



...Display, Control, Communicate



Tutorial

SMS function

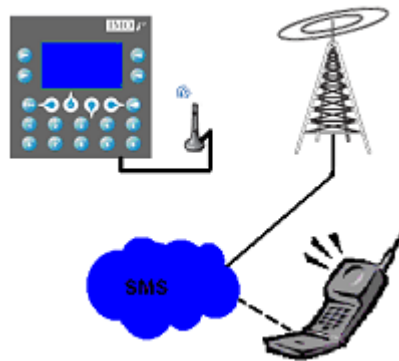
Table of Contents

TABLE OF CONTENTS	2
INTRODUCTION.....	3
INSERTING THE MODEM CARD	4
CONFIGURING THE I³.....	5
SETTING THE SMS CONFIGURATION	6
MODEM CONFIGURATION	7
MESSAGE CONFIGURATION	8
SMS DIRECTORY	8
OUTGOING MESSAGES	9
INCOMING MESSAGES	10
PROGRAMMING EXAMPLE	11
CONFIGURING THE SMS MESSAGE FUNCTION	11
LADDER LOGIC PROGRAMMING	14
SCREEN EDITOR PROGRAMMING.....	14
<i>Screen 1</i>	14
<i>Screen 2</i>	15
<i>Screen 3</i>	15
<i>Screen 4</i>	16
ADVANCED PHONEBOOK FEATURES	17
DESCRIPTION	17
PROGRAMMING	17
<i>Contacts</i>	17
<i>Logic</i>	18
<i>Messages</i>	18
<i>Theory</i>	19
RUNNING THE PROGRAM	21
APPENDICES	22
STATUS REGISTER INFORMATION	22
<i>Internal Modem Initialisation</i>	22
<i>Siemens Modem Initialisation</i>	22
<i>User Specific Initialisation</i>	22

Introduction

This tutorial, will demonstrate how the *i*³ is able to send and receive SMS messages to a users mobile (cell) telephone using the internal GSM modem.

The SMS (Short Message Service) feature allows the *i*³ to send out user-defined strings that give the current status and accept new commands from a defined user remotely via a mobile telephone.



Inserting the modem card

The GSM modem card (i3-M) can be added to any existing *i*³ controller.

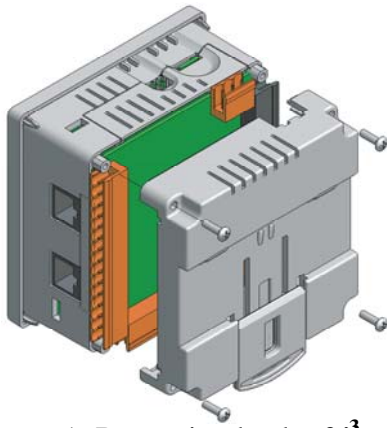


Figure 1: Removing back of *i*³

To insert the i3-M modem card, it is necessary to remove the back cover.

1. Undo the 4 fixing screws and remove the back cover.
2. Insert the i3-M, sliding the connector block onto the connector pins as shown in figure 2.
3. Replace the slim back with the larger *i*³ back cover and reuse the 4 screws, caution not to over tighten.

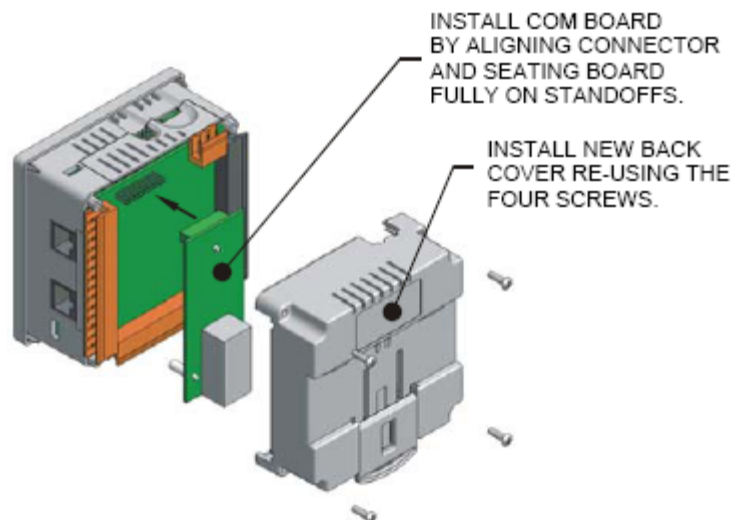



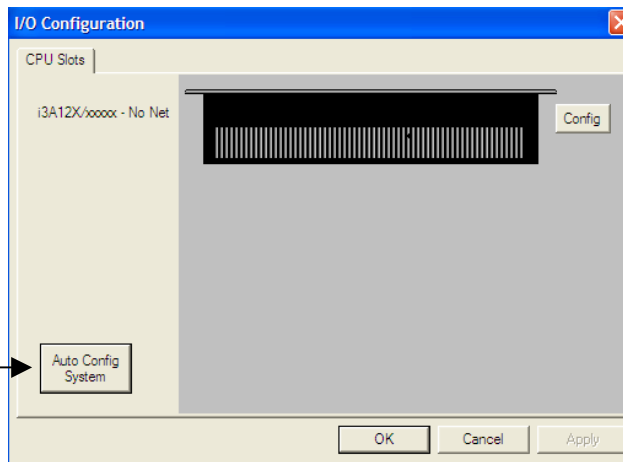
Figure 2: inserting i3-M and replacing back

Configuring the *i*³.

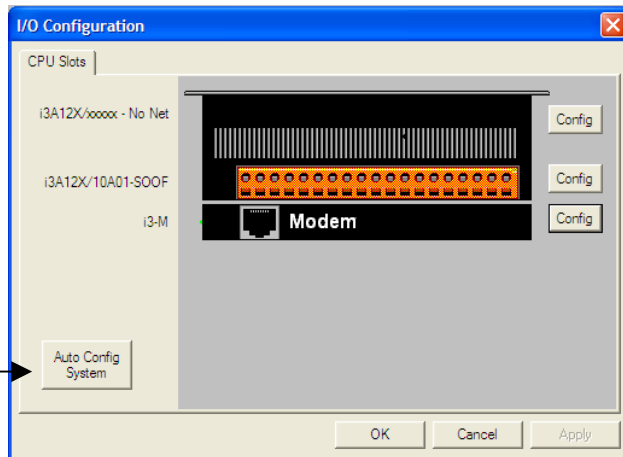
Apply power to the *i*³ controller and connect the programming cable to the port labelled MJ1. Connect the programming cable to your computer and start the i3-Configurator programming software.

A new program is started automatically so it is not necessary to create one. Click on the I/O configure icon  to open the I/O configuration menu.

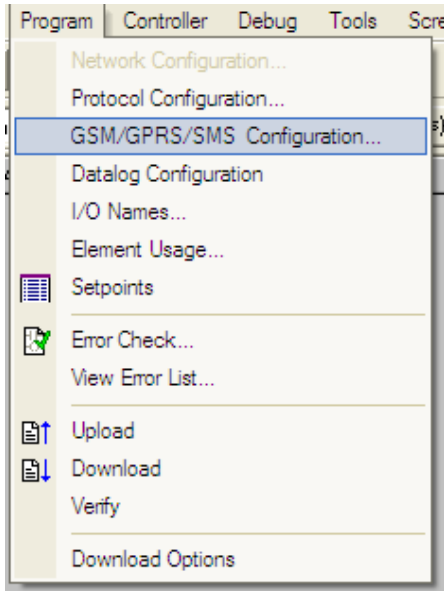
With the *i*³ connected click on the Auto Config System. This will interpret the *i*³ controller model and synchronise the programming software to the *i*³



Now the I/O format and Modem have been recognised click OK to confirm the setup and exit.



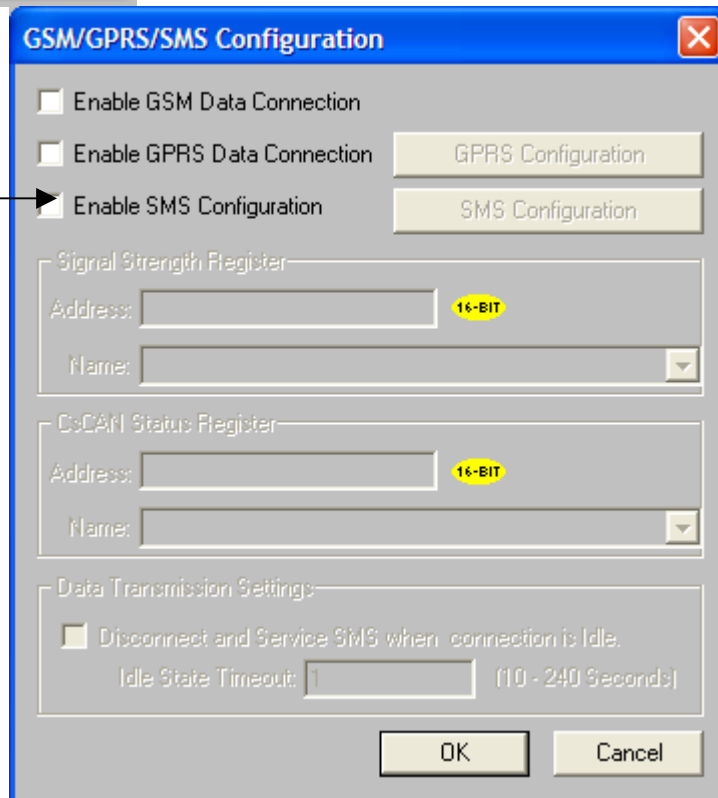
Setting the SMS configuration



The SMS configuration tool can be accessed from the Program drop down menu.

Using this tool we can configure the modem, set up a telephone directory as well as configure incoming and outgoing messages.

Tick this box to enable the SMS configuration.



Modem Configuration

Before we set up any messaging we need to configure the modem used and the registers used to store the status information.

Once the Enable button has been ticked the SMS Configuration button will become active.

Status registers used to display the status of the modem

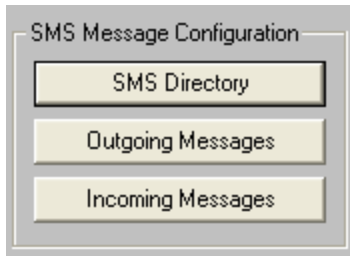
Enter the Centre number and SIM Pin if required.

The *Additional Settings* section accesses the communication parameters Modem initialisation settings. The internal modem requires the settings illustrated below.

Select *Internal Modem Initialisation* and then click ok. Then select the *Com Port Settings* button. Choose GSM QUAD mode as illustrated then click OK.

Message Configuration

Now that we have set up the status registers and configured the modem we can set up the messaging features.



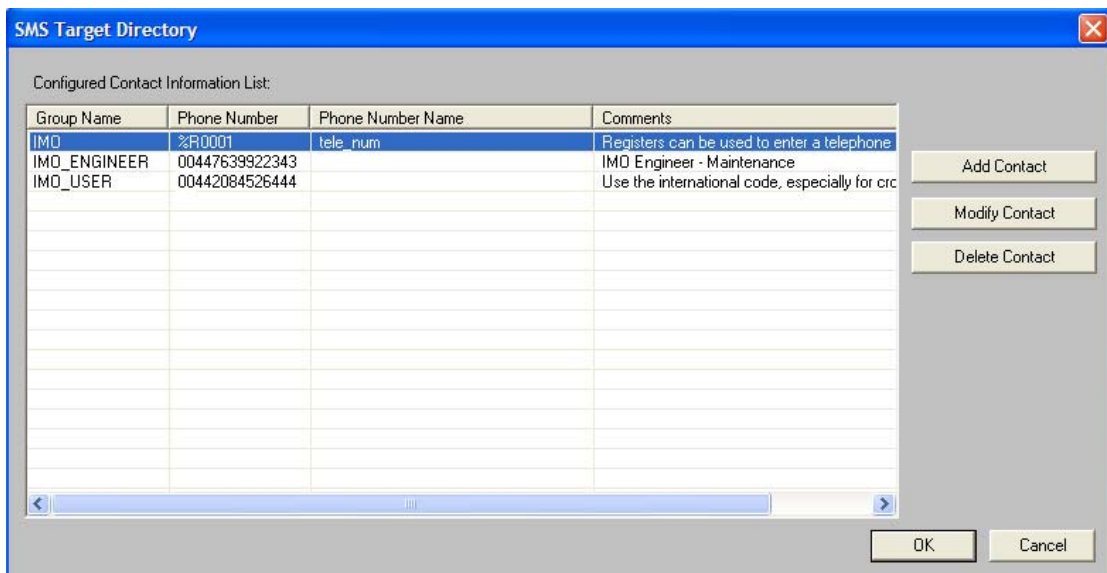
First we have to set up a SMS directory.

NOTE:

Outgoing and Incoming messages buttons cannot be accessed until the Directory is configured.

SMS Directory

Click on the *SMS Directory* button to enter mobile telephone numbers to be used.



Group Name: this is the reference we will use to send and receive messages to and from. Note if the user has not been set up in the directory they cannot send or receive messages with the *i*³ controller.

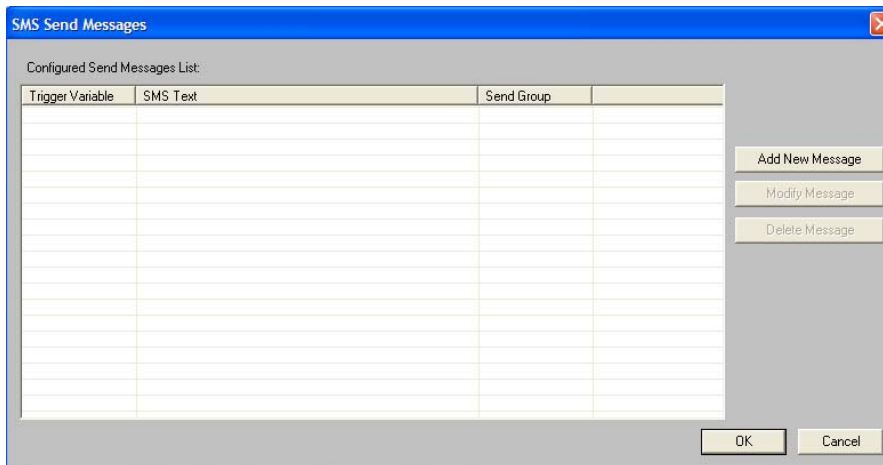
Phone Number: this can either be a register, so a variable number can be entered or it can be a fixed number.

NOTE:

It is advised to use the international dialling code. If a register is used then you can enter a variable name* in **Phone Number Name**. There is also room for some documentation in **Comments**.

*For more information on setting up a variable phone number see the Advanced section of this tutorial.

Outgoing Messages



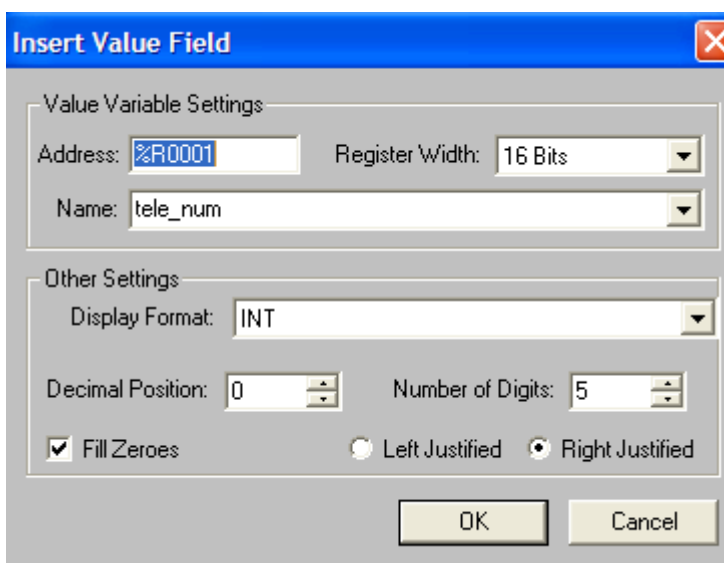
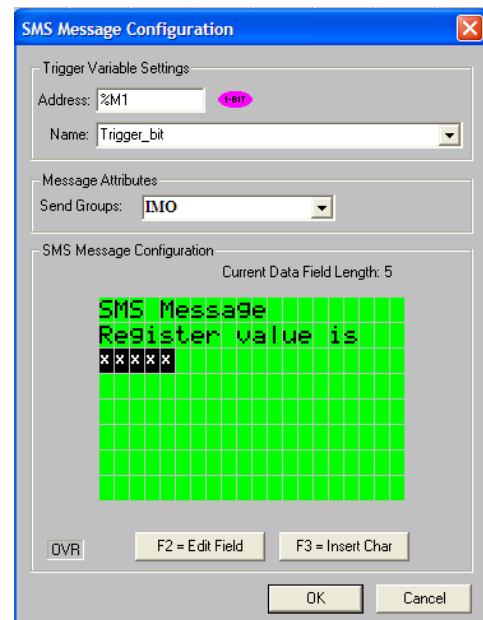
Initially, the outgoing message table will be empty.

To add new messages click on the Add New Message button, up to 32 can be configured.

The trigger variable is the Bit that will initiate send of a SMS message. It must be an individual bit but can be defined as part a bit of a word register, i.e. %R0001.2

The groups previously entered in the SMS directory can now be selected from the *Send Groups* drop down menu. Only one group can be selected per message, however the same trigger could be used to send several messages.

Construct the message using the PC keyboard. Special Characters and Variables from within the controller program can be entered by selecting either the *Edit Field* or *Insert Char* buttons.



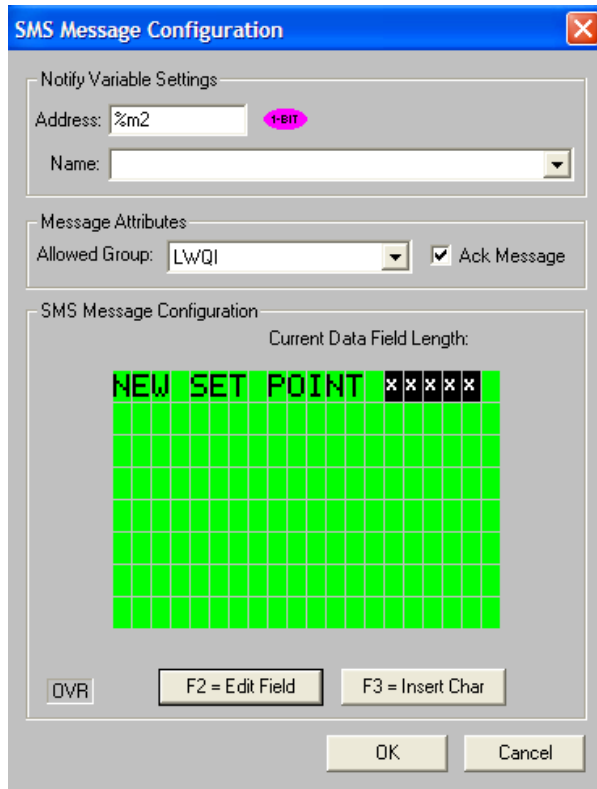
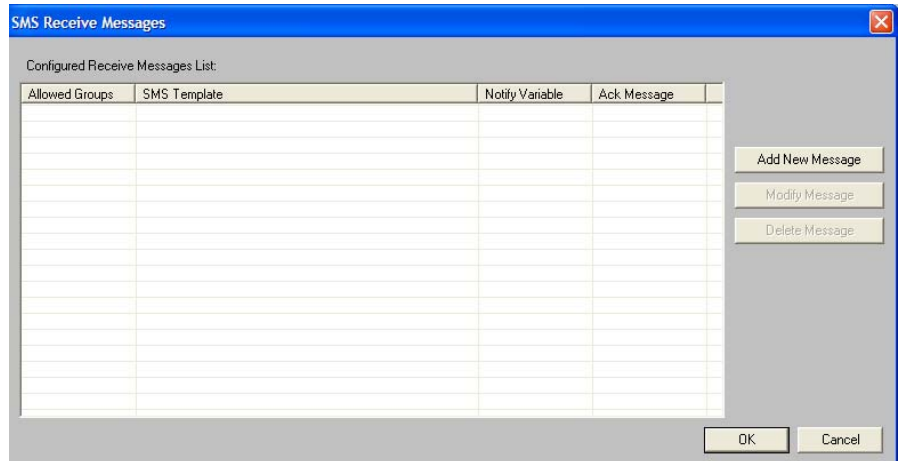
Register values can be entered into the SMS message by clicking the *Edit Field* button or by pressing F2 on your PC keyboard.

A menu appears allowing the selection and configuration of the data register to be included in the message.

Incoming Messages

The *Receive Messages* window is similar to *Send Messages* window.

The *i*³ will only accept the configured string from the configured user and will ignore all other messages that are sent to it.



The **Notify Variable Settings** is a bit that will be operated when the defined message is received. When the message is received the bit will be set to 1. It can be reset from within the controller program.

The **Allowed Group** is the permitted sender of the SMS message. An acknowledgement can be sent.

The text message is made in exactly the same way as before. Received messages are Case Specific that requires the user to send a message that matches the text exactly, case and format of the defined message.

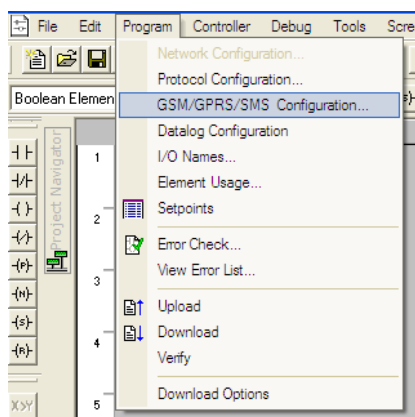
ie. If the message is in capitals, make sure you send the message in capitals.

Programming Example

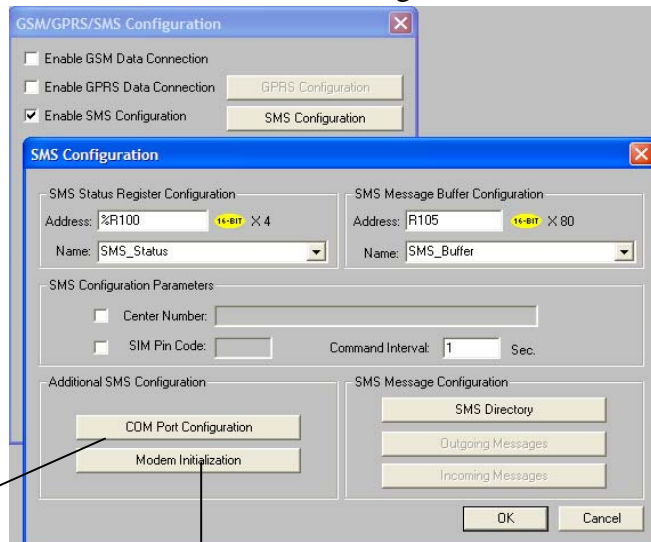
In this example we are going to set up a user page where the user can enter a number and have the i^3 text the value to a mobile number. A user with a mobile number will also be able to text a value to i^3 and change the value in the same register.

Configuring the SMS message Function

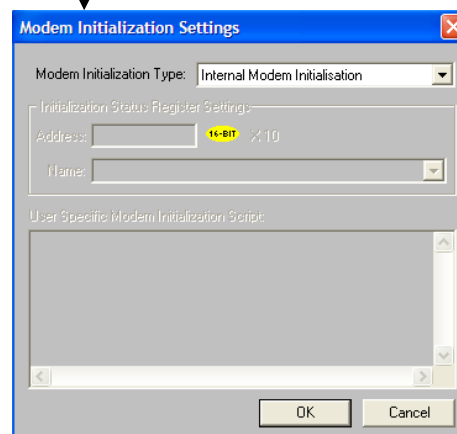
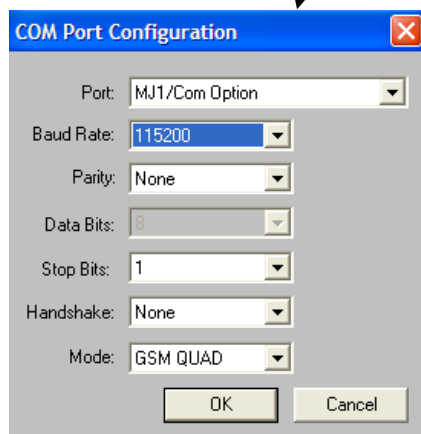
Select the SMS Config option from the Program menu.



Tick the enable SMS Configuration checkbox



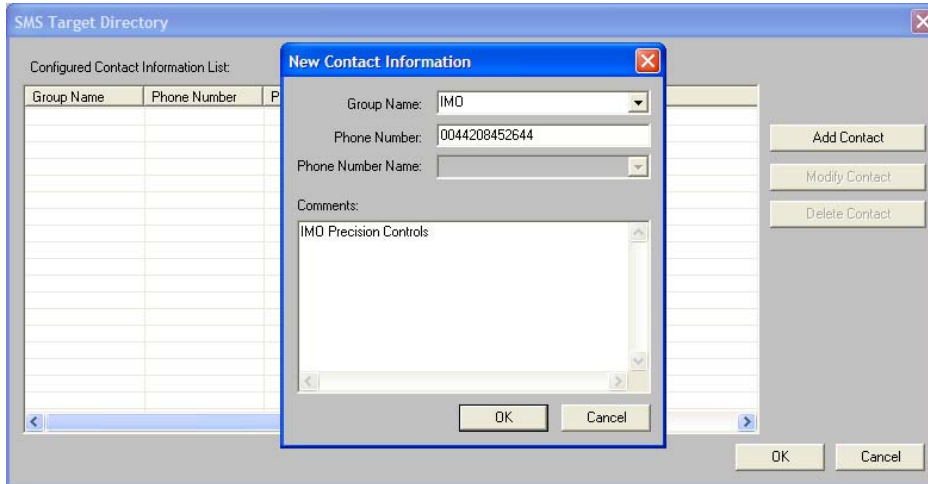
All the features will now become available. Enter the status register start as %R100 and the buffer register start at %R105.



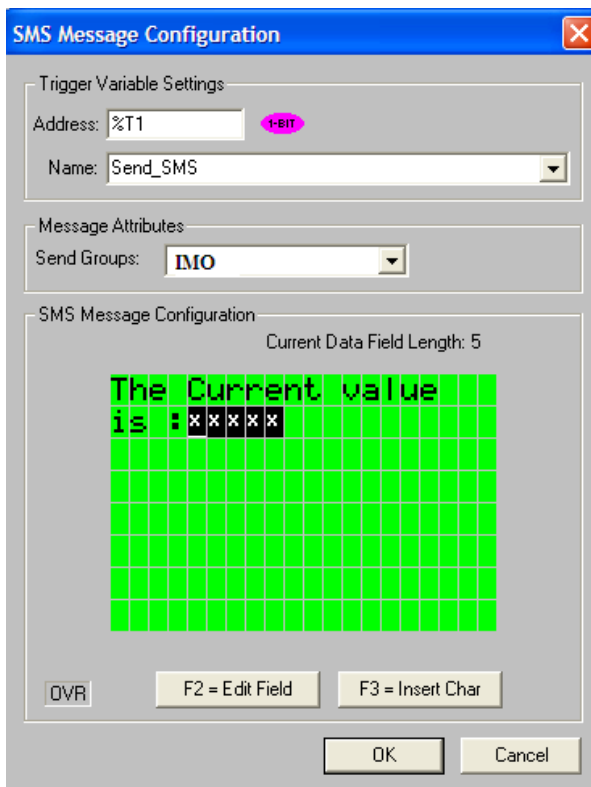
Configure the Modem initialisation to be Internal then set the Com port to be MJ1, Baud: 11520, Parity: None, Stop bits 1, handshake None and Mode GSM QUAD.

SMS via GSM Modem

Now that we have configured the addition settings we can start to configure the messages we want to transmit and receive. First we need to set up the user SMS directory.



Click Add and enter a new Group name. Insert the telephone number to send and receive messages. Note if you want to receive a message the international number needs to be used, i.e. 0044



Once the directory has been set up we can now set up the messages we want to send, and format those we want to receive.

Begin by clicking on the Outgoing Messages button to configure a message.

Set the trigger address to be %T1. Assign the Send Groups to be the group name that has just been configured in the directory.

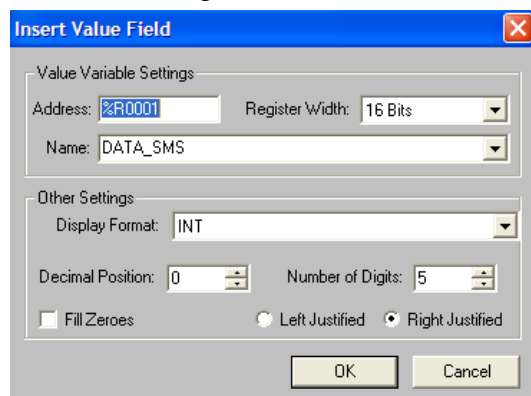
Click on the green screen and start typing the text you wish to send. To insert a register field click the Insert Field button when the cursor is where you want to enter the field.

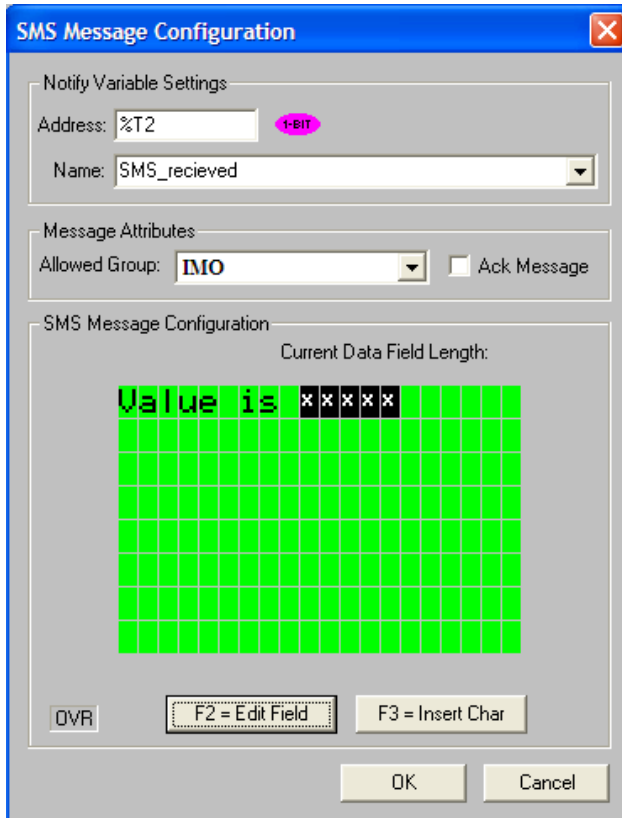
Once you have clicked on the button, a new menu will pop up so that you can enter the registers details.

Enter the address as %R1. Set the display format as INT, no decimal position, and with 5 digits.

Click OK once completed.

Once you are satisfied with the message you are going to send, click OK.



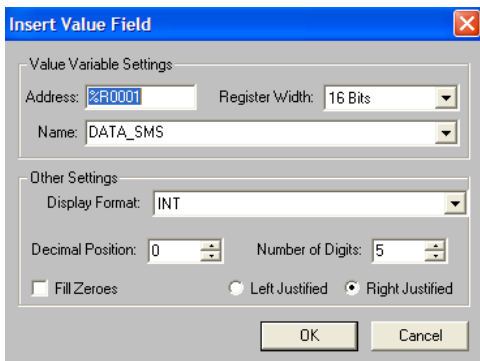


Now we want to set up a message to be received. Click the Incoming Message button.

Set the Notify address to be %T2. This bit will stay on (unless reset using ladder logic) when the exact message set up has been received.

Set the allowed group to be as set up in the directory. This will only accept this message from this user group.

Enter the text that the user will need to send and then enter a register field. The Value field edit box is the same as in the outgoing message. Set the register to be %R1.

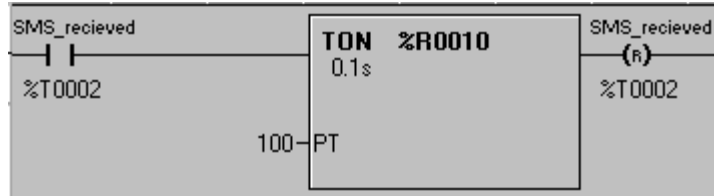


The user will need to enter the text exactly as in the set up and then they can enter a value. This value will then be sent to the register in the i^3 .

The SMS function has now been configured with an Outgoing and an Incoming message. Now we need to program some ladder logic.

Ladder Logic Programming

All that we require to program is the reset function of the message-received indicator. To do this we will use an ON delay timer.

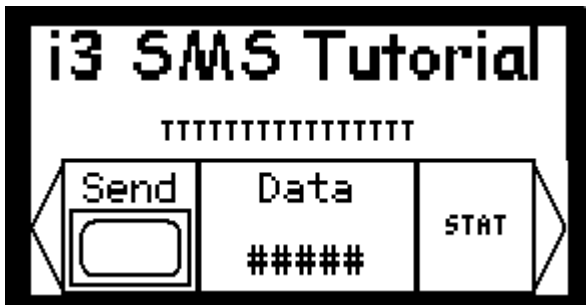


%T2 will be set on when a message is received, after 1s of being on the timer will output and reset %T2.

Screen Editor Programming

We will configure three screens. One will be for user entry, displaying received values and sending values. The other two screens will be for diagnostics use, which are optional but helpful to have, as when using the internal modem in RUN you will not be able to connect to *i*³-configurator.

Screen 1



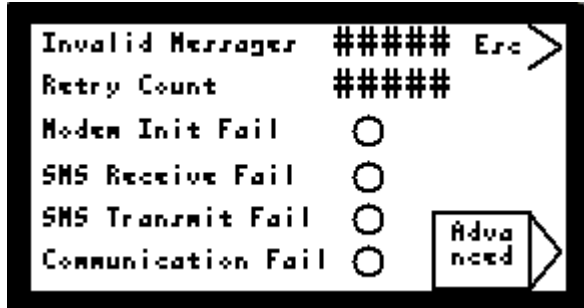
Enter some static text on the screen to display “i3 SMS Tutorial”.

Next set up a Text table function operating off of %T1. For the Value 1 have the text “Sending Message” and for the value 2 “Receiving Message”. Ensure that you will be displaying enough digitals to display the entire message.

Enter a numeric function assigned to %R1; make it editable so that the user can change the variable that is going to be sent. Also, the incoming message will change that value. Finally set up a switch to operate %T1, which will send the message and a screen jump function to jump to Screen 2, the status screen.

Screen 2

Configure four numeric data fields to display two of the status registers, %R102-103. Make sure they are all non-editable and give them appropriate legends. Pick the required Bits from status register %R100 (%R100.13 - %R100.16) and assign them to lamp objects.

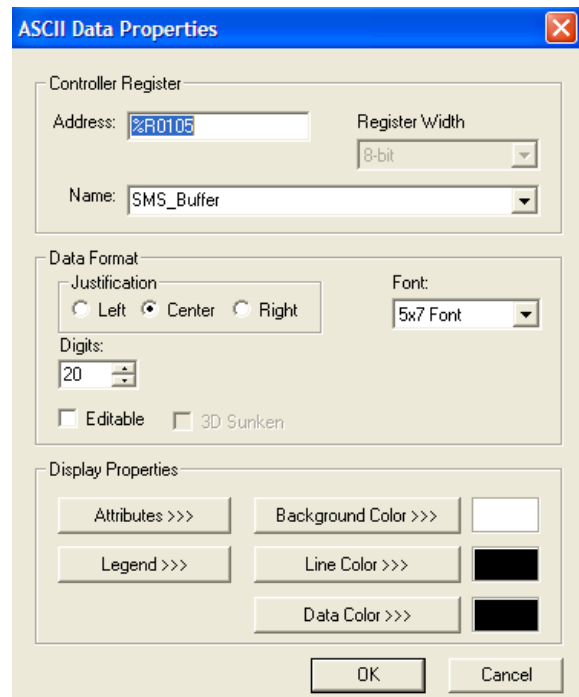
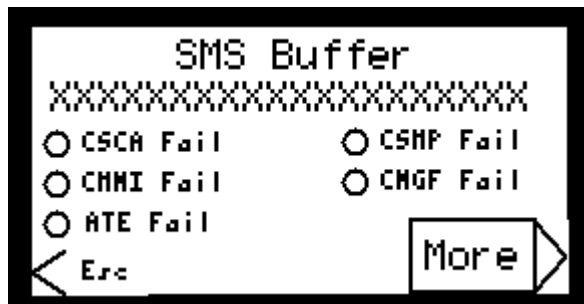


Assign a screen jump button to go to the third screen - Advanced.

And a second screen jump to return to screen 1.

Screen 3

On the advanced screen (screen 3) we require an ASCII function to display the value stored in the message buffer. We can also pick out the final bits of interest from status register %R100 that monitor the initialisation stages of the GSM modem.



Set up the ASCII data properties to show the value that is stored in %R105 and to be showing 20 digits.

Add a screen jump button to return to the main screen (screen 1). Also add an extra button to go to the advanced phonebook feature (screen 4) that will be covered in the final section of this tutorial.

Screen 4

This screen will allow an operator to edit a 10 number telephone directory. All configured SMS messages with the 'VARIABLE' send group will be sent to every number in this directory. This is a useful feature to make the most of the limit of 32 configured messages.

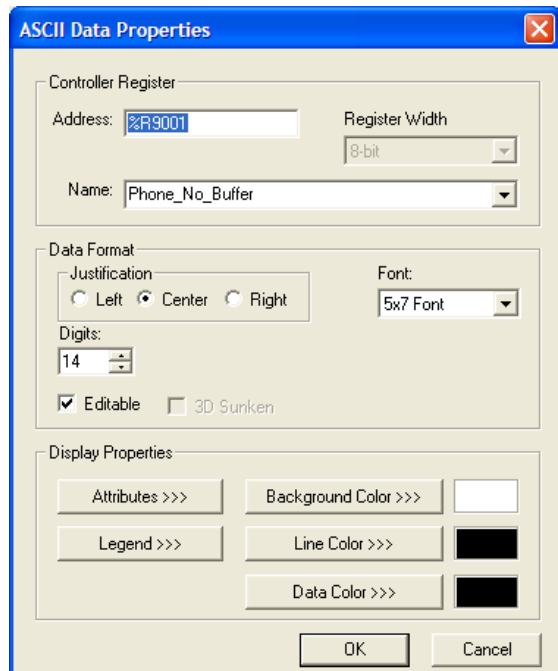
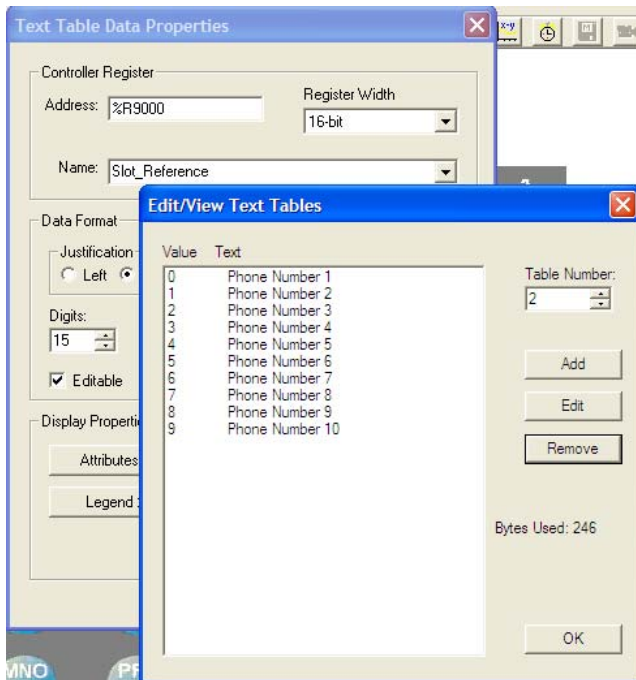
Usually, to send a message to 10 people, ten contact groups will need to be added and then 10 identical SMS messages will need to be configured for each one of the contacts. But with this technique, only one message needs to be configured with one variable contact.



At the top left is a 10 point Text Table with 15 digits.

At the bottom is an ASCII object assigned to %R9001.

There is a Jump to screen 3 soft-key and a Switch object to save the numbers after alteration.



Advanced Phonebook features

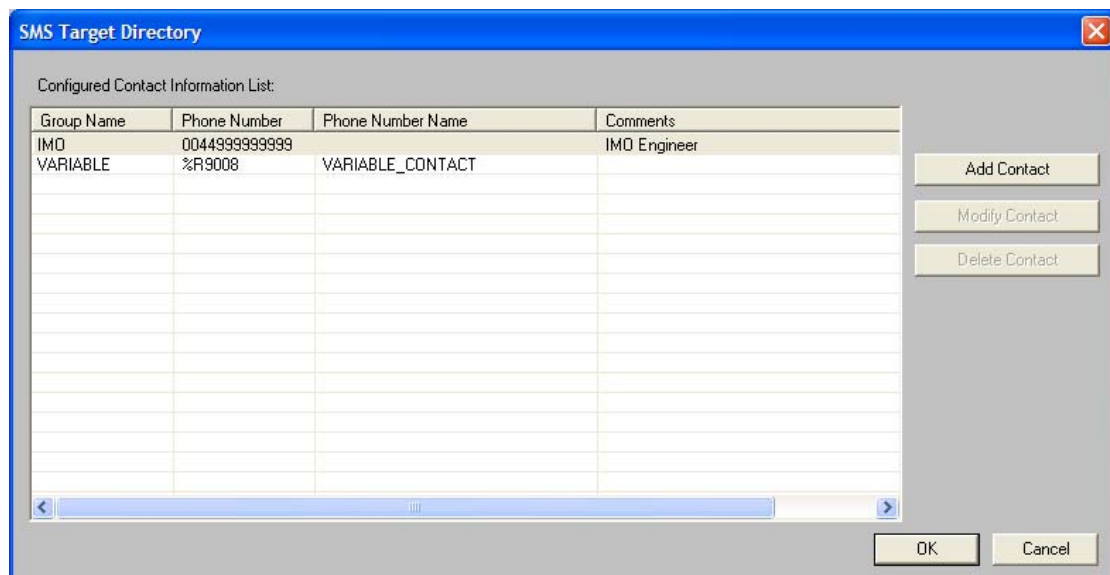
Description

The challenge to resolve with this feature is that the *i*³ scans the list of outgoing messages from top to bottom and the scan pauses to send a message before continuing. Therefore, there is no way of knowing which message is currently being scanned. If the destination phone number is changed without any control there are no guarantees that every contact is receiving every message.

Programming

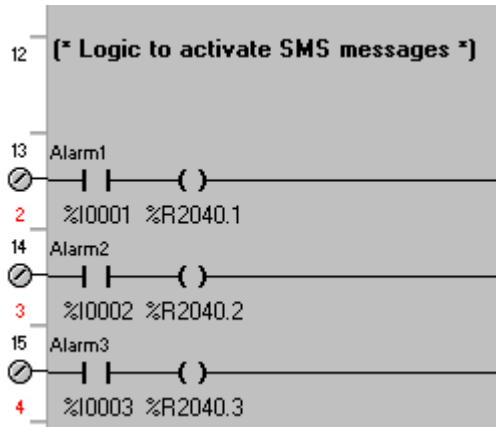
Contacts

Firstly, a variable contact number is added by allocating a register. Use register %R9008; by allocating an array of high registers (%R9000 - %R9170) the feature can then be pasted more easily into any project without interfering with more commonly used registers.



Logic

We can see below, some very basic logic is used to activate our individual SMS messages.

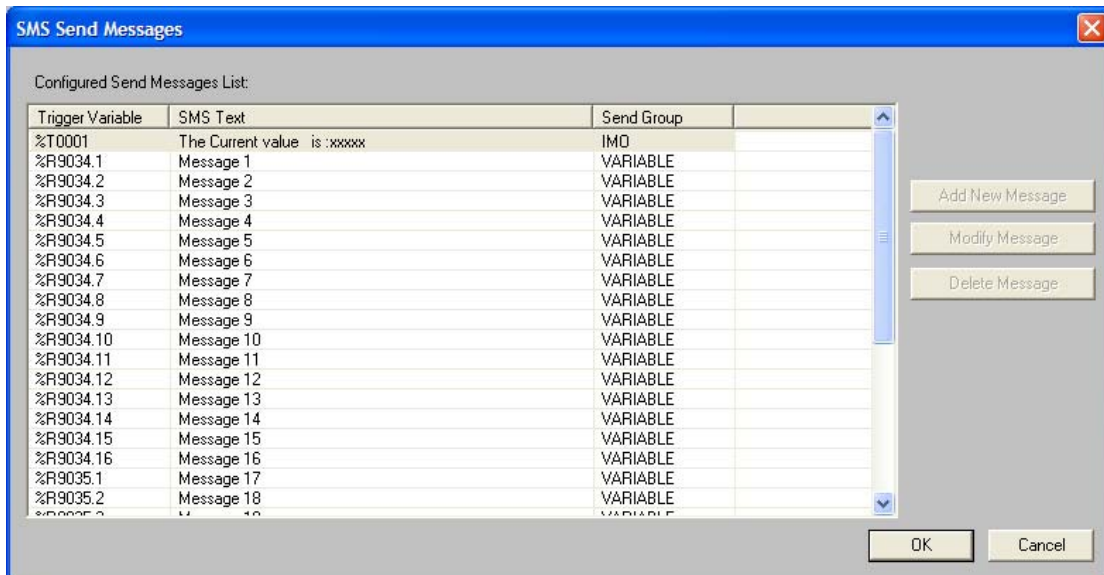


Each Bit of the 32-bit register %R2040 has a separate alarm input %I01 - %I31.

As this is just for demonstration purposes no complex logic is required.

Messages

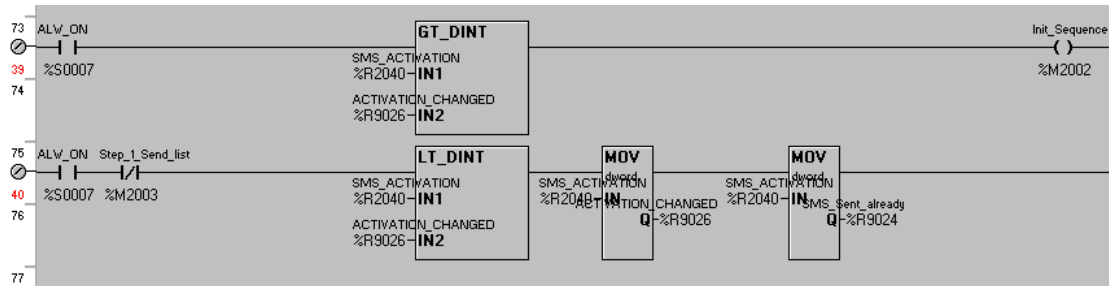
Configure a bank of outgoing messages. As we already have one configured for the basic example; we need to add another 31 SMS messages all to be sent to the variable contact.



The trigger variables are %R9034.1 through to %R9035.15. We will configure %R9034 to be a Double Integer in our program later on. As a Double Integer has 32 bits and the maximum number of messages is also 32; this provides an ideal mechanism to activate each active message in turn.

Theory

Firstly, we need to monitor %R2040 and take action when it changes. If its value increases; then more alarms are active and must be sent. If its value decreases; then alarms are disappearing and no action needs to be taken. This can be achieved as shown below.



Once the sequence is started the active alarm(s) must be found and used to send the corresponding SMS message.

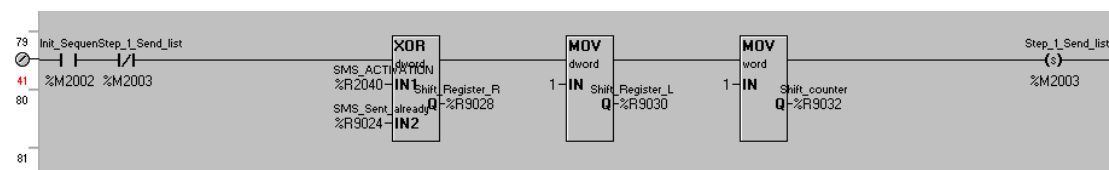
To achieve this, %M2002 initialises the sequence by moving %R2040 into register %R9028 via an XOR function paired with %R9024.

Register %R9024 is there to log any messages that have previously been sent to all contacts. The 'Exclusive OR' function cancels out any messages that have already been sent and only moves the newly activated Bits of %R2040 into %R9028.

For Example:

See the truth table below

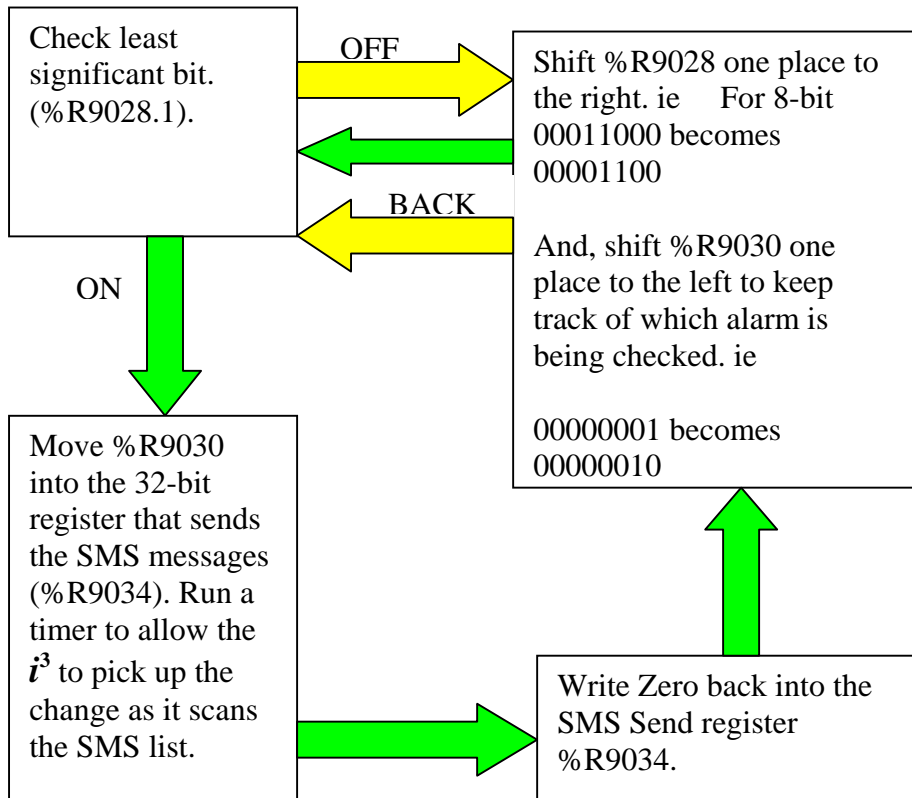
Bit Number of Register	SMS already sent %R9024	ACTIVE ALARMS %R2040	XOR Output %R9028
1	1	1	0
2	0	1	1
3..	1	1	0
32	1	1	0



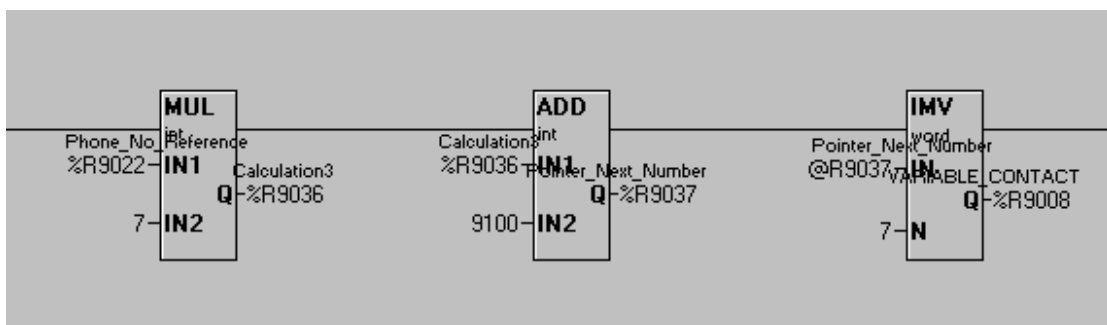
SMS via GSM Modem

For Step 1 (send message list) the easiest way to check which alarm is present within a certain Bit of the register is to always monitor the least significant bit and shift the entire register contents through that bit.


For example the flow of the program should be like this:





Once all relevant Bits of the Double Integer have been checked, and their messages sent; the next phone number in the directory can be moved into the Variable Contact %R9008. The above cycle will then begin again, and repeat for every contact in the phone directory stored in registers %R9100-69.



Running the Program

Now that all the logic has been completed we can download the program to the *i*³. Before downloading please ensure that the *i*³ is in idle mode, the traffic lights should be at RED. Click on the download icon  to download the program. This will download all the ladder logic, screens and SMS function.

Once you have downloaded you can then change the *i*³ into RUN mode by pressing on the GREEN traffic light. 

Please use the program: sms_tutorial.csp 

Appendices

Status Register Information

Status Register – 4 words – 64 bits

Status	Signal Strength	Word for Invalid Receive messages	Word for retry count
1 st word	2 nd word	3 rd word	4 th word

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Internal Modem Initialisation															
CSMP Command Failed	CSCA Service centre number setting failed	Not used	CMGF Command Failed	CNMI Command Failed	Not used	Not used	Not used	ATE command Failed	Not used	Not used	Not used	SMS Receive Failed	Modem Initialization Failed	SMS Transmit Failed	Communication Failure
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Siemens Modem Initialisation															
CPIN Command failed	CSCA service centre number failed	CReg Command failed	CMGF Command Failed	CNMI Command Failed	CSQ Command failed	COPS Command failed	CPEE Command failed	ATE Command Failed	Not used	Not used	Not used	SMS Receive Failed	Modem Initialization Failed	SMS Transmit Failed	Communication Failure
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
User Specific Initialisation															
Bit 0 – Bit 11 will be set for every failed command												SMS Receive Failed	Modem Initialization Failed	SMS Transmit Failed	Communication Failure



IMO Precision Controls Limited
 1000 North Circular Road
 Staples Corner, London
 NW2 7JP United Kingdom
 Tel: +44 (0)20 8452 6444
 Fax: +44 (0)20 8450 2274
 Email: imo@imopc.com
 Web: www.imopc.com



IMO Jeambrun Automation SAS
 165 Rue Jean Jaures,
 94700 Maisons Alfort
 Paris, France
 Tel: +33 (0)1 45 13 47 05
 Fax: +33 (0)1 45 13 47 37
 Email: info@imopc.fr
 Web: www.imopc.fr



IMO Deutschland
 Für weitere Einzelheiten
 zu IMO Agenten und
 Distributoren in Ihrer Nähe
 schreiben. Sie bitte ein E-mail
 an folgende Adresse:
 imo@imopc.com



IMO Italia
 Viale A. Volta 127/a
 50131 Firenze, Italia
 Tel: +39 800 783281
 Fax: +39 800 783282
 Email: info@imopc.it
 Web: www.imopc.it



IMO Canada
 Unit 10, Whitmore Road
 Woodbridge, Ontario.
 L4L 8G4 Canada
 Tel: +1 905 265 9844
 Fax: +1 905 265 1749
 Email: imocanada@imopc.com



- Cam Switches
- Din Terminals
- Drives
- Enclosures
- Fieldbus remote I/O
- Isolators & Switch Fuses
- MCB & RCD
- Motor Circuit Breakers
- Motor Control Gear
- Panel Meters
- Relays
- Signal Conditioning
- Sockets
- Timers
- Transformers & Power Supplies



- Drives
- Intelligent Terminals/HMI
- Limit Switches
- Photoelectric Switches
- PLCs
- Proximity Switches
- Temperature Controls



- Data Acquisition & Control
- Drives
- Intelligent Terminals/HMI
- Limit Switches
- Photoelectric Switches
- Proximity Switches
- PLCs
- Signal Conditioning
- Temperature Controls



- Lightguards
- Safety Limit Switches
- Safety Relays



- Jaguar VXM 0.37-500KW
- Jaguar VXSM 0.37-7.5KW
- Jaguar CUB 0.37-2.2KW



- Audible devices
- Chip-on-Board
- Device programmers
- LEDs & 7 seg. displays
- PCB Terminal blocks
- Relays - automotive
- Relays - power
- Relays - signal
- Switches

All IMO products are tried, tested and approved to relevant international quality standards

