

# 디스플레이 검사 장비를 위한 OPC Client 개발

## An Implementation of OPC Client for Display Inspection System

한 창 호\*, 오 춘 석, 유 영 기

(Changho Han, Choonsuk Oh, and Youngkee Ryu)

**Abstract :** OPC is the preferred communication standard for sharing process control data at all levels of the enterprise and becomes the global standard, sends/receives the data with PLC programs of inspection system via network, and manages various data using OPC server. In this paper, we introduce the developed OPC client system for FPD inspection system with OPC server. To control inspection system through OPC server, we have developed libraries which consist of stage driver, switch driver, microscope driver, image processing, alignment, and task management modules. Also we have developed the test program to verify all functions precisely. We describe the mechanism of inspection system, OPC connection method, and programs.

**Keywords :** inspection, OPC server, OPC client, control library

### I. 서론

디스플레이 검사장비 산업에도 급속도로 고도화, 정밀화, 대형화 추세로 발전하고 있어 시스템을 제어하기가 점점 복잡해 지고 있다. 이러한 시장 환경의 변화에 즉각적으로 대응 할 수 있는 자동화 생산체제로의 전환을 통하여 기업 경쟁력을 확보하고자 하는 많은 노력이 추진되고 있다. 이러한 자동화 시스템의 가장 큰 과제는 설비 시설간의 정보 및 데이터의 통합, 관리이다. 점점 다양하고 복잡해져 가는 설비 시설의 추세에서, 장비 의존적인 시스템 제어 방식은 벤더의 드라이버에 의존적이고, 개발 및 유지 보수가 어렵고, 고가라는 문제점이 있다. 이를 해결하기 위해 자동화 설비 관련 업체와 기관들이 프로세스 컨트롤의 표준에 대해 논의하게 되었고, 그 결과 OPC(OLE for Process Control)가 발표되었다. OPC는 다양한 프로세스 컨트롤 분야의 통신 표준 메커니즘을 위해 만들어졌으며, 마이크로소프트의 COM(Component Object Model)/DCOM(Distributed Component Object Model)기술을 이용한 자동화 장비와 소프트웨어 사이의 통신을 위한 규격으로서 여러 산업용 어플리케이션이 신속하고 안전하게 데이터를 공유할 수 있도록 표준화된 방식을 제공한다[1]. 이러한 배경으로 탄생한 OPC 표준은 이미 미국, 유럽, 일본, 한국 등 많은 국가에서 사용하고 있으며, 이미 산업계에서 안정화되고 있는 실정이다.

### II. 본론

#### 1. OPC 구조

OPC에서 OLE란 Object Linking and Embedding으로 윈도 우즈에서 데이터간의 연결 방법에 대한 내용이다. OLE 입력 방법에는 링크 모드와 임베드 모드가 있다. 링크 모드 데이터는 서버 어플리케이션에서 관리하며, 데이터의 링크

\* 책임저자(Corresponding Author)

논문접수 : 2008. 1. 31., 채택확정 : 2008. 2. 24.

한창호 : 선문대학교 전자공학과(liberman@paran.com)

오춘석, 유영기 : 선문대학교 전자정보통신학부

(csoh@sunmoon.ac.kr/ryu@sunmoon.ac.kr)

정보만 저장이 되며, 서버 어플리케이션에 있는 데이터는 여러 어플리케이션에서 참조가 가능하다. 임베드 모드 데이터는 OLE 컨테이너에서 관리하며, 모든 데이터가 저장되고 다른 어플리케이션에서는 참조할 수 없다. 공정 제어에서 발생하는 모든 데이터를 OLE를 통해 데이터 공유를 하게 되며, 정해진 규약에 의해 모든 데이터 통신이 가능해진다. 또한, OLE는 COM, DCOM 기반으로 설계되어 있기 때문에 분산처리가 가능하고 네트워크상에서의 작업이 수월한 장점을 가지고 있다[2,3].

OPC에서 가장 중점적으로 표준안을 작성하는 부분은 Data Access 3.0, Historical Data Access 1.2, Alarms and Events 1.1, Security 1.0, Batch 2.0, XML DA 1.01 등이 있다. OPC DA는 실시간으로 데이터를 처리하는 규정이며 가장 많이 사용되는 부분이라 할 수 있고, HDA는 분석을 위해 시간별 공정 데이터를 처리하는 규정이며 장비나 네트워크 시스템이 복잡해지고 다양해지면서 방대한 양의 데이터를 관리하기 위해 별도의 history정보를 관리한다. OPC AE는 알람과 이벤트의 교환이나 정보를 처리하는 규정이며, OPC DX는 OPC 서버간에 데이터를 교환하는 규정이며,

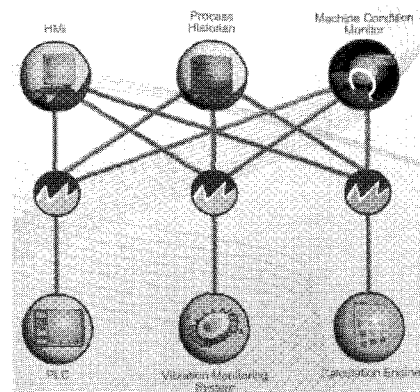


그림 1. OPC 연결구조.

Fig. 1. OPC connection structures.

OPC XMLDA는 모든 OS에서 이용하도록 공정 제어 데이터를 캡슐화하는 규정을 말한다.

그림 1은 OPC 개념에 대한 개괄도를 나타내며, 그림에서 보듯이 PLC 장비의 처리 속도가 늦은 관계로 많은 Client 측에서 데이터를 입력받아 처리할 수 있는 구조로 되어 있다. OPC는 클라이언트와 서버로 나뉘어 동작하며, 서버에서는 제어장비로부터 데이터를 실시간으로 받아 저장하며, 클라이언트에서 요청시 서버의 데이터를 넘겨준다. 따라서 많은 클라이언트의 데이터 요청에도 별 무리가 없이 전달하게 된다.

2. OPC server 구성

그림 2는 OPC COM 구조를 보여주고 있다. Automation Client를 OPC Automation Wrapper가 캡슐화해서 접근하고 있다. 이것은 OPC Custom Interface Server와 COM으로 연동되어 사용된다[4].

기본적인 OPC에 관련된 인터페이스는 표준으로 설계되어 있어 OPC Foundation에서 공개하고 있으며, 제어관리 프로그램은 이 표준 인터페이스를 이용하여 프로그램을 설계해야 한다. 본 논문의 시스템에서는 OPC DA 2.0 표준을 사용하여 개발되었다. OPC DA 구조를 살펴보면 Server는 Group을 관리하며, Group은 Item들을 관리한다. OPC Browser는 서버에 있는 Item들의 정보를 보여주는 객체이다 [5]. OPC DA에서 제공하는 인터페이스는 다음과 같은 종류가 있다.

- IO[PCDataCallback; 데이터 이벤트 처리 인터페이스
- IOPCServer; 서버 인터페이스
- IOPCServerPublicGroups; 공용그룹관리 인터페이스

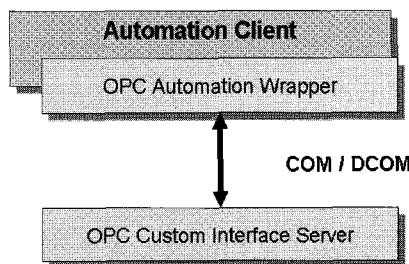


그림 2. OPC COM 구조 개요도.  
Fig. 2. COM structure outline of OPC.

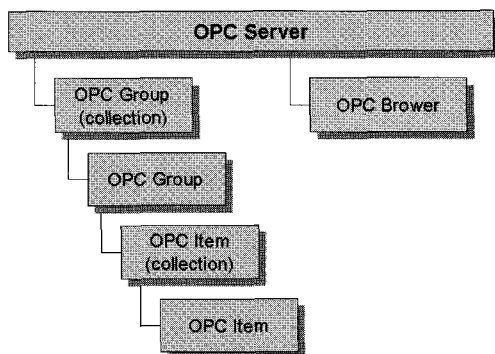


그림 3. OPC 자동화 서버 객체 모델.  
Fig. 3. Automation server object model of OPC.

- IOPCGroupStateMgt; 그룹 상태관리 인터페이스
- IOPCItemMgt; 아이템 관리 인터페이스
- IOPCItemProperties; 아이템 특성 인터페이스
- IEnumOPCItemAttributes; 아이템속성의 열거인터페이스
- IOPCBrowseServerAddressSpace; 브라우저 인터페이스
- IOPCSyncIO; 동기 입출력 인터페이스
- IOPCAsyncIO; 비동기 입출력 인터페이스
- IOPCAsyncIO2; 비동기2 입출력 인터페이스

각각의 모든 인터페이스는 COM을 통해 메소드를 처리하게 된다.

3. OPC 클라이언트의 구성

본 논문에서 개발한 전체 시스템에 대한 개요도는 그림 4와 같이 검사장비를 제어하는 PLC의 명령어를 통신하는 OPC 서버 부분, OPC 서버로부터 명령어를 전달하는 OPC 클라이언트 부분, OPC 클라이언트로부터 태그 명령어들을 받아 프로세스를 처리하고, 명령어 처리에 대한 관리를 하는 TaskManager 부분으로 세분화 된다. OPC 서버/클라이언트는 PLC와 PC제어의 통신을 담당한다. PLC로부터 현재 장비의 상태에 대한 데이터를 실시간으로 취득하고 이를 PC제어의 동작에서 활용한다.

4. 검사장비 구성

검사 장비를 제어하는 역할은 주로 PLC가 담당하며 본 시스템에서 실험하는 기종은 미쓰비시사의 MELSEC-Q시리즈이다. 이를 구동하여 데이터를 통신하기 위한 드라이버로써 KepWare사의 KepServer 프로그램을 사용하였다[6]. 이 드라이버는 OPC 서버로 구성되며 이기간의 PLC도 다룰 수 있다. 또한 OPC 표준화 모델을 지원하는 각종 시스템 관련 소프트웨어를 동시에 지원하며 시스템 모니터링이나 제어에 대한 추가 개발이 용이하게 한다. PLC 제어 외에 PC가 직접 제어하는 부분은 스테이지, 스위치, Red Lamp, 현미경 제어, 그리고 Buzzer 등이 있다.

그림 5는 검사 장비의 프로시저를 보여준다. 준비동작으로 초기동작 완료, 반입준비동작 완료, 패널 고정, 조이스틱/기능키 사용 완료 등을 거치게 되며, 검사 프로세스에서는 실제 검사가 진행된다. 검사 과정이 끝나면 후기동작으로 조이스틱/기능키 사용불가 완료, 반출준비 동작 완료의 동작을 거쳐 하나의 공정이 끝나게 된다. 검사장비의 기구를 살펴보면 유리 기판을 움직이는 X 스테이지와 검사를 위한 현미경을 움직이는 Y 스테이지가 있다. 또한 좌표계는 기판

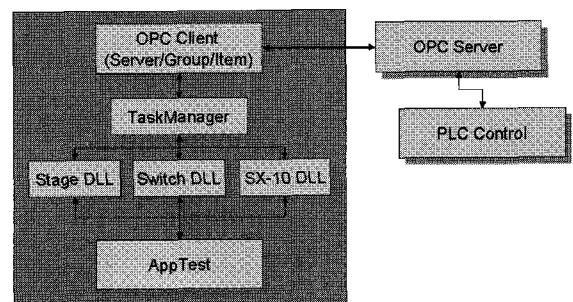


그림 4. 전체 시스템 개요도.  
Fig. 4. Whole system outline.

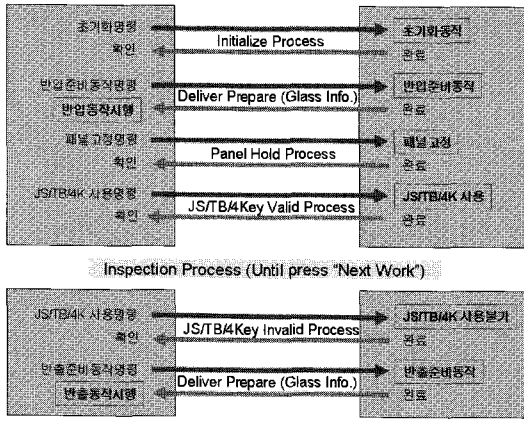


그림 5. 검사 장비 공정 과정.

Fig. 5. Work process of inspection system.

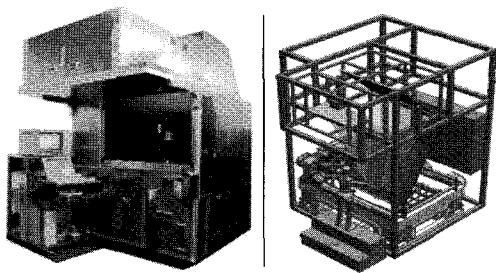


그림 6. 평판 검사 장비.

Fig. 6. Panel inspection system.

좌표계, 시스템 좌표계, 하드웨어 좌표계로 구성되어 있다.

그림 6은 검사장비 시스템이다. 이를 제어하는 OPC 클라이언트는 스테이지 구동, 스위치 구동, 현미경 구동, 화상처리, 열라이먼트, 태스크 관리, PLC 데이터 수신의 처리를 한다. 스테이지 라이브러리는 장비의 x, y, z축의 모터를 제어하고 관련된 센서를 제어하는 부분이며, 스위치는 사용자로나 PLC로부터 입력받은 스위치를 전달하거나 PLC로부터의 요청에 대한 응답을 표시한다. 현미경 구동 라이브러리는 결함 검사를 위한 현미경 제어하고, 태스크 관리 라이브러리는 수행되는 태스크에 대해 처리완료, 에러, 로그 발생 등을 수행한다. 이러한 라이브러리들을 연결하여 테스트용 OPC 클라이언트 프로그램을 개발하게 되며, 모든 프로시저를 관리한다[7,8].

5. 라이브러리 구성

본 논문에서 개발한 라이브러리는 OPC 클라이언트, 스테이지 구동, 스위치 구동, 현미경 구동, 화상처리, 열라이먼트, 태스크 관리로 나뉘어 개발되었다. OPC Client 라이브러리는 OPC서버인 KepServer를 통해 MELSEC-Q의 PLC를 구동할 수 있도록 연결성을 주로 다룬다. 스테이지 라이브러리는 장비의 x,y,z 축의 모터를 제어하고 관련된 센서를 제어하는 부분이며, 스위치는 사용자로부터 입력받은 스위치를 전달하거나 PLC로부터의 요청에 대한 응답을 표시한다. 현미경구동 라이브러리는 결함 검사를 위한 현미경 사용을 위한 구동용 라이브러리며, 화상처리 라이브러리는 MIL 라이브러리를 이용하여 장비에 맞게 설계된 라이

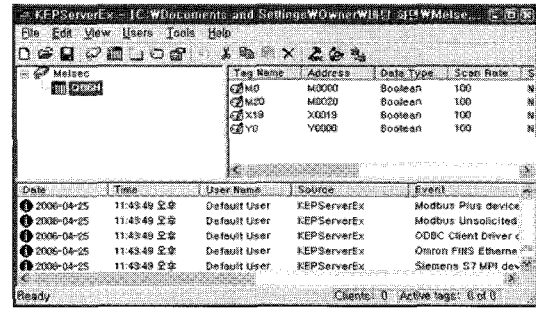


그림 7. KepServerEx 설정한 예.

Fig. 7. Examples for KepServerEx setting.

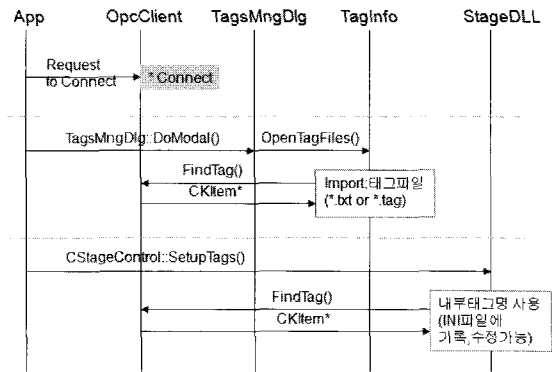


그림 8. 태그 자동 설정 흐름도.

Fig. 8. Flow chart of automatic tags setting.

브러리이다. 태스크 관리 라이브러리는 명령의 수행을 하는데 있어서 완료/에러 등을 관리하기 위한 라이브러리이다. 이 작업은 멀티태스킹으로 동작하며 각 동작에 대해 로그 파일을 생성할 수 있다. 따라서 어느 시점에 오류가 발생했는지 알 수 있다.

그림 7은 OPC Server의 설정 예를 보여주고 있다. 장비에 맞는 태그를 설정하여 입력을 하고 데이터 상태를 살펴본다. Client와의 접속상태를 살펴보기도 한다.

그림 8은 OPC Server로부터 태그정보를 전달받아 Client에서 사용할 명령어와 연결을 행한다. 태그정보에는 태그의 타입, 액세스 속도, 스캔속도, 스케일링, 등의 각종 정보를 포함하고 있다.

III. 실험

본 실험은 OPC 서버와 PC가 직접 액세스를 수행하는 것이 아니라, OPC 클라이언트와 TaskManager를 통해 데이터를 교환하고 출력 명령을 서버에 전달하는 OPC Data Manager 모듈을 구현하였다. OPC 서버 프로그램은 KepServerEx v4.0을 사용하였고, 관리 및 제어 프로그램은 Visual C++6.0 MFC 환경에서 개발하였다.

그림 9는 검사 장비의 테스트 프로그램의 실행화면이다. 위쪽은 PLC에 접속하여 OPC 서버의 실행 화면으로서 데이터를 수집하고, 공정을 테스트할 수 있게 구성되어 있고, 아래쪽의 탭 컨트롤은 Tag, Stage, Switch, SX-10, OPU 컨트롤러로 구성되어 있다. 일단 시스템이 제대로 통신이 되어

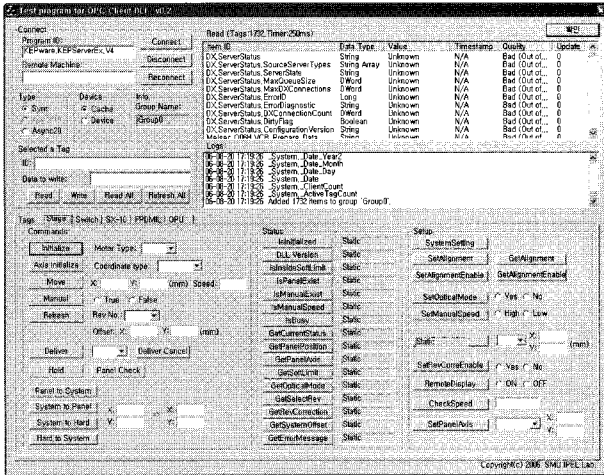


그림 9. 검사장비 테스트 프로그램.  
Fig. 9. Test program of inspection system.

작동하는지를 검사하기 위해 제어(Contorl)에서 명령을 수행한다. 통신이 확인되면 프로세스를 단계별로 시작해 전체 공정을 제대로 수행함을 결과적으로 알 수 있다[9,10].

**IV. 결론**

본 연구에서는 검사장비에 대한 공정 감시제어 및 관리용 소프트웨어에 대한 연구, 개발을 하였다. 검사장비의 제어는 기본적으로 PLC 프로그램으로 구동하게 되며, 이 PLC 프로그램과 PC 사이의 네트워크를 통해 데이터를 주고, 받을 수 있도록 OPC 서버와 클라이언트를 사용하여 데이터를 관리하였다. 제어 시스템 분야에서 세계 표준으로 자리 잡아 가고 있는 OPC 개념을 도입하여 개발의 용이성과 응용의 확장성을 동시에 얻을 수 있었다. 검사장비의 전체적인 프로세스를 관리하는 시스템은 OPC 클라이언트 모델로 개발되었으며, 태스크 관리, 로그정보 기록, 분석, 제품 결함의 검사를 실행한다. 또한, 시스템 운영의 예러 발생에 대한 처리를 지원하고 있으며, 결함 검사 시스템과 데

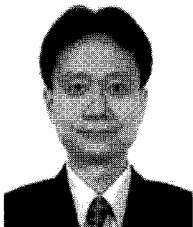
이터를 연동하여 관리한다. 나아가 전체 검사 시스템을 제어, 운용하기 위해서는 방대한 라이브러리와 프로그램이 필요하다. 초기 시스템에 맞는 기술을 확보하기 위해 많은 자료를 분석하고 국내의 기술을 파악하였다. 이번 연구로서 OPC 표준 인터페이스 기술의 타당성 및 효율성을 확인할 수 있었다. 개발된 기술은 검사장비 뿐만 아니라 일반적인 제어시스템에 모두 확대 적용 가능하다.

향후, 시스템의 통신 인터페이스의 통신 성능 향상, 시스템의 안정화를 목표로 하고 있다. 또한 인터넷을 통한 시스템 제어를 위한 HMI 분야의 연구도 진행될 예정이다.

**참고문헌**

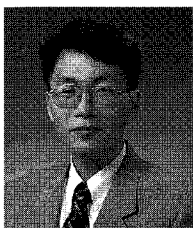
- [1] OPC Foundation, OLE for Process Control Standard, <http://opcfoundation.org/>
- [2] OPC Task Force, "OPC(OLE for Process Control) Overview," October 27, 1998.
- [3] D. W. Holley, "Understanding and using OPC for maintenance and reliability applications," *Computation & Control Engineering Journal* vol. 15, Issue 1 pp. 28-31, Feb-March 2004.
- [4] Component Object Model Specification, <http://www.microsoft.com/>
- [5] KepWare, <http://kepware.com/>
- [6] MatrikonOPC, <http://www.matrikonopc.com/>
- [7] TAKEBISHI, "DeviceXPLorer MELSEC OPC server," 2005.
- [8] 임기욱, 김홍남, "임베디드 소프트웨어분야의 기술 발전 동향" 한국정보처리학회지 제 1 권 제 6 호, 2004.
- [9] 정순보, 김효정, 윤성희, 한광록, "원도우 기반 PLC 제어용 HMI 저작도구의 설계 및 구현" 한국정보처리학회 춘계발표대회 논문집 제 11 권 제 1 호, 2003.
- [10] R. Lafore, "Object-oriented programming In C++," *Waite Group Press*, pp. 73-88, 1995.
- [11] KDT System, "CIMON-HMI," 2006.

**한 창 호**



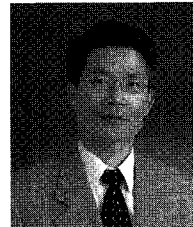
1993년 청주대 전자공학과 졸업. 1995년 동 대학원 석사. 2005년~현재 선문대학교 대학원 전자공학과 박사과정 재학중. 관심분야는 영상처리, 모션캡처, 자동검사.

**유 영 기**



1987년 한양대학교 기계공학과(학사). 1989년 한국과학기술원 생산공학과(석사). 1996년 한국과학기술원 자동화설계공학과(박사). 1989년~1992년 삼성종합기술원 전자기기연구소 연구원. 1996년~현재 선문대학교 전자정보통신공학부 조교수. 관심분야는 화상처리, 광계측 센서, 정밀측정시스템.

**오 춘 석**



1980년 서강대학교 전자공학과(학사). 1986년 Marquette University 컴퓨터공과(석사). 1992년 University of Arizona 컴퓨터공과(박사). 1980년~1993년 전자통신연구원(ETRI) 선임연구원. 2001년 미국 Optical Sciences Center Visiting Scholar. 1993년~현재 선문대학교 전자정보통신공학부 교수. 관심분야는 화상처리, 자동검사, 머신비전, 광학시스템설계.

부 조교수. 관심분야는 화상처리, 광계측 센서, 정밀측정시스템.