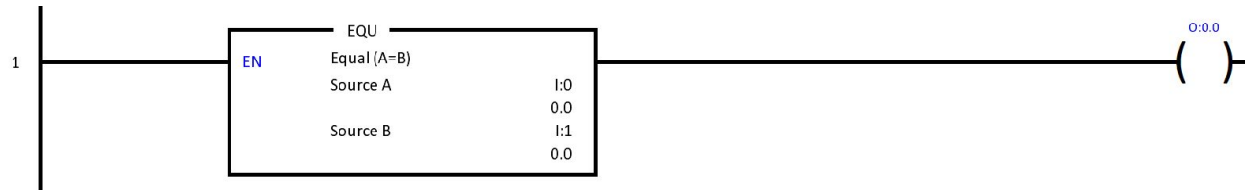
PLC LADDER SIMULATOR 2
FUNCTIONS BLOCKS

EQU (EQUAL)

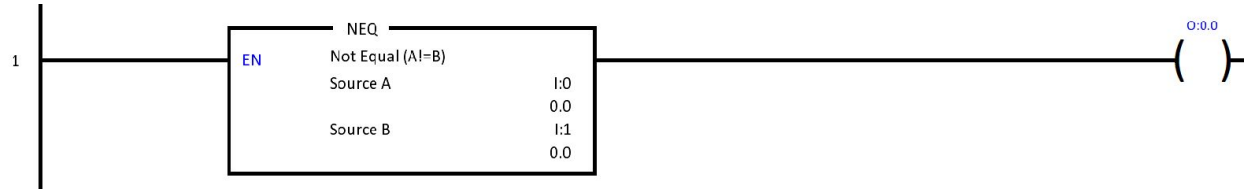
If source A it's equal to source B, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

NEQ (NOT EQUAL)

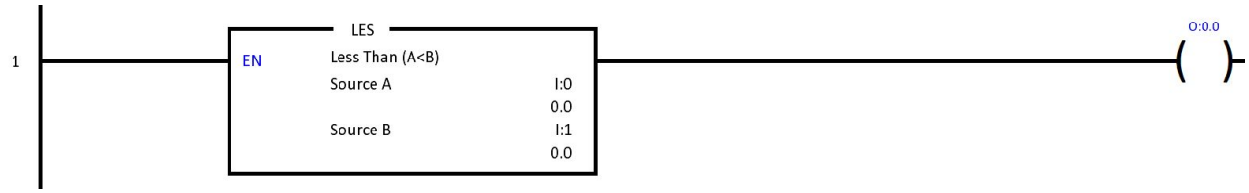
If source A it’s not equal to source B, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

LES (LESS THAN)

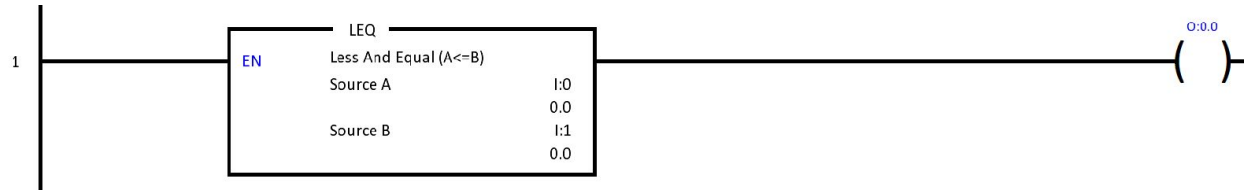
If source A is less than source B, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

LEQ (LESS AND EQUAL)

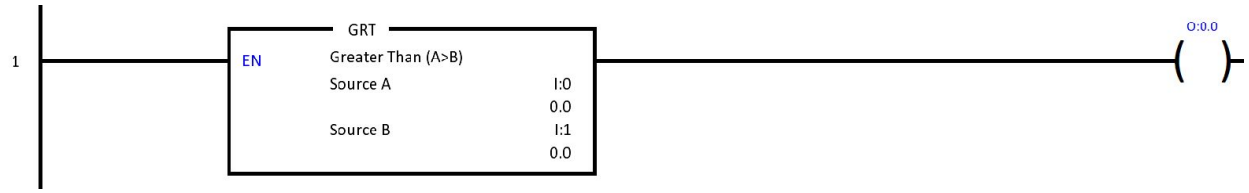
If source A is less than or equal to source B, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

GRT (GREATER THAN)

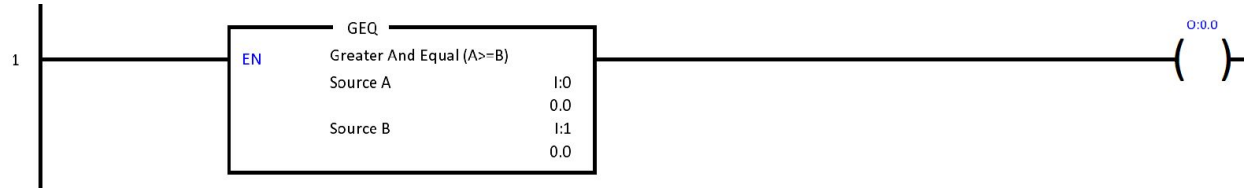
If source A is greater than source B, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

GEQ (GREATER AND EQUAL)

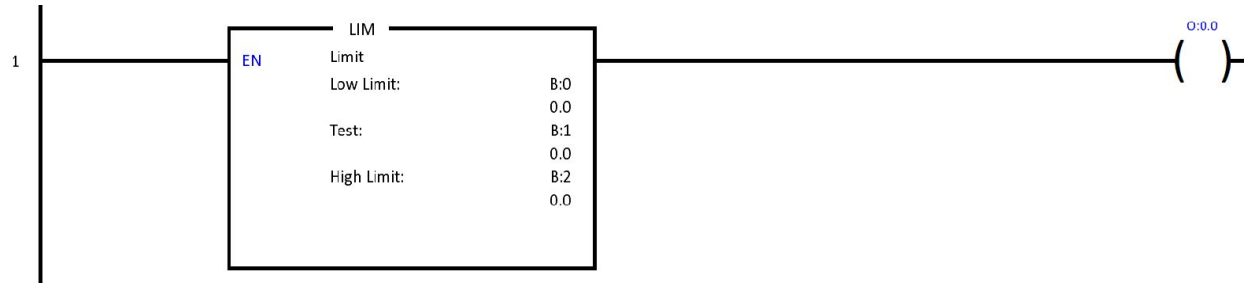
If source A is greater than or equal to source B, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

LIM (LIMIT)

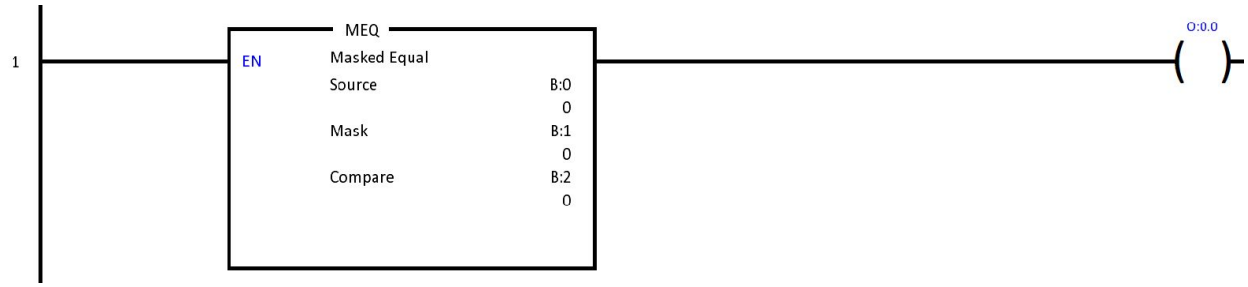
If the test value is in within the range of the low and the high value, the instruction is logically true.



		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Low Limit	Address word or a constant value	[ALL]
Test	Address word	[ALL]
High Limit	Address word or a constant value	[ALL]

MEQ (MASKED EQUAL)

Compares the source data to reference data through a mask.

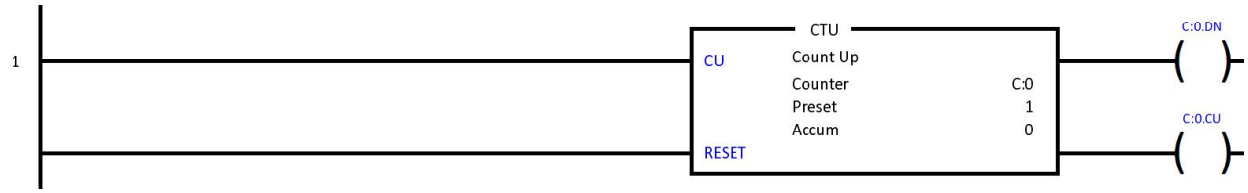


		FILE TYPE
DN (Done)	Output Coil	[O, B, T, C, R]
Source	Address word	[I, O, B, T, C, R, N, D, AI, AO, S]
Mask	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]
Compare	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]

CTU (COUNT UP COUNTER)

Increments the ACC value when the input changes it states from false to true. The DN bit is set when ACC is equal or greater than the preset and is reset when ACC is less than the preset.

If ACC is equal or greater than 32767 the **OV** (overflow) bit is set.
If ACC is equal or lower than -32768 the **UN** (underflow) bit is set.

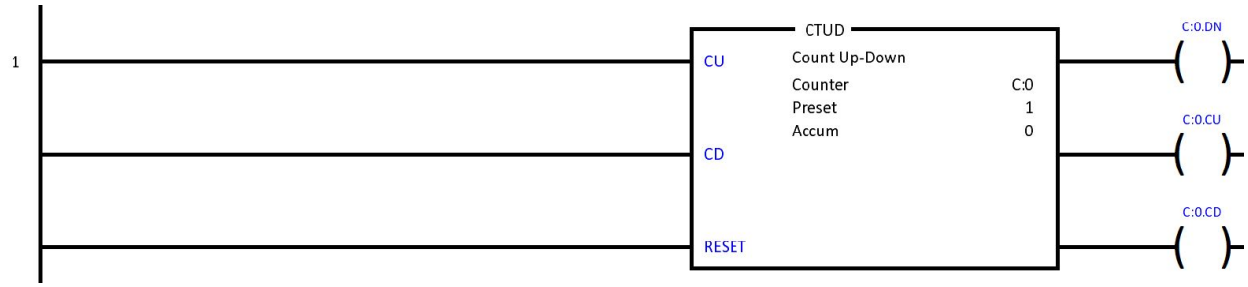


		FILE TYPE
C File	Counter File address	
Preset	Constant value	

CTUD (COUNT UP-DOWN COUNTER)

Increments the ACC value when the CU input changes it states from false to true, decrements the ACC value when the CD input changes it states from false to true. The DN bit is set when ACC is equal or greater than the preset and is reset when ACC is less than the preset.

If ACC is equal or greater than 32767 the **OV** (overflow) bit is set.
If ACC is equal or lower than -32768 the **UN** (underflow) bit is set.



		FILE TYPE
C File	Counter File address	
Preset	Constant value	

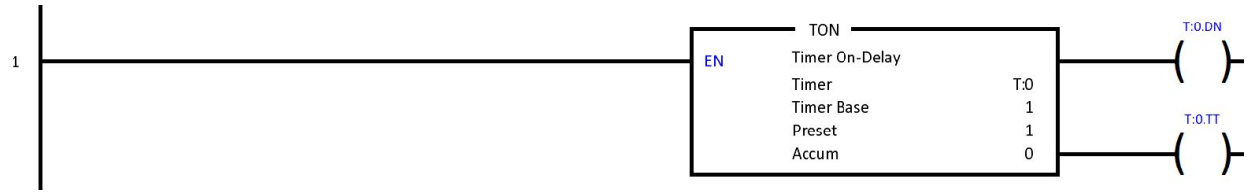
TON (TIMER ON-DELAY)

The timer clock starts counting seconds in ACC when the input is set high. When the input gets low, ACC is set back to 0.

The **BS** (base) bit set the time base of the timer if it's low the time base is 1 second if it's high the time base is 0.1 seconds (100ms). By default the **BS** bit is low.

IMPORTANT: For the Arduino Due the **BS** bit sets a time base of 0.01 seconds (10ms) and for the ESP32 0.001 seconds (1ms).

Time_Delay = Preset_value * Time_Base



		FILE TYPE
T File	Timer File address	
Preset	Constant value	

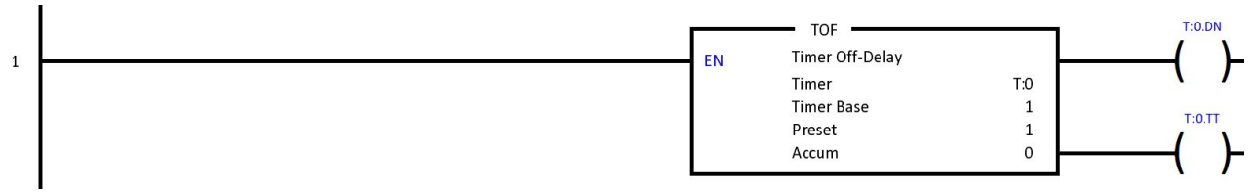
TOF (TIMER OFF-DELAY)

This timer works just like TON except that it starts when the input goes low. When the input goes high, ACC is set back to 0.

The **BS** (base) bit set the time base of the timer if it's low the time base is 1 second if it's high the time base is 0.1 seconds (100ms). By default the **BS** bit is low.

IMPORTANT: For the Arduino Due the **BS** bit sets a time base of 0.01 seconds (10ms) and for the ESP32 0.001 seconds (1ms).

Time_Delay = Preset_value * Time_Base



		FILE TYPE
T File	Timer File address	
Preset	Constant value	

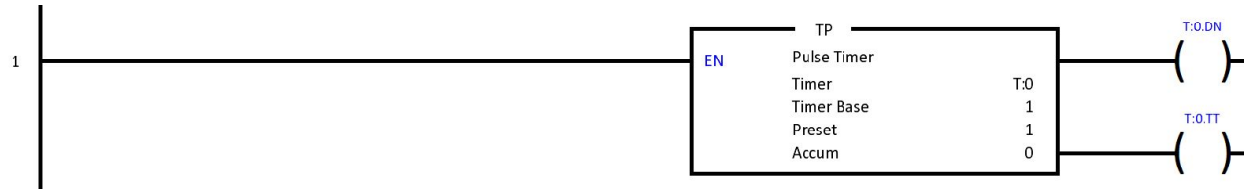
TP (PULSE TIMER)

Outputs a pulse for a duration of PRESET seconds.

The **BS** (base) bit set the time base of the timer if it's low the time base is 1 second if it's high the time base is 0.1 seconds (100ms). By default the **BS** bit is low.

IMPORTANT: For the Arduino Due the **BS** bit sets a time base of 0.01 seconds (10ms) and for the ESP32 0.001 seconds (1ms).

Time_Delay = Preset_value * Time_Base



		FILE TYPE
T File	Timer File address	
Preset	Constant value	

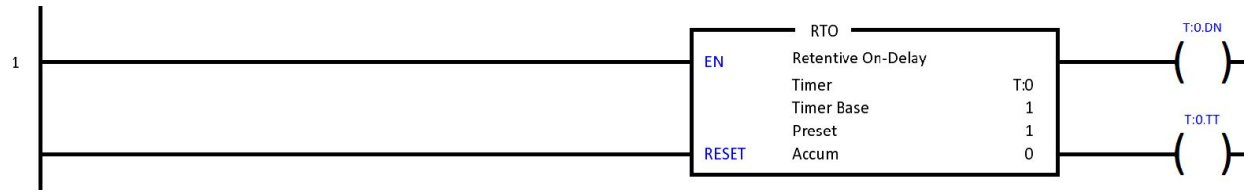
RTO (RETENTIVE TIMER ON)

This timer works like TON except that if during the counting, the input is set low, ACC retains its last value.

The **BS** (base) bit set the time base of the timer if it's low the time base is 1 second if it's high the time base is 0.1 seconds (100ms). By default the **BS** bit is low.

IMPORTANT: For the Arduino Due the **BS** bit sets a time base of 0.01 seconds (10ms) and for the ESP32 0.001 seconds (1ms).

Time_Delay = Preset_value * Time_Base



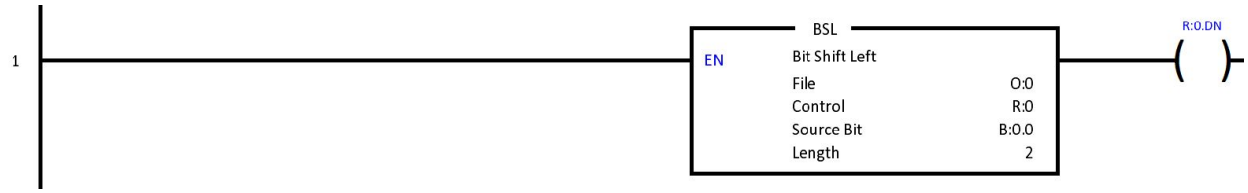
		FILE TYPE

T File	Timer File address	
Preset	Constant value	

BSL (BIT SHIFT LEFT)

It loads data into a bit array one bit at a time. The data is shifted through the bit array from the LSB to the MSB. It unloads the MSB at a time to the **UL** (unload) bit.

The **length** must be greater than 2, and less than 15 for O, B, T, C, R, N, AI, and AO file types, and less than 31 for D file type, if length is not inside this range the **ER** (error) bit is set and the block will not be executed.

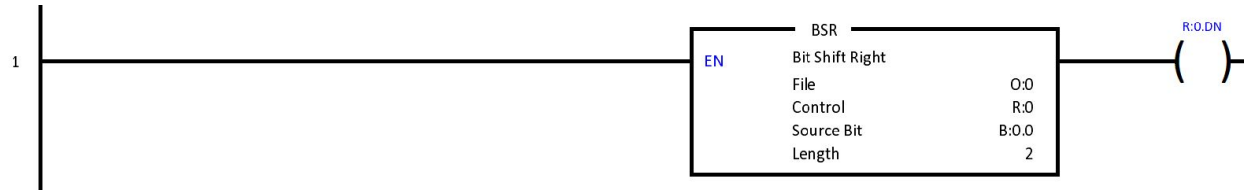


		FILE TYPE
File	Address word	[O, B, T, C, R, N, D, AO]
R File	Control Address word	
Source Bit	Address bit or a constant value (0 or 1)	[I, O, B, T, C, R]
Length	Constant value	

BSR (BIT SHIFT RIGHT)

It loads data into a bit array one bit at a time. The data is shifted through the bit array from the MSB to the LSB. It unloads the LSB at a time to the **UL** (unload) bit.

The **length** must be greater than 2, and less than 15 for O, B, T, C, R, N, AI, and AO file types, and less than 31 for D file type, if length is not inside this range the **ER** (error) bit is set and the block will not be executed.

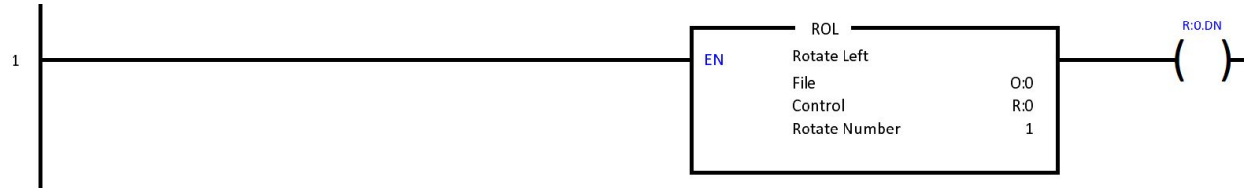


		FILE TYPE
File	Address word	[O, B, T, C, R, N, D, AO]
R File	Control Address word	
Source Bit	Address bit or a constant value (0 or 1)	[I, O, B, T, C, R]
Length	Constant value	

ROL (ROTATE LEFT)

Rotate the data in a bit array one bit at a time, n positions, from the LSB to the MSB. The MSB is also storage in the **UL** (unload) bit.

The **rotate number** must be greater than 1, and less than 15 for O, B, T, C, R, N, AI, and AO file types, and less than 31 for D file type, if the **rotate number** is not inside this range the **ER** (error) bit is set and the block will not be executed.

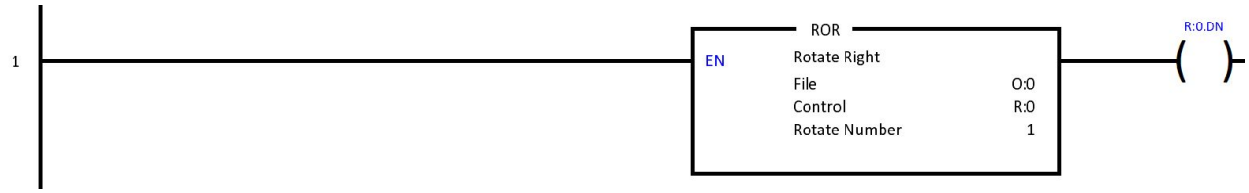


		FILE TYPE
File	Address word	[O, B, T, C, R, N, D, AO]
R File	Control Address word	
Rotate Number	Constant value	

ROR (ROTATE RIGHT)

Rotate the data in a bit array one bit at a time, n positions, from the MSB to the LSB. The MSB is also storage in the **UL** (unload) bit.

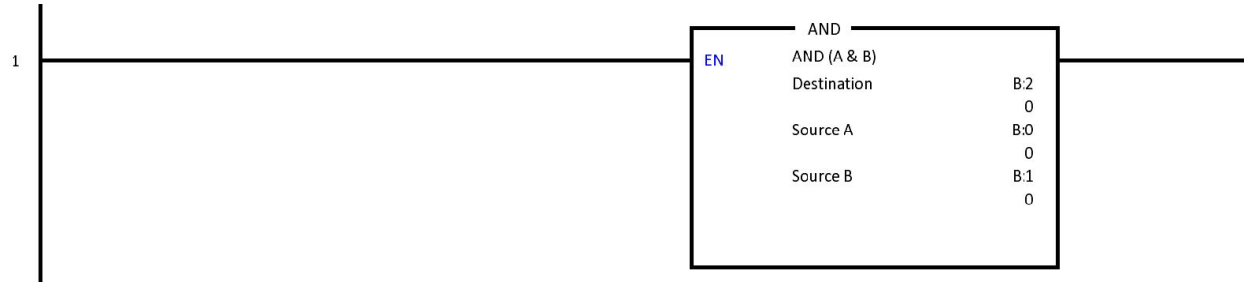
The **rotate number** must be greater than 1, and less than 15 for O, B, T, C, R, N, AI, and AO file types, and less than 31 for D file type, if the **rotate number** is not inside this range the **ER** (error) bit is set and the block will not be executed.



		FILE TYPE
File	Address word	[O, B, T, C, R, N, D, AO]
R File	Control Address word	
Rotate Number	Constant value	

AND (LOGIC AND)

Execute a bit-wise AND operation between source A and source B, the result is saved in the destination variable.

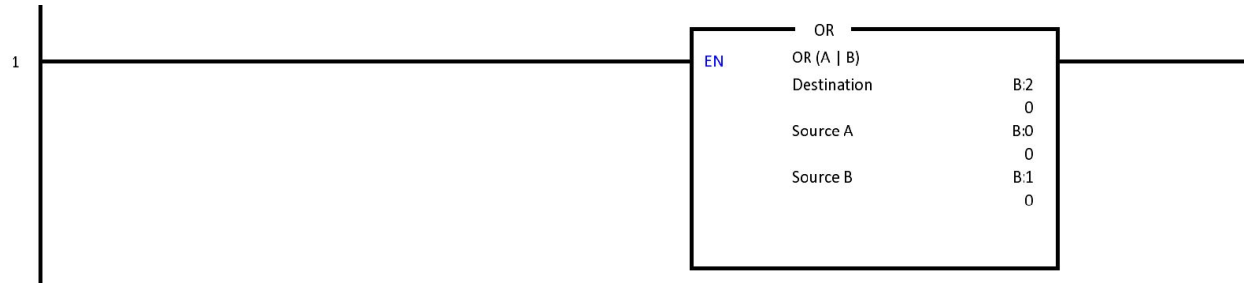


		FILE TYPE

Destination	Address word	[O, B, T, C, R, N, D, AO]
Source A	Address word	[I, O, B, T, C, R, N, D, AI, AO, S]
Source B	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]

OR (LOGIC OR)

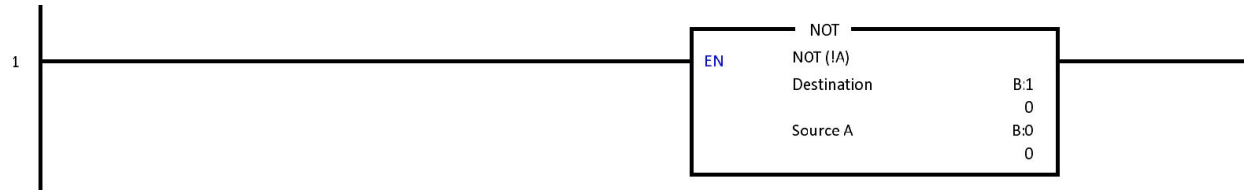
Execute a bit-wise OR operation between source A and source B, the result is saved in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, AO]
Source A	Address word	[I, O, B, T, C, R, N, D, AI, AO, S]
Source B	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]

NOT (LOGIC NOT)

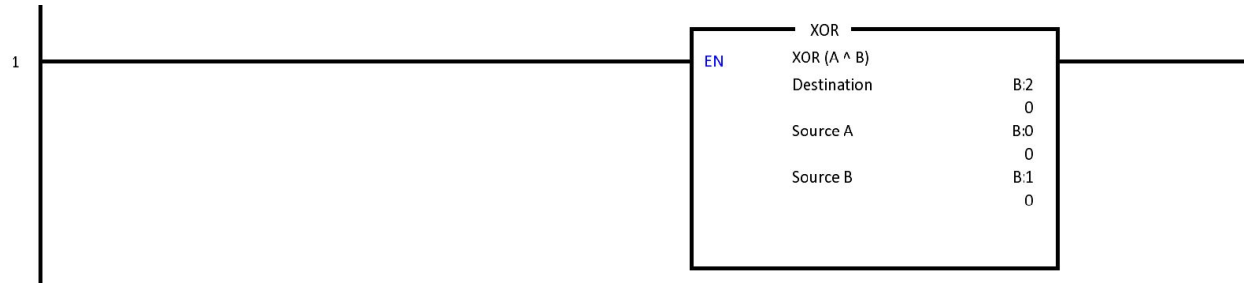
Execute a NOT operation of the source, the result is saved in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, AO]
Source A	Address word	[I, O, B, T, C, R, N, D, AI, AO, S]

XOR (LOGIC XOR)

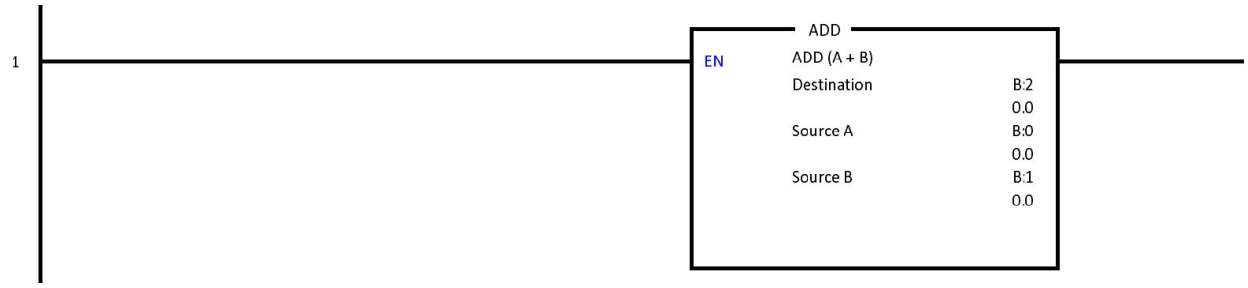
Execute a bit-wise XOR operation between source A and source B, the result is saved in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, AO]
Source A	Address word	[I, O, B, T, C, R, N, D, AI, AO, S]
Source B	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]

ADD (ADDITION)

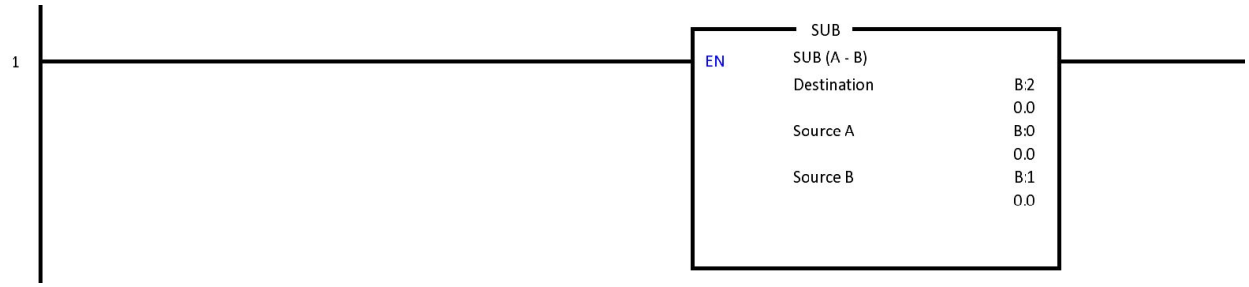
Adds source A to source B, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

SUB (SUBTRACTION)

Subtracts source A to source B, the result is stored in the destination variable.

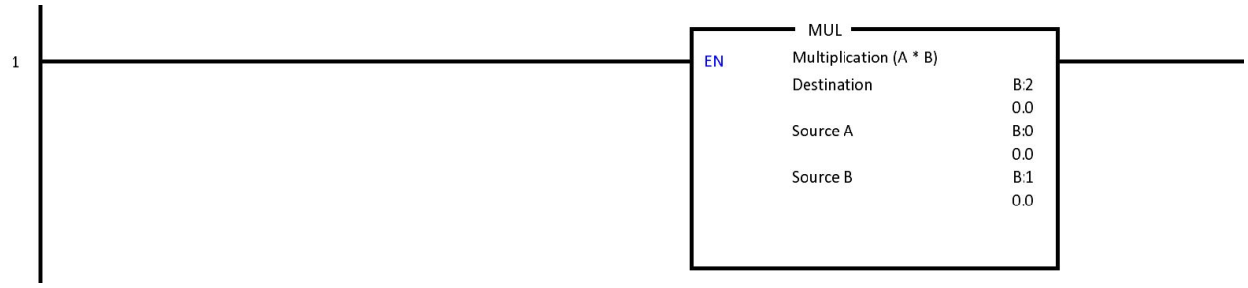


		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

Source B	Address word or a constant value	[ALL]

MUL (MULTIPLICATION)

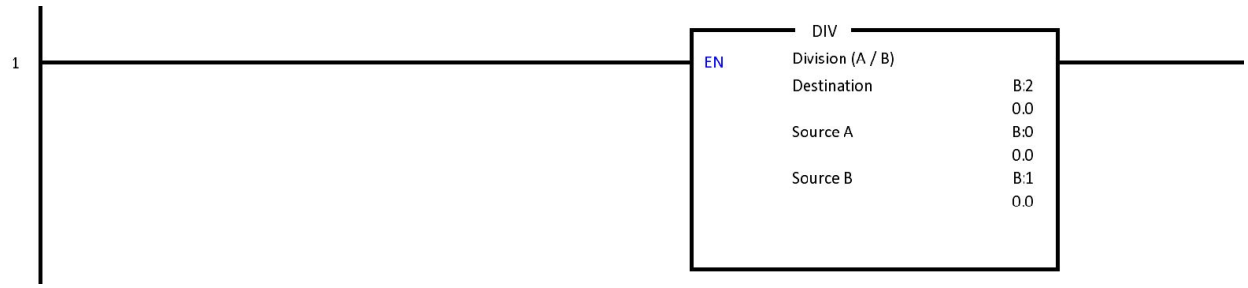
Multiplies source A by source B, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

DIV (DIVISION)

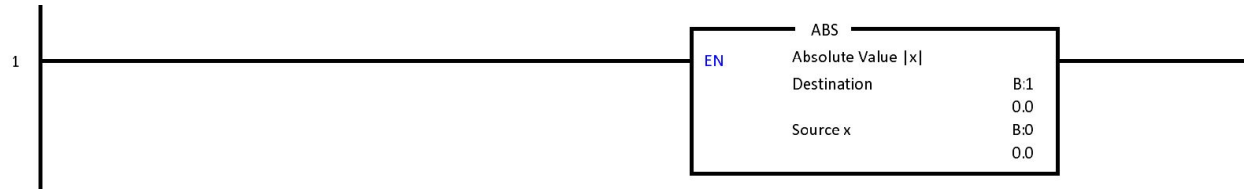
Divides source A by source B, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

ABS (ABSOLUTE VALUE)

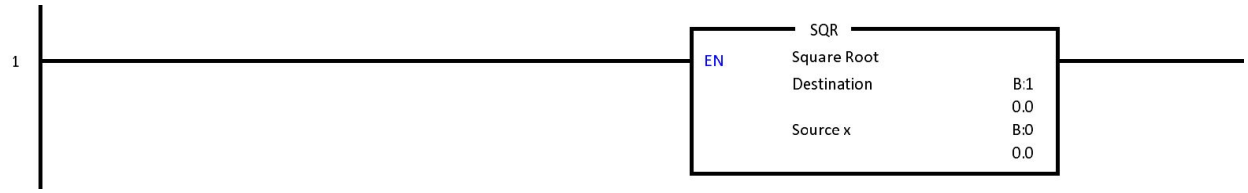
Moves the absolute value of the source into the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

SQR (SQUARE ROOT)

Calculates the positive square root of the source, the result is stored in the destination variable.

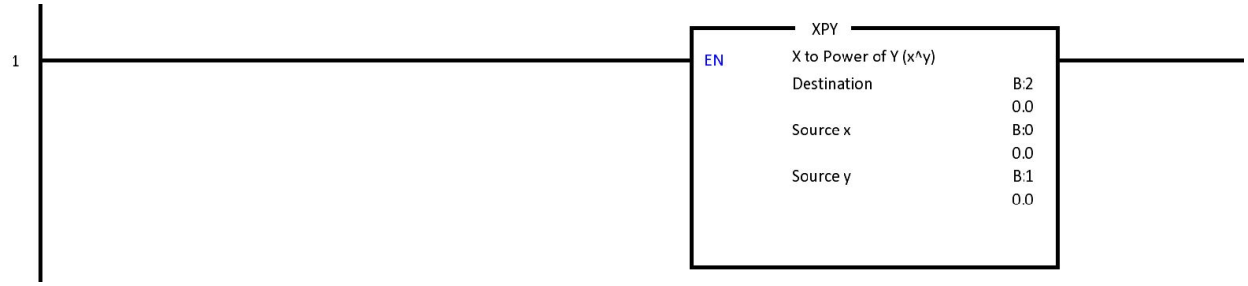


		FILE TYPE

Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

XPY (X TO THE POWER OF Y)

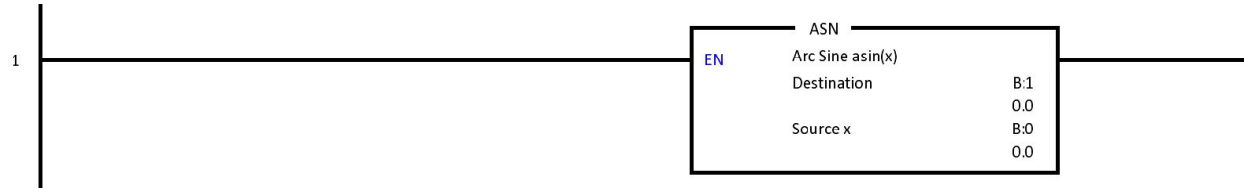
Calculates source A to the power of source B, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]
Source B	Address word or a constant value	[ALL]

ASN (ARC SINE)

Calculates the arc sine of source A, the result (in radians) is stored in the destination variable.

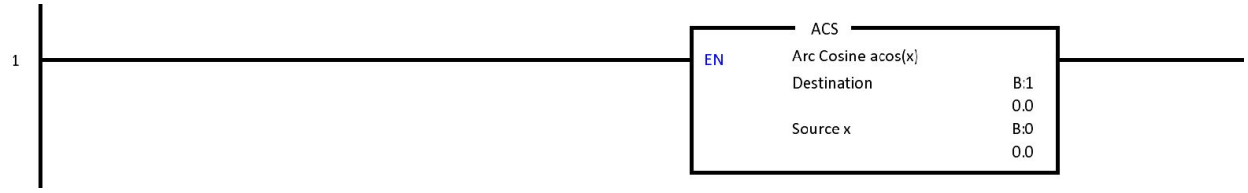


		FILE TYPE

Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

ACS (ARC COSINE)

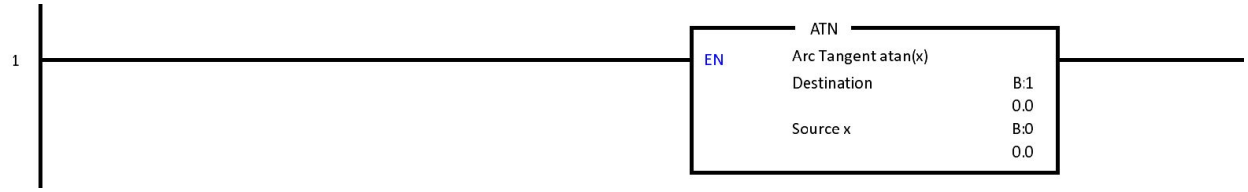
Calculates the arc cosine of source A, the result (in radians) is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

ATN (ARC TANGENT)

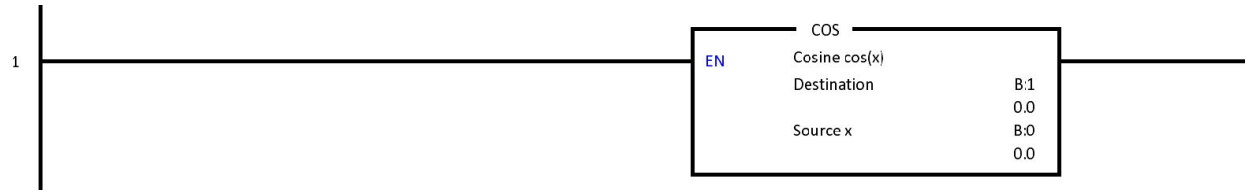
Calculates the arc tangent of source A, the result (in radians) is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

COS (COSINE)

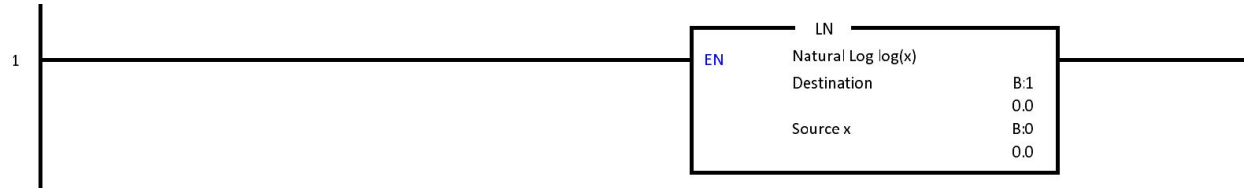
Calculates the cosine of source A, the result (in radians) is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

LN (NATURAL LOGARITHM)

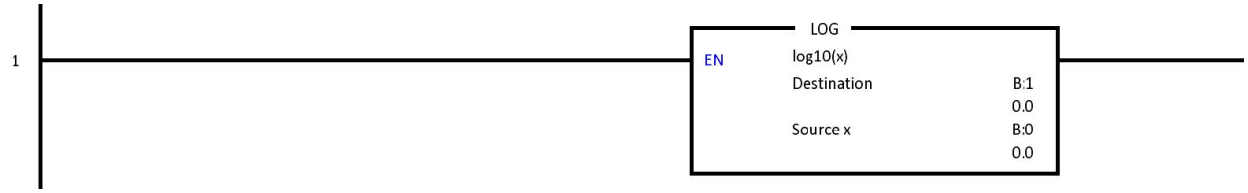
Calculates the natural logarithm of source A, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

LOG (BASE-10 LOGARITHM)

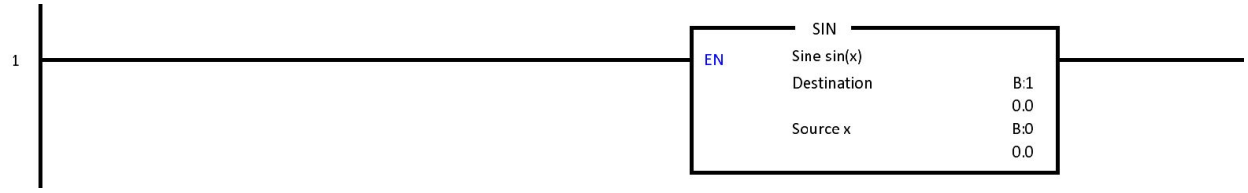
Calculates the base 10-logarithm of source A, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

SIN (SINE)

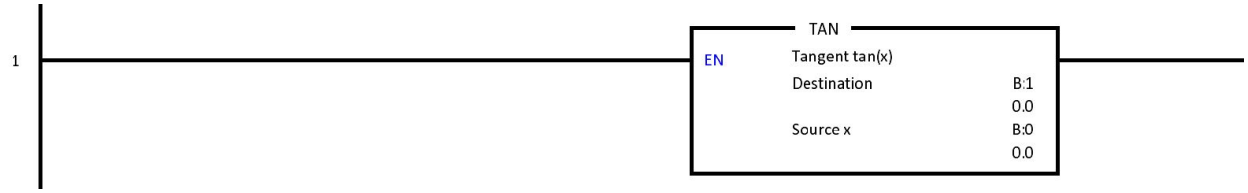
Calculates the sine of source A, the result (in radians) is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

TAN (TANGENT)

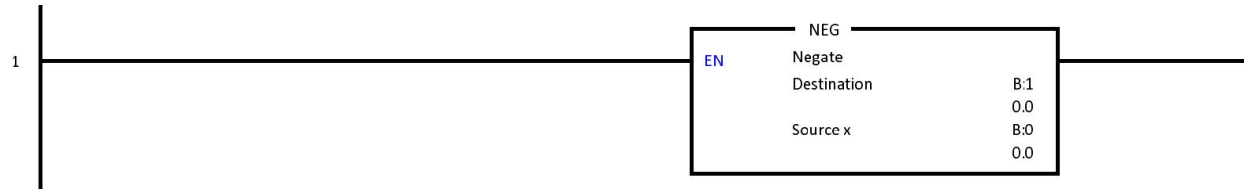
Calculates the tangent of source A, the result (in radians) is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

NEG (NEGATE)

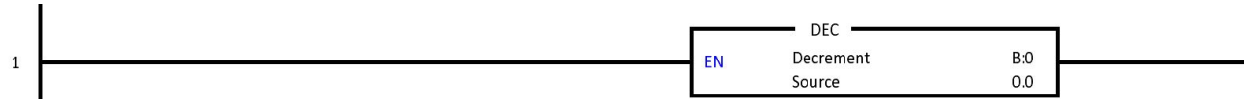
Changes the sign of source A, the result is stored in the destination variable.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

DEC (DECREMENT)

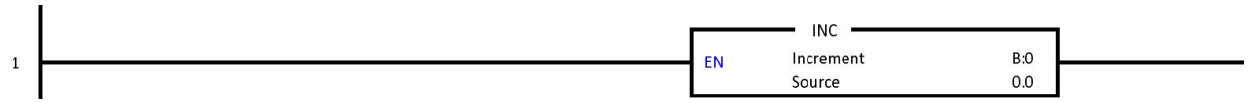
Subtracts one to the source value per program cycle, it’s recommended to use a rising or falling edge contact for the **EN** input.



		FILE TYPE
Source	Address word	[O, B, T, C, R, N, D, F, AO]

INC (INCREMENT)

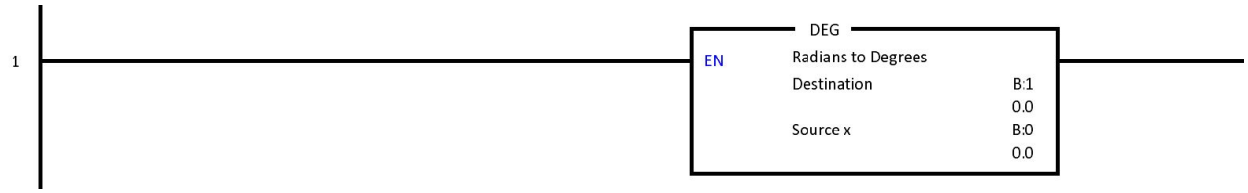
Adds one to the source value per program cycle, it's recommended to use a rising or falling edge contact for the **EN** input.



		FILE TYPE
Source	Address word	[O, B, T, C, R, N, D, F, AO]

DEG (RADIANS TO DEGREES)

Convert Radians to Degrees.

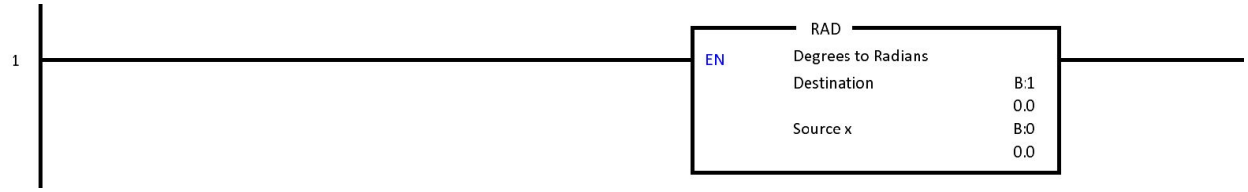


		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]

Source A	Address word	[ALL]

RAD (DEGREES TO RADIANS)

Convert Degrees to Radians.

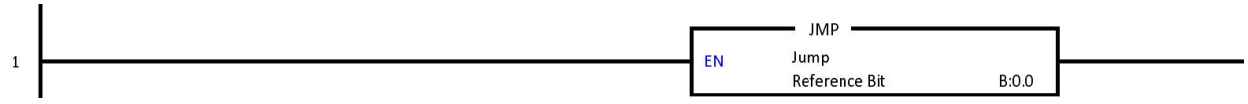


		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

JMP (JUMP TO LABEL)

It skips the next rungs and jumps to the designated label (LBL), it only jumps to labels below its declaration, any label on top of it is ignored. The jump is not made to labels declared in other routines.

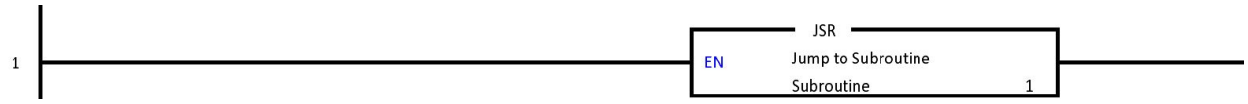
The jump block function uses bits from the B file as a reference, the state of the bit is ignored and do not affect the use of it in other block functions.



		FILE TYPE
Reference Bit	Address bit	[B]

JSR (JUMP TO SUBROUTINE)

Jumps to the beginning of the designated subroutine. The jump is not made if the routine to jump was already called without finishing all the routine.

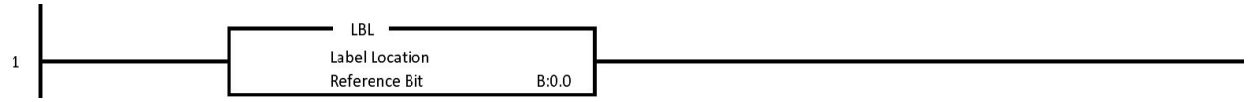


		FILE TYPE
Subroutine	Constant value	

LBL (LABEL LOCATION)

It’s the reference point for the JMP instruction.

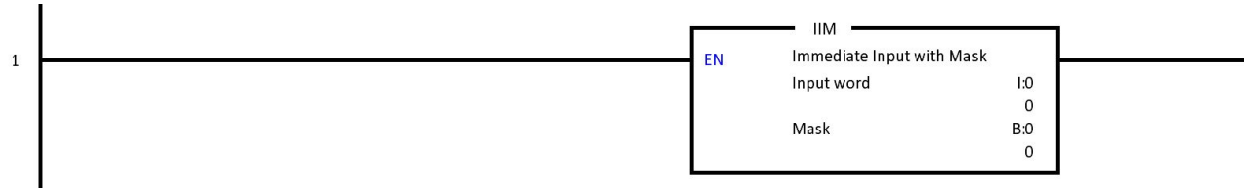
The label block function uses bits from the B file as a reference, the state of the bit is ignored and do not affect the use of it in other block functions. Any contact added at the left of the function will be ignored.



		FILE TYPE
Reference Bit	Address bit	[B]

IIM (IMMEDIATE INPUT MASK)

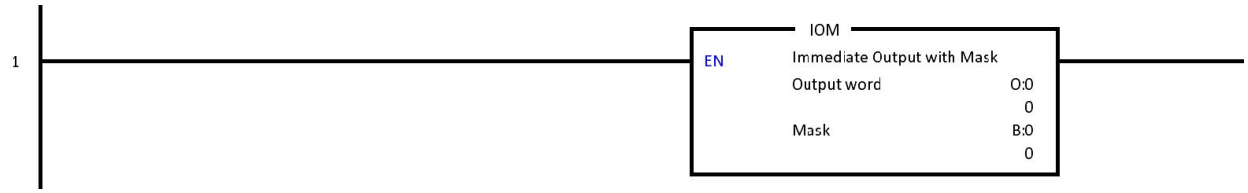
Execute an immediate input scan through a mask. This function only works with development boards projects (Arduino), in the simulation will do nothing.



		FILE TYPE
Input word	Address word	[I]
Mask	Address word or a constant value	[I, O, B, T, C, R, N, AI, AO, S]

IOM (IMMEDIATE OUTPUT MASK)

Execute an immediate output scan through a mask. This function only works with development boards projects, in the simulation will do nothing.



		FILE TYPE
Output word	Address word	[O]
Mask	Address word or a constant value	[I, O, B, T, C, R, N, AI, AO, S]

REF (I/O REFRESH)

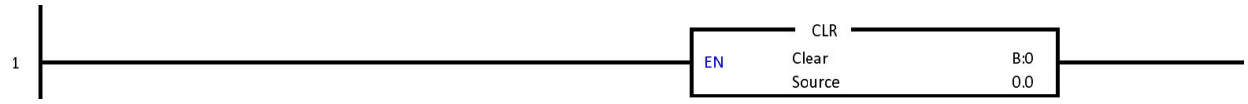
Interrupt the program scan to execute the I/O scan. This function only works with development boards projects, in the simulation will do nothing.

TND (TEMPORARY END INSTRUCTION)

Mark a temporary end that stops the program execution and restarts the scan from the beginning of the program. This function only works for the simulator, will not have any effect for development boards projects (Arduino).

CLR (CLEAR)

Set all bits of a word to zero.

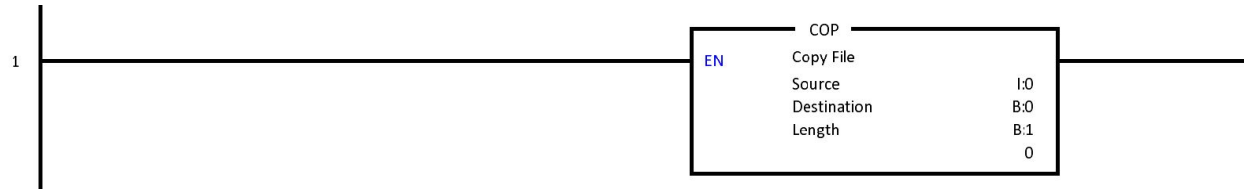


		FILE TYPE
Source	Address word	[O, B, T, C, R, N, D, F, AO]

COP (COPY)

Copies blocks of data from a source address into a destination address. The maximum words that can be copied are 50.

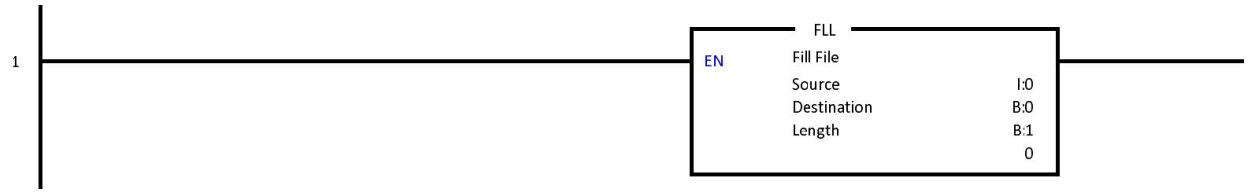
For C, T and R files, the only data that can be copied are the PRE, ACC, LEN and POS words, only the selected word type will be copied for each of the selected file type words.



		FILE TYPE
Source	Address word	[I, O, B, T, C, R, N, D, F, AI, AO]
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Length	Address word or a constant value	[I, O, B, T, C, R, N, D, F, AI, AO]

FLL (FILL)

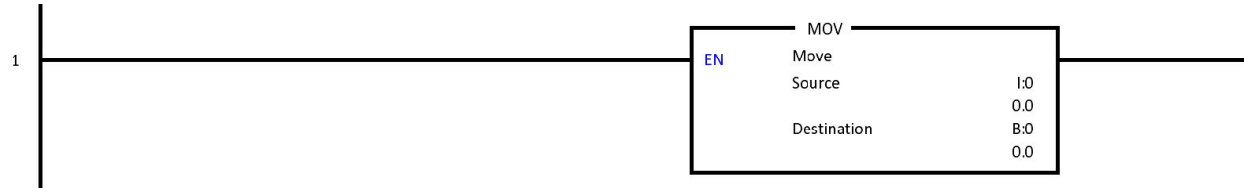
Fills the words of a file with a source value. The maximum value for length is 130 words. The Destination is the starting address of the file to fill.



		FILE TYPE
Source	Address word or a constant value	[I, O, B, T, C, R, N, D, F, AI, AO]
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Length	Address word or a constant value	[I, O, B, T, C, R, N, D, F, AI, AO]

MOV (MOVE)

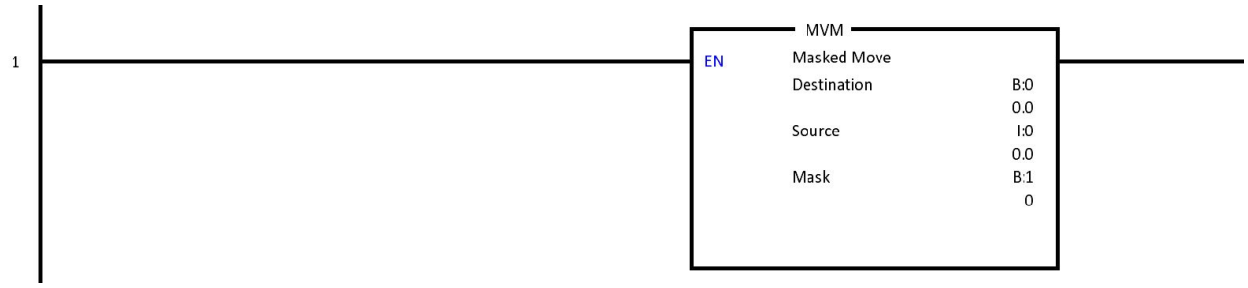
Moves a source value to a destination location without deleting the source value.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source A	Address word	[ALL]

MVM (MASKED MOVE)

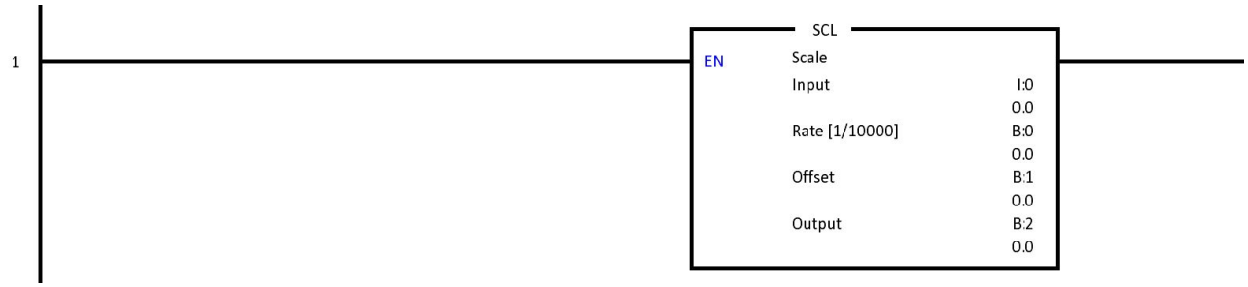
Moves a source value to a destination location through a mask value, the source value is not deleted.



		FILE TYPE
Destination	Address word	[O, B, T, C, R, N, D, F, AO]
Source	Address word or a constant value	[ALL]
Mask	Address word or a constant value	[I, O, B, T, C, R, N, D, F, AI, AO]

SCL (SCALE DATA)

Scales a given value.

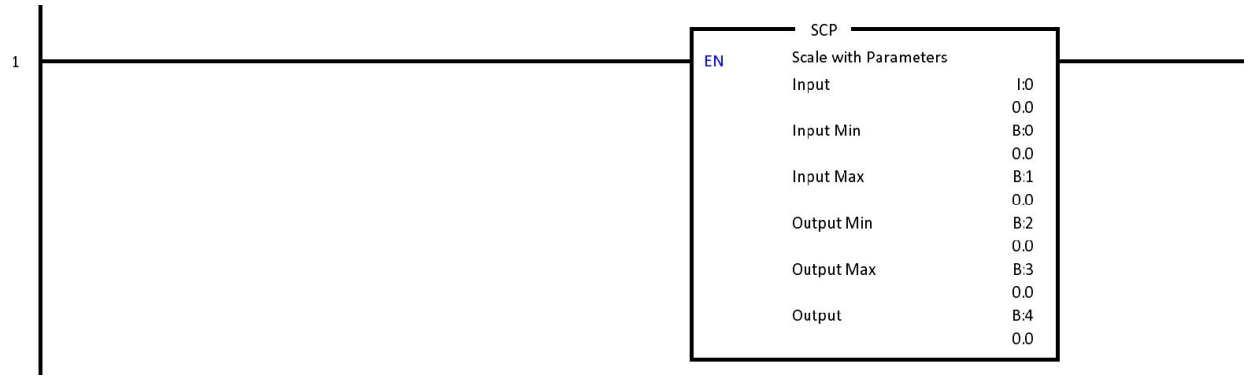


		FILE TYPE
Input	Address word	[ALL]
Rate	Address word or a constant value	[ALL]

Offset	Address word or a constant value	[ALL]
Output	Address word	[O, B, T, C, R, N, D, F, AO]

SCP (SCALE WITH PARAMETERS)

Scales a given value according to some input and output parameters.



		FILE TYPE
Input	Address word	[ALL]
Input Min	Address word or a constant value	[ALL]
Input Max	Address word or a constant value	[ALL]
Output Min	Address word or a constant value	[ALL]
Output Max	Address word or a constant value	[ALL]
Output	Address word	[O, B, T, C, R, N, D, F, AO]

SQC (SEQUENCER COMPARE)

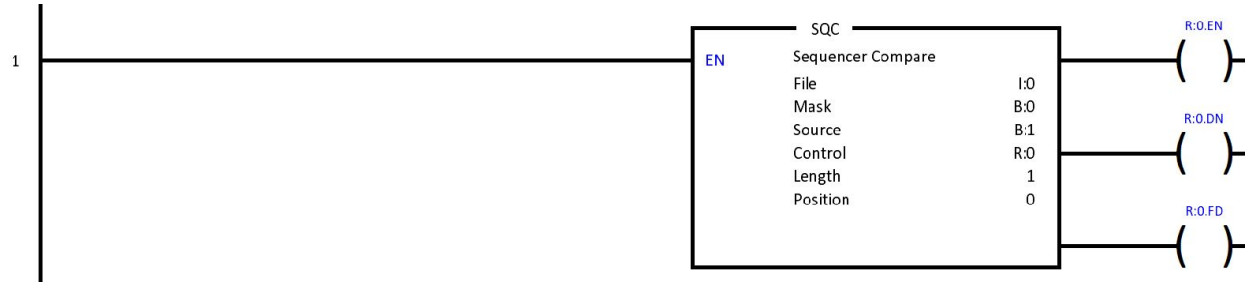
Compares a word from a data file through a mask with a source word through the increment of **POS** (position) every time the instruction state changes from false to true.

The sequencer starts comparing at **POS** 1, when **POS** is greater than **LEN** (length) the sequencer start again at **POS** 1. The maximum length is 130 words.

The **FD** (found) bit is set when the comparison is successful.

The **DN** (done) bit is set when the sequencer has operated in the last word of the selected file. It's reset on the next rung transition from false to true.

The **ER** (error) bit is set when **POS** is a negative value, or **LEN** is less or equal than zero.



		FILE TYPE
File	Address word	[I, O, B, T, C, R, N, D, AI, AO]
Mask	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]
Source	Address word or a constant value	[I, O, B, T, C, R, N, D, AI, AO, S]
R File	Control Address word	
Length	Constant value	

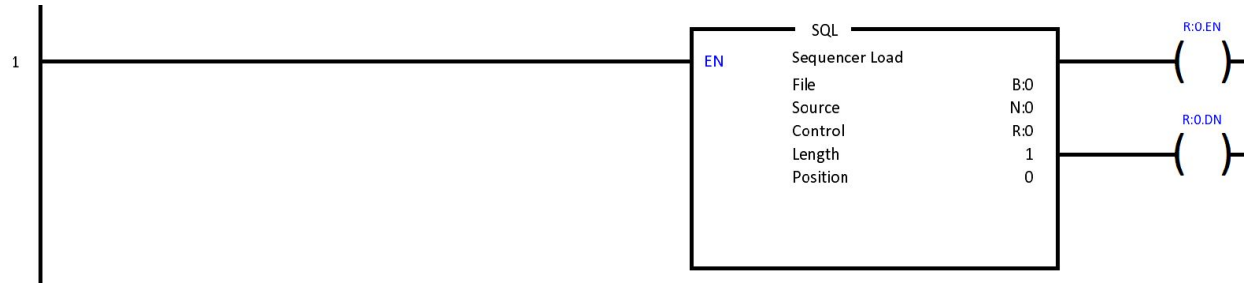
SQL (SEQUENCER LOAD)

Stores a word into a sequential load file through the increment of **POS** (position) every time the instruction state changes from false to true.

The sequencer starts loading data at **POS** 1, when **POS** is greater than **LEN** (length) the sequencer start again at **POS** 1. The maximum length Is 130 words.

The **DN** (done) bit is set when the sequencer has operated in the last word of the selected file. It's reset on the next rung transition from false to true.

The **ER** (error) bit is set when **POS** is a negative value, or **LEN** is less or equal than zero.



		FILE TYPE
File	Address word	[B, T, C, R, N, D, F, AO]
Source	Address word or a constant value	[ALL]
R File	Control Address word	
Length	Constant value	

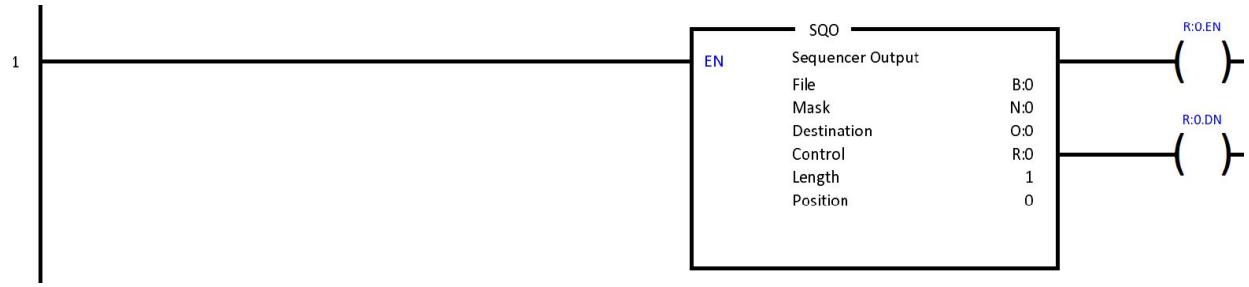
SQO (SEQUENCER OUTPUT)

Transfer a word from a data file through a mask to the destination address word every time the instruction state changes from false to true.

The sequencer starts retrieving data at **POS** 1, when **POS** is greater than **LEN** (length) the sequencer start again at **POS** 1. The maximum length is 130 words.

The **DN** (done) bit is set when the sequencer has operated in the last word of the selected file. It's reset on the next rung transition from false to true.

The **ER** (error) bit is set when **POS** is a negative value, or **LEN** is less or equal than zero.



		FILE TYPE
File	Address word	[B, T, C, R, N, AI, AO]
Mask	Address word or a constant value	[I, O, B, T, C, R, N, AI, AO, S]
Destination	Address word	[O, B, T, C, R, N, AO]
R File	Control Address word	
Length	Constant value	