📖 pv-MATH

🗁 계산의 목적 205

🗁 계산의 의미 206

🗁 알고리즘 정의 207

🗁 수식 표현 208

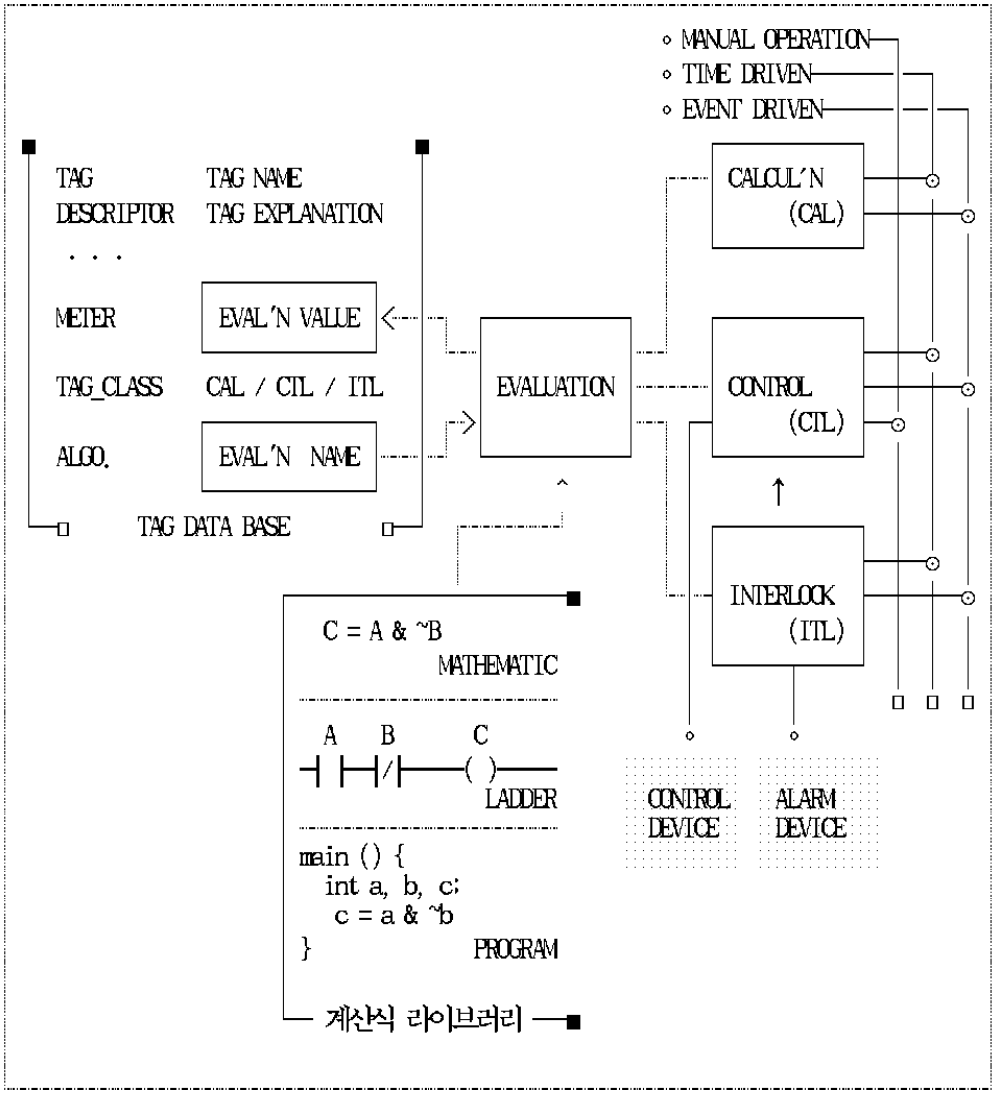
🗁 수식 등록 211

🗁 래더 다이어그램 213

계산의 목적

태그값을 계산(DERIVATION)하거나 기동시의 초기조건(CONTROL)과 동작 중의 운전 조건(INTERLOCK)을 점검하기 위한 일련의 알고리즘이 정의되고 수행 될 수 있도록 한다.

따라서 이장에서는 태그값(DERIVATION), 기동 조건(CONTROL) 그리고 운전 조건(INTERLOCK) 알고리즘(CALCULATION ALGORITHM)의 정의와 표현에 대하여 설명하겠다.



<그림1> 계산 식의 이용

계산의 의미

태그값의 계산 DERIVATION

계산 태그(CALCULATION TAG)는 실측 태그(PHYSICAL TAG)와 상반되는 의미로서, 실측 태그는 실측값인데 반하여 이 계산 태그는 하나 이상의 실측값들의 조건과 상관 관계에 의한 계산치로 표현된다, 즉 관련 실측 태그의 여러 가지의 (FACTOR)를 조합하여 실존 태그가 아닌 가상(VIRTUAL)값이 계산(CALCULATION)되어 유도(DERIVED)되기 때문에 "계산 태그 : CALCULATION TAG", "가상 태그 : VIRTUAL TAG" 혹은 " 유도 태그 : DERIVED TAG"라고 불린다.

기동조건 계산 CONTROL

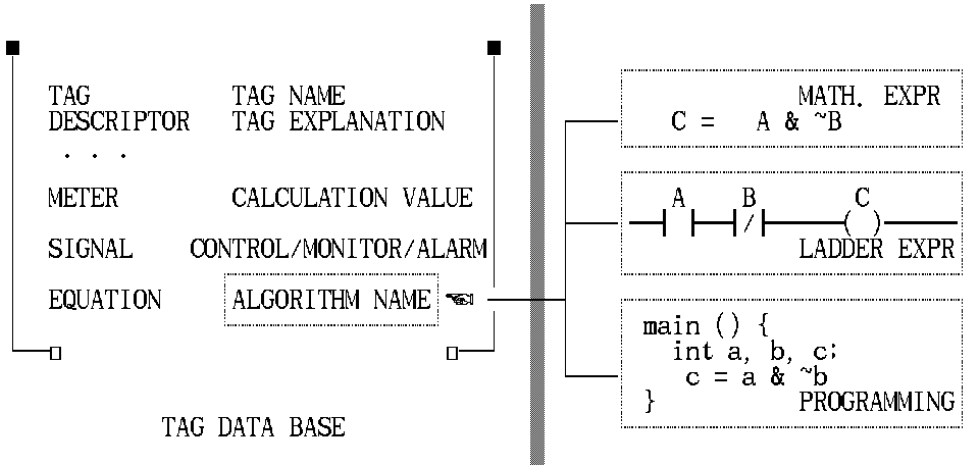
기기를 제어하기 위해서는 해당 기기가 정상으로 기동 될 수 있는지를 점검 한 후 제어를 하게 되고 기동 중에도 일련의 동작 단계마다 관련 모든 상태를 확인하면서 계속 진행 여부를
판단한다.이런 일련의 동작명령과 상태점검의 흐름을 구현하는 응용이다.

운전조건 계산 INTERLOCK

동작 중의 기기가 정상으로 운전될 수 있는 관련 신호들을 일정 시간마다 점검하고 신호들의 상관관계를 분석하고 비정상인 경우 경보를 발생하고 관련 제어를 수행할 수 있도록 한다.

알고리즘 정의

알고리즘(ALGORITHM)을 표현한 실체의 이름을 태그 데이터 베이스 내의 해당 계산 태그가 정의된 "EQUATION" 필드(FIELD)에 등록함으로써 정의된다. 여기서 알고리즘(ALGORITHM)은 래더 다이어그램(LADDER DIAGRAM)과 수식(MATHEMATIC EXPRESSION)에 의해 표현되고, 특수한 계산의 경우 프로그래밍(PROGRAMMING)으로 구현하여야 한다.



<그림2> TAG내의 ALGORITHM정의

수식 표현

식 표현의 규칙

수식 표현(MATHEMATIC EXPRESSION)은 산술 수식(ARITHMETIC EXPRESSION)과 부울 대수(BOOLEAN ALGEBRA) 양자를 의미하며 이 표기 방식들은 래더 다이어그램(LADDER DIAGRAM) 정의나 프로그래밍(PROGRAMMING)에 비해 절차상 간단하다. 연산자(OPERATOR)가 하나 이상 나오게 되면 "<표2> OPERATOR PRECEDENCE"의 연산자 우선 순위 규칙(OPERATOR PRECEDENCE)을 이해하고 적용해야 한다

**, unary +, unary -, ~
*, /
|, ^, >>, <<, +, //, &, -
>=, >, ==, <=, <, !=
&&
||
=

<OPERATOR PRECEDENCE >

	SIN( )	sine
	COS( )	cosine
삼각	TAN( )	tangent
함수	ASIN( )	inverse sine
	ACOS( )	inverse cosine
	ATAN( )	inverse tangent
	SINH( )	hyperbolic sine
쌍곡선	COSH( )	hyperbolic cosine
함수	TANH( )	hyperbolic tangent
지수	LOG10( )	logarithm
함수	LOGE( )	natural logarithm

< 표준 함수(STANDARD FUNCTION) >

대입	=	assign		
괄호	(	left parenthesis	)	left parenthesis
부호	+	unary plus	-	unary minus
산술	+	plus	-	minus
	*	multiplication	/	divide
	**	exponent		
관계	>	greater than	>=	greater than equal
	<	less than	<=	less than equal
동등	==	equal	!=	not equal
논리	&&	and	||	or
	~	not		
비트	&	and	|	bit
	<<	shift left	>>	shift right
	^	xor	~	complement
	//	bit concatenation		
조건	IF	pseudo instruction		

< 연산자(OPERATOR CONVENTION) >

피 연산자

수식(MATHEMATIC EXPRESSION)에 사용되는 피 연산자(OPERAND)는 크게 아나로그 태그(ANALOG TAG)와 디스크리트 태그(ANALOG TAG) 이름과 상수(CONSTANT)로 이루어진다.

여기서 태그명은 일반 변수명과 구분하기 위한 목적과 태그 데이터베이스는 아나로그와 디스크리트를 별도의 데이터 화일에 정의하기 때문에 이 두 데이터 화일에서 중복되는 태그명이 있을 수 있다. 따라서 수식 내에서의 태그명은 접두사(PREFIX)인 아날로그와 디스크리트에 대해 "A$"와 "D$"를 붙여 "예"와 같이 구분한다.

일반 변수명 예) : NORMAL_VARIABLE

태그 변수명 예) : DTAG_VAR_NAME, ATAG_VAR_NAME

표현의 예

밸브를 열었을 경우(FULL OPEN STATE) "INLET_VALVE_OP" 근접 스위치가 ON("1")되고 닫혔을 경우(FULL CLOSE STATE) "INLET_VALVE_CL" 근접 스위치가 ON("1")됨에 따라 밸브의 상태를 감시하기 위해 "<그림 3> 신호의 흐름"과 같이 도식화된다.

현장신호		기기상태	GUI표현	
열림 스위치	닫힘 스위치	밸브 상태	계산 태그값	밸브 색깔
0	0	RUN	0	노랑
0	1	CLOSE	1	파랑
1	0	OPEN	2	자주
1	1	ERROR	3	빨강

<그림3> 신호의 흐름

GUI(GRAPHIC USER INTERFACE)에서 밸브 상태를 표현하기 위해서는 열림 스위치와 닫힘 스위치의 실 신호들 간의 상관관계를 계산하여 밸브 상태인계산 태그값을 "<식1>밸브 상태 계산 표현"의 "표현 1" 혹은 "표현 2"와 같이 정의된다.

```
OPEN  = D$INLET_VALVE_OP
CLOSE = D$INLET_VALVE_CL
VALVE = D$INLET_VALVE
IF ( OPEN == 0 && CLOSE==0 ) VALVE = 0
IF ( OPEN == 0 && CLOSE==1 ) VALVE = 1
IF ( OPEN == 1 && CLOSE==0 ) VALVE = 2
IF ( OPEN == 1 && CLOSE==1 ) VALVE = 3
```

--- 표현 1

화일 명 : inlet-v-1.infix

```
OPEN  = D$INLET_VALVE_OP
CLOSE = D$INLET_VALVE_CL
VALVE = D$INLET_VALVE

VALVE(1:0) = OPEN(0:0) // CLOSE(0:0)
```

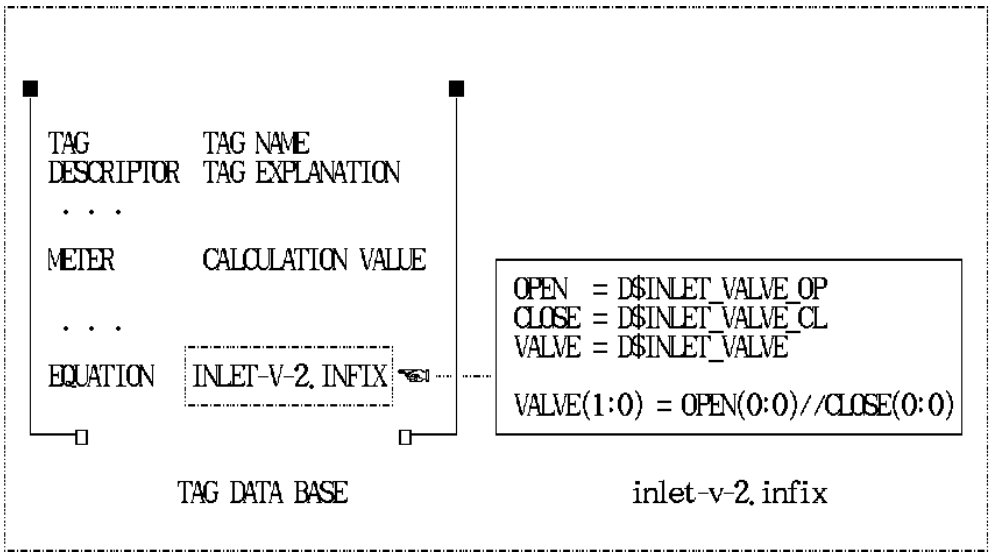
--- 표현 2

화일 명 : inlet-v-1.infix

< 식1 > 밸브 상태 계산 표현

수식 등록

구성된 수식은 "STNED(STATION EDITOR)" 툴(TOOL)에 의해 태그 데이터 베이스 내의 "EQUATION" 필드에 해당 수식의 이름을 기록해야 한다.



<그림4> 수식 등록

이 수식(INFIX EXPRESSION)은 데이터 베이스 빌더(BUILD) 과정에서 번역식(POSTFIX EXPRESSION)으로 변환된다. 그러나 수식표현이 데이터 베이스 빌드(BUILD)작업을 행한 후의 경우, 수식의 이름이 같고 수식 내용만 만 수정했다면 반드시 이 수식 화일에 대한 빌드(BUILD) 작업을 해야 한다. 왜냐하면 온라인 처리가 번역(INTERPRETER)방식이 아니라 번역된 코드를 수행 (EVALUATION)만하기 때문이다.

래더 다이어그램

래더의 의미

래더 다이어 그램(LADDER EXPRESSION)은 전기 혹은 제어 신호 배선 망 시퀜스의 구조적 표현과 논리적 의미로서 제어기기 프로그래밍 기법 중에 가장 많이 활용되고 있는 언어이다.

이는 모든 PC(PROGRAMMABLE CONTROLLER)에서 제공되어 제어 현장 실무자에 친숙한 TOOL이다.

래더 명령어

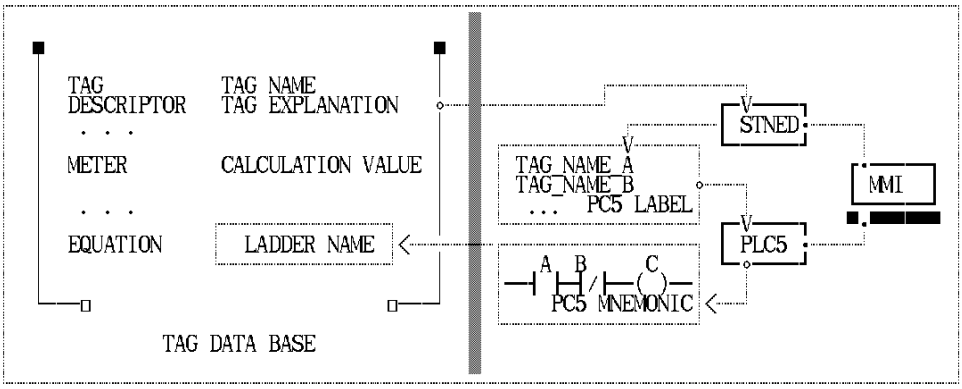
릴레이	XIC	a 접점(normal on)
	XIO	b 접점(normal off)
비교	EQU	equal(=)
	NEQ	not equal(◇)
	LES	less than(<)
	GRT	greater than(>)
	LEQ	less than or equal(<=)
	GEQ	greater than or equal(>=)
	=	n7:0 = n7:1
	◇	n7:0 ◇ n7:1
	<	n7:0 < 100
	>	n7:0 > 100
	<=	n7:0 <= n7:2
	>=	n7:0 >= n7:3
산술	ADD	' + '
	SUB	' - '
	MUL	' * '
	DIV	' / '
	NEG	' ~ '
	CLR	n7:0 = 0
대입	MOV	' = '

<표4> 래더 명령어

래더 에디터

래더 다이어 그램(LADDER EXPRESSION)은 "STNED" 내에서"PLC5"의 옵션(OPTION) 프로그램으로 작성되며 사용 가능한 명령어는 위에서 기술된 연산 자로 표현된다.

피연산자는 "PLC5" 어드레스, 태그 명, 상수 등이 사용될 수 있으며, "PLC5"에서 래이블(LABEL)사용은 "STNED"에디터에서 자동으로 마련된다.

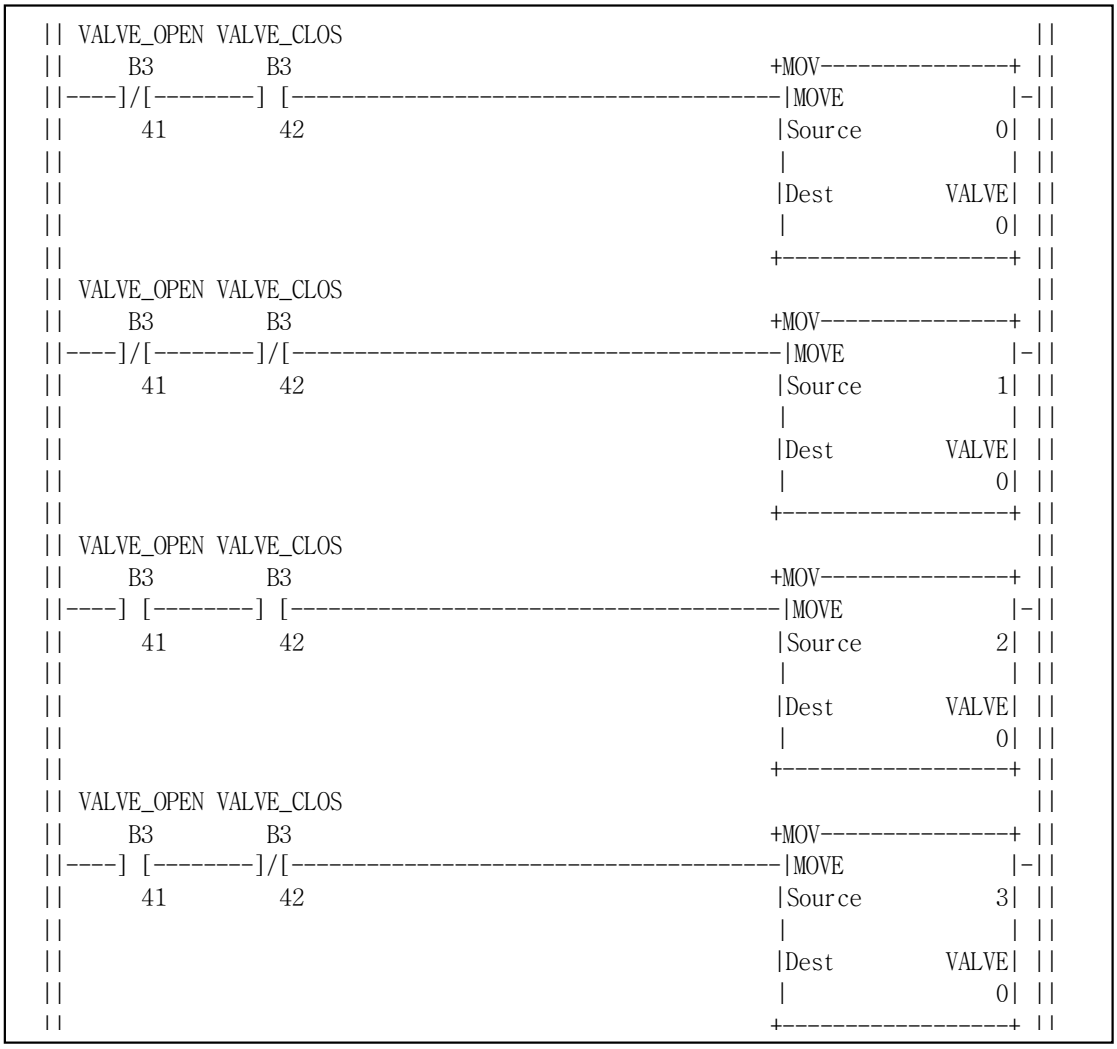


<그림 5> 래더 다이어 그램 처리 흐름

래더식의 예

```
| |                                                                    | |
| | VALVE_CLOS VALVE_OPEN +CMP---------------+          OPEN_COND | |
| |----] [-------]/[------|COMPARE           |-------------( )----| |
| |                       |Expression        |                    | |
| |                       |PRESS_01 <= 100   |                    | |
| |                       +------------------+                    | |
| | VALVE_OPEN VALVE_CLOS +CMP---------------+        CLOSE_COND | |
| |----] [-------]/[------|COMPARE           |-------------( )----| |
| |                       |Expression        |                    | |
| |                       |PRESS_01 <= 100   |                    | |
| |                       +------------------+                    | |
| |  OPEN_COND C_OPEN   DSP_01                          VALVE_OUT | |
| |----] [-------] [--+---]/[-------------------------------( )----| |
| |                   |                                           | |
| |  VALVE_OUT        |                                           | |
| |----] [------------|                                           | |
| | CLOSE_COND C_CLOSE                                    DSP_01 | |
| |----] [-------] [-----------------------------------------( )----| |
| | VALVE_OUT                                           VALVE_OPEN | |
| |----] [-------------------------------------------------( )----| |
| | VALVE_OUT                                           VALVE_CLOS | |
| |----]/[-------------------------------------------------( )----| |
```

<그림 6> "<그림 3>"과 "<식 1>"의 표현 1 에 대한 Ladder 식의 예


VALVE_OPEN VALVE_CLOS
B3 B3
41 42
MOV
MOVE
Source 0
Dest VALVE
0
VALVE_OPEN VALVE_CLOS
B3 B3
41 42
MOV
MOVE
Source 1
Dest VALVE
0
VALVE_OPEN VALVE_CLOS
B3 B3
41 42
MOV
MOVE
Source 2
Dest VALVE
0
VALVE_OPEN VALVE_CLOS
B3 B3
41 42
MOV
MOVE
Source 3
Dest VALVE
0

<그림 7> 래더식의 예 2