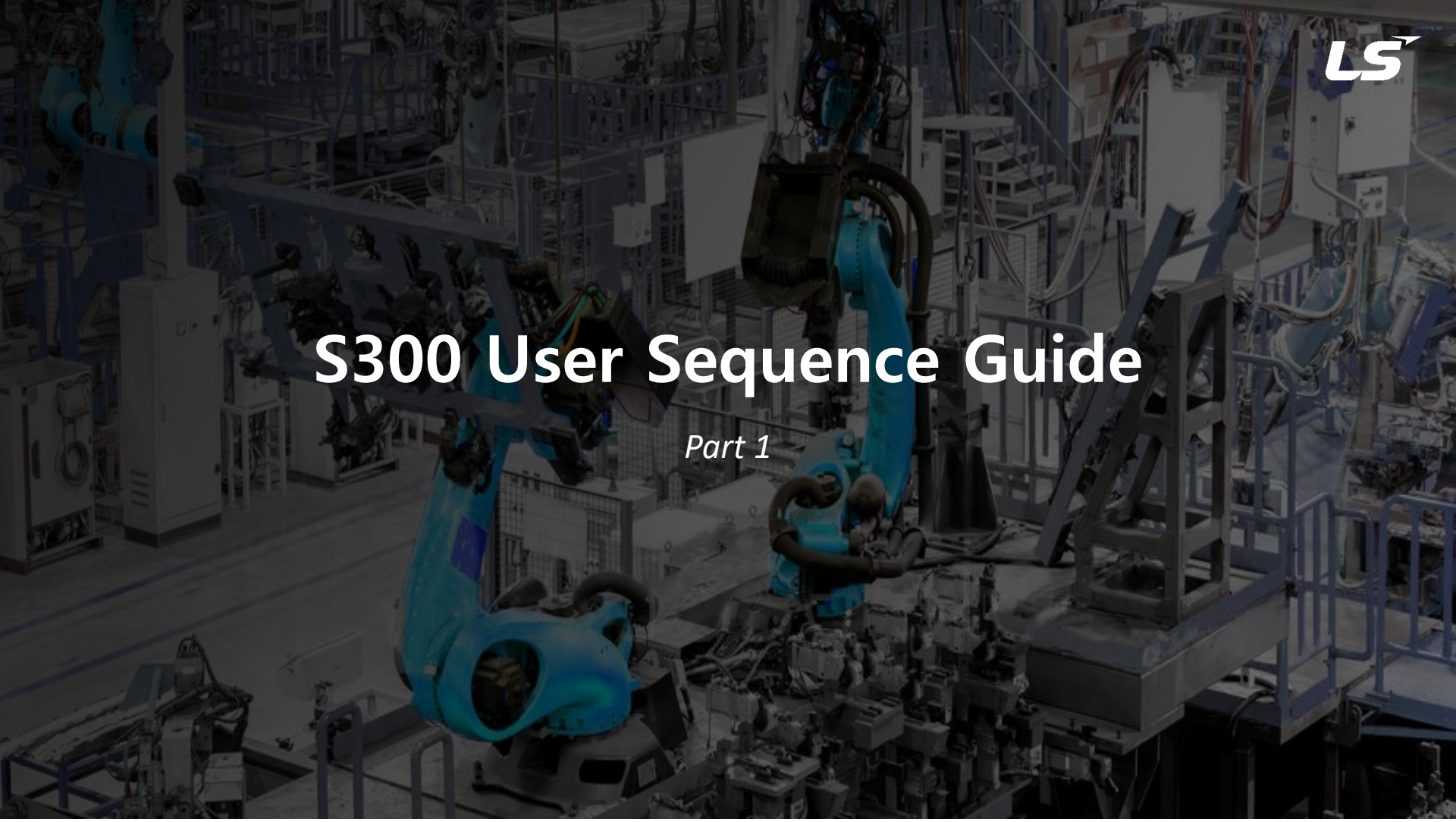
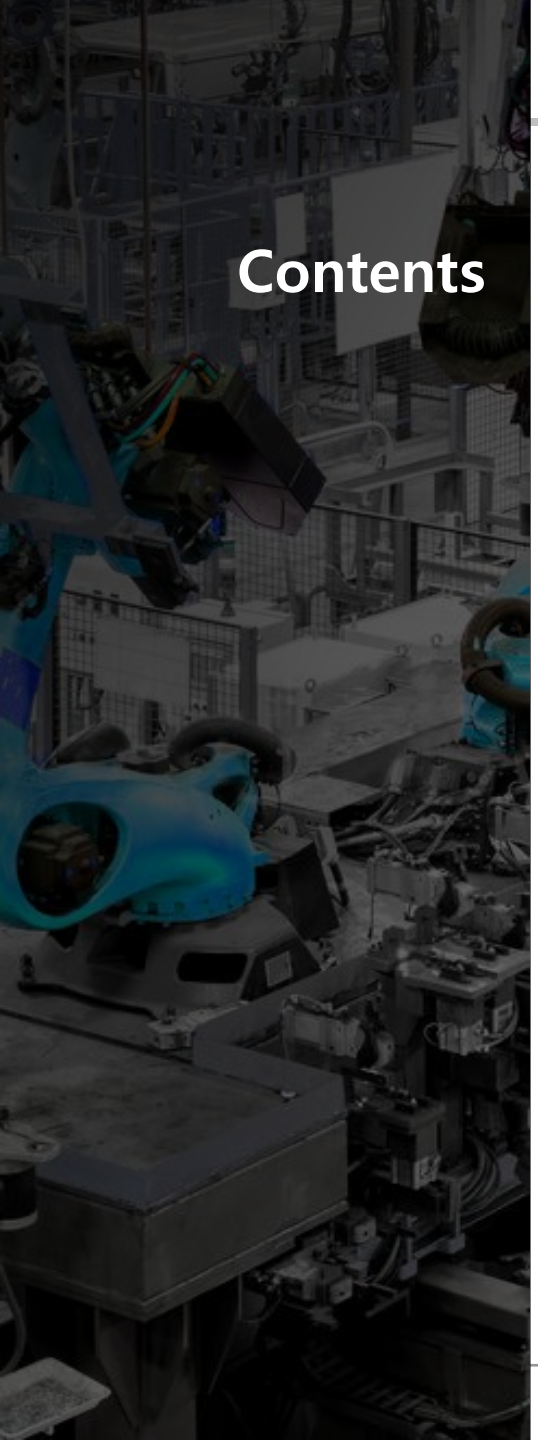


S300 User Sequence Guide

Part 1

LS ELECTRIC 기술팀
조재형M

2026-01-20



Contents

A. Theory

1. Overview
2. Basic Setup
3. Point connecting

B. Function Blocks

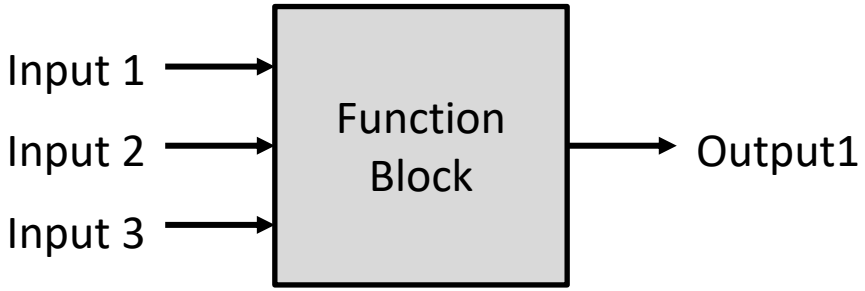
1. Value Function
2. Logic Function
3. Paraset
4. User Block
5. Source Select
6. Drive Block
7. End Block
8. Other Blocks

C. Examples

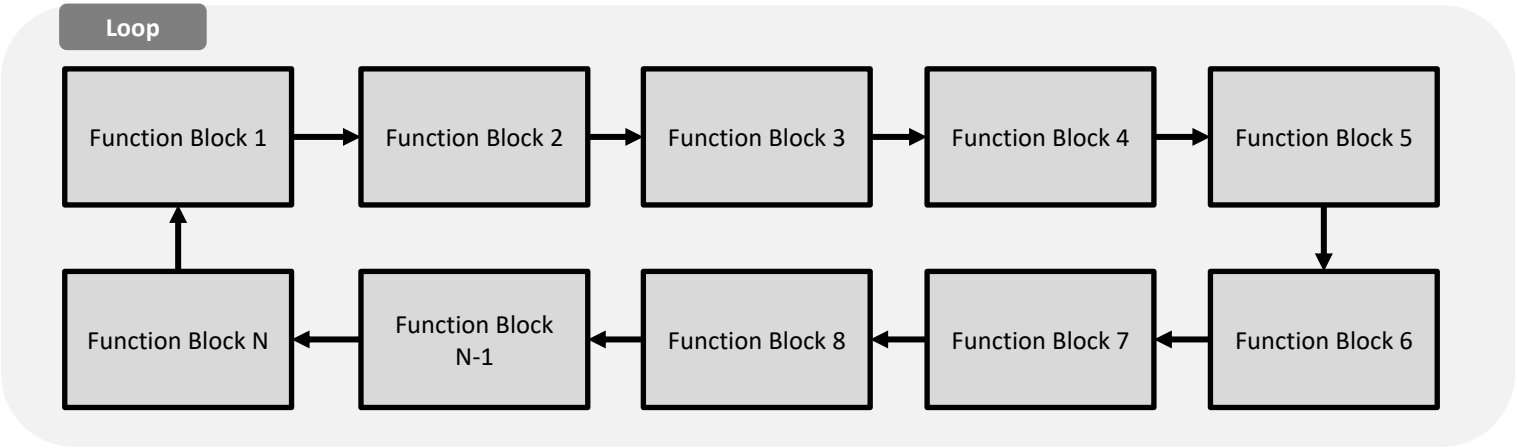
1. Traverse

Advanced User Sequence

- Overview (1)
  - User Sequence는 PLC를 사용하지 않고, 함수 블록(function block)을 이용해 내부적으로 수학/논리 시퀀스를 구성할 수 있는 기능입니다.
  - 함수 블록은 여러 개의 입력값과 여러 개의 출력값으로 구성되며, 입력과 출력의 개수는 블록 종류에 따라 달라집니다.
  - 함수 블록의 출력은 읽기 전용(read-only) 값이며, 주파수 지령(frequency reference), PID 피드백, 다른 블록의 입력 등으로 사용할 수 있습니다.



- 드라이브는 Function 1에서 시작해 마지막 함수에서 끝나는 순서로 User Sequence를 루프(반복) 형태로 실행합니다.
- User Sequence의 루프 시간(loop time)은 파라미터 US-02에서 조정할 수 있습니다.



Advanced User Sequence

- Overview (2)
  - User Sequence에는 5개의 파라미터 그룹이 있습니다.

Parameter Group	Feature
US	User Sequence 제어, 상태 모니터링
USL	로직(Logic) 함수 블록 관련 파라미터
USV	값(Value) 함수 블록 관련 파라미터
USP	Paraset 블록 관련 파라미터, 미사용(Void) 파라미터
USM	Mux/Demux/PI 제어 블록 관련 파라미터

Q & A

- Smart LCD Operator로 파라미터(설정) 복사 가능 여부

→ 불가능합니다. Smart LCD Operator로는 사용자 시퀀스 프로그램(User Sequence Program) 을 복사할 수 없습니다. 사용자 시퀀스 동작에는 DriveView의 시퀀스 도면(드로잉) 정보도 필요하므로, 복사는 DriveView 내에서만 가능합니다.

- Loop time이란?

→ Loop time은 최소 보장 실행 주기(minimum guaranteed execution time)입니다.
(예: 10 ms로 설정하면, 모든 Function Block이 10 ms 이내에 최소 1회 실행되는 것이 보장됩니다.)

- Loop time이 10 ms일 때, 한 루프 사이클 내에서 특정 파라미터/명령 값을 여러 번 바꾸면 오류가 나나요?

→ 오류는 발생하지 않습니다.
각 블록은 정해진 순서대로 실행되므로, 한 루프 사이클에서 값을 여러 번 변경하더라도 시퀀스에서 마지막에 실행된 블록이 설정한 값이 최종 결과가 됩니다.

Advanced User Sequence

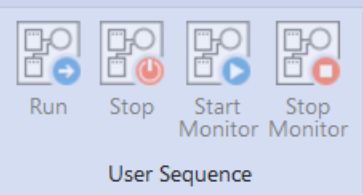
- Basic Setup Guide (1)
  - 디지털 입력을 사용해 User Sequence를 활성화 할 수 있고, 자동으로 실행되도록 설정할 수도 있습니다.

Code	Name	Default	Range
APP-03	UserSeq Enable	0	0: No
			1: Yes
US-01	UserSeqCtrl OnOff	0	0: Off
			1: ON
US-02	UserSeq Loop Time	10	10~1000 msec
US-03	All Blk Exec Time	0	0~65535 usec
US-04	User Seq Mode	0	0: Manual
			1: DI Dependent

- APP-03 UserSeq Enable → 1: Yes
- US-01 UserSeqCtrl OnOff → 사용자 시퀀스(User Sequence)를 실행할 준비가 되면 해당 값을 "1"(On) 또는 "2"(DI Dependent)로 설정하세요.
- US-02 UserSeq Loop Time → 최소 보장 실행 주기
- US-03 All Blk Exec Time →
- US-04 User Seq Mode → 값이 1(DI Dependent) 로 설정되어 있으면, 디지털 입력을 사용해 User Sequence를 On/Off 할 수 있습니다.

Advanced User Sequence

- Basic Setup Guide (2)
  - DriveView9에서는 도구 모음(Toolbar) 에서 사용자 시퀀스를 실행(Run), 정지(Stop), 모니터 시작(Start Monitor), 모니터 정지(Stop Monitor) 할 수 있습니다.



Monitoring Window

	Program	Function	Parameter	Value	Init
1	main	User Block1	1st Command Source	Keypad	Fx/Rx-1
2	main	User Block1	1st Freq Ref Src	Keypad	Keypac
3	main	User Block1	Acc Time	0.00	20.00
4	main	User Block1	Dec Time	0.00	30.00
5	main	Drive	DI1 Status	Off	Off
6	main	Drive	DI2 Status	Off	Off
7	main	Drive	Logic01 Output	0	0
8	main	Drive	Value01 Output	0	0
9	main	Drive	Acc Time	0.00	20.00
10	main	Drive	Dec Time	0.00	30.00
11	main	Drive	Forward Direction	Off	Off
12	main	Drive	Reverse Direction	Off	Off
13	main	Drive	Stopped	Off	Off
14	main	Drive	Acceleration	Off	Off
15	main	Drive	Deceleration	Off	Off
16	main	Drive	ConstRateOperating	Off	Off
17	main	Drive	Dec to stop	Off	Off
18	main	Drive	Tripped	Off	Off
19	main	Drive	Warned	Off	Off
20	main	Digital Input1	DI1 On Delay	0	0
21	main	Digital Input1	DI1 Off Delay	0	0
22	main	Digital Input1	DI1 NC/NO Sel	Normal Open	Normal Oper
23	main	Digital Input1	DI1 Define	None	FX

Advanced User Sequence

- Basic Setup Guide (3)

Code	Name	Default	Range
DRV-10	1st command source	1	0: Keypad
			1: Fx/Rx-1
			…
			7: User Sequence
DRV-11	1st Freq Ref Src	0	0: Keypad
			1: Analog Input1
			…
			9: User Sequence

User Sequence로 운전과 주파수를 제어하려면, 아래와 같이 설정해야 합니다.

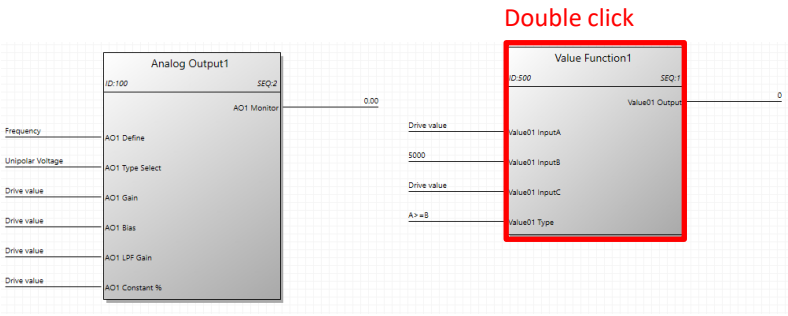
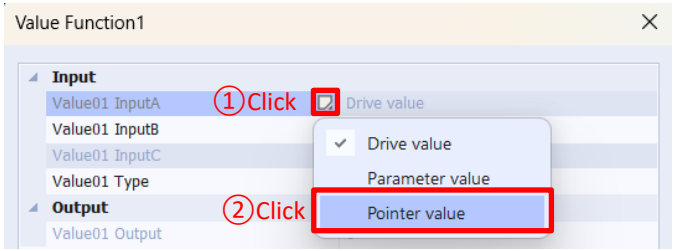
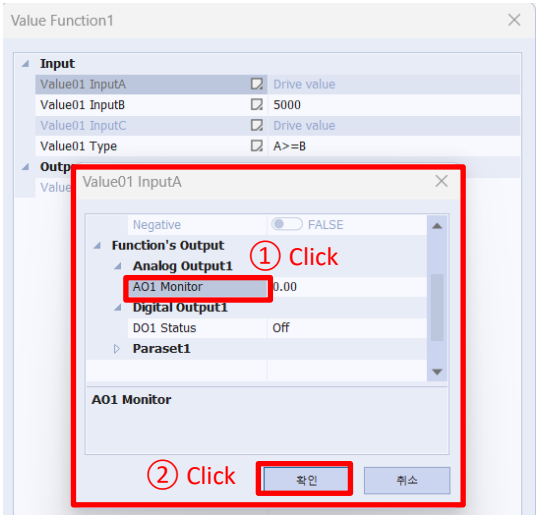
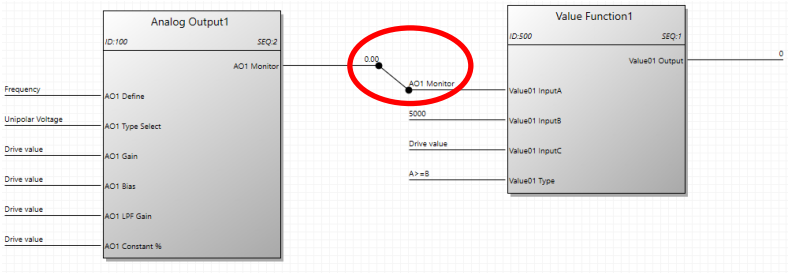
- DRV-10 1st command source → 7: User Sequence
- DRV-11 1st Freq Ref Src → 9: User Sequence

Code	Name	Default	Range
USP-61 ~ 90	Var 01 ~ 30	0	0~99,999,999

함수 블록 입력에 상수로 사용할 수 있도록, Void 파라미터를 Var 파라미터로 설정 합니다.

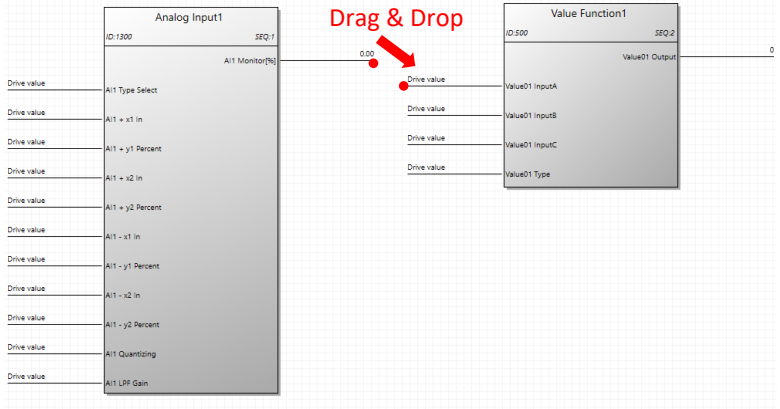
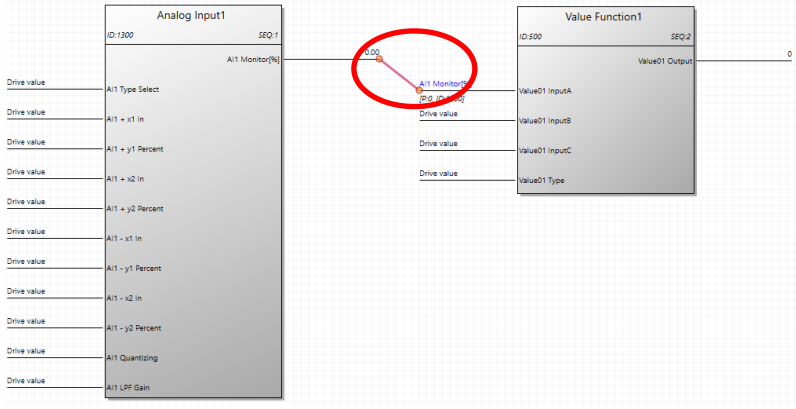
Advanced User Sequence

- Point connecting (1)
  - 입력을 pointer value로 변경한 뒤, 여러 함수 블록 출력 중 하나를 선택하면 두 지점을 연결할 수 있습니다.

Step 1	Step 2
	
Step 3	**Step 4**
	

Advanced User Sequence

- Point connecting (2)
  - 점을 하나 드래그해서 다른 점 위에 드롭하면 두 점이 연결됩니다.

Step 1	Step 2
	

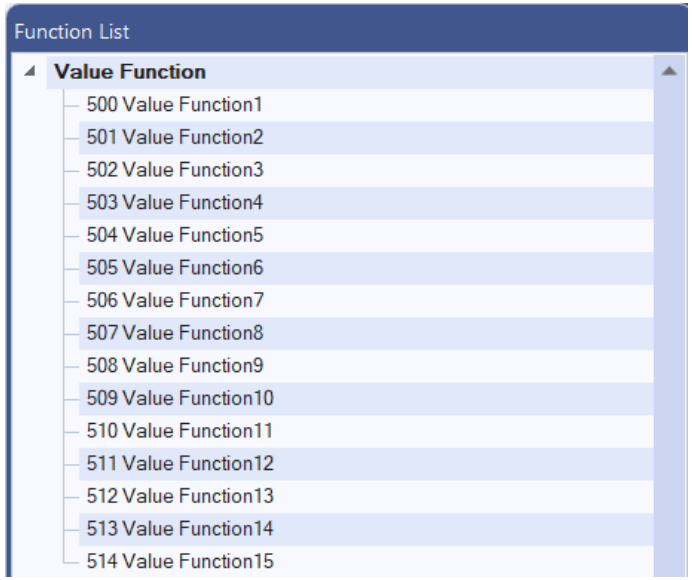
Advanced User Sequence

- Value Function Blocks (1)
  - Value 함수 블록은 입력 값 2~3개와 출력 값 1개로 구성됩니다.

Group	Name	Description	Range
USV	Value@※ Type	Choose the function to perform in the function block.	0~33
	Value@ Input A	A first input parameter.	0~99,999,999
	Value@ Input B	A second input parameter.	0~99,999,999
	Value@ Input C	A third input parameter.	0~99,999,999
	Value@ Ouptut	Output value.	Read Only

※@는 스텝 번호(1~15) 를 의미합니다.

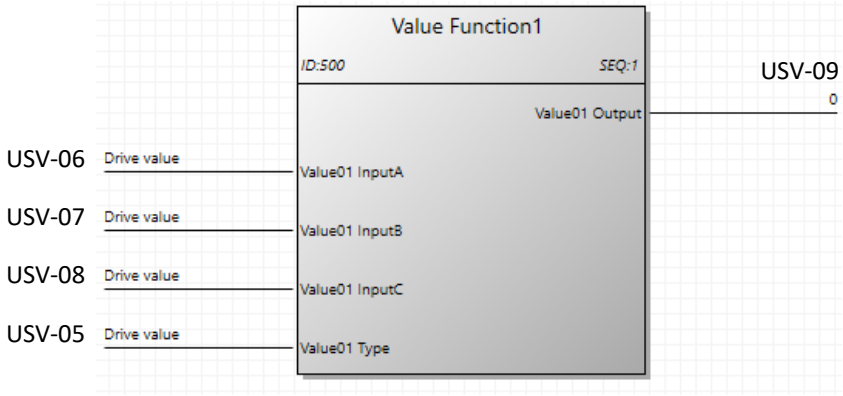
  - 34가지 기능으로 구성된 Value 함수 블록은 최대 15개까지 사용할 수 있습니다.

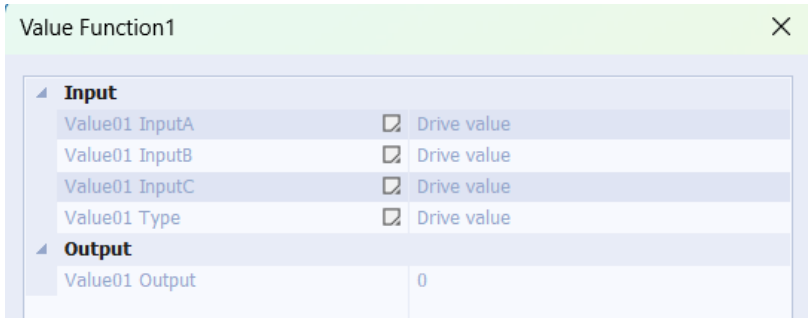


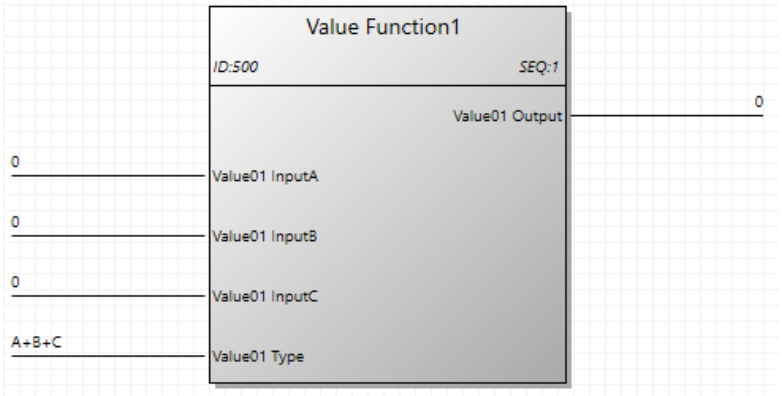
Advanced User Sequence

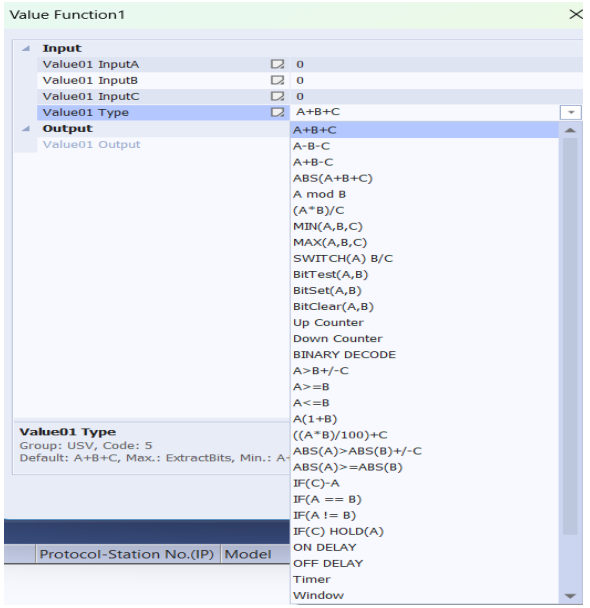
- Value Function Blocks (2)

Input Type = Drive Value	Input Type = Parameter Value









Advanced User Sequence

- Value Function Blocks (3)

No.	Type	Description
0	A+B+C	A+B+C를 출력합니다. Example) When A=3, B=2 and C=1, A+B+C=3+2+1=6
1	A-B-C	A-B-C를 출력합니다. Example) When A=3, B=2 and C=1, A-B-C=3-2-1=0
2	A+B-C	A+B-C를 출력합니다. Example) When A=3, B=2 and C=1, A+B-C=3+2-1=5
3	ABS(A+B+C)	A+B+C의 절대값을 출력합니다. Example) When A=1, B=-2 and C=-3, ABS(A+B+C)=ABS{1+(-2)+(-3)}=4
4	A mod B	A를 B로 나눈 나머지 값을 출력합니다. : A%B. Example) When A=5 and B=2, A mod B=5 mod 2=1
5	(A*B)/C	(A*B)/C를 출력합니다. Example) When A=3, B=2 and C=2, (A*B)/C=3*2/2=3 C가 0이면 0을 출력합니다.
6	MIN(A,B,C)	A, B, C 중 가장 작은 값을 출력합니다. Example) When A=3, B=2 and C=1, MIN(3,2,1)=1
7	MAX(A,B,C)	A, B, C 중 가장 큰 값을 출력합니다. Example) When A=3, B=2 and C=1, MAX(3,2,1)=3
8	SWITCH(A) B/C	A가 0이면 B, 0이 아니면 C를 출력합니다.

Advanced User Sequence

- Value Function Blocks (4)

No.	Type	Description
9	BitTest(A,B)	A의 B번째 비트가 1이면 1, 0이면 0을 출력합니다.
10	BitSet(A,B)	A의 B번째 비트를 1로 설정(Set)하여 출력합니다. B는 0~16 사이의 값으로 16 이상이면 16으로 인식합니다. B가 0이면 출력은 항상 0입니다.
11	BitClear(A,B)	A의 B번째 비트를 0으로 설정(Set)하여 출력합니다. B는 0~16 사이의 값으로 16 이상이면 16으로 인식합니다. B가 0이면 출력은 항상 A입니다.
12	Up Counter	A값의 변화에 따라 C조건으로 업을 세는 기능입니다. 출력 범위는 0~32767입니다. When C=0, upcount starts when the trigger input (A) changes from 0 to 1 (Rising edge). When C=1, upcount starts when the trigger input (A) changes from 1 to 0 (Falling edge). When C=2, upcount starts when the trigger input (A) changes.
13	Down Counter	A값의 변화에 따라 C조건으로 다운을 세는 기능입니다. 출력 범위는 0~32767입니다.

Advanced User Sequence

- Value Function Blocks (5)

<table>
<tr><th>No.</th><th>Type</th><th colspan="4">Description</th></tr>
<tr><td rowspan="9">14</td><td rowspan="9">Binary Decode</td><td>Input A</td><td>Input B</td><td>Input C</td><td>Output</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0 이 아닌 값</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0 이 아닌 값</td><td>0</td><td>2</td></tr>
<tr><td>0 이 아닌 값</td><td>0 이 아닌 값</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>0</td><td>0 이 아닌 값</td><td>4</td></tr>
<tr><td>0 이 아닌 값</td><td>0</td><td>0 이 아닌 값</td><td>5</td></tr>
<tr><td>0</td><td>0 이 아닌 값</td><td>0 이 아닌 값</td><td>6</td></tr>
<tr><td>0 이 아닌 값</td><td>0 이 아닌 값</td><td>0 이 아닌 값</td><td>7</td></tr>
<tr><td rowspan="4">15</td><td rowspan="4">A>B+/-C</td><td colspan="2">Condition</td><td colspan="2">Output</td></tr>
<tr><td colspan="2">A > (B+C)</td><td colspan="2">1</td></tr>
<tr><td colspan="2">A < (B-C)</td><td colspan="2">0</td></tr>
<tr><td colspan="2">(B-C) < A < (B+C)</td><td colspan="2">Keep</td></tr>
<tr><td>16</td><td>A>=B</td><td colspan="4">A>=B 이면 1, 아니면 0을 출력합니다.</td></tr>
<tr><td>17</td><td>A<=B</td><td colspan="4">A<=B 이면 1, 아니면 0을 출력합니다.</td></tr>
</table>

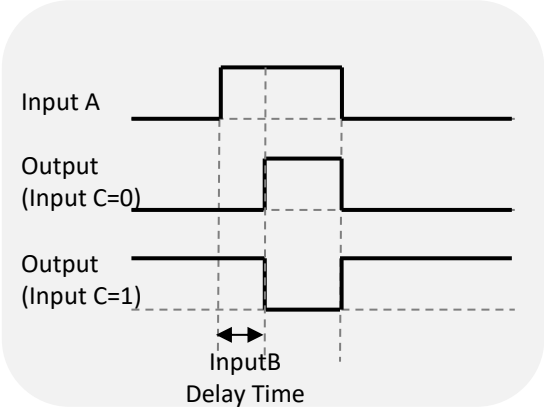
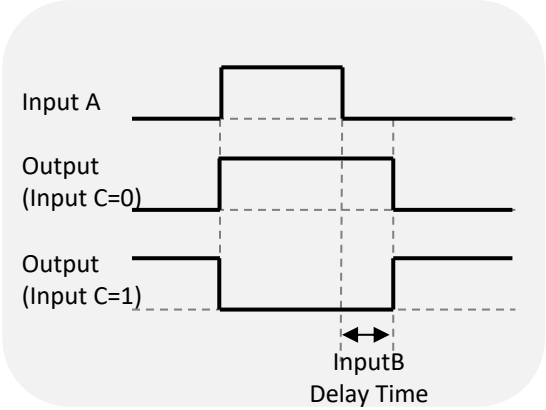
Advanced User Sequence

- Value Function Blocks (6)

<table>
<tr><th>No.</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td>18</td><td>A(1+B)</td><td colspan="2">A*(1+B/100)을 출력합니다.
Example) When A=3 and B=100, A*(1+B/100)=3*(1+100/100)=6</td></tr>
<tr><td>19</td><td>(A*B/100)+C</td><td colspan="2">(A*B)/100)+C를 출력합니다.
Example) When A=3, B=100 and C=2, (A*B)/100+C=(3*100)/100+2=5</td></tr>
<tr><td rowspan="4">20</td><td rowspan="4">ABS(A)>ABS(B)+/-C</td><td>Condition</td><td>Output</td></tr>
<tr><td>ABS(A) > ABS(B)+C</td><td>1</td></tr>
<tr><td>ABS(A) < ABS(B)-C</td><td>0</td></tr>
<tr><td>ABS(B)-C < ABS(A) < ABS(B)+C</td><td>Keep</td></tr>
<tr><td>21</td><td>ABS(A)>=ABS(B)</td><td colspan="2">절대값(A) >= 절대값(B)이면 1 아니면 0을 출력합니다.</td></tr>
<tr><td>22</td><td>IF(C)-A</td><td colspan="2">C가 0이면 A, 아니면 –A를 출력합니다.</td></tr>
<tr><td>23</td><td>IF(A==B)</td><td colspan="2">A와 B가 같으면 1 아니면 0을 출력합니다.</td></tr>
<tr><td>24</td><td>IF(A!=B)</td><td colspan="2">A와 B가 다르면 1 아니면 0을 출력합니다.</td></tr>
<tr><td>25</td><td>IF(C) HOLD(A)</td><td colspan="2">C가 1이면 이전 A값을, 0이면 현재 A값을
출력합니다.</td></tr>
</table>

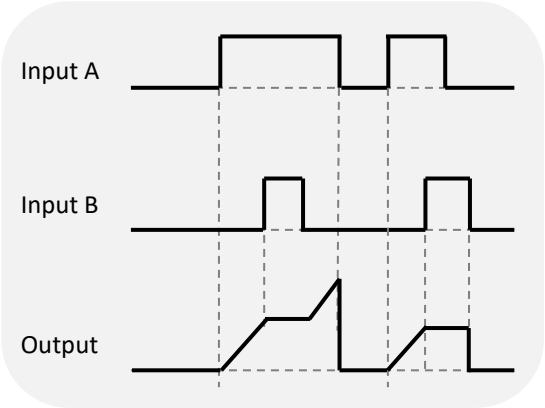
Advanced User Sequence

- Value Function Blocks (7)

No.	Type	Description
26	On Delay	A 입력에 대해 B시간만큼 온(On) 작동을 지연해서 출력합니다. 
27	Off Delay	A 입력에 대해 B시간만큼 오프(Off) 작동을 지연해서 출력합니다. 

Advanced User Sequence

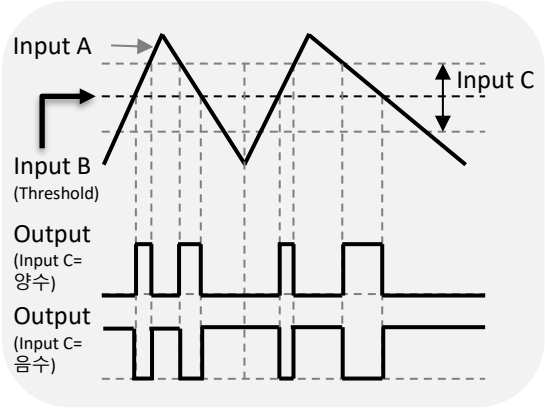
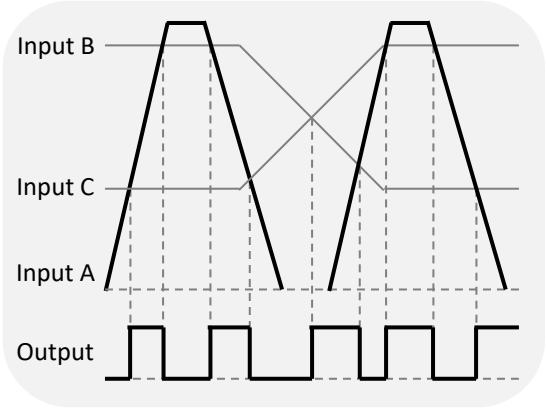
- Value Function Blocks (8)

<table>
<tr><th>No.</th><th>Type</th><th>Description</th></tr>
<tr><td rowspan="4">28</td><td rowspan="4">Timer</td><td>TIMER 함수 블록은 사용자 입력 A를 타이머 실행/정지 비트로 사용하는 규칙을 따릅니다. TIMER는 사용자 입력 A = 1일 때 시작됩니다. 사용자 입력 B는 타이머 출력 모드입니다. 사용자 입력 C는 사용되지 않습니다.
 

<table>
<tr><th>Input A</th><th>Input B</th><th>Output</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>100msec마다 1씩 증가,
99,999,999 다음에는 기존 출력값 유지</td></tr>
<tr><td>-</td><td>1</td><td>기존 출력값 유지</td></tr>
</table></td></tr>
</table>

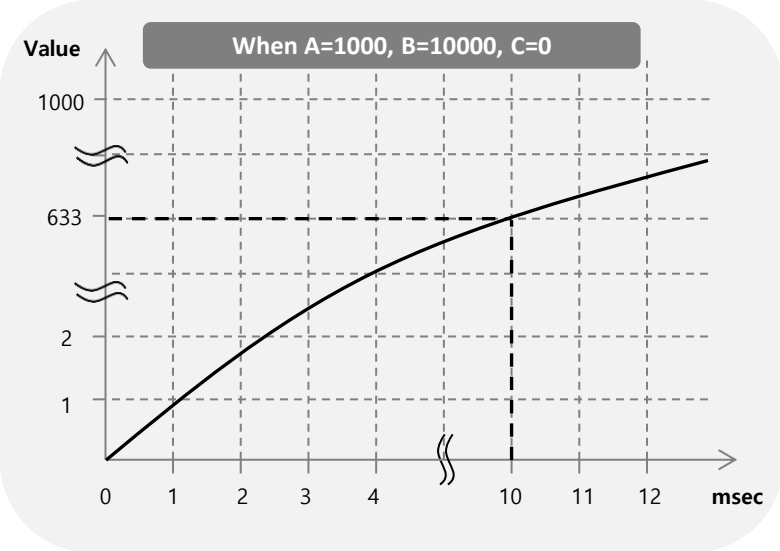
Advanced User Sequence

- Value Function Blocks (9)

No.	Type	Description
29	Window	A 입력에 대해 일정 밴드 안에 있는지 검출하는 기능입니다. 출력은 C에 의해 극성이 결정됩니다. 
30	Window C <=A <=B	C 입력에 대해 일정 밴드 안에 있는지 검출하는 기능입니다. 출력은 A와 B의 크기에 의해 극성이 결정됩니다. 

Advanced User Sequence

- Value Function Blocks (10)

No.	Type	Description
31	Lowpass Filter	LOWPASSFILTER 함수 블록은 A에 들어오는 입력값을, B를 기준으로 출력합니다. 여기서 B는 출력이 A의 63.3%에 도달하는 데 걸리는 시간이며, 단위는 msec 입니다. 사용자 입력 C가 0이면 동작을 시작하고, C=1이면 동작을 멈춥니다. 
32	Limit A upr B Lwr C	LIMIT 함수 블록은 다음 규칙을 따릅니다. (사용자 입력) C ≥ A ≥ B이면 출력은 사용자 입력 A 값입니다. 그렇지 않고 A > C이면 출력은 사용자 입력 C이고, A < B이면 출력은 사용자 입력 B입니다.
33	Extra Bits	비트 위치 B부터 시작하여 길이 C비트만큼 값 A에서 출력 비트를 추출합니다. 비트 번호는 0부터 시작합니다.

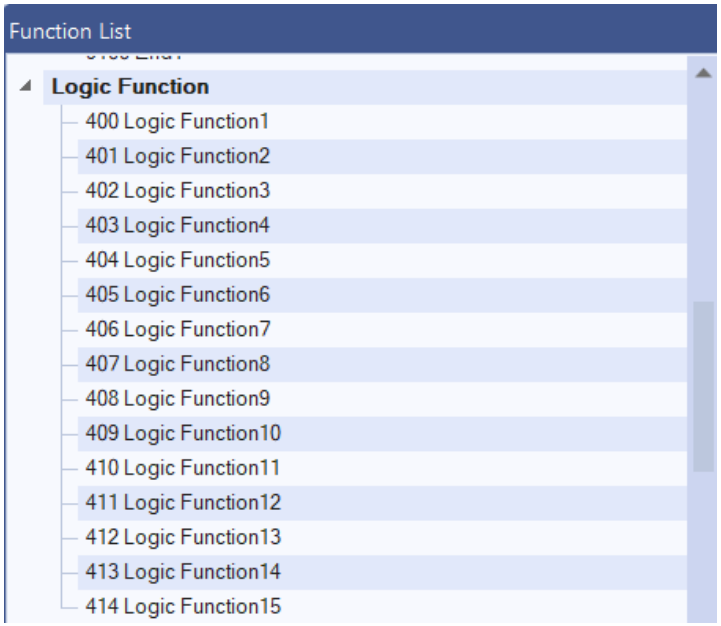
Advanced User Sequence

- Logic Function Blocks (1)
  - Logic 함수 블록은 2~3개의 입력 값과 1개의 출력 값으로 구성됩니다.

Group	Name	Description	Range
USL	Logic@※ Type	Choose the function to perform in the function block.	0~10
	Logic@ Input A	A first input parameter.	0~1
	Logic@ Input B	A second input parameter.	0~1
	Logic@ Input C	A third input parameter.	0~1
	Logic@ Ouptut	Output value.	Read Only

※ @is the step number(1~15).

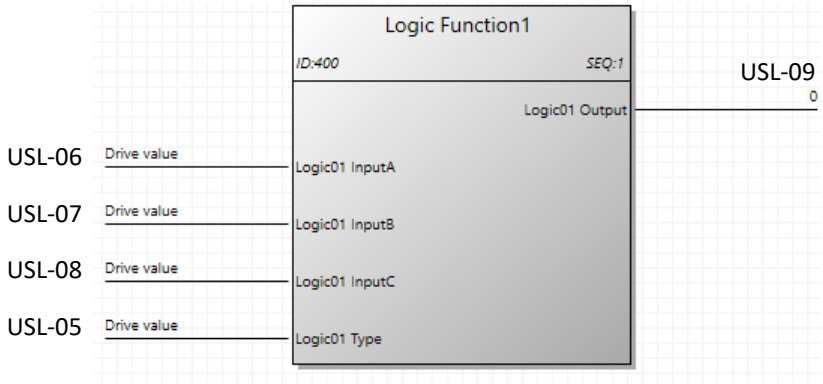
  - 최대 15개의 로직 함수 블록을 사용할 수 있으며, 11가지 서로 다른 기능을 제공합니다.

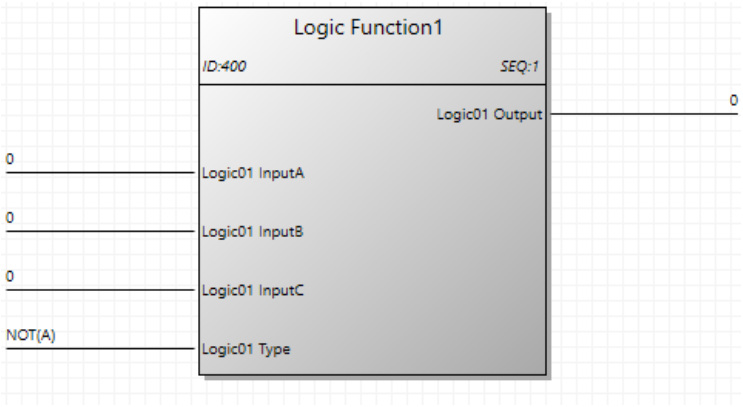


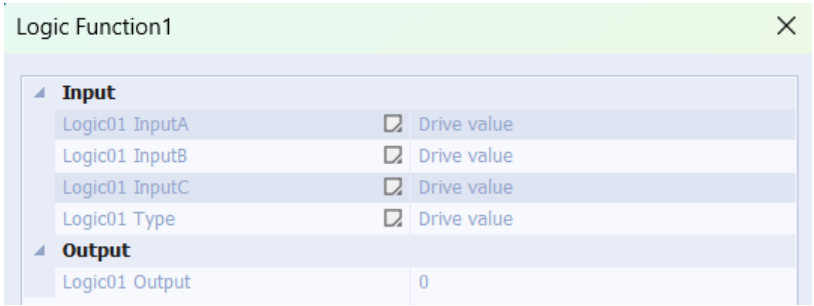
Advanced User Sequence

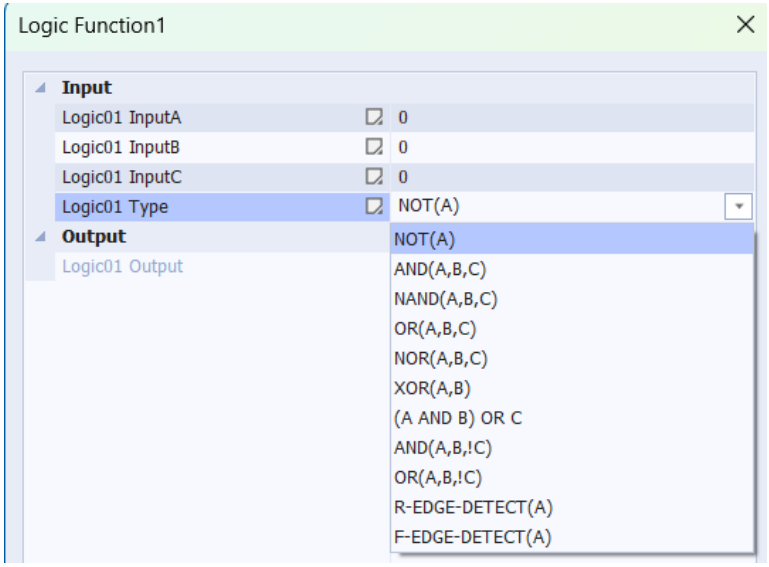
- Logic Function Blocks (2)

Input Type = Drive Value	Input Type = Parameter Value









Advanced User Sequence

- Logic Function Blocks (3)

No.	Type	Description			
0	NOT(A)	Input A		Output	
		0		1	
		1		0	
1	AND(A,B,C)	Input A	Input B	Input C	Output
		The others			0
		1	1	1	1
2	NAND(A,B,C)	Input A	Input B	Input C	Output
		The others			1
		1	1	1	0
3	OR(A,B,C)	Input A	Input B	Input C	Output
		0	0	0	0
		The others			1
4	NOR(A,B,C)	Input A	Input B	Input C	Output
		0	0	0	1
		The others			0

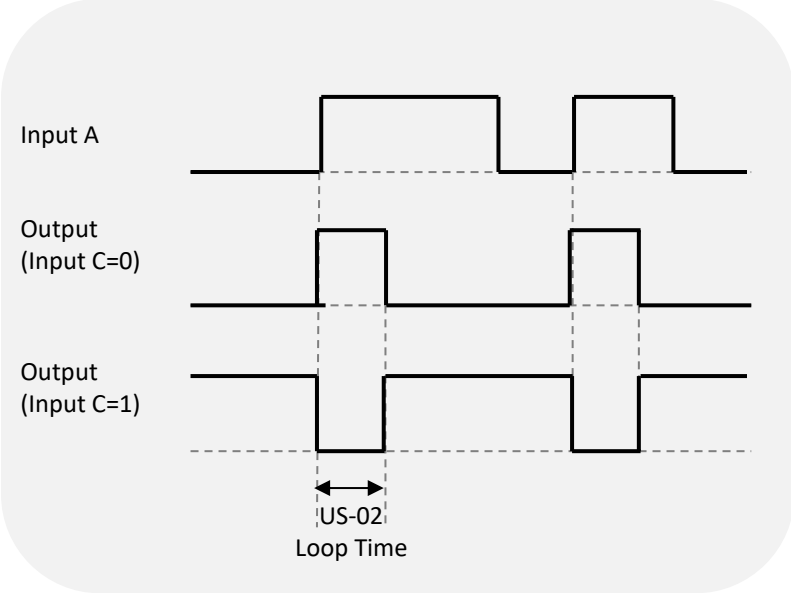
Advanced User Sequence

- Logic Function Blocks (4)

No.	Type	Description			
5	XOR(A,B)	Input A	Input B	Input C	Output
		0	0	0	0
		The others			1
		1	1	1	0
6	(A AND B) OR C	Input A	Input B	Input C	Output
		0	0	0	0
		0	0	1	1
		0	1	0	0
		0	1	1	1
		1	0	0	0
		1	0	1	1
		1	1	0	1
		1	1	1	1

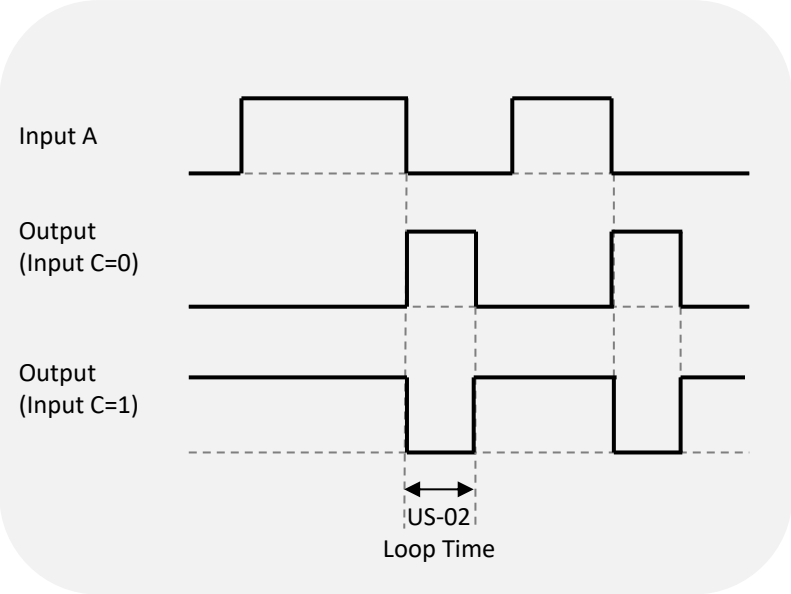
Advanced User Sequence

- Logic Function Blocks (5)

<table>
<tr><th>No.</th><th>Type</th><th colspan="4">Description</th></tr>
<tr><td rowspan="3">7</td><td rowspan="3">AND(A,B,!C)</td><td>Input A</td><td>Input B</td><td>Input C</td><td>Output</td></tr>
<tr><td colspan="3">The others</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">8</td><td rowspan="3">OR(A,B,!C)</td><td>Input A</td><td>Input B</td><td>Input C</td><td>Output</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">The others</td><td>1</td></tr>
<tr><td>9</td><td>R-EDGE-DETECT(A)</td><td colspan="4"> 
</td></tr>
</table>

Advanced User Sequence

- Logic Function Blocks (6)

No.	Type	Description
10	F-EDGE-DETECT(A)	

Advanced User Sequence

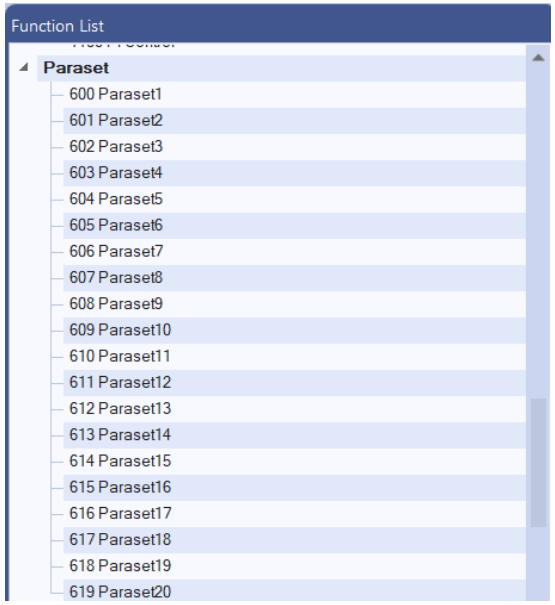
- Paraset Blocks (1)

  - Paraset 기능은 특정 조건이 충족되면 일부 파라미터를 지정된 값으로 설정합니다.
  - Paraset 기능은 입력의 상승 edge 또는 하강 edge를 감지하며, dege가 감지되면 일부 값을 일부 파라미터에 설정합니다.

Group	Name	Description	Range
USP	Paraset@※In	Paraset function input	0~1
	Paraset@ Type	파라미터 동작을 수행할 이벤트 유형을 상승 edge 또는 하강 edge 중 하나로 설정합니다.	0: 0-1 EDGE
			1: 1-0 EDGE

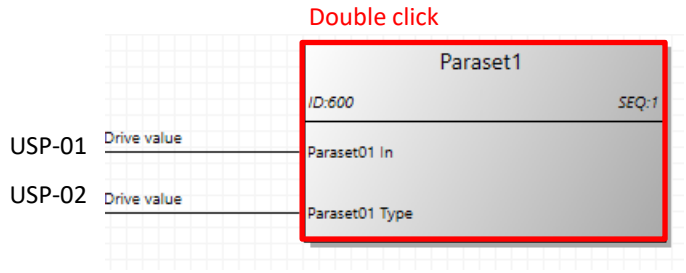
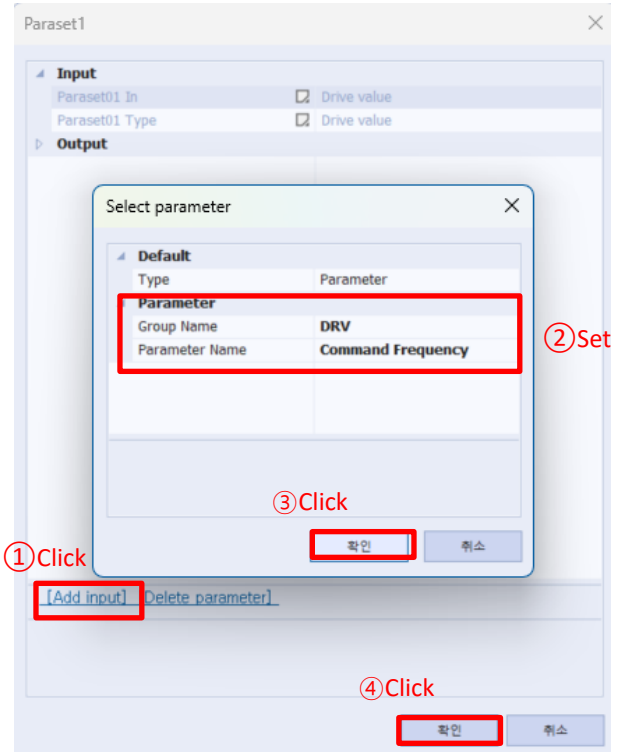
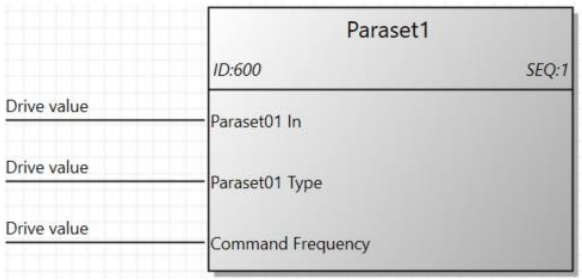
※ @is the step number(1~20).

- 최대 20개까지의 paraset 블록 함수를 사용할 수 있습니다.



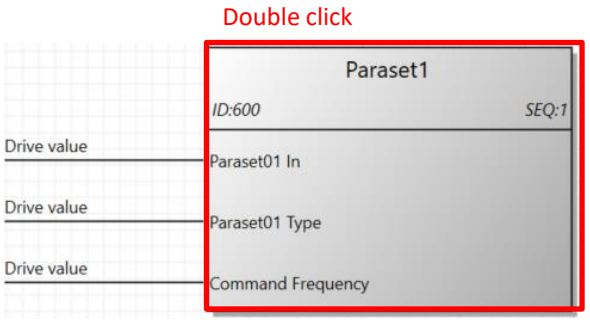
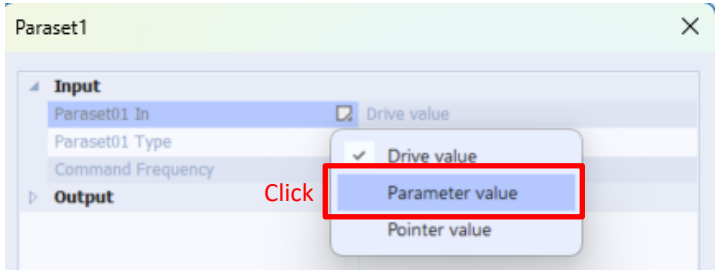
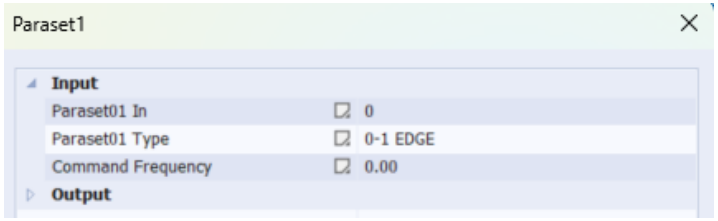
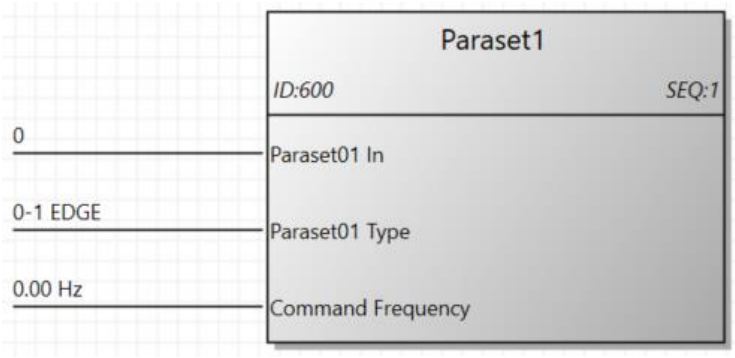
Advanced User Sequence

- Paraset Blocks (2)
  - How to add some drive value inputs

Step 1	Step 2
	
Step 3	
이 예시 에서는 Paraset01 In의 상승 edge가 감지되면, DRV-01 에 설정된 주파수가 지령 주파수로 입력됩니다. 	

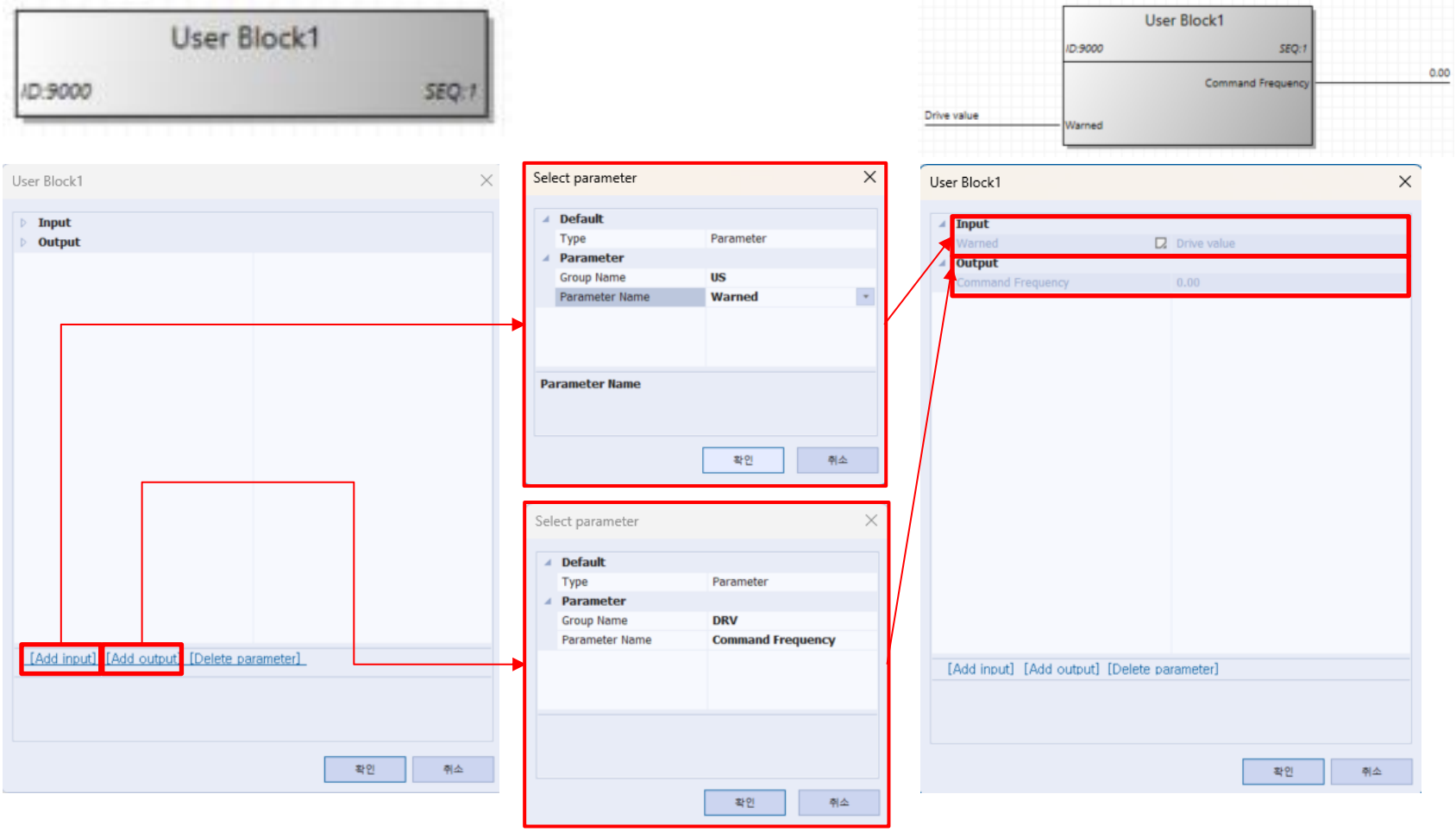
Advanced User Sequence

- Paraset Blocks (3)
  - How to add some parameter value inputs

Step 1	Step 2
	Input type을 Parameter Value로 변경. 
Step 3	**Step 4**
	이 예시 에서는 Paraset01 In의 상승 edge가 감지되면 0.00 Hz가 지령 주파수에 입력됩니다. 

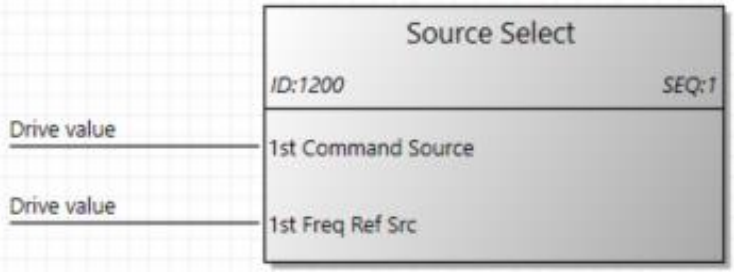
Advanced User Sequence

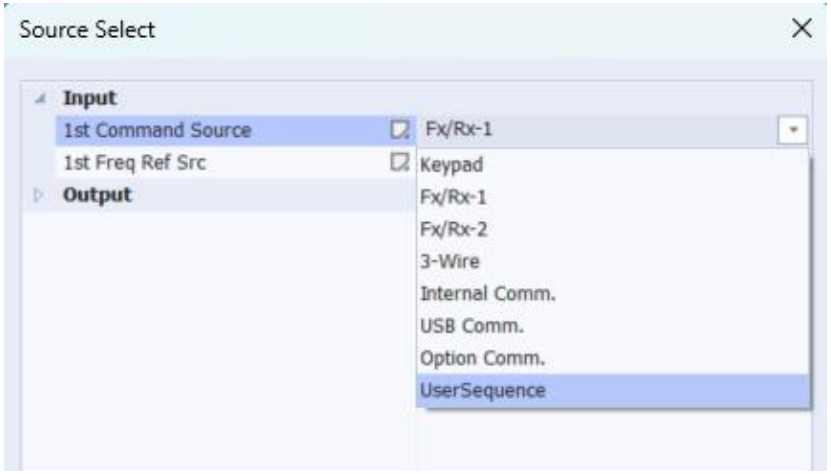
- User Blocks
  - 입력(Inputs): 사용자 시퀀스가 시작되면, 일부 파라미터를 원하는 값으로 설정할 수 있습니다.
  - 출력(Outputs): 사용자 블록의 출력은 다른 블록의 입력으로 사용할 수 있습니다.

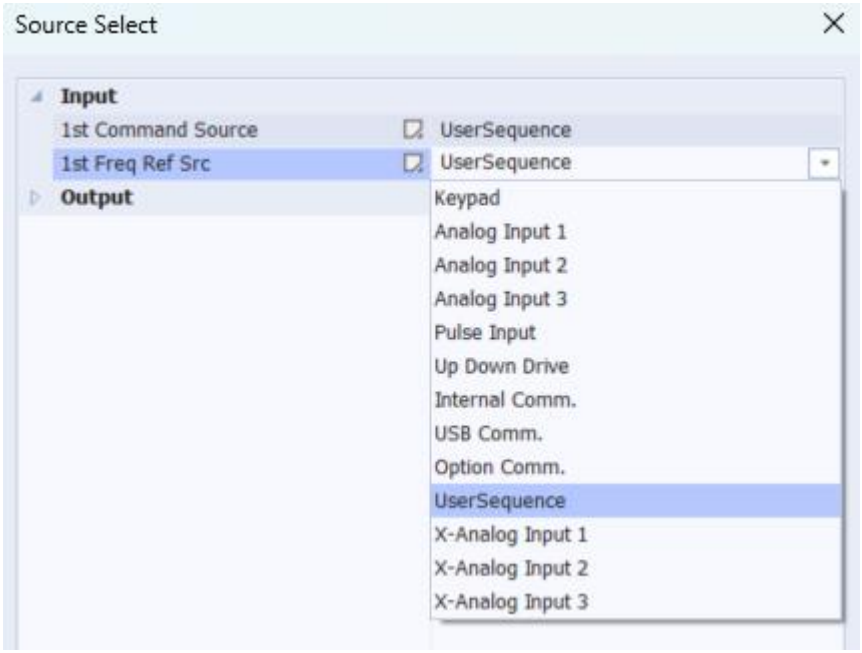


Advanced User Sequence

- Source Select
  - 이 블록은 1st command source 또는 1st Frequency reference source를 설정하는 데 사용됩니다.



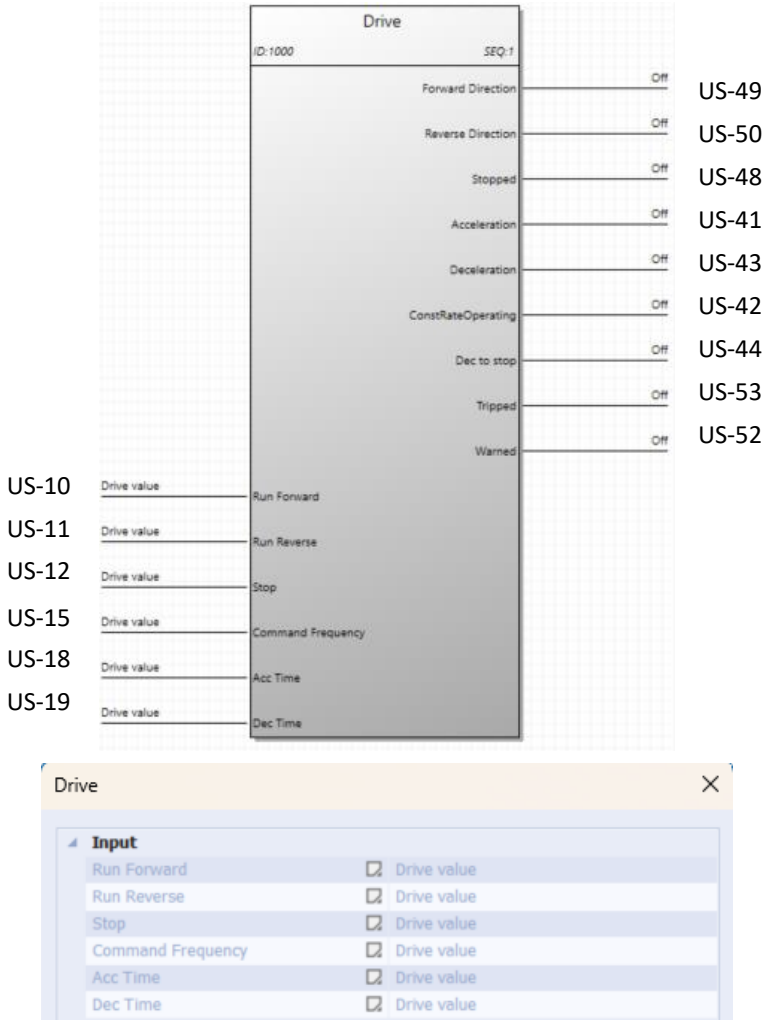


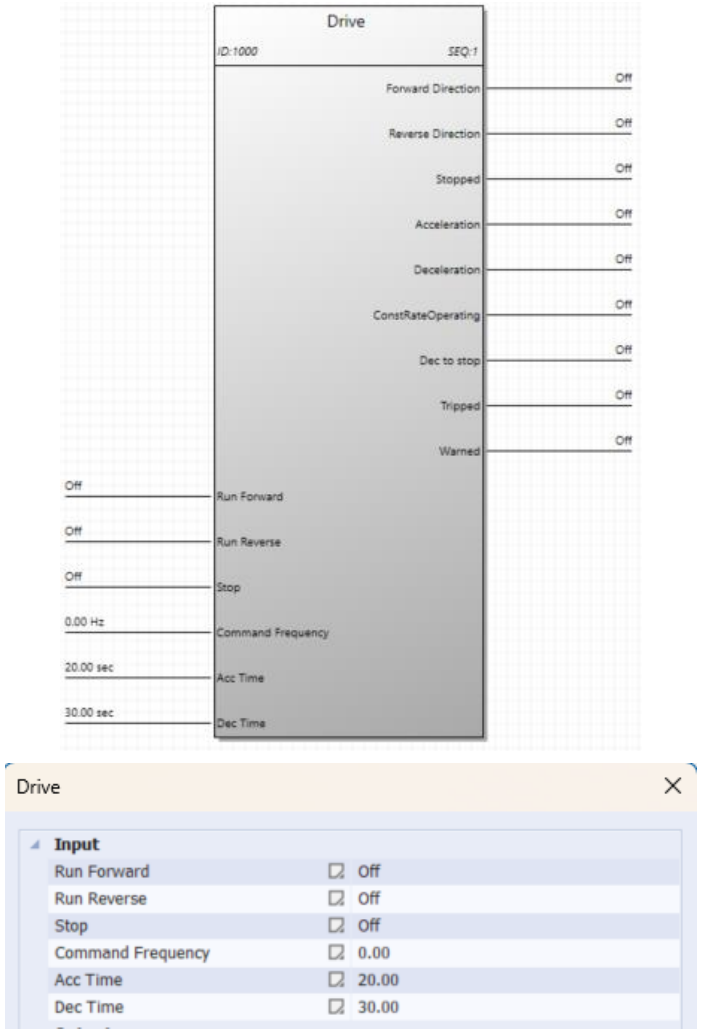


Advanced User Sequence

- Drive Block (1)

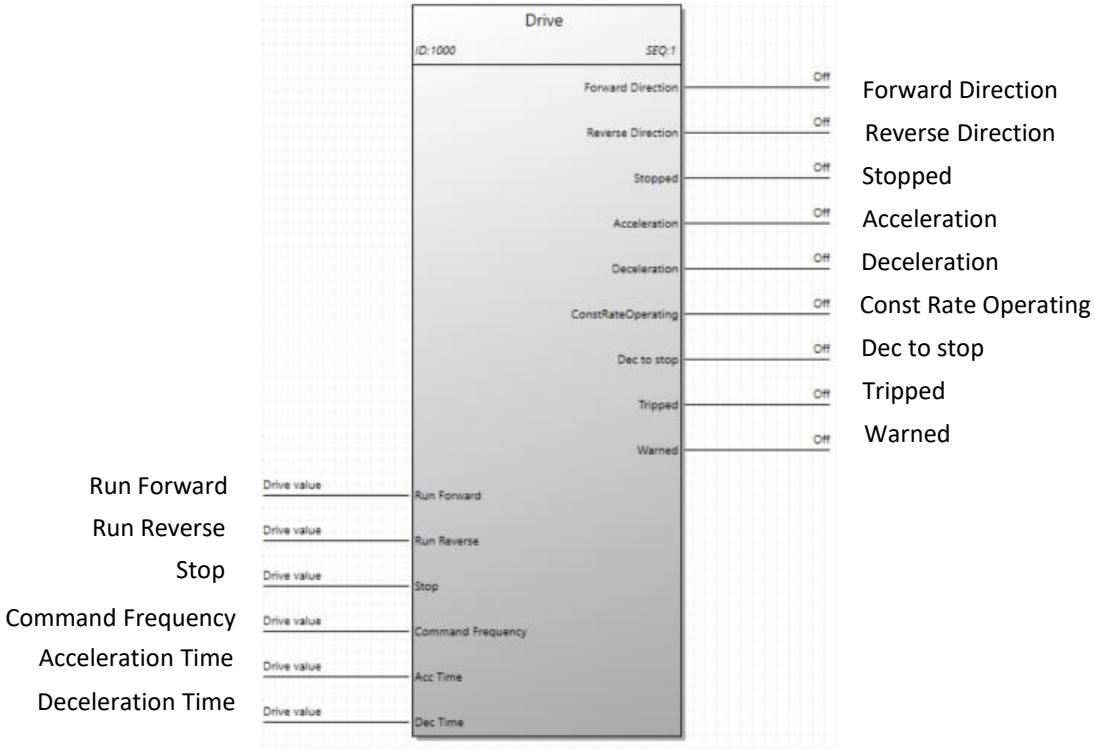
Input Type = Drive Value	Input Type = Parameter Value





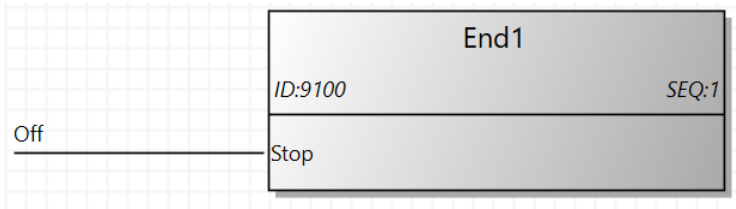
Advanced User Sequence

- Drive Block (2)
  - Drive 블록은 자주 사용하는 명령과 상태로 구성됩니다.
  - 입력(Inputs): Drive 블록의 입력을 통해 드라이브로 일부 동작(운전) 명령을 전달할 수 있습니다.
  - 출력(Outputs): Drive 블록의 출력을 통해 드라이브의 일부 상태를 읽을 수 있습니다.



Advanced User Sequence

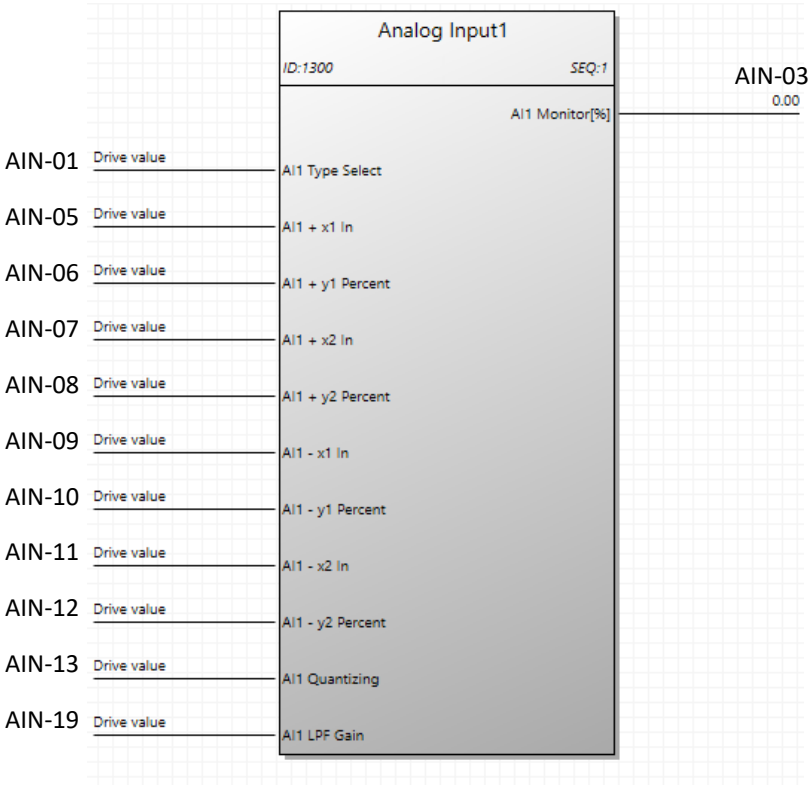
- End Block
  - End 블록은 사용자 시퀀스가 Off될 때 실행됩니다.
  - End 블록에는 기본적으로 드라이브 정지(Stop) 명령이 있으며, 기본 상태는 OFF입니다.
  - 정지 명령이 ON이고 사용자 시퀀스가 OFF로 전환되면, 사용자 시퀀스 종료와 함께 드라이브가 정지합니다.

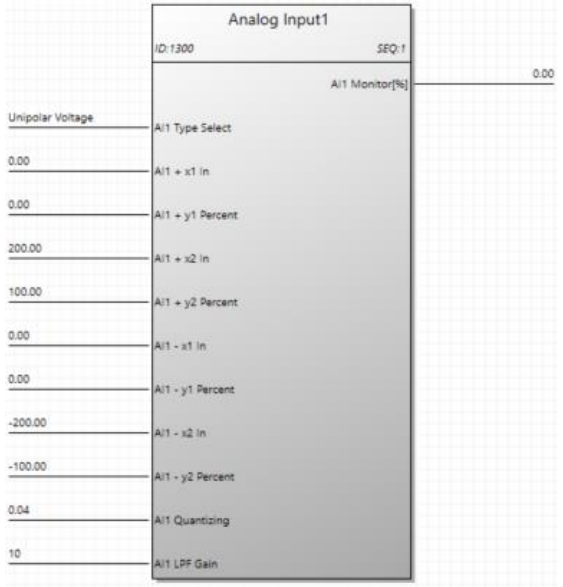


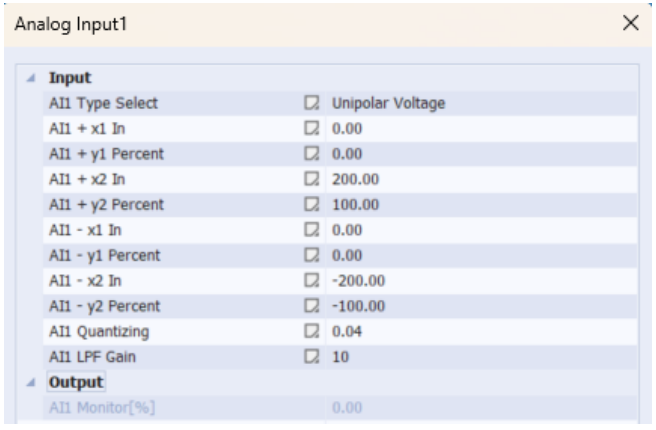
Advanced User Sequence

- Analog Input Blocks (1)

Input Type = Drive Value	Input Type = Parameter Value

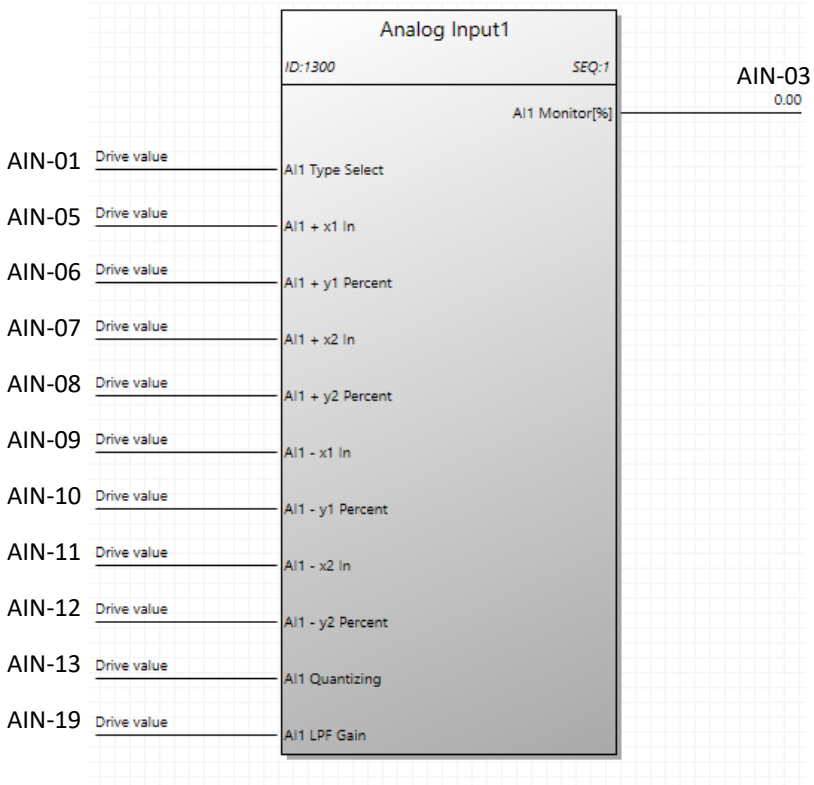






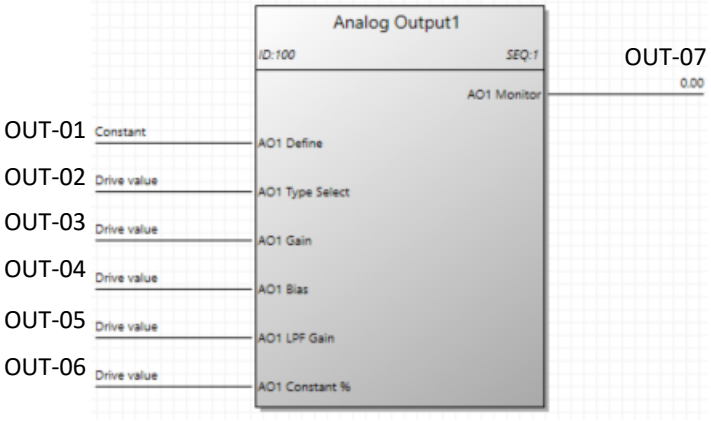
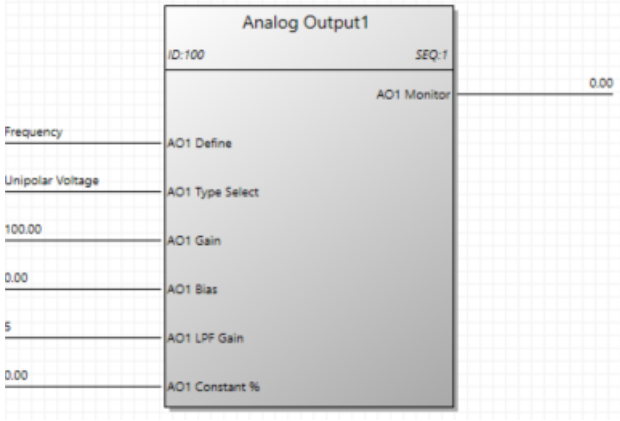
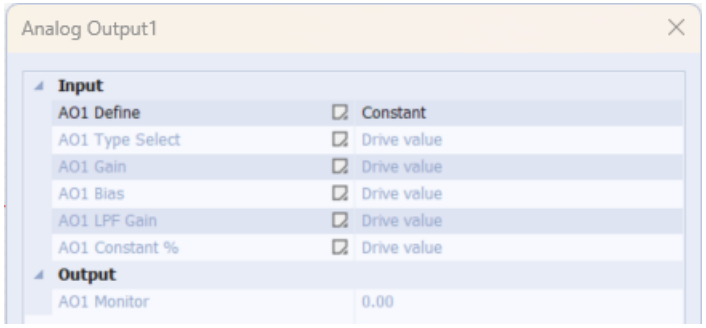
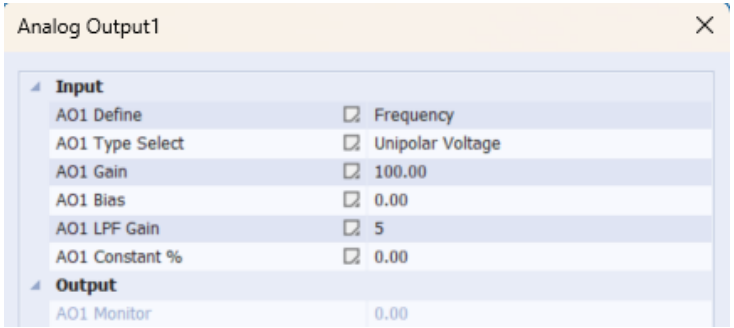
Advanced User Sequence

- Analog Input Blocks (2)
  - Analog input 블록은 아날로그 입력과 관련된 파라미터로 구성됩니다.
  - 이 블록은 아날로그 입력의 설정을 구성(설정)하는 데 사용할 수 있습니다.



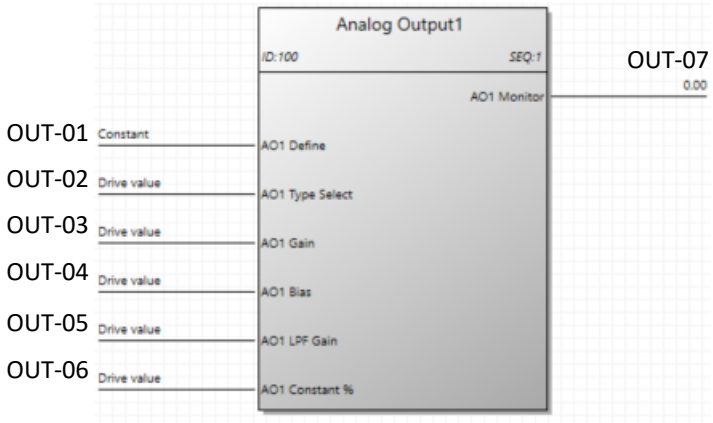
Advanced User Sequence

- Analog Output Blocks (1)

Input Type = Drive Value	Input Type = Parameter Value
	
	

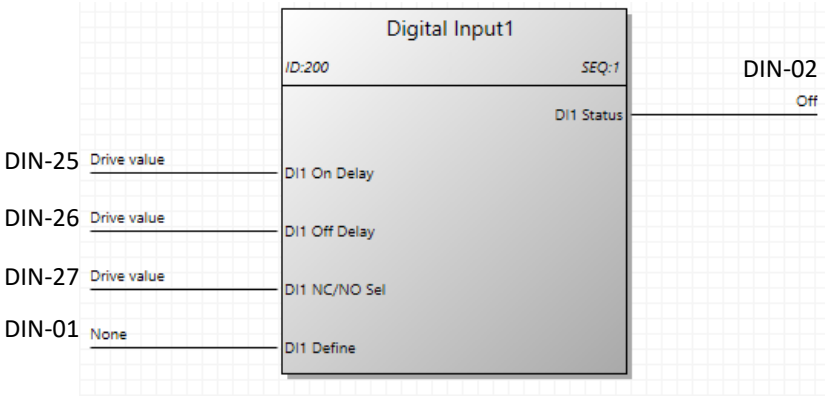
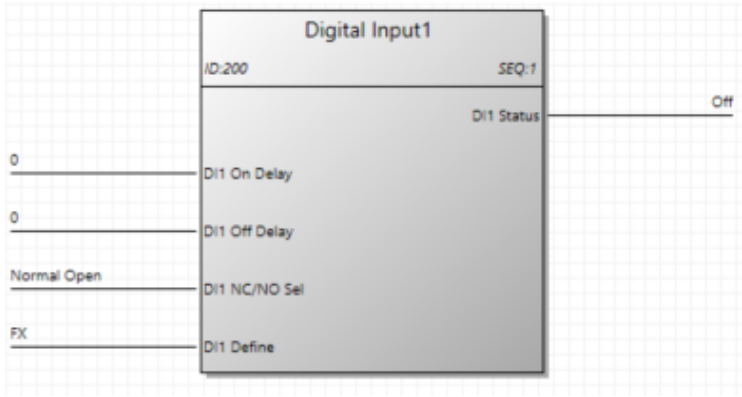
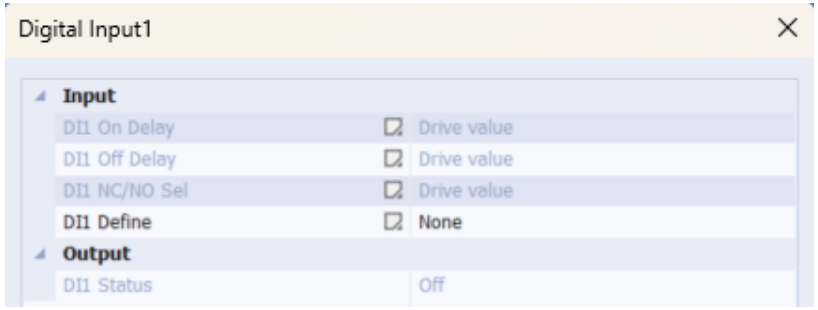
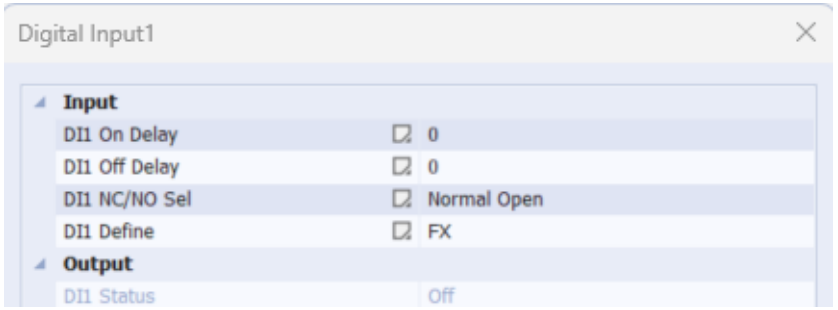
Advanced User Sequence

- Analog Output Blocks (2)
  - Analog output 블록은 아날로그 출력과 관련된 파라미터로 구성됩니다.
  - 이 블록은 아날로그 출력의 설정을 구성(설정)하는 데 사용할 수 있습니다.



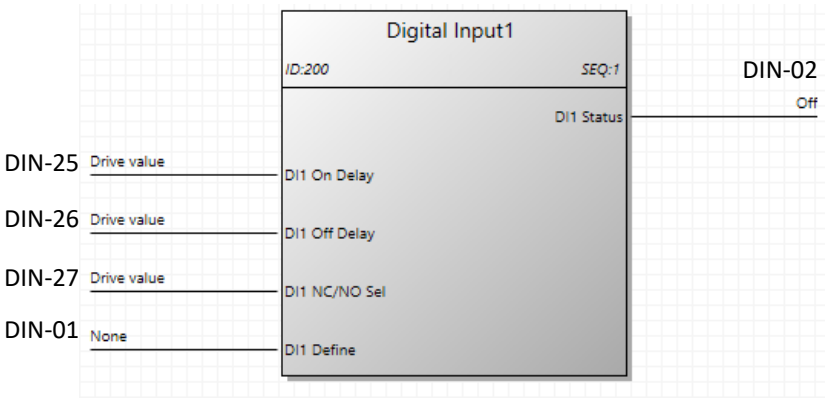
Advanced User Sequence

- Digital Input Blocks (1)

Input Type = Drive Value	Input Type = Parameter Value
	
	

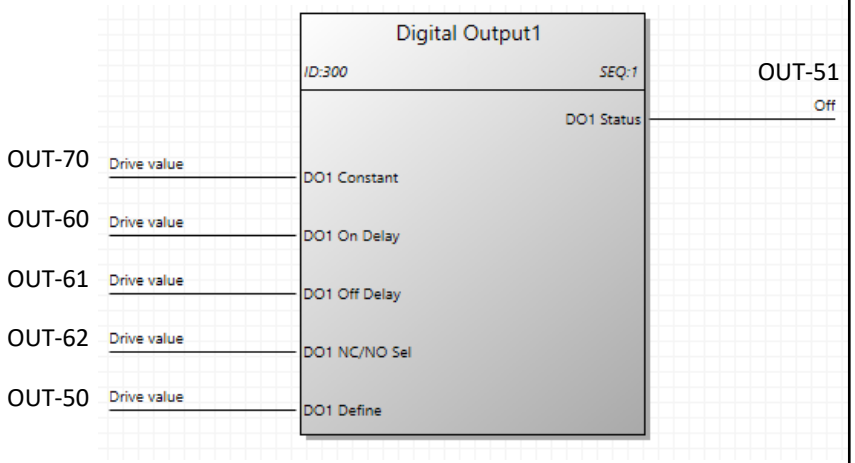
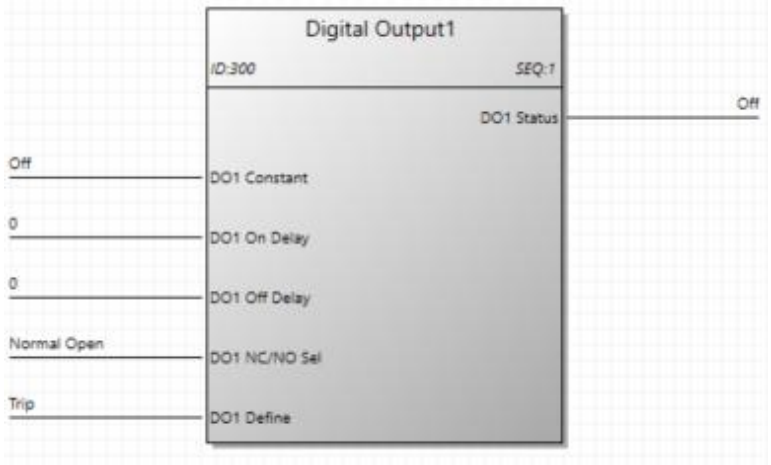
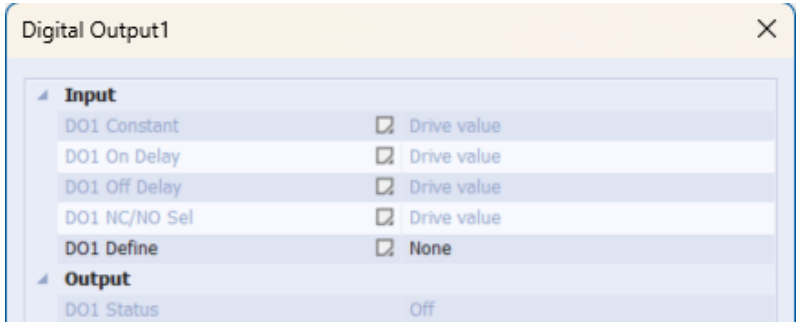
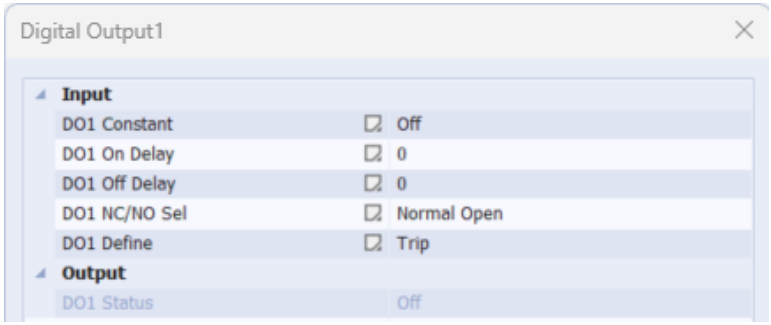
Advanced User Sequence

- Digital Input Blocks (2)
  - Digital input 블록은 디지털 입력과 관련된 파라미터로 구성됩니다.
  - 이 블록은 디지털 입력의 설정을 구성(설정)하는 데 사용할 수 있습니다.



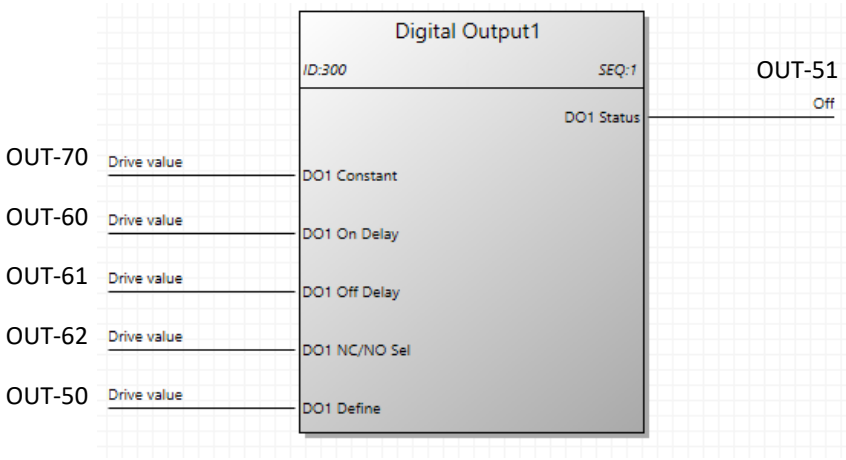
Advanced User Sequence

- Digital Output Blocks (1)

Input Type = Drive Value	Input Type = Parameter Value
	
	

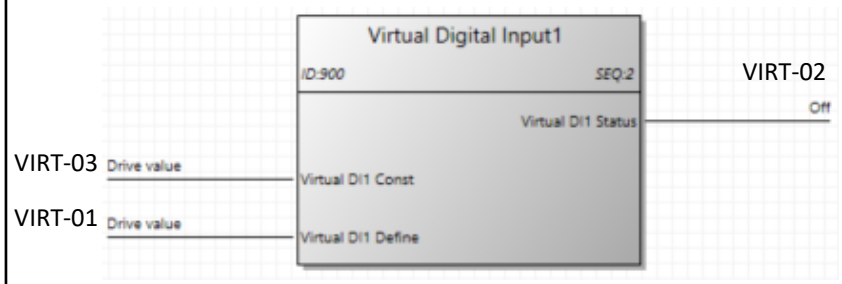
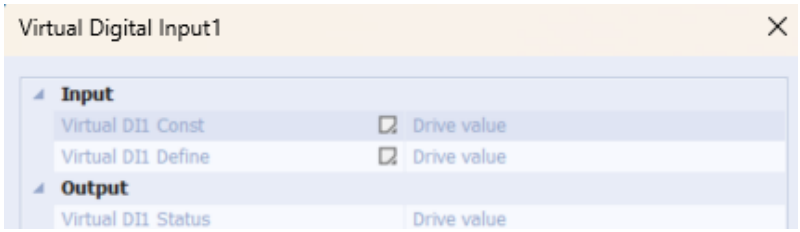
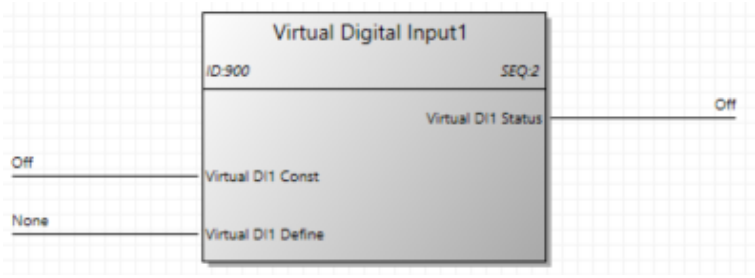
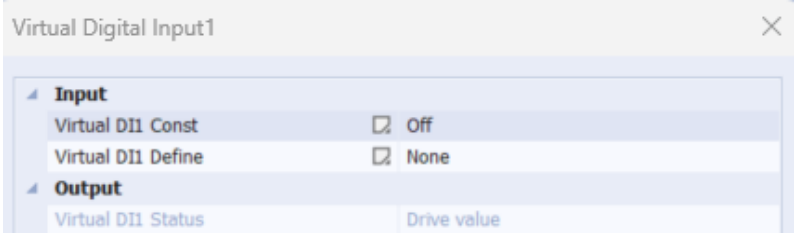
Advanced User Sequence

- Digital Output Blocks (2)
  - Digital output 블록은 디지털 출력과 관련된 파라미터로 구성됩니다.
  - 이 블록은 디지털 출력의 설정을 구성(설정)하는 데 사용할 수 있습니다.



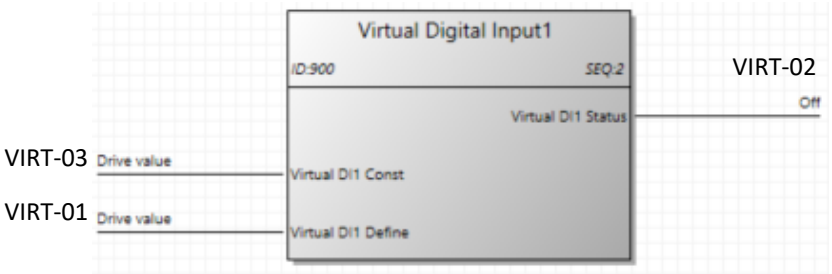
Advanced User Sequence

- Virtual Digital Input Blocks (1)

Input Type = Drive Value	Input Type = Parameter Value
 	 

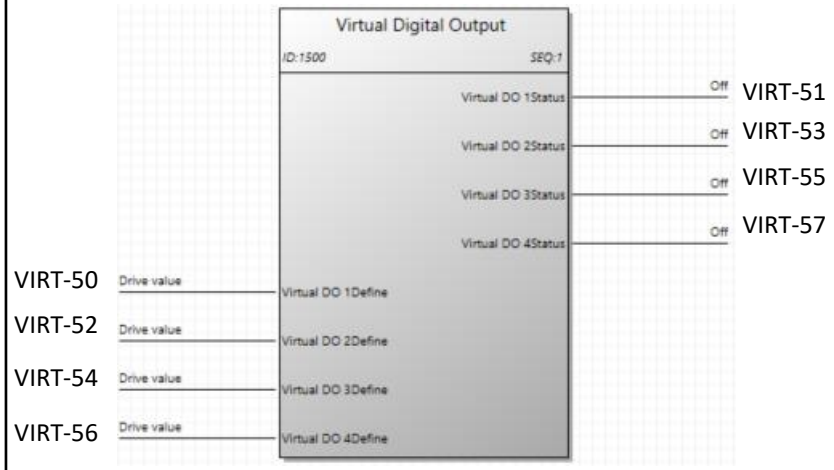
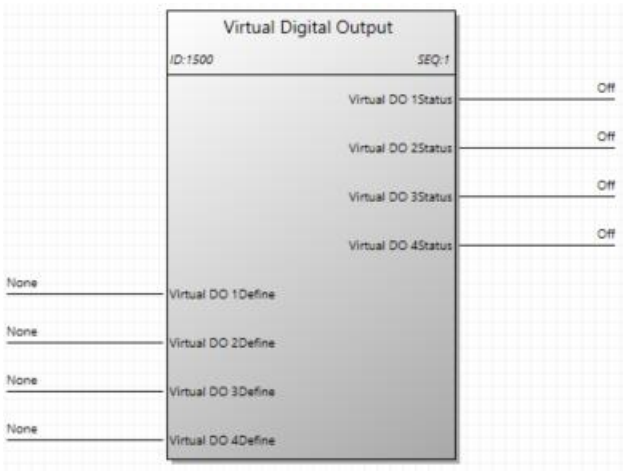
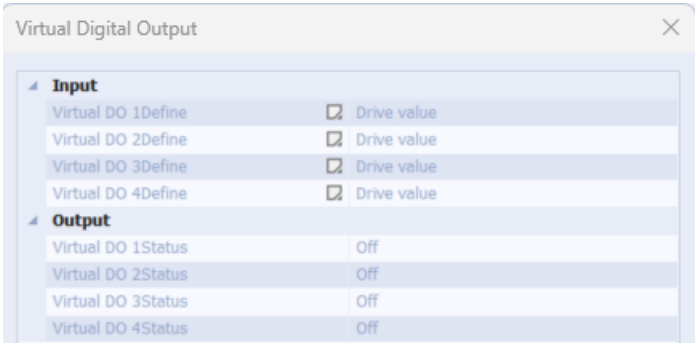
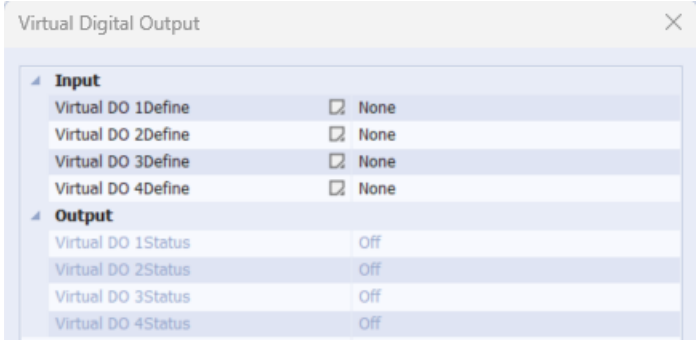
Advanced User Sequence

- Virtual Digital Input Blocks (2)
  - Virtual Digital input 블록은 가상 디지털 입력과 관련된 파라미터로 구성됩니다.
  - 이 블록은 가상 디지털 입력의 설정을 구성(설정)하는 데 사용할 수 있습니다.



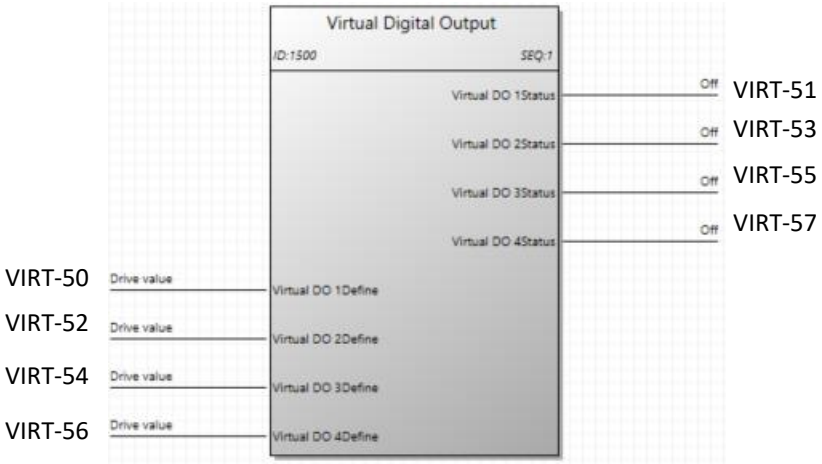
Advanced User Sequence

- Virtual Digital Output Blocks (1)

Input Type = Drive Value	Input Type = Parameter Value
	
	

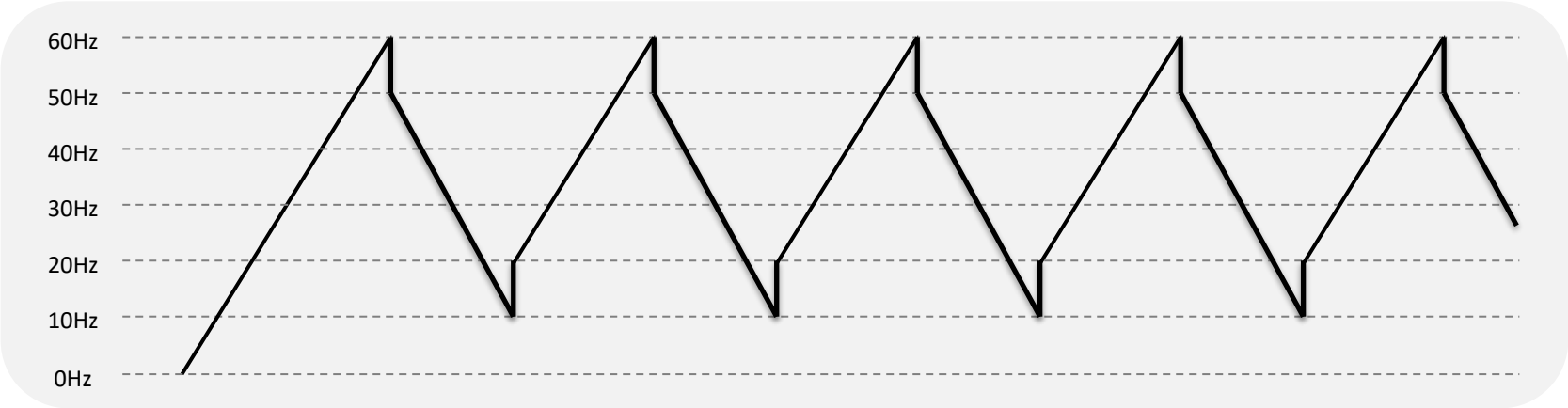
Advanced User Sequence

- Virtual Digital Output Blocks (2)
  - Virtual Digital output 블록은 가상 디지털 출력과 관련된 파라미터로 구성됩니다.
  - 이 블록은 가상 디지털 출력의 설정을 구성(설정)하는 데 사용할 수 있습니다.



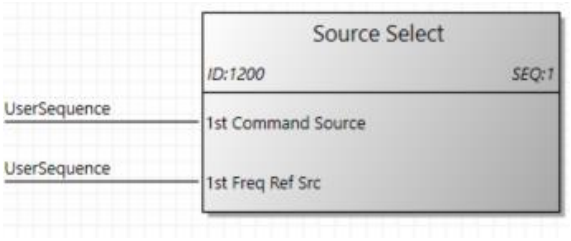
Advanced User Sequence

- Example: Advanced User Sequence를 사용한 Traverse 동작 구현.



- Requirements
  - ✓ 주파수 제한(하한): 10Hz
  - ✓ 주파수 제한(상한): 60Hz
  - ✓ 출력 주파수가 제한 주파수에 도달하면 10Hz만큼 점프한 다음, 반대쪽 제한 주파수를 향해 가속 또는 감속합니다.

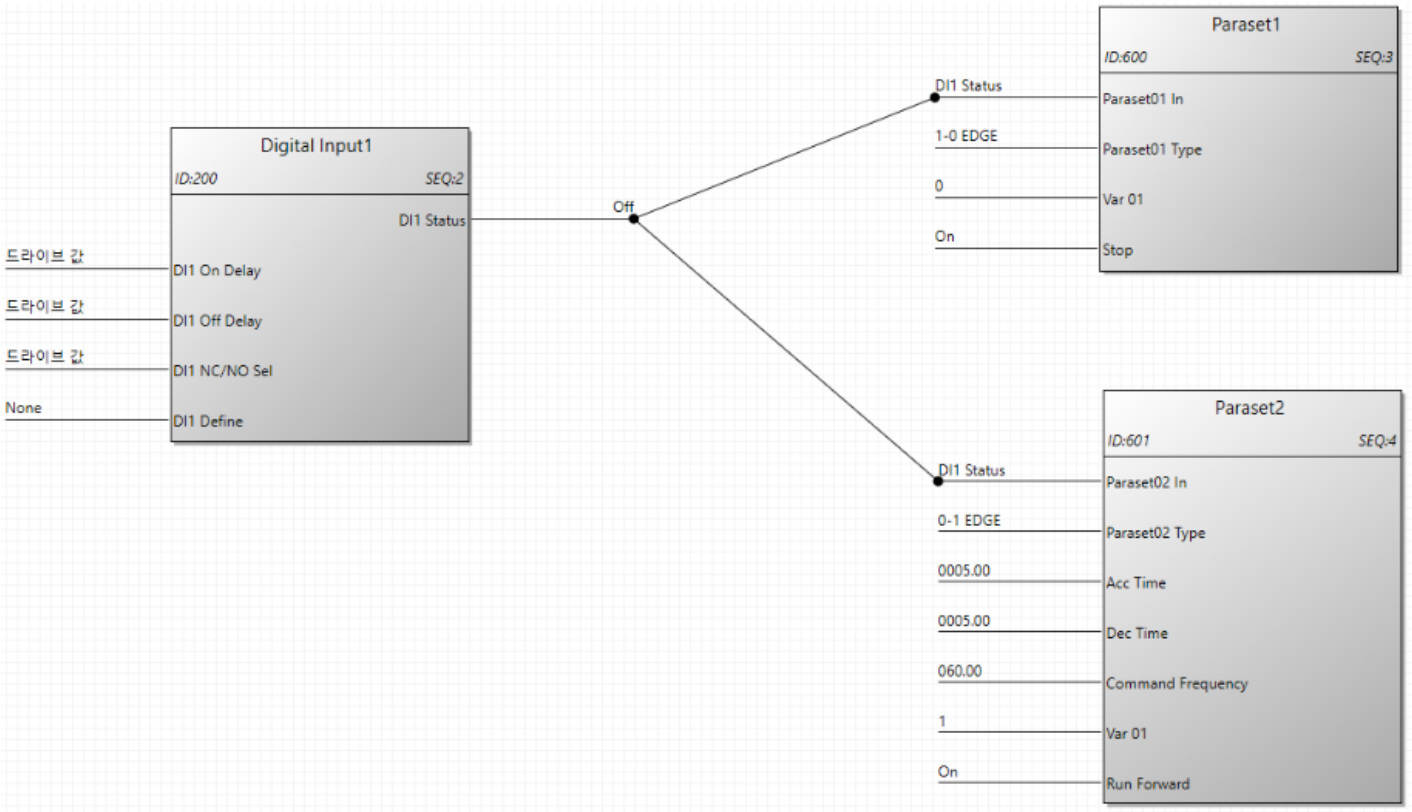
- 1. Source Select Block
  - User Sequence가 드라이브를 제어할 수 있도록, command source와 frequency reference source를 모두 'User sequence'로 설정하십시오.



Advanced User Sequence

- 2. Digital Input 1
  - 이 블록은 Digital Input 1의 상태를 모니터링하는 데 사용됩니다.
  - 디지털 입력 1이 ON 되면 상승 에지(rising edge)에서 Paraset 2 블록이 실행됩니다.
  - 반대로 디지털 입력 1이 OFF 되면 하강 에지(falling edge)에서 Paraset 1 블록이 실행됩니다.

- 3. Paraset 1,2
  - Paraset 1: 모터를 정지시키고, USP-61(Var 01)의 상태 값을 0으로 변경합니다.
  - Paraset 2: 명령 주파수를 60Hz로 설정하고, USP-61(Var 01)의 상태 값을 1로 변경합니다.

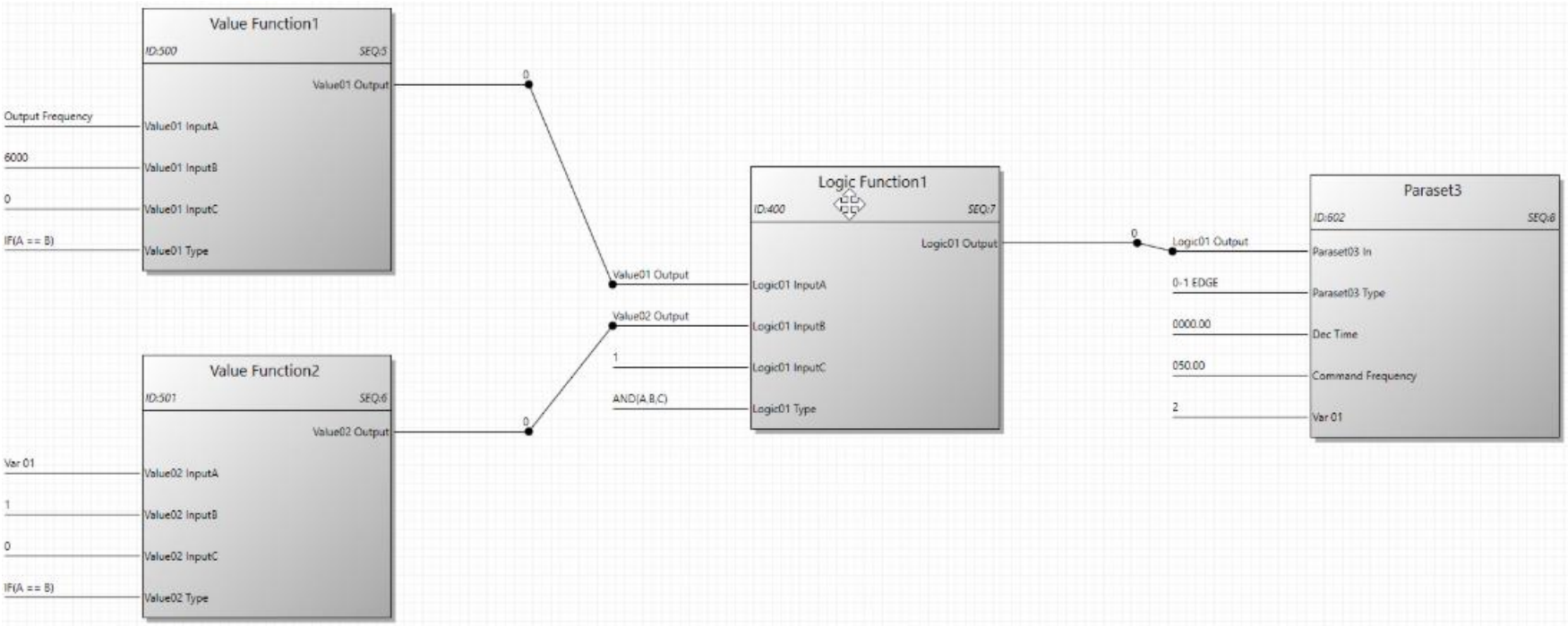


Advanced User Sequence

- 4. Value Function 1,2
  - Value Function1: 출력 주파수 = 60Hz일 때 1을 출력합니다.
  - Value Function2: Var 01 = 1일 때(디지털 입력 1이 ON일 때) 1을 출력합니다.

- 5. Logic Function 1
  - 이 블록은 AND(논리곱) 기능으로 동작합니다.
  - 즉, 출력 주파수 = 60Hz이고 Var 01 = 1이면 1을 출력합니다.

- 6. Paraset 3
  - AND 함수 결과가 1이면, 감속 시간 0초로 명령 주파수를 50Hz로 설정하고 Var 01을 2로 변경합니다.

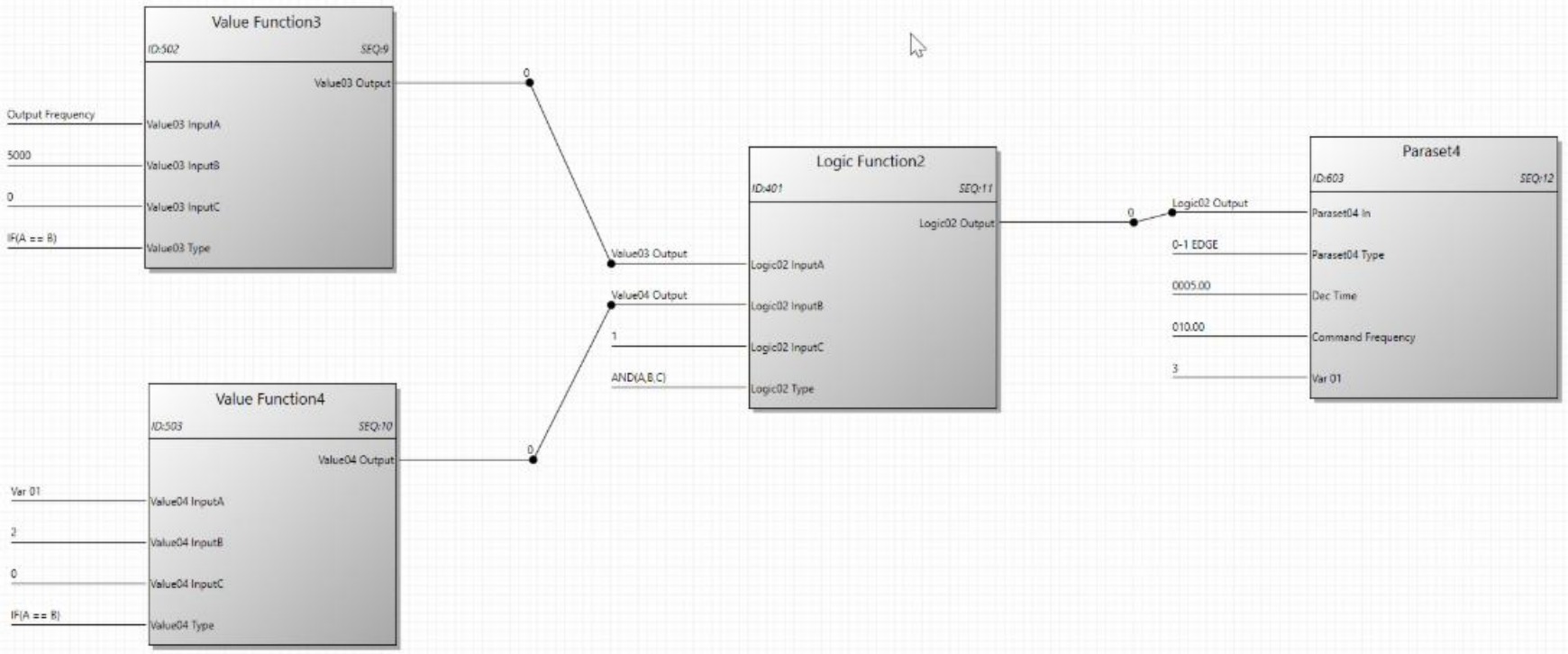


Advanced User Sequence

- 7. Value Function 3,4
  - Value Function 3: 출력 주파수 = 50Hz일 때 1을 출력합니다.
  - Value Function 4: Var 01 = 2일 때 1을 출력합니다.

- 8. Logic Function 2
  - 이 블록은 AND(논리곱) 기능으로 동작합니다.
  - 즉, 출력 주파수 = 50Hz이고 Var 01 = 2이면 1을 출력합니다.

- 9. Paraset 4
  - AND 함수 결과가 1이면, 감속 시간 5초로 명령 주파수를 10Hz로 설정하고 Var 01을 3으로 변경합니다.

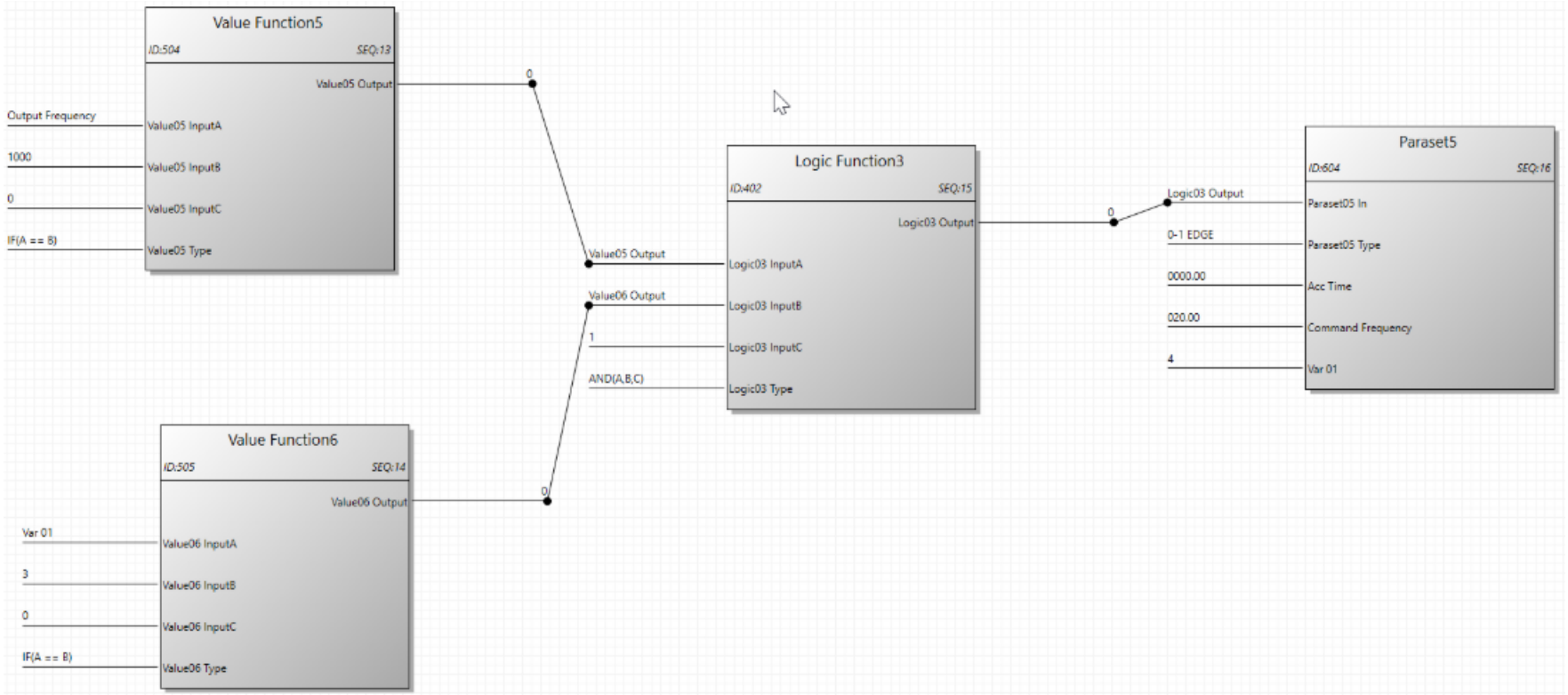


Advanced User Sequence

- 10. Value Function 5,6
  - Value Function 5: 출력 주파수 = 10Hz일 때 1을 출력합니다.
  - Value Function 6: Var 01 = 3일 때 1을 반환합니다.

- 11. Logic Function 3
  - 이 블록은 AND(논리곱) 기능으로 동작합니다.
  - 즉, 출력 주파수 = 10Hz이고 Var 01 = 3이면 1을 출력합니다.

- 12. Paraset 5
  - AND 함수 결과가 1이면, 감속 시간 0초로 명령 주파수를 20Hz로 설정하고 Var 01을 4로 변경합니다.

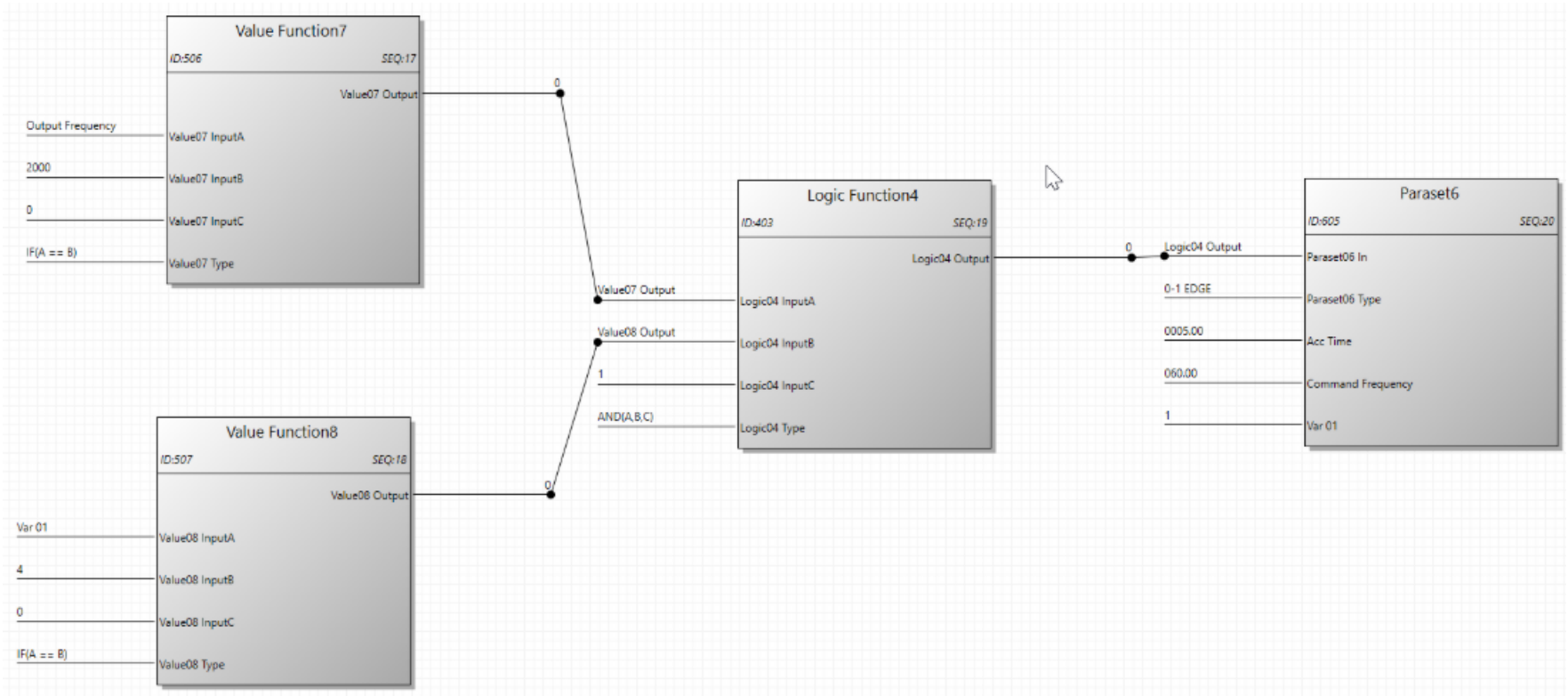


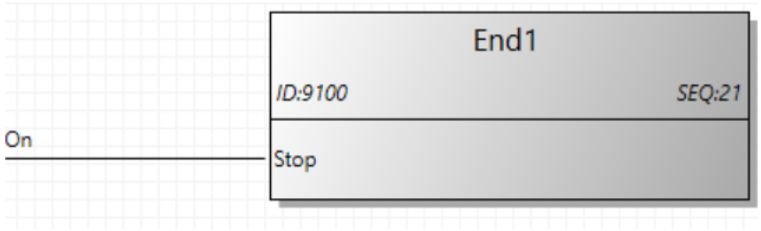
Advanced User Sequence

- 13. Value Function 7,8
  - Value Function 7: 출력 주파수 = 20Hz일 때 1을 출력합니다.
  - Value Function 8: Var 01 = 4일 때 1을 출력합니다.

- 14. Logic Function block 4
  - 이 블록은 AND(논리곱) 기능으로 동작합니다.
  - 즉, 출력 주파수 = 20Hz이고 Var 01 = 4이면 1을 출력합니다.

- 15. Paraset 6
  - AND 함수 결과가 1이면, 감속 시간 5초로 명령 주파수를 60Hz로 설정하고 Var 01을 1로 변경합니다.



Advanced User Sequence

- 16. End1
  - User Sequence가 OFF되면, END 기능이 드라이브와 모터를 정지시킵니다.