



[www.nubitek.com](http://www.nubitek.com)

# **Nubitek SMS Manager**

**First Steps**

**Quick Reference Guide**

## Nubitek SMS Manager v1.3

### First steps. Quick Reference Guide.

Last updated: July 2008

© 2006 Nubitek industrial IT S.L.  
All rights reserved.

The information in this document belongs to Nubitek industrial IT, S.L. (hereinafter referred to as NUBITEK). Authorization is granted by NUBITEK only for 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 contents 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

<b>1. INTRODUCTION</b>	<b>4</b>
INTENDED AUDIENCE	4
STRUCTURE OF THE DOCUMENT	4
NOTATIONS USED IN THIS GUIDE	4
<b>2. REQUIRED EQUIPMENT</b>	<b>5</b>
<b>3. CONNECTING THE EQUIPMENT</b>	<b>5</b>
<b>4. INSTALLING THE SMS MANAGER LIBRARY</b>	<b>6</b>
<b>5. ASSIGNING MEMORY FOR THE LIBRARY</b>	<b>7</b>
<b>6. SENDING AN SMS</b>	<b>8</b>
<b>7. RECEIVING AN SMS</b>	<b>10</b>
<b>APPENDIX 1: NUBITEK SMS MANAGER PROGRAM FUNCTIONS</b>	<b>11</b>
1. SMS_MANAGER	11
2. SEND_SMS	11
3. RECEIVE_SMS	12
<b>APPENDIX 2: THE NUBITEK KEY AND THE TEST MODE</b>	<b>12</b>

## 1. Introduction

This first steps guide will help you quickly start using the Nubitek SMS Manager library. You will be shown how to begin sending and receiving SMS messages from your S7-200 in only a few minutes.

### Intended audience

- S7-200 developers that want to begin using the Nubitek SMS Manager library to send and receive SMS messages.
- Automation clients who are not experts S7-200 development, yet want to see how simple it is to use the Nubitek SMS Manager library.

### Structure of the document

Chapters 2 and 3 provide hardware requirements for the use of the library and explain how to connect the equipment.

Chapters 4 and 5 explain how to prepare the Step-7 MicroWIN software development environment for the use of the Nubitek SMS Manager library.

Chapter 6 explains step-by-step how to send a message from your S7-200 PLC to a mobile telephone using Nubitek SMS Manager.

Chapter 7 explains how to receive a message sent from a mobile telephone with your PLC S7-200 using Nubitek SMS Manager.

Appendix 1 is the basic reference for the Nubitek SMS Manager library and offers a description of the library's functions.

Appendix 2 provides additional information concerning the key required for the use of the Nubitek SMS Manager library.

### 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
<i>Italics</i>	Menu options Example: <i>File -&gt; Add or remove library</i>
<b>Bold</b>	Important notifications for the reader. Example: <b>Attention</b>
Monospaced	File names and directory paths. Example: <code>nubitek_sms_manager_v1.3.mwl</code>
	Text referring to program code. Example: VB2000 "Alarm message."

## 2. Required Equipment

The following hardware is required to use the Nubitek SMS Manager library:

- A S7-200 CPU. (CPU 224, version xxx23, firmware 2.0 or higher)
- A TC35i or TC65 GSM modem, with antenna and power supply.
- A SIM card from any mobile telephone operator.
- A PC/PPI Multimaster cable (ref. Siemens 6ES7 901 3CB30 0XA0).
- A 9 pin, male-male Sub-D adapter.

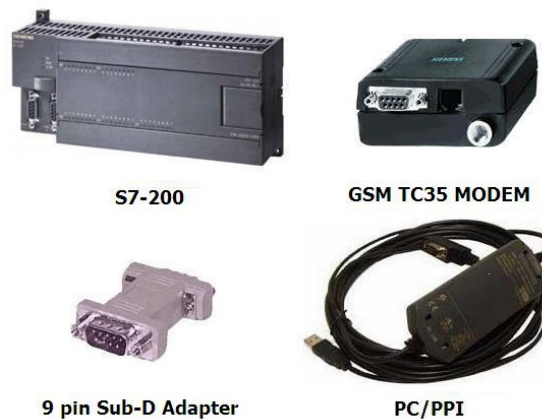


Image 2.1: Required hardware.

## 3. Connecting the Equipment

Follow the steps below to connect the equipment:

1. Insert the SIM card into the GSM modem. The slot for the SIM card is labeled on the modem as SIM and has a small yellow button next to it. Open the SIM card tray by pressing on it with a sharp object.
2. Connect the 9-pin, male-male Sub-D adapter to the female serial port on the GSM modem.
3. Connect the antenna and the power supply to the modem (in the RJ11 jack).
4. Configure the PC/PPI cable: the PC/PPI cable microswitches should be set to support a transmission rate of 9600 baud, 10 bit format, PPI/Freeport mode and remote connection:

Microswitch	1	2	3	4	5	6	7	8
Position	0	1	0	0	0	1	1	0

5. Connect the serial end of the PC/PPI cable (female) to the modem using the 9 pin, male-male adapter. The PPI end (male) should be connected to Port 0 of the S7-200 PLC.
6. Turn on power to both devices. In some versions of TC65, you must press and hold the On/Off button for a second.

## 4. Installing the SMS Manager Library

To use the Nubitek SMS Manager tool, you must first install the library in the STEP-7 MicroWIN programming environment.

- Download the Nubitek SMS Manager library from the Nubitek website ([www.nubitek.com](http://www.nubitek.com)) downloads page.
- Open STEP7-MicroWIN (STEP7 MicroWIN v4.0 SP2 or higher is required)
- From the menu bar, click on *File -> Add or remove library...*

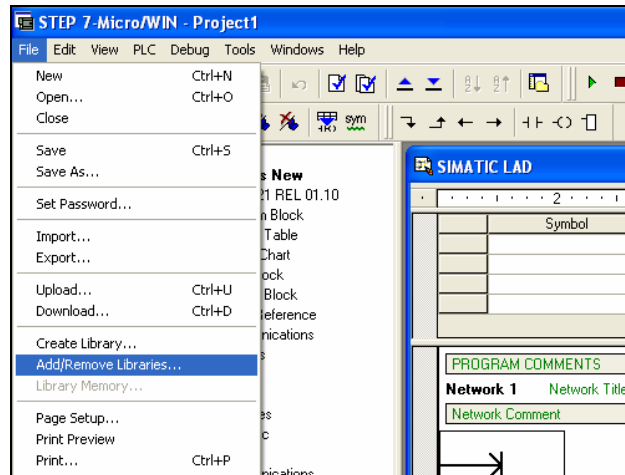


Image 4.1: Add library

- 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*.

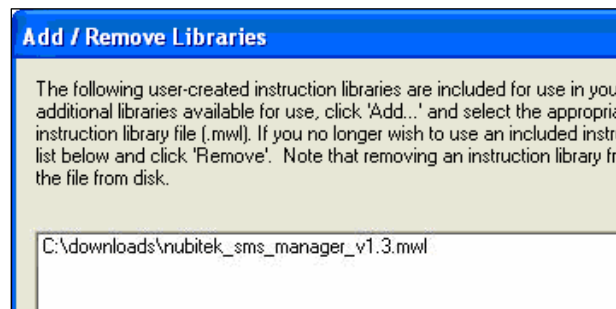


Image 4.2: Add library

- 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

## 5. Assigning Memory for the Library

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: its internal management requires 771 bytes of V memory, whose location can be defined by the user.

- Click on *File->Library memory*.

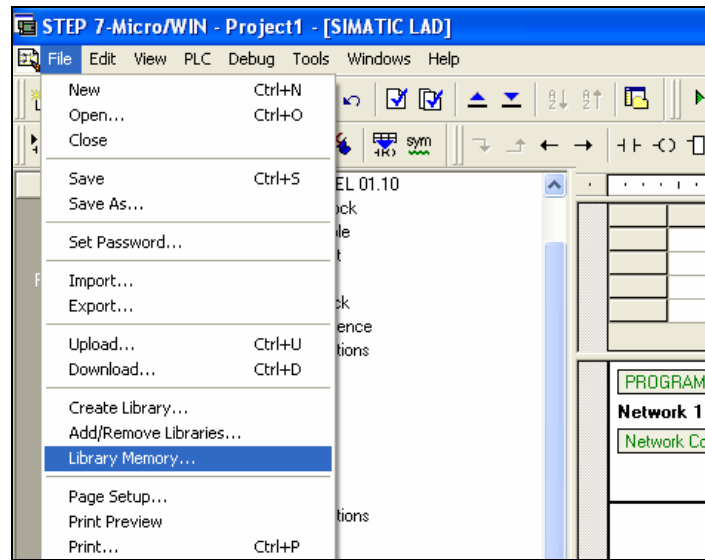


Image 5.1: Library memory

- In the window that opens, select the Nubitek\_SMS\_Manager 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*.

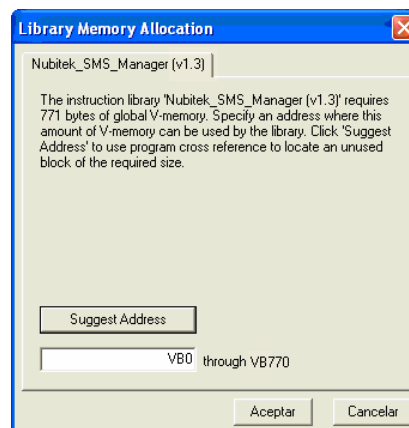


Image 5.2: Assign memory for the library

## 6. Sending an SMS

The example below shows a very quick and direct way to send an SMS. The following steps should be performed:

1. Prepare and connect the equipment according to the indications in Chapters 2 and 3 of this guide. Make sure that you have the Nubitek\_SMS\_Manager library installed, as indicated in Chapter 4.
2. In the Program Block, call the `SMS_Manager` function: it is found in the *libraries* folder at the bottom of the MicroWIN navigation tree. Use `SM0.0` enabling contact to have it executed all of the program cycles. Insert the following parameters:
  - `PIN`: the PIN for the SIM card, in double quotation marks. For example, `"2244"`.
  - `Nubitek_Key`: without quotation marks write `0`, for it is the key for test mode. If you already have a Nubitek key for your modem you may also use it now. For more information concerning the Nubitek key, see Appendix 2 of this guide.
  - `Ready`: write `Q0.0`. This way you will know when system initialization is complete and are able to begin sending and receiving SMS messages.

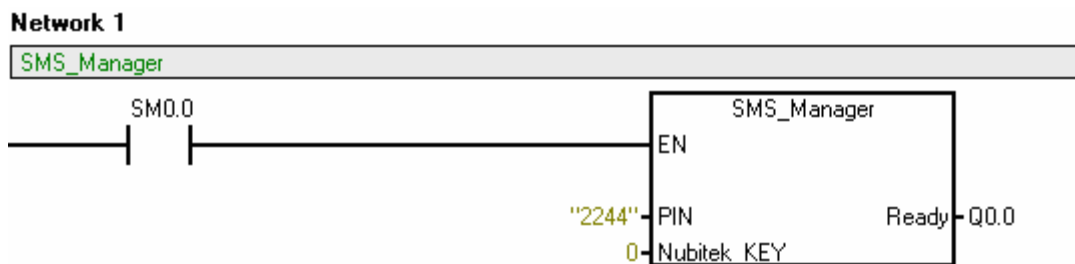


Image 6.1: SMS\_Manager function call

3. Assign memory to the library as pointed out in Chapter 5. For this example, you can use the default range, from `VB0` to `VB770`.

Up until this point, you have defined a base configuration: the general control is ready to manage the sending and receiving of messages. All applications require that you perform the previous 3 steps.

4. Define the text for the SMS as text strings in the Data Block (**Attention:** the STRING format uses double quotation marks). Write the text for the message you would like to send in the data block, for example:
  - `VB2000 "ALARM Message: installation STOPPED"`

It is recommended that you assign symbolic names to the SMS to improve the understanding of the program, so that if `VB2000` → `SMS_Alarm`, the Data Block reads:

- `SMS_Alarm "ALARM Message: installation STOPPED"`

The text should not exceed 160 characters, although you can include as many different SMS messages as the V memory space will allow.



5. Define the telephone numbers for the SMS recipient in the Data Block, just as you did with the text. Indicate the telephone number of the telephone where you would like to receive the SMS, including the country code, for example:

- VB2100 "+34666777888"

You should also assign symbolic names for telephone numbers, e.g. VB2100 → TEL\_David:

- TEL\_David "+34666777888"

Telephone numbers may require the area code or country code preceded by the + sign, although this is not always necessary: it depends on the country and the mobile telephone operator. You can also include as many different telephone numbers as the V memory space will allow.

6. Call the `Send_SMS` function. The call must be made in a way that the condition to send an SMS is only executed once: therefore use the `I0.0` and positive edge contacts. Introduce the following parameters:

- SMS: &VB2000
- Tel\_1: &VB2100
- Tel\_2: not used in this example, enter 0 for this parameter
- Value: not used in this example, enter 0 for this parameter

#### Network 2

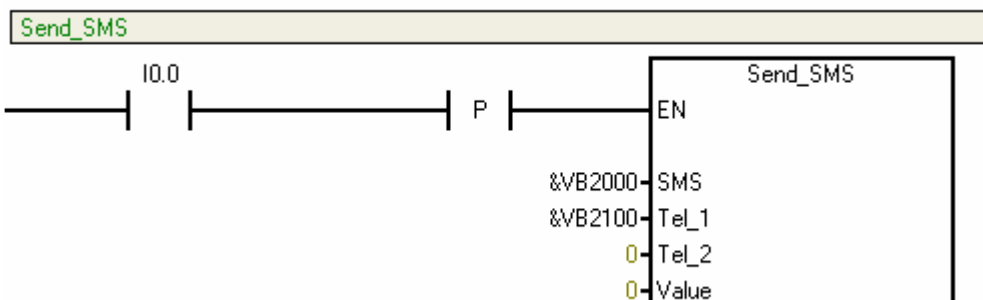


Image 6.2: Send\_SMS function call

If the symbol `I0.0` → `Failure` is assigned, you will see that the segment is very clearly understood: "when a failure occurs, send the alarm message to David's telephone".

7. Upload the program and set the CPU to RUN:
  - It should take less than one minute to activate the output `Q0.0`, which indicates that the system is ready to operate.
  - Activate the input `I0.0`. In a few seconds, you will receive the SMS you defined in the Data Block.

## 7. Receiving an SMS

This example shows how to receive an SMS. If you already tried the example from Chapter 6 (Sending an SMS), you can use that project as a base and go directly to number 2.

1. Define the base configuration of the applications as indicated in the first 3 steps of Chapter 6.
2. Call the `Receive_SMS` function. Use `SM0.0` enabling contact to have it executed all of the program cycles. Introduce the following parameters:
  - `TXT_1`: first word to be recognized, in double quotation marks. E. g. "Activate"
  - `TXT_2`: second word to be recognized, in double quotation marks. E. g. "machine"
  - `Value`: not used in this example, enter an unused memory position, for example `MD10`
  - `SMS_Rcv`: is the output activated when a message corresponding to a defined pattern is received. Enter, for example, `M0.0`.

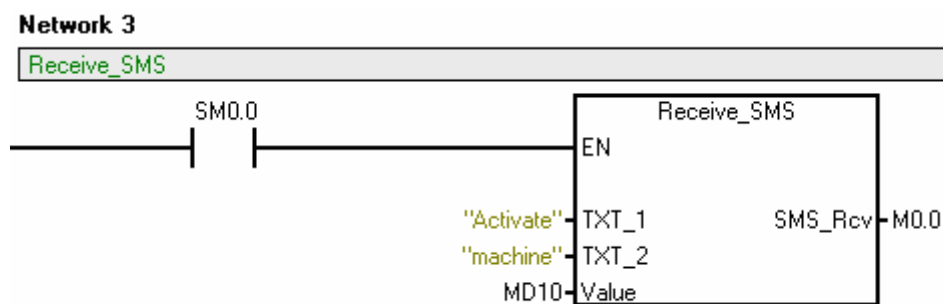


Image 7.1: Receive\_SMS function call

3. Bit `M0.0` only activates a program cycle when it receives the message. To easily observe this, we can take it to the output `Q0.1`. Add a segment as shown in the figure.

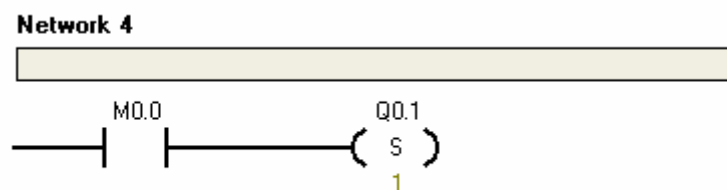


Image 7.2: Support segment

4. Upload the program and set the CPU to RUN:
  - It should take less than one minute to activate the output `Q0.0`, which indicates that the system is ready to operate.
  - Send a message with the text "activate machine", or "Machine activate" or "Must activate the machine" to the telephone number of the SIM card in the GSM modem. Any of these examples will activate the output `Q0.1`.

## Appendix 1: Nubitek SMS Manager Program Functions

The SMS Manager library is composed of 3 functions or subroutines: `SMS_Manager`, `Send_SMS` and `Receive_SMS`.

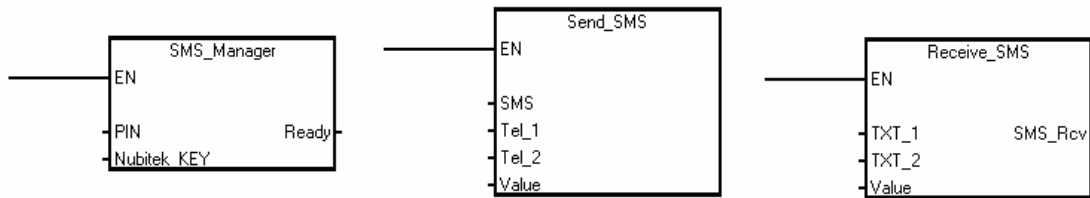


Image A1.1: `SMS_Manager`, `Send_SMS` and `Receive_SMS`

### 1. SMS\_Manager

This is the main management block of the library. It should be called one time in each program cycle, that is, using the `SM0.0` contact. It has 2 inputs and one output.

- **PIN:** STRING input. Text string with the PIN code of the SIM card inserted into the GSM modem. E.g.: "1234". If the SIM card does not have a PIN code, it should be indicated with the empty string "".
- **Nubitek\_KEY:** INT input. Numeric key provided by Nubitek for the correct functioning of the library. To work in test mode the value 0 must be set. Appendix 2 provides additional details and explains how to acquire the key.
- **Ready:** BOOL output. It is activated when the library has finished the initialization process and is ready to send messages.

### 2. Send\_SMS

The `Send_SMS` function is used once for each different SMS you wish to send.

You should call the function only one program cycle when the condition to send an SMS is produced. Therefore it is advisable to attack the enable input (EN) with the trigger condition and a positive edge.

This function has 4 input parameters, all of which are POINTER parameters, meaning that the prefix `&` should be used.

- **SMS:** DWORD input. Pointer to the text string to be sent. For example, `&VB2000`, or in symbolic format, `&SMS_Alarm`.
- **Tel\_1:** DWORD input. Pointer to the text string containing the recipient telephone number. For example, `&VB2100`, or in symbolic format, `&TEL_David`
- **Tel\_2:** DWORD input. Pointer to the text string containing the second recipient telephone number. This parameter is OPTIONAL: if you do not wish to send the SMS to another telephone number you only need to enter 0 for this input.
- **Value:** DWORD input. Pointer to the numeric variable to be inserted into the SMS text. This parameter is OPTIONAL: if you are not going to include an input value, Value should be 0. Consult the user manual for additional information concerning the inclusion of variables.

### 3. Receive\_SMS

Receive\_SMS is used once for each different incoming message you want to recognize. The function activates the SMS\_Rcv output for one program cycle upon receiving an SMS containing the key words defined in TXT 1 and 2.

The function should be called in each program cycle using the SM0.0 contact. It has 3 input parameters and 1 output parameter.

- **TXT\_1:** STRING input. Text string with the word we expect to find in the incoming SMS.
- **TXT\_2:** STRING input. Text string with another word we expect to find in the incoming SMS together with the `Txt_1` string. This parameter is OPTIONAL: if this parameter is not needed, just introduce an empty text string ("").
- **SMS\_Rcv:** BOOL output. It is activated during a program cycle when the incoming SMS includes the text strings defined in `TXT_1` and `TXT_2`, regardless of the order in which they appear and without discriminating between upper and lower case letters.
- **Value:** REAL input-output. If when `Receive_SMS` is activated, `SMS_Rcv` also detects a numeric value in the incoming SMS, either real or integer, the value is written in this variable in real format. Consult the user manual for more information concerning the capture of variables.

## Appendix 2: The Nubitek Key and the test mode

The Nubitek key is required to use the library. It must be inserted into the program as the incoming parameter `Nubitek_KEY` to the main function, `SMS_Manager`. It may be easily acquired in the Nubitek web site [www.nubitek.com](http://www.nubitek.com).

This key allows the library to work for an unlimited period of time but exclusively with a specific modem. To acquire a key you must indicate the IMEI of your modem: it is the 15-digit identifier found on the sticker on the back of the modem.

**Test mode:** Nubitek allows the user to test `Nubitek_SMS_Manager` and all of its features for **free**: just use the value 0 for the `Nubitek_KEY` parameter. Every 5 hours the application will be blocked and the modem will be switched off, but you may continue your testing by turning on the modem and restarting the CPU.