

PLC 자체 고장진단과 그의 외부 소자의 고장 진단 시스템 개발에 관한 연구

‘허 윤기’ , 변 중남*

*포항 종합 제철 기술연구소, KAIST 전기 및 전자공학과

A Study on Development of Fault diagnosis system for PLC self-diagnostics and its external devices

‘Yone Gi Hur’ , Zeungnam Bien*

POSCO Technical Research Laboratories, KAIST Dept. of Electrical Engineering

Abstract

In this paper, a fault diagnosis method is proposed for self-diagnostics of PLC(Programmable Logic Controller), process controller in industrial fields, and diagnosis of its external devices such as sensors and actuators. The aim of this research is proposition of systematic method of fault diagnosis of PLC control system and development of its equipment. A PLC fault diagnosis algorithm consists of self-diagnostics given by PLC makers, Input/Output tracking method by analyzing sequence PLC programs, searching method of past fault cases in database using an expert system, and diagnosis of PLC units such as CPU, DI, and DO board. Finally usability of PLC fault diagnostic system is verified by testing a MELSEC PLC.

대한 연구도 미비한 실정이다.

기존의 PLC 고장 진단 방법은 PLC가 제어하는 플랜트가 오동작하거나 정지하면 PLC를 재부팅(Booting)하거나 PLC의 전원을 점검하는 등의 기본적인 조작을 한 후 고장이 복구되지 않으면 PLC마다 제조 회사들이 제공하는 자기 진단 기능을 이용한다. 그러나 자기 진단의 기능만으로는 고장을 복구하는 데는 한계가 있다. 예를 들어, 자기 진단 기능의 신뢰성 문제이다. 이미 고장이 발생한 PLC의 진단 결과를 어느 정도 믿을 수 있느냐는 것이다. 실제로 자기 진단 결과의 오류에 의하여 고장 복구 시간이 더 많이 걸린 경우도 있다. 또한, 자기 진단의 기능이 제공하는 고장의 범위는 PLC 자체만의 고장 정보를 제공한다. 따라서 센서나 구동기의 오동작에 의하여 PLC가 마치 고장인 것처럼 나타날 경우 그 원인을 찾기는 실로 어렵다.

본 연구는 이와 같은 문제점을 해결하기 위한 것으로, 고장이 발생한 PLC 제어 시스템에 대하여 고장의 발생 부위와 원인을 운전원에게 제공하여 체계적인 PLC 제어 시스템의 고장 진단 방법 및 개발된 장치를 통하여 고장 복구 시간을 단축시키는 것이 목적이다. 본 연구에서는 PLC 고장진단의 한 방법과 MELSEC PLC를 대상으로 실험한 결과를 소개하고자 한다. 논문의 구성은 다음과 같다. 제1장의 서론에 이어서 2장에서는 PLC 제어 시스템의 고장 진단 알고리즘을 서술하고 3장에서는 구현된 진단용 블록에 대하여 소개한다. 제4장에서는 고장진단의 실험 결과를 밝히고 제5장에서 결론을 맺는다.

1. 서론

PLC란 산업 공정에서 널리 사용되는 공정 제어용 제어기이다. 플랜트의 센서, 구동기와 연결되어 프로세서의 신호를 입력받아서 제어 입력 값을 계산하여 구동기로 보내어 주는 역할을 하는 제어기인 PLC는 1969년 미국의 제너럴 모터(GM)사가 최초로 PLC를 설비에 도입한 이래로 현재 대규모 설비뿐만 아니라 소형 기계 및 부대 설비에도 흔히 사용되고 있다. 최근 PLC 제작 기술과 기능의 발달에 의하여 산업계에 PLC의 도입은 지속적으로 증가되고 있다. 공정의 규모에 따라서 PLC는 수 대에서 수천 대의 PLC가 운용되고 있으며 단계적으로 이루어지고 있는 공정에서 PLC의 고장은 공정의 생산 라인을 중단시킬 수 있으며 나아가서 전체 생산 라인을 마비시킬 수 있다. PLC에 의한 고장을 복구하고 고장 복구 시간을 최소화하는 것은 생산성 향상에 매우 크게 작용하여 안정된 조업을 위해서도 필수적이다. 기계적 설비에 의한 고장은 눈으로 확인이 가능하므로 비교적 빨리 찾을 수 있으나, PLC에 의한 고장은 고장 결과는 나타나지만 그의 원인을 찾기에는 매우 어렵다. 따라서 고장 복구에 따른 시간이 상당히 소요된다. 현장의 경우 PLC 고장의 원인을 파악 및 대처 방법이 체계화되어 있지 않으며 이에

2. PLC 고장진단 알고리즘

그림 1은 PLC 고장 진단 시스템의 구성도이다. PLC와 링크(Link) 유닛에 의하여 대상 PLC와 인터페이스되며 링크 유닛과 PLC 고장 진단 시스템이 연결되어서 데이터를 통신하는 구성이다. 플랜트를 구동하는 PLC는 링크 유닛을 통하여 고장 진단 시스템과 연결되어 있다. 고장 진단 알고리즘부는 PLC로부터 입/출력 정보 및 자기 진단 결과를 입력받고 고장 사례 데이터 베이스를 활용한다. 또한 PLC 언어 번역기는 PLC로부터 서비스 프로그램

을 입력받아서 사용자가 인식할 수 있는 코드로 번역하여 고장 진단 알고리즘부로 번역된 결과를 넘겨준다. 한편, 운전원은 고장 진단 알고리즘부로부터 정보를 입력받아서 PLC 고장 진단을 하게 된다.

고장 진단 알고리즘을 상세히 설명하면 다음과 같다. PLC에 고장이 발생하였을 경우 가장 먼저 확인해 보아야 할 것은 제조사에서 제공하는 자기 진단 기능(표 1에 그의 예가 나타나 있다.)이다. 자기 진단 기능이 항상 옳은 정보를 제공하는 것은 아니지만 사소한 고장의 경우에는 이 기능만으로도 문제를 해결해 낼 수 있으므로 먼저 자기 진단의 결과를 활용한다. 또한 운전원이 직접 어려 코드가 저장되어 있는 메모리를 모니터링하지 않도록 고장 진단 시스템에서 보여 준다. 자기 진단에 의하여 문제가 해결되지 않거나 자기 진단의 결과가 정상으로 나타난다면 현재 플랜트의 상태가 특정 출력점의 신호에 의한 오동작임을 확인하여 이를 역추적(Backward Tracking)하여 특정 출력점에 영향을 미치는 입력점을 찾아 운전원에게 제시하여 이를 조사하게 한다. 해당 입력 신호들의 상태가 모두 정상이라면 고장은 PLC 자체 내에 있으므로 PLC의 증상을 토대로 진단을 시작한다. 고장이 발생한 PLC 또는 같은 기종, 같은 공정에 있는 PLC를 대상으로 데이터 베이스 관리부로부터 과거 고장 사례의 증상을 중에서 핵심어를 제시하여 운전원이 이를 선택함으로써 고장 증상의 상세한 내력을 질의를 통해 확인하고 해당 증상이 있으면 과거의 고장 원인, 고장 증상, 고장 조치 상황을 제시한다. 운전원은 이를 바탕으로 고장 복구에 관한 정보를 얻는다. 과거 고장 사례가 없거나 위의 방법들로 해결이 되지 않을 때에는 유닛별 진단부로부터 PLC의 각 유닛을 진단하는 시퀀스 프로그램을 실행시켜 유닛별로 각각의 기능이 정상적으로 수행되는지를 테스트함으로써 최종 진단을 한다. 이후 현재의 고장 사례를 데이터베이스에 저장함으로써 진단을 마친다. 이상의 고장 진단 알고리즘을 구현하기 위한 시스템의 구성은 그림 2에 나타내었다.

이하 고장 진단 알고리즘부의 각 기능별로 설명하면 다음과 같다. 자기 진단부는 PLC에 어려가 발생하였을 경우에 메모리의 특정 영역에 어려 코드를 기록하는 기능이다. 운전원이 작성한 시퀀스 프로그램 상의 버그에 의하여 PLC가 정지하였을 때 이를 알려 주는 기능, 각 유닛들의 장착 상태를 확인해 주는 기능, 퓨즈의 절단 상태 및 전원 전압의 강하가 발생한 경우 이를 알려 주는 기능, 특수 기능 유닛으로 접근 불능을 알려 주는 기능 등이 있으며 구체적인 것은 PLC 제조사마다 다르다. 따라서 PLC들이 제공하는 어려 코드 번호와 고장 원인 및 부위를 정리한 테이블을 읽어서 데이터 베이스에 저장해 놓고 PLC로부터 어려 코드를 읽어서 운전원에게 정보를 제공하는 방법을 말한다. 그림 5에 이에 대한 실험 결과가 나타나 있다.

PLC 고장 진단에 있어서 PLC 외부인 입출력 디바이스의 문제에 의해서 잘못된 입력이 들어와서 마치 PLC가 오동작하는 것처럼 나타날 경우, PLC 프로그램 분석부는 오동작을 하는 출력점을 진단 시스템에 입력하고 이에 영향을 주는 입력점을 찾아서 제시함으로써 몇 개의 입력 신호선 만을 확인하여 고장의 원인을 찾아낼 수 있다.

이는 PLC에서 구동하는 프로그램을 진단 시스템이 읽어 와서 차권스 프로그램을 분석함으로써 이루어진다. 주어진 출력 신호로부터 입력 신호를 찾아내는 방법은 이중 연결 목록(Doubly Linked List; DLL)기법을 이용하였다. DLL은 하나의 노드(node)가 자신을 가리키는 노드와 자신이 가리키는 노드에 대한 어드레스(address)를 갖고 있는 구조이다. 이를 이용하면 우선 시퀀스 프로그램 내에서 사용된 입력 변수들과 내부 변수들을 각각 노드로 설정하고 프로그램을 읽으면서 각 노드들 간의 관계를 나타내는 어드레스를 저장한다. 따라서 운전원이 입력한 특정 출력점에 대하여 DLL 구조를 따라가면서 최종적으로 영향을 미치는 입력점을 찾아낸다. 그럼 6에 이에 대한 실험 결과가 나타나 있다.

고장 사례의 데이터 베이스로부터 진단하는 질의부는 과거의 고장 사례의 증상을 중에서 핵심어(key word)만 운전원에게 제시하여 해당되는 사항을 선택하면 이의 증상을 자세히 보여 주어 운전원이 현재의 증상과 일치하는지를 결정하고, 일치하면 당시의 고장 원인, 고장 부위, 그리고 조치 사항을 제시하는 방법이다. 이를 위하여 데이터 베이스에 미리 고장 증상마다의 핵심어를 설정해 놓는다. 그럼 7에 이에 대한 실험 결과가 나타나 있다.

3. PLC 진단용 Tool; PLCDocitor

개발된 고장 진단용 툴인 PLCDocitor는 고장진단의 기능 외에 평상시 PLC의 Loader기능을 대신할 수 있는 기능이 있다. PLC와 인터페이스 할 수 있는 툴인 PLCDocitor는 PC Windows에서 구동되며 소스 코드는 C로 구현하였다. 데이터베이스 관리는 Personal Oracle7을 이용하여 데이터베이스를 구축하였고 데이터베이스와의 인터페이스를 위하여 ODBC(Open DataBase Connectivity)를 이용하였다. 진단용 툴의 기능은 첫째, PLC로부터 시퀀스 프로그램을 읽어오고 시퀀스 프로그램을 코딩할 수 있으며 코딩한 시퀀스 프로그램을 PLC에 쓸 수 있으며, 읽어온 프로그램을 화면상에 보여주는 기능이 있다. 둘째, PLC의 멀레이와 레지스터의 값을 실시간으로 모니터링 할 수 있다. 셋째, 고장진단의 기능이 있다. 넷째, 데이터베이스를 이용하여 과거 고장 이력을 조사할 수 있고 추가/삭제가 가능하다. 다섯째, PLC 프로그램의 시뮬레이션 기능이다. PLC 시뮬레이터는 PLC를 이용하여 플랜트를 동작시키고자 할 때 먼저 사용자가 설계한 PLC 프로그램이 자신이 원하는 형태의 출력을 나타내고 있는지를 알아보고 오류가 있을 때는 수정하기 위하여 사용되어 진다. 또한 PLC의 고장시에는 PLC에 현재 구동되고 있는 프로그램을 읽어 와서 PLC 시뮬레이터에서 실행해 봄으로써 프로그램 상의 오류에 의하여 PLC 고장이 발생하였는지의 여부를 알고자 할 때 사용된다.

4. 고장진단 실험 및 결과

앞에서 설명한 PLC 고장 진단 시스템의 성능을 평가하기 위하여 실제로 PLC에 고장을 발생시켜 이를 찾아내는 실험을 한다. 그림 3은 PLC 고장 진단 실험 세트의 구성도이다. PLC에 직

접 고장을 발생시키는 것 대신에 PLC 혹은 출력 디바이스에 가상의 고장을 발생시키는 방법을 택하였다. 실험 대상의 PLC는 POSCO의 PLC의 한 기종인 MELSEC 계통인 금성 기전 Gold-Sec M-Series의 M2N CPU로서 최대 출력 포인트(Point)수는 512 포인티이고 1.0 sec/step의 처리 속도를 가지고 있다.

PLC 자체의 고장과 PLC 외부의 고장에 대한 진단과 고장 사례를 통한 진단의 기능을 확인하기 위하여 가상의 플랜트를 설정하였다. 그럼 4에 고장 진단 실험을 위한 전체적인 실험 세트의 구성이 나타나 있다.

실험은 다음과 같이 하였다. 첫째, 자기 진단 기능의 활용을 위하여 자기 진단으로 진단이 가능한 고장을 발생시킨다. 예를 들어 표 1의 예리 코드 10의 고장을 발생시키면 PLCDocor에서는 그림 5의 결과를 제시한다. 둘째, PLC의 구동 프로그램의 입/출력 결과를 이용하는 진단의 기능의 활용을 위하여 가상의 플랜트로부터 잘못된 센서 값을 PLC로 보내고 이를 추적해내는 실험을 한다. 그림 6은 동작 중인 PLC 시퀀스 프로그램을 읽어 와서 입출력의 관계를 보여주고 있다. 셋째, 가상의 플랜트를 제어하는 PLC의 과거 고장 사례를 미리 데이터 베이스에 넣어 두고 이를 이용하여 유사한 고장을 찾아내는 실험을 한다. 그림 7에서 상자 포장 기계가 최대 용량을 초과하여 포장하고 있다고 가정하자. 운전원은 현재 증상과 일치하는 증상의 핵심을 선택하면 고장 진단 시스템에서는 해당 증상의 원인, 부위, 조치 사항을 제시한다.

5. 결론

본 논문에서는 산업 공정의 프로세서 제어기인 PLC의 고장을 진단하는 한 가지 방법에 관하여 기술하였다. 본 연구의 목적은 고장이 발생한 PLC 제어시스템에 대하여 고장의 발생부위와 원인을 운전원에게 제공하여 체계적인 PLC 제어 시스템의 고장 진단 방법 및 그 장치의 개발이다.

본 연구에서 PLC 고장 진단 방법으로서 첫째, PLC 제조업체가 제공하는 자기 진단 결과를 이용하여 진단한다. 둘째, PLC의 시퀀스 프로그램의 입/출력 결과를 이용하여 진단한다. 셋째, 고장 사례의 데이터베이스에 의하여 진단한다. 넷째, PLC 내부의 CPU 및 I/O 보드를 진단한다.(본 논문에서는 제외하였다.)

고장 진단 시스템의 성능을 평가하기 위하여 POSCO의 PLC의 한 기종인 MELSEC PLC를 대상으로 실험하였다. 향후 현장의 운전중인 PLC에 대하여 실험할 예정이며, PLC의 기종이 달라지거나 PLC 간의 통신 및 상위 DataWay 간의 통신에 의한 고장 진단하는 방법이 다루어져야 할 것이다.

참고 문헌

- [1] 포항제철 기술연구소 시스템연구팀, "PLC 제어시스템 고장 진단 전문가 시스템을 위한 기초연구", 연구보고서, 1996
- [2] 포항제철, "고장 사례 관련 자료", 1994

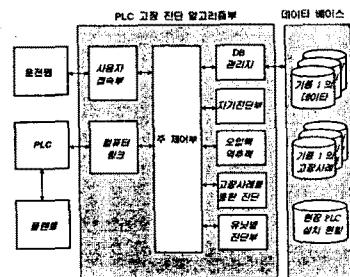


그림 1은 PLC 고장 진단 시스템의 상세 구성도이다.

표 1은 PLC의 자기 진단 기능이다.

호환형식	설정형식	설정내용	설정설명
10	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
11	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
12	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
13	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
14	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
15	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
16	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
17	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
18	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
19	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
20	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
21	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
22	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
23	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
24	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
25	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
26	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
27	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
28	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
29	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
30	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
31	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
32	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
33	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
34	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
35	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
36	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
37	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
38	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
39	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
40	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
41	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
42	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
43	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
44	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
45	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
46	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
47	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
48	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
49	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
50	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
51	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
52	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
53	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
54	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
55	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
56	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
57	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
58	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
59	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
60	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
61	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
62	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
63	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
64	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
65	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
66	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
67	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
68	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
69	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
70	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
71	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
72	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
73	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
74	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
75	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
76	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
77	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
78	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
79	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
80	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
81	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
82	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
83	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
84	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
85	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
86	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
87	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
88	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
89	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
90	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
91	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
92	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
93	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
94	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
95	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
96	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
97	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
98	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
99	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
100	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
101	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
102	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
103	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
104	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
105	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
106	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
107	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
108	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
109	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
110	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
111	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
112	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
113	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
114	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
115	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
116	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
117	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
118	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
119	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
120	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
121	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
122	설정형식	00000000000000000000000000000000	00000000000000000000000000000000
123			

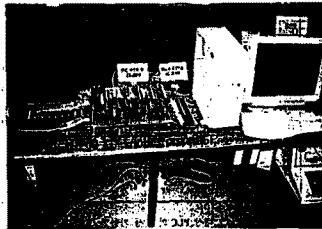


그림 4는 PLC 고장 진단 실험 세트의 구성이다.

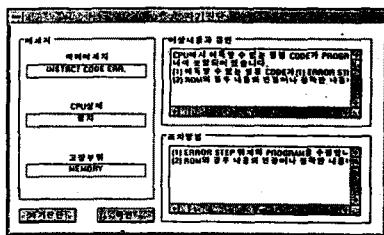


그림 5는 PLC의 자기 진단 결과이다.

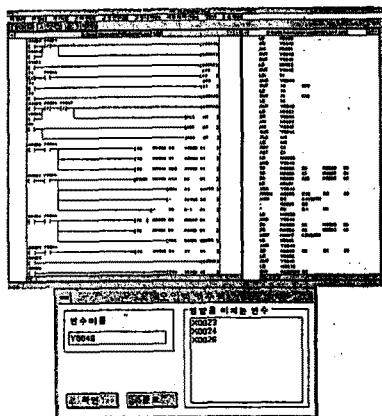


그림 6은 PLC 구동 프로그램에 의한 입/출력 주석 방법이다.

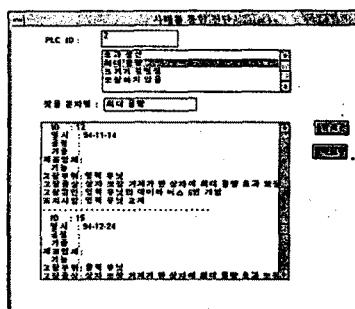


그림 7은 PLC 고장 사례 데이터베이스를 통한 진단의 결과이다.