nubitek
free utils

www.nubitek.com

Nubitek Free Utils

User Manual

Nubitek Free Utils

User manual

Last updated: september 2007

The information in this document belongs to Nubitek industrial IT, S.L. (hereinafter referred to as NUBITEK). en adelante NUBITEK). Authorization is granted by NUBITEK only fot the professional use of this document provided that the following conditions are met: (1) indication that rights are reserved appears in all copies together with the present authorization, (2) this document may be used for informative, personal and non-commercial purposes only, and may not be copied or posted on any network computer or broadcast in any media, and (3) no modification is made to this document. All other uses are specifically prohibited. NUBITEK makes no representations about the suitability of the information contained in the documents and graphics published on this website for any specific purpose. All contets and graphics are published "as is" and without warranties of any kind.

This documents may contain technical inaccuracies or typographical errors. Its contents are subject to periodic changes and updates. NUBITEK may, at any given moment, make improvements and/or modifications to this document.

Index

1. Introduction.

This guide offers clear and straightforward information on the freeware Nubitek Free Utils libraries. Its shows how to install them in STEP7- MicroWIN and how to use them to make the most of the S7-200.

To whom is this document addressed.

- S7-200 developers who wish to enrich the set of functions of STEP7 – MicroWIN by means of Nubitek's utility libraries.

How is the document organized?

Chapter 2 gives a summary of the functions included in Nubitek's utility libraries.

Chapters 3 and 4 explain how to prepare the Step-7 MicroWin software development environment to use the libraries.

The following chapters cover in detail each and every one of the functions making up the libraries, describing functionalities and parameterization methods:

- Chapter 5: Nubitek_String_Utils, with character string management routines.
- Chapter 6: Nubitek_Time_Utils, with functions relating to date and hour.
- Chapter 7: Nubitek_Math_Utils, with mathematical calculation applications.

Notations used in this guide

This document uses differences in print to identify the type of text, as shown in the following table:

Print	Use and examples
Italics	Menu options Example: *File -> Add or remove library*
Bold	Important notifications for the reader. Example: **Attention**
`Monospaced`	File names and directory paths. Example: `Nubitek_String_Utils.mwl`
	Text referring to program code.

2. Summary of Nubitek Free Utils v1.0 functions.

Nubitek Free Utils consists of 3 libraries:

Nubitek_Time_Utils

- **DATE_String**: generates a text string with current date.
- **TIME_String**: generates a text string with current hour.
- **DATETIME_String**: generates a text string with current date and hour.
- **ITIME_String**: generates a text string with conversion into hours, minutes and seconds of a time interval measures in milliseconds.
- **Time_ALARM**: gives an alert signal on the hour set, every day.
- **Day_And_Time_ALARM**: gives an alert signal on the set time and day.

Nubitek_Math_Utils

- **atan:** arctangent function calculation.
- **acos:** arccosine function calculation.
- **asin:** arcsine function calculation.
- **Mean_Variance:** mean and variance calculation of a sequence of values.
- **Analog_Adapter:** carries out the scaling of an analog input to obtain a value with physical sense of the measured dimension.

Nubitek_String_Utils

- **Small_Letter:** converts the characters of a text string into small letters.
- **Capital_Letter**: converts the characters of a text string into capital letters.
- **Get_Value:** captures a numeric value inserted in a text string.
- **S_Compare:** compares two character strings.

3. Installing libraries.

To be able to make use of any of the application libraries of Nubitek, the first step is to install it in the STEP 7 MicroWIN programming environment. Next shown is the installation of the Nubitek_String_Utils library. One must do the same for the rest.

- Download the libraries from the Nubitek website downloads page (`www.nubitek.com`): the libraries are downloaded together as a single compressed file (`nubitek_free_utils.zip`), but are installed separately.
- Decompress the .zip file and the 3 .mwl files contained within.
- Open STEP7-MicroWIN (STEP7 MicroWIN v4.0 SP2 or higher is required)
- From the menu bar, click on *File -> Add or remove library...*

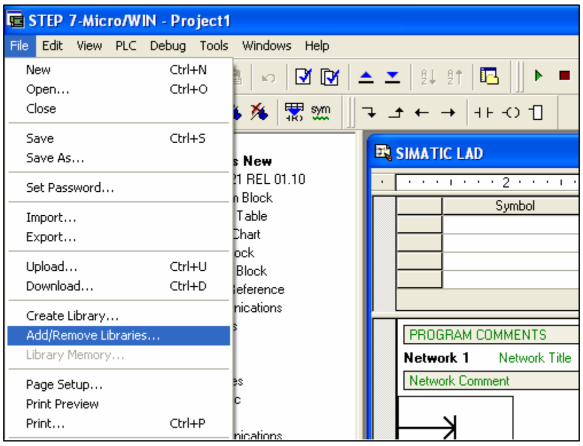
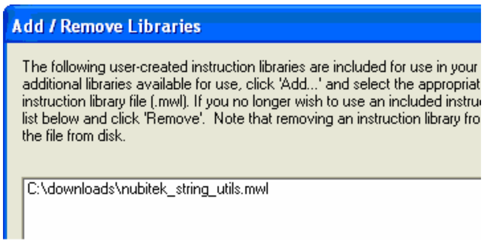


Image 4.1: Add library

- Click on *Add*, enter the directory where you downloaded SMS Manager and select the `.mwl` file of the library. Once the library has been added, click on *Accept*.



Add / Remove Libraries

The following user-created instruction libraries are included for use in your additional libraries available for use, click 'Add...' and select the appropriat instruction library file (.mwl). If you no longer wish to use an included instru list below and click 'Remove'. Note that removing an instruction library fro the file from disk.

C:\downloads\nubitek_string_utils.mwl

Image 4.2: Add library

- At this point the SMS Manager library should already be installed. You can verify that the library has been properly installed by accessing the *Libraries* icon in the operation tree: the library folder and the 3 library functions should appear.

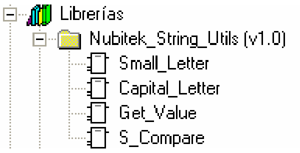


Image 4.3: Add library

4. Allocating memory to the libraries.

Nubitek_Math_Utils and Nubitek_String_Utils do not require global memory reserve for its internal use. Nonetheless, the Nubitek_Time_Utils library requires 60 bytes in V-memory to work properly.

Once you begin to use the library functions in your project, and always before you upload or compile it, you should assign a memory range for the library, whose location can be defined by the user.

- Click on *File->Library memory*:

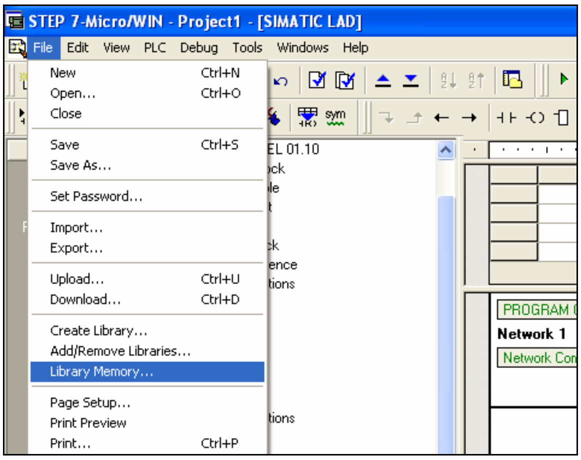
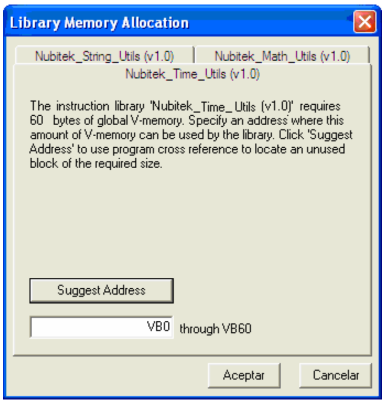


Image 5.1: Library memory

- In the window that opens, select the Nubitek_Time_Utils tab (if you do not have any other libraries installed this will be the only tab available).
- Click on the *Suggest address* button and MicroWIN will indicate a free memory range.
- If for any reason you prefer a different range, all you need to do is click *Suggest address* again or manually write the desired start address. **Important:** user variables should not be included in the memory range used by the library.
- Click on *Accept.*


Library Memory Allocation
Nubitek_String_Utils (v1.0)
Nubitek_Math_Utils (v1.0)
Nubitek_Time_Utils (v1.0)
Suggest Address
Aceptar
Cancelar

The instruction library 'Nubitek_Time_Utils (v1.0)' requires 60 bytes of global V-memory. Specify an address where this amount of V-memory can be used by the library. Click 'Suggest Address' to use program cross reference to locate an unused block of the required size.

VB0 through VB60

Image 5.2: Library memory Allocation

5. Nubitek_Time_Utils Library.

The utilities library for Nubitek_Time_Utils date and hour is composed of 6 functions:

DATE_String

Generates a text string with the current date and places it in the memory address passed as parameter `P_String` (in pointer &VBn format). The format of the generated chain is "DD-MM-YYYY", for instance "12-10-2007".

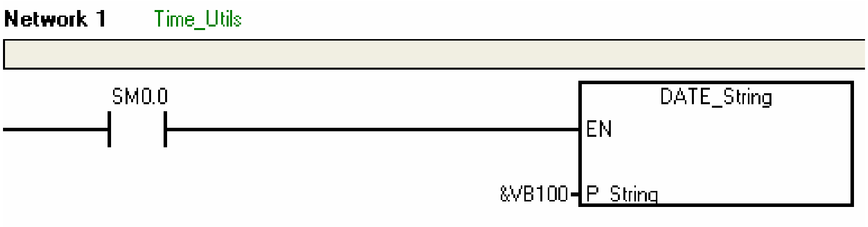


Image 5.1: DATE_String function.

TIME_String

Generates a text chain with current hour and places it in the memory address passed as parameter `P_String` (in pointer &VBn format). The format of the generated chain is "HH:MM:SS", for instance "18:45:50".

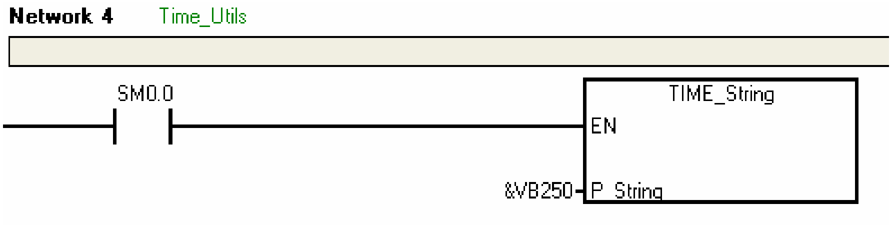


Image 5.2: TIME_String function.

DATETIME_String

Generates a text chain with current date and hour and places it in the memory address passed as parameter `P_String` (in pointer &VBn format). The format of the generated chain is "DD-MM-YYYY HH:MM:SS", for instance "12-10-2007 18:43:22".

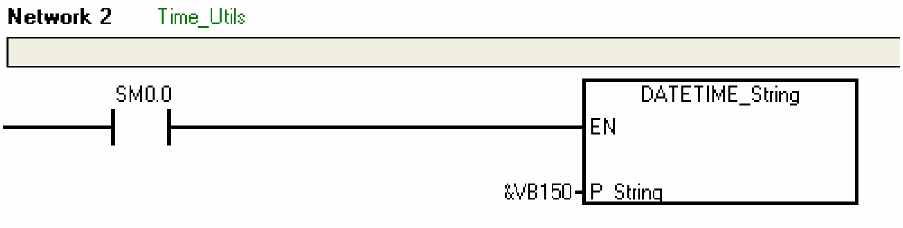


Image 5.3: DATETIME_String function.

ITIME_String

Generates a text chain with conversion into hours, minutes and seconds of a time interval expressed in milliseconds and placed in the memory address passed as parameter `P_String` (in pointer &VBn format). The format of the generated chain is "X Hours, Y Minutes, Z Seconds".

The variable with the time interval in milliseconds must be stated as an input parameter `ms` (in signless DINT format). The ITIME_String function complements the BGN_ITIME and CAL_ITIME functions of STEP7-MicroWin

Image example: if MD0 is 8621266 "2 Hours, 23 Minutes, 41 Seconds" is generated in position VB200

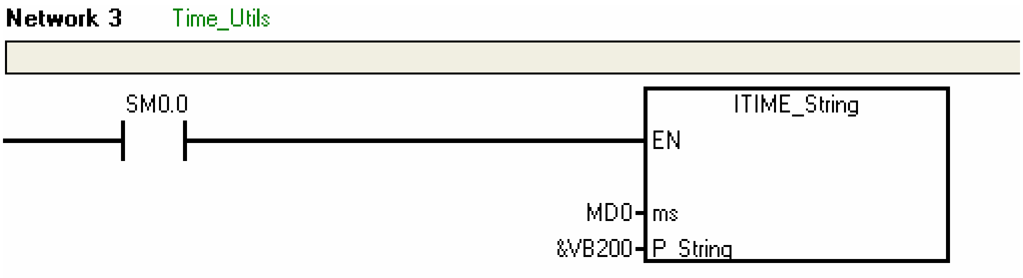


Image 5.4: ITIME_String function.

Time_ALARM

Activates the output bit `OUT` at the time shown in the input `Alarm_Time` by means of a text chain with format "HH:MM:SS".

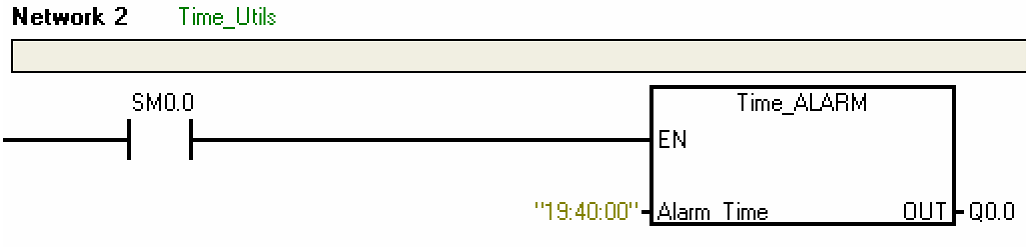


Image 5.5: Time_ALARM function

Day_And_Time_ALARM

Activates the output bit `OUT` at the time and day shown in the input `Alarm_Time` by means of a text chain with format "HH:MM:SS DD".

Weekdays are shown as: MO (Monday), TU (Tuesday), WE (Wednesday), TH (Thursday), FR (Friday), SA (Saturday) and SU (Sunday).

Image example: Q0.1 is activated on Friday at 19:42:30 p.m.

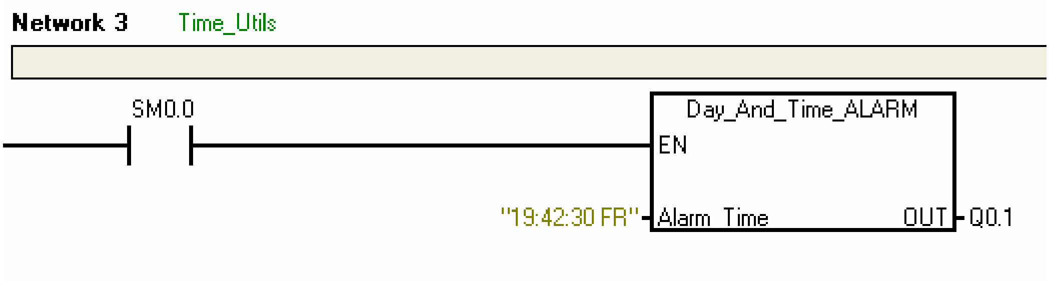


Image 5.6: Day_And_Time_ALARM function.

6. Nubitek_Math_Utils Library.

The library of mathematical and statistical utilities Nubitek_Math_Utils is composed of 5 functions:

atan

Calculates value introduced in input x in the output and rad arctangent. Both x and y are data in REAL format.

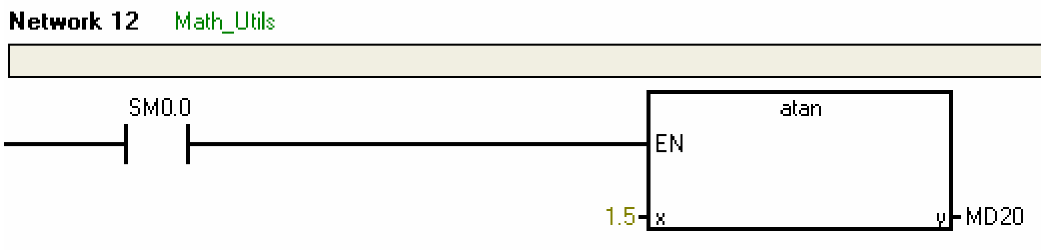


Image 6.1: atan function.

acos

Calculates value introduced in input x in the output and rad arccosine, the valid range thereof being between -1 and 1. Both x and y are data in REAL format.

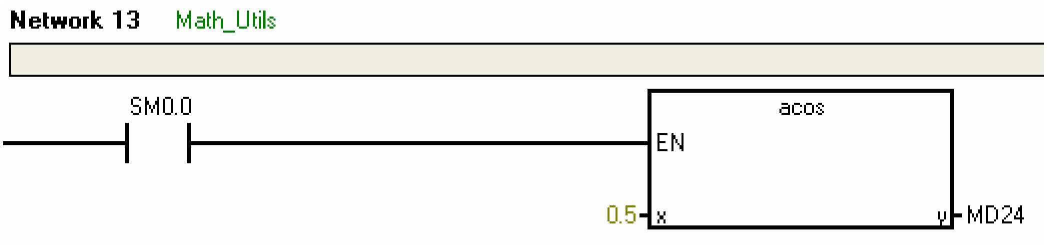


Image 6.1: acos function.

asin

Calculates value introduced in input x in the output and rad arcsine, the valid range thereof being between -1 and 1. Both `x` and `y` are data in REAL format.

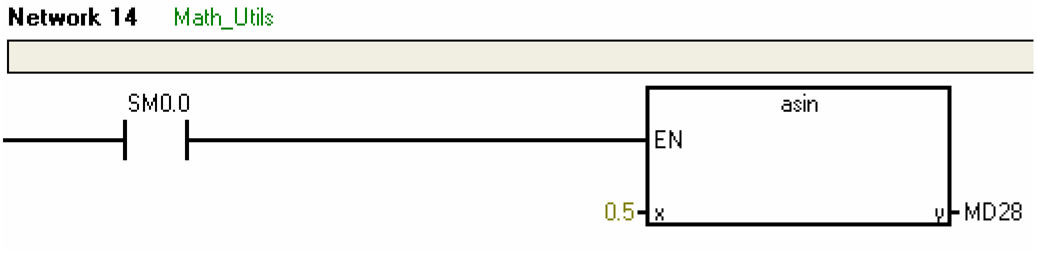


Image 6.3: asin function.

Mean_Variance

Calculates the mean and variance of a sequence of data introduced in the input IN, all thereof data in REAL format.

After each pulsation to the input `Calculate` the value in the input IN is added to the stored sequence and the output statistical calculations are updated, including the counter of stored samples, `N` (with DINT format).

The input `P_data` requires a V-memory position where samples can be stored (pointer &VBn format). **Note**: the length of the area occupied by the dynamic function depends on the number of samples stored according to the formula 16 + 4n bytes, where n is the number of samples (for instance, if you store 10 values, you need 56 bytes). This memory zone must not be used by the user.

The input `Reset` initializes the function and empties the stored list of values.

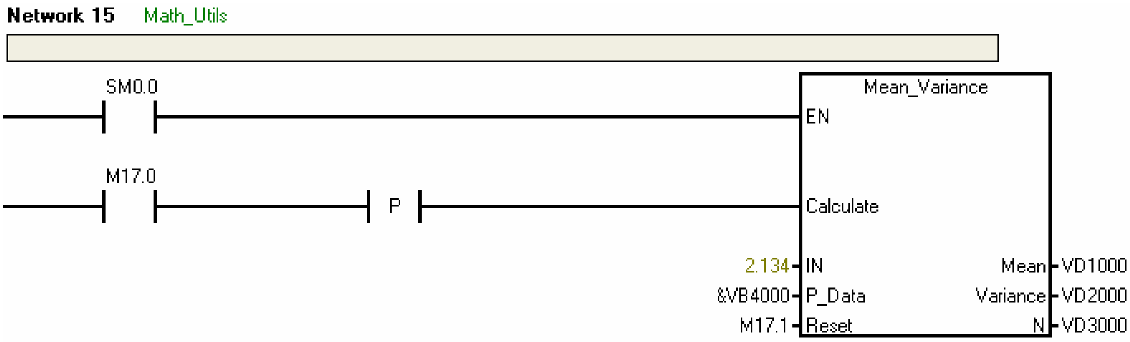


Image 6.4: Mean_Variance function.

Analog_Adapter

Converts the input of an analog sensor into a numeric value with physical sense, which can be more easily processed in the remainder of the control program. In the input parameter `Sensor_IN` you have to set the memory position in which the sensor has been cabled. The resulting scaling appears in the output `Value`, in REAL format.

The inputs MIN and MAX (both REAL type), which serve to specify the sensor's measurement range, are used to perform the scaling properly. For instance, 23.0 in MIN and 68.5 in MAX are used for a temperature sensor offering 0 – 10V for the 23 ºC to 68.5ºC range.

If the sensor is of the 4 20 mA type, it is noted marking a 1 in the input IN_4_20mA. For complete range sensors (0 – 10V, 0 – 20 mA) use a 0.

The output Error shows an error code (BYTE format) indicating:

- 0 -> No error.
- 1 -> Parameterization error: MIN>=MAX.
- 2 -> 4 20mA Sensors: the cable sensor might be broken (IN ~ 0mA)
- 4 -> 4 20mA Sensors: the sensor is not working properly (IN < 4mA)

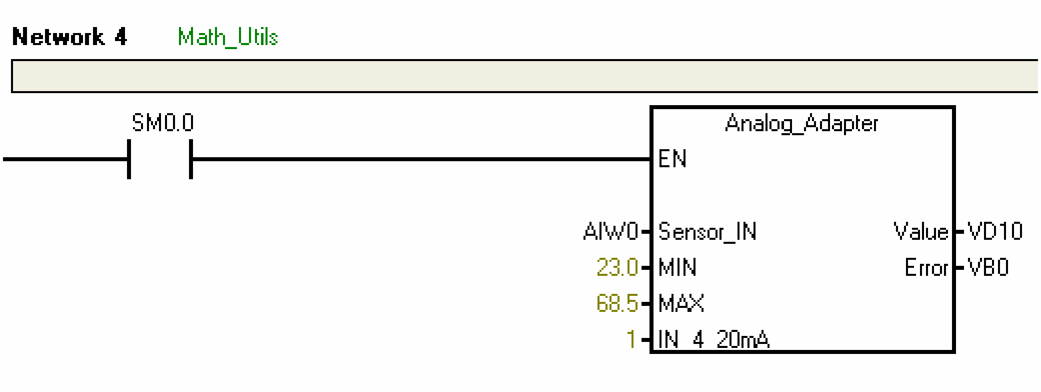


Image 6.5: Analog_Adapter function

7. Nubitek_String_Utils Library.

The utility library for Nubitek_String_Utils text chains consists of 4 functions:

Small_Letter

Converts into small letters all characters of the text, the start address of which is shown in the parameter IN (in pointer &VBn format).

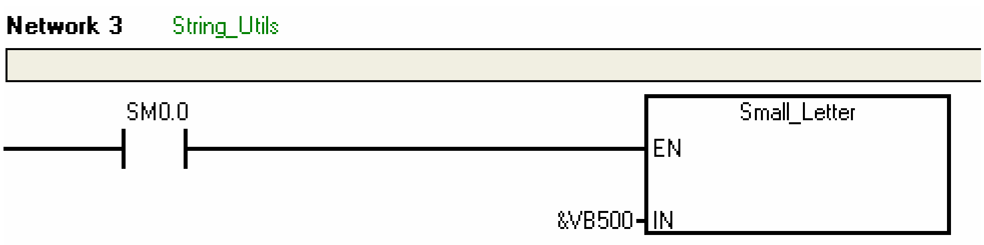


Image 4.1: Small_Letter function.

Capital_Letter

Converts into capital letters all characters of the text string, the start address of which is shown in the parameter `IN` (in pointer &VBn format).

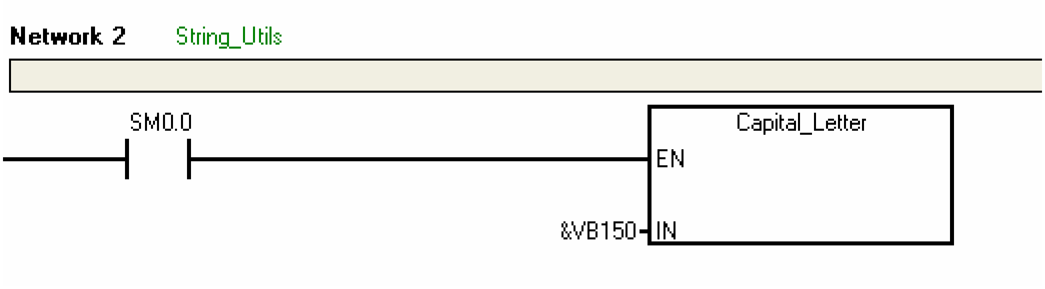


Image 4.2: Capital_Letter function.

Get_Value

Extracts the first numeric value found in the text string, the start address of which is shown in the parameter `IN` (in pointer &VBn format) and places it in the output variable `Value`, in REAL format.

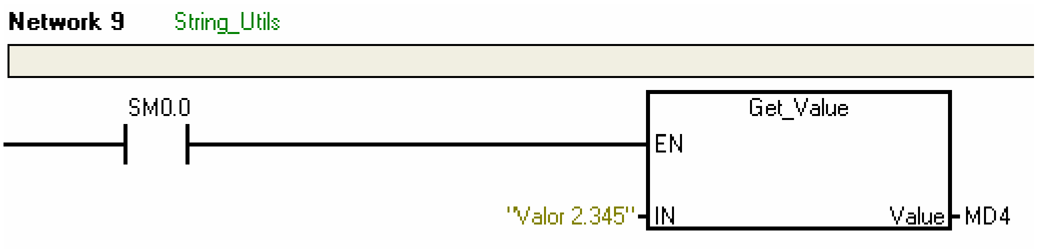


Image 4.3: Get_Value Function.

S_Compare

Compares the 2 text strings (STRING format) passed as parameter in `String_1` and `String_2` and activates the `Similar` output if equal.

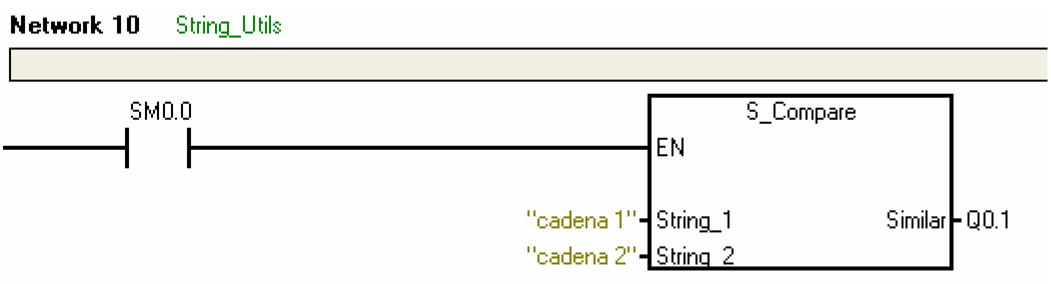


Image 4.4: S_Compare function.