



# Andhra Pradesh State Skill Development Corporation



The image is a composite of two parts. On the left, there is a diagram of a Learning Management System (LMS). It features a central computer monitor displaying the 'LMS' logo. Various icons and text labels are connected by lines to different parts of the system: 'courses' (top), 'documentation' (top right), 'tracking' (right), 'e-learning management' (bottom right), 'education' (bottom left), 'system' (left), and 'software' (top left). On the right, there is a photograph of three individuals (two men and one woman) wearing headsets and working on desktop computers in what appears to be a call center or customer service environment.

## Basics of PLC

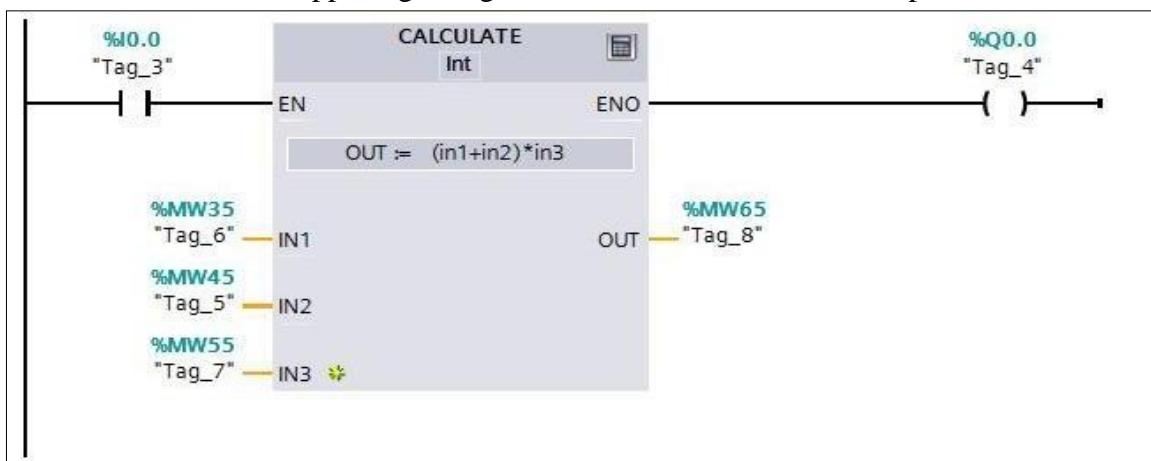
**Math Functions and Comparator Operations**



## Basic Mathematical Functions:

### *CALCULATE: Calculate*

The "Calculate" instruction is used to define and execute an expression for the calculation of mathematic You can select the data type of the instruction from the "<??>" drop-down list of the instruction box. Depending on the data type selected, you can combine the functions of certain instructions to perform a complex calculation. The information for the expression to be calculated is entered in a dialog, which you can open with the icon at the upper right edge of the instruction box. The expression can



contain names of input parameters and the syntax of the instructions. Operand names and operand addresses cannot be specified.

### Parameters

Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input
ENO	Output	BOOL	Enable Output
IN1	Input	integers, floating-point numbers	First available input
IN2	Input	Integers, floating-point numbers	Second available input
OUT	Output	Integers, floating-point numbers	Output to which the end result is to be transferred

### *1. ADD: Addition*

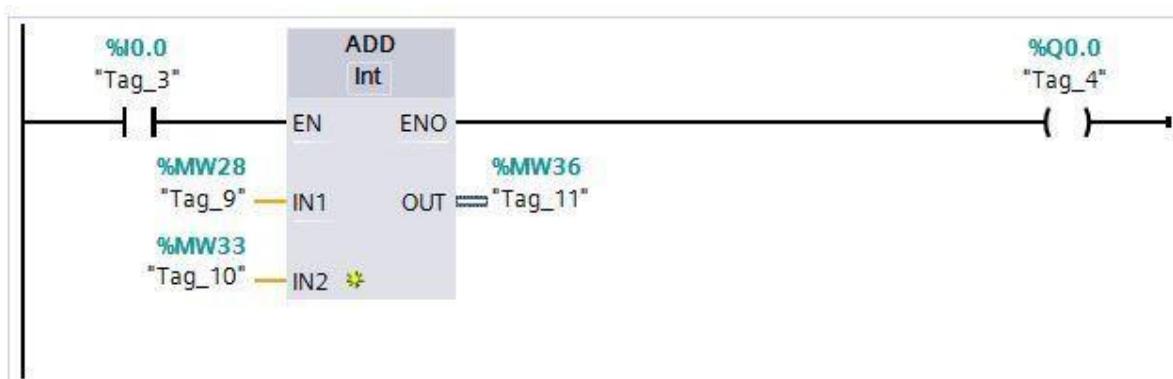
You can use the "Add" instruction to add the value at input IN1 and the value at input IN2 and query the sum at output OUT ( $OUT := IN1 + IN2$ ).



In its initial state, the instruction box contains at least 2 inputs (IN1 and IN2). The number of inputs can be extended. The inserted inputs are numbered in ascending order in the box. When the instruction is executed, the values of all available input parameters are added. The sum is stored at the OUT output.

The ENO enable output has the signal state "0" if one of the following conditions is fulfilled: The enable input EN has the signal state "0".

The result of the instruction is outside the range permitted for the data type specified at the OUT output.



A floating-point number has an invalid value.

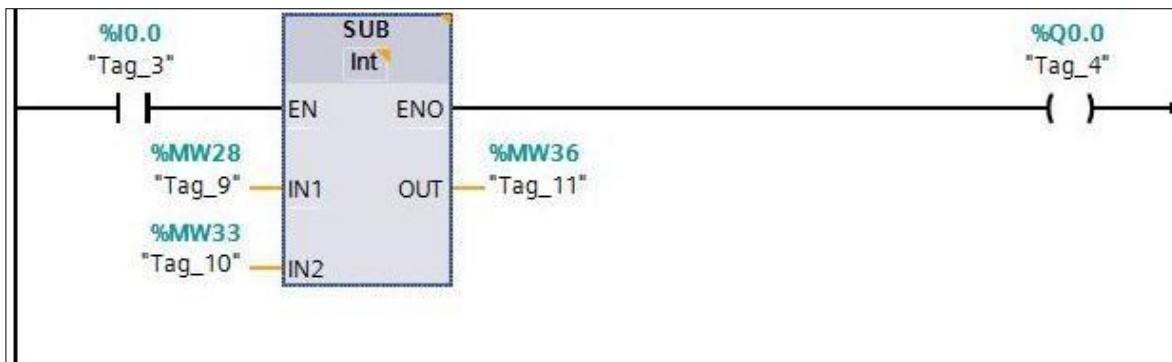
## Parameters

Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input
ENO	Output	BOOL	Enable Output
IN1	Input	integers, floating numbers	First number to be added
IN2	Input	Integers, floating numbers	Second number to be added
OUT	Output	Integers, floating numbers	Sum

## 1. SUB: Subtract

You can use the "Subtract" instruction to subtract the value at input IN2 from the value at input IN1 and query the difference at output OUT ( $OUT := IN1 - IN2$ ).

The ENO enable output has the signal state "0" if one of the following conditions is fulfilled:



The EN enable input has the signal state "0".

The result of the instruction is outside the range permitted for the data type specified at the OUT output.

A floating-point number has an invalid value.

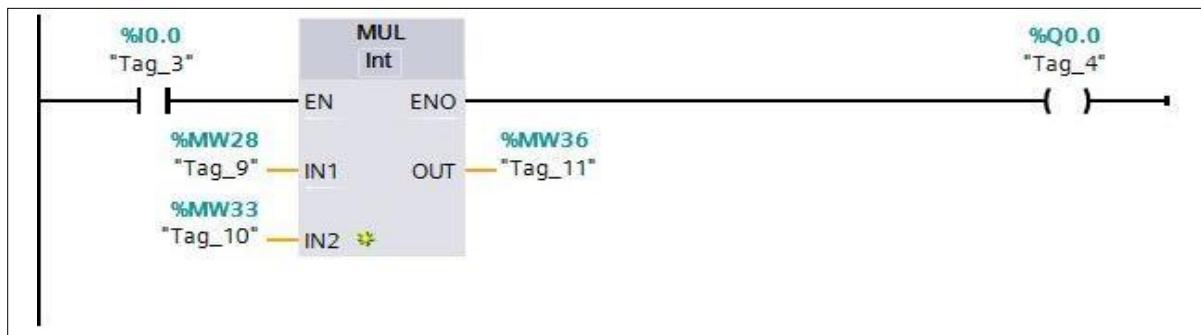
#### Parameter:

Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input
ENO	Output	BOOL	Enable Output
IN1	Input	integers, floating-point numbers	Minuend
IN2	Input	Integers, floating-point numbers	Subtracting
OUT	Output	Integers, floating-point numbers	Difference

#### 1. MUL: Multiply

You can use the "Multiply" instruction to multiply the value at input IN1 with the value at input IN2 and query the product at output OUT ( $OUT := IN1 * IN2$ ).

The number of inputs can be expanded in the instruction box. The added inputs are numbered in ascending order in the box. When the instruction is executed, the values of all available input parameters are multiplied. The product is stored at the OUT output.



The ENO enable output has the signal state "0" if one of the following conditions is fulfilled:

The EN input has the signal state "0". The result is outside the range permitted for the data type specified at output OUT. A floating-point number has an invalid value.

#### Parameters:

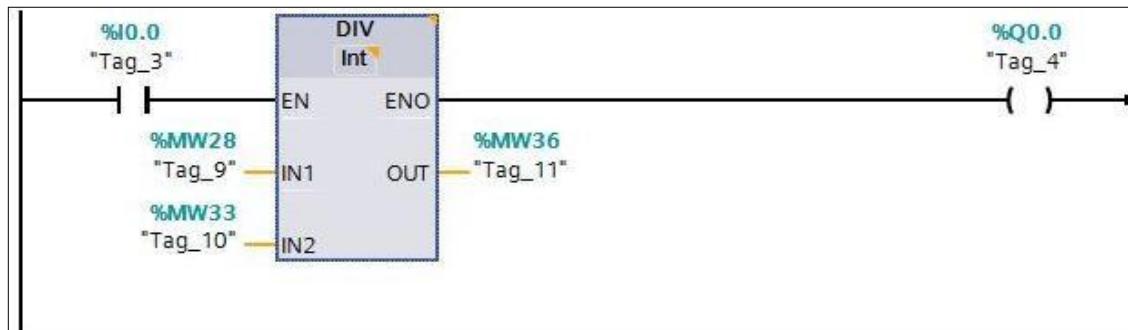
Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input
ENO	Output	BOOL	Enable Output
IN1	Input	integers, floating-point numbers	Multiplier
IN2	Input	Integers, floating-point numbers	Number being multiplied
OUT	Output	Integers, floating-point numbers	Product

## 2. DIV: Divide

You can use the "Divide" instruction to divide the value at input IN1 by the value at input IN2 and query the quotient at output OUT ( $OUT := IN1/IN2$ ).

The ENO enable output has the signal state "0" if one of the following conditions is fulfilled: The EN enable input has the signal state "0".

The result of the instruction is outside the range permitted for the data type specified at the OUT output.



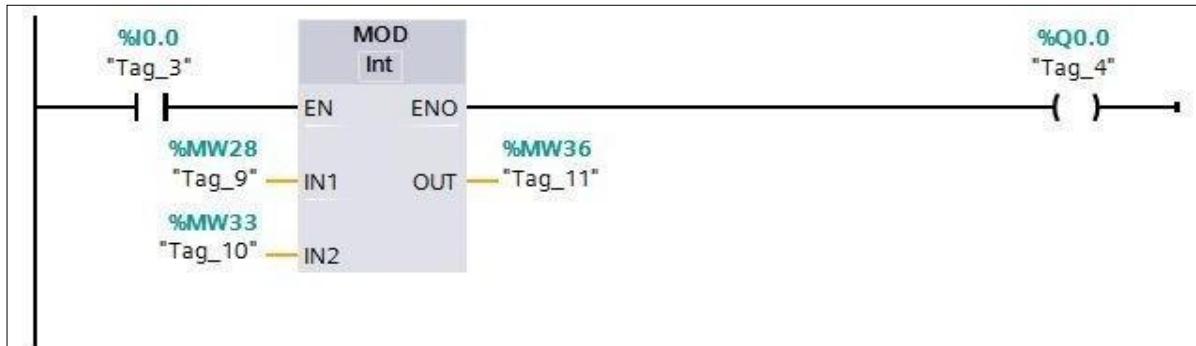
A floating-point number has an invalid value.

#### Parameters:

Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input
ENO	Output	BOOL	Enable Output
IN1	Input	integers, floating-point numbers	Dividend
IN2	Input	Integers, floating-point numbers	Divisor
OUT	Output	Integers, floating-point numbers	Quotient value

#### 1. MOD: Return remainder of division

You can use the "Return remainder of division" instruction to divide the value at input IN1 by the value at input IN2 and query the remainder of division at output OUT.



#### Parameters:



Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input
ENO	Output	BOOL	Enable Output
IN1	Input	Integers	Dividend
IN2	Input	Integers	Divisor
OUT	Output	Integers	Remainder of division

## Date and Time of day: RD\_SYS\_T



## RD\_SYS\_T

You can use RD\_SYS\_T to read the current date and current time of the CPU clock. The data is provided in DTL format at the OUT output of the instruction. The provided value does not include information about the local time zone or daylight saving time. At the RET\_VAL output, you can query whether errors have occurred during execution of the instruction.

### DTL:

Byte	Component	Data type	Value range



0	Year	UINT	1970 to 2664
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of week	USINT	1(Sunday) to 7 (Saturday) The weekday is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			