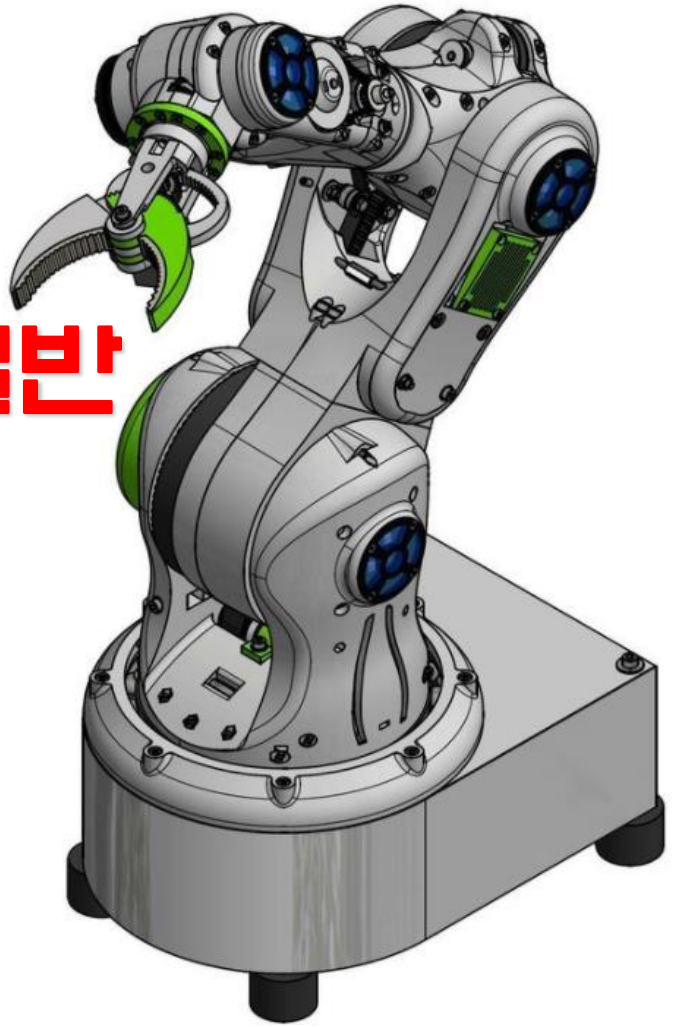
PLC 기능과 프로그램 일반



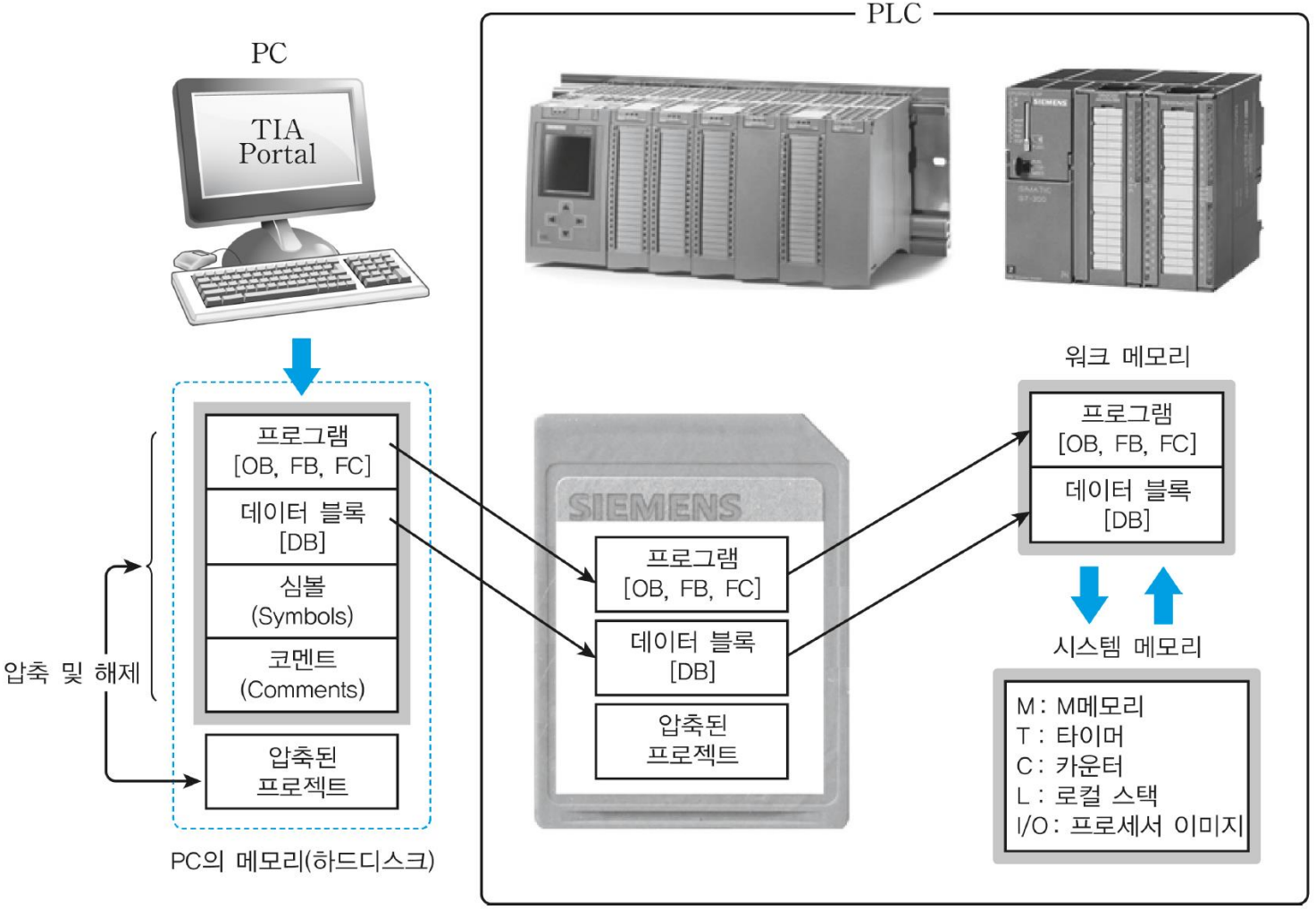
이 정 석

인하공업전문대학 메카트로닉스공학과

강의 순서 및 내용

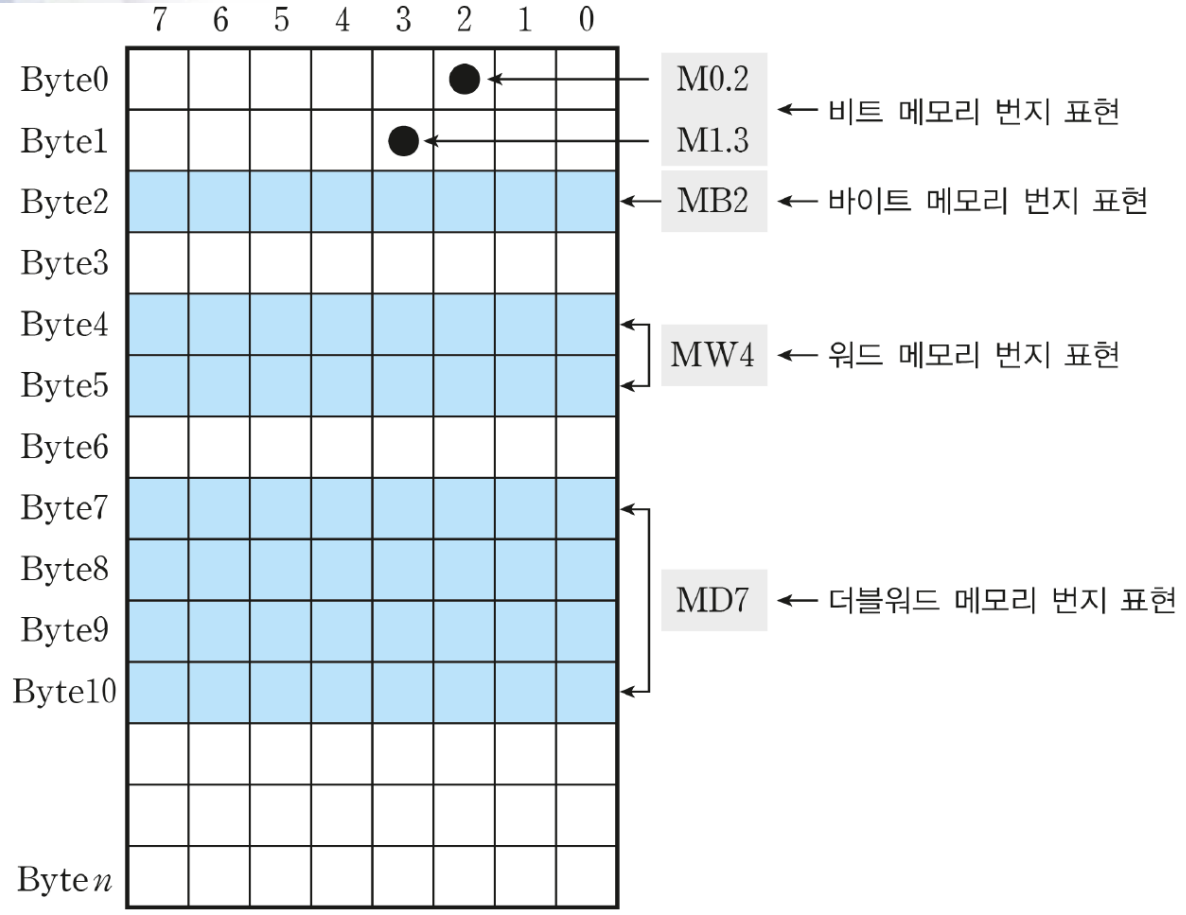
- TIA(Totally Integrated Automation) Portal 기능 및 내용
- SIEMANS PLC 메모리 및 프로그램 내용

❖ 메모리 사용법이 다르다.



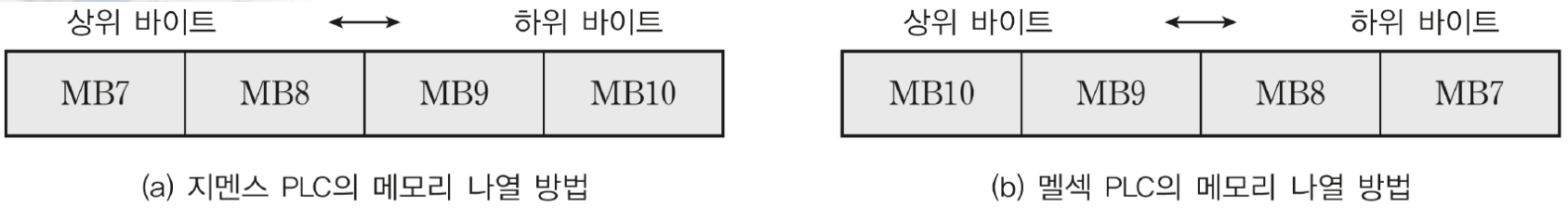
[그림 2-11] 지멘스 PLC에서 사용하는 메모리의 구조

- M메모리의 기본단위는 바이트
  - 비트, 바이트, 워드, 더블워드를 조합해서 사용

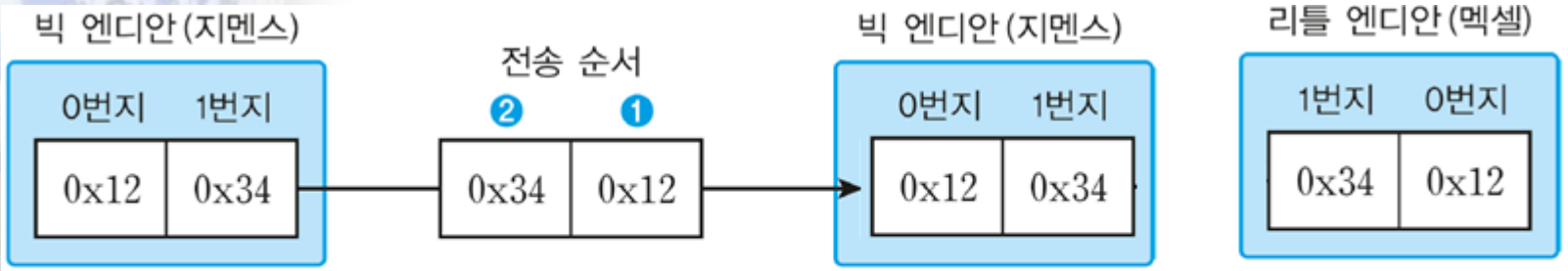


[그림 2-14] M메모리의 비트, 바이트, 워드, 더블워드 단위의 번지 표현 방식

❖ 바이트 단위의 메모리를 워드 및 더블 워드로 나열하는 방법이 다르다.



[그림 2-15] 지멘스와 멜섹의 메모리 나열 방법



[그림 2-17] 빅 엔디안 방식의 PLC에서 리틀 엔디안 방식의 PLC로의 데이터 전송

32비트 크기 16진수 : FF000205를 메모리에 저장

빅 엔디안(big-endian) 방식

메모리 번지 (바이트 단위)	7	6	5	4	3	2	1	0	비트번지
06									
05									
04									
03	0	0	0	0	0	1	0	1	
02	0	0	0	0	0	0	1	0	
01	0	0	0	0	0	0	0	0	
00	1	1	1	1	1	1	1	1	

리틀 엔디안(little-endian) 방식

메모리 번지 (바이트 단위)	7	6	5	4	3	2	1	0	비트번지
06									
05									
04									
03	1	1	1	1	1	1	1	1	
02	0	0	0	0	0	0	0	0	
01	0	0	0	0	0	0	1	0	
00	0	0	0	0	0	1	0	1	

[그림 2-16] 빅 엔디안 방식과 리트 엔디안 방식

- 프로그램 작성에 필요한 메모리를 PLC O/S로 부터 할당 받아서 사용
  - 프로그램에 필요한 메모리 사용 내역서를 작성

LAMP_FAN_FLICKER

	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Input				☐	☐	☐	☐
2	ON	Bool	false	Non-retain	☑	☑	☑	☐
3	OFF	Bool	false	Non-retain	☑	☑	☑	☐
4	<Add new>				☐	☐	☐	☐
5	Output				☐	☐	☐	☐
6	LAMP	Bool	false	Non-retain	☑	☑	☑	☐
7	FAN	Bool	false	Non-retain	☑	☑	☑	☐
8	WAR_LAMP	Bool	false	Non-retain	☑	☑	☑	☐
9	<Add new>				☐	☐	☐	☐
10	InOut				☐	☐	☐	☐
11	<Add new>				☐	☐	☐	☐
12	Static				☐	☐	☐	☐
13	K1	Bool	false	Non-retain	☑	☑	☑	☐
14	K2	Bool	false	Non-retain	☑	☑	☑	☐
15	K3	Bool	false	Non-retain	☑	☑	☑	☐
16	T1_Q	Bool	false	Non-retain	☑	☑	☑	☐
17	T1_ET	Time	T#0ms	Non-retain	☑	☑	☑	☐
18	IEC_Timer_0_Instance	TOF_TIME		Non-retain	☑	☑	☑	☐
19	IEC_Timer_0_Instance.	TON_TIME		Non-retain	☑	☑	☑	☐
20	<Add new>				☐	☐	☐	☐
21	Temp				☐	☐	☐	☐
22	<Add new>				☐	☐	☐	☐
23	Constant				☐	☐	☐	☐
24	<Add new>				☐	☐	☐	☐

[그림 2-18] FB에서 사용할 메모리를 O/S로부터 분할 받기 위한 메모리 사용 내역

자동화시스템 제어 실습(3주차)

- 메모리의 종류에 따라 보관된 데이터의 보존기간이 다르다.
  - 모든 프로그램 영역에서 사용할 수 있고 데이터의 보존기간이 PLC의 전원이 ON된 모든 시간
    1. 시스템 메모리
    2. 워크 메모리에서 만들어지는 글로벌 메모리
  - 해당 프로그램 영역(FB)에서만 사용할 수 있고 데이터의 보존기간이 PLC의 전원이 ON된 모든 시간
    1. Static
  - 해당 프로그램 영역(FB, FC)에서만 사용할 수 있고 데이터의 보존기간이 1스캔 시간
    1. Temp

❖ 메모리 크기와 데이터 타입 및 데이터 표현 방법

[표 2-6] 지멘스 PLC에서 사용하는 메모리의 크기 및 데이터 타입과 데이터 표현 방법

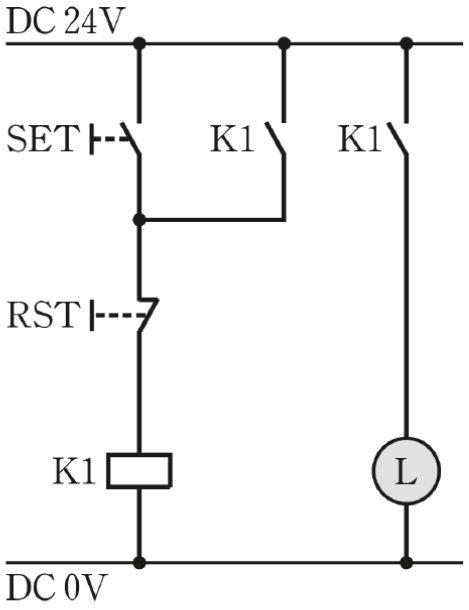
데이터 타입	크기	표현 방법	범위 및 숫자 표시 (최저값 ~ 최고값)
BOOL	1비트	비트	FALSE / TRUE
BYTE	8비트	8비트 16진수	B#16#0 ~ B#16#FF / 16#0 ~ 16#FF
CHAR	8비트	ASCII 문자	'A', 'B' 등
WORD	16비트	16비트 16진수	W#16#0 ~ W#16#FFFF 16#0 ~ 16#FFFF
		16비트 2진수	2#0000000000000000 ~ 2#1111111111111111
		카운터 값(BCD)	C#0 ~ C#999
		무부호(unsigned) 2개의 8비트	B#(0,0) ~ B#(255,255)

자동화시스템 제어 실습(3주차)

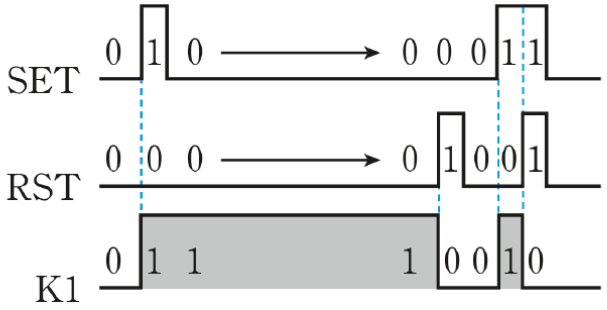
DWORD	32비트	32비트 16진수	DW#16#00000000 ~ DW16#FFFFFFFF 16#00000000 ~ 16#FFFFFFFF
		32비트 2진수	2#00000000 … 00000000 ~ 2#11111111 … 11111111
		무부호 4개의 8비트	B#(0,0,0,0) ~ B#(255,255,255,255)
INT	16비트	정수	-32768 ~ +32767
DINT	32비트	정수	L#-2147483648 ~ L#+214748367
REAL	32비트	실수(부동소수점 수)	최대 : ±3.402823e+38 최소 : ±1.175495e-38
S5TIME	16비트	SIMATIC 형식의 시간 값 10ms 간격	S5T#0H_0M_0S_10MS ~ S5T#2H_46M_30S_0MS
TIME	32비트	IEC 형식의 시간 값 1ms 간격	T#-24D20H31M23S648MS ~ T#24D20H31M23S647MS
DATE	16비트	날짜 (1일 간격)	D#1990-01-01 ~ D#2168-12-31 DATE#1990-01-01 ~ DATE#2168-12-31
TIME_OF_DAY	32비트	실제 시간	TOD#0:0:0 ~ TOD#23:59:59.999 TIME_OF_DAY#23:59:59.999

- 메모리 사용방법에 따라 프로그램 작성법이 다양
  - 자기유지회로를 PLC프로그램으로 작성할 때 사용되는 메모리에 따른 프로그램 작성방법의 차이점

- 리셋 우선 자기유지 회로



(a) 자기유지 회로



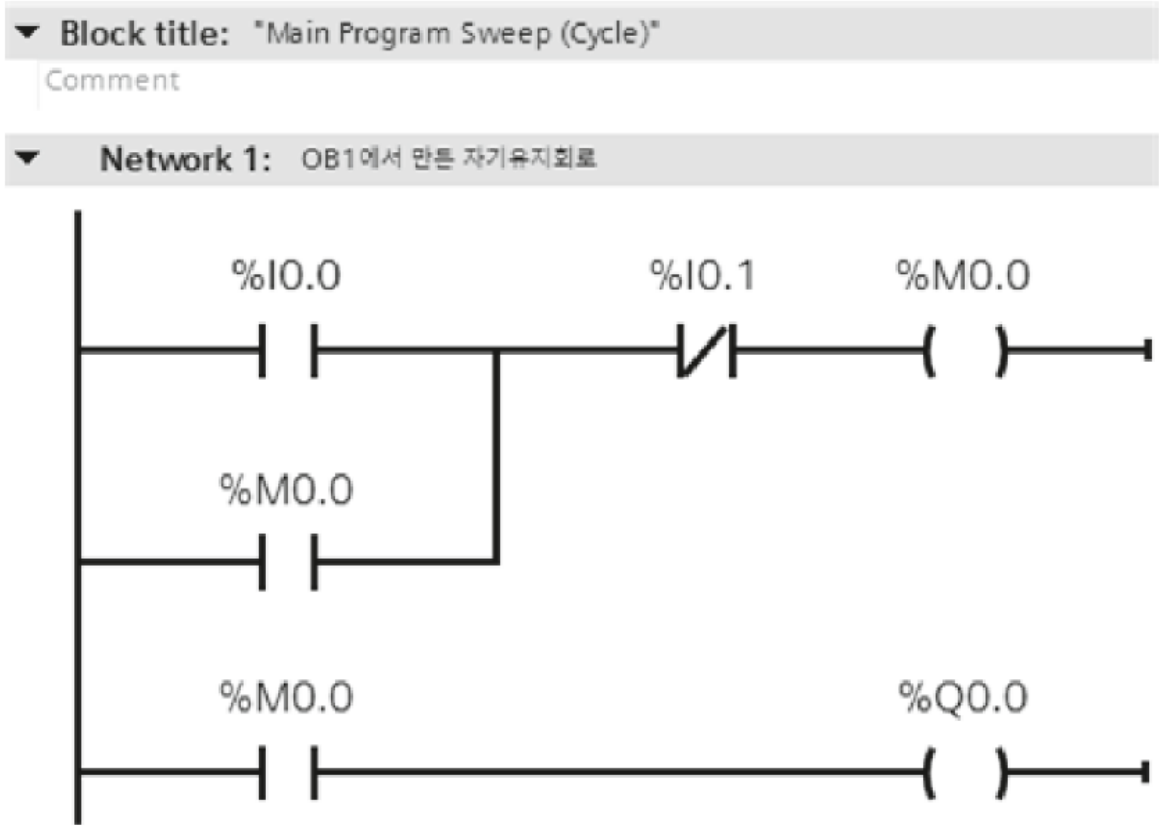
(b) 타임차트

IN		OUT	
변수명	번지	변수명	번지
SET	I0.0	L	Q0.0
RST	I0.1		

(c) 입출력 번지 할당

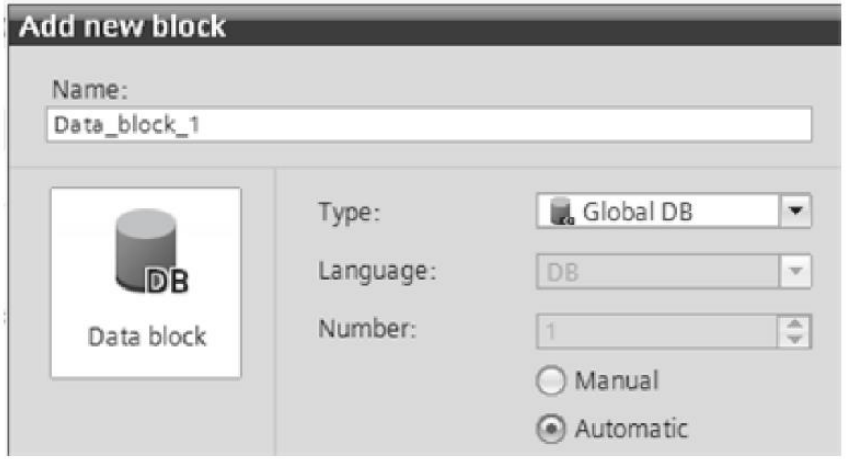
[그림 2-20] 리셋 우선 자기유지 회로

❖ 시스템 메모리 M을 이용한 자기유지회로 프로그램 작성하기

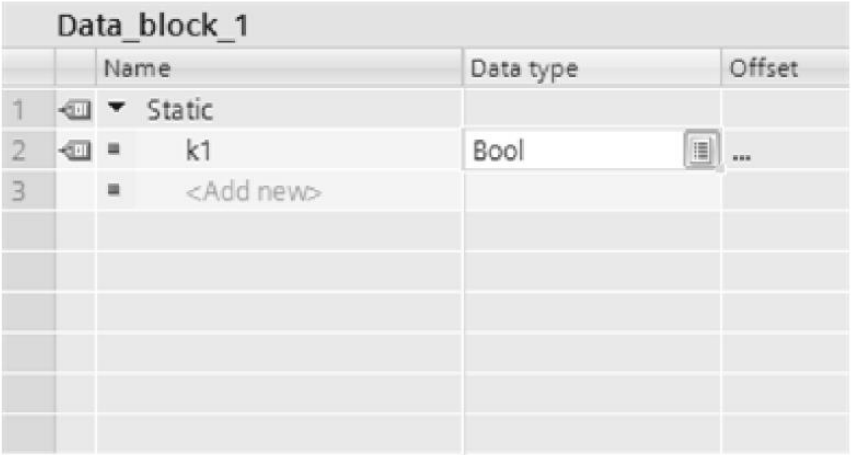


[그림 2-21] OB1에서 만든 자기유지 회로 프로그램

❖ 글로벌 DB에서 만든 K1태그를 사용한 자기유지회로 프로그램



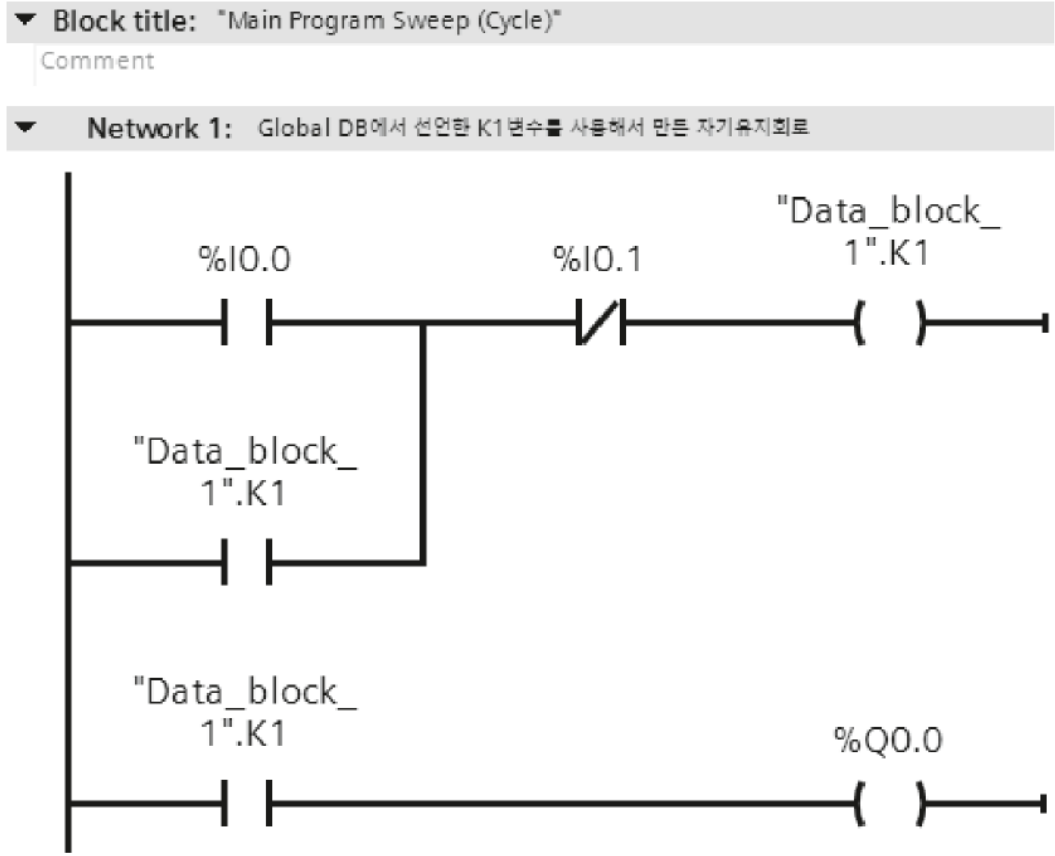
(a) DB 생성



(b) 사용할 메모리 내역서 작성

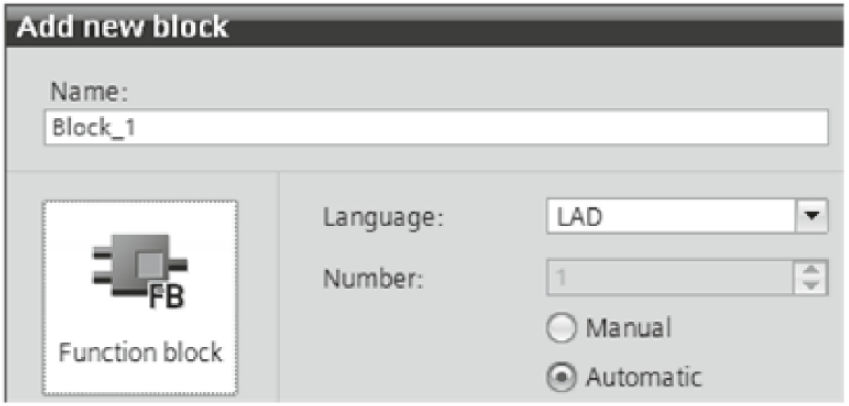
[그림 2-22] OB1에서 사용할 변수를 Global DB로 만드는 방법

❖ 글로벌 DB에서 만든 K1태그를 사용한 자기유지회로 프로그램



[그림 2-23] OB1에서 DB를 사용해서 만든 자기유지 회로

S7-1500에서 프로그램 작성 방법

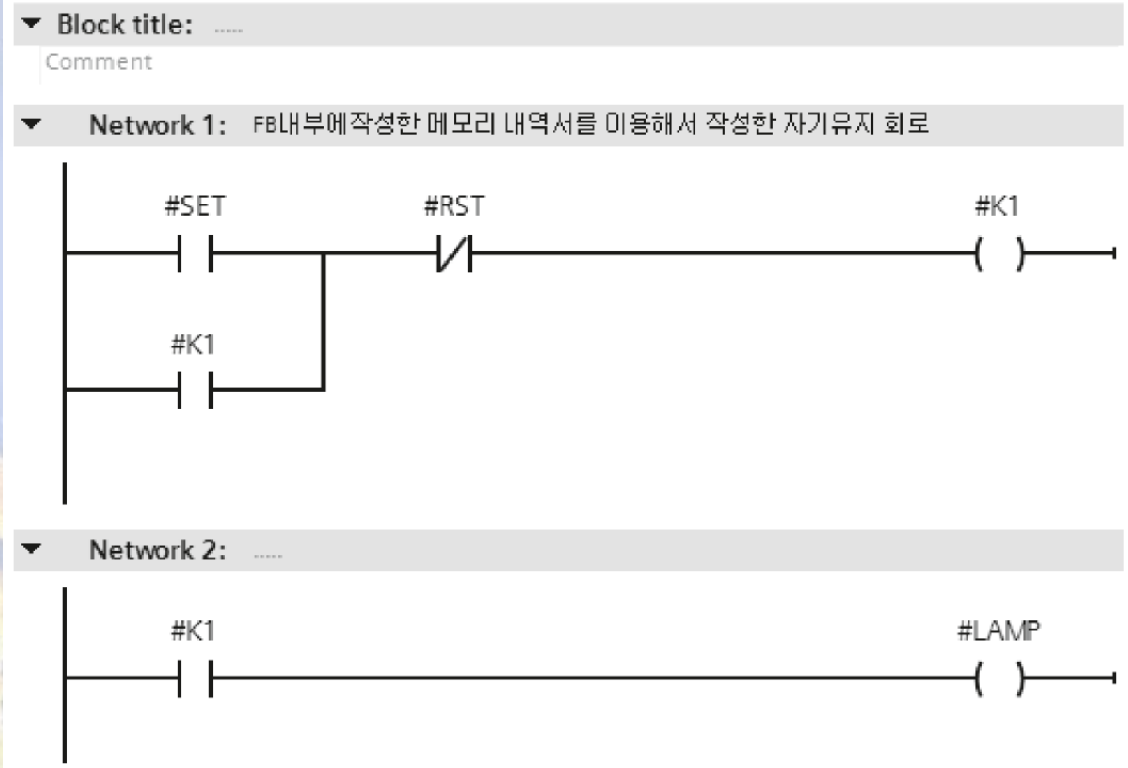


(a) FB를 생성

(b) FB1 내부에서 사용할 메모리에 대한 내역서 작성

[그림 2-24] FB에서 자기유지 회로 프로그램 작성

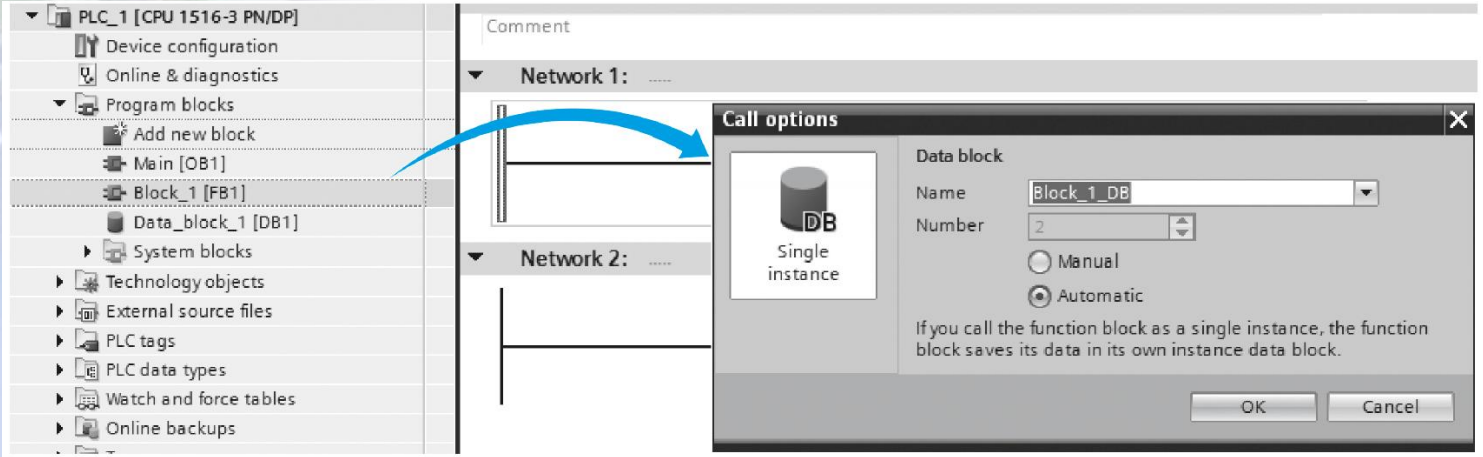
S7-1500에서 프로그램 작성 방법



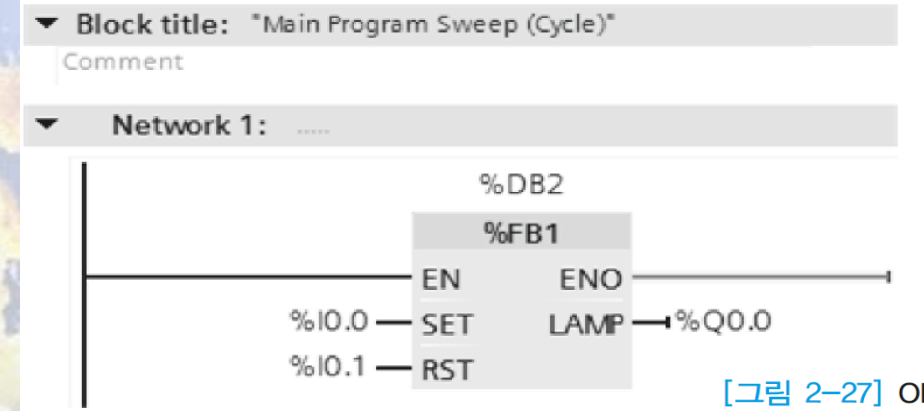
[그림 2-25] FB에서 작성한 자기유지 회로

S7-1500에서 프로그램 작성 방법

❖ OB1에서 FB호출하기 및 입출력 설정



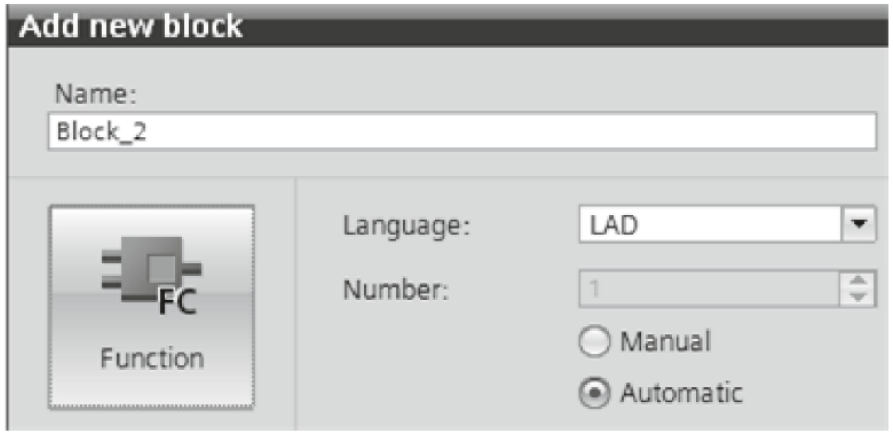
[그림 2-26] OB1에서 FB1을 호출할 때의 DB 할당



[그림 2-27] OB1에서 FB1을 호출한 모습

S7-1500에서 프로그램 작성 방법

❖ FC에서 Global DB의 K1태그를 사용한 자기유지회로



(a) FC를 생성

Block_2

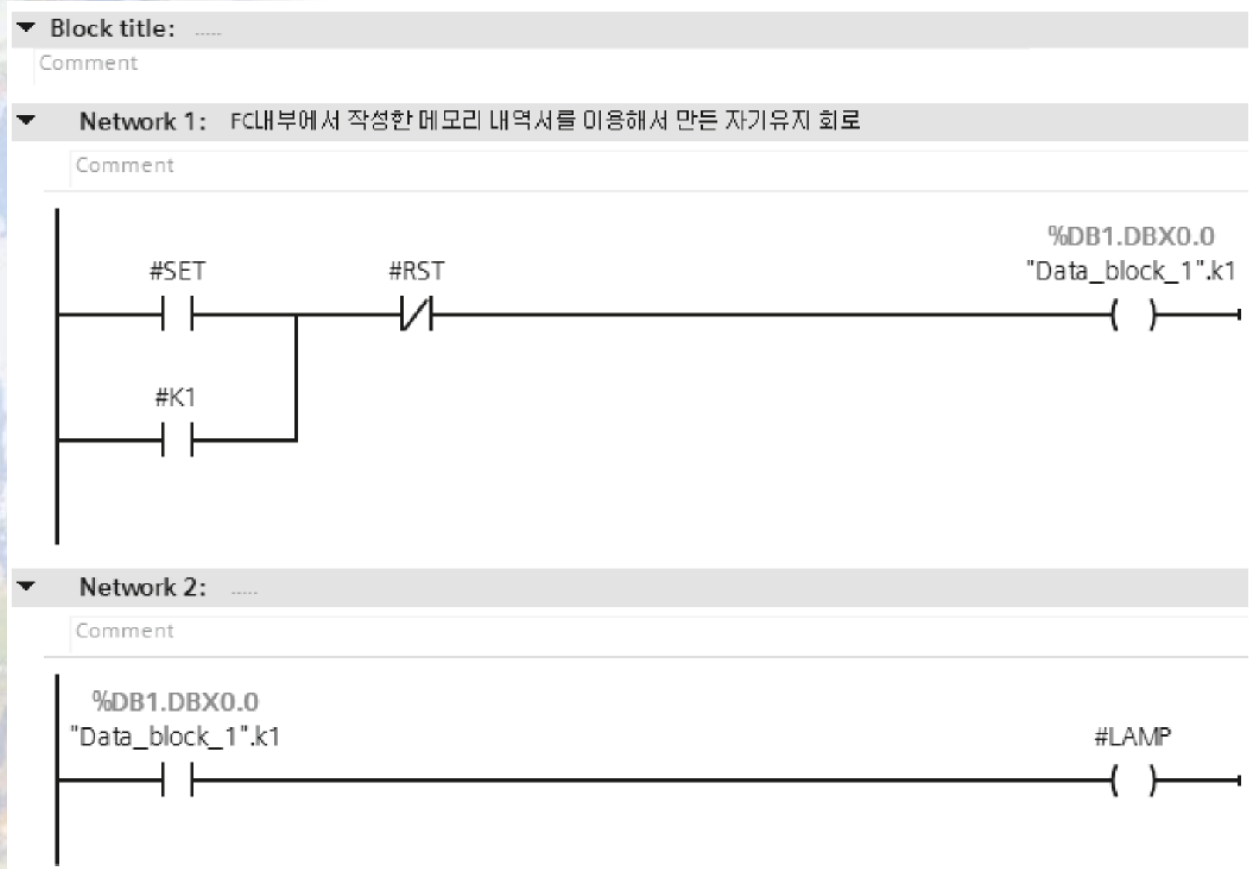
		Name	Data type	Offset
1		Input		
2		SET	Bool	
3		RST	Bool	
4		Output		
5		LAMP	Bool	
6		InOut		
7		<Add new>		
8		Temp		
9		K1	Bool	...
10		Constant		
11		<Add new>		
12		Return		
13		Block_2	Void	

(b) FC 내부에서 사용할 메모리 내역서 작성

[그림 2-28] FC에서 Global DB에 선언된 변수를 사용한 자기유지 회로 프로그램 작성

S7-1500에서 프로그램 작성 방법

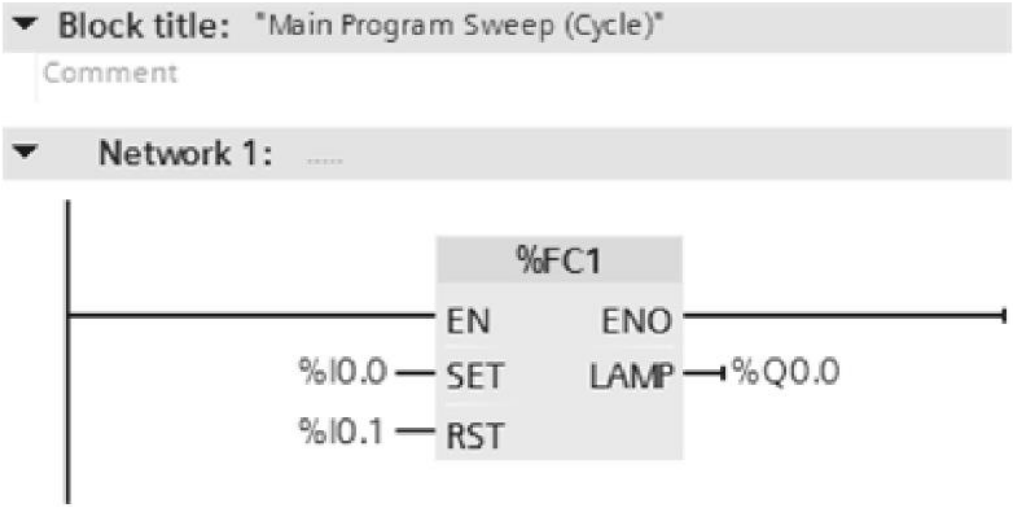
❖ FC에서 Global DB의 K1태그를 사용한 자기유지회로



[그림 2-29] FC에서 작성한 자기유지 회로 프로그램

S7-1500에서 프로그램 작성 방법

- OB1에서 FC1을 호출하고 입출력 번지 지정



[그림 2-30] OB1에서 FC1 호출


Thank You