SoMachine Basic Example Guide

SMS with GSM Modem SR2MOD03

11/2014

xSample_SMS_Send_Receive_Modem.smbe

EIO0000002088.00

www.schneider-electric.com

Schneider Electric

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, can result in death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, can result in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and the installation, and has received the safety training necessary to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This document describes a SoMachine Basic template example that allows you to send and receive SMS messages using the GSM Modem SR2MOD03 and a M221 Logic Controller via serial line communication.

The example described in this document is intended for learning purpose only; it must not be used directly on products that are part of a machine or process.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not include the code from this example in your machine or process without thoroughly testing your entire application.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This document and its related SoMachine Basic project file focus on specific functions and function blocks of the Schneider Electric libraries and on specific features of SoMachine Basic. They are intended to help you understand how to develop, test, commission, and integrate applicative software of your own design in your control systems.

The example is intended for qualified persons who are new SoMachine Basic users.

Validity Note

This document is valid for SoMachine Basic V1.3 or later.

Product Related Information

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.[1]
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

Introduction

Before You Begin

Your specific application requirements may be different from those assumed for this and any related examples described herein. In that case, you will have to adapt the information provided in this and other related documents to your particular needs. To do so, you will need to consult the specific product documentation of the hardware and/or software components that you may add or substitute for any examples specified in this documentation. Pay particular attention and conform to any safety information, different electrical requirements and normative standards that would apply to your adaptation.

⚠ WARNING

REGULATORY INCOMPATIBILITY

Be sure that all equipment applied and systems designed comply with all applicable local, regional and national regulations and standards.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only you, the user, machine builder or system integrator, can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, you must also consider any applicable local, regional or national standards and/or regulations.

Some of the major software functions and/or hardware components used in the proposed architectures and examples described in this document cannot be substituted without significantly compromising the performance of your application. Further, any such substitutions or alterations may completely invalidate any proposed architectures, descriptions, examples, instructions, wiring diagrams and/or compatibilities between the various hardware components and software functions specified herein and in related documentation. You must be aware of the consequences of any modifications, additions or substitutions.

A residual risk, as defined by EN/ISO 12100-1, Article 5, will remain if:

- it is necessary to modify the recommended logic and if the added or modified components are not properly integrated in the control circuit.
- you do not follow the required standards applicable to the operation of the machine, or if the adjustments to and the maintenance of the machine are not properly made (it is essential to strictly follow the prescribed machine maintenance schedule).
- the devices connected to any safety outputs do not have mechanically-linked contacts.

⚠ CAUTION

EQUIPMENT INCOMPATIBILITY

Read and thoroughly understand all device and software documentation before attempting any component substitutions or other changes related to the application examples provided in the document.

Failure to follow these instructions can result in injury or equipment damage.

Start-Up and Test

Before using electrical control and automation equipment after design and installation, the application and associated functional safety system must be subjected to a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such testing be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ CAUTION

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in injury or equipment damage.

Verify that the completed system, including the functional safety system, is free from all short circuits and temporary grounds, except those grounds installed according to local regulations. If temporary high-potential voltage testing is necessary, follow the recommendations in equipment documentation to help prevent injury or equipment damage.

Operations and Adjustments

Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly installed and operated.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the hands and other parts of the body are free to enter the pinch points or other hazardous areas where serious injury can occur. Software products alone cannot protect an operator from injury. For this reason, the software cannot be substituted for or take the place of point-of-operation protection.

⚠ WARNING

UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY

- Do not use this software and related automation equipment on equipment which does not have point-of -operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the examples and implementations suggested herein. It is sometimes possible to adjust the equipment incorrectly and this may produce unsatisfactory or unsafe operation. Always use the manufacturer instructions as a guide to functional adjustments. Personnel who have access to these adjustments must be familiar with the equipment manufacturer instructions and the machinery used with the electrical equipment.

Only those operational adjustments actually required by the machine operator should be accessible to the operator. Access to other controls should be restricted to help prevent unauthorized changes in operating characteristics.

Example Description

Overview

This template example helps you to program SMS (Short Message Service) management using the SR2MOD03 GSM Modem and a M221 Logic Controller.

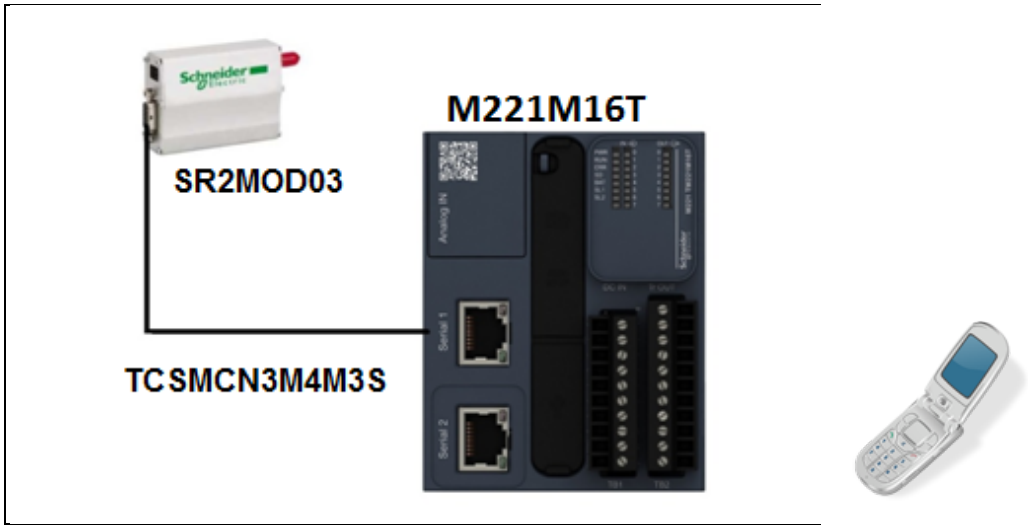
Description

The SoMachine Basic template example contains the following:

- GSM Modem (SR2MOD03) configuration
- SMS send
- SMS receive
- Controlled SMS reception via White List
- Write a register by SMS
- Set/Reset a coil by SMS
- Verify SIM card PIN
- Verify GSM Network registration
- Verify GSM signal level

Hardware configuration

The SoMachine Basic template example corresponds to the hardware implementation as follows:

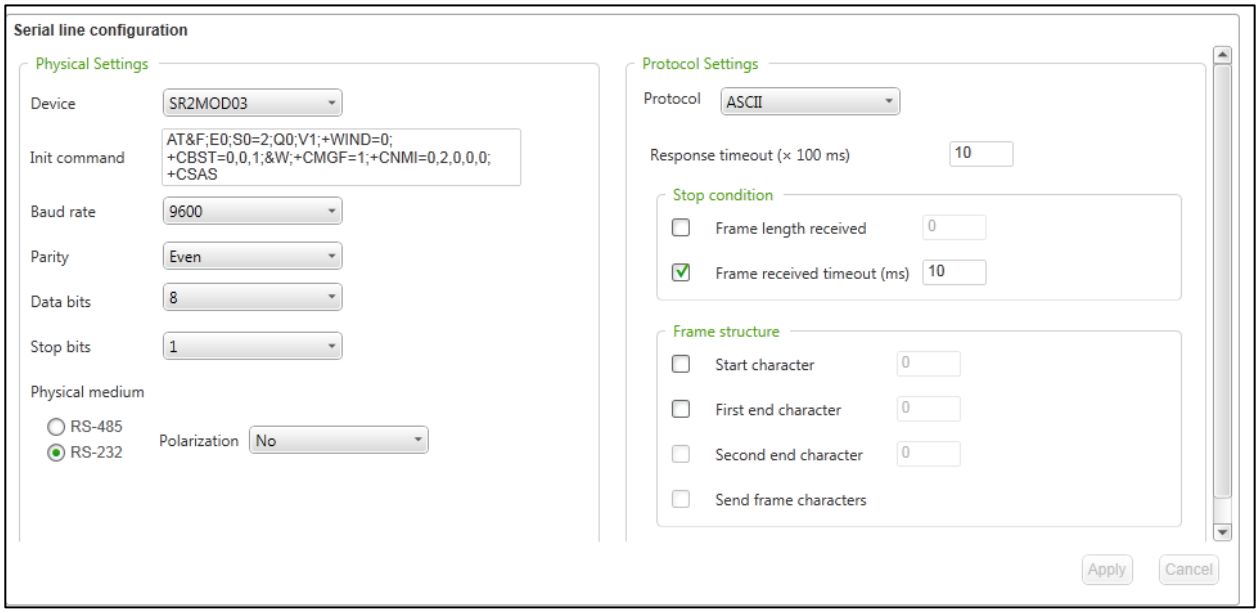


The hardware configuration is made of:

- 1 TCSMCN3M4M3S2 RJ45 DB9 Male RS232 Straight Cable
- 1 SR2MOD03 GSM Modem
- 1 M221 Logic Controller (any reference, but the template example is based on M221M16T)
- 1 mobile phone

Modem Configuration

The example template uses the following configuration. Verify the configuration in SoMachine Basic. Under the Configuration tab, select the `SL1 (Serial line)` port and verify that the parameters are set as follows:



Project Information

This SoMachine Basic template example is composed of five parts:

- Initialization sequence
- Incoming SMS verification
- SMS command reception
- Alarm SMS
- Tools to send requests

One of the key features of this template example is that you can send commands with a mobile phone by SMS to the M221 Logic Controller.

The M221 Logic Controller can as well send an alarm SMS to a mobile phone.

Program Organization

Initialization Sequence

On the first cycle of the logic controller scan (%S13 is set to 1), the Initialization sequence starts. You can also start the Initialization sequence manually by setting %M54 to 1.

The Initialization sequence is made of 7 steps.

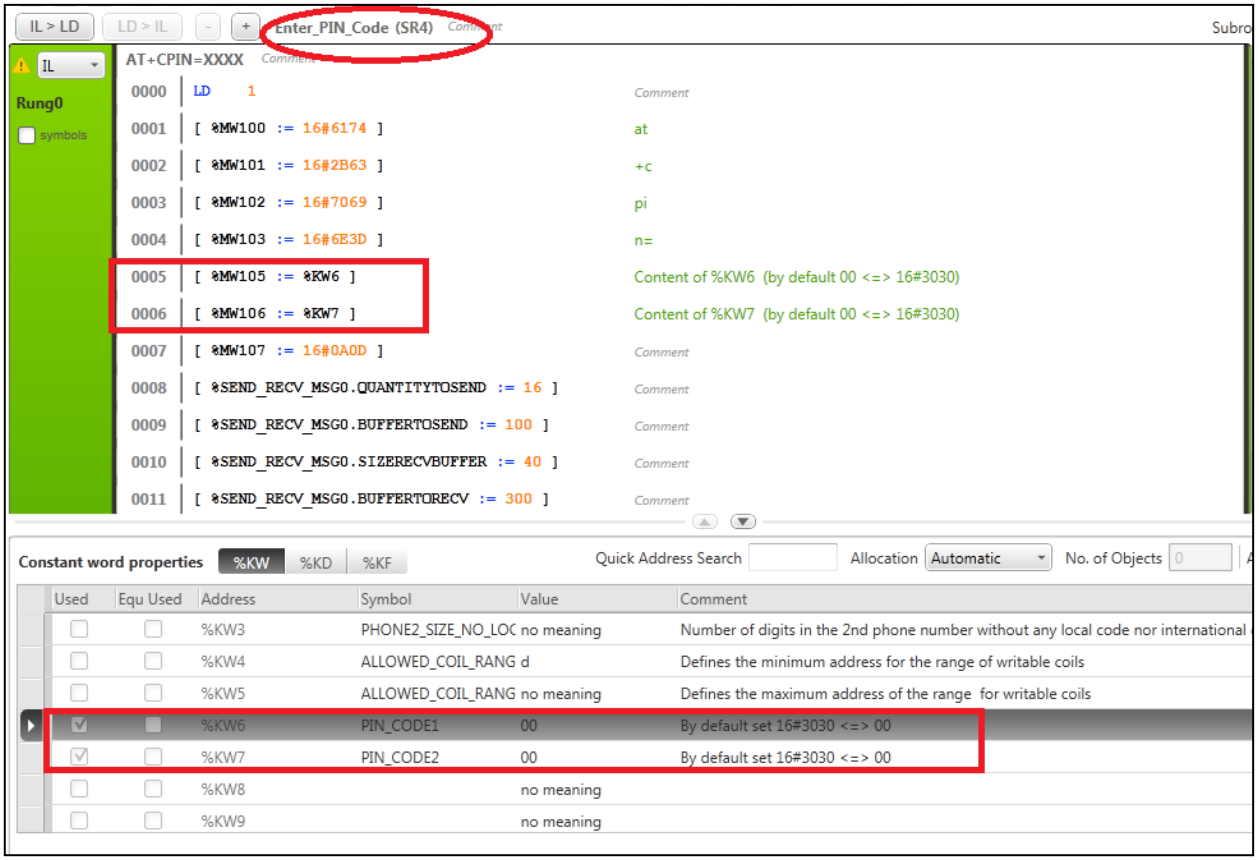
Step	Name	Description
1	GET_Signal_Quality	Verifies the quality of the reception signal
2	GET_PIN_Code_Status	Verifies whether the PIN code status is set to READY
3	GET_Network_Status	Verifies whether the modem is registred on the network
4	SET_RX_Mode	Configures the modem to receive SMS on explicit request
5	SET_Text_Mode	Sets the Text Format for SMS
6	SET_enhanced_code	Sets the answer codes detail option
7	FLUSH_SMS_Memory	Clears all existing SMS messages in the logic controller

If a step is not completed, the sequence is aborted. You can start this sequence again thereafter.

NOTE: Step 2 will only retrieve a status for the SIM Card in the modem. If the SIM Card is not READY, the Initialization sequence is aborted.

Enter PIN Code

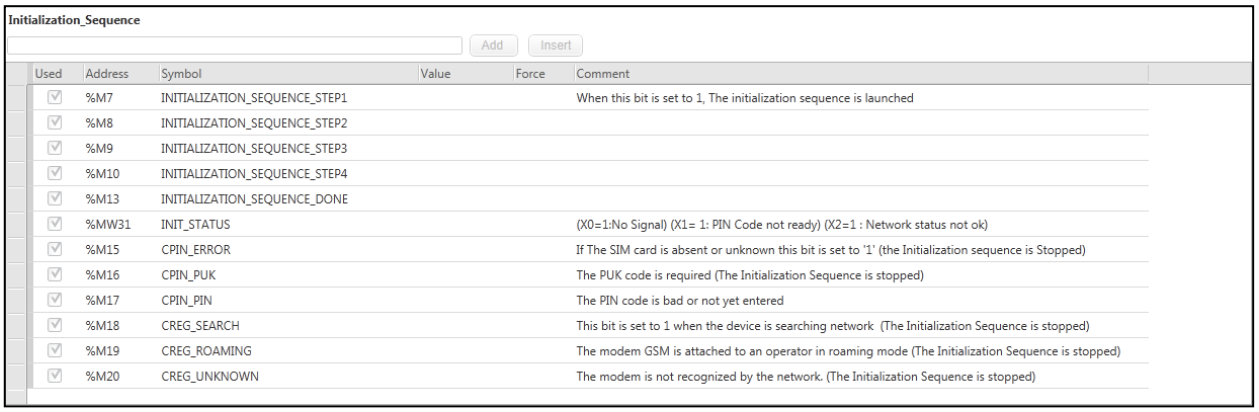
You need to adapt the template example for your specific PIN code. The PIN code is contained as ASCII characters in two constant words. Modify the value of the constant words as shown below:



Setting the memory bit %M59 to 1 sends the PIN code to the modem.

Status and Animation Table

You can monitor the Initialization sequence using the `Initialization_Sequence` animation table:



Initialization_Sequence

Add Insert

Used	Address	Symbol	Value	Force	Comment
☑	%M7	INITIALIZATION_SEQUENCE_STEP1			When this bit is set to 1, The initialization sequence is launched
☑	%M8	INITIALIZATION_SEQUENCE_STEP2			
☑	%M9	INITIALIZATION_SEQUENCE_STEP3			
☑	%M10	INITIALIZATION_SEQUENCE_STEP4			
☑	%M13	INITIALIZATION_SEQUENCE_DONE			
☑	%MW31	INIT_STATUS			(X0=1:No Signal) (X1= 1: PIN Code not ready) (X2=1 : Network status not ok)
☑	%M15	CPIN_ERROR			If The SIM card is absent or unknown this bit is set to '1' (the Initialization sequence is Stopped)
☑	%M16	CPIN_PUK			The PUK code is required (The Initialization Sequence is stopped)
☑	%M17	CPIN_PIN			The PIN code is bad or not yet entered
☑	%M18	CREG_SEARCH			This bit is set to 1 when the device is searching network (The Initialization Sequence is stopped)
☑	%M19	CREG_ROAMING			The modem GSM is attached to an operator in roaming mode (The Initialization Sequence is stopped)
☑	%M20	CREG_UNKNOWN			The modem is not recognized by the network. (The Initialization Sequence is stopped)

Sending an SMS

Once the Initialization sequence is done, you are ready to send an SMS. The Initialization sequence is done when %M13 is set to 1.

The telephone number of the mobile phone that you want to send an SMS message to is encoded in memory words (%MW) found in the subroutine WRITE_SMS (SR13). The number encoded in the template example is a fictitious number, and it needs to be modified.

In the subroutine WRITE_SMS (SR13), %MW100 through %MW111 are used to contain the dialing command and telephone number. Represented in hexadecimal in the graphic below, the ASCII representation is:

AT+CMGS="+33012334567"

Modify the number, retaining all other characters including the quotation marks. Then, adjust the number of characters to be sent to the modem, contained in %SEND_RECV_MSG0.QUANTITYTOSEND (set to 24 in the template example).

NOTE: A carriage return and line feed sequence must follow the end of the string (16#0D0A) and is included in the quantity to send.

For example, you may wish to dial a local number, such as:

AT+CMGS="5551234"

Then, set the quantity to send value to 19.

You then need to configure the message you want to send. This is done by modifying the subroutine `Write_SMS_Content (SR14)`.

For the purposes of the template example, the value of a word (`%MW211`) is incremented to simulate an alarm value. Then, the value of the word is converted into ASCII using `Convert_INT To ASCII (SR9)` into two memory words, `%MW218` and `%MW219`, which are used for the simulated alarm message to send.

You can adjust the above example to suit your needs.

Once you have established the telephone number and the message, you can send the SMS by setting the memory bit `%M55` to 1.

Receiving an SMS

Once the Initialization sequence is done, you are ready to receive an SMS. The Initialization sequence is done when memory bit %M13 is set to 1.

Working with SMS Messages

The template example includes programming code that parses and interprets incoming SMS messages for pre-defined commands. Commands are defined as instructions that the would-be application acts upon, and three such variable commands are contrived here for the purposes of the template example.

The commands are extracted from the SMS message as a string of ASCII characters. The strings are converted to text and numeric values, using the information associated with the command definition, as you can see in the following table of constant words (%KW):

Address	Symbol	Value	Comment
%KW28	VAR_QUANTITY	3	This the amount of defined variables
%KW29	VAR_STRUCT_SIZE	5	This values represents the size of each variable structure (Type, Index, value1,value2,value3 = 5 Words)
%KW30	TYPE_VAR_VAR01	1	Type of variable is defined in this Constant Word : %KW30 = 1 : %M , %KW30=2 : %MW
%KW31	VAR01_INDEX	100	The variable VAR01 (named PUMP01) will be mapped to %M100 Or %MW100 (depending on VAR TYPE)
%KW32	VAR01_01	20565	PU
%KW33	VAR01_02	19792	MP
%KW34	VAR01_03	12337	01
%KW35	TYPE_VAR_VAR02	1	Type of variable is defined in this Constant Word : %KW30 = 1 : %M , %KW30=2 : %MW. Here VAR02 is a %
%KW36	VAR02_INDEX	101	The variable VAR02 (named VALVE1) will be mapped to %M101 Or %MW101 (depending on VAR TYPE)
%KW37	VAR02_01	22081	VA
%KW38	VAR02_02	19542	LV
%KW39	VAR02_03	17713	E1
%KW40	TYPE_VAR_VAR03	2	Here VAR03 is a %MW type.Type of variable is defined in this Constant Word : %KW30 = 1 : %M , %KW30=
%KW41	VAR03_INDEX	1995	The variable VAR03 (named GAIN01) will be mapped to %MW1001 (depending on VAR TYPE)
%KW42	VAR03_01	18241	GA
%KW43	VAR03_02	18766	IN
%KW44	VAR03_03	12337	01
%KW45		0	

For the template example, there are three six-character commands (two characters per constant word) defined. The syntax of the incoming SMS command is as follows:

```
<command>=<val>
```

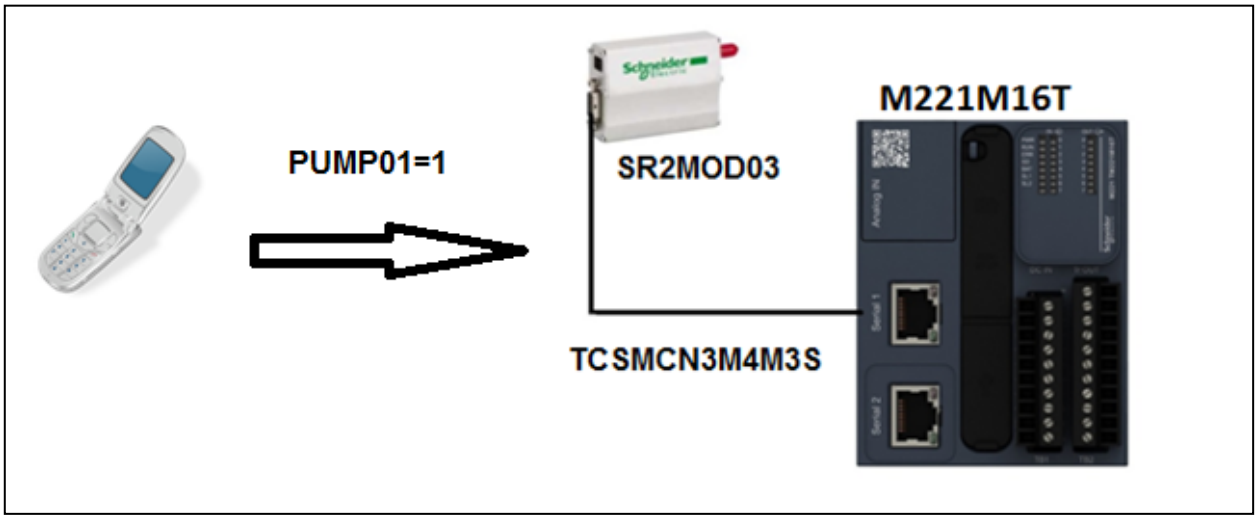
where

<command> is one of the three commands found in the command table above,

and

<val> is the numeric value you associate with the command.

The <val> is converted from the SMS message from a text to a numeric value, and stored in a memory bit or word (%M or %MW) depending on its type as seen in the command table above.

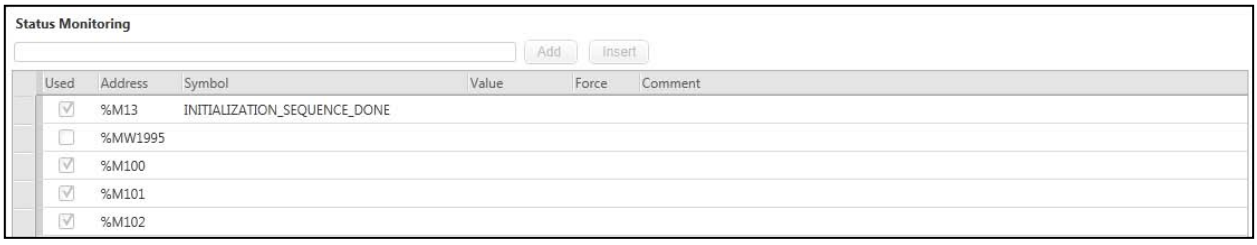


For example, you may send an SMS message to the GSM modem containing the command `PUMP01=0`, as represented in the graphic above.

%KW30	TYPE_VAR_VAR01	1	Type of variable is defined in this Constant Word : %KW30 = 1 : %M , %KW30=2 : %MW
%KW31	VAR01_INDEX	100	The variable VAR01 (named PUMP01) will be mapped to %M100 Or %MW100 (depending on VAR TYPE)
%KW32	VAR01_01	20565	PU
%KW33	VAR01_02	19792	MP
%KW34	VAR01_03	12337	01

The command is defined as type 1, meaning that what follows the equal sign is a bit value. Further, the index for the command is 100, meaning that the converted value will be stored in memory bit `%M100`.

You can monitor the effect of the SMS command using the `Status Monitoring` animation table:



In order to better understand the interpreter that parses an SMS into a command, you can monitor the different values using `Interpreter_Status` animation table:

Interpreter_Status

Add Insert

Used	Address	Symbol	Value	Force	Comment
☑	%M31	NO_SMS_RX_BUFFER			The read SMS request did not return SMS
☑	%M22	SMS_INTERPRETER_CHECK_STATUS			when this bit is set to 1, the first SMS check step is done (Is SMS present ? Is the SMS Unread ?)
☑	%M48	END_OF_INTERPRETER			This bit is set to 1 at the end of the interpreter sequence
☑	%M30	SMS_INTERPRETER_ON			This bit is set to 1 during SMS Interpreter action. During the interpretation, all RX request are blocked (p...
☑	%M32	SMS_IN_RX_BUFFER			There is a SMS in the RX Buffer
☑	%M33	SMS_ALREADY_READ			The SMS has been previously read => we should delete it
☑	%M34	SMS_UNREAD			The SMS in the RX buffer is UNREAD => we should manage it
☑	%M35	SMS_INTERPRETER_COMMA_MSB			this bit is set to 1 when we detect that the first comma after the phone number is on msb of the word
☑	%M36	SMS_INTERPRETER_COMMA_LSB			this bit is set to 1 when we detect that the first comma after the phone number is on lsb of the word
☑	%M37	SMS_INTERPRETER_PHONE_COMMA_ONGO			this bit is set to 1 during the search of the first comma after the phone number
☑	%M38	SMS_INTERPRETER_PHONE1_NOK			phone of rx SMS doesn't match with 1st allowed phone
☑	%M40	SMS_INTERPRETER_VARTYPE_COIL			this bit is set to 1 when the SMS symbol refers to a coil variable (%M)
☑	%M41	SMS_INTERPRETER_VARTYPE_REGISTER			this bit is set to 1 when the SMS symbol refers to a register variable (%MW)
☑	%M42	SMS_INTERPRETER_SET_COMMAND			this bit is set to 1 when a set coil command is detected in the SMS
☑	%M43	SMS_INTERPRETER_RESET_COMMAND			this bit is set to 1 when a reset coil command is detected in the SMS
☑	%M44	SMS_INTERPRETER_COIL_COMMAND_NOK			this bit is set to 1 when the command is not ok (for example value out of range)
☑	%M45	ASCII_TO_INT_CONV_DONE			this bit is set to 1 at the end of the ascii to int convertion process
☑	%M46	SMS_INTERPRETER_REGISTER_CMD_OK			this bit is set to 1 if the SMS register command is OK
☑	%M47	SMS_INTERPRETER_REGISTER_CMD_NOK			this bit is set to 1 if the SMS register command is not OK

White List

The template example also features a technique, referred to as a White List, to add security to your application using the GSM modem. The White List is a list of phone numbers that are authorized to transmit SMS messages to your application. When a call and subsequent message is transmitted to your application via the modem, the originating phone number is validated before acting upon the incoming message.

The White list is contained in constant words (`%KW`). The phone numbers start at `%KW10`:

☐	%KW10	PHONE_1_01	06	First TWO digits of the 1st allowed phone number
☐	%KW11	PHONE_1_02	01	
☐	%KW12	PHONE_1_03	02	
☐	%KW13	PHONE_1_04	03	
☐	%KW14	PHONE_1_05	04	

Use `%KW0` to set the length of the phone number and `%KW2` for the number of digits of the phone number (without international Code digits nor local code digits).

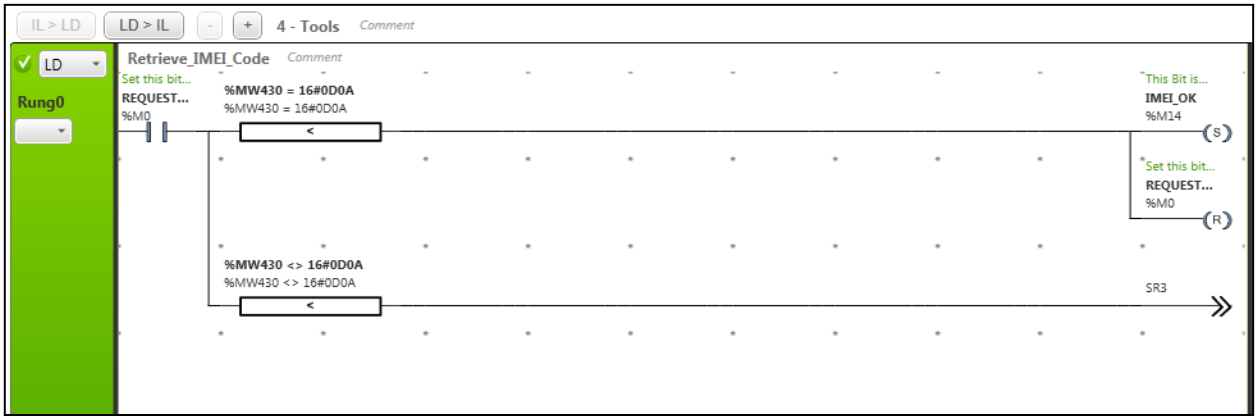
Address	Symbol	Value	Comment
%KW0	PHONE1_DIGITS_QUANTITY	10	Number of Digits in the first phone number allowed list
%KW1		0	
%KW2	PHONE1_SIZE_NO_LOCAL_PREFIX	9	Number of digits in the 1st phone number without any local code nor international code

NOTE: The template example programming code only treats a single entry in the White List. You will need to enhance the programming code to treat more telephone numbers.

Appendix

IMEI Code

You can request the IMEI (International Mobile Equipment Identity) code using the POU: Tools with the Rung – `Retrieve_IMEI_Code`, as shown below.



The `Tools` animation table can be used to request IMEI code, as shown below: