

PLC 출력 확장 디코딩 프로그램 모듈 개발

(Development of Decoding Program Module for PLC Output Expansion)

유 정 봉
Jeong Bong You

Abstract - In this paper, we proposed the program module that expand output points when increased of the output machine for the design of process control system with PLC. In order to increase output points we need the decoding. There is a hardware decoding and a software decoding. In this paper, we proposed the decoding program module which is a software decoding and confirmed feasibility through a simulation.

Keywords : PLC, Ladder diagram, Expansion module, Output unit, Decoder,

1. 서 론

PLC를 설계하여 운전하다가 출력 기기의 수가 많아져 출력을 확장하고자 할 때는 기본 베이스에 출력 유닛을 추가할 수 있는 여분이 있거나, 출력 유닛에 출력 점수를 확장할 수 있는 여분이 있을 때는 출력점수를 확장하기가 용이하다. 그러나 출력 유닛에 출력점수를 확장할 수 있는 여분이 없을 때는 출력 유닛을 추가하여야 한다. 예를 들어 출력 유닛에 3점의 추가 여유분이 있을 때 5점의 출력기기를 추가하여야 한다고 하면 출력 유닛을 추가해야 한다. 또한 기본 베이스에 유닛을 추가할 여유가 없을 때 약간의 출력 점수를 추가해야 한다면 중설베이스와 전원부등을 추가해야 하기 때문에 비용이 상당히 증가하게 된다.^[1-2] 이러한 단점을 극복하기 위해 본 논문에서는 출력 유닛을 추가하지 않고 기본으로 설치되어 있는 출력 유닛에서 출력 점수를 확장할 수 있는 확장 프로그램 모듈을 제안하고, 그의 타당성을 확인한다.

II. 출력 디코딩 프로그램 모듈 설계

2.1. 확장 시스템 구성

확장을 위해 입력력 유닛을 장착한 시스템은 그림 1과 같다. 그림 1에서 기본베이스에 16점 출력 유닛을 장착하였을 때 상위 4비트(15~12번 접점)를 확장을 위한 접점으로 사용한다. 확장을 하지 않을 때는 일반 접점으로 사용할 수 있다.

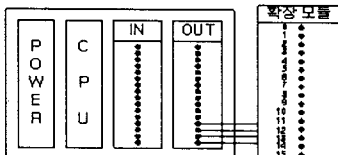


그림 1. 확장 시스템 구성도
Fig. 1. Structure of System for Expansion

확장을 위해 확장 유닛을 장착하게 되면 출력 유닛의 상위 4비트(15~12번 접점)를 확장 유닛에서 입력으로 받아들여 16개의 접점으로 확장시킨다.

따라서 그림 1의 예에서 출력 유닛을 0번 접점부터 27번 접점까지 28개의 출력 접점을 사용할 수 있다.

2.2. 프로그램 모듈 설계

기본베이스에 16점 입력 모듈 1개와 출력 모듈 1개로 구성된 시스템을 예로 하여 프로그램 모듈을 설계하였다. 프로그램 모듈의 순서도는 그림 2와 같다.^[3-4]

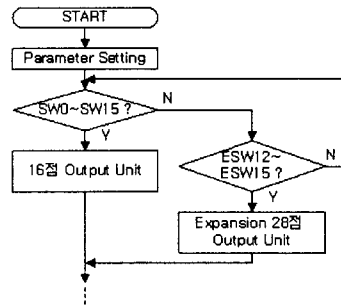


그림 2. 프로그램 모듈 순서도
Fig. 2. Flow chart of Program Module

16점 출력 모듈에서 확장하고자한다면 0~11번 접점은 그대로 사용하고, 12~15번 접점을 확장모듈의 입력으로 사용하면 된다. 프로그램시 확장모듈의 번수는 ESW12~ESW27까지로 16개를 사용한다. 그러면 16점 출력모듈을 0~27번까지 28점 출력 모듈로 확장할 수 있게 된다.

그림 3에서 ESW12~ESW27을 ED0~ED3으로 인코딩하고, 확장모듈의 입력으로 입력하여 16개의 디코딩 출력으로 출력시킨다. 인코딩된 ED0~ED3의 식은 (1)과 같다.

저자 소개
正 會 員 : 공주대학교 전기전자공학부 조교수·공학박사

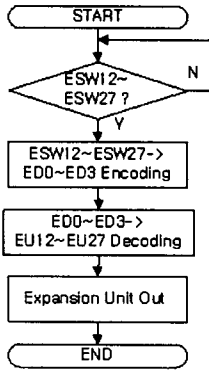


그림 3. 확장모듈 순서도
Fig. 3. Flow chart of Expansion Module

$$\begin{aligned}
 ED0 &= ESW13 + ESW15 + ESW17 + ESW19 \\
 &\quad + ESW21 + ESW23 + ESW25 + ESW27 \\
 ED1 &= ESW14 + ESW15 + ESW18 + ESW19 \\
 &\quad + ESW22 + ESW23 + ESW26 + ESW27 \\
 ED2 &= ESW16 + ESW17 + ESW18 + ESW19 \\
 &\quad + ESW24 + ESW25 + ESW26 + ESW27 \\
 ED3 &= ESW20 + ESW21 + ESW22 + ESW23 \\
 &\quad + ESW24 + ESW25 + ESW26 + ESW27
 \end{aligned} \quad (1)$$

또한, ED0~ED3을 에디터 프로그램에서 확인할 수 있도록 EU12~EU27로 디코딩한 식은 (2)와 같다.

$$\begin{aligned}
 EU12 &= \overline{ED3} \cdot \overline{ED2} \cdot \overline{ED1} \cdot \overline{ED0} \\
 EU13 &= \overline{ED3} \cdot \overline{ED2} \cdot \overline{ED1} \cdot ED0 \\
 EU14 &= \overline{ED3} \cdot \overline{ED2} \cdot ED1 \cdot \overline{ED0} \\
 EU15 &= \overline{ED3} \cdot \overline{ED2} \cdot ED1 \cdot ED0 \\
 EU16 &= \overline{ED3} \cdot ED2 \cdot \overline{ED1} \cdot \overline{ED0} \\
 EU17 &= \overline{ED3} \cdot ED2 \cdot \overline{ED1} \cdot ED0 \\
 EU18 &= \overline{ED3} \cdot ED2 \cdot ED1 \cdot \overline{ED0} \\
 EU19 &= \overline{ED3} \cdot ED2 \cdot ED1 \cdot ED0 \\
 EU20 &= ED3 \cdot \overline{ED2} \cdot \overline{ED1} \cdot \overline{ED0} \\
 EU21 &= ED3 \cdot \overline{ED2} \cdot \overline{ED1} \cdot ED0 \\
 EU22 &= ED3 \cdot \overline{ED2} \cdot ED2 \cdot \overline{ED0} \\
 EU23 &= ED3 \cdot \overline{ED2} \cdot ED1 \cdot \overline{ED0} \\
 EU24 &= ED3 \cdot ED2 \cdot \overline{ED1} \cdot \overline{ED0} \\
 EU25 &= ED3 \cdot ED2 \cdot \overline{ED1} \cdot ED0 \\
 EU26 &= ED3 \cdot ED2 \cdot ED1 \cdot \overline{ED0} \\
 EU27 &= ED3 \cdot ED2 \cdot ED1 \cdot ED0
 \end{aligned} \quad (2)$$

본 프로그램 모듈에서 ED0에서 ED3의 변수를 설계하고자 하는 PLC에 맞게 바꾸어 주지만 어떤 PLC에도 적용이 가능하다.

III. 시뮬레이션

제작된 실험용 PLC 셋트는 그림 4와 같다. PLC는 LG 산전의 GLOFA GM4 기종의 CPU를 사용하여 시뮬레이션하고 연구의 타당성을 확인하였다.^[5-6]

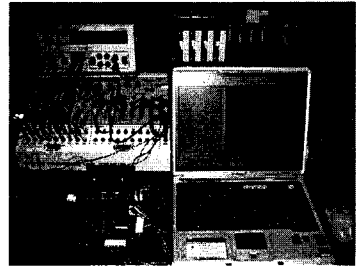


그림 4. 실험용 PLC 셋트
Fig. 4. PLC for Experiment

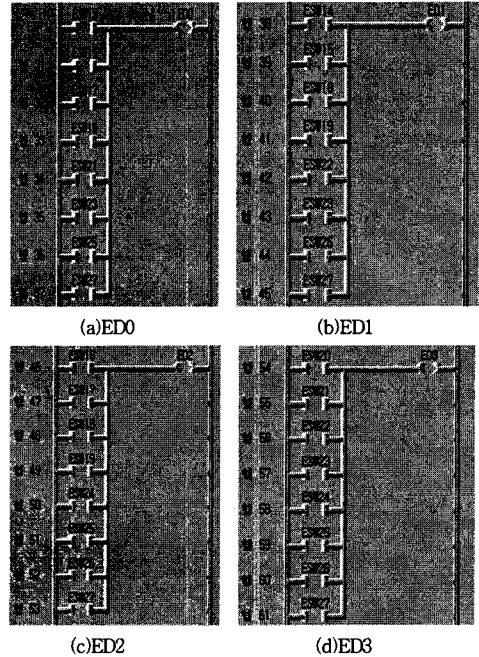


그림 5. Encoding 입력
Fig. 5. Encoding Input

확장 모듈 프로그램은 서브루틴으로 사용하였으며, 인코딩 입력 ED0~ED3은 그림 5와 같다.

또한 확장 모듈의 최종 디코딩 출력 프로그램은 그림 6과 같다.

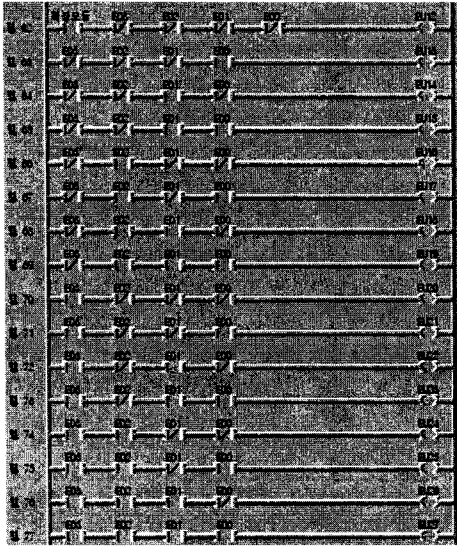


그림 6. 디코딩 출력 프로그램
Fig. 6. Program of Decoding Output

그림 6의 EU12~EU27은 확장 모듈의 최종 출력 접점을 나타낸다. ESW12 접점을 On하였을 때는 확장 모듈의 EU12번 출력이 On이 되어야 한다. 이는 그림 7에서 알 수 있다.

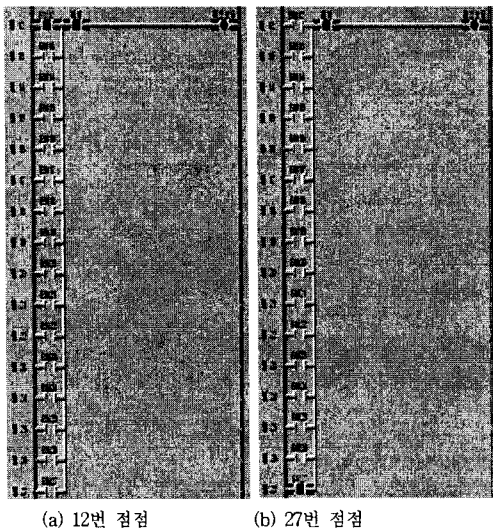


그림 7. 출력 디코딩 화면
Fig. 7. Display of Output decoding

IV. 결 론

본 논문에서 PLC의 인코딩 신호를 받아 디코딩하여 출력 점수를 확장할 수 있는 방법을 제안하였다.

PLC를 설치하여 운전하다가 소량의 출력 점수를 확장하고자 할 때 본 논문에서 제시한 방법을 사용하면 비용절감 및 시간절약 측면에서 상당한 효과가 있을 것으로 판단된다.

본 논문에서는 LG GLOFA 기종의 PLC를 사용하여 시뮬레이션을 하였지만, 변수의 값을 변경하여 주면 모든 PLC의 출력 유닛에 연결하여 사용 가능하다.

본 논문에서 제시한 방법은 PLC를 사용한 자동화 설계에서 프로그래밍시 및 디버깅시에 매우 효율적으로 사용할 수 있다.

참 고 문 헌

- [1] Masaharu Oku, Takao Kokubu, Shigeru Masuda, and Kenzo Kamiyama, "Application of the Encapsulated Actuator Model to the Sequential Control Machines", Short Papers IEEE/ASME Transactions on Mechatronics, Vol 1, No 4, pp290-294, Dec 1996.
- [2] T. Kouthon, M. A. Peraldiabd, J. D. Deco-tignie, "Distributing PLC Control", International Conf. On IEC, IEEE 21'st Vol 2, pp. 1614-1619, 1995.
- [3] M. Zhou and E Twiss, "Design of Industrial automated systems via relay ladder logic programming and Petrinets", IEEE Trans on Systems, Man and Cybernetics-part C ; Applications and Reviews, Vol 28, No 1, pp 137- 150, 1998.
- [4] R.W.Lewis, "Programming Industrial Control Systems Using IEC1131-3", The Institution of Electrical Engineers, 1992.
- [5] 지유양, 유수복, "글로벌 PLC 프로그래밍 실습", 기진연구소, 2002.
- [6] "LG Programmable Logic Controller Glofa-GM", LG Industrial Systems, 2004.