#3 전기 기능장 PLC 코딩 방법(XG5000)

1. 빠른 코딩을 위한 제언

a. **PLC입출력 변수명과 디바이스명을 정의한다.**
   => 동작설명문과 시스템도 , 플로우 차트등의 가독성이 높아진다. 즉 별도의 디바이명을 변환해서 동작설명문에 주석을 달지 않아도 된다
b. **사용할 CPU 모듈을 설정한다.**
   => 코딩중에 시뮬레이터로 작동여부를 수시시로 체크 하고 확인 하면서 완성하기 위해서
c. **시스템 모니터 창과 프로그램 코딩창은 동시에 볼수 있도록 창의 크기를 조정해 둔다.**
   => XG5000 4.2이상버전에서는 최근에 조정한 창의 위치, 싸이즈가 기억되어 다음 작업에도 동일하게 표시된다.
d. **시뮬레이션 상태에서 런 중 수정(Ctrl + Q) 런 중 쓰기(Ctrl + W)를 반복하면서 완성도를 수시로 확인 한다.**
   => 바른 코딩을 하고 있는지 의심 스러울때는 수시로 시뮬레이션 상태에서 변수등의 작동이 의도 하는데로 이루어지는지 확인 한다.
e. **주석 설명문은 가급적 달지 않는다.(변수에 주석이 필요하다면 변수명을 기입한다)**
   => 여러사람이 작업하거나 , 문서화 시켜서 다른사람에게 전달해야 하는 산출물인 경우에는 주석과 설명문을 충실히 기술하는 것이 맞다. 그러나 기능장 시험은 빠른시간내에 결과를 보여주는 1회용 시험이다.
f. **입력과 제어부분을 먼저 코딩 하고 출력 부분을 나중에 코딩한다**.
   => 입력조건이 명확한 출력은 바로 출력을 코딩한다. (ex:시퀀스 도면)
g. **입력과 제어부분은 위쪽에 출력은 아랫쪽으로 모아서 코딩한다.**
   => 출력은 이중코일을 방지하기위해서 동일조건의 출력을 병렬로 나열해야 하기 때문이다.
h. **입력과 제어 부분은 동작설명 순서에 맞게 직렬조건으로 , 병렬 조건은 계단식으로 작성 한다**
   출력부분은 출력 코일중심으로 병렬우선으로 코딩한다.
   => 코딩 샘플 참조
i. **코딩은 단축키와 방향키 ,Enter 키를 주로 이용한다.**
   => 마우스를 조작하기위해서 자판위에서 타이핑하는 손이 벗어나는 시간을 줄인다. 방향키와 Shif키로 단순반복 코딩을 줄이는 연습을 많이 해야 한다.
j. **입출력 변수 또는 시스템 디바이스명을 입력시 가능하면 소문자 입력 한다.(자동으로 대문자로 치환됨)**
   => 영문자 O(o)와 숫자 0 혹은 영문자 I(i) 와 숫자 1 등과 같은 비슷한 문자의 구별이 쉬워진다
   => 숫자와 영문이 확연하게 차이나는 폰트 선택(도구->옵션->LD/글꼴색상/텍스트글꼴=>Georgia/Bell MT
k. **Bit ,Word 등의 Data type의 Size에 주의를 해야한다.**
   => PLC는 메모리 용량에 자유스럽지 못하기 때문에 명령어 마다 결과물이 저장되는 크기가 가변적이다. Ex) MUL D0001 D0002 D0100 => D0100 , D0101으로 결과가 저장된다. 매뉴얼 참조

2. 실수 없는 프로그래밍을 위한 제언

a. 동작설명을 정독한다.
   => 눈에 익은 문제처럼 보이더라도 선입견없이 처음보는 유형인것 처럼 정독한다

b. 끝까지 해석이 난해한 부분은 표시해두고 반드시 질문해서 명확히 한다
   => 섣부른 해석은 프로그램 오류의 치명타이다. 전체 문맥이 파악되면 해석이 되는 부분도 있지만 끝까지 해석이 난해하면 질문해서 명확하게 하고 넘어 가야 한다.

c. 추가설명문,밑줄, 굵은체, 강조를 위한 조사에 특히 유의 해야한다.
   => 해당 과제의 함정은 대부분 이런 곳에 숨겨져 있다.

d. PLC입출력도는 하나 하나 짚어 가면서 변수명을 읽어 보기 바란다.
   => 램프 순서를 바꾸거나 중간에 건너 띄는 출력물 번호를 나열 해서 함정을 만들기도 한다.

e. 너무나 익숙한 패턴의 문제가 나오면 의심해 봐야한다.
   => 기능장 시험은 떨어뜨리기 위한 시험임을 명시해야 한다.

f. 측정이 모호한 문제는 출제 되지 않는다
   => 예로 버튼이 눌러지는 시간의 합이 5초이상일때 같은 조건은 측정이 모호 하다 이럴때 눌러지는 시간 만큼 초단위로 점멸하는 점등 조건을 반드시 추가 한다.

g. 기능장 시험의 입력 으로 들어 오는 신호는 기본이 a 접점 신호에 연결이 되어 있다고 가정 한다
   => PB은 NO에 연결된 a접점 신호 SS는 11시방향이 OFF 이기때문에 때문에 1시방향 ON 신호에 연결되어 있다고 가정 한다.

h. 접점(a , b) 은 Bit 신호인 상승 하강 신호와는 분명 다르다
   => 평상시 열린 접점(a contract) 과 양변환 검출 접점을 동일한 신호로 혼동해서 사용하는데 엄격히 구분할 필요가 있다.

i. 동작의 시작과 끝은 자기유지회로 보다는 Set 과 Reset 코일로 하는것이 유리하다
   => 시작(Set)을 하면 반드시 종료시점(Reset)을 언제로 할지 고민 하여 코딩 한다.

j. 해당 동작을 표시하는 값의 시효가 끝나면 반드시 초기화(Reset)해야하는 시점에 주의 한다
   => CTU ,CTD 등 Counter로 증감된 값의 Reset,INC DEC로 변경된 값의 MOV 0 초기화 등

k. Flow chart문제는 언제 변수를 초기화 할것 인가에 주의해야 한다
   => b접점을 입력조건으로 주어진 경우 그 해당 접점이 초기화 버튼이다
   직렬로 입력 접점이 발생하면 반드시 입력 접점을 병렬로 반대 접점 발생시 초기화 해야 한다

l. Flow Chart에서는 화살표의 분기점과 회기점을 주의 깊게 살핀다
   =>분기이후 신호 발생후 모든 동작이 자도 완료되면 분기된 동작 시작점으로 회귀할 가능성이 높고 동작이 계속 진행되는 경우 분기점 이전 조건이 초기화 조건일 경우가 많다

m. Flow Chart에서 동일 분기 조건에서 시퀀스 하게 동일 버튼의 조건을 반복적으로 묻는경우 는 one button 문제일 확율이 크다

n. Time Chart에서 변이가 일어 나는 시점이 상승 혹은 하강 에지인지 주의깊게 살핀다
   =>모든 입력 신호는 상승 에지가 기본이다 (특별한 언급이 없다면) 단순 a 접점보다는 상승에지로 처리하는 것이 정확하다.

o. Time Chart에서는 pattern 파악이 중요하다
   => 반복되는 형태가 가변적인지 고정적인지 1회성 혹은 n회 성인지 파악 해야한다
   반복되는 패턴앞에 지연시간을 주는 방법으로 함정을 만들거나, 1초간격을 0.5 혹은 2초등로 명시적으로 변경하여 수험자의 주의력을 시험하기도 한다