콘텐츠간 통신

상기 그림의 설명은 다음과 같다.

CDMA 2000 및 무선 망에서 데이터 통신의 방법에 대하여 나타난 것이다. 클라이언트 프로그램은 무선망을 이용하는 단말기인 PDA에서 동작하게 되고, 통신은 CDMA 2000망을 이용하여 기지국과 통신 하게 된다. 기지국은 CDMA 2000망을 통하여 전달된 데이터를 일반 인터넷 망으로 접속을 하게 되고, 일반 인터넷 망에서 접속된 서버는 기계를 동작시키거나 기계의 상태를 다시 인터넷 망을 통하여 기지국으로 전달한다.

즉 클라이언트 프로그램은 Wince상에서 일반 네트워크 프로그램을 프로그램 되어지며, 여기에서, 통신 프로토콜상에서, 기지국으로 데이터를 보내고 다시 기지국에서 서버로 데이터를 보내는 방법으로 통신이 이루어진다. 서버에서 클라이언트로의 데이터 통신 또한 동일한 구조를 가진다

CDMA 2000 및 무선 랜망을 이용하는 서버 측과 클라이언트 측 콘텐츠의 구조는 다음과 같다.

4.1 서버측 컨텐츠

다음 그림은 서버의 구조를 나타낸다.

여기서는 2가지의 서버 모델을 제공한다. 각 방법은 실제 Machine과 인터페이스 및 통신하는 방법상의 차이로 나누어 지며, 통신 방법으로는 OLE(OPC: OLE for Process Control)를 이용하는 방법과, CNC Map의 shared Memory를 이용하는 방법 2가지가 사용된다.

먼저 아래 그림은 OLE를 이용하는 방법이다.

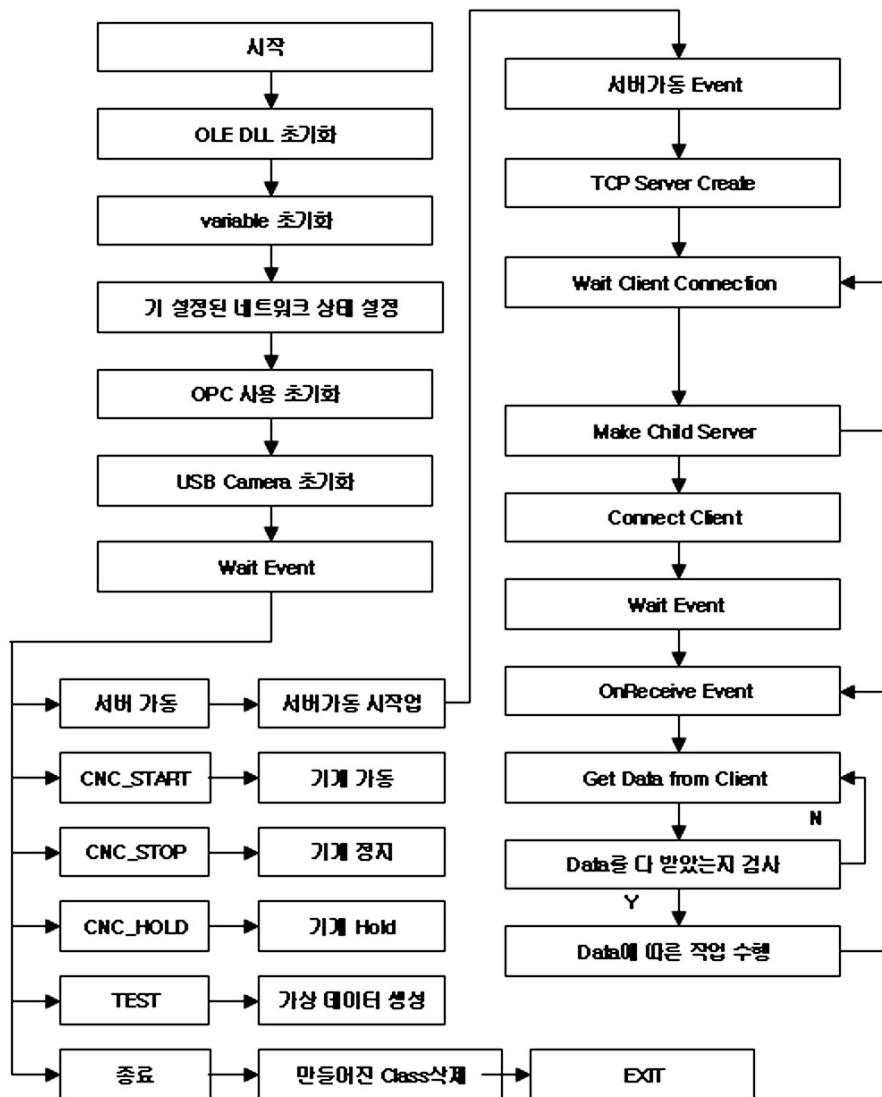


그림 2. 서버측 컨텐츠 OPC 프로그램의 구조

- . OLE DLL 초기화 : OLE를 사용하기 위하여 OLE 초기화를 실행한다.
- . Variable 초기화 : 프로그램에서 사용하는 Variable들을 초기화한다. 이 기능은 CKimmMCServerDlg::OnInitDialog() 함수내에서 이루어진다. 구체적인 사항은 GUI에 사용되는 기본 아이콘 및 변수들을 초기화 하고, OLE를 이용하여 사용되는 변수, 통신을 위한 변수 등을 초기화 하는 단계이다.
- . 기 설정된 네트워크 설정 : 서버 프로그램에서 통신을 하기 위한 상태를 설정한다. 네트워크 망에서 망의 성능에 따라, 한 패킷의 데이터가 너무 많으면, 망에서 통신이 잘 되지 않는 경향이 있어 필요에 따라 패킷의 량을 조절할 수 있도록 하였으며, 포트 번호를 설정 한다. 그리고, 이렇게 설정된 값은 하드디스크에 저장되어, 프로그램 초기화 시 이를 읽어 네트워크 상태를 초기화 한다.
- . OPC 사용 초기화 : OLE를 이용하여 OPC프로그램으로 Machine과 통신을 하게 하기 위한 OPC초기화를 수행한다. OPC프로그램에서 사용되는 변수를 초기화 하고, 필요한 Class를 생성한다.
- . USB 카메라 초기화 : 서버가 NT가 아닌 경우 USB카메라를 이용한 영상 통신이 가능하게 하기 위하여(NT의 경우, USB 카메라가 지원이 되지 않는다. 만약 영상 통신이 NY에서 이루어지게 하려면, PCI보드 타입의 영상 입력 보드를 이용하여 프로그램을 작성하여야 한다.) USB카메라를 초기화 하고 화면에 PREVIEW화면을 디스플레이 한다.
- . Wait Event : GUI부분을 화면상에 나타나 있는 상황에서, USER의 명령을 기다린다. 각 명령은 “서버가동”, “CNC START”, “CNC_STOP”, “CNC_HOLD”, “TEST”, “종료” 이벤트로 구성된다.
- . 서버가동 : 서버의 통신 프로그램을 가동하여, 클라이언트로부터 데이터 전송 요청을 기다리게 된다. 서버는 클라이언트로부터 데이터 요청이 들어오면, 해당 클라이언트와의 통신을 위하여 새로운 클래스를 만들어 통신을 전달 시킨다. 새로 만들어진 클래스는 해당 클라이언트 프로그램과 정해진 규약에 따라 통신을 한다. 통신 규약은 5-4장 통신 규약에 기술한다.
- . CNC START : CNC 운전시작 명령을 내린다.
- . CNC_STOP : CNC 중지 명령을 내린다.
- . CNC_HOL : CNC 운전을 HOLD 시키는 명령을 내린다.
- . TEST : CNC와 통신이 불가능한 경우, 통신 테스트를 위하여, 임의의 값을 발생하여, 통신 테스트를 수행한다.
- . 종료 : 프로그램을 종료한다.

상기에서 설명된 콘텐츠 프로그램은 OLE를 이용하는 OPC프로그램으로 Machine의 상태를 읽어 오거나 동작의 명령을 내리는 구조의 프로그램이다. Shared memory를 이용하는 프로그램의 경우 RTX(Realtime Kernel) API를 이용하여 프로그램 되며, OPC해당 부분에서 RTX를 이용한 프로그램으로 대체 된다.

4.2 클라이언트측 콘텐츠

다음 그림은 클라이언트 콘텐츠를 나타낸다.

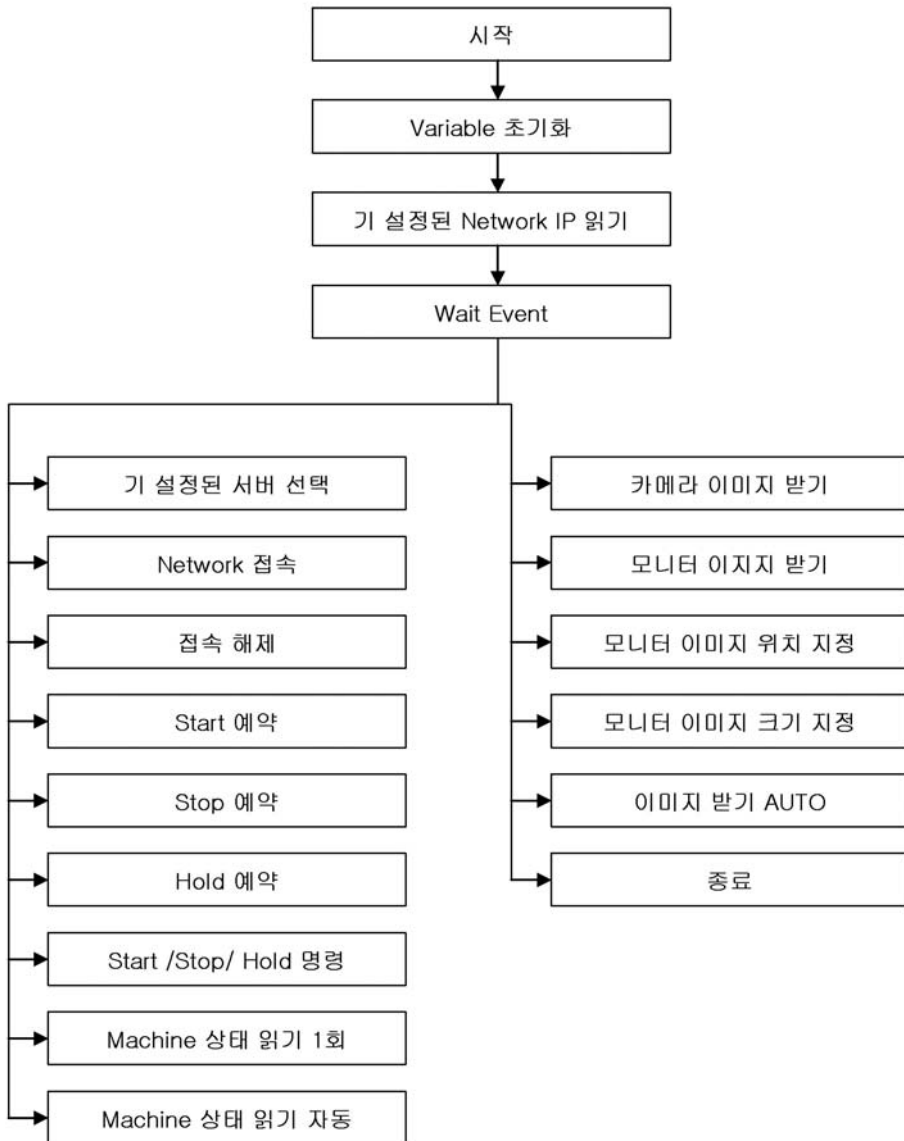


그림 3. 클라이언트측 콘텐츠 프로그램 구조

클라이언트 프로그램은 서버 프로그램에 접속하여, 서버의 정보를 읽어 오고, 필요에 따라 서버로 명령을 내리는 구조로 되어 있다.

- variable 초기화 : 시작단계에서 필요한 변수들의 초기화
- 기설정된 Network IP 읽어오기 : 등록 되어진 IP 및 포트를 읽어오는 작업을 수행한다. 프로그램에서는 사용하는 서버의 IP 및 포트를 등록할 수 있도록 하고 있으며, 등록 된 IP를 읽어와서 List Box에 표시하게 된다. 표시된 IP중 하나를 선택하여 통신을 시작하게 된다.

- 기 접속된 서버 선택 : List Box에서 접속할 서버를 선택한다.
- Network 접속 : 서버로 접속을 시도한다. 초기 접속 시도가 이루어 지면, 자동으로 서버의 기본 데이터를 읽어와서 설정하게 된다. 이 명령이 성공적으로 이루어 지고 난 후, 다른 명령들이 동작하게 된다.
- 접속 해제 : 접속을 해제 한다.
- Start/Stop/Hold 예약 : 예약 실행을 명령하기 위한 동작으로, START 예정시간/ STOP 예정 시간 / HOLD 예정시간 등을 설정하여 서버에 명령을 줌으로써, 각 명령에서 설정된 시간에 맞추어 명령이 실행 된다.
- Machine상태 읽기 1회 : Machine의 상태를 1회 읽어 온다.
- Machine상태 읽기 자동 : Machine의 상태를 주기적으로 읽어 온다.
- 카메라 이미지 받기 : USB카메라가 장착되어진 경우, USB카메라로부터 176x144(QCIF) 이미지를 전송 받아 보여 준다.
- 모니터 이미지 받기 :서버 모니터의 이미지를 받아 보여준다.
- 모니터 이미지 위치 지정 : 서버의 모니터 해상도는 데이터 송수신에 비해 큰 데이터 량을 가지고 있다. 따라서 모니터의 일부 관심 부분을 모니터링 하게 되는데, 이 경우, 받을 모니터 이미지의 영역을 설정하게 된다.
- 모니터 이미지 크기 지정 : 모니터로부터 받을 이미지의 크기를 설정한다. 전체 모니터의 크기 정보는 초기화 시 통신을 통하여 얻어 온다.
- 이미지 받기 오토 : 이미지를 주기적으로 받아 온다.
- 종료 : 프로그램을 종료한다.

5. Machine(개방형 CNC 공작기계)과의 Interface

다음 그림은 서버측 컨텐츠가 인터페이스 해야 될 기계(Machine), 즉 여기서는 개방형 CNC 공작기계의 내부 컨트롤러의 구조를 나타낸다.

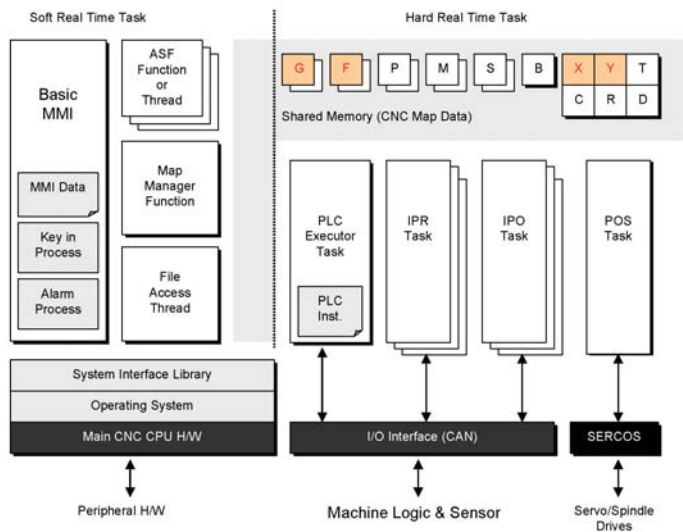


그림 4. 서버측 컨텐츠와 인터페이스되는 CNC공작기계 컨트롤러 구조

구조를 보면, PLC의 입출력 신호인 X, Y 신호, PLC로부터 CNC로의 입력신호인 G 신호, CNC로부터 PLC로의 입력신호 F 신호 등의 주요 신호가 들어 있는 CNC 내부 Map data가 오픈되어 있는 구조를 가진다. 따라서 개방형 구조를 가진 CNC 컨트롤러상에서 CNC의 웨어드메모리(Shared Memory) 영역을 공유하고 알람로그 데이터를 공유함으로써 점점 신호정보와 CNC 알람 정보를 추출하여 실제 공작기계의 주요 상황 정보를 실시간으로 공유가 가능한 형태로 서버측 컨텐츠와 CNC 운영시스템이 하나의 플랫폼 위에 탑재된다.

다음 그림은 서버측 컨텐츠가 실제 기계의 상태, 즉, 가동여부, 알람상태, NC프로그램 정보, Stage X,Y,Z 위치 등의 실제 Machine정보의 추출과 원격지 가동, 정지, EStop(긴급정지) 및 예약가동/정지 등의 정보를 적용시키기 위하여, 기계 컨트롤러와 인터페이스하기 위한 방법을 나타낸다.

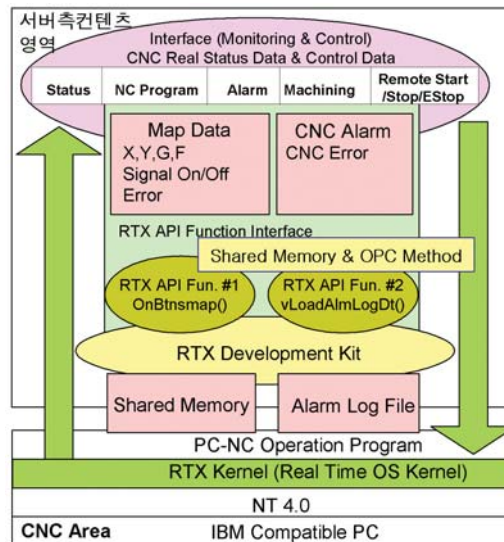


그림 5. 서버측 컨텐츠와 실제 CNC 공작기계와 Interface 구조

인터페이스를 구체적으로 살펴보면, 보통 CNC의 멀티태스킹을 위해 기반 플랫폼으로 사용되는 Realtime Kernel 에서 제공하는 RTX API Functions을 이용한 인터페이스 방식으로 Map Data와 CNC Alarm Data로의 접근이 가능하게 된다. 또한, OPC 방식으로 함수간 상호 파라미터의 전달이 가능하다.

6. 시스템 기능 및 기능테스트

본 연구의 주요 기능은 다음과 같다.

- Random 상태 생성 및 기계 PC의 상태 획득.
- 화면 캡처 및 전송기능(화면의 일부만 캡처 기능)
- USB카메라 캡처 및 전송 기능
- 실제 머신 상태 및 가공 정보 데이터 전송기능
- 외부 컨트롤 신호 전송 및 제어기능

주요 기능별 설명은 다음과 같다.

- Random 상태 생성은 실제 기계의 데이터 획득이 용이하지 않은 경우, 테스트 용으로 사용하는 것으로, Random으로 기계상태를 만들어 통신이 가능한지 테스트 하는데, 사용된다. 실 기계의 상태를 획득하는 것은 OPC를 이용하여 데이터를 획득하는 것으로, 초기 서버 프로그램의 실행 시 이를 자동으로 체크하여, 데이터를 획득하게 된다.
- 화면 캡처 기능의 경우, 서버가 설치된 PC의 화면을 캡처 하는 것으로, 클라이언트로부터 명령을 받게 된다. 클라이언트로부터 화면 캡처 명령을 받을 때, 화면의 캡처 사이즈 및 위치를 캡처 인자로 주게 된다. 서버는 이 명령에 따라, 화면의 주어진 위치부터 명령되어진 사이즈 만큼, 화면을 캡처 하여 JPEG으로 압축 후, 압축 된 데이터를 클라이언트로 보내주게 된다. 디폴트 영상 사이즈는 176x144(QCIF)사이므로, 영상의 사이즈가 클수록 데이터의 양이 많아지므로 주의하여 사용하여야 한다. 이렇게 전송된 데이터는 서버 화면에 보여지게 된다.
- USB 카메라 생성 및 전송기능은 PC에 USB카메라를 접속시킬 경우, 초기 시작 상태에서 USB카메라를 Initialize하여, 화면을 서버에 보여주게 된다. 화면 캡처와 마찬가지로, 클라이언트로부터 명령에 따라, 영상을 176x144(QCIF Size)로 캡처 하여, 압축 후 전송하게 된다. USB 카메라의 경우 현재까지, Windows NT 이외의 Windows환경에서는 사용가능하나, NT의 경우, OS에서 지원하지 않고 있다.(현재 서버가 동작되어 지는 PC는 NT 환경임)
- 실제 머신 상태 및 가공 정보 데이터 전송기능은 OPC 또는 random으로 생성되어진 기계 데이터를 클라이언트의 요구에 따라 클라이언트에게 데이터를 보내 주게 된다. 보내 주는 데이터는 기계상태, 알람상태, 가공 NC프로그램 정보, Stage 축 위치정보, Feed 상태 등이다.
- 외부 컨트롤 신호 전송 및 제어기능은 기계의 동작 정지/시작/Hold 등의 명령을 클라이언트로부터 받아 이를 서버 프로그램이 OPC를 통하여 수행하게 된다.

6.1 서버 측 컨트롤 기능 상세설명



그림 6. 서버측 컨트롤 메인프레임

서버 프로그램을 초기 실행에서, OPC의 연결, 즉 기계와의 연결에 의한 데이터 획득이 가능한지 체크하게 된다. 데이터 획득이 가능할 경우, 프로그램은 실행이 되고 가능하지 않은 경우는 Warning Message가 보여진다. OPC와의 접속 실패에 의한 메시지 확인 후 또는 OPC와의 접속 성공 후 다음 그림과 같은 본 화면이 보여진다.

- ▶ “1. 서버 가동”은 서버의 가동을 시작하는 버튼으로 가동이 되는 경우, “3. 가동상태”에 불이 들어오게 된다.
- ▶ “2번 전송설정”, 이는 1번 서버 가동 전에 설정하는 것으로, 전송 시 전송 환경을 설정하게 된다. 패킷의 크기는 한번 전송에 사용될 데이터의 단위로서 BYTE 단위이다. 지연시간은 각 패킷의 전송 사이 지연시간이다. 이 두 가지를 설정하는 이유는 많은 양의 데이터 전송 시 전송의 안정성을 확보하기 위한 것으로, 패킷의 크기가 너무 크거나, 지연시간이 너무 작은 경우, 데이터 전송의 오류를 불러 오게 된다. 포트 번호는 데이터 전송 포트로서 클라이언트를 이 포트를 사용하여 통신하여야 한다.
- ▶ “3. 가동상태”는 서버측 컨텐츠가 정상 가동이 되는 경우에 불이 들어오게 된다.
- ▶ “4. 테스트”는 OPC서버와의 연결이 실패된 경우, 데이터 전송을 테스트 하기 위한 것으로, Random 데이터를 발생한다.
- ▶ “5. 화면 캡처”는 현재 서버가 가동 중인 PC의 화면을 캡처 하여 보여주는 창으로, 클라이언트의 요구에 의해 캡처 되고 캡처 사이즈와 상관 없이 창에 맞게 표시된다. 따라서 서버 관리자는 현재 어떠한 화면이 캡처 되어 전송되었는지를 알 수 있다.
- ▶ “6. 영상 캡처”는 USB카메라가 설정되어져 있는 경우, USB카메라의 화면을 보여 준다. 이 화면 역시 클라이언트의 요구에 의해 화면이 캡처 되어 전송되어 진다. 서버에는 화면 캡처와 다르게 연속된 화면이 보여진다.
- ▶ “7. 상태획득 수”는 현재 OPC 또는 Random을 통하여, 몇 번의 데이터가 획득 되었는지를 표시하는 창으로써, 데이터의 변화가 거의 없는 경우 서버의 동작을 잘 알 수 없으므로, 현재 데이터 획득이 이루어 지고 있는지를 확인할 수 있다.
- ▶ “8. 기계상태”는 현재 기계로부터 획득되어진 데이터의 상태를 보여 준다. 보여 주는 데이터는 축 1,2,3의 위치 정보, RPM정보, Feed Rate정보, 조작모드, 설정된 NC프로그램 정보, 자동운전 상태 정보, 기계의 On/Off 정보, 알람 정보 이다.
- ▶ “9. 명령 예약”은 클라이언트로부터 예약된 명령이 있는지를 보여 주는 것으로, 날짜와 시간에 따른 예약을 보여 준다. 이러한 예약은, On/Off 및 HOLD시간을 설정할 수 있다. 예약이 다시 들어 올 경우 최종 예약에 의해 수행 된다.
- ▶ “10. 접속수”는 현재 몇 명의 클라이언트가 접속 되었는지를 보여주는 창으로써, 현재 프로그램으로 최대 5명의 클라이언트가 접속 가능 하도록 프로그램 되어 있다.
- ▶ “11.접속 정보”는 클라이언트의 접속 및 접속 해제 정보를 보여 주며, 예약 정보, 컨트롤 정보 등을 보여 주게 된다. 이러한 정보는 시간과 함께 보여 주므로, 어떠한 클라이언트가 언제 접속하여 언제 연결을 끊었는지, 그리고, 어떠한 명령을 언제 내렸는지를 판단 할 수 있다.
- ▶ “12. start”는 기계의 동작을 시작하라는 명령이다.
- ▶ “13. HOLD”는 기계에 HOLD명령을 내린다.
- ▶ “14. STOP”은 기계에 STOP명령을 내린다.
- ▶ “15. 종료”는 프로그램을 종료한다.

6.2 클라이언트측 콘텐츠 상세 기능설명

클라이언트 측은 PDA상에서 구현되어지는 프로그램으로, Embedded Visual C++ 4.0을 이용하여 프로그램 하였다. 이 프로그램의 전체적인 특징은 서버와의 연결, 서버로부터의 데이터 획득, 서버로의 명령 전달, 서버로부터 영상 데이터의 획득 및 디코딩 과 이와 수반되는 기능들이다.

먼저 클라이언트 측 메인 화면은 아래와 같다.

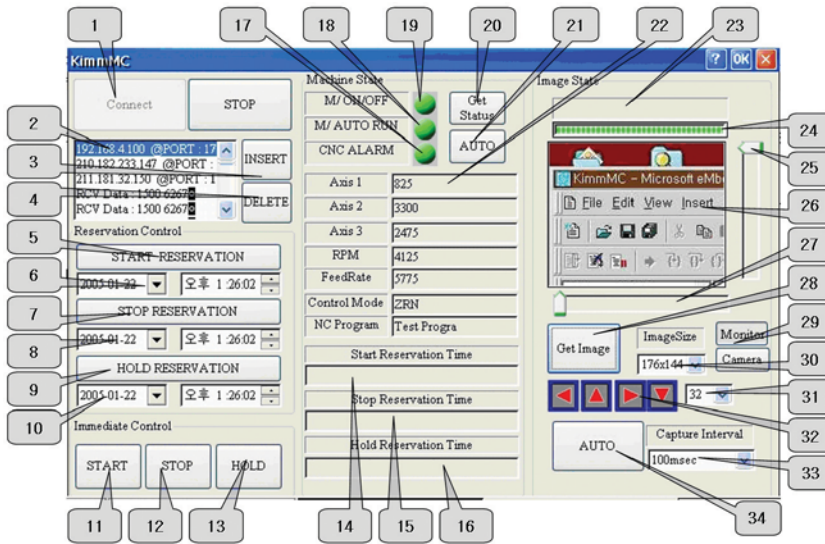


그림 7. 클라이언트측 콘텐츠 메인프레임

- ▶ 1. connect : 서버와의 연결을 시도한다. 이 버튼을 선택하기 전에 “2 List”에서 접속하고자 하는 서버를 선택한 후 접속을 시도한다.
만약 접속이 실패 하면, 접속 실패의 메시지가 나오게 된다. 접속이 성공적으로 이루어지면 필요한 명령을 수행할 수 있다.
- ▶ 2. List Box : List Box에는 “3. insert”를 통하여 입력되어진 서버의 주소들을 표시하고, 표시된 주소의 선택에 의하여, 접속 서버를 결정한다. 리스트 박스에 표시되는 형식은 “xxx.xxx.xxx.xxx @PORT:yyyy”인데 xxx는 IP ADDRESS를 나타내며, yyyy는 포트 번호를 나타낸다. 여기에 표시되는 IP들은 프로그램 실행 파일이 존재하는 디렉터리의 IP_LIST파일에 저장되어 있다.
- ▶ 3. INSERT : List Box에 IP 어드레스 및 포트를 저장한다. IP 및 포트 번호를 선택하여 SAVE를 선택하면, 지정된 IP및 포트가 IP_LIST파일에 저장되면서, LIST에 삽입된다.
- ▶ 4. delete: list Box에 표시되는 IP 및 포트 번호를 삭제한다.
- ▶ 5. START RESERVATION: “6”에 설정되어진 시간에 서버에서 “START”명령을 수행하도록 하는 명령을 서버로 보낸다.
- ▶ 6. 서버의 START 예약 시간을 설정한다.

- ▶ 7. STOP RESERVATION: “8”에 설정되어진 시간에 서버에서 “STOP”명령을 수행하도록 하는 명령을 서버로 보낸다.
- ▶ 8. 서버의 STOP 예약 시간을 설정한다.
- ▶ 9. HOLD RESERVATION : “10”에 설정되어진 시간에 서버에서 “HOLD”명령을 수행하도록 하는 명령을 서버로 보낸다.
- ▶ 10. 서버의 HOLD 예약 시간을 설정한다.
- ▶ 11. START : 서버의 START명령을 바로 실행하도록 명령을 서버로 보낸다.
- ▶ 12. STOP : 서버의 STOP 명령을 바로 실행하도록 명령을 서버로 보낸다.
- ▶ 13. HOLD : 서버의 HOLD 명령을 바로 실행하도록 명령을 서버로 보낸다.
- ▶ 14. Start Reservation Time : 서버에서 이미 예약되어진 “START”명령 시간을 보여 준다.
- ▶ 15. Stop Reservation Time : 서버에서 이미 예약되어진 “STOP”명령 시간을 보여 준다.
- ▶ 16. Hold Reservation Time : 서버에서 이미 예약되어진 “HOLD”명령 시간을 보여 준다.
- ▶ 17. M/ On/OFF : 서버에서 획득되어진 기계의 On/OFF 상태를 표시한다.
- ▶ 18. M/ AutoRun : 서버에서 획득되어진 기계의 Auto Run 상태를 표시한다.
- ▶ 19. CNC Alarm : 서버에서 획득되어진 기계의 Alarm 상태를 표시한다.
- ▶ 20. Get State : 서버로부터 데이터를 요청하여 기계의 상태를 보여준다.
- ▶ 21. AUTO : 서버로부터 주기적으로 데이터를 획득하여 보여준다.
- ▶ 22. 서버로부터 받은 데이터를 표시한다.
- ▶ 23. 서버로부터 스크린 영상을 받을 경우, 스크린의 시작 좌표를 표시한다. 스크린의 데이터는 매우 큰 데이터 이므로 영상 전송 비용 등을 감안하여, 영상의 사이즈를 설정할 수 있도록 하였으며, 따라서, 임의의 스크린 위치로부터 영상사이즈 만큼 받을 수 있도록 프로그램 되었다.
- ▶ 24. 영상 전송은 데이터의 양이 많으므로 한 장의 영상을 받기 위하여 시간이 걸리게 된다. 이 표시는 영상 전송 정도를 보여 준다.
- ▶ 25, 27. 스크린의 영상을 받을 경우, 영상의 시작 점을 설정한다. 이렇게 설정된 값은 23에 표시 된다.
- ▶ 26. 전송되어진 영상을 보여 준다.
- ▶ 28. 하나의 영상을 획득하기 위한 명령을 서버로 전송한다.
- ▶ 29. 전송 받을 영상의 종류를 선택한다, USB CAMERA(176x144)인지, 스크린의 영상인지를 결정한다.
- ▶ 30. 영상 전송을 스크린으로 설정한 경우, 받을 스크린 영상의 크기를 설정한다. 영상의 크기가 크면, 데이터의 전송량이 많아지고, 에러율이 증가한다. 본 프로그램에서는 176x144만을 지원한다.
- ▶ 31. 32에서 사용되는 STEP을 설정한다.
- ▶ 32. 31에 설정된 스텝만큼 현재의 스크린 영상에서 이동하여 영상 전송을 받는다. 이 경우, 버튼을 클릭함으로써, 데이터 전송 명령을 내리게 되는데, 초기 스크린의 주요 부분을 탐색할 경우 주로 사용된다.
- ▶ 33,34. 영상전송을 자동으로 할 경우, 주기적으로 데이터를 받을 수 있으며, 이 주기를 선택할 수 있다. 그러나 주기가 데이터 전송 속도 보다 작을 경우, 데이터 전송 속도에 맞춰 전송 되어진다.

6.3 기능테스트

개발 시스템의 기능 테스트를 위한 대상 기계는 그림과 같은 개방형 CNC 구조를 가지는 3축 밀링머신이다. 실험 시나리오는 다음과 같다. 대상 기계의 컨트롤러는 앞서 언급했듯이 개방형 구조를 가지며, 기계는 X, Y, Z의 Stage 3축과 Spindle이 있는 공작기계이다.

실험 시나리오는 다음과 같다.

- 기계 모니터링 : 정지/가동 및 알람상태 모니터링
- 작업자의 기계 작동 : 현장에서 Auto Model에서 NC 프로그램에 의한 자동운전
- 상태 모니터링 : 원격지 PDA폰에서 가동유무, NC프로그램명, Alarm여부, Stage의 X,Y,Z축 위치 등 모니터링
- 원격지에서 일시정지 명령 : PDA폰에서 무선랜이나 무선 전화망 이용한 Feed Hold 명령
- 원격지에서 재가동 명령 : PDA폰에서 무선랜이나 무선 전화망 이용한 Restart 명령
- 원격지에서 긴급정지 명령 : PDA폰에서 무선랜이나 무선 전화망 이용한 EStop 명령
- 상태 모니터링 : PDA 폰에서 Emergency Alarm 모니터링
- 원격지에서 예약가동 : PDA폰에서 무선랜이나 무선 전화망 이용한 예약시간 설정 후 가동명령
- 상태 모니터링 : PDA 폰에서 예약 시간에 기계가동 유무 확인
- 원격지에서 예약정지 : PDA폰에서 무선랜이나 무선 전화망 이용한 예약시간 설정 후 정지명령
- 상태 모니터링 : PDA 폰에서 예약 시간에 기계작동 정지 확인
- Bidirectional communication 테스트 : 클라이언트의 요구 없이도, CNC 공작기계 알람발생시 클라이언트용 PDA폰으로의 자동 통지기능 확인

이상의 실험을 통하여, 무선 환경에서도 언제 어디서나 공작기계의 상태(가동여부, NC프로그램명, Alarm여부, X,Y,Z 축 위치 등)를 모니터링 하였으며, 공작기계를 실시간 및 예약시간에 제어(가동, 일시정지, Emergency Stop, 예약가동, 예약정지 등)를 하였다. 따라서 KIMM 모델플랜트에서의 운용 결과, 업무 시간 외에도 유연성 있는 기계의 제어와 비상시 빠른 조치로 기계 가동율을 기존 보다 50%이상 높였다.

7. 결론

본 연구에서는 u-Machine을 위한 ubiquitous 기반의 CNC 공작기계 모바일 제어 및 감시기술을 개발하였다. 모바일 단말을 일반 휴대전화와 컴퓨터의 역할을 겸할 수 있는 PDA 폰을 대상으로 하여, 휴대성이 있으면서도 화면의 size를 늘임으로써, 많은 양의 데이터를 화면에서 수용할 수 있게 하였으며, 소프트웨어 컨텐츠 개발 방법 또한 WinCE .Net환경에서 Embedded Visual C++를 이용함으로써 BREW 를 이용할 때의 제약 조건, 즉, 소프트웨어 컨텐츠를 완성한 후, 이를 폰에 Up-Load 하여야 하는데, 이를 수행하기 위하여는, QUALCOMM 또는 서비스 업체(예 : 016) 등에 등록이 되어야 하는 라이선스 제약조건을 없앨 수 있었다.

그리고, 무선 전화 통신망 뿐만 아니라, 무선랜 망을 이용할 수 있게 됨으로써, 무선랜이 가능한 곳에서는 무선랜 망을 이용할 수 있도록 할 수 있었다. 따라서 언제 어디서든 전화망과 무선랜을 이용함으로써 활용의 폭이 커졌다.

또한, 개발 콘텐츠는 개방형 CNC 공작기계 컨트롤러와 정보 입출력을 위해 실제 인터페이스를 수행하였다. 또한, 클라이언트의 요청이 없을 때도, CNC 공작기계 알람발생시 클라이언트용 PDA폰으로 자동 통지 하는 양방향 통신이 가능하도록 하였다.

이상의 연구를 통하여 특정 라이선스 규제 없는 접근방법 및 시스템 구성으로 CNC에의 콘텐츠화 및 u-Machine 개발을 위한 기반을 구축하였다.

❁ 참고 문헌

- [1] Erol, N. A., Altintas, Y. and Ito, M. R., "Open system architecture modular tool kit for motion and machining process control", IEEE/ASME Transactions on Mechatronics, Vol. 5, No. 3, pp. 281-291, 2000.
- [2] Jung, M. S., Park, H. S. and Kim, B. S., "Architecture of web-based real-time monitoring system", Journal of ICASE, 7(7), pp. 632-639, 2001.
- [3] Kim, D. H., Kim, S. H. and Koh, K. S., "A Scheme for an Internet-based Checking Method of Machine-Tools with Variant CNC Architecture", Journal of Mechanical Science and Technology, Vol. 19, No. 1, pp. 97-105, 2005.
- [4] Kim, D. H., Kim, S. H. and Koh, K. S., "CNC-implemented Fault Diagnosis and Web-based Remote Services", Journal of Mechanical Science and Technology, Vol. 19, No. 5, pp. 1095-1106, 2005.
- [5] Wright, P. K., "Principles of open-architecture manufacturing", Journal of Manufacturing Systems Vol. 14, No. 3, pp. 187-202, 1995.
- [6] Yellowley, I. and Pottier, P. R., "The integration of process and geometry within an open architecture machine tool controller", International Journal of Machine Tools & Manufacture, Vol. 34, No. 2, pp. 277-293, 1994.



김 동 훈

· 한국기계연구원 지능기계연구센터 책임연구원
· 관심분야 : M2M기계복합지능화, Machine Vision
· E-mail : kdh680@kimm.re.kr



송 준 업

· 한국기계연구원 지능기계연구센터 책임연구원
· 관심분야 : 지능형정밀생산시스템, 반도체공정
장비 지능화
· E-mail : sjy658@kimm.re.kr